



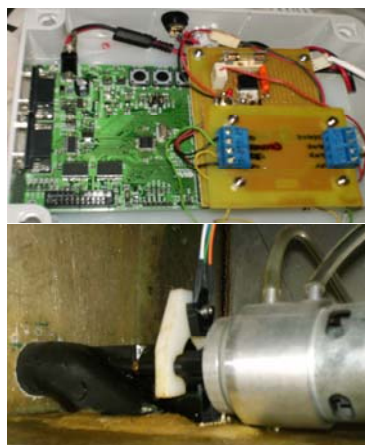
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ



ΣΧΟΛΗ ΝΑΥΠΗΓΩΝ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΠΕΡΙΟΧΗ ΝΑΥΤΙΚΗΣ ΜΗΧΑΝΟΛΟΓΙΑΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ : «ΑΝΑΠΤΥΞΗ ΚΛΙΝΗΣ ΔΟΚΙΜΩΝ ΜΕ ΣΚΟΠΟ ΤΗ
ΜΕΤΡΗΣΗ ΩΣΗΣ ΠΡΟΩΘΗΤΗΡΑ ΑΝΑΔΡΑΣΗΣ»



ΚΑΡΔΑΣΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΕΠΙΒΛΕΠΩΝ : ΙΩΑΝΝΗΣ ΠΡΟΥΣΑΛΙΔΗΣ

ΑΘΗΝΑ ΑΠΡΙΛΙΟΣ 2008

Η διπλωματική αυτή εργασία, ξεκίνησε το 2006 και ολοκληρώθηκε περί τα τέλη Απρίλη του 2008.

Με την εργασία αυτή μου δόθηκε η δυνατότητα να γνωρίσω το ιδιαίτερα ενδιαφέρον αντικείμενο της κυβερνητικής του προγραμματισμού τόσο υπολογιστών όσο και μικροεπεξεργαστών. Επιπλέον ασχολήθηκα με τη σχεδίαση ηλεκτρονικών κυκλωμάτων και εμπλούτισα τις γνώσεις μου σε αυτόν τον τομέα. Για το λόγο αυτό θα ήθελα να ευχαριστήσω τον κ. Νικόλαο Ξηρό που μου έδωσε την ευκαιρία να εκπονήσω αυτή την εργασία και τον κύριο Προυσαλίδη που δέχθηκε να αναλάβει και να ολοκληρώσει την εργασία αυτή. Η βοήθεια του είναι πολύτιμη γιατί ανέλαβε μια διπλωματική που ήταν εκτός των πεδίων ενδιαφέροντός του αλλά δεν αρνήθηκε οποιαδήποτε βοήθεια. Επίσης ευχαριστώ τους υποψήφιο διδάκτορα κ. Ελευθέριο Λόγη και τον Καθ. Πληροφορικής Ιάκωβο Μανωλά για όλη την πολύτιμη βοήθεια που μου προσέφεραν και η οποία ήταν καθοριστική.

Τέλος, ευχαριστώ εκ προοιμίου, τον κ. Γεωργίου και τον κ. Πολίτη για την παρουσία τους στην εξέταση και την παρουσίαση της εργασίας αυτής. Αφιερώνω την εργασία αυτή στους ανθρώπους που με στήριξαν ουσιαστικά με πολλούς τρόπους καθ' όλη την διάρκεια των σπουδών μου. Ιδιαίτερα στην οικογένεια μου που με ανέχθηκαν και στήριξαν.

Σύνοψη

Στην παρούσα πτυχιακή πραγματοποιήθηκε η σχεδίαση και ανάπτυξη μιας ολοκληρωμένης διάταξης μετρήσεων ώσης προωθητήρα ανάδρασης υπό κλίμακα. Παράλληλα αναπτύχθηκε το θεωρητικό υπόβαθρο των μετρούμενων μεγεθών ενώ στα πλαίσια των εργαστηριακών μετρήσεων απαιτήθηκε η ανάπτυξη ελεγκτή στροφών για τον ηλεκτρικό κινητήρα και κατ' επέκταση του προωθητήρα ανάδρασης. Επιπλέον αναπτύχθηκε η θεωρία για τη μέτρηση της ροπής αδρανείας του συστήματος πρόωσης. Η παραπάνω μελέτη οδήγησε στην ανάπτυξη μιας εύκολης μεθοδολογίας μετρήσεως της ροπής αδρανείας γραμμικών συστημάτων με την οποία δεν χρειάζονται ιδιαίτερα ακριβείς μετρήσεις της ταχύτητας ή της ροπής συναρτήσεως του χρόνου. Η ανάπτυξη του αυτοματοποιημένου συστήματος μετρήσεων απαίτησε την γραφή προγράμματος σε υπολογιστή υπό λειτουργικό περιβάλλον Windows. Ο υπολογιστής δρα ως ελεγκτής όλης της εγκατάστασης μετρήσεων ενώ για την εκτέλεση των εντολών του υπολογιστή και υλοποίησης των μετρήσεων χρησιμοποιείται μια πρότυπη ηλεκτρονική κάρτα η οποία φέρει ένα ισχυρό μικροεπεξεργαστή. Επίσης γράφηκε κώδικας για τη λειτουργία της κάρτας ώστε να συντονιστεί με το πρόγραμμα στον υπολογιστή και για να ανταποκρίνεται στις εντολές του.

Καταρχήν πραγματοποιείται μια παρουσίαση των βασικών διατάξεων της πειραματικής διάταξης των μετρήσεων ενώ μοντελοποιείται ο ηλεκτρικός κινητήρας. Στη συνέχεια αναπτύσσεται η βασική θεωρία του ελεγκτή στροφών τριών όρων «P.I.D» και εν συνεχεία μεταφέρονται οι βασικές συναρτήσεις του ελεγκτή στο διακριτό χρόνο ώστε να υλοποιηθούν στον μικροεπεξεργαστή. Στη συνέχεια παρουσιάζεται η κάρτα LPC2106 και το σκεπτικό των επιλεγμένων ρυθμίσεων ώστε να επιτευχθεί η βέλτιστη χρήση της. Οι ρυθμίσεις αυτές είναι πολύ σημαντικές για τη ροή του προγράμματος καθώς σχεδιάσθηκε έτσι ώστε η ροή του να ελέγχεται από γεγονότα, καθιστώντας το πολύ ευέλικτο. Στο επόμενο κεφάλαιο παρουσιάζονται τα υλοποιημένα πρωτόκολλα επικοινωνίας μεταξύ της κάρτας του υπολογιστή και του «Load Cell». Επίσης, παρουσιάζονται ορισμένοι βασικοί αλγόριθμοι που πραγματοποιούνται στον επεξεργαστή της κάρτας όπως της μέτρησης των στροφών και μετατροπής των δεδομένων πριν την αποστολή. Εν' συνεχεία παρουσιάζεται η εφαρμογή η οποία γράφηκε για τον υπολογιστή και χρησιμοποιήθηκε για την υλοποίηση των μετρήσεων.

Τέλος, η πτυχιακή κλείνει με τα δύο τελευταία κεφάλαια στα οποία πραγματοποιούνται οι μετρήσεις της αδρανείας και της Ώσης. Σε αυτά παρουσιάζεται όλο το θεωρητικό υπόβαθρο των μετρήσεων όπως η θεωρία για τον υπολογισμό της ροπής αδρανείας. Για κάθε μέτρηση γράφονται όλες οι παράμετροι εκτέλεσης των πειραμάτων και υπολογίζεται η αδράνεια και η καμπύλη ώσης συναρτήσεως των στροφών. Τα κεφάλαια κλείνουν με το σχολιασμό των αποτελεσμάτων.

ΠΕΡΙΕΧΟΜΕΝΑ ΚΕΦΑΛΑΙΩΝ

ΠΕΡΙΕΧΟΜΕΝΑ-----	3
ΕΙΣΑΓΩΓΗ-----	9

ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ

Η Εγκατάσταση μετρήσεων και το Σύστημα Προώσεως. -----	9
1.1 Περιγραφή Εργαστηριακής Εγκατάστασης -----	9
1.2 Κινητήρας Συνεχούς Ρεύματος Μονίμων Μαγνητών -----	12
1.3 Μοντελοποίηση Κινητήρα συνεχούς τάσης -----	13
1.4 Σύστημα τροφοδοσίας της εγκατάστασης. -----	18
1.4.1 Μετασχηματιστής Τάσης -----	18
1.4.2 Μονοφασικός Ανορθωτής Πλήρους Κύματος -----	19
1.4.3 Φίλτρο Πυκνωτή -----	20
1.5 Παλμοτροφοδοτικός Μετατροπέας Τάσης -----	20
1.6 Περιγραφή του υπέρυθρου φωτοανιχνευτή-----	25
1.7 Περιγραφή Φίλτρου σήματος από τον υπέρυθρο φώτο-ανιχνευτή -----	26

ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ

Κατασκευή Ρυθμιστή Στροφών Ηλεκτρικού κινητήρα -----	31
2.1 Αναγκαιότητα ρυθμιστή-----	31
2.2 Υπολογισμός της συνάρτησης μεταφοράς του συστήματος προωθητήρα ανάδρασης και ηλεκτρικού κινητήρα-----	33
2.3 Γενικές Σημειώσεις επί της ανάλυσης του συνολικού συστήματος -----	36
2.4 Γενικές Σημειώσεις επί της ανάλυσης του συνολικού συστήματος -----	37
2.5 Εφαρμογή αναλογικού ελεγκτή "P" στην συνάρτηση μεταφοράς του συστήματος -----	38
2.5.1 Υπολογισμός της συνάρτησης μεταφοράς του συστήματος ρύθμισης στροφών "P" όταν στο σύστημα εφαρμόζεται διαταραχή -----	38
2.5.2 Υπολογισμός της συνάρτησης μεταφοράς του συστήματος ρύθμισης στροφών "P" όταν στο σύστημα εφαρμόζεται μεταβολή στις επιθυμητές στροφές-----	49
2.6 Εφαρμογή αναλογικού ολοκληρωτικού ελεγκτή "PI" στην συνάρτηση μεταφοράς του συστήματος -----	40
2.6.1 Υπολογισμός της συνάρτησης μεταφοράς του συστήματος ρύθμισης στροφών "PI" όταν στο σύστημα εφαρμόζεται διαταραχή -----	40
2.6.2 Υπολογισμός της συνάρτησης μεταφοράς του συστήματος ρύθμισης στροφών "PI" όταν στο σύστημα εφαρμόζεται μεταβολή στις επιθυμητές στροφές-----	41
2.7 Εφαρμογή αναλογικού ολοκληρωτικού διαφορικού ελεγκτή "PID" στην συνάρτηση μεταφοράς του συστήματος -----	42
2.7.1 Υπολογισμός της συνάρτησης μεταφοράς του συστήματος ρύθμισης στροφών "PID" όταν στο σύστημα εφαρμόζεται διαταραχή-----	42
2.7.2 Υπολογισμός της συνάρτησης μεταφοράς του συστήματος ρύθμισης στροφών "PID" όταν στο σύστημα εφαρμόζεται μεταβολή στις επιθυμητές στροφές-----	43

2.8 Υπολογισμός του μονίμου σφάλματος του αναλογικού ρυθμιστή στροφών "P" -----	44
2.9 Υπολογισμός του μονίμου σφάλματος του αναλογικού και ολοκληρωτικού ρυθμιστή στροφών "PI"-----	46
2.10 Υπολογισμός του μονίμου σφάλματος του αναλογικού ολοκληρωτικού και διαφορικού ρυθμιστή στροφών "PID"-----	48
2.11 Μεταφορά των νόμων ελέγχου στον ψηφιακό επεξεργαστή -----	50
2.11.1 Επεξεργασία Διακριτών μεταβλητών-----	50
2.11.2 Θεωρία του μετασχηματισμού Z-----	51
2.12 Διακριτοποίηση του αναλογικού ελεγκτή "P" -----	52
2.13 Διακριτοποίηση του αναλογικού και ολοκληρωτικού ελεγκτή "PI" -----	52
2.14 Διακριτοποίηση του ελεγκτή τριών όρων "PID" -----	57

ΚΕΦΑΛΑΙΟ ΤΡΙΤΟ

Κεφάλαιο Αρχιτεκτονική και συνδεσμολογία κάρτας IAR -----	61
3.0 Εισαγωγή-----	61
3.1 Παρουσίαση των δυνατοτήτων και των χαρακτηριστικών της LPC2106 --	62
3.2 Επεξεργαστής ARM7TDMI-S -----	64
3.3 Χρονισμός επεξεργαστή -----	65
3.4 Χρονισμός ρολογιού περιφερειακών συσκευών και λειτουργιών επεξεργαστή (pclk)-----	67
3.5 Περιγραφή της λειτουργίας των διακοπών ροής του προγράμματος -----	67
3.6 Περιγραφή των καταχωρητών χρήσης του Ελεγκτή διακοπών VIC -----	69
3.7 Κατανομή των περιφερειακών συσκευών στον ελεγκτή διακοπών και ορισμός των προτεραιοτήτων -----	71
3.8 Περιγραφή σειριακής Θύρας UART -----	74
3.9 Περιγραφή των περιφερειακών Χρονομετρητών -----	78
3.9.1 Χρονισμός των ρολογιών Timer0 & Timer1 -----	79
3.9.2 Διαμόρφωση των λειτουργιών των χρονομετρητών -----	81
3.10 Περιγραφή λειτουργίας παλμού PWM (Pulse Width Modulator)-----	84
3.10.1 Χρονισμός του ρολογιού δημιουργίας του παλμού -----	84
3.11 Περιγραφή System Control External Interrupt 1 -----	85
3.12 Περιγραφή χρήση των ακίδων εισόδου εξόδου του επεξεργαστή «GPIO»	85
3.13 Περιβάλλον γραφής του προγράμματος της LPC2106 -----	87

ΚΕΦΑΛΑΙΟ ΤΕΤΑΡΤΟ

Κεφάλαιο Αλγόριθμοι μετρήσεων και εντολών των Προγραμμάτων-	91
4.0 Σύνοψη -----	91
4.1 Πρωτόκολλο σειριακής μετάδοσης δεδομένων από την LPC2106 στον υπολογιστή-----	92
4.2 Πρωτόκολλο σειριακής μετάδοσης από τον υπολογιστή στην LPC2106 --	99
4.3 Πρωτόκολλο λήψης της μέτρησης ώσης στην LPC2106 -----	101
4.4 Εντολές ελέγχου της LPC2106-----	102
4.5 Ο αλγόριθμος της μέτρησης και της σύνθεσης του χρόνου κατά την πλήρη λειτουργία της LPC2106 -----	104
4.6 Ο αλγόριθμος σύνθεσης του byte ελέγχου -----	106
4.7 Ο αλγόριθμος χρήσης του αρχείου των σημείων αναφοράς-----	107
4.8 Μέτρηση Στροφών στην Κάρτα Philips -----	109

ΚΕΦΑΛΑΙΟ ΠΕΜΠΤΟ

Η γλώσσα προγραμματισμού V.B και η εφαρμογή ελέγχου επί του υπολογιστή	115
5.0 Σύνοψη	115
5.1 Αντικειμενοστραφής προγραμματισμός	115
5.2 Αντικείμενα	116
5.3 Κλάσεις	116
5.4 Ιδιότητες	117
5.5 Μέθοδοι	117
5.6 Οδηγούμενος από γεγονότα προγραμματισμός	118
5.7 Διαδικασίες	118
5.8 Ροή εκτέλεσης εφαρμογής	120
5.9 Υλοποίηση εφαρμογών σε σύγχρονο προγραμματιστικό περιβάλλον	120
5.10 Στοιχεία γραφικού προγραμματιστικού περιβάλλοντος	121
5.10.1 Μενού επιλογών	122
5.10.2 Πλαίσια διαλόγου	123
5.11 Επικοινωνία με άλλες εφαρμογές	123
5.12 Χαρακτηριστικά αντικειμενοστραφών περιβαλλόντων ανάπτυξης εφαρμογών	125
5.13 Κριτήρια επιλογής της χρήση για την γραφή του προγράμματος της γλώσσας Visual Basic	126
5.14 Λίγα λόγια για την Ιστορία της Visual Basic	127
5.14.1 Το προγραμματιστικό περιβάλλον του V.Studio 2005	127
5.15 Βασικές Απαιτήσεις της εφαρμογής Visual Basic του υπολογιστή	129
5.16 Κεντρική Φόρμα Διαχείρισης της εφαρμογής	130
5.16.1 Επιλογή και χρήση υπάρχον αρχείου ρυθμίσεων	131
5.16.2 Δημιουργία νέου ή τροποποίηση αρχείου ρυθμίσεων	131
5.16.3 Εφαρμογή δημιουργίας αρχείων αποστολή σημείων αναφοράς της εφαρμογή Κλίνη Ώσης και Ελέγχου στροφών	134
5.16.4 Εισαγωγή τιμών από αρχείο κειμένου	138
5.16.5 Τυποποίηση του αρχείου αποστολής	140
5.16.6 Εφαρμογή Κλίνης Ώσης και Ελέγχου Στροφών	142
5.16.7 Αυτοματοποιημένες διαδικασίες και σημαντικές εργασίες	143
5.16.7.1 Αλγόριθμος επικοινωνίας μέσω της σειριακής θύρας με την LPC2106	143
5.16.7.2 Ανταπόκριση της εφαρμογής στο byte ελέγχου	144
5.16.7.3 Αποστολή των επιθυμητών σημείων αναφοράς από την μπάρα εισαγωγής	144
5.16.7.4 Περιοδική επανάληψη κρίσιμων διαδικασιών	145
5.17 Παρουσίαση της φόρμας της εφαρμογής Κλίνη Ώσης και ελέγχου στροφών	145
5.18 Αλλαγή καταστάσεως λειτουργίας της Κλίνης Ώσης και της LPC2106	151
5.19 Άμεση διακοπή λειτουργίας Κινητήρα	151
5.20 Αποστολή τιμών αναφοράς από αρχείο	152
5.21 Διαμόρφωση του αρχείου αποθήκευσης των εισερχόμενων δεδομένων από την LPC2106	153
5.22 Κρίσιμα μηνύματα της εφαρμογής Κλίνης Ώσης και ο χειρισμός τους	154

ΚΕΦΑΛΑΙΟ ΕΚΤΟ

Θεωρία και Υπολογισμός της Ροπής Αδρανείας του συστήματος

πρόωσης -----	157
6.0 Εισαγωγή-----	157
6.1 Υπολογισμός της ροπής αδρανείας με τη χρήση της αρχή διατήρησης της ορμής -----	157
6.2 Υπολογισμός Ροπής αδρανείας με την μέθοδο της μεταβαλλόμενης ροπής του κινητήρα-----	158
6.2.1 Εύρεση της ροπής αδρανείας όταν στο σύστημα ασκείται φορτίο -----	159
6.2.2 Εύρεση της ροπής αδρανείας όταν στο σύστημα δεν ασκείται φορτίο --	161
6.2.3 Εύρεση της ροπής αδρανείας με ημιτονοειδή μεταβολή των στροφών του κινητήρα, όταν στο σύστημα δεν ασκείται φορτίο -----	161
6.3 Σχόλια επί της πειραματικής διάταξης και τις παραμέτρους της διεξαγωγής των μετρήσεων-----	168
6.4 Εφαρμογή του Παλμοτροφοδοτικού μετατροπέα στον ηλεκτρικό κινητήρα και προβλήματα κατά την εκτέλεση του πειράματος μέτρησης της ροπής αδρανείας-----	168
6.5 Νέος αναλογικός Παλμοτροφοδοτικός μετατροπέας τάσης-----	170
6.6 Συνεργασία παλμοτροφοδοτικού μετατροπέα και συστήματος πρόωσης -	174
6.7 Διεξαγωγή μετρήσεων και αποτελέσματα -----	177

ΚΕΦΑΛΑΙΟ ΕΒΔΟΜΟ

Μέτρηση ώσεος προωθητήρα ανάδρασης -----	181
7.0 ΣΥΝΟΨΗ -----	181
7.1 Περιγραφή Προωθητήρα Ανάδρασης (Water jet).-----	181
7.2 Υπολογισμός εξερχόμενης Ισχύος Προωθητήρα Ανάδρασης-----	182
7.3 Παρουσίαση των παραμέτρων διεξαγωγής των μετρήσεων-----	185
7.4 Παρουσίαση των αποτελεσμάτων από την διεξαγωγή των μετρήσεων----	188
7.5 Σχολιασμός των αποτελεσμάτων της μέτρησης ώσης -----	194

ΚΕΦΑΛΑΙΟ ΟΓΔΩΟ

Ανακεφαλαίωση συμπεράσματα και προτάσεις επί της πτυχιακής ---	195
8.0 ΣΥΝΟΨΗ -----	195
8.1 Συνοπτική παρουσίαση των κεφαλαίων της πτυχιακής. -----	195
8.2 Παρουσίαση των προβλημάτων που παρουσιάστηκαν -----	298
8.3 Στόχοι-----	201
8.4 Επίτευξη των στόχων – Αποτελέσματα -----	202
8.5 Προτάσεις για περαιτέρω έρευνα -----	203

ΒΙΒΛΙΟΓΡΑΦΙΑ -----	205
LINKS -----	206

ΠΑΡΑΡΤΗΜΑ Α

ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ VB

ΠΑΡΑΡΤΗΜΑ Β

ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ LPC2106

ΠΑΡΑΡΤΗΜΑ Γ

ΜΕΤΡΗΣΕΙΣ ΚΑΙ ΚΩΔΙΚΑΣ ΣΕ CD

ΕΙΣΑΓΩΓΗ

Σκοπός της εργασίας

Ο σκοπός της εργασίας αυτής είναι η διερεύνηση, ο σχεδιασμός και η ανάπτυξη μιας ολοκληρωμένης εγκατάστασης μετρήσεως ώσης προωθητήρων ανάδρασης υπό κλίμακα. Ο λόγος είναι η δημιουργία ενός εργαστηριακού περιβάλλοντος και εξοπλισμού που θα έδινε στους φοιτητές τη δυνατότητα εκτέλεσης πειραμάτων μετρήσεων είτε ως μέρος πτυχιακών εργασιών είτε ως επίδειξη στα πλαίσια μαθημάτων με έμφαση στις μετρήσεις ή στον αυτόματο έλεγχο. Στην πρόκληση αυτή περιλαμβάνονταν η γραφή του κώδικα του λογισμικού της εφαρμογής που θα χρησιμοποιείται στο υπολογιστή όσο και τον προγραμματισμό μικροεπεξεργαστή που θα δρα ως μονάδα εκτέλεσης των εντολών του υπολογιστή και λήψης των μετρήσεων. Τελικά απαιτήθηκε για την επίλυση διαφόρων προβλημάτων και ο σχεδιασμός ηλεκτρονικών πλακετών για τη δημιουργία του φίλτρου των στροφών όσο και την κατασκευή τροφοδοτικού ισχύος.

Όπως θα γίνει φανερό στο τέλος της εργασίας, τελικός σκοπός είναι η δημιουργία ενός εύελκτου εργαλείου μετρήσεων και διεξαγωγής πειραμάτων για την εύρεση ιδιοτήτων κινητήρων και προωθητήρων ανάδρασης η άλλων μέσων πρόωσης. Επιπλέον αναπτύχθηκε μια μεθοδολογία υπολογισμού της ροπής αδρανείας γραμμικών συστημάτων η οποία δεν θα ήταν δυνατό να επιβεβαιωθεί και πειραματικά χωρίς την ύπαρξη αυτής της διάταξης. Η γνώση της ροπής αδρανείας έχει καθοριστικό ρόλο στο σχεδιασμό ελεγκτών για τον υπολογισμό των βέλτιστων τιμών των όρων τους. Η μη γνώση τους, ωθεί στην πραγματοποίηση σειράς πειραμάτων με σκοπό την εύρεση των βέλτιστων τιμών μέσω της λογικής «Try and Error». Επομένως ο σχεδιαστής του ελεγκτή γνωρίζοντας την τιμή της αδρανείας μπορεί να υπολογίσει το πεδίο στο οποίο θα κυμαίνονται οι βέλτιστες τιμές και με εκτέλεση ελάχιστων πειραμάτων την εύρεση της επιθυμητής βέλτιστης τιμής. Το κυριότερο όμως είναι ότι υπολογίζονται τα πεδία τιμών όπου η συμπεριφορά του ελεγκτή είναι ασταθής και αυτά απορρίπτονται με όφελος τη μείωση της πιθανότητας πρόκλησης επικίνδυνων καταστάσεων από τη χρήση τους.

Δομή της πτυχιακής.

Στο πρώτο κεφάλαιο παρουσιάζονται τα βασικότερα τμήματα της εργαστηριακής διάταξης εκτός της κάρτα LPC και του προωθητήρα. Επιπλέον πραγματοποιείται η μοντελοποίηση του κινητήρα που θα χρειαστεί για τον υπολογισμό του ελεγκτή καθώς και στην μέτρηση της ροπής αδρανείας

Στο δεύτερο κεφάλαιο παρουσιάζεται η θεωρία των ελεγκτών τριών όρων στο συνεχή χρόνο και υπολογίζεται το μόνιμο σφάλμα για κάθε τύπο. Στη συνέχεια οι σχέσεις των ελεγκτών μεταφέρονται στο διακριτό χρόνο ο οποίος είναι το πεδίο λειτουργίας των ψηφιακών ελεγκτών.

Στο τρίτο κεφάλαιο παρουσιάζεται η κάρτα LPC2106 και του χρησιμοποιούμενου επεξεργαστή. Παρουσιάζονται οι βασικές ρυθμίσεις που επιλέγονται για τον επεξεργαστή και την κάρτα και αναλύεται διεξοδικά το σκεπτικό επιλογής της σειράς προτεραιότητας των διακοπών ροής του προγράμματος. Οι διακόπτες ελέγχουν τη ροή του προγράμματος και έχουν κρίσιμο ρόλο στην ορθή λειτουργία της κάρτας και της εκτέλεσης του κώδικα.

Στο τέταρτο κεφάλαιο παρουσιάζονται τα πρωτόκολλα επικοινωνίας και αποστολής δεδομένων. Δίνονται υπό μορφή αλγορίθμου η συνάρτηση μέτρησης των στροφών και χειρισμού των μεταφερόμενων δεδομένων.

Στο πέμπτο κεφάλαιο παρουσιάζεται η δομή της λειτουργίας της εφαρμογής επί του υπολογιστή. Επιπλέον παρουσιάζονται και κρίσιμες λειτουργίες που εκτελούνται από την εφαρμογή για τον έλεγχο της εγκατάστασης.

Στο έκτο κεφάλαιο αναπτύσσεται η μεθοδολογία για τον υπολογισμό της ροπής αδρανείας γραμμικών συστημάτων. Στη συνέχεια εκτελείται το πείραμα μέτρησης της αδράνειας και υπολογίζεται η τιμή της.

Στο έβδομο κεφάλαιο αναπτύσσεται η θεωρία για τον υπολογισμό της ισχύς του προωθητήρα ανάδρασης και εκτελείται το πείραμα υπολογισμού της καμπύλης ώσης στροφών.

Η παρουσίαση των κεφαλαίων της πτυχιακής εδώ είναι πολύ σύντομη διότι έχει συμπεριληφθεί μια εκτενέστερη αναφορά στο όγδοο κεφάλαιο στην παράγραφο 8.1 και θα ήταν πλεόνασμα να επαναληφθεί. Το όγδοο κεφάλαιο υπάρχει η περίληψη της εργασίας αυτής. Ο σκοπός της είναι η υπενθύμιση των όσων έχουν προηγηθεί, ώστε να συνταχθεί μια πλήρης εικόνα της πτυχιακής πριν αναγνωστούν τα σχόλια για τα πειραματικά αποτελέσματα και οι προτάσεις για την περαιτέρω ανάπτυξη αυτής της εργασίας, με τα οποία και κλείνει .

1) Η Εγκατάσταση μετρήσεων και το Σύστημα Πρόωσης.

Εισαγωγή

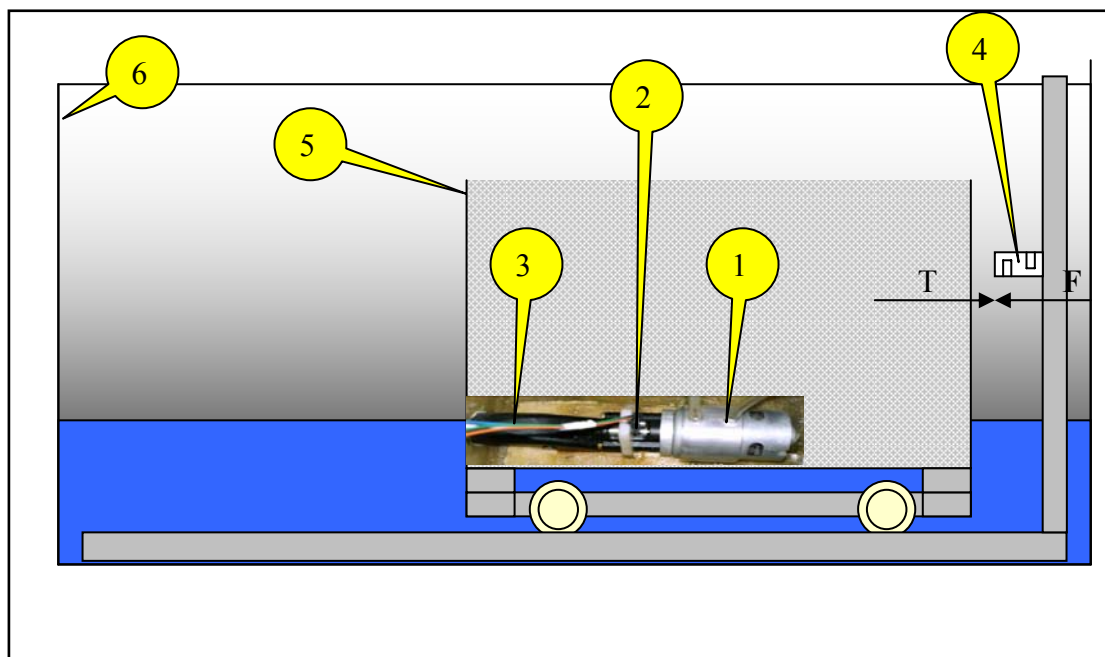
Στο κεφάλαιο αυτό περιγράφονται τα κυριότερα μέρη που απαρτίζουν την εγκατάσταση μετρήσεων καθώς και το χρησιμοποιούμενο σύστημα πρόωσης. Καταρχήν περιγράφεται η εγκατάσταση στην οποία τοποθετείται ο προωθητήρας ανάδρασης. Στη συνέχεια παρουσιάζεται ο ηλεκτρικός κινητήρας και προσδιορίζονται τα χαρακτηριστικά του ηλεκτρικού ισοδύναμου κυκλώματος. Τέλος παρουσιάζονται τα χρησιμοποιούμενα μετρικά στοιχεία και το σύστημα τροφοδοσίας της εγκατάστασης καθώς και το φίλτρο του σήματος του υπέρυθρου φωτοανιχνευτή.

1.1) Περιγραφή Εργαστηριακής Εγκατάστασης.



Εικόνα 1.1) Γενική διάταξη της πειραματική δεξαμενής.

Στην παραπάνω εικόνα αποτυπώνεται η διάταξη της δεξαμενής του υδατοστεγούς φορείου καθώς και της βάσης στήριξης της δυναμοκυψέλης .



Σχήμα 1.2) Πειραματική διάταξη του συστήματος πρόωσης και μετρήσεων.

Στο σχήμα 1.2 παρουσιάζεται η πειραματική διάταξη σε πλευρική τομή. Τα επιμέρους τμήματα από τα οποία αποτελείται είναι τα εξής:

1. Κινητήρας συνεχούς ρεύματος μονίμων μαγνητών.
2. Άξονας μετάδοσης κίνησης.
3. Προωθητήρας ανάδρασης.
4. Δυναμοκυψέλη.
5. Υδατοστεγές φορείο της εγκατάστασης.
6. Δεξαμενή νερού.

Ολόκληρη η εγκατάσταση σχεδιάστηκε με σκοπό τη μέτρηση της ώσης του προωθητήρα ανάδρασης. Η χρήση του υδατοστεγούς φορείου αποσκοπεί στην προσομοίωση του σκάφους και των εδράνων κύλισης στον περιορισμό των βαθμών ελευθερίας της κίνησης. Η βάση του μεταλλικού φορείου φέρει τα έδρανα κύλισης, τα οποία ολισθαίνουν σε σιδηροτροχιές τοποθετημένες στον πυθμένα της δεξαμενής. Τα έδρανα περιορίζουν την κίνηση του φορείου, μόνο κατά διεύθυνση παράλληλη προς τον άξονα των σιδηροτροχιών και δεν επιτρέπουν πιθανές πλευρικές μετατοπίσεις του συστήματος. Ο προωθητήρας είναι ευθυγραμμισμένος με τον άξονα κίνησης του φορείου. Το υδατοστεγές κουτί επάνω στο φορείο φέρει κατάλληλα διαμορφωμένες οπές στο κάτω τμήμα και στη μία πλευρική του επιφάνεια ώστε να προσαρμοσθεί ο προωθητήρας. Λαμβάνεται μέριμνα ώστε στα σημεία επαφής του προωθητήρα με το πολυεστερικό υλικό κάλυψης των οπών του φορείου να εφαρμοστεί κατάλληλο μονωτικό υλικό, έτσι ώστε να αποτραπεί η εισροή ύδατος στο εσωτερικό του υδατοστεγούς κουτιού. Στην εξωτερική πλευρά του φορείου και συγκεκριμένα στην οπή εξόδου του ύδατος του προωθητήρα προσαρμόζεται το ακροφύσιο εξόδου.

Η σύνδεση του ηλεκτρικού κινητήρα πραγματοποιείται σε διαμορφωμένη θέση πίσω από τον προωθητήρα. Η μεταφορά της κίνησης του κινητήρα στον προωθητήρα γίνεται μέσω της χρήσης ενός ελαστικού συνδέσμου ένωσης των δύο αξόνων κίνησης. Ο ελαστικός σύνδεσμος εκτός από τη μεταφορά της περιστροφικής κίνησης του κινητήρα χρησιμοποιείται και για την απόσβεση οποιασδήποτε άλλης σχετικής κίνησης των δύο αξόνων. Η απαιτούμενη ηλεκτρική ενέργεια παρέχεται στον κινητήρα από το δίκτυο ηλεκτρικής ενέργειας του εργαστηρίου. Επίσης μεριμνάτε η ασφαλής διεξαγωγή των μετρήσεων με την τοποθέτηση των ηλεκτρονικών διατάξεων καθώς και της τροφοδοσίας σε τέτοια απόσταση ώστε να μην υπάρχει κίνδυνος διαβροχής τους από το νερό της δεξαμενής.



Εικόνα 1.1) Φωτογραφίες του άξονα της προπέλας των εδράνων κύλισης του ελαστικού συνδέσμου και της πίσω όψης του προωθητήρα με ενσωματωμένη την προπέλα. Φαίνεται η βάση στήριξης του ηλεκτρικού κινητήρα.

Στην εμπρόσθια έδρα του μεταλλικού φορέα και στη θέση που υποδεικνύεται στο σχήμα 1.2, τοποθετείται η δυναμοκυψέλη (Load Cell) μέτρησης της ώσης. Η δυναμοκυψέλη (Load Cell) είναι ένα μετρητικό όργανο το οποίο έχει τη δυνατότητα να καταγράφει τη θλιπτική τάση που της ασκείται σε προκαθορισμένη θέση της επιφανείας του. Με τη χρήση της τιμής της ασκούμενης τάσης, της γνώση του υλικού και της γεωμετρίας του προκύπτει τελικά η ασκούμενη δύναμη επί της επιφάνειας μετρήσεως. Η έξοδος του οργάνου Load cell είναι υπό μορφή τάσης και η οποία μέσω κατάλληλων ηλεκτρονικών διατάξεων καταγραφής του ηλεκτρικού σήματος της δυναμοκυψέλης μετατρέπεται σε ψηφιακές τιμές δύναμης.

Η διεξαγωγή των πειραματικών μετρήσεων πραγματοποιείται με την παρακάτω ακολουθία. Αρχικά τοποθετείται το φορείο εντός της δεξαμενής, η οποία γεμίζεται σταδιακά με νερό. Όταν το νερό στη δεξαμενή είναι αρκετό ώστε η αναπτυσσόμενη δύναμη άνοσης να ισορροπήσει οριακά το βάρος του φορείου, τότε το φορείο επιπλέει, αλλά με τρόπο ώστε να περιορίζονται όλοι οι βαθμοί ελευθερίας του εκτός από την κίνηση του κατά μήκος των σιδηροτροχιών. Η στατική τριβή που αναπτύσσεται στα έδρανα του φορείου με αυτόν τον τρόπο ελαχιστοποιείται και πρακτικά μηδενίζεται. Όταν ο κινητήρας τεθεί σε λειτουργία, περιστρέφει την πτερωτή του προωθητήρα. Το νερό αναρροφάτε από την εισαγωγή του προωθητήρα και εκτοξεύεται με αυξημένη κινητική ενέργεια από το ακροφύσιο εξόδου σε πίεση ίση με την ατμοσφαιρική. Το αποτέλεσμα είναι να αναπτύσσεται ωστική δύναμη, η οποία κινεί το φορείο προς τα εμπρός. Το φορείο μετακινείται προς τα εμπρός έως ότου η δυναμοκυψέλη έρθει σε επαφή με το φορείο, όπως φαίνεται στο σχήμα 1.2. Αν διατηρηθεί η ταχύτητα περιστροφής του κινητήρα σταθερή και τα φαινόμενα διαταραχής του συστήματος από τους κυματισμούς θεωρηθούν αμελητέα, τότε το φορείο θα ισορροπήσει σε μία θέση όπου η αναπτυσσόμενη ωστική δύναμη T , θα είναι ίση με την κάθετη δύναμη F που αναπτύσσεται μεταξύ φορείου και εμπρόσθιας κάθετης έδρας της δεξαμενής.

$$F=T \quad (1.1)$$

Η δύναμη αυτή που μετά την αποκατάσταση της ισορροπίας του συστήματος θεωρείται σταθερή για σταθερή τιμή των στροφών του κινητήρα, καταγράφεται από τη δυναμοκυψέλη. Με αυτόν τον τρόπο για διάφορες ταχύτητες περιστροφής του κινητήρα καταγράφεται η αναπτυσσόμενη ώση και χαράσσεται η καμπύλη $F(n)$, όπου $[n]=r.p.m.$ (revolutions per minute) .

1.2) Κινητήρας Συνεχούς Ρεύματος Μονίμων Μαγνητών

Η αρχή λειτουργίας ενός ηλεκτρικού κινητήρα συνεχούς ρεύματος βασίζεται στο φαινόμενο της ανάπτυξης δύναμης Laplace σε ηλεκτρικό φορτίο κινούμενο εντός μαγνητικού πεδίου (Killian CT, 2000; Leonhard W, 1997). Σε έναν αγωγό ο οποίος διαρρέεται από συνεχές ρεύμα ,τοποθετούμενος μέσα σε μαγνητικό πεδίο, δύναμη η οποία ονομάζεται δύναμη Laplace. Η δύναμη αυτή είναι κάθετη στη διεύθυνση της μαγνητικής επαγωγής B του πεδίου και στη διεύθυνση του ηλεκτρικού ρεύματος I που διαρρέει τον αγωγό. Το μέτρο της δύναμης που δέχεται ο αγωγός υπολογίζεται από την παρακάτω εξίσωση :

$$F = I \cdot B \cdot L \cdot \sin \theta \quad (1.2)$$

όπου:

F = δύναμη στον αγωγό (Nt)

I = Ένταση του ρεύματος διαρροής του αγωγού (Amp)

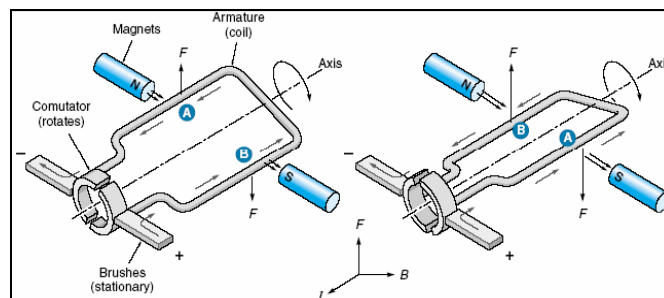
B = μαγνητική επαγωγή (Gauss)

L = μήκος του αγωγού (m)

θ = γωνία μεταξύ του μαγνητικού πεδίου και του ρεύματος(rad)

Ο κινητήρας συνεχούς ρεύματος εκμεταλλεύεται τη δύναμη αυτή κατά τέτοιο τρόπο, ώστε να προκαλείται περιστροφική κίνηση στο ρότορα. Αυτό συμβαίνει αν ο αγωγός μορφοποιηθεί σε βρόχο « πηνίο » και τοποθετηθεί σε μαγνητικό πεδίο κατά τον τρόπο που παρουσιάζεται στο σχήμα 1.3. Ο βρόχος στην περίπτωση ηλεκτρικών μηχανών ονομάζεται *περιέλιξη* «coil». Η περιέλιξη τροφοδοτείται από ηλεκτρικό ρεύμα μέσω του *συλλέκτη* «commutator» και των *ψηκτρών* «brushes». Ο συλλέκτης είναι τοποθετημένος πάνω στην περιέλιξη και περιστρέφεται μαζί με αυτή. Οι ψήκτρες είναι ακίνητες και ολισθαίνουν πάνω στον περιστρεφόμενο συλλέκτη.

Αν και η περιέλιξη τροφοδοτείται από συνεχές ρεύμα, η χωρική κατανομή της μαγνητικής επαγωγής $B(\theta)$ στο διάκενο αέρα, είναι περιοδικά μεταβαλλόμενη ως προς το δρομέα. Αυτό έχει ως αποτέλεσμα η τάση αυτεπαγωγής που αναπτύσσεται στη περιέλιξη να είναι εναλλασσόμενη. Απαιτείται λοιπόν ανόρθωση της τάσης αυτής η οποία παρέχεται μηχανικά από το συλλέκτη. Ο συλλέκτης είναι τοποθετημένος κατά τέτοιο τρόπο ώστε να συνδέει πάντα την πλευρά της περιέλιξης που είναι απέναντι από το νότιο πόλο με τη θετική ψήκτρα και την πλευρά απέναντι από τον βόρειο πόλο με την αρνητική ψήκτρα. Για να συμβαίνει αυτό κάθε φορά που η περιέλιξη διέρχεται από τον εγκάρσιο άξονα «ουδέτερη ζώνη» μεταξύ των πόλων του πεδίου, βραχυκυκλώνεται από την αντίστοιχη ψήκτρα. Αυτό έχει ως αποτέλεσμα η φορά του ρεύματος που διαρρέει την περιέλιξη να αντιστρέφεται. Κατά αυτόν τον τρόπο επιτυγχάνεται η συνεχής περιστροφή με σταθερή φορά .

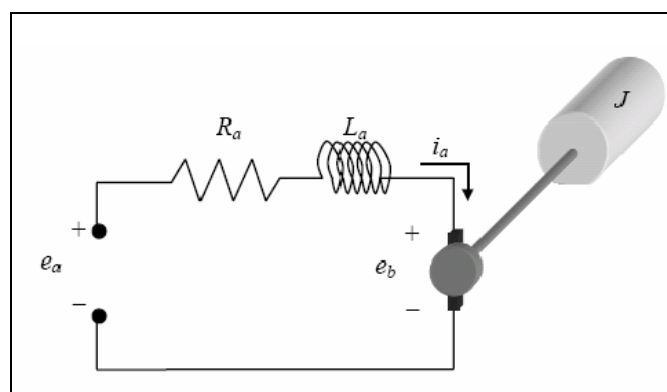


Σχήμα 1.3) Δύναμη Lorentz σε περιέλιξη ηλεκτροκινητήρα Σ.Ρ μονίμων μαγνητών.

1.3) Μοντελοποίηση Κινητήρα συνεχούς τάσης.

- **Ισοδύναμο κύκλωμα κινητήρα**

Με σκοπό τον υπολογισμό του ισοδύναμου ηλεκτρικού κυκλώματος του κινητήρα εξετάζεται η λειτουργία ενός κινητήρα συνεχούς ρεύματος μονίμων μαγνητών, σε σταθερή κατάσταση λειτουργίας. Με τον όρο σταθερή κατάσταση λειτουργίας ορίζεται η αποκατάσταση όλων των μεταβατικών φαινομένων του συστήματος. Με άλλα λόγια οι μεταβλητές που υπεισέρχονται στο σύστημα δεν έχουν καμία χρονική μεταβολή.



Σχήμα 1.4) Ισοδύναμο Κύκλωμα Ηλεκτρικού κινητήρα Συνεχούς Τάσης

Στο σχήμα 1.6 παρουσιάζεται το ισοδύναμο κύκλωμα ενός κινητήρα συνεχούς ρεύματος μονίμων μαγνητών (Killian CT, 2000; Leonhard W, 1997), αποτελούμενο από τα εξής στοιχεία:

- e_a : Τάση ακροδεκτών τροφοδοσίας [Volt]
- e_b : Αναπτυσσόμενη αντιηλεκτρεγερτική δύναμη (Counter EMF) [Volt]
- R_a : Ωμική αντίσταση του τυλίγματος τύμπανου και του συλλέκτη [Ohm]
- I_a : Ένταση ρεύματος που διαρρέει το κύκλωμα [Amp]
- L_a : Επαγωγική αντίσταση του τυλίγματος τύμπανου [H]

Η επαγωγική αντίσταση L_a του τύμπανου επιδρά μόνο στα μεταβατικά φαινόμενα, καθορίζοντας την ταχύτητα απόκρισης του κινητήρα στις μεταβολές του ρεύματος. Στη σταθερή κατάσταση λειτουργίας δεν επηρεάζει το σύστημα και θα αμεληθεί. Συνεπώς μετά την αποκατάσταση της ισορροπίας του συστήματος, συμπεριφέρεται ως βραχυκύκλωμα. Η τάση τροφοδοσίας e_a είναι συνεχής τάση, η οποία είναι ίση με τη τάση τροφοδοσίας της εγκατάστασης. Η αναπτυσσόμενη αντιηλεκτρεργετική δύναμη e_b είναι μία εσωτερικά παραγόμενη συνεχής τάση η οποία προκύπτει από τη μηχανική ανόρθωση την οποία παρέχει ο συλλέκτης του κινητήρα. Εν γένει αποδεικνύεται ότι ισχύει για τους κινητήρες συνεχούς τάσης :

$$e_b = K_e \cdot \omega \quad (1.3)$$

όπου ω είναι η γωνιακή ταχύτητα περιστροφής σε rad/sec και K_e η ηλεκτρική σταθερά του κινητήρα σε μονάδες Volt·sec/rad. Η αντιηλεκτρεργετική δύναμη αυξάνεται γραμμικά με αύξηση της γωνιακής ταχύτητας.

Με εφαρμογή του νόμου τάσης του Kirchhoff στο κύκλωμα του σχήματος 1.4 και χρήση της σχέσης 1.3 προκύπτει η παρακάτω εξίσωση των τάσεων του κλειστού κυκλώματος :

$$e_a = I_a \cdot R_a + e_b \quad \Leftrightarrow \quad \begin{matrix} e_b = K_e \cdot \omega \\ \end{matrix}$$

$$e_a = I_a \cdot R_a + K_e \cdot \omega \quad (1.4)$$

- **Προσδιορισμός των σταθερών R_a και K_e του κυκλώματος.**

Ο προσδιορισμός των σταθερών ποσοτήτων R_a και K_e ισοδυναμεί με τον προσδιορισμό της συνάρτησης $e_a = e_a(I_a, \omega_a)$ για την οποία όπως φαίνεται από τον κανόνα του Kirchhoff είναι ο γραμμικός συνδυασμός των I_a και ω . Με τη χρήση βολτόμετρου, αμπερόμετρου και ενός στροφόμετρου είναι δυνατόν για ένα πλήθος n διακριτών τιμών τάσης ακροδεκτών, να μετρηθούν και να καταγραφούν, μετά την αποκατάσταση της σταθερής κατάστασης λειτουργίας, οι αντίστοιχες τιμές των μεγεθών $e_{a,n}, I_{a,n}, \omega_n$. Τονίζεται ότι για τη μέτρηση και καταγραφή της κάθε διακεκριμένης τριάδας των παραπάνω μεγεθών πρέπει πρώτα να έχει αποκατασταθεί η ισορροπία στο κύκλωμα, δηλαδή να έχει περάσει το απαραίτητο χρονικό διάστημα των μεταβατικών φαινομένων.

Ο προσδιορισμός της συνάρτησης $e_a = e_a(I_a, \omega_a)$ μέσω της ακολουθίας των συναρτήσεων $e_{a,n}$, γίνεται με τη μέθοδο των ελαχίστων τετραγώνων.

$$e_{a,n} = I_a \cdot R_a + K_e \cdot \omega \quad (1.5)$$

Τα R_a και K_e υπολογίσθηκαν από με βάση τα παραπάνω και βρέθηκαν ότι είναι ίσα με :

$$K_e = 7.376 \cdot 10^{-3} \frac{V}{rad/s} \quad (1.6)$$

$$R_a = 0.425\Omega \quad (1.7)$$

• **Προσδιορισμός χαρακτηριστικών εξισώσεων του ισοδύναμου κυκλώματος.**

Με πολλαπλασιασμό της εξίσωσης τάσεων του Kirchhoff της σχέσης 1.4 με I_a προκύπτει η εξίσωση η οποία εκφράζει την αρχή διατήρησης της ενέργειας στο κύκλωμα.

$$e_a \cdot I_a = I_a^2 \cdot R_a + e_b \cdot I_a \quad (1.8)$$

$$\text{εισερχόμενη ηλεκτρική ισχύς} = \text{ωμικές απώλειες} + \text{μηχανική ισχύς}$$

Η στιγμιαία ηλεκτρική ισχύς που οφείλεται στην αντιηλεκτρεργετική δύναμη e_b ισούται με τη στιγμιαία μηχανική ισχύ που οφείλεται στην ηλεκτρομαγνητική ροπή. Ισχύει δηλαδή η εξίσωση :

$$e_b \cdot I_a = T \cdot \omega \xrightarrow{e_b = K_e \cdot \omega} K_e \cdot \omega \cdot I_a = T \cdot \omega \Leftrightarrow T = K_e \cdot I_a \quad (1.9)$$

Επειδή $[K_e] = \text{Volt} \cdot \text{sec} / \text{rad}$, προκειμένου να υπάρχει συμβατότητα μονάδων στην τελευταία εξίσωση θεωρείται η «**σταθερά ροπής**» K_τ για την οποία ισχύει $K_\tau = K_e$ όπου $[K_\tau] = \text{Nt} \cdot \text{m} / \text{Amp}$ (στο σύστημα μονάδων S.I).

$$K_\tau = 7.376 \cdot 10^{-3} \frac{N \cdot m}{A} \quad (1.10)$$

Επομένως ισχύει ότι η αναπτυσσόμενη ροπή του κινητήρα εκφράζεται από τη σχέση:

$$T = K_\tau \cdot I_a \quad (1.11)$$

Η σταθερά αναλογίας K_τ εκφράζει τη γραμμική εξάρτησης της αναπτυσσόμενης ροπής του κινητήρα και της έντασης του διαρρέοντος ρεύματος εντός των τυλιγμάτων. Άρα η εξίσωση διατήρησης της ενέργειας του ισοδύναμου κυκλώματος για ένα κινητήρα συνεχούς τάσης με εφαρμογή της σχέσης 1.9 στην 1.8 είναι:

$$e_a \cdot I_a = I_a^2 \cdot R_a + T \cdot \omega \quad (1.12)$$

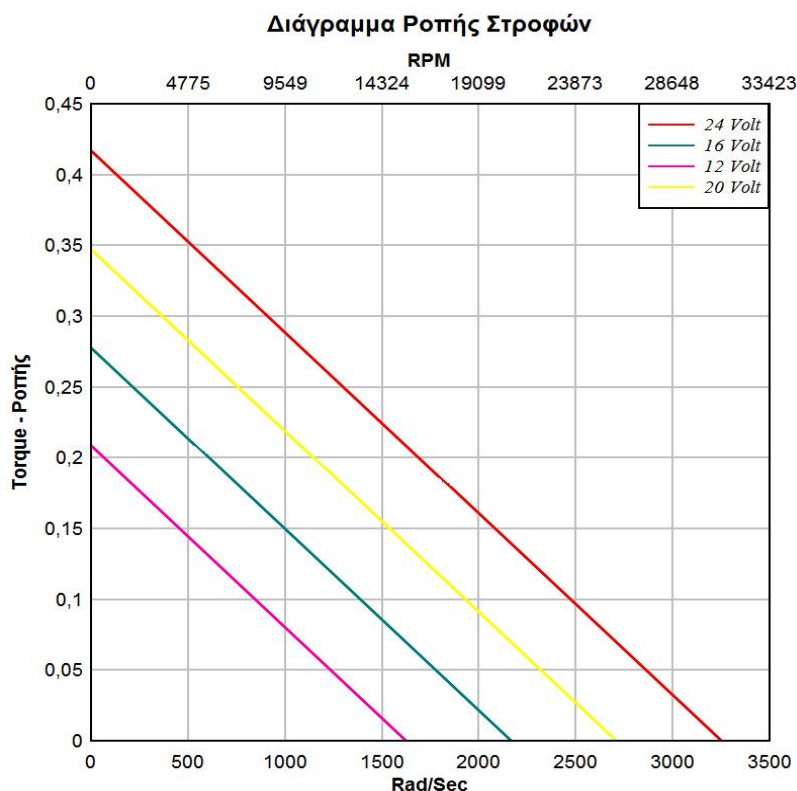
Από τις σχέσεις 1.4, 1.11 έχουμε τη σχέση της αναπτυσσόμενης ροπής του κινητήρα συναρτήσει της τάσης ακροδεκτών και της περιστροφικής του ταχύτητας.

$$I_a = \frac{T}{K_T}$$

$$e_a = I_a \cdot R_a + K_e \cdot \omega \Rightarrow e_a = \frac{T}{K_T} \cdot R_a + K_e \cdot \omega \Leftrightarrow e_a = \frac{R_a}{K_T} T + K_e \cdot \omega \Rightarrow$$

$$T(\omega, e_a) = \frac{K_T}{R_a} \cdot e_a - \frac{K_T \cdot K_e}{R_a} \cdot \omega \quad (1.13)$$

Η τελευταία εξίσωση είναι πολύ σημαντική για τη συμπεριφορά των κινητήρων συνεχούς τάσης. Από αυτήν εξάγεται το συμπέρασμα ότι για σταθερή τιμή της τάσης τροφοδοσίας, e_a , η χαρακτηριστική εξίσωση του κινητήρα $T=T(\omega)$, είναι γραμμική. Από την παραπάνω παρατήρηση προκύπτουν οι χαρακτηριστικές καμπύλες ενός κινητήρα συνεχούς τάσης για διάφορες τιμές της τάσης τροφοδοσίας e_a . Μία οικογένεια τέτοιων καμπυλών για τον κινητήρα αυτό παρουσιάζεται στο παρακάτω σχήμα.



Σχήμα 1.4) Χαρακτηριστικές καμπύλες λειτουργίας $T(\omega)$ του κινητήρα συνεχούς τάσης.

- **Μηχανικές Απώλειες Κινητήρα Συνεχούς Τάσης.**

Όταν ο κινητήρας λειτουργεί υπό την επίδραση δυναμικού φορτίου, η παραγόμενη στρεπτική ροπή του κινητήρα αντισταθμίζει τη ροπή του φορτίου, η οποία δίνεται από τη σχέση:

$$T_{motor} = J_{motor} \cdot \frac{d\omega}{dt} + T_f + T_L \quad (1.15)$$

Όπου με J_{motor} συμβολίζεται η ροπή αδρανείας του κινητήρα ,οπλισμού ,με T_f, T_L συμβολίζονται η ροπή των εσωτερικών τριβών ή απωλειών του κινητήρα και η ροπή του φορτίου αντίστοιχα .Στην περίπτωση που η δυναμική μεταβολή της ροπής του φορτίου μηδενιστεί τότε το σύστημα βρίσκεται σε σταθερή κατάσταση λειτουργίας και η επίδραση της ροπής αδράνειας του συστήματος είναι μηδενική .Αρα λοιπόν σε σταθερή κατάσταση λειτουργίας η ροπή αντίστασης των τριβών δίνεται από τη σχέση:

$$T_{motor} = J_{motor} \cdot \frac{d\omega}{dt} + T_f + T_L \Rightarrow T_f = T_{motor} - T_L \quad (1.16)$$

Επιπλέον αν ο κινητήρας λειτουργεί και υπό κενό φορτίο, $T_L = 0$ τότε η παραγόμενη στρεπτική ροπή του κινητήρα ισούται μόνο με τη ροπή αντίστασης των μηχανικών τριβών. Οι μηχανικές απώλειες ,λόγο τριβή κατά κύριο λόγο που αναπτύσσονται στα περιστρεφόμενα μηχανικά μέρη είναι γραμμικά εξαρτώμενες από τη γωνιακή ταχύτητα περιστροφής. Επομένως ισχύει ότι

$$T_f(\omega) = A_f \cdot \omega + B_f \quad (1.17a)$$

Από τις μετρήσεις που πραγματοποιήθηκαν, βρέθηκε ότι οι συντελεστές A_f και B_f της σχέσης 1.17a απωλειών των τριβών είναι $A_f = 8 \cdot 10^{-6}$ & $B_f = 11,2 \cdot 10^{-3}$. Επομένως η σχέση 1.17a γράφεται:

$$T_f(\omega) = 8 \cdot 10^{-6} \cdot \omega + 11,2 \cdot 10^{-3} \text{ (Nm)} \quad (1.17b)$$

Η ωφέλιμη παραγόμενη ροπή του κινητήρα υπολογίζεται αν αφαιρέσουμε από τη συνολική παραγόμενη τις μηχανικές απώλειες .

$$T_L = T_m - T_f \quad (1.18)$$

Πολλαπλασιάζοντας τα δύο μέρη της εξίσωσης με τη γωνιακή ταχύτητα ω , προκύπτει η αντίστοιχη σχέση για τον προσδιορισμό της αποδιδόμενης ισχύος στο φορτίο. $P_m = P_f + P_L$. Αφαιρώντας από την παραγόμενη ισχύ του κινητήρα σε κατάσταση φόρτισης, την ισχύ που παράγεται όταν ο κινητήρας λειτουργεί με κενό φορτίο υπολογίζεται η ισχύς που αποδίδεται στο φορτίο. Δηλαδή ισχύει

$$P_L = P_m - P_f .$$

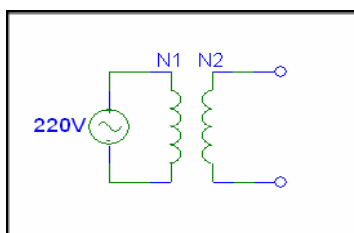
1.4) Σύστημα τροφοδοσίας ηλεκτρικής ενέργειας του συστήματος πρόωσης και της διάταξης των μετρήσεων .

Στα πλαίσια λειτουργίας της πειραματικής εγκατάστασης της κλίνης δοκιμών απαιτήθηκε η κατασκευή τροφοδοτικής διάταξης συνεχούς τάσης. Ο πρώτος λόγος της κατασκευής της είναι ότι ο χρησιμοποιούμενος κινητήρα συνεχούς τάσης απαιτεί ως γνωστών συνεχή τάση και επιπλέον η τιμή της τάσης για την επίτευξη των μεγίστων στροφών του είναι 16 Volt. Δεύτερον οι ηλεκτρονικές διατάξεις μετρήσεων της ώσης, του υπέρυθρου φωτοανιχνευτή καθώς και της LPC2106 χρειάζονται για τη λειτουργία τους, συνεχή τάση με πεδία τιμών για το καθένα από 5 έως 12 Volt. Από τους παραπάνω λόγους, τη σωστή διαμόρφωση της πειραματικής διάταξης και την κάλυψη όλων των απαιτήσεων και επιπέδων τάσης σχεδιάστηκε και κατασκευάστηκε η παρακάτω τροφοδοσία. Αποτελείται από το μετασχηματιστή ,τον ανορθωτή τον πυκνωτή εξομάλυνσης καθώς και από ένα σύνολο ρυθμιστών τάσης «Voltage Regulators» με τους οποίους λαμβάνονται τα επίπεδα τάσης 3,5,12 volt για την τροφοδοσία των ηλεκτρονικών πλακετών.

1.4.1) Μετασχηματιστής Τάσης

Στο εργαστήριο η παροχή ηλεκτρικής τάσης παρέχεται από το κοινό ηλεκτρικό δημόσιο δίκτυο και επομένως η ενεργός τιμή της τάση του εναλλασσομένου ρεύματος είναι $V_{ac} = 230Volt$. Για το μετασχηματισμό της τάσης εναλλασσομένου ρεύματος σε άλλα επίπεδα εναλλασσόμενης τάσης χρησιμοποιούνται ως γνωστών οι λεγόμενοι μετασχηματιστές. Ο μετασχηματιστής αποτελείται από δύο τουλάχιστον τυλίγματα συνεξευγμένα μέσω ενός κοινού μαγνητικού κυκλώματος ,σχήμα 1.5 . Το ένα τύλιγμα ονομάζεται πρωτεύον και θεωρείται ως είσοδος ηλεκτρικής ενέργειας, και το άλλο δευτερεύον και αποτελεί την έξοδο ηλεκτρικής ενέργειας. Αν με $N1$ συμβολιστεί ο αριθμός των ελιγμάτων του πρωτεύοντος και με $N2$ ο αριθμός των ελιγμάτων του δευτερεύοντος, ο λόγος μετασχηματισμού α του μετασχηματιστή ορίζεται ως εξής $\alpha = \frac{N1}{N2}$. (1.19)

Ο λόγος μετασχηματισμού είναι μεγαλύτερος από τη μονάδα αν ο μετασχηματιστής χρησιμοποιείται για να υποβιβάσει την τάση και μικρότερος από τη μονάδα αν ο μετασχηματιστής ανεβάζει την τάση. Άρα προκειμένου να υποβιβαστεί η τάση από τα 230V στα 16V, ο μετασχηματιστής που χρησιμοποιείται, έχει λόγο μετασχηματισμού μεγαλύτερο της μονάδος .



Σχήμα 1.5) Ιδανικός μετασχηματιστής.

Θεωρείται ότι οι ωμικές αντιστάσεις των δύο τυλιγμάτων και η μαγνητική αντίσταση του πυρήνα είναι αμελητέες. Στην περίπτωση αυτή οι απώλειες ενέργειας είναι μηδενικές και ο μετασχηματιστής καλείται ιδανικός. Σε έναν ιδανικό μετασχηματιστή η εισερχόμενη ισχύς είναι ίση με την εξερχόμενη. Επιπλέον σε κάθε χρονική στιγμή ο λόγος των στιγμιαίων τάσεων πρωτεύοντος και δευτερεύοντος ισούται με το λόγο μετασχηματισμού a . Ισχύουν δηλαδή οι σχέσεις:

$$V_1 \cdot I_1 = V_2 \cdot I_2 \text{ \& } \frac{V_1}{V_2} = \frac{N_1}{N_2} = a \quad (1.20)$$

Από τις σχέσεις αυτές προκύπτει ότι ο λόγος των στιγμιαίων ρευμάτων πρωτεύοντος και δευτερεύοντος ισούται με το αντίστροφο του λόγου μετασχηματισμού .

$$\text{Δηλαδή } \frac{I_1}{I_2} = \frac{N_2}{N_1} = 1/a \quad (1.21)$$

Συνεπώς ο λόγος του μετασχηματιστή a δίνεται από τη σχέση 1.20 και τις τιμές $V_1 = 230 \cdot \sqrt{2} = 325 \text{ Volt}$ & $V_2 = 16$, όπου V_1 το πλάτος της τάσης εισόδου στο μετασχηματιστή από το ηλεκτρικό δίκτυο και V_2 η ζητούμενη τάση εξόδου από των μετασχηματιστή για τη δημιουργία της συνεχής τάσης τροφοδοσίας του κινητήρα.

$$a = \frac{V_1}{V_2} = \frac{325}{19} \approx 20 \quad (1.22)$$

Όμως στο εμπόριο υπάρχουν μετασχηματιστές με τυποποιημένους λόγους και για αυτό επιλέχθηκε ο λόγος $a = 15$. Το τελικό πλάτος της εξαγόμενης από το μετασχηματισμένη εναλλασσόμενης τάσης είναι:

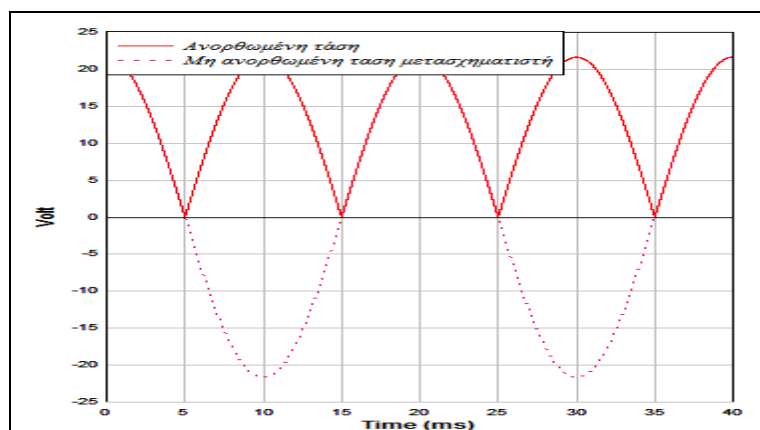
$$V_2 = \frac{V_1}{a} = \frac{325}{15} \approx 21.65 \text{ Volt} \quad (1.23)$$

Επιπλέον ορίζεται ότι πρέπει να ικανοποιεί την απαίτηση της παραλαβής φορτίου 1 KWatt.

1.4.2) Μονοφασικός Ανορθωτής Πλήρους Κύματος (τύπου γέφυρας)

Ο μετασχηματιστής χρησιμοποιείται προκειμένου η τιμή της τάσης τροφοδοσίας να υποβιβαστεί σε επίπεδα τιμών τάσης λειτουργίας του κινητήρα. Το επόμενο βήμα είναι η μετατροπή της εναλλασσόμενης κυματομορφής της εξαγόμενης τάσης από το μετασχηματιστή σε μεταβαλλόμενη. Το κύκλωμα το οποίο πραγματοποιεί τη μετατροπή του εναλλασσόμενου ρεύματος σε συνεχές καλείται «ανορθωτής». Στη συγκεκριμένη εφαρμογή χρησιμοποιείται ανορθωτής γέφυρας πλήρους κύματος (Millman G & Halkias C, 1973; Sedra AS & Smith KC, 1987). Τα βασικά στοιχεία ενός κυκλώματος ανόρθωσης γέφυρας είναι οι λεγόμενες δίοδοι. Με κατάλληλη διάταξη τους επιτυγχάνεται η τελική ανόρθωση της μετασχηματισμένης τάσης και

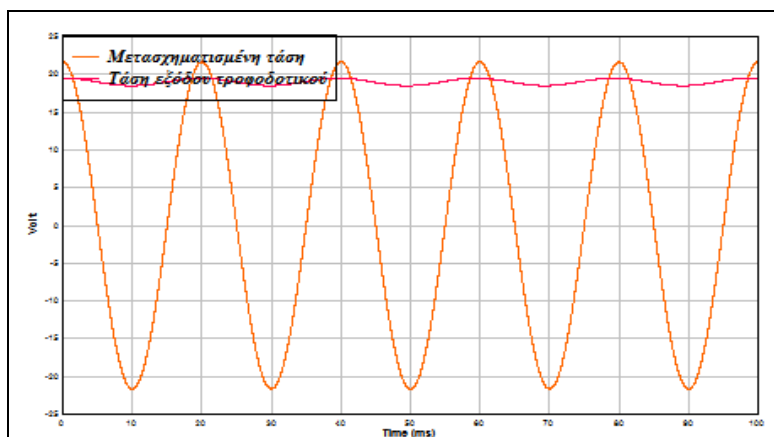
από εναλλασσόμενη μετατρέπεται σε μεταβαλλόμενη με ελάχιστη τη μηδενική τιμή. Η λειτουργία του ανορθωτή δίνεται στο παρακάτω σχήμα :



Σχήμα 1.6) Ανορθωμένη κυματομορφή της εναλλασσομένου τάσης

1.4.3) Φίλτρο Πυκνωτή

Η εξομάλυνση της τάσης που εξέρχεται από την ανορθωτική διάταξη γέφυρας, πραγματοποιείται με την παρεμβολή ενός πυκνωτή παράλληλα προς το φορτίο. Ο πυκνωτής αποθηκεύει ένα πόσο ενέργειας υπό μορφή ηλεκτρικού φορτίου και το διαθέτει προς κατανάλωση όταν του ζητηθεί με αποτέλεσμα να μειώνεται σημαντικά η κυμάτωση της τάσης εξόδου. Επιλέγεται η χρήση πυκνωτή χωρητικότητα 20mF. Στο παρακάτω σχήμα παρουσιάζεται η τελική μορφή της εξαγόμενης μορφής της τάσης από τη διάταξη τροφοδοσίας.



Σχήμα 1.7) Η καμπύλη της τελικής τάσης εξόδου της διάταξης τροφοδοσίας.

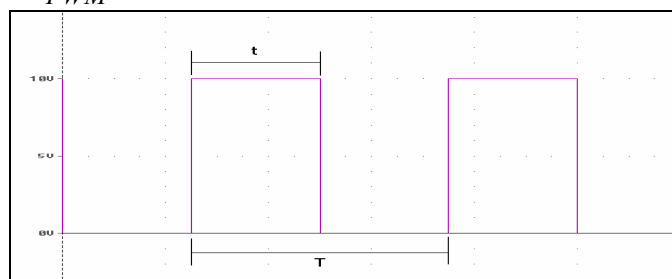
1.5) Παλμοτροφοδοτικός Μετατροπέας Τάσης.

Η τροφοδοσία του κινητήρα με συνεχή τάση πραγματοποιείται με τη χρήση διάταξης μετασχηματιστή και ανορθωτή. Όμως η τιμή της τάσης που λαμβάνεται ως έξοδος της διάταξης αυτής είναι σταθερά εξαρτώμενη από την τιμή της τάσης του εναλλασσομένου ρεύματος του δικτύου του εργαστηρίου. Η τιμή της συνεχής τάσης του τροφοδοτικού υπολογίστηκε παραπάνω και είναι ίση με 21 Volt βλέπε σχέση 1.23. Όμως για τις ανάγκες ελέγχου και ρύθμισης των στροφών του ηλεκτρικού κινητήρα απαιτείται η ύπαρξη κατάλληλης διάταξης ρύθμισης της τάσης που

εφαρμόζεται στους ακροδέκτες του κινητήρα. Η διάταξη αυτή ονομάζεται Παλμοτροφοδοτικός Μετατροπέας Τάσης ή Ρυθμιστής Στροφών και λειτουργεί με τη χρήση του παλμού PWM (Pulse Width Modulation-Διαμόρφωσης Εύρους Παλμού). Η αρχή λειτουργίας ενός PWM ρυθμιστή στροφών (Alciatore DG & Histan MB, 2003; Auslander DM & Kempf CJ, 1996), συνίσταται στην ακαριαία μετάθεση της συνεχούς τάσης τροφοδοσίας μεταξύ δύο τιμών, με μία προκαθορισμένη συχνότητα f , η οποία είναι συνήθως πολλαπλάσιο του 1kHz. Η υψηλή τιμή της τάσης ισούται με το μέτρο της συνεχούς τάσης τροφοδοσίας που εξέρχεται από την ανορθωτική διάταξη, μείον κάποιες μικρές απώλειες. Η ελάχιστη τιμή της τάσης ισούται με μηδέν. Η υψηλή τιμή της τάσης διατηρείται κατά ένα παλμό μεταβλητού πλάτους t , κατά τη διάρκεια μίας περιόδου T .

Η προκύπτουσα κυματομορφή του παλμού PWM δίνεται στο παρακάτω σχήμα και χαρακτηρίζεται από το μέγεθος «duty cycle». Το duty cycle ορίζεται ως ο λόγος μεταξύ του πλάτους του παλμού t , προς την περίοδο T της κυματομορφής. Εκφράζει δηλαδή το ποσοστό του χρόνου στο οποίο ο παλμός είναι υψηλός, στη διάρκεια μίας περιόδου. Δηλαδή ισχύει :

$$DutyCycle = \frac{t}{T_{PWM}} \cdot 100\% \quad (1.24)$$



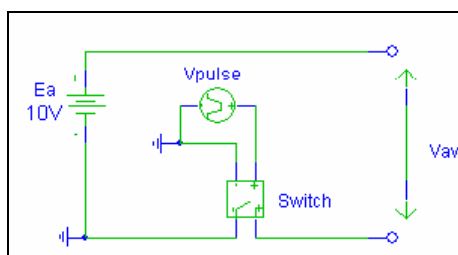
Σχήμα 1.8) Μορφή της καμπύλης του παλμού PWM.

- **Μοντελοποίηση παλμοτροφοδοτικού μετατροπέα τάσης .**



Εικόνα 1.2) Ο παλμοτροφοδοτικός μετατροπέας του ηλεκτρικού κινητήρα

Έχει βρεθεί ότι ο παλμοτροφοδοτικός μετατροπέας τάσης μοντελοποιείται με το παρακάτω ηλεκτρικό μοντέλο.



Σχήμα 1.9) Ηλεκτρικό ισοδύναμο μοντέλο του παλμοτροφοδοτικού μετατροπέα τάσης.

Δηλαδή πρόκειται για έναν ελεγχόμενο περιοδικό διακόπτη, μέσω ενός παλμού, και κλείνει ή ανοίγει το κύκλωμα στη γείωση. Για όσο διάστημα ο παλμός ελέγχου «V pulse» είναι ενεργός «on» ο διακόπτης είναι κλειστός ενώ όσο διάστημα ο παλμός είναι «0 ή Off» ο διακόπτης είναι ανοικτός.

Επομένως η τάση στα άκρα του κυκλώματος δίνεται από τη σχέση :

$$V_{av}(t) = E_a \cdot [\delta(t) - \delta(t - t_{on})] + 0 \cdot [\delta(t - t_{on}) - \delta(t - t_{Ts})] \quad (1.25)$$

Τα βασικά χαρακτηριστικά του χρησιμοποιούμενου παλμοτροφοδοτικού μετατροπέα τάσης είναι :

- Περίοδος παλμού $T_s = 0.5 \text{ msec}$ η αλλιώς η συχνότητα του PWM παλμού είναι 2KHz.
- $Duty_cycle = \frac{t_{on}}{T_s} \cdot 100\%$

Η μέση τάση σε μία περίοδο του παλμού δίνεται από τη σχέση :

$$\bar{V}_{av}(t) = \frac{1}{T_s} \int_0^{T_s} E_a \cdot [u(t) - u(t - t_{on})] + 0 \cdot [u(t - t_{on}) - u(t - t_{Ts})] dt \Rightarrow$$

$$\bar{V}_{av}(t) = \frac{1}{T_s} \cdot E_a \cdot [t_{on} - 0] + \frac{1}{T_s} \cdot 0 \cdot [T_s - t_{on}] = E_a \cdot \frac{t_{on}}{T_s} = E_a \cdot \frac{Duty_Cycle}{100\%} \quad (1.26)$$

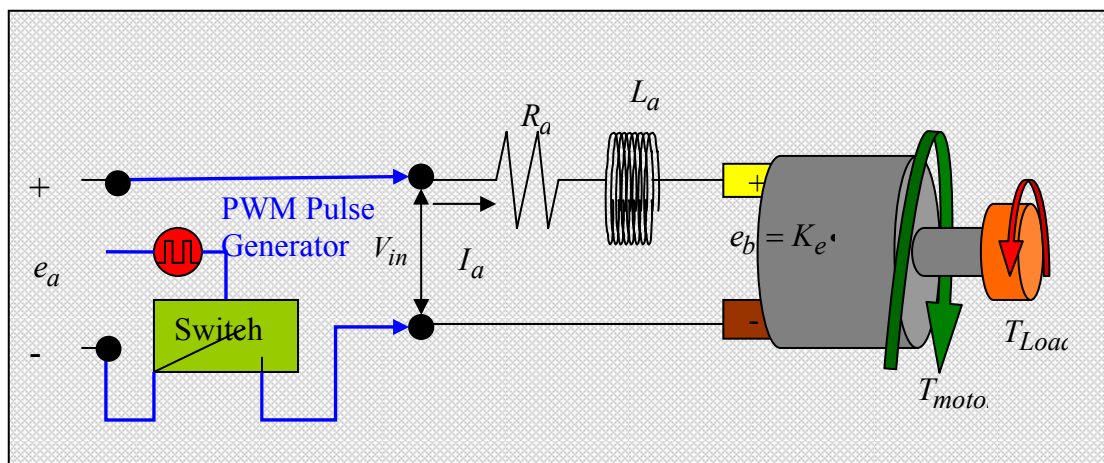
Η χρήση του παραπάνω μοντέλου ως μετατροπέας τάσης δηλαδή παροχή μεταβλητής συνεχούς τάσης προϋποθέτει τους έξι δυο όρους .

- 1) Το πλάτος του παλμού να κυμαίνεται μέσα στα όρια αντοχής του ηλεκτρικού κυκλώματος .
- 2) Τα ηλεκτρικά κυκλώματα να παρουσιάζουν αρκετή αδράνεια ως προς την τάση τροφοδοσίας τους. Με τον όρο αδράνεια εννοούμε ότι η απόκριση του ηλεκτρικού κυκλώματος στον παλμό τάσης του τροφοδοτικού «PWM» είναι όμοια με την απόκριση του ίδιου ηλεκτρικού κυκλώματος υπό σταθερή τάση μέτρου ίσο με τη μέση τάση του παλμού. Ένα ηλεκτρικό κύκλωμα για να παρουσιάζει τέτοια συμπεριφορά ,θα πρέπει στο μαθηματικό μοντέλο το οποίο το περιγράφει να χαρακτηρίζεται η απόκριση του ως απεριοδική αποσβεννύμενη ταλάντωση.

Τέτοια ηλεκτρικά κυκλώματα είναι και οι κινητήρες συνεχούς τάσης .Κρίσιμος παράγοντας για το χαρακτηρισμό της απόκρισης του μοντέλου είναι η περίοδος του παλμού η αλλιώς η συχνότητα του. Η χρήση συχνοτήτων μεγαλύτερων του 1kHz είναι συνήθως ικανοποιητικές ώστε κατά τη διάρκεια ενός παλμού η γωνιακή ταχύτητα να θεωρείται σταθερή .

- **Σύζευξη κινητήρα παλμοτροφοδοτικού μετατροπέα τάσης**

Ανάμεσα στον ηλεκτρικό κινητήρα και το τροφοδοτικό «μετασχηματιστής» συνδέεται ο παλμοτροφοδοτικός μετατροπέας (Converter) .Ο οποίος λαμβάνει το σήμα ελέγχου και παράγει το σήμα δράσης δηλαδή μετατρέπει τη συνεχή τάση σε μεταβαλλόμενη τάση της μορφής PWM με κατάλληλο Duty Cycle και πλάτος ίσο με την τιμή της συνεχούς τάσης. Στο παρακάτω σχήμα η τάση V_{in} συμβολίζει την εφαρμοζόμενη τάση επί στους ακροδέκτες του ενώ με e_b η αντιηλεκτρεγερτική δύναμη του ηλεκτροκινητήρα (Counter ElectroMotive Force).



Σχήμα 1.10) Το ισοδύναμο ηλεκτρικό κύκλωμα του συζευγμένου κινητήρα με τον παλμοτροφοδοτικό μετατροπέα τάσης.

Η αντιηλεκτρεγερτική δύναμη δίνεται από τη σχέση 1.3 και ισχύει:

$$e_b = K_e \cdot \omega = K_e \cdot 2 \cdot \pi \cdot N_{rps} \quad (1.27)$$

Όπου N είναι η ταχύτητα του κινητήρα σε στροφές ανά δευτερόλεπτο «RPS» και K_e η σταθερά αναλογία περιστροφικής ταχύτητας και αντιηλεκτρεγερτικής τάσης. Εφαρμόζοντας το νόμο του Kirchhoff στο παραπάνω κύκλωμα προκύπτει ότι :

$$V_{in} - e_b = I \cdot R + L \cdot \frac{dI}{dt} \xrightarrow{e_b = K_e \cdot \omega} V_{in} - K_e \cdot \omega = I \cdot R + L \cdot \frac{dI}{dt} \quad (1.28)$$

Η επαγωγή L καθορίζει το πόσο γρήγορα μεταβάλλεται το ρεύμα στο κύκλωμα. Όταν το ρεύμα είναι σταθερό και έχει επέλθει σταθερή κατάσταση, το πηνίο δεν επηρεάζει καθόλου το κύκλωμα. Επομένως αμελώντας το L προκύπτει η παρακάτω σχέση και η οποία είναι όμοια με την 1.4 :

$$I = \frac{V_{in} - K_e \cdot \omega}{R} \quad (1.29)$$

Κατά τη μοντελοποίηση του κινητήρα θεωρείται ότι η παρουσία του παλμού τροφοδοσίας δεν μεταβάλλει σημαντικά τη γωνιακή του ταχύτητα στο διάστημα μιας περιόδου λόγω της επίδρασης της αδράνεια των μηχανικών μερών του .Η θεώρηση

αυτή έχει ως συνέπεια ότι όλα τα μεγέθη που λαμβάνουν μέρος στο μοντέλο να αντικατασταθούν από τη μέση τιμή τους στη διάρκεια του ενός παλμού. Αν δεν ίσχυε αυτό τότε θα έπρεπε να μεταβάλλονται σημαντικά όλα τα μεγέθη γεγονός που δεν απορρέει από καμία έως τώρα μέτρηση. Επίσης γνωρίζουμε ότι το ολοκλήρωμα της τάσης ενός πηνίου σε μία περίοδο είναι πάντα μηδέν εφόσον αρχική και τελική κατάσταση είναι ίδια.

$$\frac{1}{T_s} \cdot \int_0^{T_s} L \cdot \frac{dI}{dt} dt = \frac{1}{T_s} \cdot L \cdot \int_0^{T_s} \frac{dI}{dt} dt = \frac{1}{T_s} \cdot L \cdot [I(T_s) - I(0)] = 0 \quad (1.30)$$

Επομένως οι μέσες τιμές εμφανίζονται με την ολοκλήρωση της σχέσης 1.28 για μία περίοδο του παλμού :

$$\begin{aligned} \frac{1}{T_s} \cdot \int_0^{T_s} (e_a - e_b) dt &= \frac{1}{T_s} \cdot \int_0^{T_s} I \cdot R dt + \frac{1}{T_s} \cdot \int_0^{T_s} L \cdot \frac{dI}{dt} dt \Leftrightarrow \\ \Rightarrow V_{inM} - e_{bM} &= \frac{1}{T_s} \cdot \int_0^{T_s} I \cdot R dt + 0 \Rightarrow \bar{e}_a - \bar{e}_b = \bar{I} \cdot R + 0 \end{aligned}$$

$$V_{inM} - e_{bM} = I_M \cdot R \quad (1.31)$$

με M συμβολίζονται οι μέσες τιμές.

Η μέση τιμή τη τροφοδοσία του κινητήρα που υπολογίστηκε παραπάνω προκύπτει από την παρακάτω σχέση :

$$\begin{aligned} \bar{V}_{in} = e_{aM} &= E_a \cdot \frac{Duty_Cycle}{100\%} \\ e_{bM} &= \frac{1}{T_s} \cdot \int_0^{T_s} e_b dt = \frac{e_b = K_e \cdot \omega}{T_s} \cdot \int_0^{T_s} K_e \cdot \omega dt \stackrel{\omega=const}{=} \frac{1}{T_s} \cdot K_e \cdot \omega \cdot \int_0^{T_s} dt = K_e \cdot \omega = e_b \end{aligned}$$

Επομένως αν ρυθμίσουμε η μεταβολή της μέσης τάσης και εκείνη της τιμής του ρεύματος να γίνονται με πολύ μικρότερη συχνότητα από τη συχνότητα του παλμού τροφοδοσίας 2kHz τότε η παρουσία του L μπορεί να αμεληθεί για τον υπολογισμό της απόκρισης του κινητήρα στις μεταβολές της τάσης του κινητήρα .

Αλλά ακόμα και όταν το ρεύμα μεταβάλλεται τότε εφαρμόζοντας μετασχηματισμό Laplace στην εξίσωση 1.28 προκύπτει :

$$V_{in}(s) - e_b(s) = I(s) \cdot R + L \cdot s \cdot I(s) \quad (1.32)$$

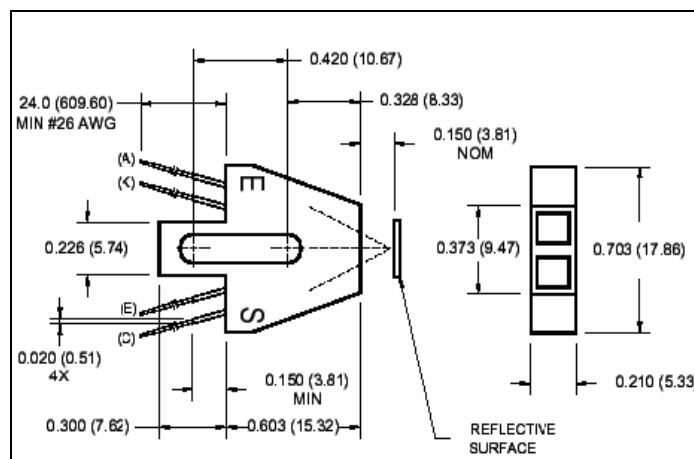
Στην περιοχή στην οποία γίνεται η μελέτη η μιγαδική συχνότητα είναι αρκετά μικρή με αποτέλεσμα το γινόμενο $L \cdot s \cdot I(s)$ της παραπάνω εξίσωσης να αμελείται. Άλλωστε η εφαρμογή του παλμοτροφοδοτικού μετατροπέα τάσης έχει ως σκοπό τη ρύθμιση της τάσης που εφαρμόζεται στο κινητήρα ως συνεχή τάση και όχι ως

μεταβαλλόμενη. Θυμίζετε ότι σκοπός είναι ο προσδιορισμός των φυσικών χαρακτηριστικών του κινητήρα ,η τιμή των οποίων στοχεύετε να είναι αντιπροσωπευτική για όλο το φάσμα των στροφών λειτουργίας του ώστε να κατασκευαστεί ικανοποιητικός ελεγκτής στροφών .

1.6) Περιγραφή του υπέρυθρου φωτοανιχνευτή.

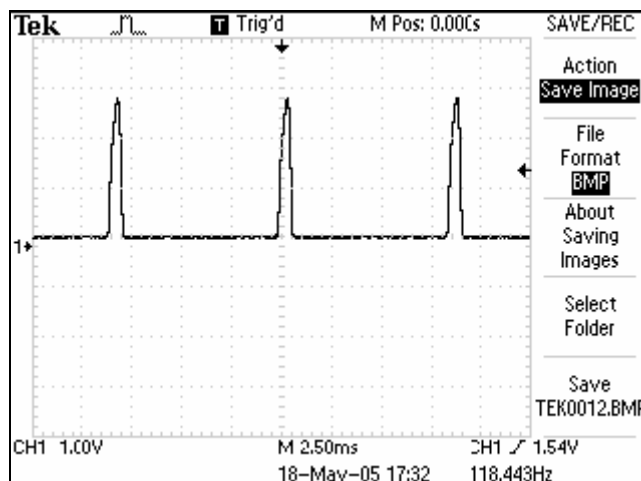
Για τη μέτρηση των στροφών χρησιμοποιήθηκε ο υπέρυθρος φωτοανιχνευτής που εμφανίζεται στην παρακάτω εικόνα. Ο υπέρυθρος φωτοανιχνευτής λειτουργεί εκπέμποντας υπέρυθρη ακτινοβολία και ανάλογα με την ένταση της ανακλώμενης ακτινοβολίας πίσω σε αυτόν παράγει μία τάση. Η τιμή της τάση είναι αντιστρόφως ανάλογη της επιστρεφόμενης ακτινοβολίας και ανάλογη της τάσης τροφοδοσίας. Δηλαδή για πλήρη επιστροφή της ακτινοβολίας η τάσης εξόδου είναι ίση με μηδέν ενώ στην αντίθετη περίπτωση της μη επιστροφής ακτινοβολίας παίρνει τιμή ίση με την τάση τροφοδοσίας. Οι διαστάσεις του αναφέρονται στην επόμενη εικόνα και η τάση τροφοδοσίας του είναι 5 V. Ο άξονας του προωθητήρα ανάδρασης αποτελείται από δυο τμήματα ανάμεσα στα οποία παρεμβάλλεται ελαστικός σύνδεσμος για να δέχεται τις παραμορφώσεις. Σε κάθε ένα τμήμα υπάρχει οπή στην οποία εισέρχεται κοχλίας για την ένωση των δύο τμημάτων. Κάθε φορά που η οπή διέρχεται μπροστά από το φωτοανιχνευτή, τότε αυτός παράγει παλμό με πλάτος 3V. Η συνδεσμολογία της συσκευής έχει ως εξής:

- Το άσπρο καλώδιο συνδέεται με την τροφοδοσία.
- Το μπλε καλώδιο μεταφέρει τον παλμό στο φίλτρο.
- Το πράσινο και το πορτοκαλί είναι οι γειώσεις. Η μια χρησιμοποιείται για να κλείσει το κύκλωμα της τροφοδοσίας και η άλλη για τη μεταφοράς του παλμού.



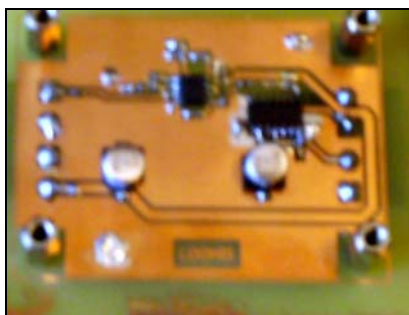
Σχήμα 1.11) Διαστάσεις και σχεδιάγραμμα του υπέρυθρου φωτοανιχνευτή.

Ο παλμός που παράγει ο φωτοανιχνευτής φαίνεται στο σχήμα 1.12. Ο παλμός εναλλάσσεται μεταξύ των τιμών τάσης μηδέν και 3 Volts περίπου. Επιπλέον η μετάβαση από τη μέγιστη τάση στην ελάχιστη δεν γίνεται ακαριαία, κάνοντας πιθανή την ύπαρξη κάποιου ποσού θορύβου στο σήμα του ανιχνευτή, με αποτέλεσμα η μέτρηση των στροφών να είναι λανθασμένη. Για να λυθεί αυτό το πρόβλημα ο παλμός του διέρχεται από το φίλτρο του επόμενου κεφαλαίου ,και εν συνεχεία αποστέλλεται στην κάρτα LPC2106.



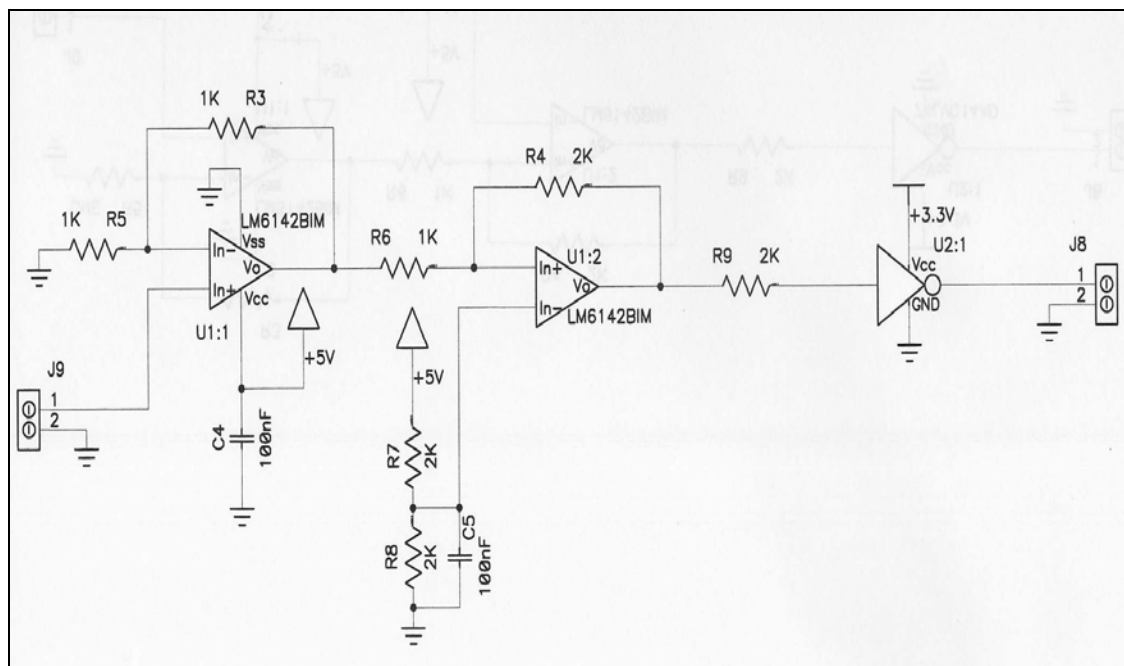
Σχήμα 1.12) Ο παραγόμενος παλμός στην έξοδο του υπέρυθρου φωτοανιχνευτή όταν διέρχεται η σπή μπροστά από αυτόν όπως φαίνεται στον παλμογράφο.

1.7) Περιγραφή Φίλτρου σήματος από τον υπέρυθρο φωτοανιχνευτή



Εικόνα 1.1) Κάτω όψη του φίλτρου.

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο το σήμα από τον ανιχνευτή περιέχει και θόρυβο με αποτέλεσμα τη λανθασμένη μέτρηση των στροφών του κινητήρα. Για την κατασκευή του φίλτρου αυτού χρησιμοποιήθηκε ένας διπλός τελεστικός ενισχυτής « operational amplifier » καθώς και ένας αντιστροφέας « Inverter » για τη ρύθμιση του ύψους του σήματος στα επιτρεπτά επίπεδα των σημάτων εισόδου στην κάρτα. Το φίλτρο λειτουργεί φιλτράροντας το πλάτος του σήματος χρησιμοποιώντας μια περιοχή του στην οποία διατηρεί σταθερή την τιμή της εξόδου του. Η μέθοδος αυτή έχει ως συνέπεια το πλάτος του θορύβου θα πρέπει να ξεπερνάει το πλάτος του πεδίου φιλτραρίσματος ώστε να επηρεάσει το εξερχόμενο σήμα από το φίλτρο προς την κάρτα LPC2106.



Σχήμα 1.13) Συνδεσμολογία φίλτρου.

Ως τελεστικός ενισχυτής χρησιμοποιήθηκε ο LM614BIM και ως αντιστροφέας ο 74LVC14AD . Για την ολοκλήρωση του φίλτρου απαιτήθηκε και χρήση διάφορων αντιστάσεων όπως φαίνονται στο παραπάνω σχήμα.

Το σήμα από τον ανιχνευτή εισέρχεται στο φίλτρο στην ακίδα 1 του J9 . Στη συνέχεια εισέρχεται στο θετικό κλάδο του πρώτου τελεστικού ενισχυτή του LM614BIM .Ο αρνητικός κλάδος συνδέεται με τη γείωση μέσω της αντίστασης R5 και ταυτόχρονα συνδέεται με την έξοδο του μέσω της αντίστασης R3. Οι αντιστάσεις R5 και R3 ορίζουν το λόγο ενίσχυσης του εξερχόμενου σήματος από αυτόν . Δηλαδή το σήμα εξόδου δίνεται από τη σχέση :

$$V_{U1:1_Out} = \frac{R_5 + R_3}{R_3} \cdot V_{U1:1_in}^+ = \frac{2}{1} \cdot V_{U1:1_in}^+ \quad (1.33)$$

Δηλαδή ο λόγος ενίσχυσης του σήματος είναι 2.

Στη συνέχεια το ενισχυμένο σήμα θα διέλθει από το δεύτερο ενισχυτή του LM614BIM στον οποίο έχει οριστεί το πεδίο τιμών «threshold» στο οποίο η έξοδος του φίλτρου παραμένει σταθερή και ουσιαστικά υλοποιείται το φιλτράρισμα του σήματος .

Ο ορισμός του πεδίου τιμών «threshold» πραγματοποιείται μέσω των αντιστάσεων R6 και R4 καθώς και της εφαρμοζόμενης τάσης στον αρνητικό κλάδο του δεύτερου τελεστικού ενισχυτή .Η τιμή της τάσης αυτής είναι 2.5 Volt που προκύπτει από τη διαίρεση της τάσης 5volt τροφοδοσίας μέσω των εν σειρά αντιστάσεων R8 και R7 και υπολογίζεται από την παρακάτω σχέση. Η ύπαρξη του πυκνωτή C5 έχει σκοπό τη σταθεροποίηση της τάσης αναφοράς.

$$V_{U1:2_in}^- = 5 \cdot \frac{R8}{R8 + R7} = 5 \cdot \frac{1}{2} = 2.5 \text{ Volt} \quad (1.34)$$

Τα όρια της χρησιμοποιούμενης περιοχής φιλτραρίσματος «threshold» προκύπτουν ξεχωριστά από τη μελέτη των δύο καταστάσεων ανόδου και καθόδου του σήματος.

➤ **Άνοδος του σήματος.**

Το σήμα κατά την έξοδό του από τον πρώτο τελεστικό ενισχυτή U1:1 εισέρχεται στο θετικό κλάδο του δεύτερου τελεστικού ενισχυτή. Η έξοδος του δεύτερου τελεστικού ενισχυτή ορίζεται από το αποτέλεσμα της σύγκρισης των εισόδων του. Αν το αποτέλεσμα είναι θετικό η έξοδος είναι +5 volt, αν όχι είναι η γείωση. Εφόσον έχουμε άνοδο του σήματος και επιπλέον η τιμή εισόδου δεν έχει ξεπεράσει την τάση στον αρνητικό κλάδο τότε στην έξοδο του ενισχυτή U1:2 εφαρμόζεται η γείωση ως αποτέλεσμα της προηγούμενης κατάστασης κατά την οποία το σήμα στο θετικό κλάδο ήταν μικρότερο του αρνητικού. Σε αυτή την περίπτωση οι εν σειρά αντιστάσεις λειτουργούν ως διαιρέτες τάσης. Η τιμή του ενισχυμένου σήματος στην οποία ο δεύτερος, ενισχυτής, θα αλλάξει την έξοδό του από γείωση σε +5Volt ,δίνεται από τη σχέση.

$$\begin{aligned} V_{U1:2_in}^+ &= V_{U1:1_Out} - \frac{V_{U1:1_Out}}{R6 + R7} \cdot R6 = 2.5 = V_{U1:2_in}^- \Rightarrow \\ V_{U1:1_Out} \left(1 - \frac{R6}{R6 + R4}\right) &= 2.5 \Rightarrow V_{U1:1_Out} = \frac{2.5}{\left(1 - \frac{R6}{R6 + R4}\right)} \Rightarrow \\ V_{U1:1_Out} &= \frac{2.5}{\left(1 - \frac{1}{1+2}\right)} = 3.75 \text{ Volt} \end{aligned} \quad (1.35)$$

Επομένως όταν η τιμή εξόδου του σήματος από τον πρώτο ενισχυτή γίνει μεγαλύτερη ίση από 3,75 volt τότε αλλάξει η έξοδος του δεύτερου τελεστικού ενισχυτή γίνεται ίση με 5 Volt. Στη συνέχεια μόλις ξεπεράσει την τιμή αυτή διατηρείται η έξοδος του δεύτερου τελεστικού ενισχυτή στην τιμή αυτή.

➤ **Μείωση του σήματος.**

Όταν βρισκόμαστε στην κατάσταση κατά την οποία μειώνεται το σήμα τότε υποθέτεται ότι πριν βρισκόταν στην κατάσταση ανόδου. Επομένως η έξοδος του δεύτερου τελεστικού ενισχυτή U1:2 είναι $V_{U1:1_Out} = 5 \text{ Volt}$. Η ύπαρξη της ανατροφοδότησης του θετικού κλάδου με την έξοδό ,μέσω της αντίσταση R4 έχει ως συνέπεια τη διατήρηση της υπάρχουσας κατάστασης εξόδου εφόσον η πτώση τάσης στην R4 δεν είναι αρκετή ώστε στην είσοδο του θετικού κλάδου να εφαρμοστεί τάσης μικρότερη του $V_{U1:2_in}^- = 2,5 \text{ volt}$.

Η τιμή της εξόδου του σήματος από την ενίσχυση του πρώτου τελεστή U1:1 στην οποία θα συμβεί η παραπάνω μεταβολή δίνεται από την παρακάτω σχέση .

$$\left. \begin{aligned}
 V_{U1:2_Out} - I \cdot R4 &= 2.5 = V_{U1:2_in}^- \\
 I &= \frac{V_{U1:2_Out} - V_{U1:1_Out}}{R6 + R4} = \frac{5 - V_{U1:1_Out}}{(1+2) \cdot 10^3} \Rightarrow \\
 V_{U1:2_Out} &= 5, \quad V_{U1:2_in}^- = 2.5
 \end{aligned} \right\} \Rightarrow$$

$$V_{U1:2_Out} - \frac{V_{U1:2_Out} - V_{U1:1_Out}}{R6 + R4} \cdot R4 = 2.5 \Rightarrow$$

$$V_{U1:2_Out} \left(1 - \frac{R4}{R4 + R6}\right) + \frac{V_{U1:1_Out}}{R6 + R4} \cdot R4 = 2.5 \Rightarrow$$

$$\frac{V_{U1:1_Out}}{R6 + R4} \cdot R4 = 2.5 - V_{U1:2_Out} \left(1 - \frac{R4}{R4 + R6}\right) \Rightarrow$$

$$\frac{V_{U1:1_Out}}{1+2} \cdot 2 = 2.5 - 5 \left(1 - \frac{1}{1+2}\right) \Rightarrow$$

$$\frac{V_{U1:1_Out}}{1+2} \cdot 2 = 5 \cdot \left(\frac{1}{6}\right) \Rightarrow$$

$$V_{U1:1_Out} = V_{U1:2_Out} \cdot \frac{1}{6} \cdot \frac{3}{2} = \frac{V_{U1:2_Out}}{4} = 1.25 \text{ volt} \quad (1.36)$$

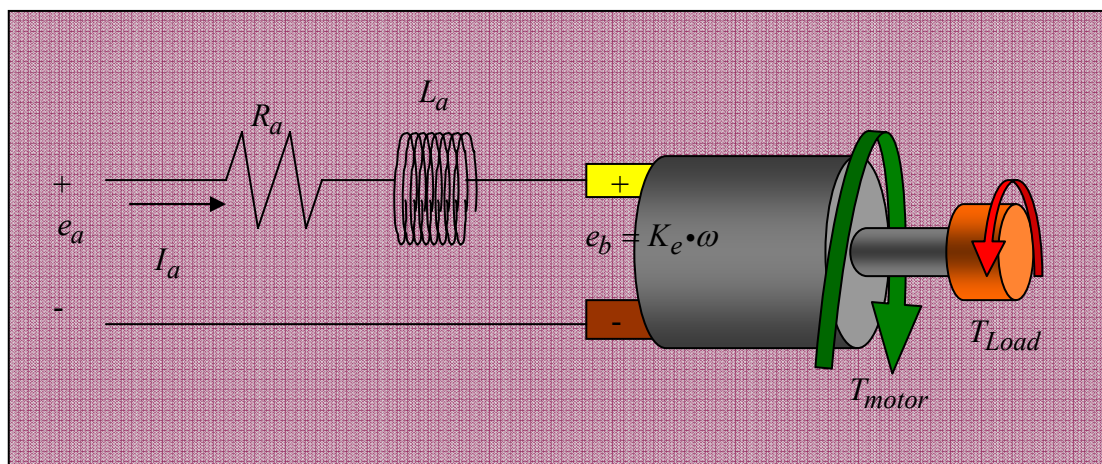
Επομένως δημιουργείται μια περιοχή του ενισχυμένου σήματος στην οποία η τελική έξοδος από το διπλό τελεστικό ενισχυτή «LM614BIM» διατηρείται σταθερή σύμφωνα με την προηγούμενη κατάσταση. Οποιαδήποτε μεταβολή στην τιμή ενισχυμένου σήματος δεν επιφέρει αλλαγή της τελικής εξόδου του, εφόσον η μεταβολή αυτή δε βρεθεί εκτός του πεδίου «threshold». Τα άκρα του πεδίου βρίσκονται από τις σχέσεις 1,62 και 1,63 και είναι (1.25,3.75).

Το παραπάνω πεδίο «threshold» αφορά το ενισχυμένο σήμα. Με τη χρήση της σχέσης 1.61 βρίσκεται το αντίστοιχο πεδίο για το σήμα από τον υπέρυθρο ανιχνευτή $\left(\frac{1.25}{2}, \frac{3.75}{2}\right) = (0.625, 1.875)$. Όταν η τιμή του σήματος ξεπεράσει την τιμή 1,875 τότε η έξοδος του τελεστικού ενισχυτή θα είναι +5 volt ενώ όταν πέσει κάτω από 0,625 είναι 0.

Το τελικό σήμα εξέρχεται από το διπλό τελεστικό ενισχυτή και στη συνέχεια εισέρχεται στον αντιστροφέα «Inverter» 74LVC14AD με σκοπό την τελική ρύθμιση του εξερχόμενου παλμού από το φίλτρο προς την κάρτα στα επιτρεπτά επίπεδα των 3,3 volt. Ο «Inverter» U2:1 δέχεται στη μία είσοδο του το σήμα από τον LM614BIM και εν συνεχεία αν το σήμα είναι μεγαλύτερο από την τάση τροφοδοσίας του «3.3 volt» δίνει στην έξοδο του 0 volt ενώ όταν η τιμή είναι 0 δίνει στην έξοδό του 3.3 volt. Δηλαδή κάνει αυτό που λει το όνομα του αντιστρέφει το σήμα και επιπλέον το ρυθμίζει στα επιθυμητά επίπεδα.

2) Ρυθμιστής Στροφών Ηλεκτρικού κινητήρα.

2.1) Αναγκαιότητα ρυθμιστή.



Σχήμα 2.1) Επίδραση της τάσης τροφοδοσίας και της ροπής του φορτίου στις στροφές του κινητήρα.

Ο σκοπός της κατασκευής του ρυθμιστή απορρέει από το γεγονός ότι ο έλεγχος των στροφών χρειάζεται για την πειραματική μέτρηση της ώσης. Η μέτρηση για να χαρακτηριστεί ως σωστή, πρέπει να διεξαχθεί υπό σταθερές στροφές της φτερωτής του προωθητήρα ανάδρασης και να διαρκέσει τόσο χρόνο ώστε οι μετρήσεις του «LoadCell» να θεωρηθεί ότι αντιπροσωπεύουν την ώση του προωθητήρα στην επιλεγμένη ταχύτητα του στροφείου.

Στους ηλεκτρικούς κινητήρες συνεχούς τάσης οι στροφές εξαρτώνται ευθέως ανάλογα από την τάση τροφοδοσίας τους. Δηλαδή η σχέση αναλογίας στροφών - τάσης περιγράφεται από τη σχέση :

$$RPM_{motor} = A_{Rpm/Volt} \cdot V_A \quad (2.1)$$

$A_{Rpm/Volt}$: Συντελεστή αναλογίας Στροφών ~ Τάσης Τροφοδοσίας

V_A : Συνεχή Τάση Τροφοδοσίας Κινητήρα.

Η εξάρτηση των στροφών του κινητήρα από την τάση θα χρησιμοποιηθεί ως το εργαλείο άσκησης της δράσης ελέγχου σε αυτόν. Θεωρητικά λοιπόν απλά ρυθμίζοντας την τάση, θα γνωρίζαμε τις στροφές, όμως η εξάρτηση αυτή δεν είναι απόλυτη. Οι στροφές του κινητήρα εξαρτώνται και από το φορτίο του, δηλαδή τη ροπή την οποία καλείτε ο κινητήρας να αναπτύξει. Στους κινητήρες συνεχούς τάσης παρατηρείται το φαινόμενο υστέρησης των πραγματικών στροφών, υπό φορτίο, σε σχέση με τις στροφές χωρίς αυτό. Κύριο γνώρισμα του φαινομένου αυτού είναι ότι ο κινητήρας υπό οποιαδήποτε φορτίο τείνει να φτάσει την ταχύτητα που θα είχε χωρίς αυτό. Το φαινόμενο αυτό ονομάζεται «Αυτορύθμιση Στροφών» του Κινητήρα «Speed

Regulation» .Η αυτορύθμιση δεν επηρεάζει την εξάρτηση των στροφών από την τάση με σχεδόν ευθεία αναλογία όπως εξηγήσαμε παραπάνω .

Η αυτορύθμιση (Killian CT, 2000; Leonhard W, 1997) είναι η δυνατότητα ενός κινητήρα να διατηρεί τη γωνιακή ταχύτητα περιστροφής του σταθερή όταν μεταβάλλεται το φορτίο που φέρει. Η γενεσιουργός αιτία που προκαλεί αυτό το φαινόμενο στους κινητήρες συνεχούς ρεύματος, είναι η αναπτυσσόμενη αντιηλεκτρεργετική δύναμη e_b . Ας υποθεθεί λοιπόν ότι ο κινητήρας καλείται να αντισταθμίσει μια ξαφνική αύξηση στο φορτίο του. Τότε όπως παρατηρείται , για συγκεκριμένη τάση τροφοδοσίας e_a , η αύξηση της μηχανικής ροπής προκαλεί ελάττωση της γωνιακής ταχύτητας περιστροφής ω , αυτό έχει ως συνέπεια τη μείωση της αναπτυσσόμενης αντιηλεκτρεργετική τάσης λόγω της γραμμικής τους σχέσης 1.13. Η μείωση όμως της αντιηλεκτρεργετική τάσης επιφέρει τελικά την αύξηση του ρεύματος τροφοδοσίας του κινητήρα, σύμφωνα με τη σχέση 1.4 από την οποία προκύπτει :

$$I_a = \frac{V_a - K_e \cdot \omega}{R_a} \quad (2.2)$$

Η αύξηση της τιμής του ρεύματος I_a είναι η αντίδραση του κινητήρα για την αύξηση της απαιτούμενης ροπής φορτίου, με αποτέλεσμα να αποτρέπεται η περαιτέρω επιβράδυνση του κινητήρα. Για κάθε κινητήρα υπάρχει μία σταθερά αναλογίας η οποία εκφράζει την παραγωγή ροπής του, συναρτήσει της έντασης του ρεύματος που το διαρρέει .

$$T_{motor} = K_I \cdot I_a \quad (2.3)$$

K_I : Σταθερά αναλογίας ροπής - έντασης ρεύματος (Nm / A) και η οποία είναι ανεξάρτητη της τάσης και των στροφών.

I_a : Ένταση ρεύματος του κινητήρα

Συνήθως η ικανότητα του κινητήρα σε μία δεδομένη κατάσταση φόρτισης ρυθμίζει τη γωνιακή ταχύτητα περιστροφής του κατά τρόπο που προαναφέρθηκε και εκφράζεται ως ποσοστό σύμφωνα με τη σχέση (Killian CT, 2000; Leonhard W, 1997).

$$Speed Regulation = \frac{\omega_{idle} - \omega_{Load}}{\omega_{Load}}, (\%) \quad (2.4)$$

ω_{idle} : Η γωνιακή ταχύτητα χωρίς το φορτίο υπό σταθερή τάση τροφοδοσίας V_A

ω_{Load} : Η γωνιακή ταχύτητα με το φορτίο υπό σταθερή τάση τροφοδοσίας V_A

Η ικανότητα αυτορύθμισης του κινητήρα ορίζεται από τα φυσικά χαρακτηριστικά του και δεν έχει σταθερή τιμή σε όλο το εύρος στροφών .Εξαρτάται από το φορτίο που καλείται να αντεπεξέλθει και από την ικανότητα του να αναπτύξει τη συγκεκριμένη ροπή υπό τη σταθερή τάση. Επομένως αν, υπό σταθερή τάση, το φορτίο είναι μικρό σε σχέση με το μέγιστο δυνατό αναπτυσσόμενο θα έχουμε περίπου

μικρό λόγο Speed Regulation ενώ όσο τείνουμε στο οριακό φορτίο ,για την τάση αυτή, ο λόγος θα αυξάνει . Αποτέλεσμα των παραπάνω είναι να εισέρχεται στη σχέση Στροφών - τάσης η τιμή της ροπής του φορτίου .

$$RPM_{motor} = A_{Rpm/Volt} \cdot V_A \cdot \left(1 - \frac{\omega_{idle} - \omega_{Load}(LOAD)}{\omega_{Load}(LOAD)}\right) \quad (2.5)$$

Από τα προαναφερθέντα είναι κατανοητό ότι πλέον οι στροφές δεν έχουν άμεση εξάρτηση μόνο από τη τάση τροφοδοσίας. Γνωρίζουμε ότι το κύριο τροφοδοτικό όλης της εγκατάστασης αποτελείται από μετασχηματιστή και ανορθωτή .Φυσικό χαρακτηριστικό του συστήματός αυτού είναι η εξάρτηση της τάσης εξόδου του από το φορτίο που καλείται να παραλάβει. Όσο αυξάνει το φορτίο τόσο η τάση εξόδου πέφτει ώστε η τελική εξαγόμενη ισχύς να παραμένει εντός των φυσικών ορίων σχεδιασμού του .

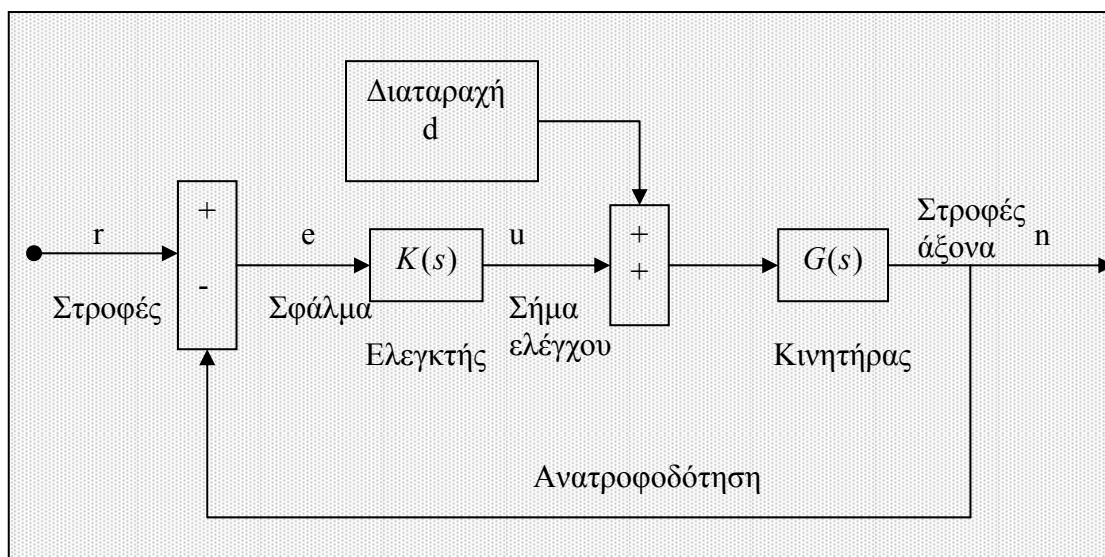
$$P_{out} = Con = V_{out} \cdot I_{out} \Leftrightarrow$$
$$V_{out} = \frac{P_{out}}{I_{out}} \quad (2.6)$$

Η ικανότητα παραγωγής ισχύος συνεχούς τάσης έχει προδιαγραφεί κατά την κατασκευή του και έχει προβλεφθεί ώστε η πτώση τάσης να καλύπτει τις λειτουργικές ανάγκες του κινητήρα ,εισάγει όμως στο σύστημα άλλη μία μεταβλητή παράμετρο .

Τελικά από τα παραπάνω κρίνεται ως αναγκαία η εισαγωγή και κατασκευή ενός ρυθμιστή στροφών ώστε οι μετρήσεις της ώσης να πραγματοποιηθούν υπό σταθερές στροφές, και κατά το δυνατόν οι παραπάνω αστάθμητοι παράγοντες να εξαλειφθούν . Θα μπορούσαμε να πραγματοποιήσουμε σειρά μετρήσεων χωρίς το ρυθμιστή στροφών σε διαφορετικές τάσης τροφοδοσίας αλλά τότε η επίδραση των μεταβλητών παραγόντων, τάσης τροφοδοτικού και αυτορύθμισης στροφών, δε μπορούν να εξομαλυνθούν .

2.2) Υπολογισμός της συνάρτησης μεταφοράς του συστήματος προωθητήρα ανάδρασης και ηλεκτρικού κινητήρα.

Το δομικό διάγραμμα του ρυθμιστή στροφών φαίνεται στο σχήμα .



Σχήμα 2.2)Δομικό διάγραμμα της εγκατάστασης πρόωσης.

- K : Ο ρυθμιστής στροφών.
- R : Το σημείο αναφοράς των επιθυμητών στροφών του συστήματος.
- e : Το σφάλμα ως η διαφορά ανάμεσα σε επιθυμητές και πραγματικές στροφές.
- u : Η δράση ελέγχου που παράγεται από το ρυθμιστή ($u = K(s) \cdot e$).
- d : Η διαταραχή στη δράση ελέγχου ως ο συνολικός θόρυβος.
- G : Η συνάρτηση μεταφοράς του συστήματος πρόωσης.

Ως είσοδο δέχεται τις επιθυμητές στροφές (setpoint) r . Υπολογίζεται το σφάλμα e ως η διαφορά των επιθυμητών στροφών r από τις πραγματικές στον άξονα n , το οποίο εισάγεται στον ελεγκτή $K(s)$ για να παραχθεί το σήμα ελέγχου. Το σήμα ελέγχου γίνεται δράση ελέγχου και προστίθεται στη εξωτερική διαταραχή και το αποτέλεσμα επιδρά στον κινητήρα $G(s)$ για να ρυθμίσει τις στροφές.

Στην προκειμένη περίπτωση του ρυθμιστή στροφών, η δράση ελέγχου που παράγεται από το σήμα ελέγχου ρυθμίζει την τάση τροφοδοσίας του ηλεκτρικού κινητήρα. Οι ελεγκτές που θα χρησιμοποιηθούν στη συγκεκριμένη εφαρμογή θα είναι τριών όρων, Αναλογικός – Ολοκληρωτικός - Διαφορικός, για τους λόγους που αναλύονται στις παρακάτω παραγράφους. Η εισαγωγή του ελεγκτή «PID» στη συνολική συνάρτηση μεταφοράς του συστήματος είναι:

Με βάση τις ιδιότητες του παραπάνω δομικού διαγράμματος προκύπτει η σχέση :

$$\begin{aligned}
 N(s) &= G(s) \cdot [U(s) + D(s)] \\
 U(s) &= K(s) \cdot [R(s) - N(s)] \Leftrightarrow \\
 N(s) &= G(s) \cdot [K(s) \cdot (R(s) - N(s)) + D(s)] \Leftrightarrow \\
 N(s) \cdot [1 + G(s) \cdot K(s)] &= G(s) \cdot K(s) \cdot R(s) + G(s) \cdot D(s) \Leftrightarrow \\
 N(s) &= \frac{G(s) \cdot K(s)}{1 + G(s) \cdot K(s)} \cdot R(s) + \frac{G(s)}{1 + G(s) \cdot K(s)} \cdot D(s) \quad (2.7)
 \end{aligned}$$

Στην τελευταία σχέση η απόκριση των στροφών του συστήματος εξαρτάται από δύο ανεξάρτητους παράγοντες, το επιθυμητό σημείο αναφοράς στροφών $R(s)$ και τη διαταραχή $D(s)$. Για το γραμμικό σύστημα αυτό η συνολική του απόκριση, στους δύο παραπάνω παράγοντες, είναι ίση με το άθροισμα των επιμέρους αποκρίσεων όταν ο ένας από του δύο παράγοντες είναι μηδενικός.

Δηλαδή το άθροισμα της απόκρισης στο σήμα $R(s)$ όταν $D(s)=0$ & της απόκρισης σε σήμα $D(s)$ όταν $R(s)=0$.

$$N_{R(s)=0}(s) = \frac{G(s) \cdot K(s)}{1 + G(s) \cdot K(s)} \cdot R(s) + \frac{G(s)}{1 + G(s) \cdot K(s)} \cdot 0 = \frac{G(s) \cdot K(s)}{1 + G(s) \cdot K(s)} \cdot R(s) \quad (2.8)$$

$$N_{R(s)=0}(s) = \frac{G(s) \cdot K(s)}{1 + G(s) \cdot K(s)} \cdot 0 + \frac{G(s)}{1 + G(s) \cdot K(s)} \cdot D(s) = \frac{G(s)}{1 + G(s) \cdot K(s)} \cdot D(s) \quad (2.9)$$

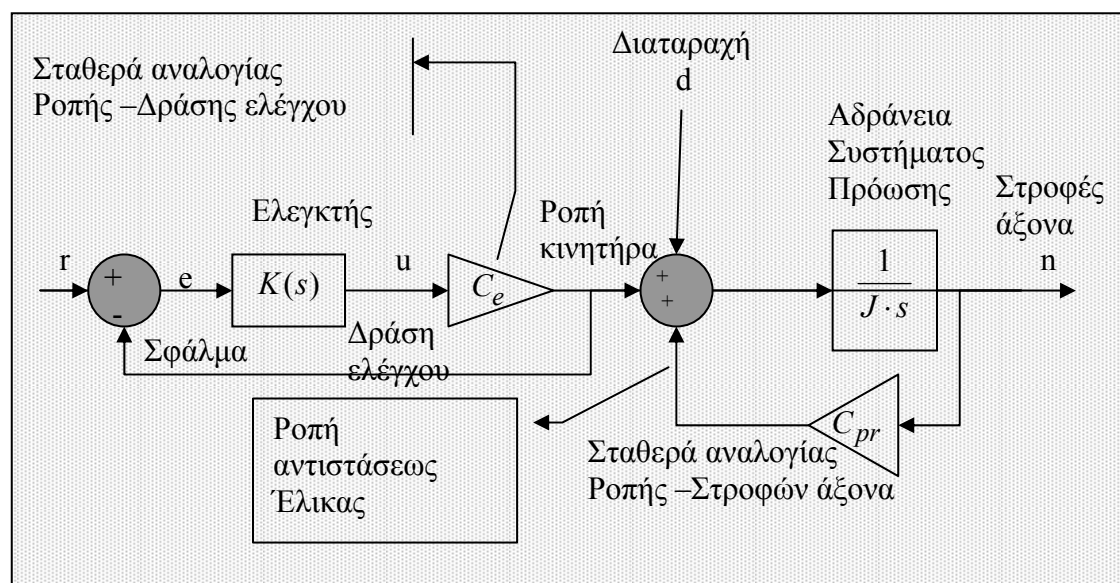
$N_{R(s)=0}(s)$: Συμβολίζεται η απόκριση του συνολικού συστήματος σε μεταβολή του επιθυμητού σημείου αναφοράς των στροφών με μηδενική διαταραχή στο σύστημα.

$N_{R(s)=0}(s)$: Συμβολίζεται η απόκριση του συνολικού συστήματος στη διαταραχή $D(s)$ με μηδενική μεταβολή επιθυμητού σημείου αναφοράς των στροφών.

Η τελευταία σχέση εκφράζει τη δυναμική συμπεριφορά του συστήματος.

$$N(s) = N_{R(s)=0}(s) + N_{R(s)=0}(s) \quad (2.10)$$

Επομένως η συνολική εκτίμηση της συμπεριφοράς του συνολικού συστήματος υπό την επίδραση και των δύο παραγόντων ταυτόχρονα ανάγεται στο άθροισμα των επιμέρους αποκρίσεων του συστήματος σε κάθε ένα παράγοντα ξεχωριστά. Το συνολικό σύστημα πρόωσης - ηλεκτρικού κινητήρα και ρυθμιστή στροφών με τη χρήση του μετασχηματισμού Laplace αναπαριστάτε με το παρακάτω σχήμα :



Σχήμα 2.3)Δομικό διάγραμμα της εγκατάστασης πρόωσης.

Η συνάρτηση μεταφοράς $G(s)$ του κινητήρα πρόωσης, όπως προκύπτει από το δομικό διάγραμμα του σχήματος 2.3 είναι :

$$G(s) = \frac{N(s)}{U(s) + D(s)} \quad (2.11)$$

Στο σχήμα, για απλότητα των υπολογισμών, ο μηχανισμός μετατροπής του σήματος ελέγχου σε δράση αναπαρίσταται με τη σταθερά C_e . Στο σημείο άθροισης των σημάτων βάσει των ιδιοτήτων του δομικού διαγράμματος ισχύει:

$$\begin{aligned} N(s) \cdot J \cdot s &= (U(s) + D(s)) \cdot C_e - N(s) \cdot C_{pr} \Leftrightarrow \\ N(s) \cdot [J \cdot s + C_{pr}] &= (U(s) + D(s)) \cdot C_e \Leftrightarrow \\ G(s) &= \frac{N(s)}{U(s) + D(s)} = \frac{C_e}{J \cdot s + C_{pr}} \end{aligned} \quad (2.12)$$

Φυσικά, η διαταραχή δεν εισέρχεται στο τμήμα που λογίζεται ως κινητήρας. Αντικαθιστώντας την τιμή των στροφών $N(s)$ στην 3.1 προκύπτει ότι:

$$G(s) = \frac{C_e}{J \cdot s + C_{pr}}$$

2.3) Γενικές Σημειώσεις επί της ανάλυσης του συνολικού συστήματος.

Όπως έχει αναφερθεί προηγουμένα η σχέση στροφών ροπής του κινητήρα και του προωθητήρα ανάδρασης δεν είναι γραμμική. Επομένως στα πλαίσια της γραμμικοποίησης του συνολικού συστήματος θεωρείται ότι το συνολικό σύστημα εξετάζεται υπό τη συνθήκη ότι βρίσκεται σε κατάσταση ηρεμίας σε μία θέση αναφοράς στροφών N_o και επιθυμητών στροφών R_o . Στο σημείο λειτουργίας αυτό μπορεί να θεωρηθεί ότι η ροπή του κινητήρα και προωθητήρα ανάδρασης είναι γραμμική συνάρτηση των στροφών N . Επομένως το συνολικό σύστημα θεωρείται γραμμικό και μπορούμε να προχωρήσουμε στη μελέτη του. Στην κατάσταση ηρεμίας θεωρείται ότι ισχύει η σχέση $R_o = N_o$. Άρα οι επιθυμητές στροφές μπορούν να γραφούν ως : $R = R_o + r = N_o + r$ όπου r η διαφορά τους από το σημείο αναφοράς R_o . Οι στροφές του κινητήρα γράφονται γύρω από το σημείο αναφοράς ως : $N = N_o + n = R_o + n$ όπου « n » η διαφορά των στροφών του κινητήρα και προωθητήρα από το σημείο ηρεμίας N_o .

Το σύστημα πλέον μπορεί να θεωρηθεί γραμμικό και εισάγοντας τις παραπάνω παραδοχές στη σχέση (2.9) έχουμε :

$$\left. \begin{aligned} N_o(s) + n(s) &= \frac{G(s) \cdot K(s)}{1 + G(s) \cdot K(s)} \cdot (R_o(s) + r(s)) + \frac{G(s)}{1 + G(s) \cdot K(s)} \cdot D(s) \\ N_o(s) &= \frac{G(s) \cdot K(s)}{1 + G(s) \cdot K(s)} \cdot R_o(s) \end{aligned} \right\} \Rightarrow$$

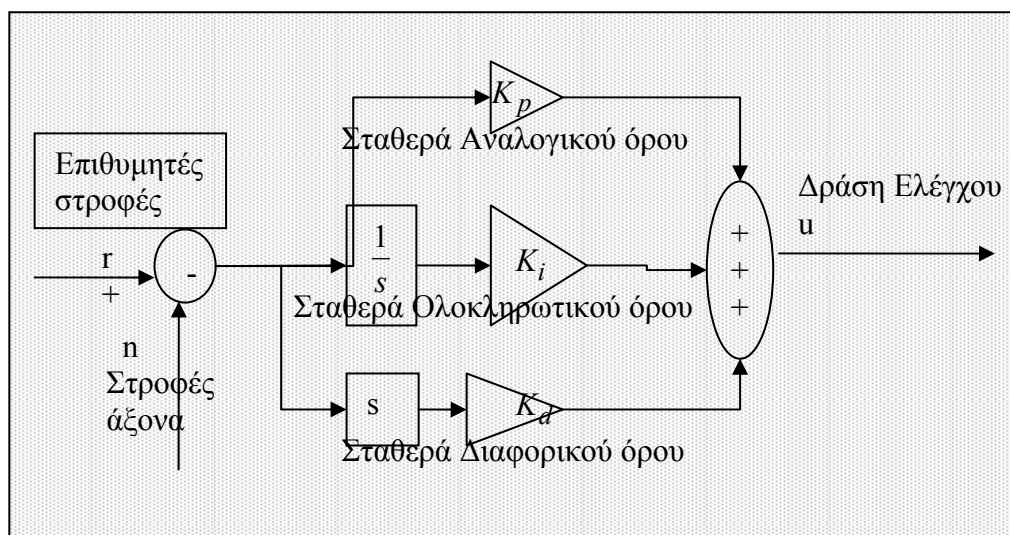
$$n(s) = \frac{G(s) \cdot K(s)}{1 + G(s) \cdot K(s)} \cdot r(s) + \frac{G(s)}{1 + G(s) \cdot K(s)} \cdot D(s) \quad (2.13)$$

Από εδώ και πέρα στην ανάλυση που θα ακολουθήσει χρησιμοποιούνται τα σύμβολα $n(s)$ & $r(s)$ που θα συμβολίζουν τις διαφορές από το σημείο αναφοράς και της γραμμικοποίησης .

2.4) Θεωρία και δομή των PID ελεγκτών.

Ο αλγόριθμος ελέγχου τριών όρων «PID – Proportional, Integral plus Derivate control» είναι ο πιο συνήθης στον έλεγχο αυτόματων διαδικασιών. Οι ελεγκτές ενός, δύο ή τριών όρων αποτελούν τον πυρήνα του σύγχρονου αυτοματισμού. Στόχος ενός ελεγκτή είναι ο συνεχής μηδενισμός του σφάλματος για κάθε συνθήκη λειτουργίας και κάθε εισαγόμενη διαταραχή. Ο στόχος αυτός είναι δύσκολος και σπάνια εφικτός, λόγω της αδράνειας και των μη γραμμικοτήτων της διαδικασίας. Κεφάλαιο 3ο: Μεθοδολογία συντονισμού ρυθμιστών στροφών P.I σελ.45.βιβλίο Συστήματα αυτομάτου ελέγχου.

Το δομικό διάγραμμα του γενικού ελεγκτή τριών όρων PID δίνεται στο παρακάτω σχήμα.



Σχήμα 2.5)Το δομικό διάγραμμα του γενικού ελεγκτή PID

Η δυναμική του συμπεριφορά περιγράφεται από μια αναλογική διαφορική και ολοκληρωτική εξίσωση, η κλασική μορφή της οποίας περιέχει τρεις όρους δηλαδή :

$$K(s) = K_p + \frac{1}{s} \cdot K_i + s \cdot K_d \quad (2.15)$$

Τα K_p, K_i, K_d είναι οι σταθερές του αναλογικού, ολοκληρωτικού και διαφορικού όρου του ελεγκτή. Οι τιμές τους καθορίζονται από τον κατασκευαστή του συστήματος έλεγχου και επηρεάζουν την ταχύτητα απόκρισης, και το εύρος του σφάλματος. Η συμπεριφορά των ελεγκτών εξαρτάται από την τιμή των συντελεστών αυτών και η οποία θα αναλυθεί στα επόμενα κεφάλαια .

Εάν με e συμβολιστεί το σφάλμα που εισέρχεται στον ελεγκτή και με u το σήμα ελέγχου τότε στο πεδίο της μιγαδικής συχνότητάς s το σήμα ελέγχου δίνεται από τη σχέση:

$$u(s) = K(s) \cdot e(s) = \left(K_p + \frac{1}{s} \cdot K_i + s \cdot K_d \right) \cdot e(s) \quad (2.16)$$

και η αντίστοιχη σχέση στο πεδίο του χρόνου t είναι:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\xi) d\xi + K_d \cdot \frac{de}{dt} \quad (2.17)$$

Το πρόβλημα σχεδίασης ενός ελεγκτή PID, που καλείται και συντονισμός, έγκειται στη βέλτιστη επιλογή των σταθερών K_p, K_i, K_d . Πολλές είναι οι μέθοδοι συντονισμού ελεγκτών PID που έχουν κατά καιρούς προταθεί από τις οποίες σε κάθε μία ικανοποιεί συγκεκριμένα κριτήρια, όπως για παράδειγμα βέλτιστη απόκριση, μεταβατική συμπεριφορά , ελάχιστο σφάλμα λειτουργίας κτλ.

2.5) Εφαρμογή αναλογικού ελεγκτή "P" στη συνάρτηση μεταφοράς του συστήματος.

Ο αναλογικός ρυθμιστής στροφών περιγράφεται από τη γενική σχέση (2.15) αν τεθούν ίσο με 0 οι συντελεστές του ολοκληρωτικού και διαφορικού όρου. Οπότε προκύπτει η παρακάτω έκφραση για τον αναλογικό ελεγκτή.

$$K(s) = K_p \quad (2.16)$$

2.5.1) Υπολογισμός της συνάρτησης μεταφοράς του συστήματος ρύθμισης στροφών "P" όταν στο σύστημα εφαρμόζεται διαταραχή.

Πρόβλημα Ρύθμισης , η τιμή αναφοράς των στροφών είναι σταθερή ($r(s)=0$) και η διαταραχή είναι μη μηδενική ($N_D(s)$).
 $R(s)=0$

Η συνάρτηση μεταφοράς της διαταραχής D υπολογίζεται από τη σχέση 2.8 θέτοντας $r(s)=0$ και λαμβάνοντας υπόψιν το κεφάλαιο 2.3 προκύπτει η σχέση:

$$GD(s) = \frac{n(s)}{D(s)} \xrightarrow{R(s)=0} GD(s) = \frac{G(s)}{1+G(s) \cdot K(s)} \quad (2.17)$$

Αντικαθιστώντας στην τελευταία σχέση τις τιμές των $G(s)$ από τη σχέση (2.12) και $K(s)$, προκύπτει η ζητούμενη συνάρτηση:

$$GD(s) = \frac{G(s)}{1+G(s) \cdot K(s)} \xrightarrow{K(s)=K_p, G(s)=\frac{C_e}{J \cdot s + C_{pr}}} GD(s) = \frac{C_e}{J \cdot s + (C_{pr} + K_p \cdot C_e)} \quad (2.18)$$

Η παραπάνω σχέση γράφεται και ως εξής:

$$GD(s) = \frac{C_e}{J \cdot s + (C_{pr} + K_p \cdot C_e)} = \frac{C_e}{s + \frac{(C_{pr} + K_p \cdot C_e)}{J}} \quad (2.19)$$

Για να είναι το σύστημα ευσταθές σύμφωνα με τη θεωρία αυτομάτου ελέγχου θα πρέπει η ρίζα του παρανομαστή της παραπάνω σχέσης να είναι αρνητική.

$$s + \frac{(C_{pr} + K_p \cdot C_e)}{J} = 0 \Leftrightarrow s = -\frac{(C_{pr} + K_p \cdot C_e)}{J} \quad (2.20)$$

$$\text{Αρκεί } (C_{pr} + K_p \cdot C_e) > 0 \Leftrightarrow K_p \cdot C_e > -C_{pr} \Leftrightarrow K_p > -\frac{C_{pr}}{C_e} \quad (2.21)$$

2.5.2) Υπολογισμός της συνάρτησης μεταφοράς του συστήματος ρύθμισης στροφών "P" όταν στο σύστημα εφαρμόζεται μεταβολή στις επιθυμητές στροφές.

Υπολογισμός της συνάρτησης μεταφοράς του συστήματος ρύθμισης στροφών όταν στο σύστημα εφαρμόζεται μεταβολή στις επιθυμητές στροφές ενώ η διαταραχή είναι μηδενική .($N_R(s)$)

Από τη σχέση 2.9 για $r(s)=0$ υπολογίζεται η συνάρτηση μεταφοράς του συστήματος για μεταβολή του επιθυμητού σημείου αναφοράς $r(s)$.

$$GR(s) = \frac{n(s)}{R(s)} \xrightarrow{D(s)=0} GR(s) = \frac{G(s) \cdot K(s)}{1+G(s) \cdot K(s)} \quad (2.22)$$

Αντικαθιστώντας στην τελευταία σχέση τις τιμές των $G(s)$ και $K(s)$, προκύπτει η ζητούμενη συνάρτηση :

$$GR(s) = \frac{G(s) \cdot K(s)}{1+G(s) \cdot K(s)} \xrightarrow{K(s)=K_p + \frac{1}{s} \cdot K_i, G(s)=\frac{C_e}{J \cdot s + C_{pr}}} GR(s) = \frac{C_e \cdot K_p}{J \cdot s + C_{pr} + K_p \cdot C_e} \quad (2.23)$$

Ομοίως όπως στη σχέση 2.21 η ευστάθεια του συστήματος εξαρτάται από τη σχέση :

$$(C_{pr} + K_p \cdot C_e) > 0 \Leftrightarrow K_p \cdot C_e > -C_{pr} \Leftrightarrow K_p > -\frac{C_{pr}}{C_e} \quad (2.24)$$

2.6) Εφαρμογή αναλογικού ολοκληρωτικού ελεγκτή "PI" στη συνάρτηση μεταφοράς του συστήματος.

Ο αναλογικός ρυθμιστής στροφών περιγράφεται από τη γενική σχέση 2.15 αν τεθεί ίσο με 0 ο συντελεστής του διαφορικού όρου. Οπότε προκύπτει η παρακάτω έκφραση για τον αναλογικό ελεγκτή.

$$K(s) = K_p + \frac{1}{s} \cdot K_i \quad (2.25)$$

2.6.1) Υπολογισμός της συνάρτησης μεταφοράς του συστήματος ρύθμισης στροφών "PI" όταν στο σύστημα εφαρμόζεται διαταραχή.

Πρόβλημα Ρύθμισης, η τιμή αναφοράς των στροφών είναι σταθερή ($r(s)=0$) και η διαταραχή είναι μη μηδενική ($N_D(s)$).

Όταν η τιμή αναφοράς είναι σταθερή και το σφάλμα στην έξοδο οφείλεται στη διαταραχή, το πρόβλημα ελέγχου ονομάζεται, ειδικότερα, πρόβλημα ρύθμισης (regulation problem).

Η συνάρτηση μεταφοράς της διαταραχής D υπολογίζεται από τη σχέση 2.8 θέτοντας $r(s)=0$ και η οποία είναι ίδια με τη σχέση 2.17.

$$GD(s) = \frac{G(s)}{1 + G(s) \cdot K(s)}$$

Αντικαθιστώντας στην τελευταία σχέση τις τιμές των G(s) και K(s) σχέση 2.25, προκύπτει η ζητούμενη σχέση :

$$\begin{aligned} GD(s) &= \frac{G(s)}{1 + G(s) \cdot K(s)} \xrightarrow{K(s)=K_p + \frac{1}{s} \cdot K_i} \\ & \xrightarrow{G(s)=\frac{C_e}{J \cdot s + C_{pr}}} \\ \rightarrow GD(s) &= \frac{s \cdot C_e}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e} \end{aligned} \quad (2.26)$$

Η παραπάνω σχέση γράφεται και ως εξής:

$$\begin{aligned} GD(s) &= C_e \cdot \frac{\frac{1}{J} \cdot s}{s^2 + \frac{(C_{pr} + K_p \cdot C_e)}{J} \cdot s + \frac{K_i \cdot C_e}{J}} \xrightarrow{\zeta = \frac{(C_{pr} + K_p \cdot C_e)}{2 \cdot \sqrt{J \cdot K_i \cdot C_e}}} \\ & \xrightarrow{\omega_n^2 = \frac{K_i \cdot C_e}{J}} \\ \rightarrow GD(s) &= C_e \cdot \frac{\frac{1}{J} \cdot s}{s^2 + 2 \cdot \zeta \cdot \omega_n + \omega_n^2} \end{aligned} \quad (2.27)$$

$$\text{με } \zeta = \frac{C_{pr} + K_p \cdot C_e}{2 \cdot \sqrt{J \cdot K_i \cdot C_e}} \quad \& \quad \omega_n^2 = \frac{K_i \cdot C_e}{J} .$$

Η παράμετρος ω_n καλείται κυκλική ιδιοσυχνότητα , $\omega_n > 0$, ενώ ζ ονομάζεται η σταθερά απόσβεσης του συστήματος αυτού .Όταν η σταθερά απόσβεσης είναι θετική το σύστημα είναι ευσταθές, όταν είναι μηδέν το σύστημα είναι οριακά ευσταθές, ενώ όταν είναι αρνητική το σύστημα παρουσιάζει αστάθεια. Για το λόγο αυτό στα συστήματα που κατασκευάζονται θα πρέπει η σταθερά απόσβεσης να είναι μεγαλύτερη του μηδενός. Εάν στην τελευταία σχέση τεθεί $\zeta=1$, τότε αυτή γίνεται :

$$GD(s) = C_e \cdot \frac{\frac{1}{J} \cdot s}{s^2 + 2 \cdot \omega_n + \omega_n^2} = C_e \cdot \frac{\frac{1}{J} \cdot s}{(s + \omega_n)^2} \quad (2.28)$$

2.6.2) Υπολογισμός της συνάρτησης μεταφοράς του συστήματος ρύθμισης στροφών "PI" όταν στο σύστημα εφαρμόζεται μεταβολή στις επιθυμητές στροφές .($\frac{N_R(s)}{D(s)=0}$)

Όταν η επιθυμητή τιμή μεταβάλλεται και η διαταραχή είναι μηδενική το πρόβλημα ελέγχου ονομάζεται, ειδικότερα, πρόβλημα παρακολούθησης (tracking problem) .

Από τη σχέση (2.13) για $r(s)=0$ υπολογίζεται η συνάρτηση μεταφοράς του συστήματος για μεταβολή του επιθυμητού σημείου αναφοράς $r(s)$.

$$GR(s) = \frac{n(s)}{R(s)} \xrightarrow{\frac{N_R(s)}{D(s)=0}} GR(s) = \frac{G(s) \cdot K(s)}{1 + G(s) \cdot K(s)}$$

Αντικαθιστώντας στην τελευταία σχέση τις τιμές των $G(s)$ και $K(s)$ στη σχέση 2.25, προκύπτει η ζητούμενη σχέση :

$$GR(s) = \frac{G(s) \cdot K(s)}{1 + G(s) \cdot K(s)} \xrightarrow{\begin{matrix} K(s) = K_p + \frac{1}{s} \cdot K_i \\ G(s) = \frac{C_e}{J \cdot s + C_{pr}} \end{matrix}} GR(s) = \frac{C_e \cdot K_p + \frac{1}{s} \cdot C_e \cdot K_i}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e} \Rightarrow$$

$$GR(s) = \frac{s \cdot C_e \cdot K_p + C_e \cdot K_i}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e} = \frac{C_e(s \cdot K_p + K_i)}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e} \quad (2.29)$$

Η παραπάνω σχέση γράφεται και ως εξής :

$$GR(s) = \frac{\frac{C_e(s \cdot K_p + K_i)}{J}}{s^2 + \frac{(C_{pr} + K_p \cdot C_e)}{J} \cdot s + \frac{K_i \cdot C_e}{J}} \xrightarrow{\zeta = \frac{(C_{pr} + K_p \cdot C_e)}{2 \cdot \sqrt{J \cdot K_i \cdot C_e}}, \omega_n^2 = \frac{K_i \cdot C_e}{J}} \rightarrow GR(s) = \frac{\frac{K_p}{K_i} \omega_n \cdot s + \omega_n}{s^2 + 2 \cdot \zeta \cdot \omega_n + \omega_n^2} \quad (2.30)$$

Ο παρανομαστής είναι ίδιος με της συνάρτησης μεταφοράς GD(s) σχέση 2.27 και επομένως ισχύουν και εδώ οι ίδιες παρατηρήσεις ως προς την ευστάθεια του συστήματος .

Όταν $\zeta > 0$ τότε το σύστημα είναι ευσταθές ενώ για $\zeta < 0$ το σύστημα είναι ασταθές . Στην ειδική περίπτωση που $\zeta = 1$ τότε έχουμε οριακή ευστάθεια .

$$\text{Για } \zeta = 1 \quad GR(s) = \frac{\frac{K_p}{K_i} \omega_n \cdot s + \omega_n}{(s + \omega_n)^2} \quad (2.31)$$

2.7) Εφαρμογή αναλογικού ολοκληρωτικού διαφορικού ελεγκτή "PID" στη συνάρτηση μεταφοράς του συστήματος.

Η σχέση του ρυθμιστή στροφών περιγράφεται από τη συνάρτηση σύμφωνα με τη σχέση 2.15 .

$$K(s) = K_p + \frac{1}{s} \cdot K_i + s \cdot K_d$$

2.7.1) Υπολογισμός της συνάρτησης μεταφοράς του συστήματος ρύθμισης στροφών "PID" όταν στο σύστημα εφαρμόζεται διαταραχή .

Πρόβλημα Ρύθμισης , η τιμή αναφοράς των στροφών είναι σταθερή ($r(s)=0$) και η διαταραχή είναι μη μηδενική ($N_D(s)$)
 $R(s)=0$

Από τη σχέση 2.13 για $r(s)=0$ υπολογίζεται η συνάρτηση μεταφοράς της διαταραχής D , $GD(s) = \frac{n(s)}{D(s)} \xrightarrow{\frac{N_D(s)}{R(s)=0}} GD(s) = \frac{G(s)}{1 + G(s) \cdot K(s)}$

Αντικαθιστώντας στην τελευταία σχέση τις τιμές των G(s) και K(s), προκύπτει η ζητούμενη σχέση:

$$GD(s) = \frac{G(s)}{1 + G(s) \cdot K(s)} \xrightarrow{K(s) = K_p + \frac{1}{s} \cdot K_i + s \cdot K_d, G(s) = \frac{C_e}{J \cdot s + C_{pr}}} \rightarrow$$

$$\rightarrow GD(s) = \frac{s \cdot C_e}{(J + C_e \cdot K_d) \cdot s^2 + (C_e \cdot K_p + C_{pr}) \cdot s + K_i \cdot C_e} \quad (2.32)$$

Η παραπάνω σχέση γράφεται και ως εξής:

$$GD(s) = \frac{s \cdot \frac{C_e}{(J + C_e \cdot K_d)}}{s^2 + \frac{(C_e \cdot K_p + C_{pr})}{(J + C_e \cdot K_d)} \cdot s + \frac{K_i \cdot C_e}{(J + C_e \cdot K_d)}} \xrightarrow{\zeta = \frac{(C_{pr} + K_p \cdot C_e)}{2 \cdot \sqrt{(J + C_e \cdot K_d) \cdot K_i \cdot C_e}}}$$

$$\omega_n^2 = \frac{K_i \cdot C_e}{J + C_e \cdot K_d}$$

$$GD(s) = \frac{s \cdot \frac{1}{K_i} \cdot \omega_n^2}{s^2 + 2 \cdot \zeta \cdot \omega_n \cdot s + \omega_n^2} \quad (2.33)$$

$$\text{με } \zeta = \frac{(C_e \cdot K_p + C_{pr})}{2 \cdot \sqrt{(J + C_e \cdot K_d) \cdot K_i \cdot C_e}} \text{ και } \omega_n^2 = \frac{K_i \cdot C_e}{(J + C_e \cdot K_d)}$$

Όταν η σταθερά απόσβεσης είναι θετική το σύστημα είναι ευσταθές, όταν είναι μηδέν το σύστημα είναι οριακά ευσταθές, ενώ όταν είναι αρνητική το σύστημα παρουσιάζει αστάθεια. Για το λόγο αυτό στα συστήματα που κατασκευάζονται θα πρέπει η σταθερά απόσβεσης να είναι μεγαλύτερη του μηδενός.

2.7.2) Υπολογισμός της συνάρτησης μεταφοράς του συστήματος ρύθμισης στροφών "PID" όταν στο σύστημα εφαρμόζεται μεταβολή στις επιθυμητές στροφές.

$$\cdot \left(\frac{N_R(s)}{D(s)=0} \right)$$

Από τη σχέση (2.13) για $r(s)=0$ υπολογίζεται η συνάρτηση μεταφοράς του συστήματος για μεταβολή του επιθυμητού σημείου αναφοράς $r(s)$

$$GR(s) = \frac{n(s)}{R(s)} \xrightarrow{\frac{N_R(s)}{D(s)=0}} GR(s) = \frac{G(s) \cdot K(s)}{1 + G(s) \cdot K(s)}$$

Αντικαθιστώντας στην τελευταία σχέση τις τιμές των $G(s)$ και $K(s)$, προκύπτει η ζητούμενη συνάρτηση :

$$GR(s) = \frac{G(s)}{1 + G(s) \cdot K(s)} \xrightarrow{\begin{matrix} K(s) = K_p + \frac{1}{s} \cdot K_i + s \cdot K_d \\ G(s) = \frac{C_e}{J \cdot s + C_{pr}} \end{matrix}}$$

$$\rightarrow GR(s) = \frac{C_e (K_i + K_p \cdot s + K_d \cdot s^2)}{(J + C_e \cdot K_d) \cdot s^2 + (C_e \cdot K_p + C_{pr}) \cdot s + K_i \cdot C_e} \quad (2.34)$$

Η παραπάνω σχέση γράφεται και ως εξής:

$$GR(s) = \frac{G(s)}{1+G(s) \cdot K(s)} \xrightarrow{K(s)=K_p+\frac{1}{s} \cdot K_i+s \cdot K_d} \frac{C_e(K_i+K_p \cdot s+K_d \cdot s^2)}{(J+C_{pr}) \cdot s} \rightarrow$$

$$GR(s) = \frac{C_e(K_i+K_p \cdot s+K_d \cdot s^2)}{(J+C_e \cdot K_d)} \cdot \frac{\zeta = \frac{(C_{pr}+K_p \cdot C_e)}{2 \cdot \sqrt{(J+C_e \cdot K_d) \cdot K_i \cdot C_e}}}{s^2 + \frac{(C_e \cdot K_p + C_{pr})}{(J+C_e \cdot K_d)} \cdot s + \frac{K_i \cdot C_e}{(J+C_e \cdot K_d)}} \xrightarrow{\omega_n^2 = \frac{K_i \cdot C_e}{J+C_e \cdot K_d}} \quad (2.35)$$

$$\rightarrow GR(s) = \frac{(K_i+K_p \cdot s+K_d \cdot s^2)}{K_i} \cdot \omega_n^2 \quad (2.36)$$

$$\rightarrow GR(s) = \frac{(K_i+K_p \cdot s+K_d \cdot s^2)}{s^2 + 2 \cdot \zeta \cdot \omega_n + \omega_n^2}$$

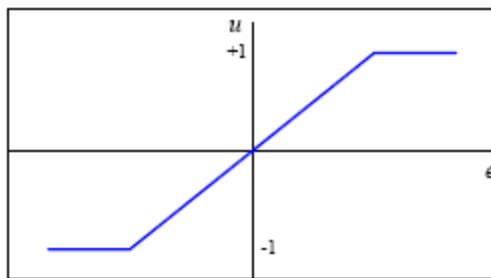
Ως προς την ευστάθεια προκύπτει ότι οι συντελεστές του χαρακτηριστικού πολυωνύμου του παρανομαστή μπορούν να πάρουν οποιαδήποτε επιθυμητή τιμή. Σημειώνεται ότι στην πράξη η χρήση του διαφορικού όρου αποφεύγεται για δύο λόγους. Καταρχήν το πολυώνυμο του αριθμητή έχει μεγαλύτερο βαθμό από εκείνο του παρανομαστή με αποτέλεσμα τη μη πραγματοποίηση του ελεγκτή με φυσικά στοιχεία και κατά δεύτερον ο διαφορικός όρος με την παραγωγή ενισχύει το θόρυβο που πάντα κάνει την εμφάνιση του στο σήμα ανατροφοδότησης. Οι υψίσυχνες συνιστώσες θορύβου που υπάρχουν στο σήμα, με σχετικά ασθενές πλάτος, ενισχύονται κατά την παραγωγή λόγω ύπαρξης του παράγοντα ω . Έτσι το διαφορικό είναι σε πολλές περιπτώσεις ένα τόσο εν θόρυβο σήμα ώστε είναι πρακτικά άχρηστο για εφαρμογές ελέγχου «Συστήματα αυτομάτου ελέγχου Κεφάλαιο 3.3».

2.8) Υπολογισμός του μονίμου σφάλματος του αναλογικού ρυθμιστή στροφών "P".

Στον αναλογικό έλεγχο (P-control) ισχύει ότι $K_p \neq 0, K_i = 0 \text{ \& } K_d = 0$. Έτσι το πρόβλημα συντονισμού περιορίζεται στον προσδιορισμό της σταθεράς K_p του νόμου ελέγχου. Η σχέση ορισμού ενός αναλογικού ελεγκτή, γύρω από ένα σημείο γραμμικοποίησης u_0 είναι:

$$u(t) = \begin{cases} u_0 + K_p \cdot e_0, & e(t) > e_0 \\ u_0 + K_p \cdot e(t), & -e_0 \leq e(t) \leq e_0 \\ u_0 - K_p \cdot e_0, & e(t) < -e_0 \end{cases} \quad (2.37)$$

Από την παραπάνω σχέση, προκύπτει το παρακάτω χαρακτηριστικό σχήμα αναλογικού ελεγκτή.



Σχήμα 2.6) Διάγραμμα αναλογικού ελεγκτή με κορεσμό.

Παρά το γεγονός ότι ο ιδανικός αναλογικός ελεγκτής (χωρίς κορεσμό) εξαλείφει το πρόβλημα των ταλαντώσεων οριακού κύκλου, δεν μπορεί να αποκλείσει την εμφάνιση σφάλματος .

Εξετάζεται καταρχήν η περίπτωση του ρυθμιστή στροφών για την οποία η διεργασία υποτίθεται ότι δεν έχει δυναμική και δεν υφίσταται διαταραχή .Δηλαδή εξετάζουμε το πρόβλημα παρακολούθησης όπως έχει ονομαστεί παραπάνω .

Η συνάρτηση μεταφοράς για το πρόβλημα αυτό έχει υπολογιστεί και εκφράζεται από τη σχέση (2.23) :

$$GR(s) = \frac{C_e \cdot K_p}{J \cdot s + C_{pr} + K_p \cdot C_e}$$

ισχύει ότι $e(s) = R(s) - N(s) = n(s) - r(s)$. Από τις δυο σχέσεις αυτές προκύπτει η έκφραση του σφάλματος $e(s)$,όπως έχει αναφερθεί στα πλαίσια της γραμμικοποίησης ,βλέπε κεφάλαιο 2.3, τα n & r εκφράζουν την απόκλιση από τα επιθυμητά σημεία αναφοράς .

Επομένως, εάν θεωρηθεί ότι η είσοδος (setpoint) είναι η βηματική συνάρτηση, τότε το σφάλμα είναι:

$$r(s) = \frac{1}{s} \text{ τότε το σφάλμα } e \text{ για } t \rightarrow \infty$$

$$GR(s) = \frac{n(s)}{r(s)} = \frac{C_e \cdot K_p}{J \cdot s + C_{pr} + K_p \cdot C_e} \Rightarrow n(s) = r(s) \cdot GR(s)$$

$$\Rightarrow n(s) = r(s) \cdot \frac{C_e \cdot K_p}{J \cdot s + C_{pr} + K_p \cdot C_e} \left. \vphantom{\frac{C_e \cdot K_p}{J \cdot s + C_{pr} + K_p \cdot C_e}} \right\} \Rightarrow$$

$$e(s) = n(s) - r(s)$$

$$e(s) = r(s) \cdot GD(s) - r(s) = r(s) \cdot \left(\frac{C_e \cdot K_p}{J \cdot s + C_{pr} + K_p \cdot C_e} - 1 \right)$$

$$e(s) = r(s) \cdot \left(\frac{\mathcal{L}_e \cdot K_p - J \cdot s + C_{pr} + K_p \cdot \mathcal{L}_e}{J \cdot s + C_{pr} + K_p \cdot C_e} \right) \quad (2.38)$$

Το μόνιμο σφάλμα υπολογίζεται από την παρακάτω σχέση σύμφωνα με τη θεωρία βλέπε Συστήματα Αυτομάτου Ελέγχου Ν.Ξηρός 2003.κεφ 1.2.

$$\lim_{t \rightarrow \infty} (e(t)) = \lim_{s \rightarrow 0} (s \cdot e(s)) = \lim_{s \rightarrow 0} \left(s \cdot \frac{1}{s} \frac{-J \cdot s + C_{pr}}{J \cdot s + C_{pr} + K_p \cdot C_e} \right) = \frac{C_{pr}}{C_{pr} + K_p \cdot C_e} \quad (2.39)$$

Το σφάλμα δε μηδενίζεται παρά μόνο μειώνεται με αύξηση του K_p .

Πρόβλημα ρύθμισης «regulation problem». Δηλαδή η τιμή αναφοράς είναι σταθερή και το σφάλμα στην έξοδο οφείλεται στη διαταραχή. Η συνάρτηση μεταφοράς για το πρόβλημα αυτό έχει υπολογιστεί και εκφράζεται από τη σχέση (2.18).

$$GD(s) = \frac{C_e}{J \cdot s + (C_{pr} + K_p \cdot C_e)} = \frac{C_e}{s + \frac{(C_{pr} + K_p \cdot C_e)}{J}} \quad (2.40)$$

ισχύει ότι $e(s) = R(s) - N(s) = n(s) - r(s) \xrightarrow{r(s)=0} e(s) = n(s)$. Από τις δυο σχέσεις αυτές προκύπτει η έκφραση του σφάλματος $e(s)$.

Επομένως, εάν θεωρηθεί ότι η είσοδος στη διαταραχή είναι η βηματική συνάρτηση, τότε το μόνιμο σφάλμα είναι: $D(s) = \frac{1}{s}$

$$\lim_{t \rightarrow \infty} (e(t)) = \lim_{s \rightarrow 0} (s \cdot e(s)) = \lim_{s \rightarrow 0} \left(s \cdot \frac{1}{s} \frac{C_e}{J \cdot s + C_{pr} + K_p \cdot C_e} \right) = \frac{C_e}{C_{pr} + K_p \cdot C_e} \quad (2.41)$$

Όπως φαίνεται από την παραπάνω σχέση το μόνιμο σφάλμα μπορεί να περιοριστεί με αύξηση του K_p αλλά δεν μπορεί να μηδενιστεί. Άλλωστε η τιμή του K_p δεν μπορεί να πάρει πολύ μεγάλες τιμές διότι η αύξηση του επιδρά αρνητικά τόσο στη δυναμική απόκριση κλειστού βρόχου, όσο και στην ευστάθεια του.

2.9) Υπολογισμός του μόνιμου σφάλματος του αναλογικού και ολοκληρωτικού ρυθμιστή στροφών "PI".

Η συνάρτηση μεταφοράς για το πρόβλημα αυτό έχει υπολογιστεί και εκφράζεται από τη σχέση 2.29.

$$GR(s) = \frac{n(s)}{r(s)} = \frac{C_e(s \cdot K_p + K_i)}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e}$$

και ισχύει ότι $e(s) = R(s) - N(s) = n(s) - r(s)$. Από τις δυο σχέσεις αυτές προκύπτει η έκφραση του σφάλματος $e(s)$.

$$\begin{aligned}
 GR(s) &= \frac{n(s)}{r(s)} = \frac{C_e(s \cdot K_p + K_i)}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e} \Rightarrow \\
 n(s) &= r(s) \cdot GR(s) = r(s) \cdot \frac{C_e(s \cdot K_p + K_i)}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e} \left. \vphantom{\frac{C_e(s \cdot K_p + K_i)}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e}} \right\} \Rightarrow \\
 e(s) &= n(s) - r(s) \\
 e(s) &= r(s) \cdot GD(s) - r(s) = r(s) \cdot \left(\frac{C_e(s \cdot K_p + K_i)}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e} - 1 \right) \Rightarrow \\
 e(s) &= r(s) \cdot \left(\frac{s \cdot \mathcal{L}_e \cdot K_p + K_i \cdot \mathcal{L}_e - J \cdot s^2 - s \cdot C_{pr} - s \cdot K_p \cdot \mathcal{L}_e - K_i \cdot \mathcal{L}_e}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e} \right) \Rightarrow \\
 e(s) &= r(s) \cdot \frac{-J \cdot s^2 - s \cdot C_{pr}}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e} \quad (2.42)
 \end{aligned}$$

Επομένως, εάν θεωρηθεί ότι η είσοδος «setpoint» είναι η βηματική συνάρτηση, τότε το σφάλμα είναι:

$$\begin{aligned}
 r(s) &= \frac{1}{s} \text{ τότε το μόνιμο σφάλμα } e \text{ δίνεται από τη σχέση :} \\
 \lim_{t \rightarrow \infty} (e(t)) &= \lim_{s \rightarrow 0} (s \cdot e(s)) = \lim_{s \rightarrow 0} \left(s \cdot \frac{1}{s} \frac{-J \cdot s^2 - s \cdot C_{pr}}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e} \right) \Rightarrow \\
 \lim_{t \rightarrow \infty} (e(t)) &= \frac{-J \cdot 0^2 - 0 \cdot C_{pr}}{J \cdot 0^2 + (C_{pr} + K_p \cdot C_e) \cdot 0 + K_i \cdot C_e} = 0 \quad (2.43)
 \end{aligned}$$

Δηλαδή το μόνιμο σφάλμα στην απόκριση του «PI» ρυθμιστή στροφών σε βηματική μεταβολή του επιθυμητού σημείου αναφοράς χωρίς την παρουσία διαταραχής είναι μηδενικό ,δηλαδή οι στροφές τείνουν στις επιθυμητές .

Εξετάζεται το πρόβλημα ρύθμισης (**regulation problem**) . Δηλαδή η τιμή αναφοράς είναι σταθερή και το σφάλμα στην έξοδο οφείλεται στη διαταραχή.

Η συνάρτηση μεταφοράς για το πρόβλημα αυτό έχει υπολογιστεί και εκφράζεται από τη σχέση 2.26 .

$$GD(s) = \frac{n(s)}{D(s)} = \frac{s \cdot C_e}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e}$$

και ισχύει ότι $e(s) = R(s) - N(s) = n(s) - r(s) \xrightarrow{r(s)=0} e(s) = n(s)$. Από τις δυο σχέσεις αυτές προκύπτει η έκφραση του σφάλματος $e(s)$.

$$e(s) = GD(s) \cdot D(s) = D(s) \cdot \frac{s \cdot C_e}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e} \quad (2.44)$$

Επομένως, εάν θεωρηθεί ότι η είσοδος στη διαταραχή είναι η βηματική συνάρτηση, τότε το σφάλμα είναι: $D(s) = \frac{1}{s}$

$$\lim_{t \rightarrow \infty} (e(t)) = \lim_{s \rightarrow 0} (s \cdot e(s)) = \lim_{s \rightarrow 0} \left(s \cdot \frac{1}{s} \cdot \frac{s \cdot C_e}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e} \right) \Rightarrow$$

$$\lim_{t \rightarrow \infty} (e(t)) = \lim_{s \rightarrow 0} (s \cdot e(s)) = \frac{0 \cdot C_e}{J \cdot 0^2 + (C_{pr} + K_p \cdot C_e) \cdot 0 + K_i \cdot C_e} = 0 \quad (2.45)$$

Δηλαδή το σύστημα αντιδρά στη διαταραχή D, παράγοντας κατάλληλο σήμα ελέγχου (σήμα δράσης) u που σταθεροποιεί την έξοδο n σε επιθυμητές στροφές R .

2.10) Υπολογισμός του μονίμου σφάλματος του αναλογικού ολοκληρωτικού και διαφορικού ρυθμιστή στροφών "PID".

Εξετάζεται το πρόβλημα παρακολούθησης (tracking problem). Η διεργασία υποτίθεται ότι δεν έχει δυναμική και δεν υφίσταται διαταραχή. Η συνάρτηση μεταφοράς για το πρόβλημα αυτό έχει υπολογιστεί και εκφράζεται από τη σχέση 2.34.

$$GR(s) = \frac{n(s)}{r(s)} = \frac{C_e(K_i + K_p \cdot s + K_d \cdot s^2)}{(J + C_e \cdot K_d) \cdot s^2 + (C_e \cdot K_p + C_{pr}) \cdot s + K_i \cdot C_e}$$

και ισχύει ότι $e(s) = R(s) - N(s) = n(s) - r(s)$. Από τις δυο σχέσεις αυτές προκύπτει η έκφραση του σφάλματος e(s) .

$$GR(s) = \frac{n(s)}{r(s)} = \frac{C_e(K_i + K_p \cdot s + K_d \cdot s^2)}{(J + C_e \cdot K_d) \cdot s^2 + (C_e \cdot K_p + C_{pr}) \cdot s + K_i \cdot C_e} \Rightarrow$$

$$\left. \begin{aligned} n(s) &= r(s) \cdot GR(s) = r(s) \cdot \frac{C_e(K_i + K_p \cdot s + K_d \cdot s^2)}{(J + C_e \cdot K_d) \cdot s^2 + (C_e \cdot K_p + C_{pr}) \cdot s + K_i \cdot C_e} \\ e(s) &= n(s) - r(s) \end{aligned} \right\} \Rightarrow$$

$$e(s) = r(s) \cdot GD(s) - r(s) = r(s) \cdot \left(\frac{C_e(K_i + K_p \cdot s + K_d \cdot s^2)}{(J + C_e \cdot K_d) \cdot s^2 + (C_e \cdot K_p + C_{pr}) \cdot s + K_i \cdot C_e} - 1 \right) \Rightarrow$$

$$e(s) = r(s) \cdot \left(\frac{K_i \cdot \mathcal{L}_e + s \cdot \mathcal{L}_e \cdot K_p + K_d \cdot s^2}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e} + \frac{-J \cdot s^2 - C_e \cdot K_d \cdot s^2 - s \cdot K_p \cdot \mathcal{L}_e - s \cdot C_{pr} - K_i \cdot \mathcal{L}_e}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e} \right) \Rightarrow$$

$$e(s) = r(s) \cdot \frac{-[(J + C_e \cdot K_d - K_d) \cdot s^2 + s \cdot C_{pr}]}{J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e} \quad (2.46)$$

Επομένως, εάν θεωρηθεί ότι η είσοδος «setpoint» είναι η βηματική συνάρτηση, τότε το σφάλμα είναι:

$r(s) = \frac{1}{s}$ τότε το μόνιμο σφάλμα e δίνεται από τη σχέση :

$$\lim_{t \rightarrow \infty} (e(t)) = \lim_{s \rightarrow 0} (s \cdot e(s)) = \lim_{s \rightarrow 0} \left(s \cdot \frac{1 - [(J + C_e \cdot K_d - K_d) \cdot s^2 + s \cdot C_{pr}]}{s J \cdot s^2 + (C_{pr} + K_p \cdot C_e) \cdot s + K_i \cdot C_e} \right) \Rightarrow$$

$$\lim_{t \rightarrow \infty} (e(t)) = \frac{-[(J + C_e \cdot K_d - K_d) \cdot 0^2 + 0 \cdot C_{pr}]}{J \cdot 0^2 + (C_{pr} + K_p \cdot C_e) \cdot 0 + K_i \cdot C_e} = 0 \quad (2.47)$$

Δηλαδή το μόνιμο σφάλμα στην απόκριση του PID ρυθμιστή στροφών σε βηματική μεταβολή του επιθυμητού σημείου αναφοράς είναι μηδενικό. Συνεπώς οι στροφές τείνουν στις επιθυμητές.

Εξετάζεται το πρόβλημα ρύθμισης «**regulation problem**». Δηλαδή η τιμή αναφοράς είναι σταθερή και το σφάλμα στην έξοδο οφείλεται στη διαταραχή. Η συνάρτηση μεταφοράς για το πρόβλημα αυτό έχει υπολογιστεί και εκφράζεται από τη σχέση 2.32.

$$GD(s) = \frac{n(s)}{D(s)} = \frac{s \cdot C_e}{(J + C_e \cdot K_d) \cdot s^2 + (C_e \cdot K_p + C_{pr}) \cdot s + K_i \cdot C_e}$$

και ισχύει $e(s) = R(s) - N(s) = n(s) - r(s) \xrightarrow{r(s)=0} e(s) = n(s)$. Από τις δυο σχέσεις αυτές προκύπτει η έκφραση του σφάλματος $e(s)$.

$$e(s) = GD(s) \cdot D(s) = D(s) \cdot \frac{s \cdot C_e}{(J + C_e \cdot K_d) \cdot s^2 + (C_e \cdot K_p + C_{pr}) \cdot s + K_i \cdot C_e} \quad (2.48)$$

Επομένως, εάν θεωρηθεί ότι η είσοδος στη διαταραχή είναι η βηματική συνάρτηση, τότε το σφάλμα είναι: $D(s) = \frac{1}{s}$

$$\lim_{t \rightarrow \infty} (e(t)) = \lim_{s \rightarrow 0} (s \cdot e(s)) = \lim_{s \rightarrow 0} \left(s \cdot \frac{1}{s} \cdot \frac{s \cdot C_e}{(J + C_e \cdot K_d) \cdot s^2 + (C_e \cdot K_p + C_{pr}) \cdot s + K_i \cdot C_e} \right) \Rightarrow$$

$$\lim_{t \rightarrow \infty} (e(t)) = \lim_{s \rightarrow 0} (s \cdot e(s)) = \frac{0 \cdot C_e}{(J + C_e \cdot K_d) \cdot 0^2 + (C_e \cdot K_p + C_{pr}) \cdot 0 + K_i \cdot C_e} = 0 \quad (2.49)$$

Δηλαδή το σύστημα αντιδρά στη διαταραχή D , παράγοντας κατάλληλο σήμα ελέγχου (σήμα δράσης) u που σταθεροποιεί την έξοδο n επιθυμητές στροφές R . Από την παραπάνω ανάλυση καταλήγουμε στο συμπέρασμα ότι η χρήση μόνο αναλογικού ελέγχου έχει ως συνέπεια την ύπαρξη σφάλματος στην τελική μόνιμη κατάσταση του. Δηλαδή την απόκλιση των τελικών στροφών σε σχέση με τις επιθυμητές. Η χρήση

ρυθμιστή στροφών (PI) & (PID) έχει ως συνέπεια την επίτευξη μηδενικού σφάλματος στροφών.

2.11) Μεταφορά των νόμων ελέγχου στον ψηφιακό επεξεργαστή.

Εισαγωγή

Οι πρώτοι βιομηχανικοί ελεγκτές τριών όρων κατασκευάζονταν με αναλογικές τεχνικές και περιείχαν πνευματικά ή υδραυλικά στοιχεία όπως φυσερά, βαλβίδες, έμβολα κ.λπ. Στα μέσα της δεκαετίας του '60 εμφανίστηκαν οι πρώτοι ψηφιακοί ελεγκτές σε σχήμα και λειτουργικότητα ακριβώς όμοια με τους προγενέστερους αναλογικούς ελεγκτές τους οποίους και αντικατέστησαν. Με την εμφάνιση των μικροεπεξεργαστών στα μέσα της δεκαετίας του '70, ο χώρος του αυτόματου ελέγχου άλλαξε ριζικά και οι ελεγκτές εμφανίστηκαν πλέον σε μορφή λογισμικού και χρονικά διακριτοποιημένων αλγορίθμων. Οι ψηφιακοί ελεγκτές, με τη μεγαλύτερη τους ακρίβεια και αξιοπιστία, αποτελούν πλέον τον πυρήνα του αυτόματου ελέγχου. Κατά το σχεδιασμό και υλοποίηση του αλγορίθμου ενός ψηφιακού ελεγκτή πρέπει να ληφθούν υπόψιν ορισμένοι βασικοί παράγοντες που δεν ισχύουν στη μορφή του ελεγκτή στο συνεχή χρόνο. Θέματα όπως η συχνότητα της δειγματοληψίας, η ψηφιοποίηση και η διακριτοποίηση στο χρόνο των μεταβλητών πρέπει να εξεταστούν προσεκτικά προκειμένου να μην προκύψουν προβλήματα κατά την εφαρμογή ελεγκτών. Η ευκολία προγραμματισμού, η ευελιξία που προσφέρει το λογισμικό και η μεγάλη ταχύτητα επεξεργασίας των σύγχρονων επεξεργαστών επιτρέπουν σύνθετους αλγορίθμους ελέγχου και ελεγκτές που ήταν αδιανόητοι πριν λίγα μόνο χρόνια.

2.11.1) Επεξεργασία Διακριτών μεταβλητών.

Οι πράξεις του αλγόριθμου εξαρτώνται από τις τιμές εισόδου και εξόδου της διαδικασίας σε τακτές χρονικές στιγμές. Συνεπώς οι τιμές των μεταβλητών της διαδικασίας πρέπει να μετριοούνται και να μετατρέπονται σε ψηφιακές με συχνότητα σχετικά υψηλή ώστε να μην υπάρχει απώλεια της πληροφορίας. Η δειγματοληψία των αναλογικών μεταβλητών της διαδικασίας αποτελεί τη βάση του σύγχρονου ψηφιακού ελέγχου διακριτού χρόνου (digital discrete time control).

Ως ψηφιακό σύστημα ελέγχου νοείται ένα σύστημα στο οποίο οι βασικές (μαθηματικές) λειτουργίες, συμπεριλαμβανομένου και του νόμου ελέγχου, υλοποιούνται σαν ένα ενσωματωμένο «embedded» πρόγραμμα που εκτελείται σε ένα ψηφιακό μικροεπεξεργαστή με προδιαγραφές πραγματικού χρόνου (real – time). Επειδή ο μικροεπεξεργαστής είναι ψηφιακός, η χρήση του έχει ένα σημαντικό συνακόλουθο: τα σήματα που μπορεί τόσο να επεξεργαστεί όσο και να παράγει δεν μπορούν να είναι συνεχούς χρόνου. Αντί αυτού επεξεργάζεται και παράγει πίνακες «διανύσματα» τιμών του κάθε σήματος εισόδου ή εξόδου. Οι πίνακες αυτοί περιέχουν τις τιμές των αντίστοιχων σημάτων σε διακεκριμένες χρονικές στιγμές και αναφέρονται με τον όρο σήματα διακριτού χρόνου. Οι διακεκριμένες χρονικές στιγμές στις οποίες οι τιμές των σημάτων συνεχούς χρόνο u είναι γνωστές είναι στη συντριπτική πλειοψηφία των περιπτώσεων ισαπέχουσες μεταξύ τους και μπορούν κατά συνέπεια να χαρακτηρισθούν σαν ακέραια πολλαπλάσια ενός βασικού χρονικού

διαστήματος που ονομάζεται περίοδος δειγματοληψίας. Με ελάχιστες εξαιρέσεις ο αλγόριθμος ενός ελεγκτή δύο όρων, σήμερα υλοποιείται με ψηφιακό επεξεργαστή. Η εκτέλεση του αλγόριθμου πραγματοποιείται σε διακριτό χρόνο και επαναληπτικά. Είναι ευνόητο ότι οι συνεχείς μεταβλητές της ελεγχόμενης διαδικασίας πρέπει πρώτα να μετατραπούν σε ψηφιακή μορφή για να είναι επεξεργάσιμες. Παρόμοια, ο συνεχής κλασικός αλγόριθμος τριών όρων, που αναπτύχθηκε στα προηγούμενα κεφάλαια, πρέπει τώρα να διακριτοποιηθεί. Ο επεξεργαστής αρχικά καταγράφει σε κάθε περίοδο δειγματοληψίας την τιμή της ελεγχόμενης μεταβλητής, η διαδικασία αυτή ονομάζεται διακριτοποίηση. Η δυαδική μορφή της μεταβλητής ταυτίζεται με τη ψηφιοποιημένη τιμή της και είναι αυτή που χρησιμοποιεί ο επεξεργαστής.

Στην περίπτωση του μετρητή στροφών και της εφαρμογής στην LPC2106 και της υλοποίησης του αλγόριθμου Έλεγχου PID, ορίζεται ως περίοδο δειγματοληψίας τα 20 msec, με σκοπό την επίτευξη του βέλτιστου δυνατού ρυθμού μετρήσεων και λήψης δεδομένων όπως θα εξηγηθεί σε παρακάτω κεφάλαια. Η διακριτοποίηση των στροφών υλοποιείται μέσω της χρήσης του «interrupt» της LPC2106. Το «interrupt» αυτό είναι επιφορτισμένο με το χρονοισμό της περιοδικής δειγματοληψίας των μετρούμενων μεγεθών και της μετάδοσης τους στον υπολογιστή. Για τη μέτρηση των στροφών χρησιμοποιείται κατάλληλη πύλη του επεξεργαστή η οποία παρακολουθεί το σήμα του υπέρυθρου ανιχνευτή και καταγράφει σε ψηφιακή μορφή της περιστροφές του σφονδύλου. Ομοίως και οι υπόλοιπες μετρήσεις. Τα σημεία αναφοράς στροφών και δράσης ελέγχου της LPC2106 λαμβάνονται απευθείας σε ψηφιακή μορφή. Αυτές οι διακριτοποιημένες ψηφιακές τιμές των μετρούμενων μεγεθών και σημείων αναφοράς χρησιμοποιούνται στο ψηφιακό επεξεργαστή της LPC2106 για την υλοποίηση του ελεγκτή P.I.D

2.11.2) Θεωρία του μετασχηματισμούς Z .

Υπενθυμίζεται ο ορισμός του μετασχηματισμού Z και παρουσιάζονται ορισμένες ιδιότητες οι οποίες θα χρησιμοποιηθούν παρακάτω κατά την παραγωγή και μεταφορά των συναρτήσεων του ελεγκτή P.I.D από το συνεχές χρόνο στο διακριτό. Περισσότερες πληροφορίες σχετικά με το μετασχηματισμό Z αναφέρονται στα Συστήματα Αυτομάτου Ελέγχου και Ρύθμισης Μηχανών ,Συμπληρωματικές σημειώσεις θεωρίας σημάτων χρόνου και συστημάτων διακριτού χρόνου Αθήνα 2003.

Ο μετασχηματισμός Z είναι ένας γραμμικός μετασχηματισμός που ορίζεται ως ακολούθως για ένα σήμα διακριτού χρόνου (ακολουθία) $x(n)$:

$$X(z) = Z\{x(n)\} = \sum_{n=0}^{+\infty} x(n) \cdot z^{-n} \quad (2.50)$$

Ο μετασχηματισμός Z χρησιμοποιείται για την απεικόνιση των μεταβλητών τη διακριτή χρονική στιγμή k στην προηγούμενη περίοδο δειγματοληψίας $k-1$. Δηλαδή:

$$y(n) = x(n-1) \xrightarrow{Z} Y(z) = z^{-1} X(z) \quad (2.51)$$

Μερικές από τις σημαντικότερες ιδιότητες δίνονται παρακάτω :

1) Γραμμικότητα

$$\left. \begin{array}{l} X_1(z) = Z\{x_1(n)\} \\ X_2(z) = Z\{x_2(n)\} \\ k_1, k_2 \in \mathfrak{R} \end{array} \right\} \Rightarrow k_1 \cdot X_1(z) + k_2 \cdot X_2(z) = Z\{k_1 \cdot x_1(n) + k_2 \cdot x_2(n)\} \quad (2.52)$$

2) Καθυστέρηση η προχώρηση κατά Μ μονάδες διακριτού χρόνου .

$$Z\{x(n)\} = X(z) \Rightarrow \left\{ \begin{array}{l} Z\{x(n-N)\} = z^{-N} \cdot X(z) \\ Z\{x(n+N)\} = z^N \cdot X(z) - \sum_{k=0}^{N-1} z^{N-k} \cdot x(k) \end{array} \right. \quad (2.53)$$

3) Θεώρημα διακριτής συνέλιξης .

$$\left. \begin{array}{l} X_1(z) = Z\{x_1(n)\} \\ X_2(z) = Z\{x_2(n)\} \\ x(n) = x_1(k) \otimes x_2(k) \triangleq \sum_{k=-\infty}^{+\infty} x_1(k) \cdot x_2(n-k) \end{array} \right\} \Rightarrow X(z) = Z(x(n)) = X_1(z) \cdot X_2(z) \quad (2.54)$$

2.12) Διακριτοποίηση του αναλογικού ελεγκτή "P".

Στο συνεχή χρόνο ο αναλογικός ελεγκτής γράφεται ως $P(t) = K_p \cdot e(t)$. Με χρήση του μετασχηματισμού Z ο αναλογικός ελεγκτής περιγράφεται από τη σχέση :

$$Z(P(t)) = Z(K_p \cdot e(t)) \Rightarrow P(z) = K_p \cdot e(z) \quad (2.55)$$

Επομένως ο διακριτοποιημένος αλγόριθμος του αναλογικού ελεγκτή δίνεται από τη σχέση :

$$u(k) = K_p \cdot e(k) \quad (2.56)$$

,όπου ο k ορίζει τον αριθμό της δειγματοληψίας του περιοδικού υπολογισμού της δράσης ελέγχου u .

2.13) Διακριτοποίηση του αναλογικού και ολοκληρωτικού ελεγκτή "PI".

Υπενθυμίζεται ότι η σχέση μεταξύ της εισόδου και της εξόδου ενός κλασικού ελεγκτή δύο όρων στο συνεχή χρόνο δίνεται από τη σχέση:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\xi) d\xi = P(t) + I(t) \quad (2.57)$$

Για τη μετατροπή της παραπάνω σχέσης στο διακριτό χρόνο, κάθε ένας από τους δύο παραπάνω όρους της, διακριτοποιείται και στη συνέχεια σχηματίζουν τον αλγόριθμο του ψηφιακού ελεγκτή δια της αθροίσεως τους.

Διακριτοποίηση του Αναλογικού όρου P .

Στο συνεχή χρόνο ο αναλογικός όρος του ελεγκτή ορίζεται ως : $P = K_p \cdot e(t)$. Ο αλγόριθμος διακριτού χρόνου έχει ήδη υπολογιστεί στο προηγούμενο κεφάλαιο υλοποιείται με απλή αντικατάσταση των συνεχών μεταβλητών με τις διακριτοποιημένες τιμές τους την ίδια στιγμή, δηλαδή $P(k) = K_p \cdot e(k), k = 0, 1, \dots$

Διακριτοποίηση του Ολοκληρωτικού όρου I .

Ο ολοκληρωτικός όρος του συνεχούς χρόνου ορίζεται από το ολοκλήρωμα:

$$I(t) = K_i \cdot \int_0^t e(\xi) d\xi$$

Ο υπολογισμός τους για το διακριτό χρόνο θα γίνει με δύο τρόπους .

α) Ο υπολογισμός του ολοκληρωτικού όρου θα πραγματοποιηθεί με τη χρήση της μεθόδου της Οπισθόδρομης διαφοράς (**Backwards difference**). Επομένως το ολοκλήρωμα γράφεται :

$$\frac{dI}{dt} \approx \frac{I(k) - I(k-1)}{T_s} = K_i \cdot e(k) \quad (2.59)$$

Σε επαναληπτική μορφή η παραπάνω εξίσωση γράφεται :

$$\begin{aligned} I(k) &= T_s \cdot K_i \cdot \sum_0^k e(k) \Rightarrow \\ \Rightarrow I(k) &= I(k-1) + T_s \cdot K_i \cdot e(k) \end{aligned} \quad (2.60)$$

όπου, T_s η περίοδος δειγματοληψίας .

Με εφαρμογή του μετασχηματισμού Z στον ολοκληρωτικό όρο $I(k)$ επί της σχέσης 2.60 προκύπτει η σχέση του Ολοκληρωτικού όρου I στο πεδίο του Z .

$$\left. \begin{aligned} I(k) &= K_i \cdot \int_0^t e(\xi) d\xi = K_i \cdot \int_0^{(n-1)T_s} e(\xi) d\xi + K_i \cdot \int_{(n-1)T_s}^{nT_s} e(\xi) d\xi = I(k-1) + \Delta I(k) \\ \Delta I(k) &= I(k) - I(k-1) = K_i \cdot T_s \cdot e(k) \end{aligned} \right\} \Rightarrow$$

$$\Rightarrow I(z) = \frac{T_s}{(1-z^{-1})} \cdot K_i \cdot e(z) \quad (2.61)$$

β) Υπολογισμός του ολοκληρώματος με τον κανόνα του τραπεζίου.

Υπενθυμίζεται ότι η σχέση του κανόνα τραπεζίου για την $I(t)$ μεταξύ δύο χρονικών στιγμών δίνεται από τη σχέση :

$$I(t) = K_i \cdot \int_{T_1}^{T_2} e(t) dt = K_i \cdot \left[\frac{e(T_2) + e(T_1)}{2} \right] \cdot [T_2 - T_1] \quad (2.62)$$

Επομένως η διακριτοποιημένη σχέση του ολοκληρωτικού όρου είναι :

$$\left. \begin{aligned} I(k) &= K_i \cdot \int_0^t e(\xi) d\xi = K_i \cdot \int_0^{(k-1)T_s} e(\xi) d\xi + K_i \cdot \int_{(k-1)T_s}^{kT_s} e(\xi) d\xi = I(k-1) + \Delta I(k) \\ \Delta I(k) &= K_i \cdot T_s \cdot \frac{e(k) + e(k-1)}{2} \end{aligned} \right\} \Rightarrow$$

$$I(k) = I(k-1) + K_i \cdot T_s \cdot \frac{e(k) + e(k-1)}{2} \quad (2.63)$$

Ο μετασχηματισμός Z της ολοκλήρωσης $I(s)$ με τον κανόνα του τραπεζίου της παραπάνω σχέσης είναι :

$$I(z) = \frac{T_s \cdot (1 + z^{-1})}{2 \cdot (1 - z^{-1})} \cdot K_i \cdot e(z) \quad (2.64)$$

Η συνολική μορφή του ψηφιακού αλγόριθμου του ελεγκτή P.I .

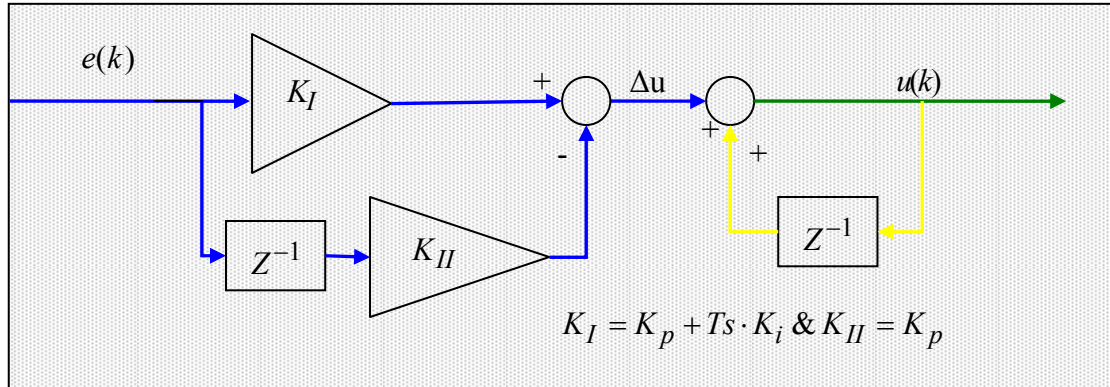
Η συνολική μορφή του ψηφιακού αλγόριθμου του P.I ελεγκτή, που θα υλοποιηθεί από το μικροεπεξεργαστή προκύπτει από την άθροιση των διακριτοποιημένων όρων του ελεγκτή $u(k) = P(k) + I(k)$. Για κάθε μία από τις δύο μεθόδους υπολογισμού του ολοκληρωτικού όρου θα προκύψουν διαφορετικές τελικές σχέσεις του αλγόριθμου του ελεγκτή.

A) Αλγόριθμος PI ελεγκτή με χρήση οπισθόδρομης διαφοράς της σχέσης 2.60.

$$\left. \begin{aligned} u(k) &= P(k) + I(k) \\ I(k) &= I(k-1) + T_s \cdot K_i \cdot e(k) \\ P(k) &= K_p \cdot e(k) \end{aligned} \right\} u(k) = K_p \cdot e(k) + I(k-1) + T_s \cdot K_i \cdot e(k) \quad (2.65)$$

B) Αλγόριθμος PI ελεγκτή με χρήση οπισθόδρομης διαφοράς της σχέσης 2.63.

$$\left. \begin{aligned} u(k) &= P(k) + I(k) \\ I(k) &= I(k-1) + K_i \cdot T_s \cdot \frac{e(k) + e(k-1)}{2} \\ P(k) &= K_p \cdot e(k) \end{aligned} \right\} u(k) = K_p \cdot e(k) + I(k-1) + T_s \cdot K_i \cdot e(k) \quad (2.66)$$



Σχήμα 2.13)Δομικό διάγραμμα του αναλογικού και ολοκληρωτικού ελεγκτή στροφών PI ,στο πεδίο του Z, με τη χρήση οπισθόδρομης διαφοράς του ολοκληρωτικού όρου I.

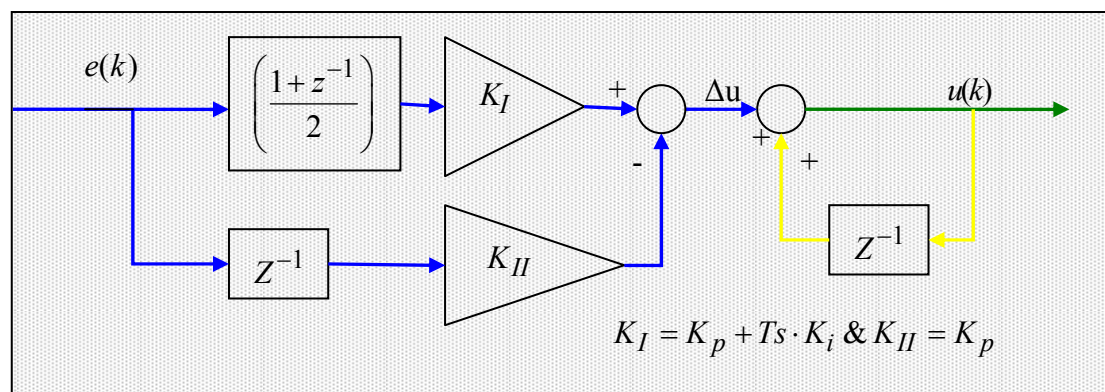
Ο όρος $I(k-1)$, που περιέχεται στις τελευταίες εξισώσεις 2.65 και 2.66 συσσωρεύει το σφάλμα $e(k)$. Όμως ο υπολογισμός του ολοκληρώματος δεν είναι εύκολο να πραγματοποιηθεί. Επιπλέον με την πάροδο του χρόνου το άθροισμα ίσως να αυξάνεται ανεξέλεγκτα. Εάν ληφθεί υπόψη ότι όλες οι πράξεις θα γίνουν από το μικροεπεξεργαστή και ότι οι θέσεις μνήμης του δεν έχουν άπειρη χωρητικότητα, τότε είναι πιθανό το άθροισμα κάποια στιγμή να υπερχειλίσει (overflow) και να καταστραφεί η ροή του αλγόριθμου .

Το $u(k)$ μπορεί να γραφεί και ως εξής, προκειμένου να αντικατασταθεί ο όρος $I(k-1)$, ο οποίος δεν μπορεί να αναπαρασταθεί κατά την υλοποίηση του αλγόριθμου:

$$\begin{aligned}
 u(k-1) &= P(k-1) + I(k-1) \xrightarrow{I(k)=I(k-1)+T_s \cdot K_i \cdot e(k)} \\
 u(k-1) &= P(k-1) + I(k) - T_s \cdot K_i \cdot e(k) \xrightarrow{I(k)=u(k)-P(k)} \\
 u(k-1) &= P(k-1) + u(k) - P(k) - T_s \cdot K_i \cdot e(k) \xrightarrow{P(k-1)=K_p \cdot e(k-1)} \\
 u(k-1) &= K_p \cdot e(k-1) + u(k) - K_p \cdot e(k) - T_s \cdot K_i \cdot e(k) \Rightarrow \\
 u(k) &= u(k-1) + K_p \cdot e(k) - K_p \cdot e(k-1) + T_s \cdot K_i \cdot e(k) \quad (2.67)
 \end{aligned}$$

Ομοίως με χρήση του ολοκληρωτικού τύπου β) κανόνα τραπεζίου προκύπτει η σχέση για τον αλγόριθμο του ελεγκτή:

$$u(k) = u(k-1) - K_p \cdot e(k-1) + K_p \cdot e(k) + K_i \cdot T_s \cdot \frac{e(k) + e(k-1)}{2} \quad (2.68)$$



Σχήμα 2.14)Δομικό διάγραμμα του αναλογικού και ολοκληρωτικού ελεγκτή στροφών PI ,στο πεδίο του Z, με τη χρήση του κανόνα του τραπεζίου στον υπολογισμό του ολοκληρωτικού όρου I.

Οι τελευταίες εξισώσεις υλοποιούνται στο μικροεπεξεργαστή για τη δημιουργία του σήματος ελέγχου των στροφών. Το δομικό διάγραμμα του σχήματος 2.13 υλοποιεί την παραπάνω σχέση 2.67 ενώ το σχήμα 2.14 υλοποιεί τη σχέση 2.68 .Από τα σχήματα 2.13 και 2.14 και από τον ορισμό του μετασχηματισμού Z στο κεφάλαιο 13.2 θα γίνει επαλήθευση των εξισώσεων 2.67 και 2.68.

Αρχικά για το μπλε τμήμα του σχήματος 2.13 ισχύει ότι :

$$E(z) \cdot (K_p + T_s \cdot K_i) - z^{-1} \cdot E(z) K_p = \Delta u(z) \quad (2.69)$$

Η σχέση 2.67 με τη χρήση του μετασχηματισμού Z γράφεται:

$$\begin{aligned} \Delta u = u(k) - u(k-1) &= K_p \cdot e(k-1) - K_p \cdot e(k) + T_s \cdot K_i \cdot e(k) \Rightarrow \\ Z\{\Delta u\} &= (K_p \cdot (1 - z^{-1}) + T_s \cdot K_i) \cdot e(z) \end{aligned} \quad (2.70)$$

Ομοίως για το σχήμα 2.14 η συνάρτηση μεταφοράς του μπλε τμήματος δίνεται από τη σχέση :

$$E(z) \cdot (K_p (1 - z^{-1}) + T_s \cdot K_i \cdot \frac{1 + z^{-1}}{2}) = \Delta u(z) \quad (2.71)$$

Από τη χρήση του μετασχηματισμού Z στη σχέση 2.68 προκύπτει η παρακάτω εξίσωση :

$$\begin{aligned} \Delta u = u(k) - u(k-1) &= -K_p \cdot e(k-1) + K_p \cdot e(k) + K_i \cdot T_s \cdot \frac{e(k) + e(k-1)}{2} \Rightarrow \\ Z\{\Delta u\} &= (K_p (1 - z^{-1}) + K_i \cdot T_s \cdot \frac{1 + z^{-1}}{2}) \cdot e(z) \end{aligned} \quad (2.72)$$

Συγκρίνοντας τις εξισώσεις 2.69 με την 2.70 και 2.71 με την 2.72 φαίνεται ότι η μεταβολή του σήματος ελέγχου μεταξύ δυο διαδοχικών περιόδων δειγματοληψίας είναι η ποσότητα των εξισώσεων 2.67 και 2.68 αντίστοιχα. Το μπλε τμήμα του δομικού διαγράμματος παράγει το νέο σήμα έλεγχου, το οποίο προστίθεται με αυτό της προηγούμενης περιόδου στο υπόλοιπο διάγραμμα :

$$\Delta U(z) + z^{-1} \cdot U(z) = U(z) \Rightarrow$$

$$\Delta u(k) + u(k-1) = u(z)$$

2.14) Διακριτοποίηση του ελεγκτή τριών όρων "P.I.D".

Η υλοποίηση του νόμου ελέγχου ενός βιομηχανικού ελεγκτή συνεχούς χρόνου σε λογισμικό για εκτέλεση με ψηφιακό υπολογιστή απαιτεί την προσέγγιση κάθε όρου του κλασικού αλγορίθμου τριών όρων .

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\xi) d\xi + K_d \frac{d}{dt} e(t)$$

Στις προηγούμενες παραγράφους των κεφαλαίων 2.12 και 2.13 υπολογίστηκαν οι δύο πρώτοι όροι ,αναλογικός και ολοκληρωτικός στο πεδίο του Z. Επομένως απαιτείται μόνο ο υπολογισμός του διαφορικού όρου ώστε να εξαχθεί η τελική διακριτοποιημένη σχέση του αλγορίθμου του ελεγκτή P.I.D.

Διακριτοποίηση του Ολοκληρωτικού όρου D .

Ο διαφορικός όρος στο συνεχή χρόνο ορίζεται από το διαφορικό :

$$D(t) = K_d \frac{d}{dt} e(t) \tag{2.73}$$

Ο υπολογισμός του διαφορικού όρου πραγματοποιείται με τη χρήση της μεθόδου της Οπισθόδρομης διαφοράς (Backwards difference).

$$\frac{d}{dt} e = \frac{e(k) - e(k-1)}{T_s} \tag{2.74}$$

όπου, T_s η περίοδος δειγματοληψίας .

Άρα η διακριτοποιημένη σχέση του διαφορικού όρου είναι :

$$D(k) = K_d \cdot \frac{e(k) - e(k-1)}{T_s} \tag{2.75}$$

Ο μετασχηματισμός Z της παραγώγου $\frac{d}{dt} e$ με τη μέθοδο της Οπισθόδρομης διαφοράς είναι :

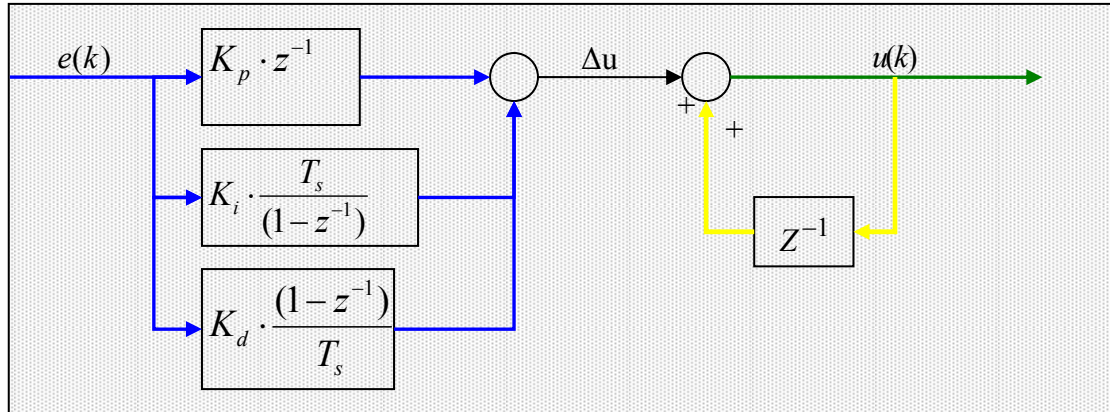
$$\frac{de}{dt} \simeq \frac{e(k) - e(k-1)}{T_s} = \frac{(1 - z^{-1})}{T_s} \cdot e(z) \tag{2.76}$$

Επομένως ο μετασχηματισμένος διαφορικός όρος δίνεται από τη σχέση

$$Z\{D(t)\} = Z\left\{K_d \frac{d}{dt} e(t)\right\} = K_d \cdot \frac{(1 - z^{-1})}{T_s} \cdot e(z) \tag{2.77}$$

Η συνολική σχέση του διακριτοποιημένου αλγόριθμου του P.I.D ελεγκτή δίνεται από την άθροιση των επιμέρους σχέσεων των όρων αυτού. Διακρίνονται δύο περιπτώσεις ανάλογα με τη χρησιμοποιούμενη σχέση για τον ολοκληρωτικό όρο.

A) Χρήση της οπισθόδρομης διαφορά για τον υπολογισμό του ολοκληρωτικού όρου.



Σχήμα 2.15)Δομικό διάγραμμα του ελεγκτή στροφών PID ,στο πεδίο του Z, με τη χρήση οπισθόδρομης διαφοράς του ολοκληρωτικού όρου I.

Η συνολική μορφή του ψηφιακού αλγόριθμου του PID ελεγκτή, που θα υλοποιηθεί από το μικροεπεξεργαστή προκύπτει από την άθροιση των σχέσεων 2.58, 2.60 και 2.75 .

$$u(k) = P(k) + I(k) + D(k) \xrightarrow{I(k)=I(k-1)+T_s \cdot K_i \cdot e(k)} \begin{matrix} P(k)=K_p \cdot e(k), D(k)=K_d \cdot \frac{e(k)-e(k-1)}{T_s} \\ \end{matrix}$$

$$\rightarrow u(k) = K_p \cdot e(k) + I(k-1) + T_s \cdot K_i \cdot e(k) + K_d \cdot \frac{e(k) - e(k-1)}{T_s} \quad (2.78)$$

Ο όρος $I(k-1)$ στην τελευταία σχέση δεν είναι επιθυμητός όπως έχει εξηγηθεί στο προηγούμενο κεφάλαιο και επομένως θα απαλειφθεί σύμφωνα με τις παρακάτω πράξεις.

$$u(k-1) = P(k-1) + I(k-1) + D(k-1) \Rightarrow$$

$$\Rightarrow u(k-1) = P(k-1) + I(k) - T_s \cdot K_i \cdot e(k) + K_d \cdot \frac{e(k-1) - e(k-2)}{T_s} \left. \vphantom{\Rightarrow} \right\} \Rightarrow$$

$$I(k) = u(k) - P(k) - D(k)$$

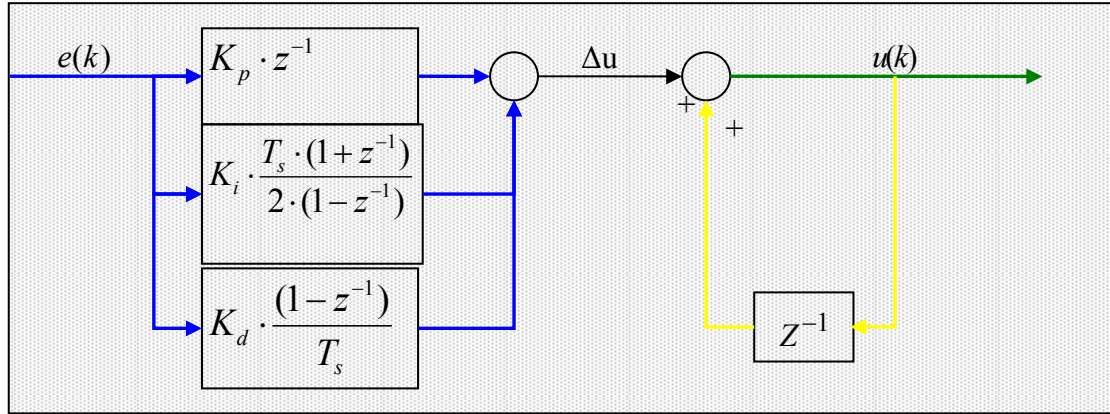
$$u(k-1) = P(k-1) + u(k) - P(k) - D(k) - T_s \cdot K_i \cdot e(k) \xrightarrow{P(k-1)=K_p \cdot e(k-1)}$$

$$u(k-1) = K_p \cdot e(k-1) + u(k) - K_p \cdot e(k) - K_d \cdot \frac{e(k) - e(k-1)}{T_s} -$$

$$T_s \cdot K_i \cdot e(k) + K_d \cdot \frac{e(k-1) - e(k-2)}{T_s} \Rightarrow$$

$$u(k) = u(k-1) - K_p \cdot e(k-1) + K_p \cdot e(k) + T_s \cdot K_i \cdot e(k) + K_d \cdot \frac{e(k) - 2 \cdot e(k-1) + e(k-2)}{T_s} \quad (2.79)$$

B) Χρήση του κανόνα του τραπεζιού για τον υπολογισμό του ολοκληρωτικού όρου.



Σχήμα 2.16) Δομικό διάγραμμα του ελεγκτή στροφών PID ,στο πεδίο του Z, με τη χρήση του κανόνα τραπεζιού στον υπολογισμό του ολοκληρωτικού όρου I.

Η συνολική μορφή του ψηφιακού αλγόριθμου του P.I.D ελεγκτή, που θα υλοποιηθεί από το μικροεπεξεργαστή προκύπτει από την άθροιση των σχέσεων 2.58, 2.63 και 2.75.

$$u(k) = P(k) + I(k) + D(k) \rightarrow \begin{aligned} P(k) &= K_p \cdot e(k), D(k) = K_d \cdot \frac{e(k) - e(k-1)}{T_s} \\ I(k) &= I(k-1) + K_i \cdot T_s \cdot \frac{e(k) + e(k-1)}{2} \end{aligned}$$

$$\rightarrow u(k) = K_p \cdot e(k) + I(k-1) + K_i \cdot T_s \cdot \frac{e(k) + e(k-1)}{2} + K_d \cdot \frac{e(k) - e(k-1)}{T_s} \quad (2.80)$$

Για την απαλοιφή του όρου $I(k-1)$ της παραπάνω σχέσης ακολουθείται όμοια σειρά πράξεων όπως στην προηγούμενη παράγραφο για την εξαγωγή της σχέσης 2.79.

$$u(k) = u(k-1) + K_p \cdot (e(k) - e(k-1)) + K_i \cdot T_s \cdot \frac{e(k) + e(k-1)}{2} + K_d \cdot \frac{e(k) - 2 \cdot e(k-1) + e(k-2)}{T_s} \quad (2.81)$$

Για την επαλήθευση των σχέσεων 2.79 και 2.81 υπολογισμού του ελεγκτή στροφών PID θα χρησιμοποιηθεί ο μετασχηματισμός Z.

A) Επαλήθευση του αλγόριθμου υπολογισμού του ελεγκτή της σχέσης 2.79.

$$Z\{u(k)\} = Z\left\{u(k-1) - K_p \cdot e(k-1) + K_p \cdot e(k) + T_s \cdot K_i \cdot e(k) + K_d \cdot \frac{e(k) - 2 \cdot e(k-1) + e(k-2)}{T_s}\right\} \Rightarrow$$

$$\begin{aligned}
 \Rightarrow u(z) &= z^{-1} \cdot u(z) - K_p \cdot z^{-1} \cdot e(z) + K_p \cdot e(z) + \\
 &+ T_s \cdot K_i \cdot e(z) + K_d \cdot \frac{e(z) - z^{-1} \cdot e(z)}{T_s} - K_d \cdot \frac{z^{-1} \cdot e(z) - z^{-2} \cdot e(z)}{T_s} \Rightarrow \\
 u(z) \cdot (1 - z^{-1}) &= K_p \cdot (1 - z^{-1}) \cdot e(z) + T_s \cdot K_i \cdot e(z) + K_d \cdot \frac{(1 - z^{-1})}{T_s} \cdot e(z) - K_d \cdot \frac{(1 - z^{-1}) \cdot z^{-1}}{T_s} \cdot e(z) \Rightarrow \\
 u(z) \cdot (1 - z^{-1}) &= K_p \cdot (1 - z^{-1}) \cdot e(z) + T_s \cdot K_i \cdot e(z) + K_d \cdot \frac{(1 - z^{-1})(1 - z^{-1})}{T_s} \cdot e(z) \Rightarrow \\
 u(z) &= K_p \cdot e(z) + K_i \cdot \frac{T_s}{(1 - z^{-1})} \cdot e(z) + K_d \cdot \frac{(1 - z^{-1})}{T_s} \cdot e(z) \quad (2.82)
 \end{aligned}$$

Χρησιμοποιώντας τους αντίστροφους μετασχηματισμούς Z , $u(z) = u_\delta(s)$
 $e(z) = e_\delta(s)$, $I_\delta(s) = \frac{T_s}{(1 - z^{-1})} \cdot K_i \cdot e(z)$, $D_d(s) = K_d \cdot \frac{(1 - z^{-1})}{T_s} \cdot e(z)$ &
 $P_\delta(s) = K_p \cdot e(z)$ επί της σχέσης 2.82 καταλήγουμε :

$$\left. \begin{aligned}
 u(s) &= K_p e(s) + I(s) + K_d e(s) \cdot \frac{(1 - z^{-1})}{T_s} \cdot e(z) \\
 X(z) \Big|_{z^{-1} = \exp(-sT_s)} &= X_\delta(s) \\
 \Rightarrow u_\delta(s) &= P_\delta(s) + I_\delta(s) + D_\delta(s)
 \end{aligned} \right\} \Rightarrow \quad (2.83)$$

Δηλαδή η αλγόριθμός **επαληθεύεται** .

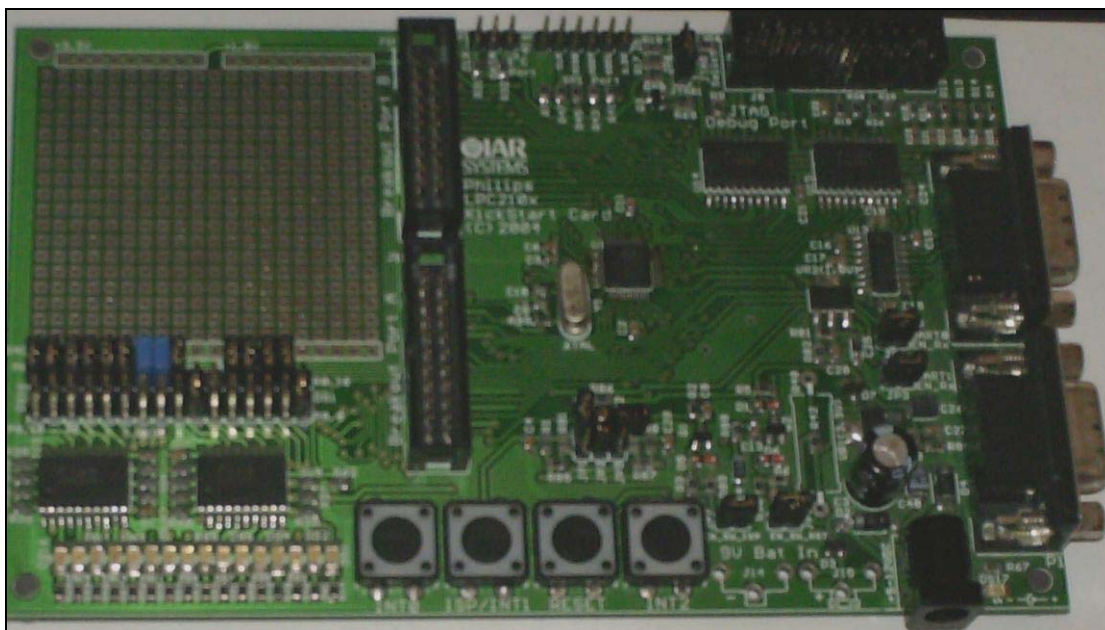
B) Επαλήθευση του αλγόριθμου υπολογισμού του ελεγκτή της σχέσης 2.81.

Η σχέση 2.81 μετασχηματίζεται με τη χρήση του Z όμοια με το μετασχηματισμό της εξίσωσης 2.79 στην προηγούμενη παράγραφο. Εκτελώντας τις πράξεις καταλήγουμε στην τελική σχέση :

$$u(z) = K_p \cdot e(z) + K_i \cdot \frac{T_s \cdot (1 + z^{-1})}{2 \cdot (1 - z^{-1})} \cdot e(z) + K_d \cdot \frac{(1 - z^{-1})}{T_s} \cdot e(z) \quad (2.84)$$

Ομοίως χρησιμοποιώντας τους αντίστροφους μετασχηματισμούς, $u_\delta(z) = u(s)$,
 $e_\delta(z) = e(s)$, $P_\delta(s) = K_p \cdot e(z)$, $I_\delta(s) = \frac{T_s \cdot (1 + z^{-1})}{2 \cdot (1 - z^{-1})} \cdot K_i \cdot e(z)$, $D_d(s) = K_d \cdot \frac{(1 - z^{-1})}{T_s} \cdot e(z)$
 επί της σχέσης 2.84 αποδεικνύεται ότι ισχύει η σχέση
 $u_\delta(s) = P_\delta(s) + I_\delta(s) + D_\delta(s)$, δηλαδή στην **επαλήθευση** της σχέσης 2.81.

3) Αρχιτεκτονική και συνδεσμολογία κάρτας IAR.



3.0) Εισαγωγή.

Ο μικροεπεξεργαστής είναι ένα στοιχείο LSI (Large Scale Intergration) που υλοποιεί τις περισσότερες από τις λειτουργίες ενός παραδοσιακού επεξεργαστή με ένα και μόνο chip. Ιστορικά ο πρώτος 8 – bits μικροεπεξεργαστής εισήχθη στην αγορά από την Intel το 1972 για τον έλεγχο οθονών CRT. Η παρούσα γενιά σχεδίασης μικροεπεξεργαστών άρχισε το 1983 με την εισαγωγή του πρώτου 32 – bits μικροεπεξεργαστή από την National Semiconductors.

Τα τρία ουσιώδη πλεονεκτήματα των μικροεπεξεργαστών, που συνετέλεσαν στη ραγδαία εξάπλωση τους είναι ,πρώτον απαιτούν λιγότερα εξαρτήματα, δεύτερον αποτελούν μια λύση χαμηλού κόστους και τρίτον προσφέρουν την ευελιξία του προγραμματισμού. Ο μικρός αριθμός των εξαρτημάτων που απαιτεί ένα σύστημα με μικροεπεξεργαστή οδηγεί στα παρακάτω συγκεκριμένα πλεονεκτήματα :

- Μειώνει το φυσικό όγκο και καταλήγει στην ελαχιστοποίηση των διαστάσεων.
- Μειωμένη κατανάλωση ισχύος και μειωμένη σπατάλη ενέργειας.
- Μεγαλύτερη αξιοπιστία λόγω του μικρότερου αριθμού εξαρτημάτων.

Τα παραπάνω χαρακτηριστικά καταλήγουν σε ένα σημαντικό αποτέλεσμα: Το χαμηλό κόστος. Τα ουσιαστικά πλεονεκτήματα του προγραμματισμού είναι ότι απλοποιεί το σχεδιασμό, ελαχιστοποιεί το χρόνο ανάπτυξης και επιτρέπει εύκολες μετατροπές. Τα πλεονεκτήματα αυτά έχουν οδηγήσει στην ευρύτερη χρήση των ηλεκτρονικών μικροεπεξεργαστών στον τομέα του αυτόματου ελέγχου. Στα πλαίσια

λοιπόν της πτυχιακής θα χρησιμοποιήσουμε την ηλεκτρονική κάρτα IAR LPC2106, η οποία παρέχει μεγάλη επεξεργαστική ισχύ και πλήθος λειτουργιών χρήσιμες για διάφορες εργασίες.

3.1) Παρουσίαση των δυνατοτήτων και των χαρακτηριστικών της LPC2106.

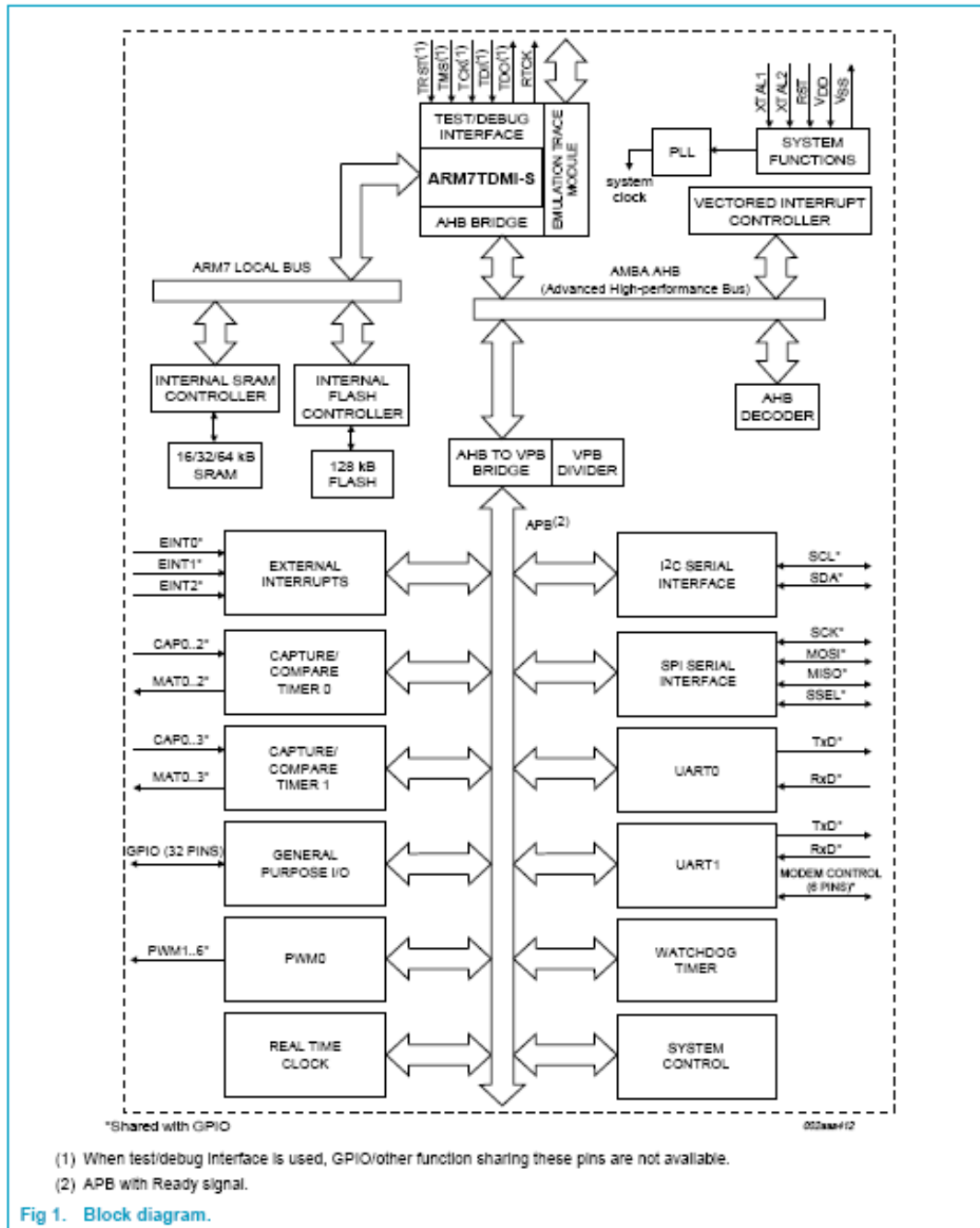
Η κάρτα LPC2106 είναι βασισμένη στον επεξεργαστή 16/32 bit ARM7TDMI-S μαζί με 128 kbytes (kB) υψηλής ταχύτητας μνήμης . Η 128bit εύρους διασύνδεση μνήμης και η αρχιτεκτονική επιτάχυνσης καθιστούν δυνατή την εκτέλεση 32bit κώδικα σε μέγιστη ταχύτητα επεξεργαστή. Η χρήση ιδιαίτερα μεγάλου και απαιτητικού κώδικα παρέχει τη δυνατότητα της εκτέλεση του υπό τη λειτουργία του επεξεργαστή σε16-bit Thumb, μειώνοντας το μέγεθος του κώδικα κατά 30% με ελάχιστο αντίκτυπο στην απόδοση.

Διαθέτει μεγάλο εύρος σειριακής διασύνδεσης με τον επεξεργαστή ικανό να χρησιμοποιηθεί ως μετατροπείς πρωτοκόλλων επικοινωνίας, αναγνώρισης φωνής και επεξεργασία εικόνας παρέχοντας ταυτόχρονα μεγάλο μέγεθος διαθέσιμη μνήμης και υψηλής επεξεργαστικής ισχύς. Διαθέτει διάφορους 32bit χρονομετρητές, κανάλια παλμών PWM και 32 εισόδους εξόδους γενικής χρήσης που την καθιστούν ιδανική για χρήση στο βιομηχανικό έλεγχο.

Τα κυριότερα χαρακτηριστικά της LPC2106 είναι :

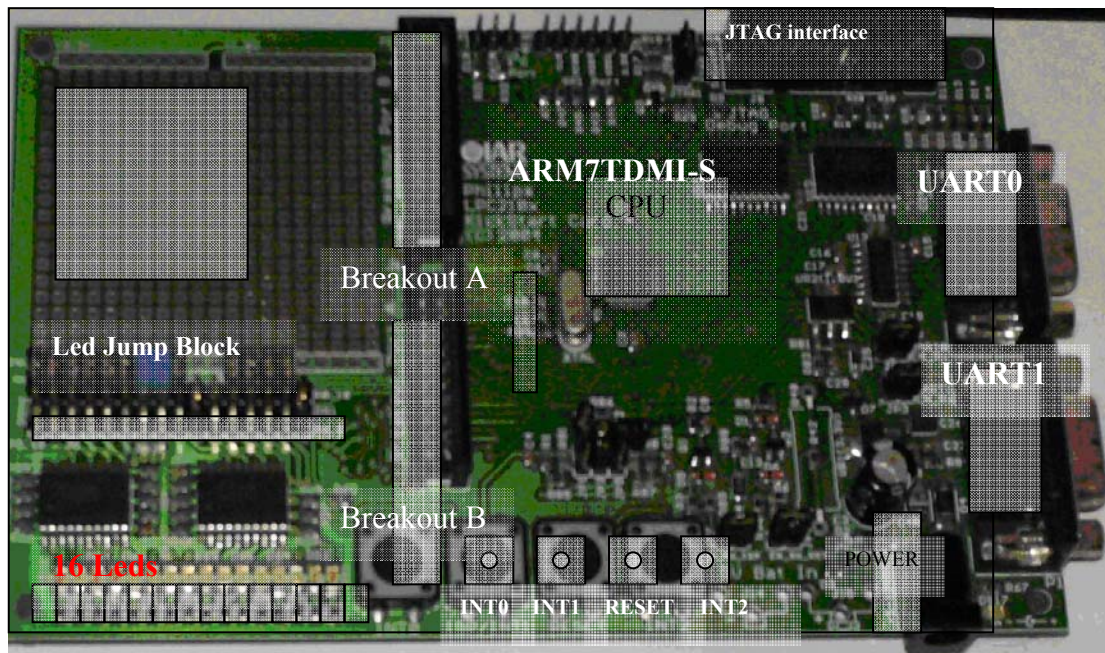
- 16/32 bit ARM7TDMI-S επεξεργαστής.
- 64 kB στατικής μνήμης RAM (SRAM).
- 128 kB διαθέσιμης μνήμης προγράμματος ,128 bit διασύνδεση / επιτάχυνσης επιτρέποντας λειτουργίες υψηλής ταχύτητας 60MHz.
- Ελεγκτής διακοπών ροής προγράμματος με καθοριζόμενες προτεραιότητες.
- Πολλαπλές σειριακές διασυνδέσεις 2 σειριακής επικοινωνίας (UART) , I2C και SPI.
- 32 χρονομετρητές (7 σύλληψης και σύγκρισης) μονάδα κατασκευής PWM παλμού, Real Time Clock and Watchdog.
- 32 γενικής χρήσης έξοδοι είσοδοι ,5V αντοχής,
- 60 MHz μέγιστη ταχύτητα επεξεργαστή.
- 2 χαμηλής κατανάλωσης λειτουργίες IDLE & Power down.
- Αφύπνιση του επεξεργαστή από την κατάσταση Power down μέσω εξωτερικής διακοπής(Interrupt).
- Δυνατότητα 2 παροχών τροφοδοσίας ,είτε μέσω 9V μπαταρίας η
- Τάση λειτουργία επεξεργαστή από 1.65 V έως 1.95 V (1.8 V \square 8.3%).
- Τάση γενικής χρήσης είσοδοι έξοδοι από3.0 V έως 3.6 V (3.3 V \square 10%) με 5 V αντοχή.

Το δομικό διάγραμμα της κάρτας και του επεξεργαστή είναι :



Σχήμα 3.1) Δομικό Διάγραμμα LPC2106

Η κατανομή των ηλεκτρονικών διατάξεων στην επιφάνεια της κάρτας και οι θέσεις των pins δείχνονται στο παρακάτω σχέδιο:

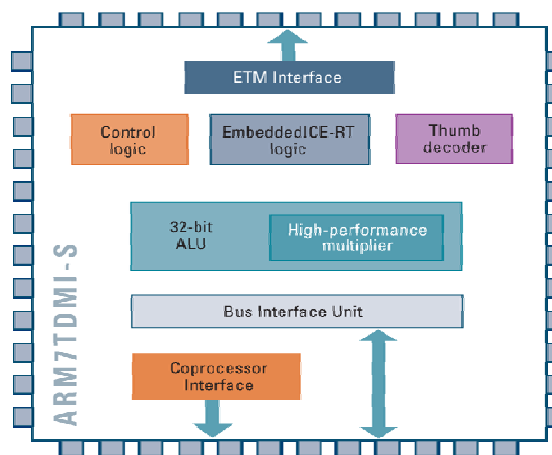


Σχήμα 3.2) Γενική διάταξη LP2106 επεξεργαστή και περιφερειακών θυρών επικοινωνίας.

Στο παραπάνω σχέδιο διακρίνονται οι θέσεις των δύο σειριακών θυρών ,UART0&1 στη δεξιά πλευρά της κάρτας. Στην κάτω περιοχή στα αριστερά είναι τοποθετημένα 16 leds κάτω από το μπλοκ αντιστοίχισης Pins και Leds. Στη μέση υπάρχουν 4 κουμπιά τα οποία λειτουργούν ως εξωτερικοί διακόπτες του προγράμματος ή επανεκκίνησης. Στα δεξιά υπάρχει η υποδοχή τροφοδοσίας της κάρτας. Στην επάνω αριστερή πλευρά διακρίνεται η πρότυπη περιοχή ενώ στο κέντρο διακρίνεται ο επεξεργαστής.

3.2) Επεξεργαστής ARM7TDMI-S

Η αρχιτεκτονική δομή του επεξεργαστή και η επικοινωνία των ενσωματωμένων μονάδων σε αυτόν παρουσιάζονται στο παρακάτω σχήμα.

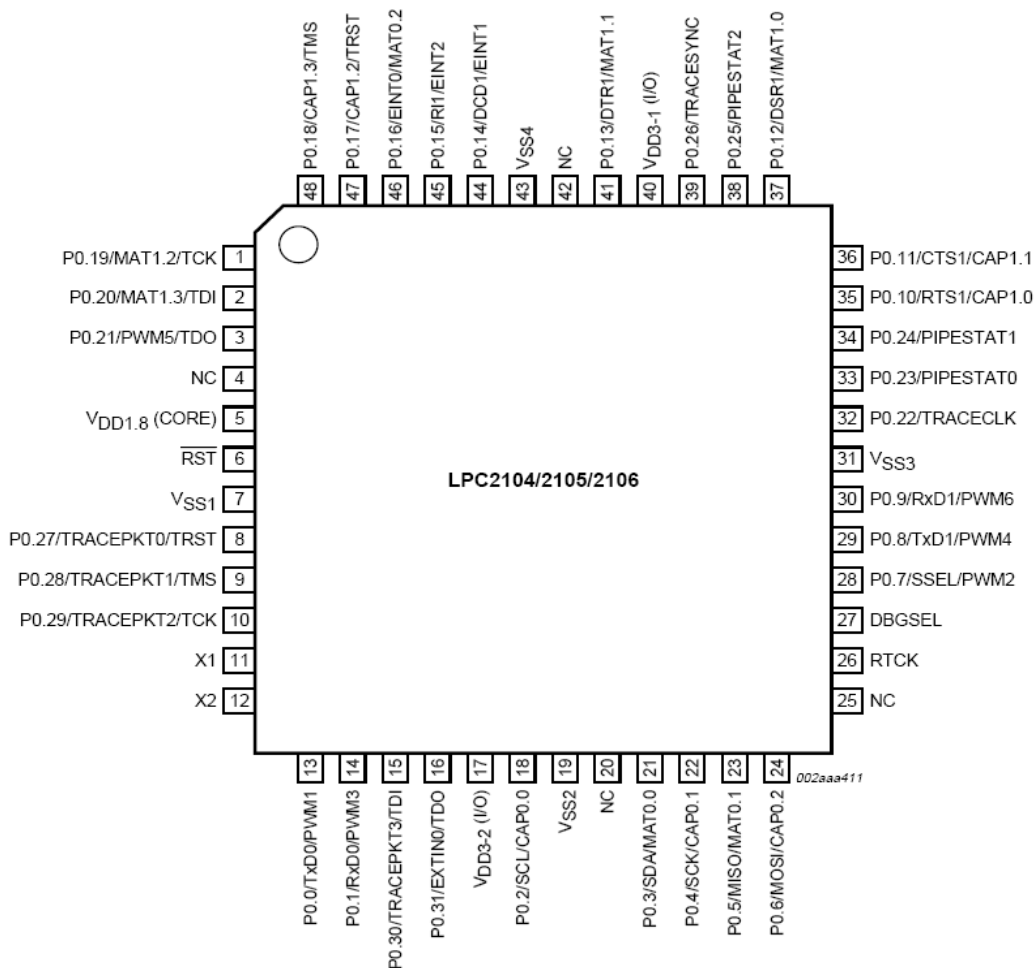


Σχήμα 3.3) Αρχιτεκτονική δομή λειτουργικών μονάδων εντός του επεξεργαστή.

Διακρίνονται οι μονάδες :

- ❖ Control logic Μονάδα λογικού ελέγχου.
- ❖ ALU(Arithmetic Logical Unit) μονάδα αριθμητικών πράξεων.
- ❖ Coprocessor interface Διασύνδεση συνεπεξεργαστή
- ❖ Bus interface unit

Η σύνδεση των ακίδων του επεξεργαστή στην κάρτα και οι διαθέσιμες λειτουργίες κάθε ακίδας «pin» παρουσιάζονται στο σχήμα.



Σχήμα 3.4) Σύνδεση και χρήση των ακίδων του επεξεργαστή.

Ρύθμιση της LPC2106 και χρήση των λειτουργιών της.

3.3) Χρονισμός επεξεργαστή.

Ο επεξεργαστής χρησιμοποιεί τον εξωτερικό κρύσταλλο XTAL ο οποίος παράγει το βασικό παλμό χρονισμού του επεξεργαστή. Ο παραγόμενος παλμός συνδέεται με τα pin 11 & 12 ή X1 & X2. Στο πρώτο Pin ο επεξεργαστής δέχεται την είσοδο του παλμού από τον κρύσταλλο και στο δεύτερο είναι η έξοδος του.

Ο σκοπός του κρυστάλλου ταλαντωτή, XTAL, είναι ο συντονισμός της εσωτερικής γεννήτρια χρονισμού του επεξεργαστή. Η γεννήτρια του ρολογιού του επεξεργαστή ορίζει την ταχύτητα λειτουργίας του επεξεργαστή (cclk). Η συχνότητα της γεννήτριας δεν έχει καθορισμένη σχέση με τη συχνότητα του κρυστάλλου αλλά δύναται να οριστεί με αποτέλεσμα την επιλογή διαφορετικών ταχυτήτων λειτουργίας του, **ARM7TDMI-S**.

Οι σχέσεις υπολογισμού της ταχύτητας του επεξεργαστή δίνονται παρακάτω.

- F_{osc} συχνότητα εξωτερικού κρυστάλλου.
- $F_{cco} = 14745600$ συχνότητα εσωτερικής γεννήτριας.
- $cclk$ συχνότητα λειτουργίας επεξεργαστή.
- M πολλαπλασιαστής της συχνότητας του εξωτερικού κρυστάλλου.
- P διαιρέτης της συχνότητας της εσωτερικής γεννήτριας.

Η ταχύτητα του επεξεργαστή συναρτήσει των παραπάνω δίνεται από τις σχέσεις:

$$cclk = M \cdot F_{osc} \quad \text{ή} \quad cclk = \frac{F_{cco}}{2 \cdot P}$$

$$\text{Ή } F_{cco} = cclk \cdot 2 \cdot P \quad \& \quad F_{cco} = F_{osc} \cdot M \cdot 2 \cdot P$$

$$F_{osc} \in (10, 25)MHz$$

$$cclk \in (10, 150)MHz$$

$$F_{cco} \in (156, 320)MHz$$

Ορίζεται ότι η συχνότητα λειτουργίας του επεξεργαστή είναι ίση με του εξωτερικού κρυστάλλου $cclk = 14745600Hz$ και επομένως ο πολλαπλασιαστής έχει τιμή .

$$M = \frac{cclk}{F_{osc}} = 1. \tag{3.1}$$

$$\& \quad cclk = M^1 \cdot F_{osc} = 14745600$$

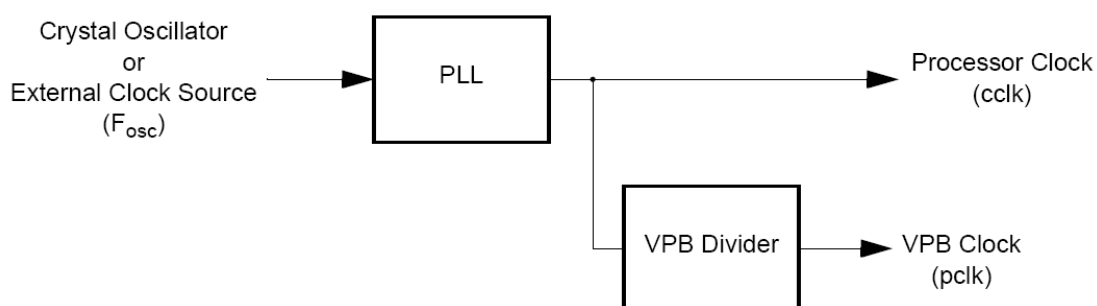
θέτω $P = 6$ και επομένως η συχνότητα της εσωτερικής γεννήτρια

$$F_{cco} = 14745600 \cdot 1 \cdot 2 \cdot 6 = 176.9MHz \tag{3.2}$$

Η συχνότητα λειτουργίας του επεξεργαστή σχέση 3.1 κρίνεται ότι είναι ικανοποιητική για την ταχύτητα εκτέλεσης του προγράμματος .

3.4) Χρονισμός ρολογιού περιφερειακών συσκευών και λειτουργιών επεξεργαστή (pclk).

Η χρήση και επικοινωνία του επεξεργαστή με τις περιφερειακές συσκευές απαιτεί τον ορισμό του ρολογιού λειτουργίας των περιφερειακών λειτουργιών «pclk». Ο διαιρέτης VPD ελέγχει τη σχέση μεταξύ της ταχύτητας του επεξεργαστή «cclk» του «pclk». Ο σκοπός του διαιρέτη είναι να ρυθμίσει την ταχύτητα των περιφερειακών λειτουργιών στην επιθυμητή και οπωσδήποτε να είναι μικρότερη από εκείνη του επεξεργαστή. Επιπλέον η μείωση της ταχύτητας αποσκοπεί στη μείωση της κατανάλωσης όταν στην εφαρμογή δεν απαιτείται σε καμία από τις περιφερειακές λειτουργίες ταχύτητα ίση με την ταχύτητα του επεξεργαστή.



Σχήμα 3.5) Διάγραμμα χρονισμού των περιφερειακών.

Ως εξορισμού η τιμή εκκίνησης για το διαιρέτη είναι ίση με 4. Αυτό έχει ως αποτέλεσμα η συχνότητα του περιφερειακού ρολογιού να είναι :

$$pclk = \frac{cclk}{VPD} = \frac{14745600}{4} = 3686400Hz \quad (3.3)$$

Η συχνότητα του περιφερειακού ρολογιού είναι ικανοποιητική για τις ανάγκες της εφαρμογής.

3.5) Περιγραφή της λειτουργίας των διακοπών ροής του προγράμματος.

Ο επεξεργαστής διαθέτει έναν ελεγκτή ο οποίος ονομάζεται VIC «Vectored Interrupt Controller». Ο λόγος της ύπαρξης του είναι ο έλεγχος της σειράς εκτέλεσης των αιτήσεων διακοπής του προγράμματος από τις περιφερειακές συσκευές. Παράδειγμα μιας τέτοια χρήσης είναι η ανά τακτά χρονικά διαστήματα εκτέλεση μια συγκεκριμένης σειράς ενεργειών όπως ο υπολογισμός της ταχύτητας του κινητήρα ή η λήψη δεδομένων από τη σειριακή θύρα .

Ο ελεγκτής «VIC» ελέγχει και οργανώνει τη σειρά εκτέλεσης των αιτημάτων διακοπής της ροής του προγράμματος. Έχει τη δυνατότητα καταχώρησης σε αυτόν 32 πηγών αίτησης διακοπής οι οποίες οργανώνονται σε τρεις κατηγορίες, γρήγορης εκτέλεσης διακοπή FIQ, VIQ «vectored interrupt request» και NonVIQ «Non-Vectored interrupt request» .

Ως γρήγορης εκτέλεσης διακοπής αίτηση FIQ «fast interrupt request» χαρακτηρίζονται οι αιτήσεις με την υψηλότερη προτεραιότητα εκτέλεσης. Η ελάχιστη καθυστέρηση στην εκτέλεση της διακοπής επιτυγχάνετε μόνο όταν μια πηγή έχει χαρακτηριστεί ως FIQ. Η διακοπή ροής του προγράμματος κατευθύνεται προς την εκτέλεση της συνάρτησης εξυπηρέτησης της διακοπής ISR «interrupt service routine» χωρίς έλεγχο της πηγής αίτησης.

Ως «Αιτήσεις Κατευθυνόμενης Διακοπής» (VIQ) χαρακτηρίζονται οι αιτήσεις με μεσαία προτεραιότητα. Συνολικά υπάρχουν 16 συνολικά θέσεις στην κατηγορία αυτή. Κάθε μία από τις 32 δυνατές αιτήσεις μπορούν να οριστούν στις 16 διαθέσιμες θέσεις. Μεταξύ των αιτήσεων της ίδιας κατηγορίας η προτεραιότητα ορίζεται σύμφωνα με τον αριθμό της θέσης στην οποία έχουν καταχωρηθεί. Όταν δύο ή περισσότερα αιτήματα της κατηγορίας αυτής έχουν ζητηθεί τότε ορίζεται η σειρά εκτέλεση τους με βάση τη θέση που κατέχουν και στη συνέχεια εκτελείται η συνάρτηση η οποία έχει οριστεί για τη συγκεκριμένη αίτηση. Ως «Αιτήσεις Μη Κατευθυνόμενης Διακοπής» (NonVIQ) χαρακτηρίζονται οι αιτήσεις με τη χαμηλότερη προτεραιότητα οι οποίες δεν έχουν τη δυνατότητα επιλογής, της συνάρτησης εκτέλεσης ανάλογα με την πηγή της αίτησης αλλά οδηγούνται προς μία συγκεκριμένη συνάρτηση για όλες τις πηγές.

Στον παρακάτω πίνακα παρουσιάζονται οι πηγές διακοπής για τις περιφερειακές λειτουργίες. Κάθε περιφερειακή συσκευή έχει μια γραμμή συνδεδεμένη με τον ελεγκτή των διακοπών VIC ,αλλά κάθε συσκευή μπορεί να έχει περισσότερες της μία εσωτερικές αιτήσεις.

Block	Flag(s)	VIC Channel #
WDT	Watchdog Interrupt (WDINT)	0
-	Reserved for software interrupts only	1
ARM Core	Embedded ICE, DbgCommRx	2
ARM Core	Embedded ICE, DbgCommTx	3
Timer 0	Match 0 - 3 (MR0, MR1, MR2, MR3) Capture 0 - 3 (CR0, CR1, CR2, CR3)	4
Timer 1	Match 0 - 3 (MR0, MR1, MR2, MR3) Capture 0 - 3 (CR0, CR1, CR2, CR3)	5
UART 0	Rx Line Status (RLS) Transmit Holding Register empty (THRE) Rx Data Available (RDA) Character Time-out Indicator (CTI)	6
UART 1	Rx Line Status (RLS) Transmit Holding Register empty (THRE) Rx Data Available (RDA) Character Time-out Indicator (CTI) Modem Status Interrupt (MSI)	7
PWM0	Match 0 - 6 (MR0, MR1, MR2, MR3, MR4, MR5, MR6) Capture 0 - 3 (CR0, CR1, CR2, CR3)	8
I2C	SI (state change)	9
SPI	SPIF, MODF	10
-	reserved	11
PLL	PLL Lock (PLOCK)	12
RTC	RTCCIF (Counter Increment), RTCALF (Alarm)	13
System Control	External Interrupt 0 (EINT0)	14
System Control	External Interrupt 1 (EINT1)	15
System Control	External Interrupt 2 (EINT2)	16

Πίνακας 3.1) Πηγές Διακοπής Προγράμματος

3.6) Περιγραφή των καταχωρητών χρήσης του Ελεγκτή Διακοπών VIC.

Αρχικά με τον καταχωρητή VICIntEnable επιλέγονται τα κανάλια του ελεγκτή διακοπών τα οποία είναι ενεργά και θα ζητούν από τον ελεγκτή αιτήματα διακοπής. Κάθε bit του καταχωρητή αντιστοιχείται στο αντίστοιχο κανάλι του VIC. Αντίθετα ο VICIntEnClr επιτρέπει την αποεπιλογή των επιθυμητών καναλιών στον VICIntEnable ως πηγές αιτήσεων στον VIC.

Ο καταχωρητής VICIRQStatus επιστρέφει την κατάσταση των αιτήσεων οι οποίες δεν είναι της κατηγορίας FIQ και έχουν ενεργοποιηθεί. Κάθε bit του καταχωρητή αντιστοιχεί και στο αντίστοιχο κανάλι του ελεγκτή διακοπών VIC σύμφωνα με τον παραπάνω πίνακα. Ομοίως ο VICFIQStatus επιστρέφει την κατάσταση των αιτήσεων διακοπής της κατηγορίας FIQ. Ο VICRawIntr επιστρέφει

την κατάσταση των αιτήσεων διακοπής ανεξάρτητα την κατηγορία στην οποία έχουν καταχωρηθεί.

Ο καταχωρητής VICVectAddr περιέχει τη διεύθυνση της συνάρτησης η οποία θα εκτελεστεί όταν μια διακοπή του προγράμματος έχει πραγματοποιηθεί. Αντίθετα ο VICDefVectAddr περιέχει τη διεύθυνση της συνάρτησης για την κατηγορία των διακοπών της κατηγορίας NonVIQ.

Με την ομάδα των καταχωρητών VICVectCntl0~ VICVectCntl15 ορίζονται οι θέσεις των πηγών των διακοπών που ανήκουν στην κατηγορία VIQ. Όπως έχει προαναφερθεί ανάλογα με τη θέση που λαμβάνουν στην ομάδα αυτή ορίζεται και η προτεραιότητα εκτέλεσης τους. Η πηγή του αιτήματος που βρίσκεται στη θέση VICVectCntl0 έχει υψηλότερη προτεραιότητα έναντι της επόμενης και ούτω καθ' εξής. Αντίστοιχα η ομάδα των καταχωρητών VICVectAddr0~15 περιέχει τις διευθύνσεις των συναρτήσεων οι οποίες εκτελούνται κατά τη διακοπή του προγράμματος. Στη πρώτη θέση VICVectAddr0 γράφεται η διεύθυνση της συνάρτησης που αντιστοιχεί στην πηγή η οποία είναι καταχωρημένη στην πρώτη θέση στην ομάδα VICVectCntl0, και οι υπόλοιπες καταχωρήσεις ακολουθούν την ίδια λογική.

Η εκτέλεση μιας διακοπής του προγράμματος δεν μπορεί να διακοπεί από μία νέα αίτηση της ίδιας κατηγορίας και για να εκτελεστεί πρέπει να τελειώσει πρώτα η προηγούμενη. Η χρήση των διακοπών πρέπει να γίνει με ιδιαίτερη προσοχή γιατί όταν η εκτέλεση της ολοκληρωθεί πρέπει το σήμα της διακοπής να σβήνεται. Σε αντίθετη περίπτωση η ροή του προγράμματος δεν θα συνεχιστεί και δεν θα επανέλθει στην κανονική του ροή. Εκτός από το σβήσιμο του σήματος διακοπής πρέπει να σβήνεται και το σήμα της συσκευής η οποία προκάλεσε τη διακοπή ώστε να θεωρηθεί ότι έχει εκτελεστεί. Αν δε σβηστεί τότε μετά το πέρας της διακοπής θα συνεχίσει να εκκρεμεί η αίτηση με αποτέλεσμα την ενεργοποίηση του ελεγκτή διακοπών VIC ,ξεκινώντας από την αρχή τη διαδικασία εκτέλεσης της διακοπής.

Η ροή εκτέλεσης μια διακοπής ξεκινά από την αίτηση μίας περιφερειακής συσκευής προς τον VIC η οποία έχει ορισθεί ως πηγή διακοπών στον VICIntEnable. Στη συνέχεια στέλνεται σήμα στον επεξεργαστή προς διακοπή της εκτέλεσης του προγράμματος. Ο επεξεργαστής σταματά τη ροή και ζητά τη διεύθυνση της συνάρτησης η οποία πρόκειται να εκτελεστεί. Ο VIC ανάλογα με την κατηγορία στην οποία είναι καθορισμένη η πηγή της αίτησης αποφασίζει ποια από τις υπάρχουσες αιτήσεις θα εκτελεστεί. Ειδικά αν ανήκει στην κατηγορία VIQ τότε η προτεραιότητα ορίζεται από τη θέση της. Στη συνέχεια επιστρέφεται στο επεξεργαστή η διεύθυνση της συνάρτησης η οποία αντιστοιχεί στην επιλεγμένη αίτηση με τη μεγαλύτερη προτεραιότητα και εκτελείται. Με τον τερματισμό της εκτέλεσης καθαρίζεται το σήμα αίτησης της περιφερειακής συσκευής VIC και στη συνέχεια καθαρίζεται και το σήμα διακοπής προς τον επεξεργαστή. Ο επεξεργαστή επανέρχεται στην εκτέλεση του προγράμματος από το σημείο στο οποίο είχε διακοπεί.

3.7) Κατανομή των περιφερειακών συσκευών στον ελεγκτή διακοπών και ορισμός των προτεραιοτήτων.

Η λειτουργία των περιφερειακών συσκευών πραγματοποιείται με τη χρήση των διαθέσιμων διακοπών «interrupts» κάθε συσκευής. Η ταυτόχρονη λειτουργία πολλών συσκευών εισάγει το πρόβλημα της κατανομής προτεραιοτήτων στην εξυπηρέτησή τους. Η εκτέλεση των σημαντικότερων λειτουργιών πρέπει να πραγματοποιείται ανεξάρτητα από τη συχνότητα εκτέλεσης των λιγότερων σημαντικών. Σε αντίθετη περίπτωση κινδυνεύει η εκτέλεση του προγράμματος και κατ' επέκταση ο έλεγχος της εγκατάστασης από την εκτέλεση λιγότερων σημαντικών λειτουργιών, η συχνότητα των οποίων δεν θα αφήνει καθόλου επεξεργαστικό χρόνο. Αντίθετα η μη χρήση διακοπών θα είχε ως συνέπεια τη σειριακή εξυπηρέτηση κάθε συσκευής διαδοχικά και την αύξηση του χρόνου εκτέλεσης ολόκληρης της εφαρμογής. Η σειριακή εκτέλεση ορίζει ότι όλες έχουν την ίδια προτεραιότητα και σημασία σε ένα αλγόριθμο. Επιπλέον μία απαιτητική εφαρμογή, εξυπηρέτησης μιας συσκευής θα καθυστερούσε την εκτέλεση και των υπόλοιπων με αποτέλεσμα τη σημαντική επιβράδυνση του κώδικα. Η επιβράδυνση θα τις οδηγήσει να εκτελούνται σε χρόνους οι οποίοι δεν θα πληρούν τις προϋποθέσεις ελέγχου της εγκατάστασης. Ένα τέτοιο πρόβλημα θα δημιουργούνταν αν κατά την αποστολή των δεδομένων μέσω της σειριακής ήταν αναγκαία η ολοκλήρωση της η αποστολή όλων των δεδομένων, πριν εκτελεστεί ο έλεγχος του σήματος του οπτικού αισθητήρα. Ο έλεγχος του σήματος θα ήταν αδύνατος διότι η ταχύτητα της σειριακής είναι σημαντικά μικρότερη από την απαιτούμενη ταχύτητα για τον αισθητήρα.

- **Ανάλυση προτεραιοτήτων περιφερειακών συσκευών.**

Οι χρησιμοποιούμενες περιφερειακές συσκευές του επεξεργαστή είναι :

1. UART0.
2. UART1.
3. Timer0.
4. Timer1.
5. PWM0.
6. System Control External Interrupt 1.

Η εξήγηση της λειτουργίας κάθε μίας συσκευής γίνεται στα παρακάτω κεφάλαια. Όπως έχει προαναφερθεί κάθε περιφερειακή συσκευή μπορεί να διαθέτει περισσότερες της μιας εσωτερικές πηγές αιτήσεων, οι οποίες αναλύονται στα κεφάλαια περιγραφής κάθε μιας ξεχωριστά καθώς και η κατανομή των εσωτερικών προτεραιοτήτων τους. Η συνολική εποπτεία της κατανομής προτεραιοτήτων δίνεται στο τέλος του κεφαλαίου. Προς το παρόν θα ασχοληθούμε με την κατανομή των προτεραιοτήτων ως προς τις περιφερειακές συσκευές μεταξύ τους.

Ο ορισμός προτεραιοτήτων είναι δυνατός για την κατηγορία VIQ η οποία έχει την ιδιότητα κατεύθυνσης της διακοπής προς την εκτέλεση διαφορετικών συναρτήσεων για διαφορετικές πηγές αιτήσεων με προγραμματιζόμενη αντιστοιχία. Επιπλέον αν και η κατηγορία VIQ δεν είναι η ταχύτερη δυνατή, το πλήθος των συσκευών δεν μας επιτρέπει τη χρήση τους στην κατηγορία FIQ διότι θα δημιουργηθεί καθυστέρηση ανάλογη της αύξησης του πλήθους των πηγών κατάταξης στην κατηγορία αυτή. Επιπλέον δεν υπάρχει η δυνατότητα επιλογής της εκτελούμενης συνάρτησης ανά πηγή παρά μόνο μιας σταθερής συνάρτησης, στην οποία προγραμματιστικά θα πρέπει να αναγνωριστούν οι πηγές των αιτήσεων. Η επιλογή αυτή θα αύξανε το χρόνο σε

σχέση με το λογικό κύκλωμα το οποίο διαθέτει ο επεξεργαστή στον ελεγκτή διακοπών VIC. Η χρήση της κατηγορίας NonVIQ έχει όμοια λειτουργία με την κατηγορία FIQ, δηλαδή δεν παρέχεται ορισμός προτεραιοτήτων και επιπλέον χαμηλότερη προτεραιότητα εκτέλεσης έναντι όλων των άλλων κατηγοριών.

Ως σημαντικότερη όλων κρίνεται η επικοινωνία με τον υπολογιστή. Ο λόγος είναι ότι ο υπολογιστής και κατ' επέκταση ο χρήστης αυτού έχει τον έλεγχο της όλης εγκατάστασης. Ο έλεγχος πραγματοποιείται μέσω εντολών οι οποίες στέλνονται δια της σειριακής θύρας. Οι εντολές αναγνωρίζονται από το πρόγραμμα της κάρτας και εκτελούνται. Επομένως η απώλεια της επικοινωνίας λόγω της μη εξυπηρέτησης της σειριακής σε εύλογο διάστημα έχει ως συνέπεια την απώλεια του ελέγχου, οδηγεί στην υπερφόρτιση της μνήμης «buffer» της σειριακής και την απώλεια δεδομένων. Η σειριακή θύρα που χρησιμοποιείται για την επικοινωνία με τον υπολογιστή είναι η UART0.

Δεύτερη σε προτεραιότητα περιφερειακή συσκευή τίθεται η δεύτερη σειριακή θύρα. Ο λόγος είναι ότι όπως και στην παραπάνω παράγραφο η καθυστέρηση στην εξυπηρέτηση της θα είχε ως συνέπεια την απώλεια των εισερχόμενων δεδομένων από τη δυναμοκυψέλη. Τα δεδομένα της σειριακής αυτής δεν είναι απαραίτητα για τον έλεγχο της εγκατάστασης ωστόσο η περιορισμένη χρήση της και οι μικροί χρόνοι εξυπηρέτησης της δεν επηρεάζουν καθοριστικά την εξυπηρέτηση των άλλων με αποτέλεσμα να δύναται να γραφεί στη δεύτερη θέση προτεραιότητας η UART1.

Τρίτος σε προτεραιότητα επιλέγεται να χρησιμοποιηθεί ο περιφερειακός χρονομετρητής Timer1. Η χρήση του ορίζεται ως η καταγραφή του σήματος του υπέρυθρου φωτοανιχνευτή για τη μέτρηση των περιστροφών του κινητήρα καθώς και της χρήση του ως ρολόι της όλης εφαρμογής. Περισσότερες πληροφορίες ανατρέξτε στο κεφάλαιο λειτουργία Timer1. Για να χρησιμοποιηθεί ο Timer1 ως μετρητής των περιστροφών πρέπει η προτεραιότητα της εκτέλεσης του να είναι μεγαλύτερη της προτεραιότητας έναντι της συνάρτησης μέτρησης των στροφών και του υπολογισμού του χρόνου. Η απαίτηση αυτή απορρέει από το γεγονός ότι εφόσον προκύψουν ταυτόχρονα δύο αιτήματα το ένα από τον καταγραφέα του σήματος του υπέρυθρου ανιχνευτή και το άλλο από τη συνάρτηση υπολογισμού της ταχύτητας είναι εύλογο να εξυπηρετηθεί πρώτα η καταγραφή του σήματος και εν' συνεχεία ο υπολογισμός της ταχύτητας. Ειδικά η τελευταία καταγραφή του σήματος, αύξηση του μετρητή, μεταφέρεται στην επόμενη μέτρηση των στροφών με αποτέλεσμα τη δημιουργία λογικού λάθους της εφαρμογής. Η μέτρηση των στροφών είναι πολύ σημαντική για την εξαγωγή των αποφάσεων του χρήστη του υπολογιστή και για τον αλγόριθμο του ελεγκτή στροφών. Ο ελεγκτής δεν μπορεί να χρησιμοποιηθεί εφόσον η μέτρηση της περιστροφικής ταχύτητας είναι λανθασμένη ενώ ο χρήστης της εγκατάστασης θα οδηγηθεί σε λάθος απόφαση. Για τους παραπάνω λόγους η προτεραιότητα του χρονομετρητή Timer1 επιλέγεται να λάβει θέση αμέσως μετά τις σειριακές θύρες UART0 & UART1.

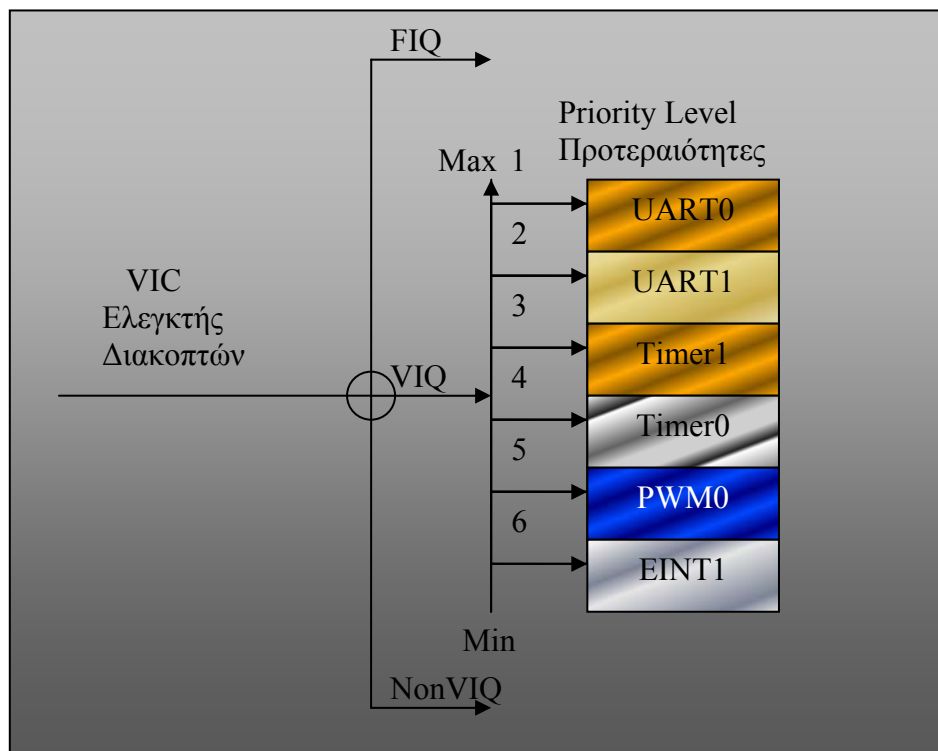
Τέταρτος σε προτεραιότητα επιλέγεται ο περιφερειακός χρονομετρητής Timer0. Η χρήση του αποσκοπεί στο χρονισμό της περιοδική επικοινωνία και αποστολή των δεδομένων στον υπολογιστή. Επίσης χρησιμοποιείται και στον περιοδικό έλεγχο της σειριακής θύρας, για τη λήψη εντός εύλογου χρονικού διαστήματος των δεδομένων από τον υπολογιστή. Στην περιοδική αποστολή των μετρήσεων εντάσσεται και η εκτέλεση της συνάρτησης μέτρησης των στροφών. Όπως αναφέρθηκε παραπάνω η

εκτέλεση της μέτρησης των στροφών πρέπει να πραγματοποιηθεί σε χαμηλότερη προτεραιότητα από τη λήψη του σήματος του φωτοανιχνευτή στον Timer1. Η δυνατή θέση λοιπόν που μπορεί να λάβει η προτεραιότητα του Timer0 είναι η αμέσως χαμηλότερη θέση από τη θέση του Timer1. Επιπλέον η χρήση του Timer0 για την περιοδική επικοινωνία και αποστολή των δεδομένων λειτουργίας της εγκατάστασης στον υπολογιστή τον καθιστά ως το αμέσως σημαντικότερο περιφερειακό σύστημα. Η μη έγκαιρη εκτέλεση του θα έχει ως αποτέλεσμα την καθυστερημένη αποστολή των δεδομένων στον υπολογιστή ή τον υπολογισμό της ταχύτητας σε μεγαλύτερα διαστήματα από αυτά που χρειάζονται για τη σωστή λειτουργία του ελεγκτή. Είναι σημαντική η τοποθέτηση της προτεραιότητας του Timer0 κατά μία θέση χαμηλότερη του Timer1 και της μη παρεμβολής άλλων μεταξύ αυτών. Η σχεδίαση και γραφή της συνάρτησης υπολογισμού της περιστροφικής ταχύτητας έχει γίνει με βάση τη μικρότερη δυνατή καθυστέρηση στην εκτέλεση των διακοπών «interrupts» μεταξύ των δύο χρονομετρητών Timer0 & Timer1. Μεγάλες καθυστερήσεις μπορεί να έχουν αποτέλεσμα τη μη ακριβή μέτρηση της ταχύτητας, εφόσον δεν θα περιγράφει πλέον την περίοδο μέτρησης αλλά μεγαλύτερη αυτής. Επιπλέον πολύ μεγάλες καθυστερήσεις μπορεί να έχουν ως αποτέλεσμα τη θεώρηση ότι έχουν πληρωθεί τα κριτήρια απώλειας της σειριακής επικοινωνίας και τον τερματισμό του προγράμματος, όπως εξηγείται αναλυτικότερα στα κεφάλαιο περιγραφής του Timer0.

Πέμπτη σε προτεραιότητα επιλέγεται η χρήση της περιφερειακής συσκευής δημιουργίας παλμού PWM2 .Ο παλμός χρησιμοποιείται για τον έλεγχο του παλμοτροφοδοτικού μετατροπέα τάσης και κατ' επέκταση σύμφωνα με την ανάλυση στο κεφάλαιο 1 τον έλεγχο των στροφών του κινητήρα. Η διακοπή του «PWM2» έχει ως σκοπό από τη στιγμή που είναι γνωστή η ταχύτητα του κινητήρα να εφαρμόσει τα νέα χαρακτηριστικά του παλμού «Duty Cycle» είτε μέσω της μεθόδου του κανόνα ελέγχου είτε μέσω του ελεγκτή στροφών. Επειδή η επικοινωνία με τον παλμοτροφοδοτικό μετατροπέα πραγματοποιείται σε σταθερή περιοδικότητα χρησιμοποιείται η περίοδος αυτή για την εφαρμογή της νέας τιμής του «Duty Cycle». Η περιοδική αλλαγή του «Duty Cycle» μηδενίζει τις άσκοπες αλλαγές ενδιάμεσα στα διαστήματα εφόσον αυτές, ούτως ή άλλως δεν πρόκειται να εφαρμοσθούν. Μόνο η τελευταία εισαγωγή ,πριν την εκκίνηση της νέας περιόδου του παλμού «PWM» ,της τιμής του «Duty Cycle» στον καταχωρητή θα χρησιμοποιηθεί .Κάθε εγγραφή στον καταχωρητή που είναι υπεύθυνος για τον ορισμό του «Duty Cycle» σβήνει την προηγούμενη .Η διακοπή «Interrupt» αυτή λοιπόν χρησιμοποιείται ως περιοδικός ελεγκτής της πρόσβασης στις ιδιότητες του παλμού PWM2. Αν κατά την ίδια χρονική στιγμή ζητηθούν αιτήματα διακοπής στον ελεγκτή διακοπών,VIC, από τον Timer0 και από τον PWM0 θα πρέπει πρώτα να υπολογιστεί η ταχύτητα λειτουργίας του κινητήρα. Στη συνέχεια να εκτελεστεί η διακοπή για τον PWM0,η οποία προϋποθέτει τη γνώση της ταχύτητας του κινητήρα για τη χρήση του ελεγκτή στροφών.

Έκτη και τελευταία επιλέγεται η χρήση της περιφερειακής συσκευής External Interrupt 1(EINT1). Ο σκοπός της χρήσης της είναι η παροχή της δυνατότητας εναλλαγής μεταξύ του ελεγκτή στροφών και της απευθείας ρύθμισης του Duty Cycle ,κανόνας ελέγχου. Η λειτουργία αυτή του EINT1 δεν είναι περιοδική και επιπλέον η χρήση της δεν αναμένεται να είναι εκτεταμένη ούτε κρίσιμη για τον έλεγχο της εγκατάστασης. Η ταχύτητα εκτέλεσης της είναι αδιάφορη πρακτικά. Από τους παραπάνω λόγους λοιπόν τίθεται η προτεραιότητα της ως τελευταία όλων.

Η γραφική καταχώρηση των προτεραιοτήτων των περιφερειακών συσκευών στον ελεγκτή VIC αναπαριστώνται στο παρακάτω σχεδιάγραμμα.



Σχήμα 3.6) Κατανομή προτεραιοτήτων των πηγών διακοπής των περιφερειακών συσκευών.

3.8) Περιγραφή σειριακής Θύρας UART (universal asynchronous receiver / transmitter).

Οι σειριακές θύρες που διαθέτει η κάρτα LPC2106 είναι δύο, του τύπου UART. Η σύνδεση της σειριακής θύρας με τον υπολογιστή απαιτεί τη χρήση καλωδίου «null modem», ενώ για τη σύνδεση με το σύστημα μέτρησης της ώσης απαιτεί καλώδιο τύπου «RS232».

Η χρήση και η ρύθμιση των σειριακών θυρών πραγματοποιείται μέσω των καταχωρητών των σειριακών θυρών οι οποίοι παρουσιάζονται στα βοηθήματα. Περιγράφονται παρακάτω οι απαιτούμενοι καταχωρητές για την επίτευξη της επικοινωνίας και υπολογίζονται οι απαραίτητες παράμετροι. Ανάλογα με την επιλεγμένη σειριακή θύρα το δεύτερο γράμμα του ονόματος κάθε καταχωρητή ορίζει τον αριθμό της θύρας δηλαδή U0RBR για την πρώτη σειριακή θύρα και U1RBR για τη δεύτερη.

Ο καταχωρητής U0RBR ή U1RBR περιέχει το εισερχόμενο byte είτε από τον υπολογιστή είτε από το σύστημα μέτρησης της ώσης αντίστοιχα. Στον καταχωρητή U0THR γράφεται το byte το οποίο θέλουμε να στείλουμε στον υπολογιστή. Στον καταχωρητή U0IER ή U1IER ορίζονται οι εσωτερικές πηγές των σειριακών που μπορούν να ζητήσουν από τον ελεγκτή διακοπών «VIC» τη διακοπή του προγράμματος. Η επικοινωνία της LPC2106 με τον υπολογιστή απαιτεί την αποστολή

και λήψη δεδομένων. Επομένως στον UOIER θα ενεργοποιηθούν οι εσωτερικές πηγές διακοπής 1) αίτηση διακοπής όταν έχει ληφθεί 1byte RDA «receive data available» 2) αίτηση διακοπής όταν ο καταχωρητής αποστολής U1THR είναι άδειος. Δηλαδή έχει ολοκληρωθεί η αποστολή του χαρακτήρα στο υπολογιστή THRE «transmitter holder register Empty». Ο λόγος της χρήσης της αίτησης διακοπής λόγω λήψης δεδομένων είναι η πληροφόρηση του προγράμματος ότι ένα «byte» πληροφορίας από τη σειριακή έχει ληφθεί και η ανάγνωση του byte από τον καταχωρητή U0RBR. Η συνάρτηση ανάγνωσης χρησιμοποιεί την πληροφορία αυτή σύμφωνα με το προσυμφωνημένο και σχεδιασμένο πρωτόκολλο για την ανασυγκρότηση της μεταφερόμενης πληροφορίας. Το πρωτόκολλο παρουσιάζεται στο κεφάλαιο 3. Σημειώνεται ότι τα πρωτόκολλα επικοινωνίας μεταξύ λήψης και αποστολής είναι διαφορετικά τόσο μεταξύ τους όσο και μεταξύ των χρησιμοποιούμενων θυρών.

Η δεύτερη σειριακή θύρα «UART1» δεν χρησιμοποιείται για την αποστολή δεδομένων προς το σύστημα μετρήσεων ώσης «load cell» και επομένως θα ενεργοποιηθεί μόνο η αίτηση λήψης δεδομένων RDA «receive data available» στον καταχωρητή των εσωτερικών αιτήσεων διακοπής UOIER. Η χρήση της αίτησης είναι όμοια με τη χρήση της αιτήσεως της πρώτης σειριακής «UART0» και διέπεται από τις ίδιες αρχές.

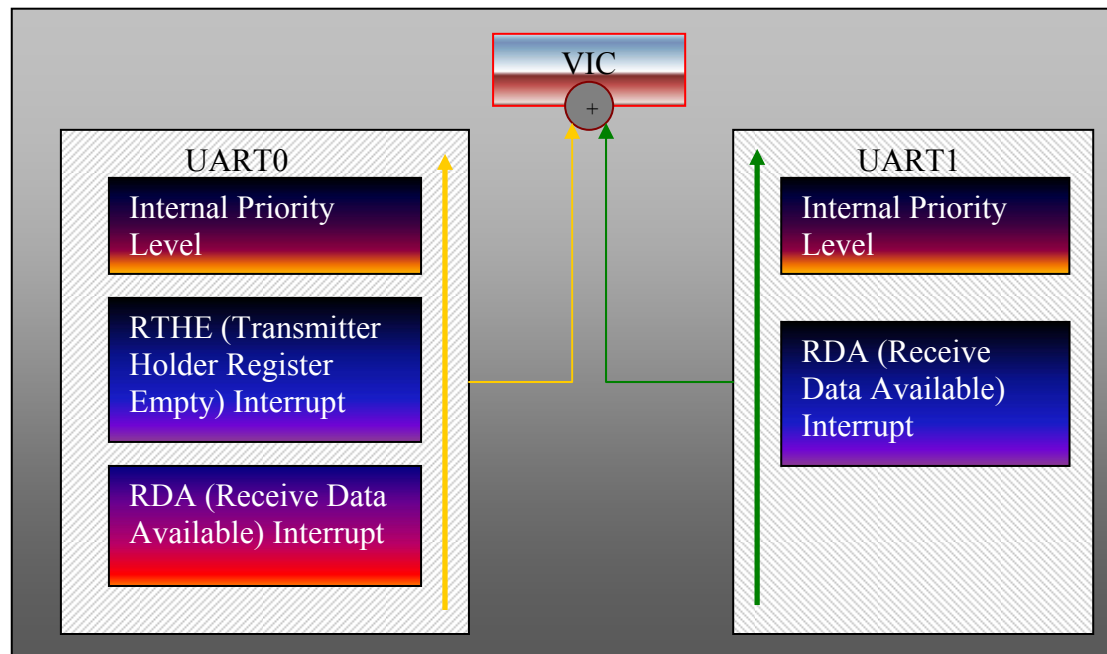
Ο καταχωρητής U0IIR χρησιμοποιείται από τη συνάρτηση η οποία εκτελείται κατά την αίτηση διακοπής από την πηγή UART0 με σκοπό την εξακρίβωση της εσωτερικής πηγής η οποία αιτήθηκε ώστε να εκτελεστούν οι αντίστοιχες ενέργειες. Η σειρά με την οποία γίνεται ο έλεγχος στη συνάρτηση εξυπηρέτησης της καθορίζει και την προτεραιότητα ανάμεσα στις εσωτερικές πηγές αιτήσεων αυτής. Αν οι πηγές είναι περισσότερες της μιας τότε εκτελούνται όλες οι απαραίτητες ενέργειες σύμφωνα με τη σειρά γραφής τους πριν ολοκληρωθεί η διακοπή και επανέλθει το πρόγραμμα στη ροή του. Ομοίως η λειτουργία του καταχωρητή U1IIR της σειριακής δεύτερης θύρας «UART1».

Στη σειριακή «UART0» επιλέγεται να τεθεί υψηλότερη προτεραιότητα στον έλεγχο για την αποστολή δεδομένων «THRE» έναντι της λήψης δεδομένων «RDA». Η ρύθμιση των προτεραιοτήτων των εσωτερικών πηγών πραγματοποιείται υπό το γνώμονα ότι η εκτέλεση της αποστολής δεδομένων είναι συγκριτικά ταχύτερη. Ο κώδικας που χρησιμοποιεί είναι απλούστερος και μικρότερος με αποτέλεσμα μικρότερες καθυστερήσεις έναντι της λήψης. Η σειριακή 2 «UART1» έχει μία μόνο εσωτερική πηγή αιτήσεων την «RDA» επομένως πραγματοποιείται μόνο έλεγχος για αυτήν.

Ο καταχωρητής U0FCR ελέγχει τη χρήση της προσωρινής μνήμης, «buffer», της σειριακής θύρας. Η ορθή λειτουργία της σειριακής απαιτεί τη χρήση της προσωρινής μνήμης. Στην προσωρινή μνήμη αποθηκεύονται τα εισερχόμενα και εξερχόμενα δεδομένα. Η διαχείριση της πραγματοποιείται από τον επεξεργαστή σύμφωνα με τη λογική FIFO. Οι επιλογές στον καταχωρητή αυτό, ενεργοποιούν τη χρήση της και επίσης ορίζονται ξεχωριστά οι εντολές καθαρισμού των καναλιών αποστολής και λήψης. Ορίζεται το πλήθος το λαμβανόμενων bytes πριν ζητηθεί αίτηση διακοπής λόγω λήψης δεδομένων RDA στον «VIC». Το μέγεθος της μνήμης σε κάθε κανάλι είναι 16 byte. Ο σκοπός της είναι η παροχή αρκετού χρόνου ώστε ακόμα και αν καθυστερήσει η εκτέλεση της διακοπής τα δεδομένα τα οποία στέλνονται και λαμβάνονται να μην χαθούν από τα επόμενα μέχρι την ανάγνωση και γραφή τους

στους καταχωρητές U0RBR και U0THR αντίστοιχα. Για τον καταχωρητή U1FCR της δεύτερης σειριακής ισχύει ότι και για αυτόν της πρώτης. Για τις δύο αυτές σειριακές επιλέγεται η ενεργοποίηση της «RDA» διακοπής κάθε byte που λαμβάνεται ,δηλαδή το επίπεδο ενεργοποίησης αφήνεται ως το προεπιλεγμένο 0.

Το σχηματικό διάγραμμα των διακοπών κάθε σειριακής είναι :



Σχήμα 3.7) Κατανομή των εσωτερικών προτεραιοτήτων των σειριακών θυρών.

Οι παράμετροι σύνδεσης της σειριακής θύρας ορίζονται στον καταχωρητή U0LCR. Σε αυτόν ορίζεται η ταχύτητα της σειριακής θύρας ,ο έλεγχος ισοτιμίας «parity» ,ο αριθμός των bits στο τέλος κάθε μεταφερόμενης λέξης, το μέγεθος της καθώς και η χρήση του σήματος τερματισμού της επικοινωνίας. Στις δυο σειριακές θύρες δεν χρησιμοποιείται το σήμα τερματισμού. Ο σκοπός του είναι να θέτει τη σειριακή θύρα σε κατάσταση αναμονής περιμένοντας το κατάλληλο σήμα εκκίνησης. Η χρήση του ελέγχου για parity «bit ισοτιμίας» χρησιμοποιείται από το μηχανισμό της σειριακή για τον έλεγχο των μεταφερόμενων δεδομένων και την ανίχνευση λάθους κατά τη μεταφορά. Στα πρωτόκολλα επικοινωνίας όπως παρουσιάζονται στο κεφάλαιο 4 ,και των δύο σειριακών θυρών έχει ενσωματωθεί αλγόριθμος ανίχνευσης λαθών στις συναρτήσεις ανάγνωσης των σειριακών και επομένως δεν κρίνεται σκόπιμη η χρήση του ελέγχου ισοτιμίας. Η μη χρήση της ισοτιμίας και για τις δύο θύρες δεν θα δημιουργήσει το bit ισοτιμίας μειώνοντας έτσι τον αριθμό μεταφερόμενων bits ανά λέξη. Ορίζεται ότι το μέγεθος της μεταφερόμενης λέξης θα είναι ίσο με 1 byte δηλαδή 8 bits ή ένας χαρακτήρας. Επομένως για κάθε λέξη που μεταφέρεται το πλήθος των bits που χρησιμοποιούνται είναι :

$$Bits_per_Byte = 1 \frac{startbit}{Word} + 8 \frac{bits}{word} + 1 \frac{stopbit}{word} = 10 \frac{bits}{word} \quad (3.5)$$

Η ταχύτητα λειτουργίας κάθε σειριακής εξαρτάται από το μέγεθος των μεταφερόμενων δεδομένων από και προς αυτήν. Στον υπολογισμό της ταχύτητας για

τη σειριακή UART0 εκτός του όγκου των μεταφερόμενων δεδομένων πρέπει να συνυπολογιστεί και η ταχύτητα ολοκλήρωσης της μεταφοράς του συνόλου των δεδομένων υπό μορφή πακέτου όπως ορίζεται από το πρωτόκολλο επικοινωνίας. Μικρές ταχύτητες θα έχουν ως συνέπεια τη χρήση των διακοπών της σειριακής για μεγαλύτερο χρόνο με αποτέλεσμα, λόγω της δεσπόζουσας θέσης που κατέχει στον ελεγκτή διακοπών VIC, τον αποκλεισμό της εκτέλεσης των υπολοίπων. Το φαινόμενο αυτό παρατηρήθηκε κατά την ανάπτυξη του προγράμματος οπότε αν και η επιλεγμένη ταχύτητα ικανοποιούσε τον όγκο των μεταφερόμενων δεδομένων, δημιουργούσε μεγάλες καθυστερήσεις κυρίως στην εκτέλεση της διακοπής του Timer1 για τη λήψη του σήματος από τον υπέρυθρο ανιχνευτή.

Στο πρωτόκολλο αποστολής των δεδομένων ,από την κάρτα προς τον υπολογιστή, έχει οριστεί ότι τα δεδομένα οργανώνονται σε μια συνεχή ομάδα των 27 bytes και τα οποία αποστέλλονται διαδοχικά. Περισσότερες πληροφορίες στο κεφάλαιο 4. Επιπλέον ορίζετε ότι η μέγιστη συχνότητα αποστολής των δεδομένων είναι 50 πακέτα ανά δευτερόλεπτο . Επομένως ο μέγιστος αριθμός αποστελλόμενων χαρακτήρων ανά δευτερόλεπτο δίνεται από τη σχέση :

$$UART0_{MaxBps} = 50Hz \cdot 27 \frac{Bytes}{DataPackage} = 1350 \frac{Bytes}{Sec} \quad (3.6)$$

Λαμβάνοντας υπόψη τις ρυθμίσεις της σειριακής που παρουσιάστηκαν στην προηγούμενη παράγραφο, ορίζεται ότι για κάθε ένα αποστελλόμενο χαρακτήρα-byte απαιτούνται 10bits σύμφωνα με τη σχέση 3.5 ,ο μέγιστος τελικός αριθμός μεταφερόμενων bits είναι :

$$UART0_{MaxBits / Sec} = 1350 \frac{Byte}{Sec} \cdot 10 \frac{Bits}{Byte} = 13500 BitsPerSec \quad (3.7)$$

Η τιμή 13500 ορίζει και την ελάχιστη απαιτούμενη ταχύτητα της σειριακής ως προς την ικανοποίηση της μεταφοράς όλων των δεδομένων. Πρέπει να αναφερθεί ότι ο έλεγχος οφείλει να γίνεται και για τις δύο γραμμές επικοινωνίας ,δηλαδή και για την ικανοποίηση της λήψης. Στο πρωτόκολλο λήψης ορίζεται ότι τα λαμβανόμενα δεδομένα οργανώνονται σε μια ομάδα των 6 bytes ανά πακέτο το οποίο είναι αντίστοιχα μικρότερο του μεγέθους εκείνου προς αποστολή. Η συχνότητα λήψης είναι αντίστοιχα μικρότερη ή ίση της αποστολής, 50Hz. Άρα η μέγιστη ταχύτητα που ικανοποιεί τη λήψη είναι :

$$UART0_{MaxBits / Sec} = 6 \frac{bytes}{datapackage} \cdot 50 \frac{datapackage}{sec} \cdot 10 \frac{Bits}{Byte} = 3000 BitsPerSec \quad (3.8)$$

,η οποία είναι μικρότερη της ελάχιστης για αποστολή σχέση 3.7 και επομένως αγνοείται.

Όπως αναφέρθηκε παραπάνω η διακοπή εκτός της ικανοποίησης του μεταφερόμενου όγκου δεδομένων δεν θα πρέπει να επηρεάζει σημαντικά και τις υπόλοιπες διακοπές. Αυτή που είναι πιο ευαίσθητη έναντι της σειριακής είναι του timer1. Για την καταγραφή του σήματος από τον υπέρυθρο φωτοανιχνευτή, θα ζητείται μία διακοπή ανά περιστροφή του κινητήρα. Επομένως γνωρίζοντας ότι οι

μέγιστες στροφές είναι μικρότερες ή ίσες των 20000 rpm, συνεπάγεται ότι η συχνότητα είναι:

$$RPS_{Motor} = \frac{20000rpm}{60sec} = 333.3rps \text{ OR } F_{Maxspeed} = 333.3Hz .$$

Άρα η περίοδος είναι περίπου 3 msec.

Επομένως θεωρείται ότι για να έχουμε ορθή λειτουργία των διακοπών θα πρέπει η αποστολή κάθε πακέτου ως σύνολο προς τον υπολογιστή να έχει διάρκεια μικρότερη ή ίση της περιόδου της μέγιστης περιστροφικής ταχύτητας. Επομένως η ταχύτητα της σειριακής θα πρέπει να είναι μεγαλύτερη ή ίση :

'Εστω $\Rightarrow SerialBusSpeed = S.B.S$

$$T_{Send} \left(\frac{sec}{DataPackage} \right) = \frac{270 \frac{bits}{DataPackage}}{S.B.S \frac{bits}{Sec}} \Leftrightarrow S.B.S \frac{bits}{Sec} = \frac{270 \frac{bits}{DataPackage}}{T_{Send} \left(\frac{sec}{DataPackage} \right)} \Rightarrow$$

$$S.B.S \frac{bits}{Sec} = \frac{270}{3 \cdot 10^{-3}} = 90000bps \quad (3.9)$$

Επιλέγεται η ταχύτητα της σειριακής UART0 να είναι ίση με 115200bps.

Ομοίως η επιλογή της ταχύτητας της σειριακή «UART1», η οποία χρησιμοποιείται για την επικοινωνία με το σύστημα μέτρησης ώσης ,διέπεται από τα ίδια κριτήρια. Από το πρωτόκολλο επικοινωνίας για τη λήψη δεδομένων της σειριακής UART1, βλέπε κεφάλαιο 4.3 ,το μέγεθος κάθε πακέτου δεδομένων είναι εννέα bytes .Επιπλέον ο ρυθμός μεταφοράς των εισερχόμενων πακέτων είναι 10Hz με αποτέλεσμα να έχουμε μέγιστο αριθμό μεταφερόμενων bits ανά δευτερόλεπτο ίσο με 900 bps. Όλες οι ρυθμίσεις διατηρούνται ίδιες με αυτές της πρώτης, εκτός της ταχύτητας σύνδεσης. Επομένως εφόσον έχουμε μόνο λήψη δεδομένων τότε η ταχύτητα ποιοτικά θα αντιστοιχεί στη διπλάσια. Άρα μια ταχύτητα μεγαλύτερη ίση της τάξης 9600 bps κρίνεται ικανοποιητική.

Συνοπτικά δίνονται στον παρακάτω πίνακα οι ρυθμίσεις των σειριακών θυρών.

Ρυθμίσεις Σειριακών Θυρών	UART0	UART1
Ρυθμός μετάδοσης Ταχύτητα	115200 bps	9600 bps
Bit ανά χαρακτήρα	8	8
Ισοτιμία	Καμία	καμία
Αριθμός Bits τέλους χαρακτήρα	1	1

Πίνακας 3.2) Ρυθμίσεις σειριακών θυρών UART0 και UART1

3.9) Περιγραφή των περιφερειακών Χρονομετρητών .

Στην LPC2106 υπάρχουν δύο περιφερειακές συσκευές οι οποίες χρησιμοποιούνται υπό την ευρεία έννοια ως ρολόγια ή χρονομετρητές. Τα ρολόγια γενικά είναι κυκλώματα που παράγουν παλμούς ,με ακριβές πλάτος σε ακριβή χρονικά διαστήματα μεταξύ τους. Τα ρολόγια χρησιμοποιούνται για την εκτέλεση

συγκεκριμένων διαδικασιών σε συγκεκριμένους χρόνους δηλαδή το χρονισμό των ενεργειών αυτών. Οι περιφερειακοί χρονομετρητές Timer1 και Timer0 χρησιμοποιούν το περιφερειακό ρολόι του επεξεργαστή «pclk» ως βασικό παλμό χρονισμού τους. Επομένως η βασική συχνότητα λειτουργίας τους είναι η ίδια με του περιφερειακού ρολογιού. Οι δυνατές λειτουργίες που μπορούν να εκτελέσουν είναι αρκετές και επομένως θα αρκεστούμε στην παρουσίαση μονό των χρησιμοποιούμενων από την εφαρμογή στην LPC2106. Όπως έχει αναφερθεί και σε προηγούμενες παραγράφους ο Timer 1 καλείται να εκτελέσει δύο λειτουργίες ,την τήρηση του χρόνου λειτουργίας του προγράμματος και την καταγραφή του εισερχόμενου παλμού από τον υπέρυθρο φωτοανιχνευτή. Παρακάτω αναλύονται οι απαιτούμενες ρυθμίσεις του Timer 1 για την εκτέλεση των παραπάνω δυο λειτουργιών.

3.9.1) Χρονισμός των ρολογιών Timer0 & Timer1.

Οι δύο χρονομετρητές διαθέτουν ένα καταχωρητή T0TC & T1TC αντίστοιχα οι οποίοι λειτουργούν ως μετρητές ,δηλαδή σε κάθε παλμό του ρολογιού τους αυξάνουν την τιμή τους κατά 1 μονάδα μέχρι τη μέγιστη τιμή τους. Ο χρονισμός του κάθε ρολογιού επειδή χρησιμοποιεί το περιφερειακό ρολόι του επεξεργαστή pclk, του οποίου η ταχύτητα του μπορεί να είναι πολύ υψηλή, διαθέτει τον καταχωρητή TOPR & T1PR οι οποίοι λειτουργούν ως διαιρέτες της ταχύτητας του περιφερειακού ρολογιού.

TIMER1

Έχουμε ορίσει ότι η ταχύτητα λειτουργίας του περιφερειακού ρολογιού είναι $pclk=3686400$ Cycles per Sec ή Hz. Για να χρησιμοποιηθεί ο Timer1 ως χρονομετρητής ορίζεται ότι η διακριτική ικανότητα του αρκεί να είναι της τάξης των 10^{-3} Sec. Επιπλέον για να λειτουργήσει ως δέκτης καταγράφοντας μια ιδιότητα του σήματος από τον φωτοανιχνευτή, βλέπε κεφάλαιο 4 μέτρηση στροφών, η ταχύτητα του πρέπει να είναι ικανή να καταγράψει την τιμή του χρόνου της ανόδου ή καθόδου του σήματος με ακρίβεια ώστε να επιτύχουμε τη μέτρηση των στροφών. Υπενθυμίζεται ότι η σχέση υπολογισμού της ταχύτητα δίνεται από τη σχέση $\bar{V} = \frac{N}{T \pm t}$, όπου N ο αριθμός των καταγραφών των περασμάτων από το υπέρυθρο ανιχνευτή ,T η διάρκεια των περασμάτων και t το σφάλμα στη μέτρηση του χρόνου και ισούται με την περίοδο της συχνότητας λειτουργίας του ρολογιού. Επομένως το σφάλμα στη μέτρηση της ταχύτητας δίνεται από τη σχέση ,βλέπε Εργαστηριακές ασκήσεις Φυσική Κεφ3.6 έκδοση 1999.

$$\delta \bar{V} = \frac{N}{T} \cdot \sqrt{\left(\frac{\delta N}{N}\right)^2 + \left(\frac{\delta T}{T}\right)^2} \xrightarrow{\frac{\delta N=0}{\delta T=t}} \delta \bar{V} = \bar{N} \cdot \frac{t}{T^2} \Rightarrow$$

$$\delta V = N \cdot \frac{t}{T^2} \quad (3.10)$$

Επομένως για επιθυμητό σφάλμα $2 \cdot 10^{-1}$ RPM ή $2 \cdot 10^{-1}/60 = 3.333 \cdot 10^{-3}$ rps θα πρέπει στη μέγιστη ταχύτητα των 20.000 RPM η 333.3 rps δηλαδή N=20.000 σε ένα λεπτό θα πρέπει το σφάλμα στην ανάγνωση του χρόνου να είναι :

$$\delta V = \bar{N} \cdot \frac{t}{\bar{T}^2} \Leftrightarrow t = \frac{\delta \bar{V} \cdot \bar{T}^2}{\bar{N}} = \frac{3.333 \cdot 10^{-3} \cdot 1^2}{333.3} = 10^{-5} \text{ sec} \quad (3.11)$$

Για την ικανοποίηση και των δυο λειτουργιών στο ρολόι Timer1 υπερισχύει η μικρότερη τιμή ,δηλαδή θα πρέπει η συχνότητα του ρολογιού να είναι μεγαλύτερη ίση των 10^{-5} Hz .Επομένως ο καταχωρητής T1PR μπορεί να πάρει τις παρακάτω τιμές. Για την εξαγωγή της παρακάτω σχέσης βλέπε LPC2106/2105/2104 USER MANUAL , preliminary Supersedes data of 2003 Sep 17.

$$Timer1_{Clock_Rate} (Hz) = \frac{PLCK}{T1PR+1} \geq 10^5 \Rightarrow T1PR+1 \leq \frac{PLCK}{10^5} \Leftrightarrow$$

$$T1PR \leq \frac{PLCK}{10^5} - 1 = \frac{3686400}{10^5} - 1 = 35,864 \quad (3.12)$$

Ο καταχωρητής T1PR δέχεται μόνο ακέραιες τιμές και επιλέγω την τιμή $3 < 35$. Η συχνότητα λειτουργίας του Timer1 είναι :

$$Timer1_{Clock_Rate} (Hz) = \frac{3686400}{3+1} = 921600 \text{ Hz} \quad (3.13)$$

TIMER 0

Οι λειτουργίες που εκτελεί ο χρονομετρητής είναι α) ο χρονοισμός της επικοινωνίας με τον υπολογιστή στέλνοντας της μετρήσεις β) εκτέλεση της μέτρησης της περιστροφικής ταχύτητας του κινητήρα και γ) περιοδικός έλεγχος της ύπαρξης σειριακής επικοινωνίας με τον υπολογιστή .Δηλαδή ο ρόλος του περιορίζεται στο χρονοισμό των προαναφερόμενων λειτουργιών. Οι καταχωρητές που περιγράφουν το χρονοισμό του ρολογιού έχουν την ίδια λειτουργικότητα με τον Timer1 .Τίθεται ως περίοδος εκτέλεσης των προαναφερθέντων διαδικασιών, α-γ, τα $20 \cdot 10^{-3} \text{ Sec}$ ή 50Hz. Η τιμή αυτή προκύπτει από την περίοδο επικοινωνίας με τον παλμοτροφοδοτικό μετατροπέα τάσης η οποία είναι η μεγαλύτερη απαιτούμενη σε σχέση με όλα τα χρησιμοποιούμενα συστήματα. Η χρήση μικρότερης περιόδου είναι ανούσια εφόσον ο έλεγχος του παλμοτροφοδοτικού μπορεί να πραγματοποιηθεί μόνο με την παραπάνω συχνότητα. Εξίσου σημαντική είναι η ακριβής προσέγγιση της περιόδου αυτής, με το μικρότερο δυνατόν σφάλμα. Το μικρότερο σφάλμα μπορεί να επιτευχθεί με τη λειτουργία του timer0 στη μεγαλύτερη δυνατή συχνότητα δηλαδή ίση με τη συχνότητα του περιφερειακού ρολογιού PLCK.

$$Timer0_{Clock_Rate} (Hz) = \frac{PLCK}{TPR+1} = PLCK \Rightarrow T0PR+1 = \frac{PLCK}{PLCK} \Leftrightarrow$$

$$T0PR+1=1 \Rightarrow T0PR=0 \quad (3.14)$$

Άρα επιλέγεται για τον καταχωρητή T0PR η τιμή 0.

3.9.2) Διαμόρφωση των λειτουργιών των χρονομετρητών.

Timer1

Η δυνατότητα μέτρησης του μετρητή του «Timer1» έχει όριο τον αριθμό 2^{32} , με αποτέλεσμα για να συνεχιστεί ορθά το πρόγραμμα, την επανεκκίνηση της καταμέτρησης του χρονομετρητή από την τιμή 0, ειδικά θα υπερχειλίσει και η εκτέλεση του προγράμματος θα είναι απρόβλεπτη. Η επανεκκίνηση του T1TC από τη τιμή 0, όταν φτάσει τη μέγιστη τιμή του, ρυθμίζεται στον καταχωρητή TIMCR ο οποίος ελέγχει τις λειτουργίες του T1TC. Στον TIMCR παρέχεται η δυνατότητα επιλογής τριών λειτουργιών που θα εκτελέσει ο T1TC α) αίτηση διακοπής στον VIC, β) επανεκκίνηση γ) παύση λειτουργίας όταν η τιμή του μετρητή γίνει ίση με μία καθορισμένη. Επίσης οι επιλογές αυτές δίνονται να ορισθούν για μέχρι 4 τιμές του T1TC. Η λειτουργία του TIMCR έχει ως εξής. Σε κάθε χρονική στιγμή ελέγχεται η τιμή του T1TC και αν είναι ίση με μία από τις ορισμένες TIMR0~3 τότε εκτελεί τις αντίστοιχες επιλεγμένες λειτουργίες για την τιμή αυτή του T1TC. Η τιμή του T1TC στην οποία ορίζονται να εκτελεστούν οι παραπάνω λειτουργίες εισάγονται στους καταχωρητές TIMR0~3. Για τη λειτουργία του ως χρονομετρητής πρέπει να μπορεί να μετρήσει για ικανό διάστημα ώστε να ολοκληρωθεί η λειτουργία του προγράμματος χωρίς επανεκκίνηση, διότι τότε θα χαθεί η συνέχεια μέτρησης του χρόνου. Όμως σύμφωνα με την ταχύτητα του Timer1 η μέγιστη τιμή του T1TC= 2^{32} θα επιτευχθεί σε $2^{32}/921.600 = 4660,337\text{Sec} \approx 77,67\text{min}$, χρονικό διάστημα το οποίο δεν κρίνεται αρκετό. Επιπλέον για να ξεπεραστεί αυτός ο περιορισμός επιλέγεται η χρήση της αίτησης διακοπής όταν ο T1TC φτάσει τη μέγιστη τιμή του με σκοπό την καταγραφή σε μία μεταβλητή του αριθμού των επανεκκινήσεων του T1TC, βλέπε κεφάλαιο 4, αλγόριθμος καταγραφής του χρόνου. Επομένως ελλείψει άλλης χρονικής στιγμής στην οποία επιθυμείται η εκτέλεση άλλων λειτουργιών επιλέγεται να τεθεί στον καταχωρητή TIMR0 η μέγιστη τιμή του T1TC= 2^{32} και στον TIMCR επιλέγονται οι λειτουργίες επανεκκίνησης και διακοπής «interrupt» όταν η τιμή του T1TC γίνει, ίση με τον καταχωρητή TIMR0, μέγιστη.

Η δεύτερη λειτουργία του Timer1 είναι να καταγράφει τον παλμό από τον υπέρυθρο ανιχνευτή με σκοπό την ανίχνευση της περιστροφικής ταχύτητας του κινητήρα. Η εκτέλεση του αλγόριθμου μέτρησης της ταχύτητας απαιτεί από τον Timer1 την καταγραφή των κορυφών του παλμού οι οποίες αντιστοιχούν σε ένα πέρασμα, περιστροφή, και επιπλέον της χρονικής στιγμής στην οποία πραγματοποιήθηκε αυτή. Από τις δυνατότητες των χρονομετρητών δεν μπορεί να βρεθεί λειτουργία αναγνώρισης της κορυφής του παλμού άμεσα, παρά μόνο χρησιμοποιώντας κώδικα αυξάνοντας την επεξεργαστική ισχύ. Το πρόβλημα αυτό επιλύεται με την παρατήρηση ότι εφόσον ο παλμός είναι περιοδικός, με περίοδο ίση με της περιστροφής του κινητήρα, και διαθέτη μία μόνο κορυφή καθ' όλη την περιστροφή τότε η μετάβαση του παλμού προς την κορυφή και το αντίθετο θα είναι και αυτή περιοδική και ίση με την ίδια περίοδο του παλμού. Η απαίτηση της ίδιας περιόδου πηγάζει από διατήρηση της μορφής του παλμού ανεξάρτητα της περιστροφικής ταχύτητας εφόσον τα γεωμετρικά χαρακτηριστικά του σφονδύλου από τα οποία δημιουργείται ο παλμός παραμένουν τα ίδια. Δηλαδή αν $F(t)$ περιγράφει τον παλμό όπου $t \in [T, T + 1]$ και T η περίοδος, πρέπει να ισχύει η σχέση :

$$F\left(\frac{t}{T_1}\right) = F\left(\frac{t}{T_2}\right), T_1 \neq T_2. \quad (3.15)$$

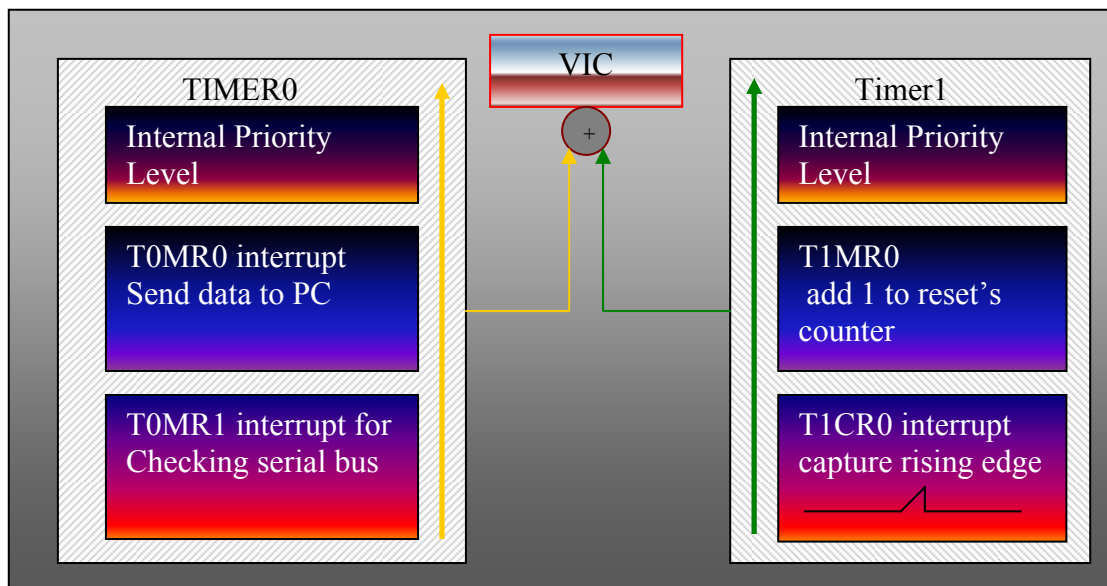
Επομένως η καταγραφή της κορυφής αντικαθίσταται από την καταγραφή της ανόδου του παλμού και της χρονικής στιγμής στην οποία συμβαίνει αυτή. Ο Timer1 διαθέτει τη δυνατότητα αυτή στον οποίο ο καταχωρητής T1CCR ελέγχει τις λειτουργίες καταγραφής «Capture» των εισερχόμενων σημάτων. Ο καταχωρητής αυτός ελέγχει τη λειτουργία καταγραφής μέχρι τεσσάρων σημάτων ταυτόχρονα όπου για κάθε σήμα ορίζονται τρεις επιλογές. Η τιμή του μετρητή T1TC αποθηκεύεται στον αντίστοιχο καταχωρητή T1CR0~3 για κάθε σήμα από το αντίστοιχο ενεργό κανάλι όταν συμβεί μια από τις παρακάτω επιλογές. Οι επιλογές είναι α) όταν το σήμα αυξάνεται ,μεταβαίνει από κατάσταση 0 σε κατάσταση 1 β) όταν το σήμα μειώνεται, μεταβαίνει από κατάσταση 1 σε κατάσταση 0 γ) αίτηση διακοπής όταν πληρωθούν τα κριτήρια οποιασδήποτε από τις καταστάσεις α,β. Επομένως για την εκτέλεση του αλγόριθμου μέτρησης των στροφών θα χρησιμοποιηθεί το κανάλι 0,Capture1.0 και για την καταγραφή του σήματος επιλέγονται οι ρυθμίσεις α και γ . Με τις ρυθμίσεις αυτές καταγράφεται η άνοδος του σήματος και ζητείται διακοπή σε κάθε σύλληψη ώστε να αθροιστεί το γεγονός της ανόδου σε μια μεταβλητή που έχει τον ρόλο μετρητή των περιστροφών του κινητήρα.

Η λειτουργία του αλγόριθμου μέτρησης του χρόνου, εφόσον ο timer1 χρησιμοποιεί δύο εσωτερικές πηγές διακοπής, απαιτεί στη συνάρτηση εκτέλεσης της διακοπής πρώτα να εκτελείται η διακοπή λόγω επανεκκίνησης του μετρητή T1TC και στη συνέχεια η διακοπή λόγω λήξης του σήματος σύμφωνα με το κριτήριο ανόδου.

Timer0

Όπως έχει προαναφερθεί ο σκοπός του χρονομετρητή είναι ο χρονισμός της επικοινωνίας και του ελέγχου της σειριακής θύρας. Η εκπλήρωση των δυο παραπάνω λειτουργιών επιτυγχάνεται με την επαναληπτική λειτουργία του ρολογιού μεταξύ των τιμών 0 και 20 msec .Επομένως χρησιμοποιούνται οι δύο καταχωρητές ,T0MR0 & T0MR1,η λειτουργία των οποίων είναι όμοια με τους αντίστοιχους καταχωρητές του TIMER1. Ο πρώτος χρησιμοποιείται για το χρονισμό ,στα 20 msec, δηλαδή της εκτέλεση της επικοινωνίας και της μέτρηση της ταχύτητας του κινητήρα καθώς και για την επανεκκίνηση του χρονομετρητή από την τιμή 0. Ο δεύτερος θα χρησιμοποιηθεί για τον έλεγχο της σειριακής επικοινωνίας όταν η τιμή του μετρητή T0TC γίνει ίση με το ήμισυ της περιόδου των 20msec. Ο λόγος που εκτελείται ο έλεγχος σε διαφορετική χρονική στιγμή είναι, ότι αν ο έλεγχος ενσωματώνονταν και αυτός στη συνάρτηση επικοινωνίας θα αύξανε το χρόνο εκτέλεσης της. Επιπλέον εφόσον οι λειτουργίες αυτές εκτελούνται περιοδικά έχει ως συνέπεια και η κατανάλωση της επεξεργαστικής ισχύος να κατανέμεται περιοδικά. Επομένως θέτοντας την εκτέλεση του ελέγχου σε διαφορετική στιγμή επιτυγχάνεται η κατανομή της επεξεργαστικής ισχύος σε περισσότερες από μία στιγμές. Αποτέλεσμα είναι η ομαλότερη ζήτηση και λειτουργία, μειώνοντας τον κίνδυνο καθυστερήσεων. Γενικά η χρήση μικρότερων και ταχύτερων αλγόριθμων επιτυγχάνει μεγαλύτερη ευελιξία σε σχέση με τη δημιουργία λιγότερων σε αριθμό αλλά απαιτητικότερων αλγορίθμων λόγω των μεγαλύτερων χρόνων ολοκλήρωσης τους. Ισχύει γενικά η αρχή ότι σε μία ομάδα εκτελούμενων λειτουργιών η βραδύτερη καθορίζει αποφασιστικά και το συνολικό χρόνο εκτέλεσης όλης της ομάδας.

Με τον καταχωρητή T0MCR ορίζονται οι εκτελούμενες λειτουργίες στις χρονικές στιγμές T0MR0 & T0MR1. Στο μεν πρώτο επιλέγεται, συμπληρώνοντας τα αντίστοιχα bit, η ζήτηση διακοπής, για την εκτέλεση της επικοινωνίας και επιπλέον επανεκκίνηση του μετρητή από την τιμή μηδέν «reset». Για το δεύτερο επιλέγεται η αίτηση διακοπής στον VIC για την εκτέλεση του ελέγχου της σειριακής. Εφόσον στον Timer0 ορίζονται δύο αιτήματα διακοπής ορίζεται ότι τη μεγαλύτερη εσωτερική προτεραιότητα στην εκτέλεση των αιτημάτων από τη συνάρτηση εξυπηρέτησης του timer0 έχει το αίτημα του καταχωρητή T0MR0 και απαιτείται ο T0MR1. Τέλος οι καταχωρητές T0TCR & T1TCR χρησιμοποιούνται για την εκκίνηση, τερματισμό και επαναφορά των μετρητών στην αρχική τιμή 0. Συνοπτικά η κατανομή των εσωτερικών διακοπών των χρονομετρητών δίνονται στο παρακάτω σχήμα.



Σχήμα 3.8) Κατανομή εσωτερικών προτεραιοτήτων διακοπής «interrupts» για τους χρονομετρητές Timer0 και Timer1.

Στον πίνακα που ακολουθεί συνοψίζονται οι ρυθμίσεις των χρονομετρητών TIMER1 & TIMER0.

	Timer0	Timer1	Functions at Timer's MCR & CCR control registers
Prescale Register (PR)	0	3	
Timer speed	$PLCK/(PR+1)=3686400$	$PLCK/(3+1)=921600$	

MR0	$20 \cdot 10^{-3} \text{ sec} * 3686400$ or 20mSec	$MaxT1TC = 2^{32}$ 77.67min	Interrupt & reset
MR1	$10 \cdot 10^{-3} \text{ sec} * 3686400$ or 10mSec	Not Used	Interrupt
CR0	Not Used	Rising edge	Interrupt

Πίνακας 3.3) Ρυθμίσεις των χρονομετρητών Timer0 & 1.

3.10) Περιγραφή λειτουργίας παλμού PWM (Pulse Width Modulator)

Η δημιουργία παλμού PWM χρησιμοποιείται στην οδήγηση του παλμοτροφοδοτικού μετατροπέα τάσης και επιπλέον για στο χρονισμό της εκτέλεσης του υπολογισμού του νέου Duty Cycle που θα εφαρμοστεί. Ο επεξεργαστής διαθέτει συνολικά 6 εξόδους παλμού PWM από τους οποίους θα χρησιμοποιηθεί μόνο η έξοδος 2 «PWM2». Από τις απαιτήσεις της μορφής που πρέπει να πληροί ο παλμός στην έξοδο επιλέγεται η δημιουργία παλμού μονής ακμής «single edge». Τα χαρακτηριστικά αυτά ρυθμίζονται στον καταχωρητή PWMPCR συμπληρώνοντας τα κατάλληλα πεδία bit του.

3.10.1) Χρονισμός του ρολογιού δημιουργίας του παλμού.

Ο επεξεργαστής για τη δημιουργία του παλμού χρησιμοποιεί το μετρητή PWMTC. Ο μετρητής χρησιμοποιείται για το χρονισμό του παλμού σε συνεργασία με τους υπόλοιπους καταχωρητές και παίζει ουσιαστικά το ρόλο του εσωτερικού ρολογιού δημιουργίας του παλμού. Η συχνότητα αύξησης της τιμής του είναι ίση με τη συχνότητα του περιφερειακού ρολογιού διαιρεμένη με τον PWMPR καταχωρητή και αυξημένο κατά μία μονάδα, $PWMTC_{Clock_Rate}(Hz) = \frac{PLCK}{PWMPR + 1}$.

Η συχνότητα λειτουργίας του μετρητή είναι η μέγιστη δυνατή ώστε να επιτευχθεί η ανάλογη ανάλυση στη μέτρηση του χρόνου. Επιπλέον θα πρέπει η ταχύτητα αυτή να μην οδηγεί το μετρητή PWMTC σε υπερχείλιση πριν επιτευχθεί η ολοκλήρωση του παλμού. Οι δύο παραπάνω αρχές έχουν ως συνέπεια ο PWMPR να πάρει την τιμή 0, ώστε η συχνότητα του PWMTC να είναι ίση με τη συχνότητα του περιφερειακού ρολογιού, καλύπτοντας και τη δεύτερη απαίτηση. Η περίοδος του παλμού ορίζεται στο καταχωρητή PWMMR0.

Η περιοδική επανάληψη του παλμού ορίζεται στον καταχωρητή PWMMCR. Ο καταχωρητής αυτός ελέγχει τις λειτουργίες που θα εκτελεστούν όταν η τιμή του μετρητή γίνει ίση με τους αντίστοιχους καταχωρητές PWMMR0~6. Για να είναι δυνατή η επαναληπτική δημιουργία του παλμού επιλέγεται η επανεκκίνηση του μετρητή «reset» όταν γίνει ίση με την τιμή του PWMMR0. Η τιμή του PWMMR0 ορίζει τη χρονική στιγμή της ολοκλήρωσης εξαγωγής του παλμού ή αλλιώς την περίοδο του. Ο παλμός μονής ακμής ξεκινά από τη μέγιστη τιμή του «on» και μεταβαίνει στη χαμηλότερη τιμή του «off» τη στιγμή που ορίζεται στον καταχωρητή PWMMR2, όπου και διατηρείται μέχρι το τέλος της περιόδου του παλμού, PWMMR0. Επομένως στον καταχωρητή PWMMR2 αποθηκεύεται η τιμή του επιθυμητού «Duty Cycle» στις αντίστοιχες μονάδες χρόνου. Πρέπει να ισχύει ότι $PWMMR2 \leq PWMMR0$ ειδάλλως ο παλμός θα βρίσκεται συνέχεια στην κατάσταση «on».

Ο χρονισμός της εκτέλεσης της συνάρτησης υπολογισμού του νέου duty cycle επιτυγχάνεται με την αίτηση διακοπής όταν ο μετρητής γίνει ίσος με τον PWMMR2. Στον PWMMCR επιλέγεται η αίτηση διακοπής για τον καταχωρητή PWMMR2.

PWM registers	PWM2 channel	Functions of PWMMCR
PWMPR	0	
PWMTC clock speed	PLCK/ (PWMPR+1)=3686400	
PWMPCR	Single edge output	
PWMMR0	$20 \cdot 10^{-3} \text{ sec} * 3686400$ or 20mSec	Reset
PWMMR2	1~2 msec *3686400	Interrupt

Πίνακας 3.4) Ρυθμίσεις του περιφερειακού συστήματος δημιουργίας παλμού PWM του επεξεργαστή.

3.11) Περιγραφή System Control External Interrupt 1.

Ο επεξεργαστής διαθέτει 3 εξωτερικές εισόδους οι οποίες λειτουργούν ως πηγές αιτημάτων στον ελεγκτή διακοπών. Μπορούν να χρησιμοποιηθούν για την εκτέλεση ποικίλων λειτουργιών όπως αναφέρονται στο εγχειρίδιο χρήσης της LPC2106. Εδώ θα χρησιμοποιηθεί μόνο η είσοδος EINT1 για την εκτέλεση διακοπής η οποία χρησιμοποιείται για αλλαγή της μεθόδου υπολογισμού του «Duty Cycle». Δηλαδή την αλλαγή της συνάρτησης η οποία εκτελείται στη διακοπή της περιφερειακής γεννήτριας παλμού «PWM». Ο σκοπός είναι η επιλογή ανάμεσα στη λειτουργία του ελέγχου των στροφών και της δράσης ελέγχου «Duty Cycle» απευθείας από το πρόγραμμα στον υπολογιστή. Η εναλλαγή ανάμεσα στις δύο επιλογές πραγματοποιείται κατά τη διάρκεια εκτέλεσης της εφαρμογής χωρίς να χρειάζεται ο τερματισμός και φόρτωση νέου προγράμματος στην κάρτα, αυξάνοντας την ευελιξία του έλεγχου της εγκατάστασης.

Ο μόνος καταχωρητής που χρησιμοποιείται είναι ο EXTINT και χρησιμοποιείται από τη συνάρτηση εξυπηρέτησης της διακοπής για εξακρίβωση ποιές από τις τρεις συσκευές εισόδου EXTINT0~2 έχει ζητήσει τη διακοπή.

3.12) Περιγραφή χρήση των ακίδων εισόδου εξόδου του επεξεργαστή «GPIO».

Η λειτουργία των περιφερειακών συσκευών του επεξεργαστή εκτελείται εντός των κυκλωμάτων του επεξεργαστή και οι ακίδες επικοινωνίας του με την κάρτα δεν αντανακλούν υποχρεωτικά τις λειτουργίες αυτές εφόσον δεν έχουν οριστεί κατάλληλα. Κάθε ακίδα «pin», δύναται να χρησιμοποιηθεί για την πρόσβαση σε περιφερειακές συσκευές του. Ορισμός της λειτουργίας κάθε ακίδας του επεξεργαστή πραγματοποιείται με τους καταχωρητές PINSEL0 & PINSEL1 ο καθένας από τους οποίους ελέγχει 16 ακίδες του επεξεργαστή. Περισσότερες πληροφορίες δίνονται στο εγχειρίδιο LPC2106 User manual Appendix .

Η πρόσβαση στις ακίδες του επεξεργαστή πραγματοποιείται μέσω των θυρών BreakoutA & B .

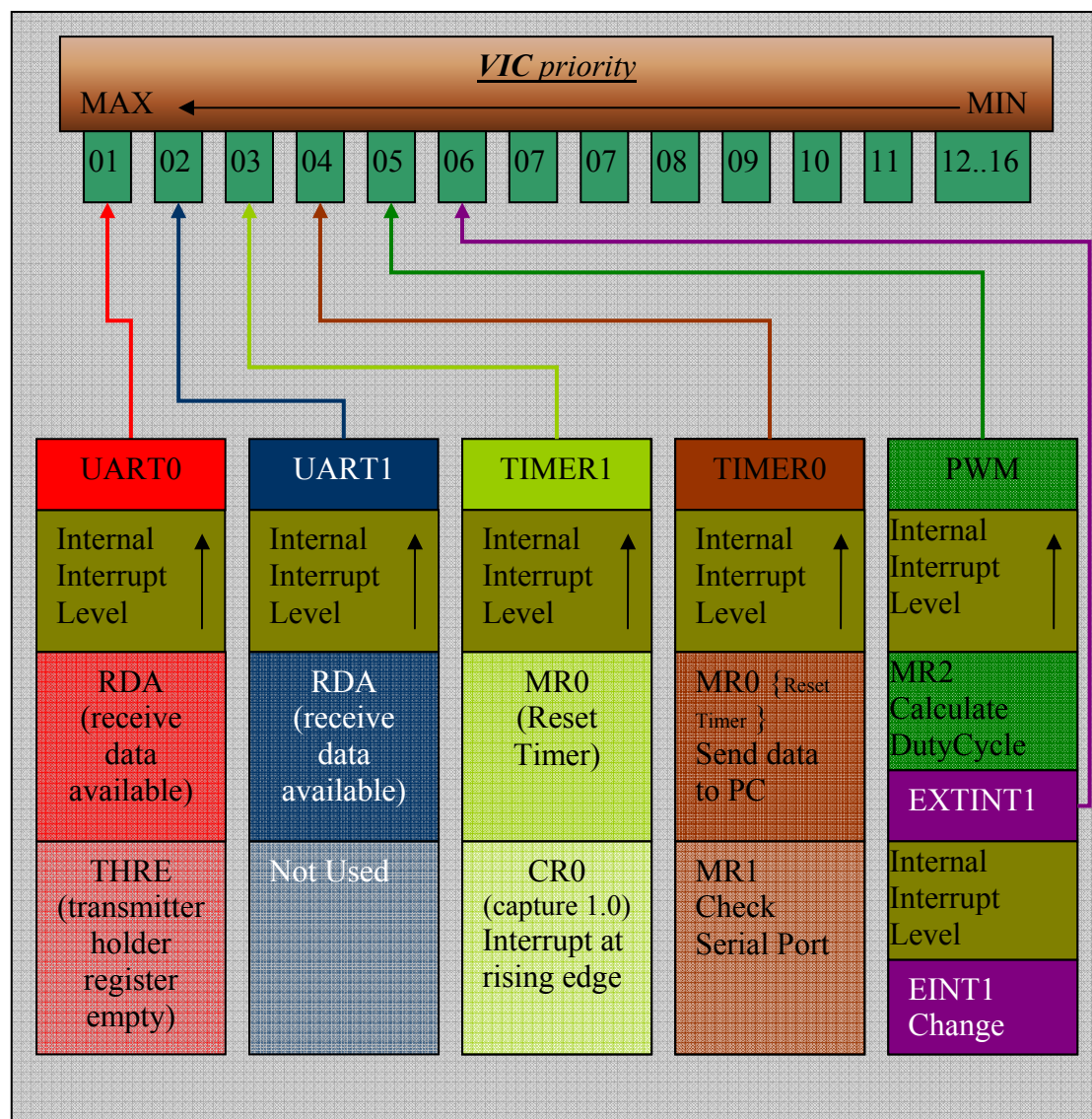
Στον παρακάτω πίνακα δίνονται οι ρυθμίσεις των pins του επεξεργαστή .

Pin Name	Function	Pin Name	Function
P0.0	TxD(UART0)	P0.16	GPIO PORT P0.16
P0.1	RxD(UART0)	P0.17	GPIO PORT P0.17
P0.2	GPIO PORT P0.2	P0.18	GPIO PORT P0.18
P0.3	GPIO PORT P0.3	P0.19	GPIO PORT P0.19
P0.4	GPIO PORT P0.4	P0.20	GPIO PORT P0.20
P0.5	GPIO PORT P0.4	P0.21	GPIO PORT P0.21
P0.6	GPIO PORT P0.5	P0.22	GPIO PORT P0.22
P0.7	PWM2	P0.23	GPIO PORT P0.23
P0.8	TxD(UART1)	P0.24	GPIO PORT P0.24
P0.9	RxD(UART0)	P0.25	GPIO PORT P0.25
P0.10	Capture1.0(Timer1)	P0.26	GPIO PORT P0.26
P0.11	GPIO PORT P0.11	P0.27	GPIO PORT P0.27
P0.12	GPIO PORT P0.12	P0.28	GPIO PORT P0.28
P0.13	GPIO PORT P0.13	P0.29	GPIO PORT P0.29
P0.14	EINT1	P0.30	GPIO PORT P0.30
P0.15	GPIO PORT P0.15	P0.31	GPIO PORT P0.31

Πίνακας 3.5) Ρυθμίσεις των ακίδων του επεξεργαστή για τη χρήση των περιφερειακών συσκευών αυτού.

Σε όλες τις ακίδες στις οποίες έχει οριστεί η λειτουργία Γενικής Εισόδου Εξόδου GPIO(General Purpose Input Output) πρέπει να οριστούν οι παράμετροι λειτουργίας τους. Αρχικά ορίζεται αν θα λειτουργήσουν ως είσοδος ή έξοδος .Λόγοι ασφαλείας καθορίζουν τη λειτουργία όλων ως είσοδο. Η κατεύθυνση λειτουργίας ορίζεται στον καταχωρητή IODIR όπου συμπληρώνονται όλα τα bits του κατάλληλα.

Σύνοψη των Ρυθμίσεων λειτουργίας του επεξεργαστή & σχήμα κατανομής Διακοπών.



Σχήμα 3.9) Σύνοψη ρύθμισης των πηγών διακοπής στον VIC και της εσωτερικής κατανομής τους

3.13) Περιβάλλον γραφής του προγράμματος της LPC2106

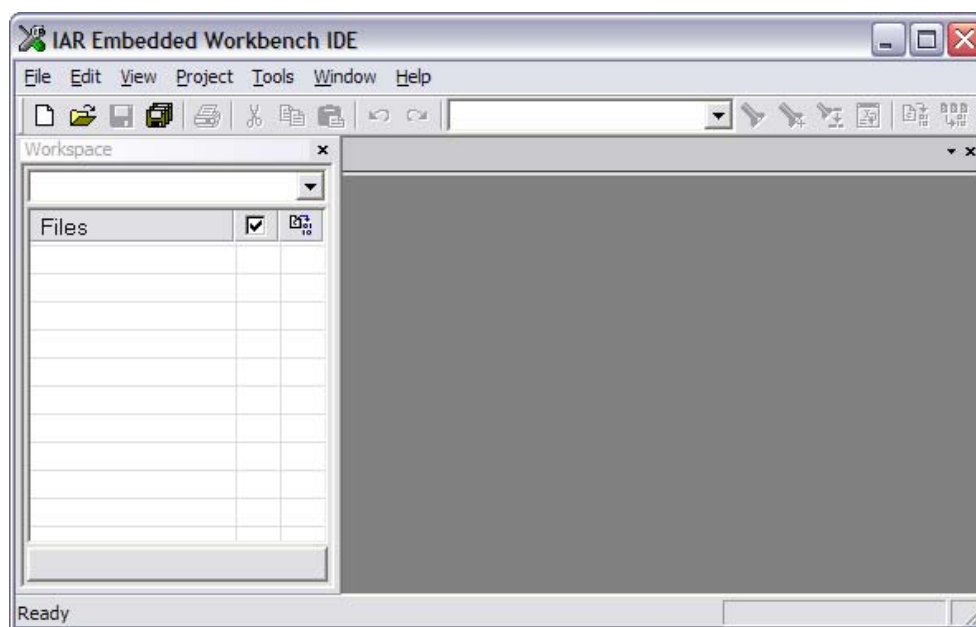
Ο προγραμματισμός της κάρτας LPC2106 πραγματοποιείται μέσω του λογισμικού «IAR Embedded Workbench IDE» το οποίο παρέχεται μαζί με την κάρτα. Στο πρόγραμμα αυτό παρέχεται η δυνατότητα εξαγωγής του πηγαίου κώδικα ο οποίος μεταφέρεται στην κάρτα προς εκτέλεση. Επίσης παρέχει τη δυνατότητα παρακολούθησης της πορείας του και της άμεσης πρόσβασης σε όλες τις περιοχές της μνήμης ενώ εκτελείται με σκοπό τη παροχή βοήθειας στην ανάπτυξη του κώδικα και την εύρεση λογικών σφαλμάτων στη γραφή του δηλαδή την εκσφαλμάτωση, η κοινός debug. Η επικοινωνία αυτή πραγματοποιείται είτε μέσω σειριακού καλωδίου είτε με τη χρήση του συστήματος J-link του οποίου η σύνδεση είναι τύπου USB.

Το πρόγραμμα αυτό υποστηρίζει τη γραφή του κώδικα σε δύο γλώσσες προγραμματισμού την C και την επέκτασή αυτή της C++ καθώς και τη γλώσσα assembly. Οι γλώσσες αυτές δεν είναι άμεσα υλοποιήσιμες στον επεξεργαστή και τη

μετάφραση τους σε γλώσσας μηχανής αναλαμβάνει ο λεγόμενος compiler assembler. Στην περίπτωση που ο κώδικας είναι γραμμένος σε C ή C++ υλοποιείται ένα ενδιάμεσο στάδιο μετάφρασης κατά το οποίο ο κώδικας μεταφράζεται σε assembly.

Για τη γραφή του κώδικα χρησιμοποιήθηκε η γλώσσα C ως γλώσσα ανώτερου επιπέδου έναντι της assembly και η οποία είναι ευρύτερα διαδεδομένη στη συγγραφή προγραμμάτων για τη λειτουργία επεξεργαστών. Δεν είναι γλώσσα «πολύ υψηλού επιπέδου» και δεν εξειδικεύεται σε κανένα συγκεκριμένο τομέα εφαρμογών. Όμως η απουσία περιορισμών και η γενικότητα της την καθιστούν πιο δυνατή και αποτελεσματική από άλλες γλώσσες. Η C δεν έχει δεσμούς με οποιοδήποτε υλικό ή σύστημα και είναι εύκολο να γράφουν προγράμματα που θα τρέχουν χωρίς αλλαγές σε οποιοδήποτε μηχάνημα που την υποστηρίζει. Την υποστήριξη αυτή όπως προαναφέρθηκε την αναλαμβάνει ο μεταφραστής που περιέχεται στο χρησιμοποιούμενο περιβάλλον και ο οποίος έχει αναπτυχθεί ειδικά για την υποστήριξη των επεξεργαστών του τύπου ARM. Ο μεταφραστής εκτός από το σύνολο των εντολών C και μερικών εξαιρέσεων έχει συμπληρωθεί με την εισαγωγή ορισμένων επιπλέον «μακροεντολών» επεκτείνοντας τις διαθέσιμες εντολές τις οποίες αναγνωρίζει για την απλούστευση ορισμένων διαδικασιών. Οι εντολές αυτές αναφέρονται στο εγχειρίδιο του μεταφραστή EWARM Compiler Reference

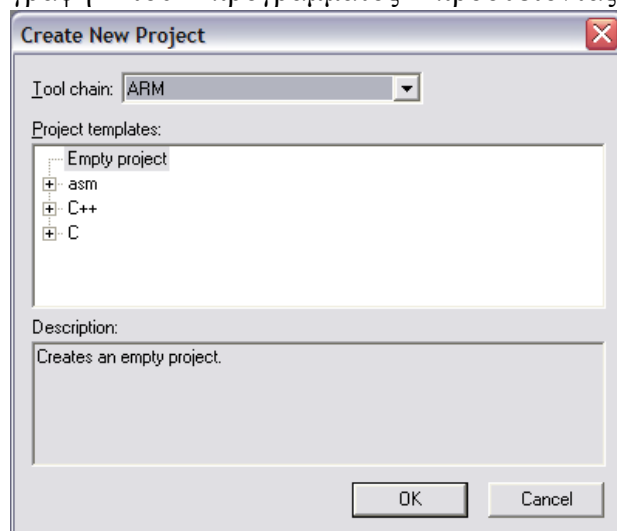
Η χρήση του προγράμματος για τη γραφή του κώδικα απαιτεί να πραγματοποιηθούν συγκεκριμένες ρυθμίσεις ώστε να μεταφρασθεί ο κώδικας σωστά και να είναι εκτελέσιμος από τον επεξεργαστή. Οι ρυθμίσεις για τη γραφή του κώδικα της κάρτας LPC2106 στην εφαρμογή IAR Embedded Workbench IDE δίνονται αναλυτικά στο κεφάλαιο των βοηθημάτων .



Εικόνα 3.3) Στιγμιότυπο από την εκκίνηση του παραθύρου του προγράμματος IAR «Embedded Workbench IDE»

Αρχικά κατά την εκκίνηση του προγράμματός εμφανίζεται το περιβάλλον χρήσης του προγράμματος στο οποίο εκτός από την περιοχή γραφής του προγράμματος εμφανίζονται το μενού περιεχομένων του προγράμματος καθώς και η μπάρα η οποία περιέχει διάφορες γνωστές εντολές. Για την εκκίνηση ορίζεται ο λεγόμενος χώρος

εργασίας , «workspace» . Στο χώρο αυτό αποθηκεύονται όλες οι απαραίτητες ρυθμίσεις του προγράμματος και τα δημιουργούμενα αρχεία. Για τη δημιουργία του workspace ακολουθούμε τις εντολές File → New → Workspace. Δημιουργούμε νέο Project ακολουθώντας τις εντολές Project → Create New Project και εμφανίζεται η οθόνη στην οποία επιλέγεται Tool chain: → ARM & Project Templates → C . Στη συνέχεια εφόσον έχουμε δημιουργήσει ένα project μπορούμε να ξεκινήσουμε τη γραφή του προγράμματος προσθέτοντας ένα αρχείο στο οποίο θα πραγματοποιήσουμε τη γραφή του κώδικα. Η εισαγωγή του αρχείου γίνεται με τις εντολές File → New → Source/Text.

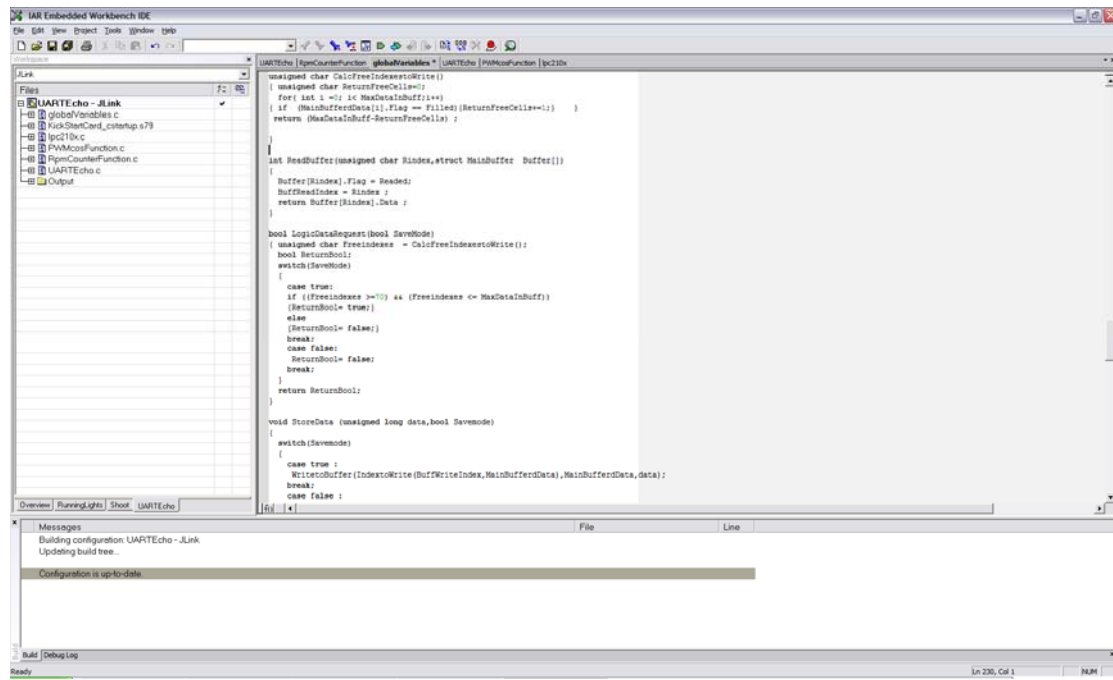


πραγματοποιήσουμε τη γραφή του κώδικα. Η εισαγωγή του αρχείου γίνεται με τις εντολές File → New → Source/Text.

Στη συνέχεια εμφανίζεται το επιλεγμένο αρχείο για τη γραφή του προγράμματος .Στο αριστερό μέρος της οθόνης εμφανίζονται με μορφή δένδροδιαγράμματος τα αρχεία που έχουν ενσωματωθεί στο τρέχον «Project» ,ενώ στο αριστερό εμφανίζονται τα περιεχόμενα κάθε επιλεγμένου αρχείου από το Project.

Στην πάνω γραμμή εμφανίζεται η μπάρα των εργαλείων εκσφαλμάτωσης και εκτέλεσης του προγράμματος .Επίσης για τον έλεγχο των σφαλμάτων και ενημέρωσης του χρήστη στο κάτω μέρος της οθόνης κάτω από το παράθυρο εμφάνισης του αρχείου εμφανίζεται το παράθυρο ενημέρωση της κατάστασης του εκσφαλματωμένου αρχείου καθώς και υπόδειξης του σημείου στο κώδικα που υπάρχει λάθος .Ο χρήστης απλά κάνοντας αριστερό κλικ στο μήνυμα λάθους κατευθύνεται αυτόματα στο σημείο του λάθους για τη διόρθωση του. Σημειώνεται ότι ο έλεγχος των λαθών πραγματοποιείται μόνο για τη σύνταξη του κώδικα και δεν διορθώνει σφάλματα που προκύπτουν κατά την εκτέλεση του κώδικα όπως διαιρέσεις με το μηδέν ή άλλα λάθη που θα προκαλέσουν υπερχείλιση της μνήμης και την καταστροφή της ροής του προγράμματος .Ένα συνήθης λάθος είναι οι πράξεις μεταξύ μεταβλητών διαφορετικού τύπου. Πρέπει να δίνεται ιδιαίτερη προσοχή και να ορίζεται σαφώς ο τύπος του αποτελέσματος της πράξης. Στην παρακάτω εικόνα δίνεται ένα στιγμιότυπο από τη γραφή του κώδικα ,με το παραπάνω πρόγραμμα.

Κεφάλαιο 3: Αρχιτεκτονική και συνδεσμολογία κάρτας IAR.



4) Κεφάλαιο Αλγόριθμοι μετρήσεων και εντολών των Προγραμμάτων.

4.0) Σύνοψη.

Στην επικοινωνία μεταξύ συστημάτων χρησιμοποιούνται τα λεγόμενα πρωτόκολλα επικοινωνίας. Η ανάγκη διασύνδεσης διαφορετικών συστημάτων μεταξύ τους οδήγησε στην ανάπτυξη κανόνων οι οποίοι ρυθμίζουν την επικοινωνία των συστημάτων. Η μεταφορά οργανώνεται με τέτοιο τρόπο ώστε τα συνδεδεμένα συστήματα να είναι ικανά να αναγνωρίσουν τα μεταφερόμενα δεδομένα σύμφωνα με ένα κοινά επιλεγμένο πρότυπο επικοινωνίας. Ένα τέτοιο πρότυπο επικοινωνίας, όπως έχει παρουσιαστεί στο προηγούμενο κεφάλαιο, είναι η ρύθμιση των σειριακών θυρών (ταχύτητα, ισοτιμία, μέγεθος λέξης, Bit τερματισμού). Το πρωτόκολλο αυτό ρυθμίζει την επικοινωνία ανά μεταφερόμενο byte αλλά δεν ορίζει τους κανόνες για την αποστολή δεδομένων των οποίων η πληροφορία δε μπορεί να παρουσιαστεί με ένα μόνο byte. Επομένως για τη μεταφορά δεδομένων με μεγαλύτερο μέγεθος πρέπει να παρεμβληθεί ένα κοινό πρωτόκολλο, το οποίο μορφοποιεί την πληροφορία σε ένα σύνολο από « bytes » και στέλλονται μέσω της σειριακής. Το ίδιο αυτό πρωτόκολλο χρησιμοποιείται από το δέκτη ώστε να επαναδομήσει το σύνολο των bytes στην αρχική μορφή της πληροφορίας.

Στα πλαίσια δημιουργίας των προγραμμάτων παρουσιάστηκε η ανάγκη ανάπτυξης κωδικών επικοινωνίας μεταξύ των προγραμμάτων για λόγους ασφαλείας των μεταφερομένων δεδομένων καθώς και την εξοικονόμηση του συνολικού όγκου δεδομένων που μεταφέρονται. Η χρήση γενικών αλγορίθμων οι οποίοι χρησιμοποιούνται ευρέως στην επικοινωνία μεταξύ συστημάτων θα οδηγούσε στην αύξηση των διακινούμενων bytes γεγονός που έρχεται σε αντίθεση με την αρχή της λιγότερης δυνατής χρήσης των σειριακών θυρών. Οι παραπάνω λόγοι οδήγησαν στην κατασκευή εξατομικευμένων πρωτοκόλλων επικοινωνίας μεταξύ της LPC2106 και της εφαρμογής στον υπολογιστή καθώς και με την επικοινωνία με το σύστημα μέτρησης της ώσης. Οι αλγόριθμοι παρουσιάζονται για το λόγο ότι ο αναγνώστης της πτυχιακής πρέπει να γνωρίζει τα πρωτοκόλλα και τους αλγόριθμους ώστε να του δοθεί η δυνατότητα να μπορεί να συνδέσει οποιοδήποτε σύστημα επιθυμεί και να επικοινωνήσει με κάποιο από τα χρησιμοποιούμενα συστήματα. Ο έλεγχος των συνεργαζόμενων συστημάτων εκτός από το κοινό πρωτόκολλο απαιτεί και τον ορισμό των εντολών οι οποίες ρυθμίζουν τη συνεργασία μεταξύ τους. Η παρουσίαση των εντολών αυτών είναι απαραίτητη για τον αναγνώστη για το χειρισμό των συσκευών.

Στη συνέχεια παρουσιάζονται οι αλγόριθμοι υλοποίησης των μετρήσεων και διεξαγωγής βασικών λειτουργιών της LPC2106 ώστε ο αναγνώστης να μπορεί να χρησιμοποιήσει τα αποτελέσματα γνωρίζοντας τη μέθοδο υπολογισμού τους.

4.1) Πρωτόκολλο σειριακής μετάδοσης δεδομένων από την LPC2106 στον υπολογιστή.

Η αποστολή προς τον υπολογιστή έχει δομηθεί με τον εξής τρόπο :

Η σειριακή θύρα UART διαθέτει για την ορθή λειτουργία της 16 byte μνήμης ξεχωριστά για την αποστολή και λήψη δεδομένων , «buffer». Το μέγεθος αυτό όμως δεν είναι ικανοποιητικό και για το λόγο αυτό μία περιοχή της μνήμης χωρητικότητας 100 bytes χρησιμοποιείται για την προσωρινή αποθήκευση των εξερχόμενων δεδομένων προς τον υπολογιστή. Στη συνέχεια τα δεδομένα από τη θέση αυτή μεταφέρονται στον εσωτερικό buffer της σειριακής για αποστολή μέσω της κλίσης της συνάρτησης η οποία εξυπηρετεί τη μεταφορά των δεδομένων που υπάρχουν στον buffer.

Αρχικά ορίζονται τα δεδομένα που πρόκειται να αποσταλούν στην εφαρμογή του υπολογιστή. Τα δεδομένα αυτά είναι :

1. Το σημείο αναφοράς των στροφών.
2. Οι μετρούμενες στροφές.
3. Το σημείο αναφοράς της δράσης ελέγχου.
4. Η τιμή της δράσης ελέγχου.
5. Η μέτρηση της ώσης από τη δυναμοκυψέλη.
6. Η μέτρηση της τάσης τροφοδοσίας.
7. Η μέτρηση της έντασης ρεύματος.
8. Η χρονική στιγμή κλήσης της συνάρτησης σε (msec) για τον υπολογισμό των παραπάνω 7 μεγεθών
9. Η τιμή της μεθόδου λειτουργίας της εκτέλεσης του προγράμματος επί της κάρτας.
10. Η ένδειξη καταστάσεων λάθους της εφαρμογής.
11. Τιμή ελέγχου εγκυρότητας των απεσταλμένων δεδομένων.

Ως γνωστόν η αναπαράσταση οποιαδήποτε τιμής ή πληροφορίας στους ηλεκτρονικούς επεξεργαστές πραγματοποιείται στο δυαδικό σύστημα. Ο αριθμός των bits που χρησιμοποιούνται για να αναπαραστήσουν μια πληροφορία ορίζεται από τον τύπο της πληροφορίας αυτής. Στο συγκεκριμένο επεξεργαστή η αναπαράσταση ενός πραγματικού αριθμού χρειάζεται 32bits ενώ η αναπαράσταση ενός ακεραίου απαιτεί από 8 έως 32 bits.

Παρακάτω παρουσιάζονται οι πίνακες μεγέθους και εύρους τιμών για κάθε τύπο δεδομένων ακεραίων και πραγματικών.

Type	Size	Range (+/-)	Decimals	Exponent	Mantissa
float	32 bits	$\pm 1.18E-38$ to $\pm 3.39E+38$	7	8 bits	23 bits
double	64 bits	$\pm 2.23E-308$ to $\pm 1.79E+308$	15	11 bits	52 bits

Πίνακας 4.1) Μέγεθος και Εύρος Πραγματικών Τιμών

Data type	Size	Range	Alignment
char	8 bits	0 to 255	1
signed char	8 bits	-128 to 127	1
unsigned char	8 bits	0 to 255	1
short	16 bits	-32768 to 32767	2
signed short	16 bits	-32768 to 32767	2
unsigned short	16 bits	0 to 65535	2
int	32 bits	-2^{31} to $2^{31}-1$	4
signed int	32 bits	-2^{31} to $2^{31}-1$	4
unsigned int	32 bits	0 to $2^{32}-1$	4
long	32 bits	-2^{31} to $2^{31}-1$	4
signed long	32 bits	-2^{31} to $2^{31}-1$	4
unsigned long	32 bits	0 to $2^{32}-1$	4
long long	64 bits	-2^{63} to $2^{63}-1$	4
signed long long	64 bits	-2^{63} to $2^{63}-1$	4
unsigned long long	64 bits	0 to $2^{64}-1$	4

Πίνακας 4.2) Μέγεθος και Εύρος Ακεραίων Τιμών

Από τους παραπάνω πίνακες φαίνεται ότι η αναπαράσταση ενός αριθμού σε μορφή πραγματικού και η αποστολή του θα απαιτούσε τουλάχιστον 32 bits με αποτέλεσμα για τα 11 διαφορετικά δεδομένα που θέλουμε να στείλουμε θα απαιτούνται 352 bits ή 44 bytes. Από το δεύτερο πίνακα γίνεται φανερό ότι η αναπαράσταση ενός ακεραίου είναι πιο ευέλικτη και εξαρτάται άμεσα από την τιμή μου και τον αντίστοιχο τύπο. Η παρατήρηση αυτή και συνυπολογίζοντας την απαίτηση της οικονομίας των μετακινούμενων bytes οδήγησε στην απόφαση της χρήσης των ακεραίων τιμών των δεδομένων εφόσον δεν έχουμε απώλεια πληροφορίας. Τελικός σκοπός είναι η χρήση των λιγότερων δυνατών bytes ανά δεδομένο ώστε το τελικό σύνολο των απαιτούμενων bits και για τα 11 να είναι ελάχιστο. Επομένως όλα τα δεδομένα τα οποία πρόκειται να αποσταλούν μετατρέπονται στις ακεραίες τιμές τους. Η ακεραία τιμή ενός αριθμού οποιουδήποτε τύπου είναι ίση με τη μικρότερη από τις πλησιέστερες ακεραίες τιμές του αριθμού αυτού. Το μόνο δεδομένο το οποίο δεν είναι τύπου ακεραίου είναι η τιμή της μετρούμενης ώση η οποία όμως με κατάλληλο μετασχηματισμό μετατρέπεται σε ακεραίο χωρίς απώλεια της πληροφορίας.

Στον παρακάτω πίνακα δίνονται οι ελάχιστες και μέγιστες τιμές για κάθε δεδομένο το οποίο πρόκειται να αποσταλεί μέσω της σειριακής.

Δεδομένο	τύπος	Ελάχιστη τιμή	Μέγιστη τιμή
Σημείο Αναφοράς Στροφών ανά λεπτό	Ακέραιος	0	20000
Μετρούμενες Στροφές ανά λεπτό του Κινητήρα	Ακέραιος	0	20000
Σημείο αναφοράς δράσης	Ακέραιος	0	10000

ελέγχου			
Τρέχουσα δράση ελέγχου	Ακέραιος	0	10000
Μετρούμενη ώση	Δεκαδικός με 4 δεκαδικά ακρίβεια	0	10.0000
Μέτρηση τάσης τροφοδοσίας ακροδεκτών κινητήρα	Δεκαδικός με 3 δεκαδικά ακρίβεια	0	20.000
Μέτρηση έντασης ρεύματος κινητήρα	Δεκαδικός με 2 δεκαδικά ακρίβεια	0	50.00
Χρόνος μέτρησης	Ακέραιος	0	2^{32}
Λέξη ελέγχου εγκατάστασης	χαρακτήρας	0	255
Λέξη λαθών	χαρακτήρας	0	255
Λέξη ελέγχου εγκυρότητας αποστελλόμενων δεδομένων	χαρακτήρας	0	255

Πίνακας 4.3) Μέγιστες και ελάχιστες τιμές αποστελλόμενων δεδομένων

Έχει υπολογιστεί ότι 27 bytes αρκούν για τη μεταφορά των δεδομένων μέσω της σειριακής σύμφωνα με την παρακάτω αναφερόμενη διάταξή τους. Τα δεδομένα συγκεντρώνονται από τη συνάρτηση που είναι υπεύθυνη για την αποστολή των δεδομένων στον «buffer» της σειριακής με κατάλληλη διαμόρφωση. Έχει οριστεί ότι κάθε φορά που θα καλείται η συνάρτηση αποστολής δεδομένων θα δημιουργείται μια ομάδα των 27 bytes .

Στη σύνθεση και αποστολή της ομάδας των bytes λαμβάνουν χώρα προσπελάσεις των σημείων της μνήμης οι οποίες περιέχουν τα δεδομένα αυτά με σκοπό τη διάσπαση των πληροφοριών αυτών σε bytes ώστε να σταλούν μέσω της σειριακής θύρας. Για λόγους ορθής αποστολής των δεδομένων η πρόσβαση στη σειριακή επικοινωνία γίνεται μόνο μέσω της συνάρτησης αποστολή δεδομένων. Περισσότερες πληροφορίες στο επόμενο κεφάλαιο.

Παρακάτω εξηγείται η διαμόρφωση της ομάδας των 27 bytes και η αντιστοιχία κάθε byte στα δεδομένα προς αποστολή. Σε κάθε ένα των 27 αντιστοιχείται ένας αριθμός, ως διακριτικό π.χ. Byte (0) ,ο οποίος δείχνει τη σειρά αποστολής από τη σειριακή. Το Byte (0) αντιστοιχεί ως το πρώτο byte προς αποστολή ενώ το byte (26) ως το τελευταίο αντίστοιχα.

0~1

Τα πρώτα δύο byte, δηλαδή [(0,1 από τα 27) Byte(0) & Byte(1)], εκφράζουν το σημείο αναφοράς των στροφών. Η τιμή του σημείου αναφοράς παίρνει τιμές από 0 έως 20000 για την οποία αρκούν 16bits για την έκφραση & αποστολή. Επομένως η τιμή του χωρίζεται σε δύο byte το οποίο το Byte(0) περιέχει τα bits από MSB (15) έως 8bit και το Byte(1) = {7bit~ LSB} .

Π.χ.

16 bit = 15300 Rpm	Byte(0)	Byte(1)
(MSB) 0011 1011 1100 0100 (LSB) = 15300	0011 1011	1100 0100

Πίνακας 4.4) Μετασχηματισμός 16bit ακεραίου σε δύο byte.

2~3

Η μέτρηση των στροφών παίρνει ακέραιες τιμές από 0 έως 30000 και επομένως αρκούν 16bits για την έκφραση & αποστολή της. Όμοια με την προηγούμενη διαδικασία ανάλυσης 16bit ακεραίου αριθμού σε δύο byte μετατρέπεται η τιμή των στροφών στα Byte(2) και Byte(3). Δηλαδή το Byte(2) = { MSB~8bit} και το Byte(3) = {7bit~ LSB} .

4~5

Το σημείο αναφοράς της δράσης ελέγχου παίρνει τιμές από 0 έως 10000 επομένως αρκούν 16bits για την έκφραση και αποστολή της .Αντίστοιχα με τις παραπάνω διαδικασίες η δράση ελέγχου μετατρέπεται σε 2 byte τα Byte(4) = { MSB~8bit} & Byte(5) = {7bit~ LSB} .

6~7

Η εφαρμοζόμενη δράση ελέγχου παίρνει τιμές από 0 έως 10000 & επομένως αρκούν 16Bits για την έκφρασή της και αποστολής της .Αντίστοιχα με τις προηγούμενες διαδικασίες μετατρέπεται η 16 Bit τιμή σε δύο byte ,Byte(6) = { MSB~8} & Byte(7)={7bit~ LSB} .

8~11

Η μετρούμενη τιμή της δυναμοκυψέλης παίρνει τις ακέραιες τιμές από 0 έως 10.0000 . Η τιμή αυτή πολλαπλασιάζεται με το 10000 ώστε να προκύψουν ακέραιες από 0 έως 100000 .Επομένως η τιμή της δυναμοκυψέλης ,για να εκφραστεί και να αποσταλεί ,χρειάζεται περισσότερα από 20 bits .Επιλέγεται η χρήση ενός 32bit ακεραίου για την έκφραση της. Για την αποστολή χωρίζεται σε 4 bytes τα οποία είναι Byte(8) = { MSB~ 24bit} ,Byte(9)={23bit~16bit},Byte(10)={15bit~8bit} & Byte(11)={7 bit~LSB} .

32 Bit = 99000	Byte(8)	Byte(9)	Byte(10)	Byte(11)
(MSB) 0000 0000 - 0000 0001 - 1000 0010 - 1011 1000 (LSB)	0000 0000	0000 0001	1000 0010	1011 1000

Πίνακας 4.5) Μετασχηματισμός 32bit ακεραίου σε τέσσερα byte.

12~15

Η μετρούμενη τάση παίρνει τιμές από 0 έως 20,000 .Για την αποστολή η τιμή αυτή πολλαπλασιάζεται με 1000 ώστε να προκύψουν ακέραιοι από 0 έως 20000 .Αρκούν δηλαδή 16bits. Επιλέγεται όμως η χρήση 32bit ακεραίου για την έκφραση της τροποποιημένης τιμής της τάσης .Για την αποστολή χωρίζεται σε 4 bytes όμοια με την παραπάνω διαδικασία , Byte(12) = { MSB~ 24bit} ,Byte(13)={23bit~16bit},Byte(14)={15bit~8bit} & Byte(15)={7 bit~LSB} .

16~19

Η μετρούμενη ένταση ρεύματος του κινητήρα παίρνει τιμές από 0 έως 50,00 .Για τη μεταφορά της πολλαπλασιάζεται με 100 ώστε να προκύψουν ακέραιες τιμές από 0 έως 5000 .Αρκούν 16 bits για την έκφραση της όμως θα χρησιμοποιηθούν 32Bits .Για την αποστολή χωρίζεται σε 4 bytes όμοια με την παραπάνω διαδικασία , Byte(16) = {MSB~ 24bit} ,Byte(17)={23bit~16bit},Byte(18)={15bit~8bit} & Byte(19)={7 bit~LSB} .

20~23

Η τιμή του χρόνου είναι ένας 32bit ακέραιος με μονάδες msec .Για την αποστολή του λοιπόν χωρίζεται σε 4 bytes όμοια με τις προηγούμενες διαδικασίες , Byte(20) = {MSB~ 24bit} ,Byte(21)={23bit~16bit},Byte(22)={15bit~8bit} & Byte(23)={7 bit~LSB}. Για τη διαμόρφωση των τεσσάρων Bytes του χρόνου χρησιμοποιείται μια ελαφρά παραμετροποιημένη μέθοδος και η οποία παρουσιάζεται στο κεφάλαιο 4.5 .

24~25

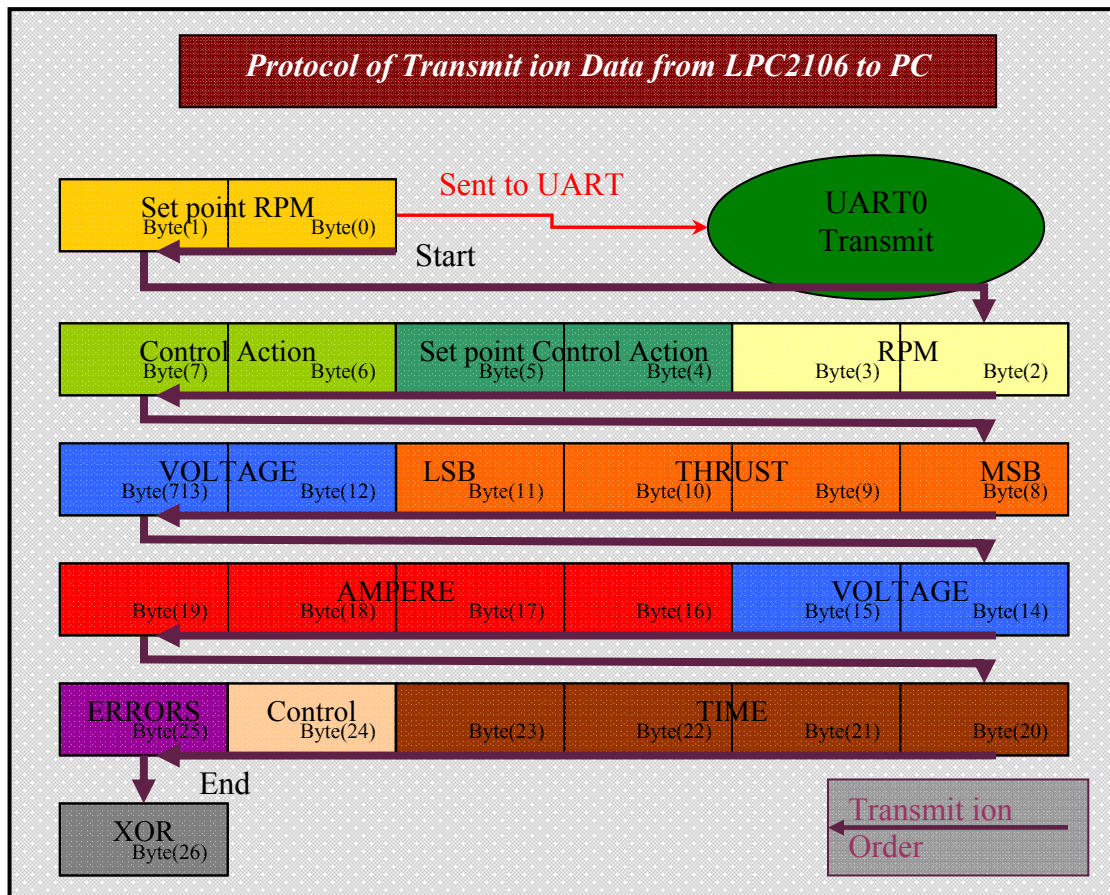
Η κατάσταση λειτουργίας του προγράμματος εκφράζεται από 1 byte το οποίο αποστέλλεται με το byte(24). Το byte αυτό παίρνει την τιμή του με την εκτέλεση του αλγόριθμου που δίνεται στο κεφάλαιο 4.6 .Το «byte» αυτό ονομάζεται «control byte». Η αποστολή των καταστάσεων λάθους εκφράζεται επίσης με 1 byte το οποίο αποστέλλεται με το byte(25).

26

Το Byte ελέγχου εγκυρότητας δημιουργείται από τα προηγούμενα 26 bytes .Το byte ελέγχου είναι ίσο με την XOR σύνθεση του προηγούμενου με το επόμενο byte, Byte(26) = Byte(0) XOR Byte(1) XOR XOR Byte(24) XOR Byte(25) .

Η ομάδα αποστολής έχει τη μορφή :

[Byte(0), Byte(1) , Byte(2) , Byte(3) , Byte(4) , Byte(5) , Byte(6) , Byte(7) , Byte(8), Byte(9) , Byte(10) , Byte(11) , Byte(12) , Byte(13) , Byte(14) , Byte(15) , Byte(16), Byte(17) , Byte(18) , Byte(19) , Byte(20) , Byte(21) , Byte(22) , Byte(23) , Byte(24), Byte(25) , Byte(26)] .



Σχήμα 4.1 Σχεδιάγραμμα πρωτοκόλλου και σειράς αποστολής των δεδομένων του αλγόριθμου αποστολής από την LPC2106 στον υπολογιστή. Με ίδιο χρώμα εμφανίζονται τα bytes τα οποία χρησιμοποιούνται για την ίδια μέτρηση.

Με σκοπό την αύξηση της ασφάλειας των μεταφερόμενων δεδομένων οι τιμές των bytes 8,12&16 στην παραπάνω σειρά αποστολής αντικαθίστανται από τις τιμές 3,4,5 αντίστοιχα, στο δεκαδικό σύστημα. Οι τιμές 3,4,5 υπολογίζονται επίσης για τη δημιουργία του χαρακτήρα ελέγχου. Ο χαρακτήρας ελέγχου στο τέλος της ομάδας των μεταφερόμενων δεδομένων χρησιμοποιείται από την εφαρμογή στον υπολογιστή να ελέγξει τα εισερχόμενα δεδομένα ως προς τη σωστή μεταφορά τους από τη σειριακή και την απόρριψη τους σε περίπτωση λάθους.

Σημείωση.

Η μετατροπή ενός ακεραίου αριθμού σε μια σειρά από bytes στηρίζεται στη γραφή όλων των αριθμών στο δυαδικό σύστημα στη μνήμη. Η μνήμη αποτελείται από ένα πλήθος κελιών ή θέσεων κάθε ένα από τα οποία μπορεί να περιέχει μία πληροφορία. Το κάθε κελί έχει ένα αριθμό που ονομάζεται διεύθυνση, με το οποίο τα προγράμματα αναφέρονται σε αυτό. Στους επεξεργαστές που χρησιμοποιούν το δυαδικό σύστημα οι διευθύνσεις της μνήμης εκφράζονται επίσης με το δυαδικό σύστημα. Το κελί είναι η μικρότερη προσπελάσιμη μονάδα μνήμης και έχει μέγεθος στον επεξεργαστή αυτό 8bits ή byte. Επομένως κάθε πληροφορία που αποθηκεύεται ή χρησιμοποιείται μετατρέπεται στην αντίστοιχη της στο δυαδικό σύστημα και στη συνέχεια αποθηκεύεται στη διεύθυνση μνήμης που της υποδεικνύεται. Επομένως γνωρίζοντας τη διεύθυνση μπορούμε να μεταβούμε στην περιοχή της μνήμης και να

διαβάσουμε οποιαδήποτε από τα bits της πληροφορίας που περιέχονται. Με τον ορισμό του τύπου μιας πληροφορίας ορίζεται και το μέγεθος της μνήμης η οποία χρειάζεται για να αποθηκευτεί η πληροφορία. Τα byte μιας λέξης μπορεί να αριθμούνται από τα αριστερά προς τα δεξιά ή αντίστροφα. Το πρώτο σύστημα ονομάζεται Μεγάλου άκρου (big endian) και το δεύτερο μικρού άκρου (little endian). Στον επεξεργαστή αυτό χρησιμοποιείται το σύστημα μεγάλου άκρου. Η γνώση του παραπάνω συστήματος είναι σημαντική διότι η φορά της γραφής των bits στη μνήμη ορίζει και τη φορά ανάγνωσης της. Γνωρίζοντας τα παραπάνω για να διαβαστεί ένας αριθμός σε μία περιοχή της μνήμης χρειάζεται η διεύθυνση, η φορά ανάγνωσης και ο αριθμός των bits τα οποία θέλουμε να διαβάσουμε. Με τον όρο MSB (Most Significant Bit) συμβολίζεται το bit το οποίο βρίσκεται πρώτο στη φορά γραφής και το LSB (List Significant Bit) τελευταίο. Όταν αποθηκεύεται ένας τύπος δεδομένου σε μια διεύθυνση στη μνήμη τότε δεσμεύεται μία περιοχή της με αρχή την διεύθυνση και μήκος ίσο με το απαιτούμενος μήκος σε bit από τύπο του δεδομένου. Όπως αναφέρεται παραπάνω όλα τα δεδομένα προς αποστολή μετατρέπονται στην ακέραια μορφή τους με μέγεθος 16 ή 32bits. Επομένως ο καθένας από αυτούς τους αριθμούς μπορεί να διασπαστεί σε 2 ή 4 Bytes. Ένας ακέραιος γράφεται στη μνήμη στη δυαδική μορφή του ξεκινώντας από το πρώτο bit που ορίζει η διεύθυνση και συμπληρώνοντας διαδοχικά τα υπόλοιπα bits μέχρι το τελευταίο. Άρα γνωρίζοντας το μέγεθος που καταλαμβάνει ο ακέραιος μπορούμε να χωρίσουμε την περιοχή της μνήμης σε διαδοχικές περιοχές των 8bits αρχίζοντας το πρώτο bit ή MSB και προχωρώντας προς το τέλος. Έτσι προκύπτει μια σειρά από bytes διαδοχικά μεταξύ τους τα οποία αν διαβαστούν με τη σωστή σειρά θα προκύψει πάλι ο ίδιος ακέραιος που είναι αποθηκευμένος στην περιοχή της μνήμης. Π.χ για τον αριθμό 20000 με χρήση 16 bits γράφεται στη μνήμη ως 0100 1110 0010 0000 διασπάται σύμφωνα με τα παραπάνω στα δυο bytes 01001110 και 00100000. Επομένως αν ξαναγραφτούν στη μνήμη με τη σωστή σειρά τότε θα προκύψει πάλι ο 20000.

Address	MSB	16 bits Integer														LS B
Bits No	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Int 20000	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0
Bytes No	Byte(1) = 01001110								Byte(2) = 00100000							

Πίνακας 4.6) Παράδειγμα μετατροπής του ακεραίου αριθμού 20000 σε δύο bytes με βάση τη φορά ανάγνωσης της μνήμης (big endian).

Η χρήση του ψηφιακού τελεστή, XOR για τη δημιουργία του byte ελέγχου χρησιμοποιείται ευρέως στη μεταφορά δεδομένων. Ο ψηφιακός αυτός τελεστής ονομάζεται τελεστής αποκλειστικής σύζευξης και θέτει 1 στα bit των θέσεων όπου οι δύο τελεστέοι έχουν διαφορετικά bit και μηδενίζει τα bit των θέσεων που είναι ίδια. Η επίδραση επί των τεσσάρων διακεκριμένων ζευγαριών των 0 και 1 παρουσιάζεται στον παρακάτω πίνακα.

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Πίνακας 4.7) Επίδραση του ψηφιακού τελεστή XOR επί των bits.

Παράδειγμα της επίδρασης του τελεστή αποκλειστικής σύζευξης σε δυο byte με τιμή 12 και 5 είναι η επιστροφή ενός byte με τιμή 9. Αναλυτικότερα το byte 12 γράφεται στο δυαδικό ως 0000 1100 και το 9 ως 0000 0101.

A	0	0	0	0	1	1	0	0	=12
B	0	0	0	0	0	1	0	1	=5
A XOR B	0	0	0	0	1	0	0	1	=9

Πίνακας 4.8) Παράδειγμα επίδρασης του ψηφιακού τελεστή XOR στα bytes 12 & 5.

Σύμφωνα με τα παραπάνω λοιπόν η χρήση του τελεστή αποκλειστικής σύζευξης μεταξύ των bytes τα οποία στέλνονται θα δημιουργήσει ένα byte το οποίο εξαρτάται άμεσα από τα «bytes» με τα οποία υπολογίστηκε. Αν και η χρήση πολλών bytes αυξάνει την πιθανότητα να δημιουργηθούν περισσότερα λάθη του ενός τέτοια ώστε να οδηγήσουν στη μη ανίχνευση των λαθών, λόγω αλληλοαναιρέσεις τους, ωστόσο είναι προτιμότερο να υπάρχει κάποιος έλεγχος.

4.2) Πρωτόκολλο σειριακής μετάδοσης δεδομένων από τον υπολογιστή στην προς την LPC2106

Τα δεδομένα τα οποία πρόκειται να στέλνονται στην LPC2106 είναι κατά κύριο λόγο τα σημεία αναφοράς των στροφών και δράσης ελέγχου. Είναι δηλαδή αριθμητικές τιμές. Για το λόγο αυτό το πρωτόκολλο αποστολής θα δομηθεί ανάλογα με το πρωτόκολλο αποστολής από την LPC2106 προς τον υπολογιστή όπως παρουσιάστηκε στο 4.1. Δηλαδή κάθε δεδομένο το οποίο πρόκειται να σταλεί θα διασπάται σε ένα σύνολο bytes και τα οποία θα μεταφέρονται. Στη συνέχεια με κατάλληλες πράξεις στην LPC2106 bytes θα ανασυνταχθούν στον πραγματικό αριθμό.

Τα δεδομένα αποστολής είναι το σημείο αναφοράς της δράσης ελέγχου, το σημείο αναφοράς των στροφών και οι εντολές ελέγχου της LPC2106. Το πρώτο έχει εύρος τιμών από 0 ~ 100,00. Για την αποστολή του μετατρέπεται σε ακέραιο πολλαπλασιαζόμενος με το 100 και επομένως το εύρος τιμών γίνεται 0~10000. Το δεύτερο έχει εύρος τιμών 0~20000. Για την έκφραση των παραπάνω ακεραίων και αποστολή αρκούν 16bits. Οι εντολές ελέγχου αντιστοιχούνται με ακεραίους των οποίων η τιμή είναι μεγαλύτερη του 20000. Ο λόγος της χρήσης των μεγαλύτερων αριθμών για εντολές είναι ότι για τη μετάφραση των εισερχόμενων δεδομένων χρησιμοποιείται ο ίδιος αλγόριθμός όπως και για τα σημεία αναφοράς των στροφών και της δράσης ελέγχου. Έτσι δεν απαιτείται η χρήση ειδικών χαρακτήρων οι οποίοι ορίζουν τον τύπο του εισερχόμενου δεδομένου, αν δηλαδή αφορά τα σημεία αναφοράς ή πρόκειται για εντολή. Το διακριτικό γνώρισμα ανάμεσα στις εντολές ελέγχου και τα σημεία αναφοράς είναι η τιμή του μεταφερόμενου δεδομένου απλουστεύοντας τις διαδικασίες λήψης. Επιπλέον η χρήση κοινού πρότυπου μετάδοσης των δεδομένων αυξάνει την αξιοπιστία της εφαρμογής και αποτρέπει την απώλεια της επικοινωνίας στις μεταβατικές καταστάσεις του προγράμματος όπως θα οριστούν σε άλλο κεφάλαιο. Επομένως οι εντολές ελέγχου αρκεί να αντιστοιχηθούν σε ακέραιες τιμές εκτός του πεδίου τιμών των σημείων αναφοράς. Έτσι επιλέγεται η

χρήση 32bit ή 4 bytes για κάθε τιμή αναφοράς ή εντολής ,η όποια πρόκειται να αποσταλεί .Λόγω της χρήσης της τιμής ως το γνώρισμα ανάμεσα στα σημεία ελέγχου και εντολών πρέπει να χρησιμοποιηθούν και επιπλέον bytes στο πρωτόκολλο με σκοπό τον έλεγχο των μεταφερόμενων bytes ώστε να ανιχνευθούν έγκαιρα τυχόν σφάλματα στη μετάδοση. Επομένως επιλέγεται η οργάνωση των δεδομένων σε ομάδα των 6 bytes στα οποία το πρώτο είναι ένα σταθερό byte το οποίο εκφράζει την αρχή της ομάδας και το τελευταίο χρησιμοποιείται ως το byte ελέγχου για τα μεταφερόμενα δεδομένα. Τα 4 ενδιάμεσα χρησιμοποιούνται για την έκφραση της τιμής αναφοράς η εντολής.

Για την αναλυτική και σχηματική παρουσίαση του πρωτοκόλλου που ακολουθεί ,σε κάθε byte της δημιουργούμενης ομάδας εκχωρούνται τα ονόματα Byte(0) έως Byte(5). Ο αριθμός εντός της παρένθεσης ορίζει τη σειρά με την οποία θα σταλούν ξεκινώντας από το 0 και τελειώνοντας στο 5. Τονίζεται ότι η σειρά αποστολής δεν πρέπει να διακοπεί και να παρεμβληθεί από κάποιο άλλο byte διότι αυτό θα θεωρηθεί ως λάθος στη διαδικασία ελέγχου με το τελευταίο byte(5) και η μεταφερόμενη πληροφορία θα χαθεί.

Η οργάνωση της ομάδας των bytes παρουσιάζεται παρακάτω.

0

Το Byte(0) της ομάδας ορίζεται ότι είναι ο χαρακτήρας « * » .

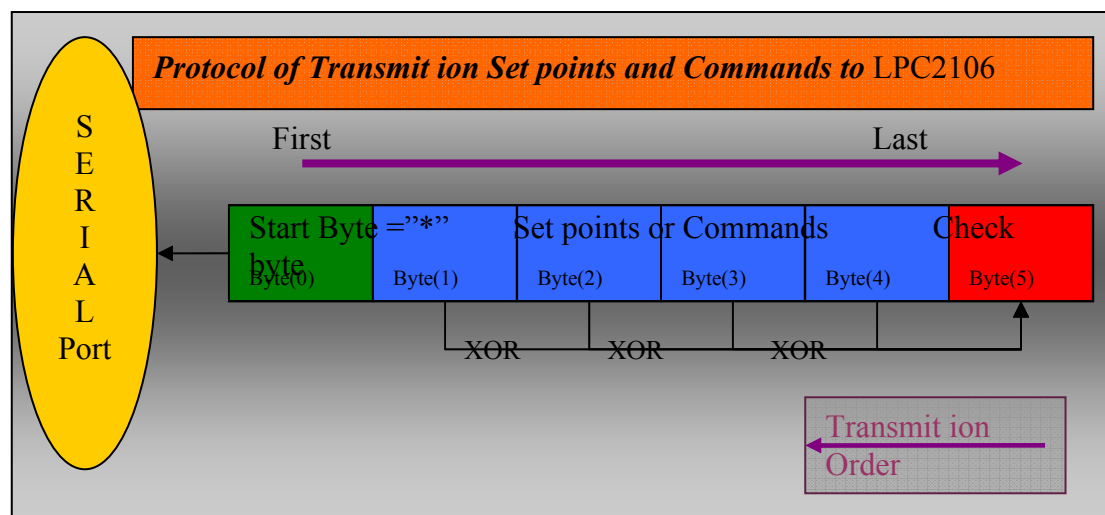
1~4

Τα byte από 1~4 έχουν την παρακάτω δομή .Ο 32bit ,προς αποστολή ,ακέραιος χωρίζεται σε 4 bytes με την παρακάτω μέθοδο Byte(1) = {MSB~24bit} ,Byte(2)={23bit~16bit},Byte(3)={15bit~8bit} & Byte(4)={7 bit~LSB} .

5

Το byte ελέγχου εγκυρότητας δημιουργείται από τα Byte(1)~Byte(4) ως η XOR σύνθεση του προηγούμενου με το επόμενο Byte(i) , Byte(6) = Byte(1) XOR Byte(3) XOR Byte(3) XOR Byte(4) .

Η ομάδα αποστολής έχει τη μορφή [Byte(0), Byte(1), Byte(2), Byte(3), Byte(4), Byte(5)] .



Σχήμα 4.2 Σχεδιάγραμμα πρωτοκόλλου αποστολής των σημείων αναφοράς και εντολών στην LPC2106 από PC.

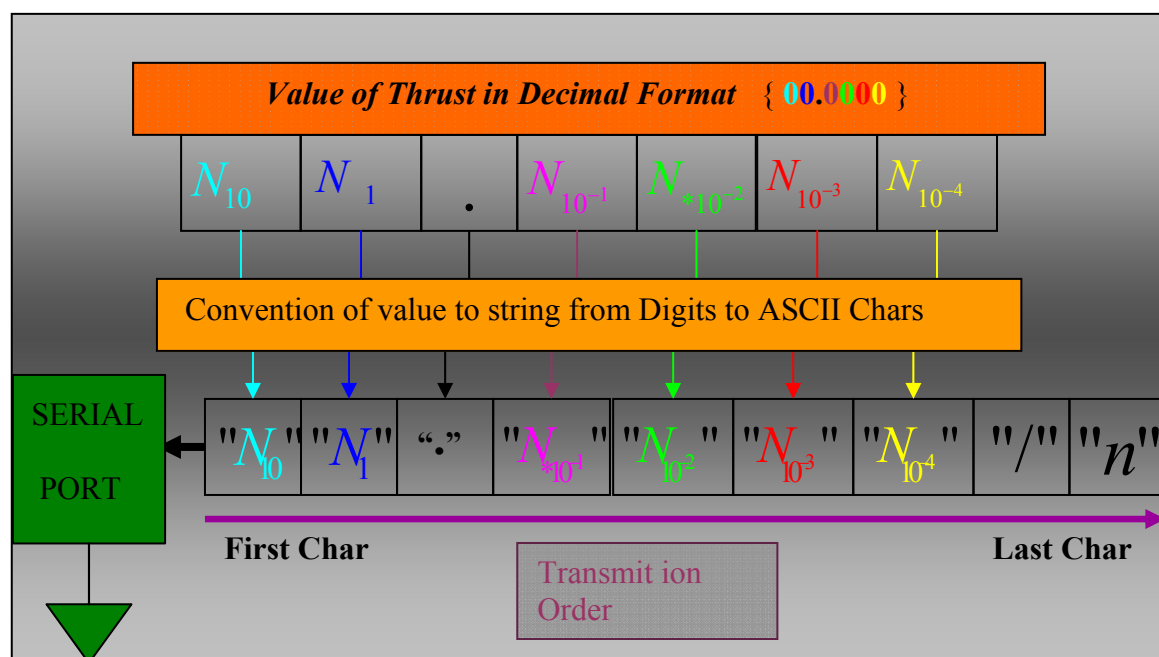
4.3) Πρωτόκολλο λήψης της μέτρησης ώσης στην LPC2106

Από τη συσκευή μέτρησης του «Load Cell», προς την LPC2106 στέλνεται η τιμή της ώσης. Εκτός από την καθεαυτή τιμή στέλνονται επιπλέον χαρακτήρες οι οποίοι χρησιμοποιούνται στο σχηματισμό λέξεων, οι οποίες αφορούν την κατάσταση λειτουργίας της εφαρμογής της μέτρησης της ώσης. Για την εξυπηρέτηση των δύο αυτών σκοπών κάθε αποστέλλόμενο byte από τη συσκευή μέτρησης μεταφράζεται στον αντίστοιχο χαρακτήρα σύμφωνα με το σύστημα κωδικοποίησης χαρακτήρων ASCII και οποίος δίνεται στα παραρτήματα.

Με τον όρο συμβολοσειρά ή «String» ονομάζεται μία σειρά από διαδοχικούς χαρακτήρες. Κάθε χαρακτήρας έχει μήκος ενός byte ή 8 bits. Οι τιμές των 8 bits ορίζουν ένα μοναδικό θετικό ακέραιο στο δεκαδικό σύστημα και σύμφωνα με αυτόν, μετατρέπεται στο αντίστοιχο χαρακτήρα όπως προκύπτει από τον πίνακα ASCII. Χρήση της κωδικοποίησης ASCII στη μετάδοση συμβολοσειρών μέσω σειριακής χρησιμοποιείται ευρύτατα στη βιβλιογραφία για τη μεταφορά αρχείων κειμένου. Κατά τη μέθοδο αυτή η συμβολοσειρά διαβάζεται ξεκινώντας από την αρχή μέχρι το τέλος της και κάθε χαρακτήρας οποίος διαβάζεται μετατρέπεται στο αντίστοιχο byte της κωδικοποίησης ASCII. Με τον τρόπο αυτό δημιουργείται μια διαδοχική σειρά από χαρακτήρες ASCII και στέλνονται μέσω της σειριακής με την ίδια σειρά που έχουν διαβαστεί. Επομένως όλα τα δεδομένα σε αυτήν τη μέθοδο αντιμετωπίζονται ως χαρακτήρες. Πχ, αν θέλω να στείλω τον αριθμό 12, αυτός διαβάζεται από το πρωτόκολλο επικοινωνίας ως το string «12» με αποτέλεσμα τη δημιουργία μιας σειράς από δύο χαρακτήρες, από την οποία ο πρώτος είναι ο χαρακτήρας «1» και δεύτερος ο χαρακτήρας «2». Σημειώνεται ότι ο χαρακτήρας «1» σε ASCII έχει διαφορετική δυαδική τιμή από τη δυαδική τιμή του ακεραίου 1. Η μέθοδος αυτή είναι διαφορετική από τη μέθοδο αποστολής που παρουσιάστηκε στα δύο προηγούμενα πρωτόκολλα, θυμίζετε ότι κάθε δεδομένο προς αποστολή διαβάζεται σύμφωνα με το δυαδικό σύστημα και ανάλογα με τον αριθμό των χρησιμοποιούμενων bits, χωρίζονταν σε διαδοχικές οκτάδες bits ή bytes. Έτσι λοιπόν ο αριθμός 12 θα αντιμετωπίζονταν ως ακέραιος, ο οποίο χωράει σε 1 byte, για την τιμή του οποίου στη

δυναμική της μορφή θα στέλλονταν στη σειριακή χωρίς να χρησιμοποιηθεί ο κωδικοποίηση ASCII.

Το σύστημα μέτρησης της ώσης έχει την ικανότητα της μέτρησης μέχρι 50 κιλών δύναμης με 4 δεκαδικά ανάλυση. Επομένως η ώση υπολογίζεται ως δεκαδικός αριθμός με 4 δεκαδικά ψηφία. Σύμφωνα με το παραπάνω πρωτόκολλο η μορφή του δεκαδικού αριθμού θα είναι [00.0000] όπου με την « . » ορίζεται η υποδιαστολή των δεκαδικών. Επομένως για να σταλεί η τιμή της ώσης θα χρησιμοποιηθεί σύμφωνα με το πρωτόκολλο το string « 00.0000 ». Για τον έλεγχο των σταλμένων χαρακτήρων ως προς το ότι ορίζουν την τιμή της ώσης και δεν προέρχονται από τα χρησιμοποιούμενα λεκτικά της ενημέρωσης της κατάστασης του συστήματος μέτρησης, προστίθεται στο τέλος των επτά χαρακτήρων άλλοι δύο, οι “/” & “n” σε ASCII. Επομένως κάθε φορά που στέλνεται η τιμή της ώσης στέλνονται 9 χαρακτήρες συνολικά σχηματίζοντας την παρακάτω μορφή λεκτικού « 00.0000/n ». Άρα πρώτος στέλνεται ο χαρακτήρας του πρώτου ψηφίου της τιμής 00.0000 όπου εκφράζει τη δεκάδα ,δεύτερος σε σειρά στέλνεται ο χαρακτήρας του δεύτερου ψηφίου της τιμής που εκφράζει τις μονάδες και τρίτος ο χαρακτήρας « . » .Τέταρτος στέλνεται ο χαρακτήρας του ψηφίου του πρώτου δεκαδικού, πέμπτος ο χαρακτήρας του δεύτερου δεκαδικού ,έκτος ο χαρακτήρας του τρίτου δεκαδικού ,έβδομος ο χαρακτήρας του τέταρτου δεκαδικού. Ογδοοι και ένατοι στέλνονται οι χαρακτήρες « / » & « n » αντίστοιχα.



Σχήμα 4.3 Σχεδιάγραμμα πρωτοκόλλου αποστολής της τιμής ώσης στην LPC2106 από το σύστημα μέτρησης.

4.4) Εντολές ελέγχου της LPC2106.

Ο σχεδιασμός όλων των εφαρμογών έχει γίνει υπό την αρχή ότι ο έλεγχος όλων των εφαρμογών και των συστημάτων να διεξάγεται από τον υπολογιστή και κατ' επέκταση το χρήστη αυτού. Επομένως όλα τα συστήματα τα οποία είναι συνδεδεμένα με τον υπολογιστή πρέπει να είναι ικανά να αναγνωρίσουν τις εντολές του, και να αντιδράσουν σύμφωνα με τον καθορισμένο τρόπο. Τα μόνο σύστημα το οποίο είναι

συνδεδεμένο απευθείας με τον υπολογιστή είναι η LPC2106 και στην οποία είναι συνδεδεμένα όλα τα υπόλοιπα συστήματα μετρήσεων και ελέγχου. Όπως έχει προαναφερθεί όλες οι εντολές αντιστοιχούνται σε ακεραίους για τις ανάγκες επικοινωνίας.

Αρχικά ορίζονται οι εντολές που στέλνει η LPC2106 στον υπολογιστή. Κατά την εκκίνηση της δεν ενεργοποιούνται όλες οι λειτουργίες των συνδεδεμένων συσκευών σε αυτή αλλά μόνο η πρώτη σειριακή θύρα « UART0 ». Στην αρχική αυτή φάση στέλνει περιοδικά στον υπολογιστή το χαρακτήρα κατά ASCII «κενό» ή Space ή “ “ ή κατά την αντιστοιχία του στο δεκαδικό σύστημα τον απρόσημο ακεραίο 255. Σε περίπτωση που για κάποιον λόγο δεν έχει εκκινήσει σύμφωνα με τα σχεδιασμένα τότε στέλνει τον χαρακτήρα «□» οποίος αντιστοιχεί στον ακεραίο 254. Για να εκκινήσει ο έλεγχος όλων των συστημάτων ο υπολογιστής πρέπει να στείλει την εντολή εκκίνησης στην LPC2106 η οποία δίνεται παρακάτω. Επιπλέον εκτός των δύο αυτών σημάτων εκκίνησης ορίζονται οι εντολές με τις οποίες ζητεί από τον υπολογιστή την αλλαγή της μεθόδου λειτουργίας σε έλεγχο στροφών ή σε δράση ελέγχου. Οι εντολές αυτές είναι ενσωματωμένες στο «byte ελέγχου» βλέπε 4.1 και δίνονται στο κεφάλαιο παρουσίασης του αλγόριθμου σύνθεσης του Byte ελέγχου.

Στη συνέχεια εφόσον ορίστηκαν οι εντολές που στέλνει η LPC2106 ορίζονται οι εντολές που στέλνει ο υπολογιστής προς αυτήν. Τονίζεται ότι όλες οι εντολές για να σταλούν προς την LPC2106 πρέπει να μορφοποιηθούν σύμφωνα με το πρωτόκολλο αποστολής όπως ορίζεται στο κεφάλαιο 4.2 .Πρώτη δίνεται η εντολή εκκίνησης της πλήρους λειτουργίας της LPC2106 στην οποία χρησιμοποιούνται όλα τα περιφερειακά συστήματα μετρήσεων, ελέγχου και επικοινωνίας .Η εντολή αυτή αντιστοιχείται στον ακεραίο του δεκαεξαδικού συστήματος HEX, 0x52455345. Στη συνέχεια ορίζεται η εντολή τερματισμού της λειτουργίας της LPC2106 η οποία είναι 0xFFFFFFF0 . Με την εντολή αυτή τερματίζονται οι μετρήσεις και τα περιφερειακά συστήματα και επανέρχεται στην αρχική κατάσταση κατά την έναρξή της. Επόμενη εντολή είναι η επείγουσα κράτηση του κινητήρα ,έως τη λήψη της εντολής εκκίνησης του, και η οποία είναι 0xFFFFFFFF. Με την εντολή 0xFFFFFFF0 έχετε η κατάσταση επείγουσας κράτησης και εκκινείτε ο κινητήρας σύμφωνα με την τρέχουσα μέθοδο λειτουργίας. Για να επιβεβαιώσει η LPC2106 την ύπαρξη της σειριακής σύνδεσης στέλνεται σε αυτήν ανά τακτά διαστήματα η εντολή 0x00004E21 όταν είναι αναγκαίο.

Έχει αναφερθεί ότι χρησιμοποιούνται δύο μέθοδοι υπολογισμού της δράσης ελέγχου του κινητήρα η οποία ασκείται μέσω του παλμοτροφοδοτικού μετατροπέα τάσης. Ο πρώτος είναι ο έλεγχος στροφών και δεύτερος ο απευθείας έλεγχος του «duty cycle» ή κανόνας ελέγχου. Με την εντολή 0xFFFFFFFF1 ορίζεται ως τρέχουσα μέθοδος ο έλεγχος στροφών ενώ με την εντολή 0xFFFFFFFF2 θέτεται η μέθοδος απευθείας ελέγχου του Duty Cycle «control action». Με σκοπό την επανάληψη μετρήσεων δίνεται η δυνατότητα ανάγνωσης των επιθυμητών σημείων αναφοράς «Set Points» από αρχείο και την αποστολή του προς εκτέλεση στην LPC2106. Με την εντολή 0xFFFFFFFF3 ενεργοποιείται η επιλογή ανάγνωσης των «set points» από το αρχείο και με την εντολή 0xFFFFFFFF4 συμβαίνει το αντίστροφο .

Συνοπτικά όλες οι διαθέσιμες εντολές παρουσιάζονται στον παρακάτω πίνακα.

Κατεύθυνση	Εντολές σε	Εντολές σε	Λειτουργία εντολής
------------	------------	------------	--------------------

αποστολής	δεκαδικό Σύστημα	δεκαεξαδικό Σύστημα	
LPC2106 προ υπολογιστή	255	0xFF	Αίτηση εκκίνησης μετρήσεων και πλήρους λειτουργίας
	254	0xFE	Ένδειξη εσφαλμένης εκκίνησης
Υπολογιστής προς LPC2106	1380275013	0x52455345	Εντολή εκκίνησης μετρήσεων και πλήρους λειτουργίας
	4294967293	0xFFFFFFFFD	Εντολή τερματισμού μετρήσεων και λειτουργίας
	4294967295	0xFFFFFFFFF	Επείγουσα κράτηση του κινητήρα «σταμάτημα»
	4294967294	0xFFFFFFFFE	Τερματισμός της επείγουσας κράτησης του κινητήρα
	4294967281	0xFFFFFFFF1	Επιλέγεται ο έλεγχος στροφών
	4294967282	0xFFFFFFFF2	Επιλέγεται ο απευθείας έλεγχος Duty Cycle
	4294967283	0xFFFFFFFF3	Χρήση των Set points από αρχείο
	4294967284	0xFFFFFFFF4	Χρήση των Set points από χειριστήρια στον υπολογιστή
	20001	0x00004E21	Έλεγχος ύπαρξης της σειριακής σύνδεσης

Πίνακας 4.9) Σύνοψη εντολών ελέγχου της LPC2106 και υπολογιστή

4.5) Ο αλγόριθμος της μέτρησης και της σύνθεσης του χρόνου κατά την πλήρη λειτουργία της LPC2106.

Σε προηγούμενο κεφάλαιο ορίστηκε ότι για τη μέτρηση του χρόνου στον υπολογισμό της ταχύτητας του κινητήρα, της αποστολής των δεδομένων και γενικά τη λειτουργίας ως ρολόι όλης της εφαρμογής επί της LPC2106 χρησιμοποιείται ο χρονομετρητής Timer1. Σύμφωνα με τις ρυθμίσεις οι οποίες έχουν επιλεγεί ο Timer1 θα εκκινήσει την καταμέτρηση του χρόνου πάλι από την αρχή όταν ο μετρητής φτάσει την μέγιστη τιμή του. Για να συνεχιστεί η μέτρηση του χρόνου και πέρα του χρόνου επανεκκίνησης « reset » χρησιμοποιείται μια μεταβλητή στην οποία αποθηκεύεται ο αριθμός των επανεκκινήσεων του Timer1. Επομένως ο χρόνος υπολογίζεται από το άθροισμα της τιμή του Timer1 και του γινομένου του πλήθους των επανεκκινήσεων επί την περίοδο.

Δηλαδή δίνεται από την παρακάτω σχέση:

$$T(ms) = Timer1_{Count} + N_{Reset} \cdot T_{Reset}$$

, όπου με $Timer1_{Count}$ συμβολίζεται η τρέχουσα τιμή σε msec του Timer1 ,με N_{Reset} το πλήθος των επανεκκινήσεων του Timer1 και T_{Reset} η περίοδος σε msec .Η τιμή του χρόνου $Timer1_{Count}$ υπολογίζεται από τη σχέση:

$$Timer1_{Count}(mSec) = \frac{T1TC}{921600} \cdot \frac{\cancel{Cycles} * \frac{1000mSec}{\cancel{Second}}}{\frac{\cancel{Cycles}}{\cancel{Second}}} = \frac{T1TC \cdot 1000}{921600}$$

Η περίοδος επανεκκίνησης υπολογίζεται από το λόγο της μέγιστης τιμής που είναι ικανός να μετρήσει προς τη συχνότητας λειτουργίας του.

$$T_{Re.set} = \frac{MaxCount}{Timer1_{Speed}(Hz)} \xrightarrow{MaxCount=4294967295} T_{Re.set} = \frac{4294967295}{921600} \Rightarrow$$

$$T_{Re.set} = 4660,337Sec$$

Με σκοπό την αποστολή του χρόνου δειγματοληψίας των μετρήσεων των δεδομένων στον υπολογιστή χρησιμοποιείται μια εξειδικευμένη διαμόρφωση των bytes που αφορούν το χρόνο στο πρωτόκολλο αποστολής όπως παρουσιάστηκε στο κεφάλαιο 4.1 .

Ο σκοπός της διαμόρφωσης των bytes αποστολής του χρόνου είναι η μείωση του υπολογιστικού φόρτου για τον υπολογισμό σύμφωνα με την παραπάνω σχέση. Επιλέγεται να μεταφερθεί με την ανεπεξέργαστη μορφή του. Στην παραπάνω σχέση για το τελικό υπολογισμό του χρόνου χρησιμοποιούνται δύο παράμετροι η τιμή του timer1 και του πλήθος των επανεκκινήσεων .Επομένως μεταφέρονται οι δύο παράμετροι αυτοί στον υπολογιστή ο οποίος αναλαμβάνει τον τελικό υπολογισμό του χρόνου σύμφωνα με τη σχέση

$$T(ms) = Timer1_{Count} + N_{Re.set} \cdot T_{Re.set} \quad \begin{matrix} T_{Re.set}=4660338msec \\ \Rightarrow \end{matrix}$$

$$T(ms) = Timer1_{Count} + N_{Re.set} \cdot 4660338$$

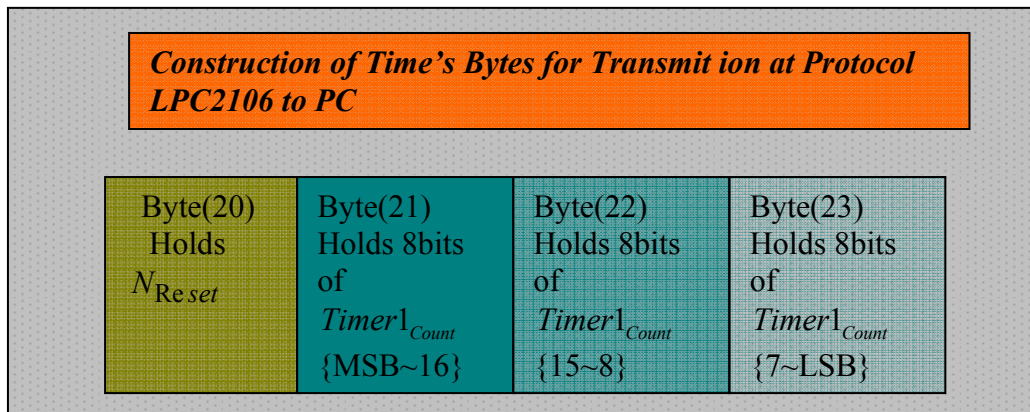
Για τη μεταφορά του χρόνου χρησιμοποιούνται 4 bytes όπως έχει προαναφερθεί. Ο ακέραιος **4660338** ,που αντιστοιχεί στη μέγιστη τιμή των msec που μπορεί να μετρήσει ο Timer1 ,χρησιμοποιείται για την αναπαράσταση του σε δυαδική μορφή λιγότερα από 24bits. Επομένως ορίζονται τα 3 τελευταία bytes από τα 4 χρησιμοποιούμενα να διατεθούν για τη μεταφορά της τιμή σε msec του Timer1.Το ένα επιπλέον byte χρησιμοποιείται για τη μεταφορά του αριθμού των επανεκκινήσεων του «Timer1», ή αλλιώς της τιμής της μεταβλητής « $N_{Re.set}$ ». Εφόσον για την τιμή αυτή χρησιμοποιείται 1 byte ορίζεται και ότι η μέγιστη τιμή της είναι $MAX_N_{Re.set} = 2^8 - 1 = 255$. Το μέγιστο διάστημα το οποίο μπορεί να μετρηθεί χωρίς να προκύψει πρόβλημα είναι:

$$T(ms) = Timer1_{Count} + N_{Re.set} \cdot 4660338 \quad \begin{matrix} Timer1=4660338 \\ \Rightarrow \\ N_{Re.set}=255 \end{matrix} \quad Max_T(ms) = 1193046528msec$$

$$Max_T(ms) = 1193046528msec_or_1193046,528Sec_or_331,4018Hours$$

Άρα η διαμόρφωση των τεσσάρων bytes αποστολής του χρόνου είναι :
Time = {[Byte(1)= $N_{Re.set}$] [Byte(2)~Byte(4) περιέχουν τα 3 bytes Timer1] σε αντιστοιχία με το πρωτόκολλο μεταφοράς του κεφαλαίου 4.1 «Bytes» κατανέμονται:

Byte(20)={ N_{Reset} },Byte(21)={MSB=23bit~16bit} του $Timer1_{Count}$
 Byte(22)={15bit~8bit} & Byte(23)={7 bit~LSB=0} του $Timer1_{Count}$.



Σχήμα 4.4 Σχεδιάγραμμα της διαμόρφωσης των "Bytes" αποστολής για την τιμή του χρόνου στο πρωτόκολλο επικοινωνίας.

4.6) Ο αλγόριθμος σύνθεσης του byte ελέγχου.

Το «byte ελέγχου» χρησιμοποιείται για την πληροφόρηση της εφαρμογής του υπολογιστή της τρέχουσας κατάστασης του προγράμματος της LPC2106. Επιπλέον χρησιμοποιείται για την αποστολή αιτήσεων προς εξυπηρέτηση από τον υπολογιστή είτε αλλαγής κατάστασης είτε ζήτησης δεδομένων όπως θα εξηγηθεί αναλυτικά στο κεφάλαιο αποστολής αρχείου αναφοράς από τον υπολογιστή. Παρακάτω αναλύεται η οργάνωση και διαμόρφωση του byte αυτού προκειμένου να εκτελεστούν οι παραπάνω λειτουργίες. Καταρχήν ως γνωστόν ένα byte αποτελείται από 8bits όπου το πρώτο bit από αριστερά αντιστοιχεί στο MSB και το τελευταίο στο LSB. Η θέση των bits και η αρίθμηση τους στον επεξεργαστή και τη μνήμη παρουσιάζεται στο παρακάτω πίνακα.

Address	Byte							
	MSB							LSB
Bits No	7	6	5	4	3	2	1	0
Usage	Mode	Mode	Stop Motor	Request data	Not used	Not used	Read File	Not used

Πίνακας 4.10) Χρήσης των bits του byte ελέγχου.

Για τη μεταφορά όλων των παραπάνω πληροφοριών δεν θα αντιμετωπιστεί ως μία ενιαία ποσότητα αλλά σε κάθε bit αντιστοιχείται μια πληροφορία. Πρώτα ορίζονται τα bits που θα χρησιμοποιηθούν για την πληροφόρηση της τρέχουσας μεθόδου που χρησιμοποιείται για τον υπολογισμό του «duty cycle» ,έλεγχου των στροφών και δράσης ελέγχου με απευθείας ορισμό. Επίσης θα χρησιμοποιηθούν και για την ζήτηση εναλλαγής ανάμεσα στις δύο παραπάνω μεθόδους. Χρησιμοποιούνται τα bits 7 και 6. Τα δύο bits ορίζουν τις τέσσερις εξής τιμές με τις αντίστοιχες ερμηνείες λειτουργίας.

Control Byte	Bit 7	Bit 6	Λειτουργία
	0	0	Τρέχουσα μέθοδος Έλεγχος στροφών
	0	1	Τρέχουσα μέθοδος Έλεγχος Duty Cycle
	1	0	Αίτηση χρήσης της μεθόδου Έλεγχος στροφών
1	1	Αίτηση χρήσης της μεθόδου Έλεγχος Duty Cycle	

Πίνακας 4.11) Πίνακας τιμών των bits του Byte ελέγχου.

Το «Bit» 5 ορίζεται για την πληροφόρηση του pc αν η επείγουσα κράτηση του κινητήρα είναι ενεργή. Παίρνει την τιμή 1 στο bit όταν **δεν** είναι ενεργή η επείγουσα κράτηση και 0 όταν **είναι** ενεργή.

Το «Bit» 4 χρησιμοποιείται από τον αλγόριθμο χρήσης των σημείων αναφοράς από αρχείο. Με 1 αιτείται στον υπολογιστή να στείλει τις τιμές αναφοράς από το αρχείο και με 0 δεν ζητεί.

Με το «Bit» 1 πληροφορείται ο υπολογιστής αν η λειτουργία ανάγνωσης από αρχείο είναι ενεργή η όχι. Με 1 είναι ενεργή η λειτουργία ανάγνωσης από αρχείο και με 0 δεν είναι.

Τέλος τα Bits 3,2,0 δεν χρησιμοποιούνται και οι τιμές τους μπορεί να έχουν οποιαδήποτε από τις τιμές 0 και 1.

4.7) Ο αλγόριθμος χρήσης του αρχείου των σημείων αναφοράς.

Σκοπός

Η διεξαγωγή των πειραμάτων και μετρήσεων εκτός από τη δυνατότητα του ελέγχου των σημείων λειτουργίας της εγκατάστασης από το χρήστη περιλαμβάνει και τη δυνατότητα χρήσης των σημείων αναφοράς από αρχείο ώστε να είναι δυνατή η επανάληψη των πειραμάτων υπό τις ίδιες συνθήκες.

- Περιορισμοί

Στα χαρακτηριστικά της LPC2106 όπως αναφέρθηκαν στο κεφάλαιο 3 ορίζεται ότι η διαθέσιμη μνήμη της έχει μέγεθος 128 KBytes. Επομένως η μεταφορά όλων των σημείων στην κάρτα θα ήταν δυνατή μόνο για αρχεία τα οποία θα χωρούν στη διαθέσιμη ελεύθερη μνήμη. Για την επίλυση του περιορισμού αυτού επιλέγεται η δέσμευση μέρους της μνήμης και της αποθήκευση σε αυτή μικρών τμημάτων του αρχείου, τα οποία όταν εκτελεστούν ανανεώνονται με νέα τμήματα. Κάθε μια τιμή του αρχείου που χρησιμοποιείται αφαιρείται από τη μνήμη ώστε να χρησιμοποιηθεί η θέση της από τα νέα σημεία όταν αυτά θα ληφθούν. Όταν οι άδειες θέσεις είναι αρκετές ζητείται από τον υπολογιστή να στείλει νέες μέχρι να πληρωθούν.

- Παρατηρήσεις

Η αποθήκευση των σημείων αναφοράς και επανάληψη των πειραμάτων έχει νόημα όταν ως παράμετρος τεθεί ο χρόνος. Δηλαδή όταν τα σημεία εκτελούνται σε συγκεκριμένες χρονικές στιγμές. Για χάρη απλότητας της λειτουργίας της LPC2106

επιλέγεται η «απαλοιφή» της παραμέτρου του χρόνου. Με τον όρο απαλοιφή εννοείται ότι κανονικοποιείται η παράμετρος του χρόνου για τα σημεία αναφοράς έτσι ώστε να είναι γνωστή εκ' το προτέρων ο χρόνος εφαρμογής τους. Έτσι αποφεύγεται ο υπολογιστικός φόρτος της εξακρίβωσης του χρόνου και δεν απαιτείται κάποιου είδους χρονισμού της LPC2106 με τον υπολογιστή. Αποτέλεσμα είναι η απλούστευση και μείωση του χρησιμοποιούμενου κώδικα. Ορίζεται ότι κάθε σημείο αναφοράς στο αρχείο απέχει χρονικά από το προηγούμενο και το επόμενο 20msec, ως ο ελάχιστος χρόνος ανανέωσης του «DutyCycle» του παλμοτροφοδοτικού. Δηλαδή για κάθε σημείο είναι γνωστή η παράμετρος του χρόνου από τη θέση του στο αρχείο αν υποτεθεί ότι τη στιγμή της εφαρμογής του πρώτου σημείου ο χρόνος είναι μηδέν. Επομένως ο χρόνος δεν είναι συνεχής αλλά διακριτός, δηλαδή $t(n) = n \cdot T, T = 20mSec, n \in (0, \infty)\mathbb{N}$. Στο αρχείο περιέχονται οι επιθυμητές τιμές των σημείων αναφοράς στον πλησιέστερο διακριτό χρόνο προς τον επιθυμητό.

Γνωρίζοντας λοιπόν όλα τα παραπάνω εξηγείται ο αλγόριθμος χρήσης του αρχείου στην LPC2106. Καταρχήν ορίζεται ότι ο μέγιστος αριθμός αποθηκευμένων τιμών του αρχείου στη μνήμη είναι 100, πίνακας 100 θέσεων, για κάθε θέση χρησιμοποιούνται 32bits. Επιπλέον για κάθε θέση δεσμεύεται άλλη μία θέση μνήμης ίση με ένα byte το οποίο ονομάζεται «byte» κατάστασης. Ορίζεται ότι σε κάθε ζήτηση της «LPC2106» για αποστολή νέων τιμών από το αρχείο, θα στέλνονται τα δέκα επόμενα σημεία του αρχείου. Επομένως για να ζητήσει η LPC2106 νέα σημεία θα πρέπει να έχει τουλάχιστο δέκα κενές θέσεις. Επιπλέον ο ελάχιστος αριθμός των κενών θέσεων αυξάνεται κατά δέκα επιπλέον θέσεις φτάνοντας συνολικά τις είκοσι «20». Οι είκοσι θέσεις παρέχουν αρκετό χρόνο στον υπολογιστή ώστε να επεξεργαστεί το αίτημα αποστολής και να στείλει τις τιμές πριν εξεταστεί ξανά η αποστολή αιτήματος νέων τιμών από την LPC2106. Η αποστολή του αιτήματος πραγματοποιείται στον αλγόριθμο επικοινωνίας με τον υπολογιστή και χρονίζετε μέσω του Timer0 κάθε 20 msec. Η χρήση του ελάχιστου αριθμού κενών θέσεων ίσο με 10 θα οδηγούσε στην περίπτωση που ο υπολογιστής καθυστερούσε αρκετά, ώστε να μην έχει στείλει έστω ένα σημείο στην LPC2106, να ζητήσει και νέα αποστολή ανεβάζοντας το συνολικό αριθμό αποστολών $2 \cdot 10$ σημεία για 10 θέσεις με αποτέλεσμα την υπερχειλίση. Η χρήση 20 κενών θέσεων παρέχει ικανοποιητική χωρητικότητα έναντι του παραπάνω φαινομένου αφού ο αριθμός των διαθέσιμων θέσεων είναι διπλάσιος από τον πλήθος των αποστελλόμενων ανά αίτηση.

Η αίτηση προς τον υπολογιστή ακολουθεί τον παρακάτω αλγόριθμο. Η εξέταση για την αίτηση νέων τιμών πραγματοποιείται κατά τη δημιουργία του πακέτου επικοινωνίας με τον υπολογιστή σύμφωνα με το πρωτόκολλο στο κεφάλαιο 4.1. Κάθε φορά που εξετάζεται η ζήτηση νέων τιμών διαβάζονται όλες οι θέσεις «100» της χρησιμοποιούμενης μνήμης και υπολογίζεται το πλήθος των κενών θέσεων. Αν είναι περισσότερες ή ίσες με 20 τότε συμπληρώνεται στο bit 4 του Byte ελέγχου σύμφωνα με το προηγούμενο κεφάλαιο με την τιμή 1 ώστε να ζητήσει από τον υπολογιστή νέες τιμές. Σε αντίθετη περίπτωση βάζει 0. Για την εξέταση των κενών θέσεων χρησιμοποιείται το επιπλέον byte «κατάστασης» που ορίστηκε σε κάθε θέση στον πίνακα. Το byte αυτό κάθε φορά που γράφεται μία νέα τιμή σε μία θέση συμπληρώνεται με την ένδειξη «νέα τιμή», ενώ κάθε φορά που χρησιμοποιείται από την LPC2106 συμπληρώνεται η ένδειξη «άδειο». Με τον τρόπο αυτό μπορεί να βρεθούν οι κενές θέσεις της μνήμης.

Η ανάγνωση και χρήση των τιμών από τη μνήμη χρησιμοποιεί πάλι το επιπλέον byte «κατάστασης» της μνήμης αποθήκευσης. Κάθε φορά που ζητείται η ανάγνωση ενός σημείου αναφοράς από μια θέση της μνήμης εξετάζεται αν το αντίστοιχο byte «κατάστασης» της έχει την ένδειξη «Νέα τιμή». Αν ναι επιστέφει την τιμή του σημείου αναφοράς θέτοντας στο byte κατάστασης την ένδειξη «άδειο» και θέτει ως νέα θέση ανάγνωσης την επόμενη. Αν όχι τότε επιστέφει την τιμή αναφοράς της προηγούμενης θέσης και θέτει ως νέα θέση ανάγνωσης την ίδια.

Για τη γραφή μιας τιμής στον πίνακα μνήμης σε μία θέση εξετάζεται αν αυτή η θέση είναι χαρακτηρισμένη ως άδεια και αν ναι γράφεται, συμπληρώνοντας το αντίστοιχο Byte «κατάστασης» ως «Νέα τιμή». Αν όχι τότε γράφεται στην τελευταία θέση στην οποία πραγματοποιήθηκε εγγραφή ,προηγούμενη, διαγράφοντας την τιμή την οποία περιέχει και θέτει ως νέα θέση εγγραφής την ίδια.

4.8) Μέτρηση Στροφών στην Κάρτα Philips .

Σκοπός του αλγόριθμου είναι η επίτευξη της μεγαλύτερης δυνατής ακρίβειας και ταυτόχρονα τη μείωση του θορύβου στον υπολογισμό της, από την αβεβαιότητα του σήματος από τον υπέρυθρο φωτοανιχνευτή. Με το όρο αβεβαιότητα εννοούμε την τυχαία καθυστέρηση της αποστολή του παλμού στην LPC2106 καθώς και στην αντίληψη αυτού. Για την αντιμετώπιση του θορύβου χρησιμοποιείται συχνά στη βιβλιογραφία ο μέσος όρος μεταξύ διαδοχικών μετρήσεων εντός του χρονικού διαστήματος αναφοράς. Το διάστημα αυτό πρέπει να είναι πολύ μεγαλύτερο της χρονικής διάρκειας της αβεβαιότητας ώστε να αλληλοαναιρεθούν μεταξύ τους. Για την επίτευξη της αφαίρεσης του θορύβου θα χρησιμοποιήσω τη μέθοδο του κινητού μέσου όρου, δηλαδή ένα σταθερό αριθμό μετρήσεων πχ, 5, οι οποίες αποθηκεύονται με χρονική σειρά σε ένα πίνακα στη μνήμη. Κάθε νέα μέτρηση αποθηκεύετε στη θέση της παλαιότερης ώστε ο συνολικός αριθμός των αποθηκευμένων μετρήσεων να παραμένει σταθερός. Ο υπολογισμός της ταχύτητας θα προκύψει ως ο μέσος όρος μεταξύ των αποθηκευμένων μετρήσεων. Η παραπάνω μέθοδος αντιστοιχεί στη διεξαγωγή μιας και μόνο, με χρονική διάρκεια ίση με τη συνολική διάρκεια και των 5. Όμως έχει το μειονέκτημα σε σχέση με τον κινητό μέσο όρο ότι η κάθε μέτρηση ολοκληρώνεται σε μεγάλα διαστήματα τα οποία δεν είναι αποδεκτά .

Στη μέτρηση των στροφών υπάρχουν πολλές μέθοδοι οι οποίοι μπορούν να ακολουθηθούν. Οι δύο βασικότερες μέθοδοι υπολογισμού είναι, η θεώρηση ενός σταθερού χρονικού διαστήματος στο οποίο καταγράφονται τα περάσματα του σφονδύλου της μηχανής από το καταγραφικό, αισθητήρας υπέρυθρων, και η μέση ταχύτητα προκύπτει ως ο λόγος του πλήθους των καταγραφών προς το χρονικό διάστημα μέτρησης .Δεύτερη είναι η μέθοδος κατά την οποία υπολογίζεται το χρονικό διάστημα ,«περίοδος» ,μεταξύ δύο περασμάτων του σφονδύλου της μηχανής και η μέση ταχύτητα προκύπτει ως ο λόγος του ενός περάσματος προς το διάστημα της περιστροφής που είχε μετρηθεί προηγουμένως.

Στην πρώτη η τυπική απόκλιση ή σφάλμα σύμφωνα με τη θεωρία σφαλμάτων στο βιβλίο Εργαστηριακές ασκήσεις Φυσική Κεφ3.6 έκδοση 1999 δίνεται από τη σχέση :

$$\delta V = \frac{N}{T} \cdot \sqrt{\left(\frac{\delta N}{N}\right)^2 + \left(\frac{\delta T}{T}\right)^2} \xrightarrow{\frac{\delta N=1}{\delta T=0}} \delta \bar{V} = \frac{N}{T} \cdot \frac{\delta N}{N} = \frac{\delta N}{T} \quad (4.1)$$

,Θεωρείται ότι το σφάλμα μέτρησης της περιόδου T είναι μηδενικό. Αν το σφάλμα στην καταγραφή είναι «1» μια περιστροφή και η περίοδος μέτρησης 500 msec τότε το σφάλμα της μέτρησης είναι:

$$\delta\bar{V} = \frac{\delta N}{T} \xrightarrow[T=0.5Sec]{\delta N=1} \delta\bar{V} = \frac{1}{0.5Sec} = 2Rps \text{ _ or _ } 120RPM$$

Σύμφωνα με τη δεύτερη μέθοδο το σφάλμα ,υπό τη συνθήκη ότι έχουμε μηδενικό σφάλμα της καταγραφής, δίνεται από τη σχέση :

$$\delta V = \frac{N}{T} \cdot \sqrt{\left(\frac{\delta N}{N}\right)^2 + \left(\frac{\delta T}{T}\right)^2} \xrightarrow[\delta T=t]{\delta N=0 \& N=1} \delta V = 1 \cdot \frac{t}{T^2} \quad (4.2)$$

Έχει υπολογιστεί ότι το σφάλμα μέτρησης του χρόνου είναι : $t = 921600^{-1} Sec$ και για $T = 0,003Sec$ ίση με την περίοδο μία περιστροφής με ταχύτητα 20000RPM.

$$\delta V = N \cdot \frac{t}{T^2} = 1 \cdot \frac{921600^{-1}}{0.003^2} = 0.1205Rps \text{ _ or _ } 7.233RPM$$

Η προκύπτουσα τιμή του σφάλματος είναι πολύ μικρότερη της πρώτης μεθόδου. Όμως στη μέθοδο αυτή η επίδραση του θορύβου θα είναι σημαντική εφόσον υπολογίζεται με τη χρήση μόνο δύο καταγραφών του παλμού. Παρατηρώντας την επίδραση των δύο παραπάνω μεθόδων στο τελικό σφάλμα επιλέγεται να εφαρμοστεί μια σύνθετη μέθοδος από τις δύο παραπάνω .Η κάρτα έχει τη δυνατότητα να μετρά με μεγάλη ακρίβεια τη χρονική στιγμή στην οποία η κατάσταση του σήματος αλλάζει Timer1 capture π.χ, από On σε Off κλπ. Η σύνθετη μέθοδος μέτρησης στροφών δομείται ως εξής : Από την πρώτη μέθοδο κληρονομείται το σταθερό χρονικό διάστημα στο οποίο διαρκεί μία μέτρηση και η αθροιστική καταγραφή των περασμάτων στο διάστημα αυτό, ενώ από τη δεύτερη διατηρείται η καταγραφή της χρονικής στιγμής του περάσματος της οπής από τον υπέρυθρο φωτοανιχνευτή, αλλαγής της κατάστασης του Pin off->on .

Η ιδέα είναι ότι αν γνωρίζω στο χρονικό διάστημα ολοκλήρωσης της μέτρησης τη χρονική στιγμή του περάσματος της πρώτης και τελευταίας καταγραφής τότε η μέση ταχύτητα προκύπτει ως ο λόγος του αθροίσματος των καταγραφών μείον μία περιστροφή προς το χρονικό διάστημα μεταξύ πρώτης και τελευταίας καταγραφής και όχι προς το χρονικό διάστημα μέτρησης. Δηλαδή ο λόγος αυτός ισούται με τη μέση ταχύτητα του κινητήρα μεταξύ της πρώτης και της τελευταίας πλήρους περιστροφής κατά τη διάρκεια μίας μέτρησης, Έχει το μειονέκτημα ότι χρειάζονται τουλάχιστο δύο περιστροφές για τον υπολογισμό της ταχύτητας αλλά η άθροιση των περιστροφών, έχει ως συνέπεια τη μείωση του σφάλματος στην τελική μέτρηση .Επιπλέον το διάστημα ολοκλήρωσης ή άθροισης των περιστροφών δεν ισούται με το διάστημα μέτρησης αλλά με το διάστημα μεταξύ πρώτης και τελευταίας καταγραφής.

Το σφάλμα θα δίνεται ομοίως από τη σχέση στην οποία όμως η τιμή του χρόνου δεν είναι σταθερή αλλά εξαρτάται από τη μέση ταχύτητα μεταξύ της πρώτης και της τελευταίας χρονικά περιστροφής. Η ποιοτική διαφορά ανάμεσα στις δύο τελευταίες μεθόδους είναι ότι ο T δεν αντιπροσωπεύει την περίοδο μιας περιστροφής αλλά τη διάρκεια μεταξύ παλαιότερης και νεότερης καταγραφής.

$$\left. \begin{aligned} \delta V &= \frac{N}{T} \cdot \sqrt{\left(\frac{\delta N}{N}\right)^2 + \left(\frac{\delta T}{T}\right)^2} \xrightarrow[\delta T=t]{\delta N=0} \delta V = N \cdot \frac{t}{T^2} \\ \bar{V} &= \frac{N}{T} \Leftrightarrow T = \frac{N}{\bar{V}} \end{aligned} \right\} \Rightarrow$$

$$\Rightarrow \delta V = \bar{V} \frac{t}{T} \xrightarrow[T=\frac{N}{\bar{V}}]{} \delta V = \bar{V}^2 \frac{t}{N} \quad (4.3)$$

Επομένως για την ίδια μέση μετρούμενη ταχύτητα όσο αυξάνει ο αριθμός των καταγραφών δηλαδή το διάστημα μέτρησης τόσο μειώνεται το σφάλμα. Έστω ο κινητήρας περιστρέφεται με 20000RPM ή 333,3 rps και το διάστημα μέτρησης είναι 0,5 sec τότε το σφάλμα είναι

$$\delta \bar{V} = \bar{V}^2 \frac{t}{N} \underset{N \approx 333.3 \cdot 0.5}{=} = 333.3^2 \cdot \frac{921600^{-1}}{156} = 0.000772rps \text{ _or_ } 0.046RPM$$

❖ *Αλγόριθμος υπολογισμού Περιστροφικής ταχύτητας.*

Για την υλοποίηση της σύνθετης μεθόδου μέτρησης της ταχύτητας περιστροφής του κινητήρα χρησιμοποιούνται οι παρακάτω μεταβλητές και τύποι δεδομένων.

1. Μετρητής των περιστροφών από το καταγραφικό Counter.
2. Ο χρονομετρητής Timer1 στον οποίο είναι ενεργοποιημένη η λειτουργία Timer1Capture. Χρησιμοποιείται η λειτουργία καταγραφής του χρόνου όταν ο εφαρμοζόμενος παλμός μεταβαίνει από την κατάσταση off σε on.
3. Πίνακας αποθήκευσης στη μνήμη των δεδομένων ,για υπολογισμό της ταχύτητας στον ο οποίος στην πρώτη στήλη αποθηκεύεται το πλήθος των μετρήσεων και στην δεύτερη η τιμή χρόνου στην οποία έγινε καταγραφή της τελευταίας περιστροφής πριν την αποθήκευση.
4. Χρονικός καταμετρητής της διάρκειας μέτρησης Timer0 για το χρονισμό αποθήκευσης των μετρήσεων και υπολογισμού της ταχύτητας.

Για την πραγματοποίηση της μέτρησης πρέπει να έχουν εκκινήσει και οι δύο χρονομετρητές Timer0 & Timer1. Εκκινείτε ο Timer1 με τιμή εκκίνησης τη μηδενική (0). Όταν ολοκληρώνεται ή εκκινείτε η μέτρηση των στροφών, μηδενίζεται η τιμή του μετρητή περιστροφών ,Counter. Υπενθυμίζεται ότι η μέτρηση των στροφών πραγματοποιείται κατά τη δημιουργία του πακέτου δεδομένων αποστολής στον υπολογιστή. Επομένως η μέτρηση των στροφών θα έχει την ίδια περίοδο εκτέλεσης με την περίοδο επικοινωνίας με τον υπολογιστή η οποία είναι 20 msec και η οποία ελέγχεται από τον Timer0.

Κάθε φορά που καταγράφεται ένα πέρασμα , αυξάνεται η τιμή του counter κατά 1 και ταυτόχρονα αποθηκεύεται η τιμή της χρονικής στιγμής της καταγραφής στον καταχωρητή, Timer1capture, του Timer1. Όταν η χρονική διάρκεια μέτρησης ολοκληρωθεί, δηλαδή ο Timer0 έχει ενεργοποιήσει τη διακοπή της επικοινωνίας με τον υπολογιστή, εκκινείτε η εκτέλεση του αλγόριθμου μέτρησης. Καταρχήν αποθηκεύονται στον πίνακα αποθήκευσης των δεδομένων μέτρησης ,RPMArray, η τιμή του μετρητή περιστροφών και η τιμή της χρονικής στιγμής στην οποία

σημειώθηκε η τελευταία περιστροφή ,η οποία συγκρατείται στον καταχωρητή T1CR0.

Οι τιμές αυτές αποθηκεύονται στη θέση του πίνακα που περιέχει τις παλαιότερες τιμές όπως έχει εξηγηθεί παραπάνω. Με τον όρο θέση εννοείται η γραμμή του πίνακα αποθήκευσης, όπου στην ίδια γραμμή αποθηκεύονται στην πρώτη στήλη η τιμή του Counter, περιστροφές και στη δεύτερη η τιμή του καταχωρητή Timer1capture. Στη συνέχεια μηδενίζεται ο Counter = 0 ενώ ο Timer0 όταν ολοκληρώσει τη μέτρηση , θα επανεκκινήσει αυτόματα από τη θέση 0 ως το σταθερό διάστημα μέτρησης ,περίοδο μέτρησης.

Επειδή οι θέσεις του πίνακα είναι περιορισμένες δεν μπορούμε να χρησιμοποιούμε συνεχώς νέες θέσεις για την αποθήκευση των νέων τιμών ,και επομένως οι θέσεις του πίνακα ανακυκλώνονται. Επιπλέον αν ορίζονταν μία και μόνο θέση του πίνακα στην οποία θα αποθηκεύονταν οι νέες τιμές τότε θα αναγκαζόμασταν στη συνεχή μετακίνηση των δεδομένων του πίνακα ώστε να εξασφαλίσουμε ότι στην επιλεγμένη η θέση πριν την νέα εγγραφή θα υπάρχουν τα παλαιότερα δεδομένα. Για την αποφυγή των συνεχών μετακινήσεων δεδομένων επιλέγεται η κυκλική ανακύκλωση των θέσεων. Για κάθε νέα εγγραφή επιλέγεται η επόμενη θέση από την προηγούμενη εκτός από την περίπτωση όπου η προηγούμενη θέση ήταν ίση με τη μέγιστη θέση του πίνακα όπου και θέτεται ως νέα θέση η πρώτη. Με τον τρόπο αυτό χρησιμοποιούνται οι θέσεις του πίνακα με κυκλική επαναλαμβανόμενη σειρά. Η φορά συμπλήρωσης των θέσεων του πίνακα είναι από τη θέση με τη μικρότερη τιμή προς τη μεγαλύτερη με εξαίρεση τη μέγιστη στην οποία ακολουθεί μικρότερη, κλείνοντας τον κύκλο. Έστω δηλαδή ότι ο πίνακας διαθέτει 5 θέσεις ή γραμμές τότε η συμπλήρωση των θέσεων πραγματοποιείται με την εξής σειρά για ένα κύκλο , γραμμή $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 1$.

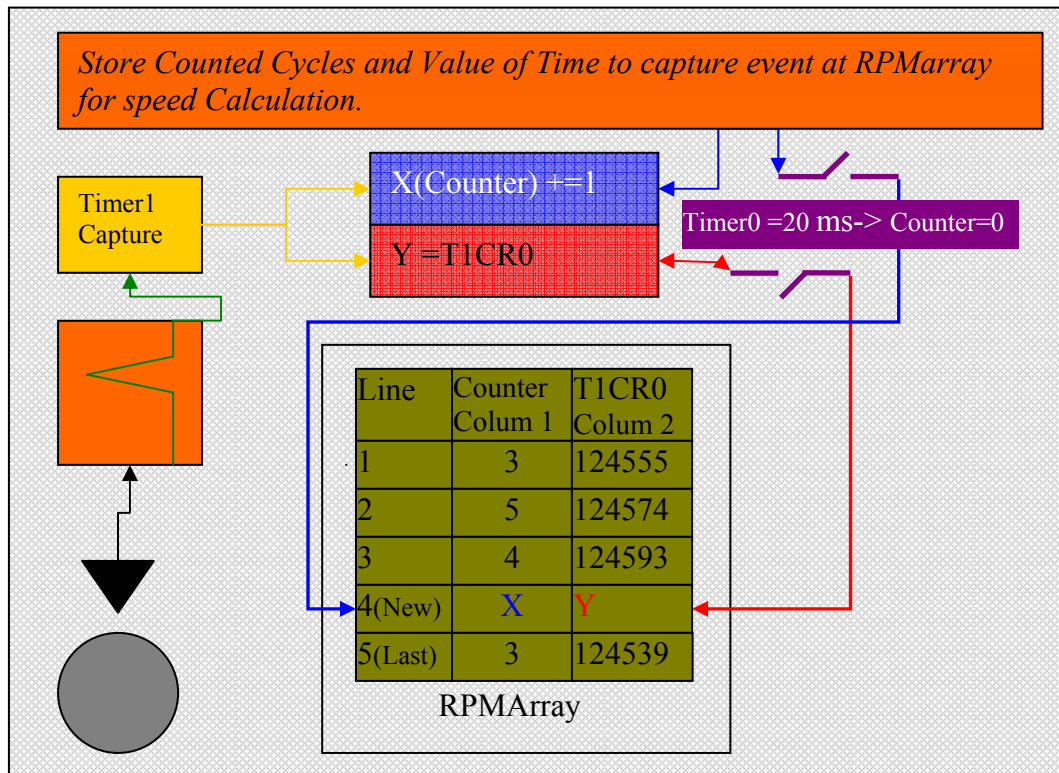
Επομένως αν γνωρίζω την τελευταία θέση αποθήκευσης μπορώ να υπολογίσω άμεσα τη θέση της παλαιότερης καταγραφής .Έστω ότι η νεότερη μέτρηση βρίσκεται στη γραμμή (v) του πίνακα τότε η παλαιότερη βρίσκεται στη γραμμή (v+1) σύμφωνα με την κυκλική διαχείριση των θέσεων. Αν ειδικά η νεότερη βρίσκεται στην τελευταία γραμμή τότε η παλαιότερη βρίσκεται στην πρώτη.

Ο υπολογισμός της μέσης ταχύτητας προκύπτει από την εκτέλεση της εξής σειράς πράξεων :

Γνωρίζοντας τη θέση της τελευταίας αποθήκευσης στον πίνακα αποθήκευσης RPMArray υπολογίζω πόσες περιστροφές «περάσματα» πραγματοποιήθηκαν μεταξύ της παλαιότερης και νεότερης μέτρησης ,αθροίζοντας όλες τις τιμές της στήλης 1 και αφαιρώ την παλαιότερη τιμή της. Επομένως ξέρω πόσες περιστροφές έχουν πραγματοποιηθεί από την παλαιότερη μέτρηση μέχρι την νεότερη .

Για τον υπολογισμό της ταχύτητας χρειάζομαι το χρονικό διάστημα που διάρκεσαν οι περιστροφές αυτές .Αυτό υπολογίζεται από τη στήλη 2 αποθήκευσης του χρόνου καταγραφής αφαιρώντας από τη νεότερη τιμή την παλαιότερη και επομένως γνωρίζω το χρονικό διάστημα μεταξύ των δύο. Άρα η περιστροφική ταχύτητα προκύπτει ως ο λόγος των περιστροφών που πραγματοποιήθηκαν δια της διάρκειας των περιστροφών και όχι με το συνολικό χρονικό διάστημα μέτρησης. Η ταχύτητα σε στροφές ανά λεπτό δίνεται από τη σχέση.

$$RPS = \frac{\sum_0^n Counter(i) - Counter(first)}{(T_{Last} - T_{First})(\cancel{Sec})} \cdot 60 \left(\frac{\cancel{Sec}}{Min} \right) \quad (4.4)$$



Σχήμα 4.5) Παράδειγμα αλγόριθμου αποθήκευσης των μετρήσεων στον πίνακα αποθήκευσης και ανακύκλωσης των θέσεων του για το υπολογισμό της ταχύτητας του κινητήρα.

Όταν ο Timer0 φτάσει την τιμή του διαστήματος μέτρησης πχ.20ms μηδενίζεται και εκκινεί τον υπολογισμό της ταχύτητας. Καταρχήν αποθηκεύει τις καταγεγραμμένες τιμές του X,Y στον πίνακα στη γραμμή έστω 4 ως η γραμμή που περιέχει τα παλαιότερα δεδομένα. Επομένως μετά την εγγραφή στον πίνακα ως παλαιότερη είναι η γραμμή 5. Η νέα αποθήκευση στον πίνακα την επόμενη φορά θα γίνει στη γραμμή 5. Άρα η ταχύτητα υπολογίζεται με τα παρακάτω βήματα.

1. Αθροίζω τη στήλη 1 Rev.Count =3+5+4+X+6 και στη συνέχεια αφαιρείται η παλαιότερη 3 .Ο λόγος της αφαίρεσης είναι να μην προκύψουν αρνητικές τιμές. Η νεότερη μέτρηση που βρίσκεται στη θέση (v) ,έστω v =4 τότε παλαιότερη βρίσκεται στη 4+1 = 5 γραμμή .Άρα Rev.Count =3+5+4+X+3 – 3 = 3+5+4+X=12+X.
2. Η διάρκεια των περιστροφών μεταξύ της νεότερης και παλαιότερης αποθήκευσης είναι ίση με τη διαφορά της νεώτερης καταγραφής μείον της παλαιότερης. $T_{Last} - T_{First} = Colum2(4) - Colum2(5) = Y - 1244539$

$$3. \quad RPS = \frac{\sum_0^n Counter(i) - Counter(first)}{(T_{Last} - T_{First})(\cancel{Sec})} \cdot 60 \left(\frac{\cancel{Sec}}{Min} \right) = \frac{12+X}{Y-1244539} \cdot 60$$

Time.Count=[{(στήλη 2 γραμμή4) = Y} – {(στήλη 2 γραμμή5) = 1221235}]
Αρα Rps = Rev.Count(σε Rev.) / Time.Coun(σε sec) Rps*60 = Rpm

5) Η γλώσσα προγραμματισμού V.B και η εφαρμογή ελέγχου επί του υπολογιστή.

5.0) ΣΥΝΟΨΗ.

Στο κεφάλαιο αυτό παρουσιάζονται τα δομικά στοιχεία του προγραμματισμού με τη χρήση προγραμματιστικών αντικειμένων και των πλεονεκτημάτων που προσφέρει έναντι της παλαιότερης μορφής χωρίς τη χρήση αντικειμένων «objects». Στη συνέχεια παρουσιάζεται η εκτέλεση, η μορφή και ροή εκτέλεσης των προγραμμάτων οδηγούμενα από γεγονότα. Η πορεία των προγραμμάτων αυτών δεν είναι προκαθορισμένη αλλά είναι οδηγούμενα από τα γεγονότα που οφείλουν να ανταποκριθούν. Η παρουσίαση έχει σκοπό την εξοικείωση του αναγνώστη με τη μορφή αυτή εκτέλεσης ενός προγράμματος

Στη συνέχεια αναφέρονται οι λόγοι επιλογής της γλώσσας γραφής του προγράμματος και παρουσιάζεται η γενική δομή του προγράμματος και των δυνατοτήτων της υλοποιημένης εφαρμογής. Λόγο του εκτεταμένου κώδικα που απαιτήθηκε για τη γραφή της δεν θα παρουσιαστεί όλος ο κώδικας αλλά των σημαντικότερων σημείων αυτού. Άλλωστε η περαιτέρω ερευνά επί του κώδικα μπορεί να γίνει και στα βοηθητικά κεφάλαια στα οποία παρουσιάζεται ολόκληρος. Τέλος δίνονται μερικά στιγμιότυπα από τη λειτουργία της εφαρμογής και εξηγούνται οι βασικές λειτουργίες των γραφικών επί της εφαρμογής.

5.1) Αντικειμενοστραφής Προγραμματισμός

Τα κλασσικά προγραμματιστικά περιβάλλοντα εργασίας, οδήγησαν στην ανάπτυξη πολύπλοκων προϊόντων λογισμικού, που η κατασκευή τους και η συντήρησή τους απαιτούσε ειδικούς με βαθιές γνώσεις σε επιμέρους θέματα, όπως τεχνικές διαχείρισης μνήμης, πρότυπα επικοινωνίας, ειδικές γλώσσες προγραμματισμού και λειτουργικά συστήματα. Η πολυπλοκότητα κατασκευής του λογισμικού δημιούργησε ταυτόχρονα προβληματισμούς και δρομολόγησε απόπειρες εξεύρεσης φιλικότερων τεχνικών και μεθόδων σχεδιασμού προγραμμάτων. Από την άλλη πλευρά η μοντελοποίηση του ανθρώπινου συλλογισμού μέσω της τεχνητής νοημοσύνης, έκανε φανερό την ανάγκη της δημιουργίας ενός διαφορετικού προγραμματιστικού περιβάλλοντος που θα ομαδοποιούσε στην ίδια οντότητα όλες τις πληροφορίες και ιδιότητες που αφορούσαν την ίδια έννοια.

Ο αντικειμενοστραφής σχεδιασμός προγραμμάτων γεννιέται από την προσπάθεια αντιμετώπισης αυτών ακριβώς των ζητημάτων. Χρησιμοποιώντας τον όρο αντικειμενοστραφής προγραμματισμός (object oriented programming) δεν αναφερόμαστε σε κάποιο συγκεκριμένο προϊόν ή μια γλώσσα προγραμματισμού, αλλά σε ένα διαφορετικό τρόπο προσέγγισης προβλημάτων με τον υπολογιστή.

Ο αντικειμενοστραφής προγραμματισμός ή αντικειμενοστραφής σχεδίαση προέκυψε από τη στιγμή που απαντήθηκε διαφορετικά από ότι απαντιόταν μέχρι το

ερώτημα «Η δομή των προγραμμάτων είναι προτιμότερο να στηρίζεται στις ενέργειες ή στα δεδομένα;». Η νέα απάντηση που δόθηκε, η απάντηση «δεδομένα», προσδιορίζει και τη βασική διαφορά ανάμεσα στις παραδοσιακές προγραμματιστικές τεχνικές και στην αντικειμενοστραφή προσέγγιση η οποία περιγράφει «ενέργειες» (επεξεργασία) που εφαρμόζονται πάνω σε δεδομένα.

Η αντικειμενοστραφής σχεδίαση λοιπόν εκλαμβάνει σαν πρωτεύοντα δομικά στοιχεία ενός προγράμματος τα δεδομένα, από τα οποία δημιουργούνται με κατάλληλη μορφοποίηση τα αντικείμενα. Αυτή η σχεδίαση αποδείχθηκε ότι επιφέρει καλύτερα αποτελέσματα, αφού τα προγράμματα που δημιουργούνται είναι περισσότερο ευέλικτα, επαναχρησιμοποιήσιμα και περισσότερο φιλικά.

Οι θεμελιώδεις αρχές του αντικειμενοστραφούς προγραμματισμού, πηγάζουν από τον καθημερινό μας φυσικό κόσμο. Στηρίζονται στο γεγονός, ότι για να μπορέσει κάποιος να κατανοήσει άγνωστες σε αυτόν έννοιες, θα πρέπει να καθοδηγηθεί μέσω της προσομοίωσης των άγνωστων αυτών εννοιών αντιστοιχίζοντας αυτές σε πρακτικές γνώσεις και εικόνες από το περιβάλλον του, τις οποίες γνωρίζει και μπορεί πολύ εύκολα να χειριστεί. Σύμφωνα με την αντικειμενοστραφή θεωρία ανάπτυξης εφαρμογών, η προσέγγιση κάθε προβλήματος πρέπει να γίνεται με φυσική ερμηνεία και να μην στηρίζεται σε πολύπλοκα τεχνικά ζητήματα.

Σύγχρονα αντικειμενοστραφή προγραμματιστικά περιβάλλοντα είναι η C++, η Java, η Visual Basic, το Delphi και πολλά άλλα.

5.2) Αντικείμενα

Βασικά στοιχεία του αντικειμενοστραφούς προγραμματισμού αποτελούν τα αντικείμενα (objects). Αν ρίξουμε μια ματιά γύρω μας, θα παρατηρήσουμε ότι το περιβάλλον μας αποτελείται από αντικείμενα, τα οποία μπορούμε εύκολα να αντιληφθούμε και να χειριστούμε. Κάθε αντικείμενο είναι ευδιάκριτο και άταρκες, ενώ το σύνολο των χαρακτηριστικών που προσδιορίζουν με λεπτομέρεια τη φυσική του υπόσταση.

5.3) Κλάσεις

Ένα αντικείμενο είναι άταρκες, έχει δηλαδή τη δική του φυσική υπόσταση και ταυτότητα. Αυτό πρακτικά σημαίνει, ότι ακόμη και αν δύο αντικείμενα έχουν ακριβώς τις ίδιες τιμές στις ιδιότητες τους εξακολουθούν να παραμένουν δύο ανεξάρτητα αντικείμενα. Κάθε ανεξάρτητο αντικείμενο, αποτελεί ένα στιγμιότυπο «instance» του γενικού τύπου. Ο γενικός τύπος ενός αντικειμένου καλείται κλάση «class» και καθορίζει τις αρχικές ιδιότητες και τη συμπεριφορά κάθε αντικειμένου που προέρχεται από αυτή.

Σε ένα αντικειμενοστραφές προγραμματιστικό περιβάλλον η υποστήριξη κλάσεων αποτελεί κυρίαρχο στοιχείο. Η κλάση είναι στατική περιγραφή ενός συνόλου αντικειμένων. Όλα τα αντικείμενα δημιουργούνται ως ακριβή αντίγραφα της κλάσης τους.

5.4) Ιδιότητες.

Κατά τη δημιουργία τους τα διάφορα αυτά αντικείμενα έχουν ίδιες ιδιότητες και αντιδρούν με τον ίδιο τρόπο σε όποιο μήνυμα ανιχνευτεί από το περιβάλλον τους. Καθένα από τα αντικείμενα εικόνας κληρονομεί τις αρχικές ιδιότητες και τη συμπεριφορά του από την κλάση εικόνα. Ωστόσο ο προγραμματιστής έχει τη δυνατότητα, μετατρέποντας τις τιμές των ιδιοτήτων τους «attributes ή properties», να διαμορφώσει διαφορετικά την εξωτερική εμφάνιση των αντικειμένων. Μπορεί επίσης με τη συγγραφή των κατάλληλων εντολών κώδικα να αντιστοιχήσει διαφορετικές εργασίες σε καθένα από αυτά τα αντικείμενα εικόνας.

Σε ένα γραφικό αντικειμενοστραφές περιβάλλον ανάπτυξης εφαρμογών, οι ιδιότητες των αντικειμένων είναι δυνατό να πάρουν αρχικές τιμές κατά το σχεδιασμό, αλλά τις περισσότερες φορές έχουμε τη δυνατότητα να τις μετατρέψουμε και κατά την εκτέλεση της εφαρμογής.

Κατά το σχεδιασμό μιας εφαρμογής πολλά από τα σύγχρονα περιβάλλοντα ανάπτυξης εφαρμογών μας παρέχουν κατάλληλα εργαλεία. Όπως παράθυρο ιδιοτήτων (properties window), που εμφανίζουν τη τιμή κάθε ιδιότητας ενός αντικειμένου και μας επιτρέπουν τη μετατροπή τους.

Όταν θέλουμε να μετατρέψουμε τη τιμή μιας ιδιότητας κατά την εκτέλεση μιας εφαρμογής, επειδή δεν έχουμε πρόσβαση στα εργαλεία που μας παρέχει το περιβάλλον ανάπτυξης, πρέπει να επέμβουμε δυναμικά με τη χρήση εντολών κώδικα. Στα περισσότερα αντικειμενοστραφή προγραμματιστικά περιβάλλοντα, όπως είναι η Visual C++ και η Visual Basic, μια εντολή έχει τη μορφή:

$$\text{Αντικείμενο.Ιδιότητα} = \text{Τιμή}$$

Στην παραπάνω εντολή ο όρος Αντικείμενο αντιπροσωπεύεται από μια ιδιότητα η οποία χαρακτηρίζει μοναδικά το επιθυμητό αντικείμενο. Συνήθως αυτή η ιδιότητα είναι το όνομα του αντικειμένου.

5.5) Μέθοδοι

Σε πολλά αντικειμενοστραφή προγραμματιστικά περιβάλλοντα, όπως είναι η Visual C++, Smalltalk και η Visual Basic, ένα αντικείμενο είναι δυνατόν εκτός από ιδιότητες να περιέχει και μεθόδους «methods». Μια μέθοδος είναι η εφαρμογή μιας ενέργειας επάνω σε ένα αντικείμενο. Οι μέθοδοι είναι εσωτερικά στοιχεία του αντικειμένου όπως και οι ιδιότητες.

Όλα τα αντικείμενα μιας κλάσης υποστηρίζουν τις ίδιες μεθόδους. Το αποτέλεσμα μιας μεθόδου επάνω σε ένα αντικείμενο συνήθως επηρεάζει κάποιες από τις τιμές των ιδιοτήτων του. Οι μέθοδοι μπορούν να ενεργήσουν σε ένα αντικείμενο μόνο κατά την εκτέλεση της εφαρμογής, άρα μέσα από εντολές κώδικα. Η σύνταξη μιας εντολής εκτέλεσης μεθόδου είναι:

$$\text{Αντικείμενο.Μέθοδος}$$

5.6) Οδηγούμενος από Γεγονότα Προγραμματισμός

Ο οδηγούμενος από γεγονότα προγραμματισμός «event-driven programming» είναι μια μεθοδολογία προγραμματισμού που μας επιτρέπει, σε συνδυασμό με τον αντικειμενοστραφή προγραμματισμό, να εκμεταλλευτούμε τα πλεονεκτήματα των σύγχρονων περιβαλλόντων εργασίας.

Για να μπορέσουν να κατανοηθούν καλύτερα τα πλεονεκτήματα του οδηγούμενου από γεγονότα προγραμματισμού, θα πρέπει να πραγματοποιηθεί πρώτα μια σύγκρισή του με το δομημένο προγραμματισμό, που αποτελεί την παραδοσιακή μορφή προγραμματισμού.

Σε μια εφαρμογή που έχει αναπτυχθεί με τη φιλοσοφία του δομημένου προγραμματισμού, η εκτέλεση της ξεκινά από την αρχική εντολή του προγράμματος και η ροή εκτέλεσής της είναι καθορισμένη από τις διαδικασίες και τις συναρτήσεις που περιλαμβάνει το πρόγραμμα. Σύμφωνα με τα παραπάνω το δομικό στοιχείο του προγράμματος αποτελούν οι διαδικασίες και οι συναρτήσεις. Σε όλη τη διάρκεια της εκτέλεσης της εφαρμογής, το πρόγραμμα διατηρεί τον έλεγχό της, ενώ στο χρήστη έχει ανατεθεί δευτερεύων ρόλος και απλά πληκτρολογεί κάποια δεδομένα, όταν το πρόγραμμα το απαιτεί.

Αντίθετα με τον οδηγούμενο από γεγονότα προγραμματισμό, ο χρήστης με την έναρξη της εκτέλεσης της εφαρμογής αποκτά τον έλεγχό της και αποφασίζει, χρησιμοποιώντας το πληκτρολόγιο ή το ποντίκι, ποιο τμήμα του προγράμματος θα εκτελεστεί. Κάθε ενέργεια του χρήστη δημιουργεί μηνύματα με τη μορφή των γεγονότων, τα οποία αντιλαμβάνεται το πρόγραμμα και ανταποκρίνεται σε αυτά.

Γεγονός «event» είναι μια ενέργεια που αναγνωρίζεται από την εφαρμογή και συγκεκριμένα από τα αντικείμενά της και την αναγκάζει να ανταποκριθεί. Η πρόκληση ενός γεγονότος μπορεί να γίνει είτε από το χρήστη με τη χρήση του πληκτρολογίου ή του ποντικιού είτε από το σύστημα.

Παραδείγματα γεγονότων που προκαλούνται από το χρήστη είναι το πάτημα κάποιου πλήκτρου εντολής, η επιλογή ενός στοιχείου μίας λίστας, η πληκτρολόγηση κάποιων χαρακτήρων σε ένα πλαίσιο κειμένου, η επιλογή ενός αντικειμένου μενού επιλογών και πολλά άλλα. Ως γεγονότα που προκαλούνται από το σύστημα μπορούμε να αναφέρουμε την αποστολή μηνυμάτων από το ρολόι του υπολογιστή ή από άλλες εφαρμογές και ο τερματισμός του συστήματος.

5.7) Διαδικασίες

Σε πολλά από τα σύγχρονα προγραμματιστικά περιβάλλοντα, ο τρόπος που ο προγραμματιστής καθορίζει την εκτέλεση του προγράμματος, είναι με συγγραφή των κατάλληλων εντολών κώδικα που ενεργοποιούν τα αντικείμενα της εφαρμογής και τα υποχρεώνουν να αντιδράσουν στα γεγονότα που προκαλούνται στο περιβάλλον τους. Ο κώδικας για κάθε γεγονός που πρόκειται να συμβεί, σε άλλα προγραμματιστικά περιβάλλοντα γράφεται με τη μορφή ανεξάρτητων τμημάτων κώδικα του προγράμματος, που καλούνται διαδικασίες γεγονότων «event procedures», ενώ σε κάποια άλλα προγραμματιστικά περιβάλλοντα γράφεται ενιαία.

Κάθε αντικείμενο της εφαρμογής αναγνωρίζει ένα συγκεκριμένο αριθμό γεγονότων. Κατά συνέπεια, για εκείνη την κατηγορία των προγραμματιστικών περιβαλλόντων που υποστηρίζουν διαδικασίες γεγονότων, αυτές καθορίζονται από την ενέργεια και το αντικείμενο που τις υποστηρίζει.

Η σύνταξη μιας διαδικασίας γεγονότος είναι:

Διαδικασία Αντικείμενο_Γεγονός

....

Εντολές κώδικα

....

Τέλος Διαδικασίας

Το μεγάλο πλεονέκτημα των σύγχρονων περιβαλλόντων προγραμματισμού, είναι ότι κάθε αντικείμενο αναγνωρίζει αυτόματα τα γεγονότα που το αφορούν και ξέρει πως θα αντιδράσει σε αυτά. Ο προγραμματιστής σε κανένα σημείο του προγράμματος δεν χρειάζεται να ασχοληθεί με τον τρόπο που το αντικείμενο θα ανιχνεύσει από το περιβάλλον το γεγονός, ούτε με την κλάση του αντίστοιχου κατάλληλου κώδικα, η οποία θα γίνει αυτόματα από το αντικείμενο.

Σε μια εφαρμογή οδηγούμενη από τα γεγονότα δεν συμπεριλαμβάνουμε μόνο διαδικασίες γεγονότων, αλλά έχουμε τη δυνατότητα να συμπεριλάβουμε και γενικές διαδικασίες ή συναρτήσεις που καλούνται μέσα από διαδικασίες γεγονότων.

Για παράδειγμα αν θέλουμε σε πολλά σημεία ενός προγράμματος να γίνεται ο ίδιος υπολογισμός, αποφεύγουμε να χρησιμοποιούμε τις ίδιες εντολές προγράμματος σε κάθε διαδικασία γεγονότος, αλλά δημιουργούμε μια γενική διαδικασία «Υπολογισμός» εφαρμόζοντας τις αρχές του δομημένου προγραμματισμού.

Παρατηρούμε λοιπόν ότι για την ανάπτυξη μιας σύγχρονης εφαρμογής πρέπει να εφαρμόσουμε και τις τρεις τεχνικές προγραμματισμού. Σε μια σύγχρονη εφαρμογή συνδυάζουμε τον τμηματικό «modular» προγραμματισμό, αξιοποιώντας τις τεχνικές του αντικειμενοστραφούς και του οδηγούμενου από γεγονότα προγραμματισμού, αλλά εξακολουθούμε να χρησιμοποιούμε και το δομημένο «structural» προγραμματισμό, αφού τα περιεχόμενα των διαδικασιών γεγονότων είναι δομημένες εντολές κώδικα. Η εφαρμογή του δομημένου προγραμματισμού φυσικά δε διαφοροποιείται και ισχύουν όσα γνωρίζουμε για τις μεταβλητές, τις δομές ελέγχου, τους πίνακες και πολλά άλλα. Τέλος μέσα σε ένα πρόγραμμα έχουμε τη δυνατότητα χρησιμοποίησης γενικών διαδικασιών και συναρτήσεων που επίσης στηρίζονται στις αρχές του δομημένου προγραμματισμού.

5.8) Ροή εκτέλεσης εφαρμογής

Αν προσπαθήσουμε να αναλύσουμε τη ροή εκτέλεσης μιας εφαρμογής που ακολουθεί την τεχνική του οδηγούμενου από γεγονότα προγραμματισμού και υποστηρίζει διαδικασίες γεγονότων, μπορούμε να τη παρουσιάσουμε με τα παρακάτω βήματα:

1. Ο χρήστης ξεκινά την εκτέλεση της εφαρμογής.
2. Τα αντικείμενα της εφαρμογής περνούν σε κατάσταση αναμονής γεγονότων.
3. Μόλις ένα αντικείμενο αναγνωρίσει μέσα στο περιβάλλον του κάποιου γεγονότος που το αφορά, καλεί την αντίστοιχη διαδικασία γεγονότος.
4. Αν στη διαδικασία γεγονότος έχουμε συμπεριλάβει εντολές κώδικα, εκτελούνται διαφορετικά η εφαρμογή αγνοεί το γεγονός.
5. Τα αντικείμενα της εφαρμογής περνούν πάλι σε κατάσταση αναμονής γεγονότων (Βήμα 2).

Ο προγραμματιστής δεν είναι απαραίτητο να συμπεριλάβει εντολές κώδικα σε κάθε διαδικασία γεγονότος που υποστηρίζουν τα αντικείμενα της εφαρμογής. Απλά επιλέγει αυτές για τις οποίες τον ενδιαφέρει το πρόγραμμα να αντιδρά. Αν μια διαδικασία γεγονότος δεν περιέχει καμία εντολή κώδικα όπως είναι φυσικό αυτό το γεγονός αγνοείται τελείως από την εφαρμογή, σαν να μην συνέβη ποτέ.

5.9) Υλοποίηση εφαρμογών σε σύγχρονο προγραμματιστικό περιβάλλον

Στο παρελθόν ένας προγραμματιστής κατά το σχεδιασμό της εφαρμογής έδινε ιδιαίτερη έμφαση στην ορθή υλοποίηση του αλγόριθμου του προγράμματος και δεν ασχολούνταν σχεδόν καθόλου με τον τρόπο επικοινωνίας του χρήστη με την εφαρμογή, αφού άλλωστε το περιβάλλον εργασίας δεν του παρείχε και ιδιαίτερες δυνατότητες επάνω σε αυτό το τμήμα του προγράμματος.

Αντιθέτως σε ένα σύγχρονο περιβάλλον εργασίας δίνεται ιδιαίτερη έμφαση στον τρόπο που επικοινωνεί ο χρήστης με το σύστημα. Κατά συνέπεια ο προγραμματιστής είναι υποχρεωμένος να ασχοληθεί ιδιαίτερα με τον τρόπο επικοινωνίας του χρήστη με την εφαρμογή, τη διεπαφή χρήστη «user interface». Κατά το σχεδιασμό της διεπαφής χρήστη, ο προγραμματιστής πρέπει να αντιλαμβάνεται την εφαρμογή από την πλευρά του χρήστη. Ένας προγραμματιστής για να δημιουργήσει μια σύγχρονη, φιλική και ευέλικτη εφαρμογή θα πρέπει να δώσει οπτική μορφή στο περιβάλλον εργασίας που παρέχει στο χρήστη. Τα σύγχρονα προγραμματιστικά περιβάλλοντα παρέχουν στον προγραμματιστή όλα εκείνα τα εργαλεία που χρειάζεται για να δημιουργήσει ένα τέτοιο περιβάλλον. Και φυσικά τα εργαλεία αυτά είναι αντικείμενα.

Στον κόσμο των αντικειμένων, ένας προγραμματιστής είναι φυσικό να ξεκινά το σχεδιασμό της εφαρμογής από τα αντικείμενα που πρόκειται να χρησιμοποιήσει. Η σωστή επιλογή των αντικειμένων εξασφαλίζει σε μεγάλο βαθμό και την επιτυχία εκτέλεσης της εφαρμογής.

Μετά από την επιλογή των κατάλληλων αντικειμένων, ο προγραμματιστής γνωρίζει ποιες θα είναι οι πιθανές ενέργειες του χρήστη ή του συστήματος επί της εφαρμογής. Επομένως είναι ευδιάκριτο ποια είναι τα γεγονότα που είναι δυνατό να ανιχνεύσουν τα επιλεγμένα αντικείμενα. Από τα πιθανά γεγονότα που θα συμβούν στο περιβάλλον της εφαρμογής, ο προγραμματιστής επιλέγει αυτά που θέλει να αξιοποιήσει μέσα από την εφαρμογή του και γράφει τον κατάλληλο κώδικα.

Συνοψίζοντας τα παραπάνω μπορούμε να περιγράψουμε την ανάπτυξη μιας εφαρμογής σε ένα σύγχρονο προγραμματιστικό περιβάλλον σαν διαδικασία τριών βημάτων:

1. Σχεδιασμός του τρόπου επικοινωνίας χρήστη-εφαρμογής, επιλέγοντας τα κατάλληλα εργαλεία.
2. Καθορισμός της αρχικής συμπεριφοράς των αντικειμένων μέσω των ιδιοτήτων που υποστηρίζουν.
3. Δημιουργία και εκσφαλμάτωση του κώδικα.

5.10) Στοιχεία γραφικού προγραμματιστικού περιβάλλοντος

Η φιλικότητα και η ευελιξία μιας εφαρμογής καθορίζεται κυρίως από τον τρόπο που επικοινωνεί με το χρήστη και συνεργάζεται με το περιβάλλον εργασίας. Το περιβάλλον εργασίας μιας εφαρμογής δεν πρέπει να είναι κουραστικό και δύσχρηστο. Αντίθετα θα πρέπει να είναι ευχάριστο, να καθοδηγεί και να διευκολύνει το χρήστη στην εργασία που θέλει να επιτελέσει.

Όλα τα σύγχρονα προγραμματιστικά εργαλεία, όπως η Visual C++, το Delphi ή η Visual Basic, μας προσφέρουν τη δυνατότητα να δίνουμε στις εφαρμογές μας τέτοια χαρακτηριστικά μέσα από τη χρησιμοποίηση των γραφικών. Με τον προγραμματισμό σε γραφικό προγραμματιστικό περιβάλλον, παρέχοντας ένα εύχρηστο περιβάλλον εργασίας στους χρήστες, αυξάνουμε τις δυνατότητες τους και μεγιστοποιούμε τα αποτελέσματα. Επιπλέον, μειώνεται δραστικά ο χρόνος και το κόστος εκμάθησης των χρηστών σε μια καινούργια εφαρμογή.

Ένα γραφικό προγραμματιστικό περιβάλλον μας δίνει ευελιξία και διαφορετικές δυνατότητες δημιουργίας γραφικού τρόπου επικοινωνίας του χρήστη με την εφαρμογή, αλλά ταυτόχρονα μας καθοδηγεί να δημιουργήσουμε γνώριμες εφαρμογές, αφού τα αντικείμενα, όπως πλήκτρα εντολής, λίστες, πλαίσια ελέγχου, πλήκτρα επιλογής, που χρησιμοποιούμε στις δικές μας εφαρμογές είναι ήδη γνωστά στους χρήστες από εφαρμογές γενικής χρήσης. Έτσι οι εφαρμογές μας, διατηρούν κάποια φιλοσοφία που είναι ήδη γνωστή στους χρήστες από άλλες εφαρμογές, όπως λογιστικά φύλλα, επεξεργαστές κειμένου ή βάσεις δεδομένων, που χρησιμοποιούν παρόμοια γραφικά αντικείμενα.

Ακόμη, ένα γραφικό προγραμματιστικό περιβάλλον προσφέρει στον προγραμματιστή τα κατάλληλα εργαλεία για γρήγορη ανάπτυξη εφαρμογών. Έτοιμα εργαλεία που παρέχονται από πολλά προγραμματιστικά περιβάλλοντα, όπως εργαλειοθήκες «toolboxes», εργαλεία εκσφαλμάτωσης «debugging tools», εύχρηστα παράθυρα συγγραφής κώδικα «code windows», γεννήτριες εκτυπωτικών αναφορών «report generators» συνθέτουν ένα περιβάλλον που διευκολύνει τον προγραμματιστή στο έργο του. Ακόμη διάφοροι οδηγοί εκτέλεσης εργασιών «wizards» τον βοηθούν

στον εκτέλεση συχνών εργασιών, όπως η δημιουργία φορμών ή εκτυπώσεων. Τέλος ένα σύγχρονο προγραμματιστικό εργαλείο, επιτρέπει στον προγραμματιστή να εκμεταλλευτεί τις ήδη υπάρχουσες εφαρμογές του συστήματος, χωρίς να χρειάζεται να δημιουργεί τμήματα κώδικα στην εφαρμογή του για την εκτέλεση εργασιών που ήδη εκτελεί σωστά κάποια άλλη εφαρμογή.

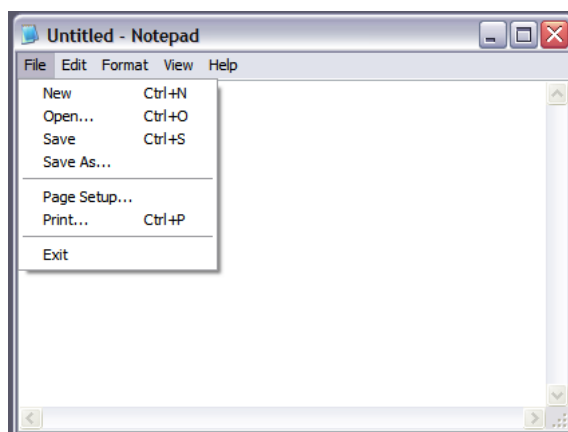
Το αποτέλεσμα εργασίας με ένα γραφικό προγραμματιστικό εργαλείο, είναι η γρήγορη υλοποίηση δυναμικών εφαρμογών που επικοινωνούν φιλικά με το χρήστη και συνεργάζονται αρμονικά με το περιβάλλον και τις υπόλοιπες εφαρμογές.

Βασικά χαρακτηριστικά στοιχεία των γραφικών προγραμματιστικών περιβαλλόντων είναι τα μενού επιλογών και τα πλαίσια διαλόγου.

5.10.1) Μενού επιλογών

Τα μενού επιλογών (menus) μας επιτρέπουν, κατά την εκτέλεση μιας εφαρμογής, τη γρήγορη πρόσβαση σε μια εύκολα προσπελάσιμη λίστα επιλογών. Οι περισσότεροι χρήστες γνωρίζουν ήδη τις δυνατότητες των μενού επιλογών αφού τα έχουν ήδη χρησιμοποιήσει μέσα από γνωστές εφαρμογές γενικής χρήσης.

Τα κλασικά μενού επιλογών εμφανίζονται στη γραμμή μενού «menu bar» με την ακόλουθη μορφή:



Σχήμα 5.1) Μενού επιλογών τυπικής εφαρμογής

Πολλά περιβάλλοντα παρέχουν τη δυνατότητα εμφάνισης μενού επιλογών με τη μορφή πτυσσόμενων μενού «popup menu». Τα πτυσσόμενα μενού επιλογών, συνήθως εμφανίζονται με το πάτημα του δεξιού πλήκτρου του ποντικιού και τα χρησιμοποιούμε για την εμφάνιση σύντομων επιλογών σε συγκεκριμένα σημεία της εφαρμογής.

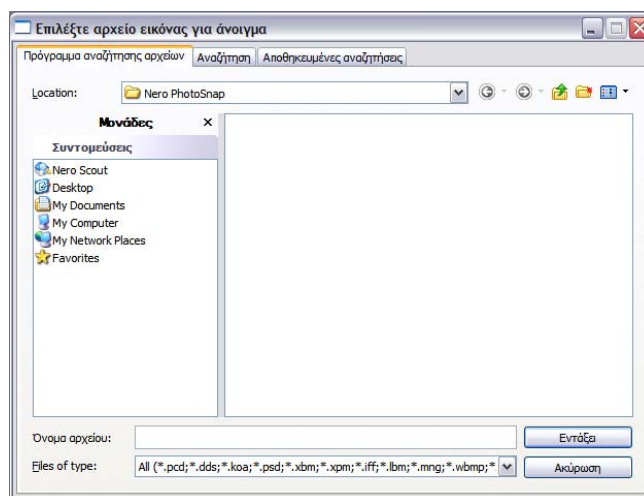
Στις εφαρμογές αν θέλουμε να δώσουμε ένα σύντομο τρόπο πρόσβασης στις πιο συχνά εκτελέσιμες επιλογές ενός μενού, ο καλύτερος τρόπος είναι η δημιουργία γραμμών εργαλείων «toolbars». Μια γραμμή εργαλείων μπορεί να υλοποιηθεί πολύ εύκολα με την ομαδοποίηση αντικειμένων εικόνας ή πλήκτρων εντολής που τα τοποθετούμε κάτω από τη γραμμή μενού και τον αντίστοιχο κώδικα.

5.10.2) Πλαίσια διαλόγου

Τα πλαίσια διαλόγου «dialog boxes» εξασφαλίζουν ένα άμεσο διάλογο της εφαρμογής με το χρήστη. Πολλές φορές κατά την εκτέλεση μιας εφαρμογής είναι αναγκαίο ο χρήστης να επικοινωνεί με την εφαρμογή. Μέσα από τα πλαίσια διαλόγου είναι δυνατό να εμφανιστούν μηνύματα προς το χρήστη, αλλά και ο χρήστης μπορεί να εισάγει κάποια δεδομένα στην εφαρμογή.

Τα περισσότερα σύγχρονα προγραμματιστικά εργαλεία παρέχουν στο προγραμματιστή έτοιμα προκαθορισμένα πλαίσια διαλόγου που μπορεί να χρησιμοποιήσει στις εφαρμογές του. Τα προκαθορισμένα πλαίσια διαλόγου, έχουν το πλεονέκτημα ότι διατηρούν την ίδια φιλοσοφία με το περιβάλλον εργασίας, αφού είναι πανομοιότυπα με τα πλαίσια που εμφανίζει το σύστημα όταν πρέπει να επικοινωνεί με το χρήστη, για παράδειγμα σε περίπτωση λάθους. Σε ένα προκαθορισμένο πλαίσιο διαλόγου έχουμε τη δυνατότητα να εμφανίσουμε διάφορα πλήκτρα, όπως πλήκτρο επικύρωσης «OK», ακύρωσης «Cancel» και επανάληψης «Retry». Η εμφάνιση των προκαθορισμένων πλαισίων διαλόγου γίνεται μέσω συναρτήσεων της γλώσσας προγραμματισμού.

Αν κανένα από τα προκαθορισμένα πλαίσια διαλόγου, που μας προσφέρει το περιβάλλον προγραμματισμού δεν καλύπτει κάποια ανάγκη της εφαρμογής μας ή δεν μας εκφράζει η τυποποίησή τους, έχουμε φυσικά τη δυνατότητα τοποθετώντας αντικείμενα επάνω σε μια φόρμα να δημιουργήσουμε τα δικά μας πλαίσια διαλόγου.



Σχήμα 5.2) Τυπικός πλαίσιο διάλογο εφαρμογής των windows

5.11) Επικοινωνία με άλλες εφαρμογές

Οι προγραμματιστές στον παραδοσιακό προγραμματισμό τις περισσότερες φορές αγνοούσαν υπάρχουσες εφαρμογές, γιατί το περιβάλλον εργασίας καθιστούσε δύσκολη τη συνεργασία τους. Έτσι κατέφευγαν στη συγγραφή τμημάτων κώδικα για την εκτέλεση εργασιών, που ήδη εκτελούσε ορθά μια άλλη εφαρμογή.

Όμως τα σύγχρονα περιβάλλοντα εργασίας παρέχουν τους μηχανισμούς για υψηλού επιπέδου επικοινωνία μεταξύ διαφορετικών εφαρμογών, του λειτουργικού

συστήματος και της εφαρμογής μας και μας επιτρέπουν έτσι την εκμετάλλευση έτοιμων βιβλιοθηκών προγραμμάτων, ελαττώνοντας και τις απαιτήσεις σε συγγραφή κώδικα.

Επιπλέον μέσα από την εφαρμογή μιας μπορούμε να έχουμε πρόσβαση και επικοινωνία με άλλου τύπου εφαρμογές για την ανάκτηση και ανάλυση δεδομένων διαφορετικής μορφής. Συγκεκριμένα, μέσα από μια εφαρμογή μας, μπορούμε να επεξεργαστούμε τα δεδομένα ενός λογιστικού φύλλου, να μορφοποιήσουμε και να εκτυπώσουμε τμήματα κειμένου που έχουν γραφτεί με έναν επεξεργαστή κειμένου, να χρησιμοποιήσουμε ένα πρόγραμμα επικοινωνίας για να στείλουμε ή να δεχτούμε μηνύματα, να αναλύσουμε τα δεδομένα μιας υπάρχουσας βάσης δεδομένων, να απεικονίσουμε γραφικά τα δεδομένα μας που προέρχονται από ένα πρόγραμμα παρουσιάσεων και πολλά άλλα.

Οι σύγχρονες τεχνολογίες σύνδεσης, μας επιτρέπουν μέσα από τη δική μας εφαρμογή την αξιοποίηση των δεδομένων μιας άλλης ανεξάρτητης εξωτερικής εφαρμογής. Υπάρχουν δύο τρόποι διασύνδεσης εφαρμογών, φαινομενικά ίδιοι από την πλευρά του χρήστη αλλά με τεχνικές διαφοροποιήσεις.

Με τον πρώτο τρόπο, έχουμε τη δυνατότητα διασύνδεσης «linking» των δεδομένων μιας άλλης εφαρμογής με τη δική μας εφαρμογή. Τα δεδομένα τη άλλης εφαρμογής που χρησιμοποιούμε στη δική μας, βρίσκονται σε κάποιο ξεχωριστό αρχείο του συστήματος. Με την τεχνική αυτή, η εφαρμογή μας καταφεύγει στο αρχείο και αντλεί τα δεδομένα, όποτε τα χρειαστεί, χωρίς να επιβαρύνεται σε χωρητικότητα. Τα δεδομένα αυτά της άλλης εφαρμογής παραμένουν προσπελάσιμα σε οποιαδήποτε τρίτη εφαρμογή μπορεί να τα χρειαστεί. Για παράδειγμα ένα αρχείο εικόνας είναι δυνατό να προσπελάζεται από την εφαρμογή δημιουργίας του φυσικά και ταυτόχρονα να είναι διασυνδεδεμένο τόσο στην εφαρμογή μας, όσο και σε κάποια τρίτη εφαρμογή. Κάθε χρονική στιγμή και στις τρεις εφαρμογές θα εμφανίζεται το πιο πρόσφατο ενημερωμένο αρχείο εικόνας.

Η τεχνική της διασύνδεσης, είναι ιδιαίτερα χρήσιμη για την υποστήριξη συνδέσεων πραγματικού χρόνου (real-time) μεταξύ εφαρμογών και για την κοινή προσπέλαση δεδομένων από διαφορετικές εφαρμογές με άμεση ενημέρωση των τροποποιήσεων του αντικειμένου. Μειονέκτημα αυτής της τεχνικής είναι ότι πιθανή μετακίνηση των δεδομένων από το σημείο του συστήματος στο οποίο βρίσκονται, έχει ως αποτέλεσμα την αδυναμία πρόσβασης της εφαρμογής μας στα δεδομένα αυτά, αν προηγουμένως δεν έχει γίνει ενημέρωση του συνδέσμου. Χαρακτηριστικό του τρόπου αυτού σύνδεσης είναι ότι η τροποποίηση των δεδομένων της εξωτερικής εφαρμογής μπορεί να γίνει μόνο μέσα από την εφαρμογή που τα δημιούργησε με ενεργοποίηση της είτε εξωτερικά είτε εσωτερικά από την εφαρμογή μας.

Ο δεύτερος τρόπος μας επιτρέπει την ενσωμάτωση (embedding) των δεδομένων μέσα στη δική μας εφαρμογή. Με την τεχνική αυτή εξασφαλίζεται ο αποκλειστικός έλεγχος των δεδομένων μέσα από την εφαρμογή μας. Τα δεδομένα δημιουργούνται μέσα από την εφαρμογή μας ή αντιγράφονται από κάποιο υπάρχον αρχείο και αποθηκεύονται στην εφαρμογή μας. Αυτό πρακτικά σημαίνει επιβάρυνση της εφαρμογής μας από την άποψη πως το αρχείο που δημιουργείται έχει μεγάλο μέγεθος, αλλά ταυτόχρονα εξασφαλίζεται η ασφάλεια των δεδομένων, αφού δεν επιτρέπεται να τροποποιηθούν από κάποια άλλη εφαρμογή.

Με την τεχνική της ενσωμάτωσης δύο εκδοχές είναι δυνατές. Τα δεδομένα να δημιουργηθούν εκείνη τη στιγμή ή να είναι ήδη έτοιμα. Στην πρώτη περίπτωση τα δεδομένα δημιουργούνται με τη χρήση μιας άλλης εφαρμογής που ενεργοποιείται μέσα από τη δική μας και ενσωματώνονται κατ' ευθείαν στην εφαρμογή μας. Για παράδειγμα με την ενεργοποίηση μέσα από την εφαρμογή μας και τη χρήση ενός προγράμματος επεξεργασίας εικόνας μπορούμε να δημιουργήσουμε ένα γραφικό που θα ενσωματωθεί στα δεδομένα της εφαρμογής μας. Σε αυτήν την περίπτωση δεν δημιουργείται κάποιο ξεχωριστό αρχείο στο σύστημα και επιπλέον δεν υπάρχει η δυνατότητα πρόσβασης από τη μεριά μιας τρίτης εφαρμογής σε αυτό το γραφικό.

Στη δεύτερη περίπτωση υπάρχει ανεξάρτητο αρχείο στο σύστημα, από το οποίο και προέρχονται τα δεδομένα που χρησιμοποιούμε στην εφαρμογή μας. Αυτό βέβαια δεν σημαίνει ότι το ανεξάρτητο αυτό αρχείο δεσμεύεται από την εφαρμογή μας. Ένα αντίγραφο του ενσωματώνεται στο αρχείο της εφαρμογής μας. Αν για παράδειγμα έχουμε διασυνδέσει την εφαρμογή μας με ένα λογιστικό φύλλο και κάποια στιγμή θέλουμε να επιφέρουμε κάποιες αλλαγές, ενεργοποιούμε μέσα από την εφαρμογή μας την εφαρμογή επεξεργασίας λογιστικών φύλλων και πραγματοποιούμε τις επιθυμητές αλλαγές. Αν στη συνέχεια “ανοίξουμε” το ανεξάρτητο αρχείο του λογιστικού φύλλου που υπάρχει στο σύστημα, θα διαπιστώσουμε ότι οι αλλαγές αυτές δεν έχουν καταγραφεί σε αυτό.

Ανακεφαλαιώνοντας, τα δεδομένα που συνδέονται με την εφαρμογή μας με τη τεχνική της διασύνδεσης αποθηκεύονται στην εφαρμογή η οποία τα δημιούργησε, ενώ αντίθετα τα δεδομένα που συνδέονται στην εφαρμογή μας με την τεχνική της ενσωμάτωσης, περιέχονται στην εφαρμογή μας και αποθηκεύονται μαζί της.

5.12) Χαρακτηριστικά αντικειμενοστραφών περιβαλλόντων ανάπτυξης εφαρμογών

Το βασικότερο χαρακτηριστικό ενός αντικειμενοστραφούς περιβάλλοντος ανάπτυξης εφαρμογών είναι η κλάση. Ένα πραγματικά αντικειμενοστραφές περιβάλλον ανάπτυξης εφαρμογών πρέπει να διαθέτει ακόμη τα παρακάτω χαρακτηριστικά:

- Υποστήριξη αφηρημένων τύπων (data abstraction). Κάθε αντικείμενο περιγράφεται ως υλοποίηση αφηρημένων τύπων. Κάθε αφηρημένος τύπος έχει ως δομικά του στοιχεία δεδομένα και συγκεκριμένους κανόνες συμπεριφοράς. Ένα αντικείμενο περιγράφεται από συγκεκριμένες ιδιότητες και κατά την εκτέλεση της εφαρμογής λαμβάνει μόνο τα μηνύματα που το αφορούν αγνοώντας οτιδήποτε άλλο συμβαίνει στο περιβάλλον του. Αυτό εξασφαλίζει ότι κάθε τμήμα του προγράμματος επηρεάζει μόνο τα αντικείμενα που το αφορούν και έχουμε πλήρη έλεγχο της εφαρμογής.
- Ενθυλάκωση (encapsulation). Σε μια αντικειμενοστραφή εφαρμογή κάθε αντικείμενο αποτελεί ξεχωριστή οντότητα και περιέχει ενσωματωμένα τα χαρακτηριστικά (δεδομένα) και τους κανόνες συμπεριφοράς του (μεθόδους). Η δυνατότητα ενός αντικειμένου να συνδυάζει εσωτερικά τα δεδομένα και τις μεθόδους χειρισμού του καλείται ενθυλάκωση. Την ενθυλάκωση μπορούμε να την παρομοιάσουμε σαν ένα κέλυφος που υπάρχει γύρω από κάθε αντικείμενο και διαχωρίζει τον εσωτερικό από τον

εξωτερικό του κόσμο. Με την ενθυλάκωση ο χειρισμός κάθε αντικειμένου είναι φυσικά πιο εύχρηστος και ασφαλής, γιατί τα περιεχόμενα του προστατεύονται και είναι δυνατό να μεταβληθούν μόνο με την αλλαγή των τιμών των ιδιοτήτων του ή με τη δράση των μεθόδων που υποστηρίζει.

- **Κληρονομικότητα (inheritance).** Χαρακτηριστικά και μέθοδοι μπορούν να είναι κοινά σε διαφορετικές κλάσεις που είναι ιεραρχικά συνδεδεμένες. Μια κλάση μπορεί να περιγραφεί γενικά και στη συνέχεια μέσω αυτής της κλάσης να οριστούν υποκλάσεις (subclasses) αντικειμένων. Η κλάση απόγονος (υποκλάση) κληρονομεί και μπορεί να χρησιμοποιήσει όλα δεδομένα και τις μεθόδους που περιέχει η κλάση πρόγονος. Για παράδειγμα αν περιγράψουμε μια γενική κλάση γεωμετρικό σχήμα, μπορούμε στη συνέχεια να δημιουργήσουμε από αυτή υποκλάσεις αντικειμένων τριγώνου, τετραγώνου κ.λπ. Η δυνατότητα δημιουργίας ιεραρχιών αντικειμένων καλείται κληρονομικότητα.
- **Πολυμορφισμός (polymorphism).** Με το χαρακτηριστικό του πολυμορφισμού παρέχεται η δυνατότητα στο ίδιο αντικείμενο να αναφέρεται, στο επίπεδο εκτέλεσης της εφαρμογής, σε διαφορετικές κλάσεις, να επιδρά διαφορετικά σε διαφορετικά αντικείμενα. Δύο ή περισσότερες κλάσεις αντικειμένων μπορούν να υποστηρίξουν συμπεριφορές με κοινό όνομα και ίδιο βασικό σκοπό αλλά με διαφορετική εφαρμογή. Για παράδειγμα σε ένα αυτοκίνητο τα πεντάλ του γκαζιού και του φρένου υποστηρίζουν τη μέθοδο πάτησε. Με τη μέθοδο πάτησε ο οδηγός δίνει μια εντολή για την εκτέλεση μιας εργασίας, αλλά η εφαρμογή της μεθόδου είναι διαφορετική σε κάθε πεντάλ. Όταν ο οδηγός πατήσει γκάζι, το αυτοκίνητο αναπτύζει ταχύτητα, ενώ το πάτημα του φρένου το επιβραδύνει. Η ίδια συμπεριφορά του οδηγού επιφέρει διαφορετικό αποτέλεσμα. Με τη χρήση του πολυμορφισμού ο προγραμματιστής απαλλάσσεται από τη σύνταξη και την επανάληψη πολύπλοκων δομών ελέγχου μέσα στην εφαρμογή.

5.13) Κριτήρια επιλογής της χρήσης για τη γραφή του προγράμματος της γλώσσας Visual Basic

Η φιλικότητα και η ευελιξία που προσφέρουν τα προγράμματα γραφικής επικοινωνίας με το χρήστη καθόρισαν τον αποκλεισμό των υπόλοιπων προγραμματιστικών περιβαλλόντων τα οποία ήταν διαθέσιμα για την ανάπτυξη της εφαρμογής λειτουργίας και επικοινωνίας μεταξύ της LPC2106 και του υπολογιστή με αποτέλεσμα να μπορεί να χρησιμοποιηθεί στο λειτουργικό περιβάλλον των Windows με βασικό χαρακτηριστικό τη γρήγορη και φιλική ανάπτυξη της εφαρμογής. Ως βασικό προγραμματιστικό περιβάλλον επιλέχθηκε η χρήση του Visual Studio 2005 το οποίο διαθέτει τη δυνατότητα ανάπτυξης εφαρμογών τόσο σε C++ ,C# και Visual Basic. Το Visual Studio 2005 είναι η συνέχεια του 2003 και φέρει σημαντικές αλλαγές καθώς και επιπλέον διαθέτει νέα εργαλεία και δυνατότητες. Πρόκειται για την τελευταία και τρέχουσα έκδοση προγραμματιστικού περιβάλλοντος της Microsoft για το λειτουργικό της περιβάλλον Windows υποστηρίζοντας το με πληθώρα άρθρον στη δικτυακή βιβλιοθήκη υποστήριξης. Το Visual Studio χρησιμοποιείται για την εξαγωγή εφαρμογών όλων των τύπων οι οποίες αφορούν εφαρμογές του λειτουργικού των «windows». Η τελική επιλογή της γλώσσας στην οποία αναπτύσσεται ο κώδικας της εφαρμογής είναι η Visual Basic. Στο «V.S» 2005 η χρησιμοποιούμενη έκδοση της Visual Basic είναι η 8.

5.14) Λίγα λόγια για την Ιστορία της Visual Basic.

Η «Basic» ήταν μια νεότερη γλώσσα προγραμματισμού και εξακολουθεί να είναι ακόμα μεταξύ των απλούστερων και δημοφιλέστερων γλωσσών προγραμματισμού. Αρχικά σχεδιασμένη ως διαλογική γλώσσα κατανομής χρόνου κεντρικών υπολογιστών από τους John Kemeney και Thomas Kurtz το 1963 προέκυψε ως ευρέως χρησιμοποιούμενη στους προσωπικούς υπολογιστές. Στον πρώτο "οικογενειακό" υπολογιστή της IBM, το PCjr, με μια κασέτα Basic. Λόγω της απλότητας της η «Basic» έχει χρησιμοποιηθεί συχνά στη διδασκαλία των εισαγωγικών εννοιών του προγραμματισμού.

Η Basic συνεχίζει να χρησιμοποιείται ευρέως επειδή μπορεί να μαθευτεί γρήγορα, οι δηλώσεις της είναι εύκολο να διαβαστούν από άλλους προγραμματιστές, και η υποστήριξη είναι της διαθέσιμη στα περισσότερα λειτουργικά συστήματα. Η τεκμηρίωση του της «Basic» έχει μεταφραστεί σε πολλές εθνικές γλώσσες. Έρχεται συχνά με την υποστήριξη ήχου και γραφικής παράστασης. Μια δημοφιλής έκδοση της «Basic» είναι σήμερα QBASIC και πλέον η Visual Basic.

Η «BASIC» χρησιμοποιείται σε πολλές επιχειρησιακές εφαρμογές και θεωρείται ακόμα έγκυρη επιλογή ως γλώσσα προγραμματισμού. Η Visual Basic προσθέτει στη «Basic» αντικειμενοστραφή χαρακτηριστικά γνωρίσματα και γραφική επικοινωνία με το χρήστη διευρύνοντας έτσι τη χρηστικότητα της ως γλώσσα προγραμματισμού.

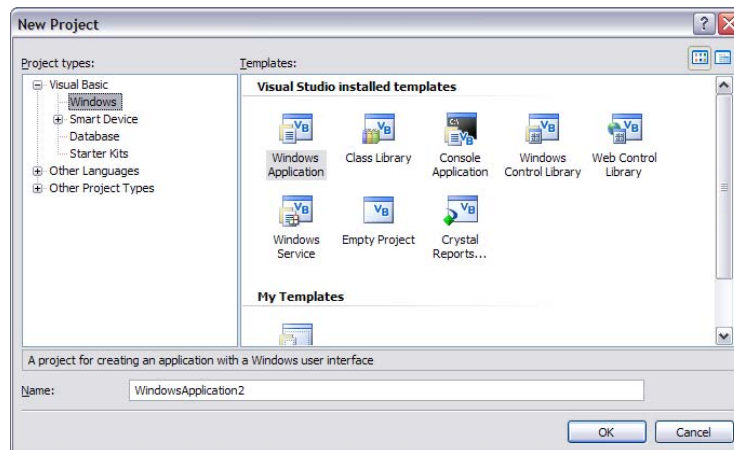
5.14.1) Λίγα λόγια για το προγραμματιστικό περιβάλλον του Visual Studio 2005

Ανοίγοντας το περιβάλλον εργασίας του Visual Studio 2005 εμφανίζεται η παρακάτω οθόνη.



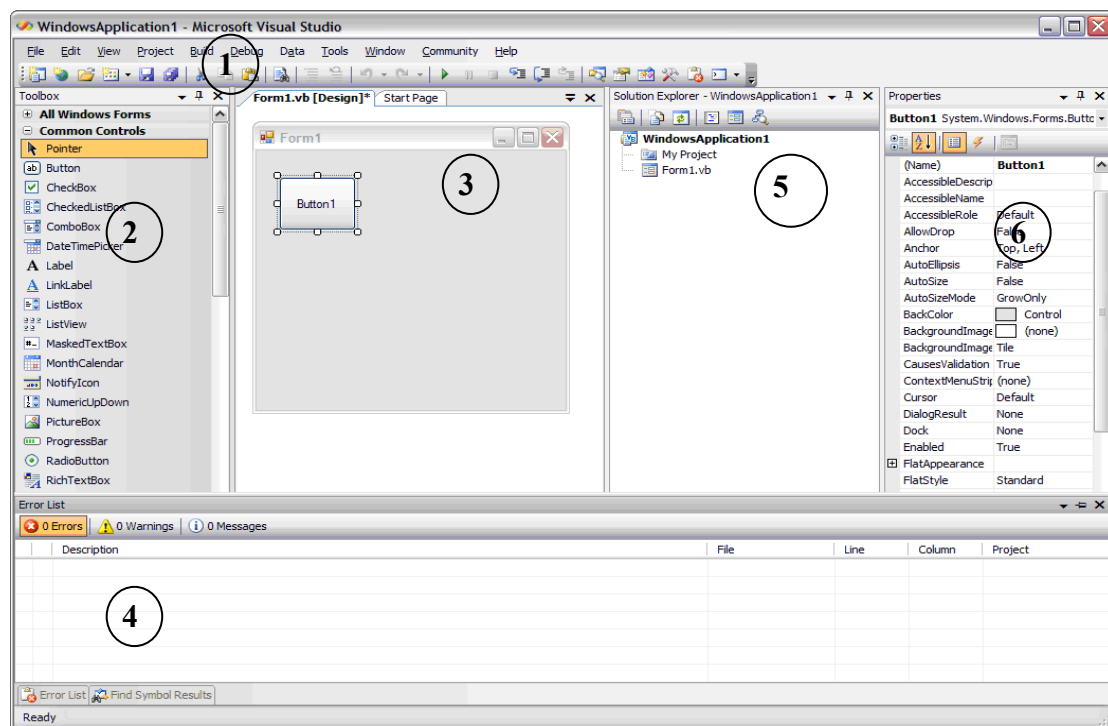
Σχήμα 5.3) Οθόνη έναρξης του Visual Studio 2005

Στην οθόνη δίνονται στην αριστερή επάνω στήλη τα τελευταία χρησιμοποιημένα projects. Για τη δημιουργία ενός νέου επιλέγω από το κουμπί File->New Project και στη συνέχεια εμφανίζεται η φόρμα οθόνη επιλογής του τύπου του νέου project που θέλουμε να δημιουργηθεί.



Σχήμα 5.4) Οθόνη δημιουργίας και καινούριας εργασίας «project»

Τη συνέχεια θα εμφανιστεί η τυπική οθόνη προγραμματισμού της Visual Basic .



Σχήμα 5.5) Οθόνη γραφής του κώδικα V.B

Ανοίγοντας τη VB, στην οθόνη εμφανίζεται το παραπάνω παράθυρο. Όπως φαίνεται το βασικό παράθυρο της γλώσσας (1) βρίσκεται στο πάνω μέρος της οθόνης. Περιέχει τη γραμμή των μενού, καθώς και μία εργαλειοθήκη. Αν και πολύ μικρό, εμπεριέχει όλα τα εργαλεία που είναι διαθέσιμα στον προγραμματιστή.

Στην αριστερή μεριά της οθόνης εμφανίζεται ένα μακρόστενο παράθυρο το οποίο ονομάζεται «toolbox» (2). Το παράθυρο αυτό το χρησιμοποιούμε για να προσθέσουμε τα απαραίτητα χειριστήρια στο πρόγραμμά μας.

Συνήθως στη μέση της οθόνης εμφανίζεται η φόρμα « form » (3), η οποία είναι το αντικείμενο στο οποίο θα προγραμματίσουμε και θα δημιουργήσουμε.

Στο κάτω μέρος της οθόνης εμφανίζεται το παράθυρο εκσφαλμάτωσης «immediate window ή debug window» (4). Στο παράθυρο αυτό μπορούμε να βρούμε τα λάθη που αναφέρονται κατά την εκσφαλμάτωση debugging στο πρόγραμμα.

Στη δεξιά μεριά της οθόνης στο πάνω μέρος βρίσκεται ο «project explorer»(5). Με το παράθυρο αυτό μπορούμε να βρίσκουμε εύκολα και να επεξεργαζόμαστε τα αντικείμενα που έχουμε προσθέσει στο πρόγραμμά μας όπως οι φόρμες.

Επιπλέον ένα ακόμα σημαντικό παράθυρο θα το βρούμε στην κάτω δεξιά μεριά της οθόνης που είναι το παράθυρο ιδιοτήτων των αντικειμένων «properties window» (6). Από το παράθυρο αυτό μπορούμε πολύ απλά και εύκολα να ρυθμίζουμε τα διάφορα αντικείμενα του προγράμματος, πριν καν ακόμα αρχίσουμε να προγραμματίζουμε.

Τέλος αν για κάποιο λόγο δεν εμφανίζεται στην οθόνη κάποιο από τα παράθυρα, θα εμφανιστεί χρησιμοποιώντας το μενού «View» ,από το βασικό παράθυρο της γλώσσας προγραμματισμού. Για παράδειγμα αν δεν εμφανίζεται το «toolbox» μπορούμε να ανοίξουμε το μενού «View» ,και να επιλέξουμε « toolbox ». Αυτομάτως το παράθυρο θα εμφανιστεί στην οθόνη.

Ουσιαστικά με κάθε νέα φόρμα δημιουργείται μία αυτοτελής κλάση στην οποία όλα τα τοποθετημένα εργαλεία της ανήκουν. Με την εντολή από το μενού «View» και πατώντας «Code» βλέπουμε τον κώδικα που ανήκει στην κλάση της φόρμας αυτής και όλων των αντικειμένων επί αυτής.

5.15) Βασικές Απαιτήσεις της εφαρμογής Visual Basic του υπολογιστή.

Οι λειτουργίες που απαιτείται να εκτελέσει η εφαρμογή στον υπολογιστή χωρίζονται στις παρακάτω κατηγορίες :

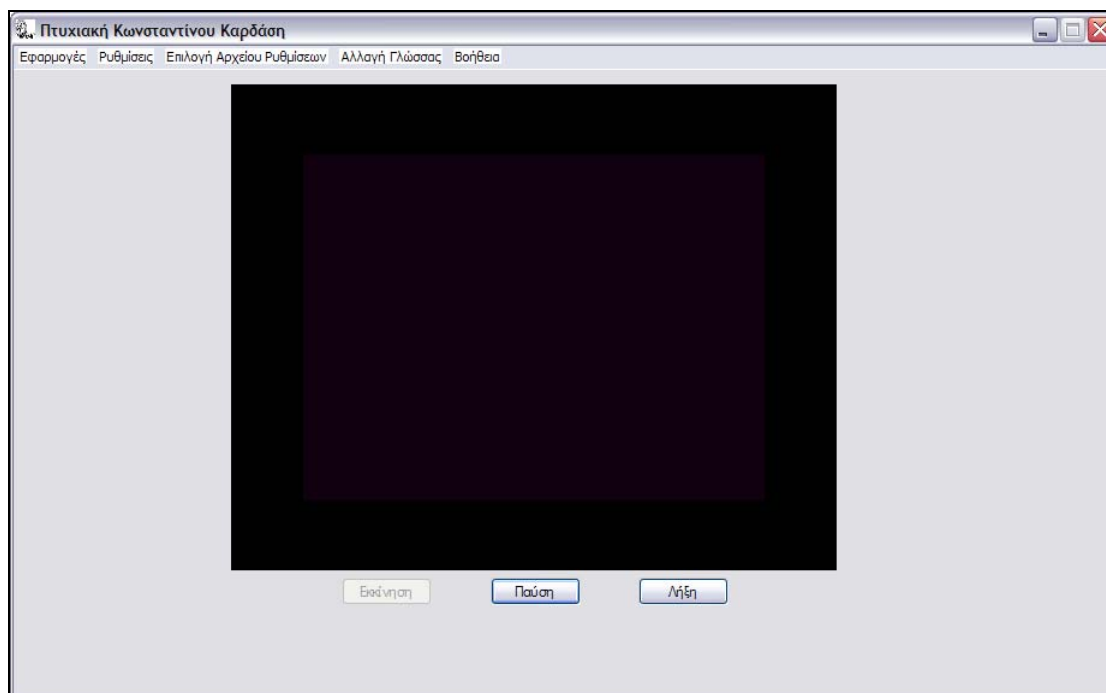
- ❖ Επικοινωνίας μέσω της σειριακής θύρας με την LPC2106.
- ❖ Παρακολούθηση της λειτουργίας της LPC2106 και εμφάνιση των εισερχόμενων μετρήσεων σε γραφική μορφή κειμένου και παραστάσεων.
- ❖ Αποθήκευση των εισερχόμενων δεδομένων σε αρχείο κειμένου.
- ❖ Έλεγχος της ορθής λειτουργίας της κάρτας μέσω της αποστολής των κατάλληλων εντολών διαμέσων της σειριακής θύρας.
- ❖ Εναλλαγή ανάμεσα στον έλεγχο της δράσης ελέγχου στον κινητήρα και του έλεγχου της περιστροφικής ταχύτητας του κινητήρα. Δηλαδή τον έλεγχο του εφαρμοζόμενου PWM στον κινητήρα ή τον έλεγχο του σημείου αναφοράς των στροφών του ελεγκτή επί της LPC2106.
- ❖ Γραφικά χειριστήρια των σημείων αναφοράς λειτουργίας της εγκατάστασης.
- ❖ Δυνατότητα αποστολής των σημείων αναφοράς μέσω της ανάγνωσης τους από αρχείο κειμένου,txt.
- ❖ Εφαρμογή δημιουργίας του αρχείου κειμένου ,txt στο οποίο θα περιέχονται τα σημεία αναφοράς προς αποστολή.

- ❖ Γραφική προεπισκόπηση του αρχείου προς αποστολή και τροποποίηση του.
- ❖ Εισαγωγή των τιμών αναφοράς στο αρχείο προς αποστολή από αρχείο κειμένου ,txt.
- ❖ Αποθήκευση των ρυθμίσεων λειτουργίας ολόκληρης της εφαρμογής.
- ❖ Φόρμα επιλογής των ρυθμίσεων λειτουργίας της εφαρμογής

Οι παραπάνω κατηγορίες λειτουργίας παρουσιάζονται στα επόμενα κεφάλαια, ενώ αναλυτικές πληροφορίες δίνονται στα βοηθητικά κεφάλαια στο εγχειρίδιο χρήσης της εφαρμογής.

5.16) Κεντρική Φόρμα Διαχείρισης της εφαρμογής

Για την εκτέλεση όλων των παραπάνω λειτουργιών δημιουργήθηκε η κεντρική φόρμα διαχείρισης όλης της εφαρμογής υπό το όνομα «Πτυχιακή Καρδάση Κωνσταντίνου». Με την εκκίνηση της εφαρμογής εμφανίζεται η παρακάτω κεντρική φόρμα.

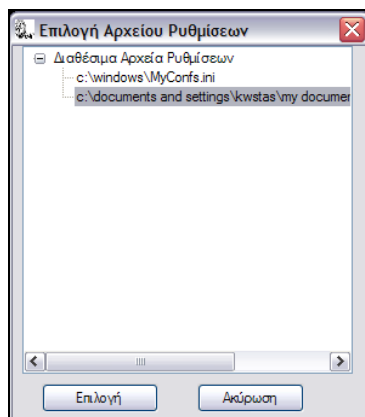


Σχήμα 5.6) Κεντρική φόρμα διαχείρισης της εφαρμογής του υπολογιστή.

Στη φόρμα αυτή εμφανίζεται στην κορυφή της οι κατάλογοι ή κοινός μενού από τους οποίους παρέχεται η πρόσβαση σε όλες τις φόρμες ρυθμίσεων και εκτέλεσης της επικοινωνίας και ελέγχου της LPC2106. Επιπλέον έχει προστεθεί η δυνατότητα εμφάνισης πλαισίου διαλόγου και ανάγνωσης του κειμένου βοήθειας ολόκληρης της εφαρμογής. Δίνεται η δυνατότητα δημιουργίας πολλαπλών αρχείων αποθήκευσης των ρυθμίσεων της εφαρμογής και χρήσή τους. Τέλος στο πρόγραμμα συμπεριλήφθηκε η γραφή όλων των λεκτικών και εμφανιζόμενων μηνυμάτων σε πολλαπλές γλώσσες οι οποίες είναι Ελληνικά, Αγγλικά, Ιταλικά, Ισπανικά και Γαλλικά. Η προσωρινή επιλογή της γλώσσας γίνεται από το μενού αλλαγή γλώσσας.

5.16.1) Επιλογή και χρήση του υπάρχοντος αρχείου ρυθμίσεων.

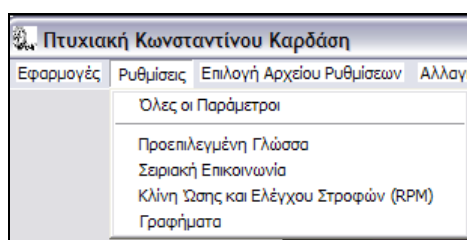
Στον κατάλογο Επιλογή Αρχείου Ρυθμίσεων της κεντρικής φόρμα διαχείρισης εμφανίζονται όλα τα διαθέσιμα αρχεία ρυθμίσεων ανάμεσα στα οποία μπορούμε να επιλέξουμε το επιθυμητό. Κάθε αρχείο περιέχει όλες τις απαραίτητες ρυθμίσεις της εφαρμογή οι οποίες γίνονται άμεσα ενεργές με την αποδοχή και έξοδο από τον κατάλογο αυτό.



Σχήμα 5.7) Φόρμα επιλογής του αρχείου ρυθμίσεων.

5.16.2) Δημιουργία νέου ή τροποποίηση αρχείου ρυθμίσεων.

Για την επίτευξη της τροποποίησης των παραμέτρων λειτουργίας της εφαρμογής ή της ρύθμισης νέου αρχείου ρυθμίσεων εισήχθη στην εφαρμογή η φόρμα ρυθμίσεων μέσα από την οποία δίνεται η πρόσβαση σε όλες τις διαθέσιμες ρυθμίσεις. Η φόρμα των ρυθμίσεων εμφανίζεται από τον κατάλογο «Ρυθμίσεις» του μενού της φόρμας διαχείρισης «Πτυχιακή Κωνσταντίνου Καρδάση». Επίσης στον κατάλογο αυτό περιέχονται οι επιλογές πρόσβασης σε επιλεγμένες περιοχές της φόρμας ρυθμίσεων κλειδώνοντας τις υπόλοιπες επιλογές για ταχύτερη πρόσβαση.

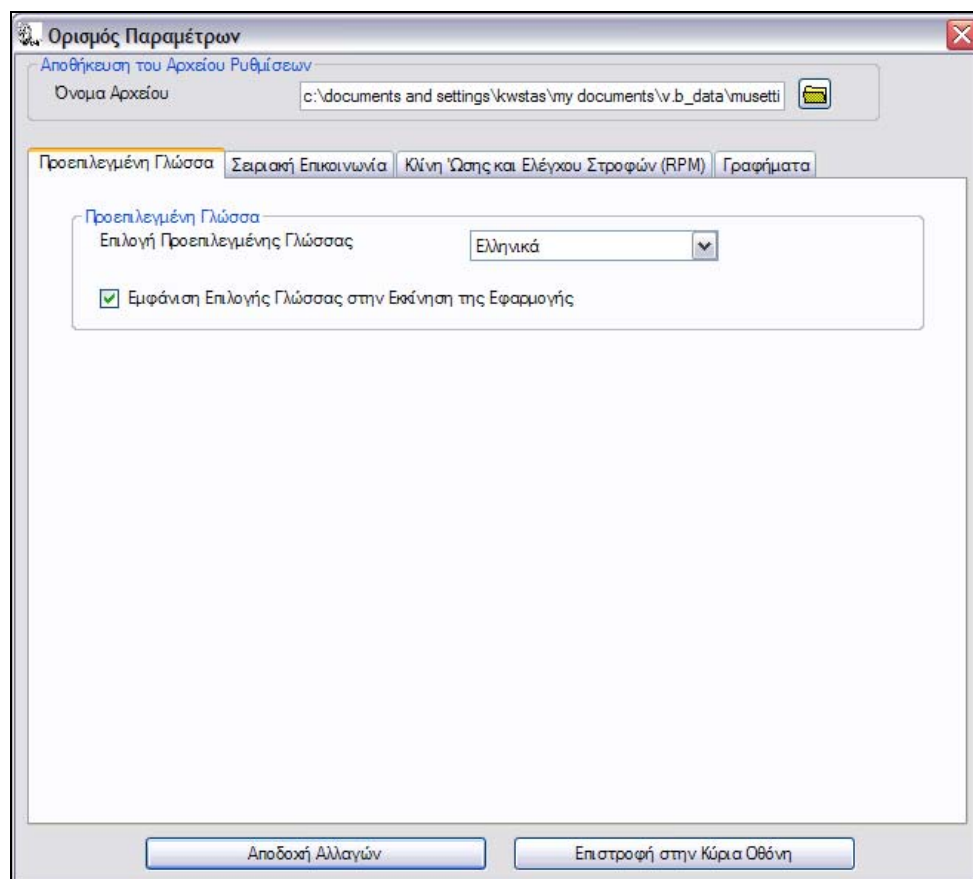


Σχήμα 5.8) Φόρμα επιλογής των παραμέτρων λειτουργίας της εφαρμογή του υπολογιστή Πτυχιακή Καρδάση Κωνσταντίνου.

Επιλέγοντας από τον κατάλογο «Όλες οι παράμετροι» εμφανίζεται η φόρμα εισαγωγής «Ορισμός παραμέτρων». Στη φόρμα αυτή εμφανίζονται όλες οι ρυθμίσεις του χρησιμοποιούμενου αρχείου ρυθμίσεων και η διεύθυνση του εμφανίζεται στο ανώτερη περιοχή της φόρμα αυτής. Σε περίπτωση επιθυμίας δημιουργίας νέου αρχείου τότε πατώντας το πλήκτρο με το φάκελο εμφανίζεται ο γνωστός διάλογος ανοίγματος αρχείων των Windows και δημιουργούμε ένα νέο αρχείο κειμένου «*.txt»

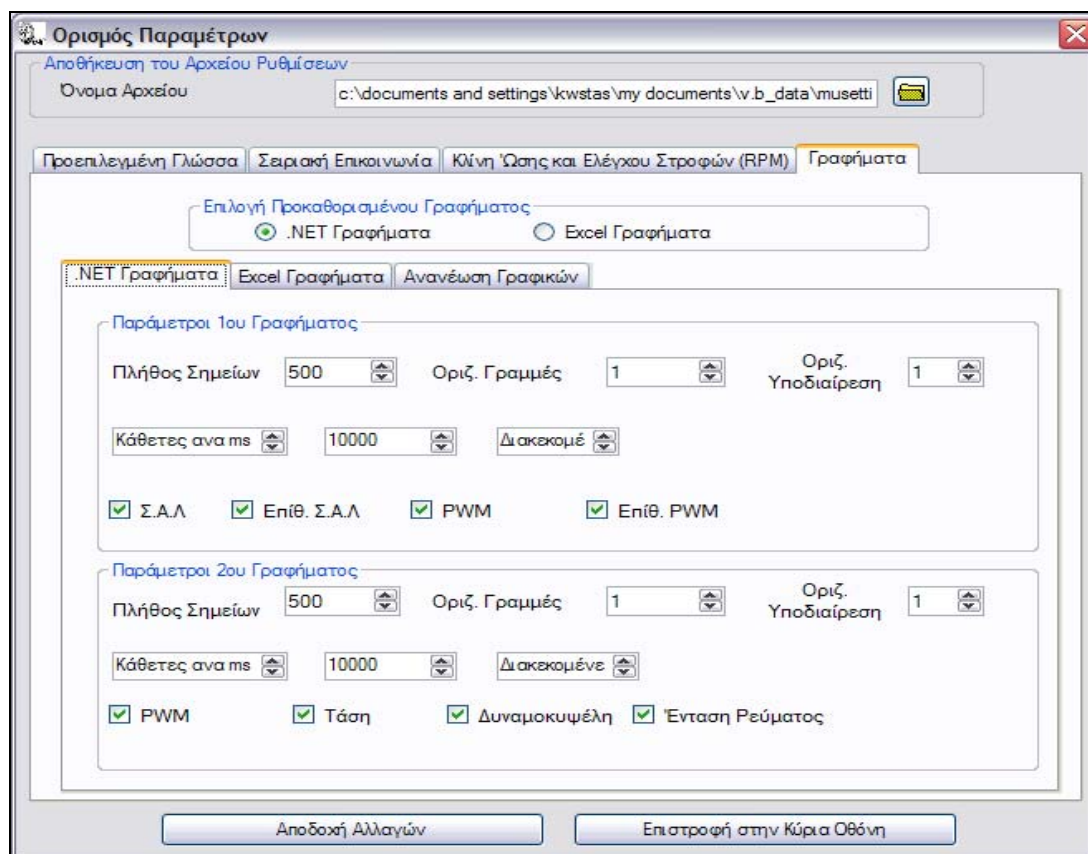
,ενώ τα πεδία της φόρμας ρυθμίσεων θα συμπληρωθούν με τις ρυθμίσεις του βασικού αρχείου «ασφαλείας» των ρυθμίσεων «MyConfs.ini».

Στη φόρμα εισαγωγής των ρυθμίσεων εμφανίζονται οι φάκελοι ρυθμίσεων οι οποίοι περιέχουν τις ρυθμίσεις για κάθε κατηγορία ρυθμίσεων. Οι εμφανιζόμενοι φάκελοι είναι «**Προεπιλογή γλώσσας**» λειτουργίας της εφαρμογής, ρυθμίσεων «**Σειριακής επικοινωνίας**», επιλογής των αρχείων αποθήκευσης και ανάγνωσης των σημείων αναφοράς της «**Κλίνης Ώσης και Ελέγχου στροφών**» και ο φάκελος ρύθμισης των «**Γραφημάτων**» και γραφικών της φόρμας Κλίνης Ώσης και Ελέγχου στροφών.



Σχήμα 5.9) Φόρμα επιλογής της γλώσσας λειτουργίας της εφαρμογής «Πτυχιακή Καρδάση Κωνσταντίνου».

Στο φάκελο «**Σειριακή επικοινωνία**» ρυθμίζονται όλες οι παράμετροι χρήσης της σειριακής όπως όνομα, ταχύτητα, ισοτιμία κ.λ.π. Οι ρυθμίσεις της θύρας αυτή χρησιμοποιούνται γενικά από όλες της διαθέσιμες εφαρμογές «**Σειριακής επικοινωνίας**» και «**Κλίνης Ώσης και Ελέγχου στροφών**». Επιπλέον ρυθμίζεται και η διεύθυνση των αρχείων αποθήκευσης δεδομένων και αποστολής για την εφαρμογή σειριακής επικοινωνίας. Ο φάκελος «**Κλίνης Ώσης και Ελέγχου στροφών**» περιέχει της διευθύνσεις των αρχείων αποθήκευσης εισερχόμενων δεδομένων από την LPC2106 και αποστολής των σημείων αναφοράς.



Σχήμα 5.10) Φόρμα επιλογής των παραμέτρων σχεδίασης των εμφανιζόμενων γραφικών επί της εφαρμογής Κλίνη ώσης και ελέγχου στροφών.

Τέλος στο φάκελο «**Γραφήματα**» περιέχονται όλες οι διαθέσιμες ρυθμίσεις των γραφικών και γραφημάτων τα οποία παρουσιάζονται στη φόρμα της εφαρμογής «**Κλίνης Ώσης και Ελέγχου στροφών**». Σκοπός της ρύθμισης των γραφικών είναι η δυνατότητα τροποποίησης των γραφικών ανάλογα με τη διαθέσιμη ταχύτητα και ισχύ του υπολογιστή ώστε να μην παρατηρούνται καθυστερήσεις στην εκτέλεση των εντολών σε περίπτωση μεγάλου φόρτου επεξεργασίας. Το πρόβλημα αυτό εμφανίστηκε σε παλιότερους υπολογιστές και το οποίο επιλύθηκε είτε με τη μη εμφάνιση των γραφικών παραστάσεων είτε με τη μείωση του ρυθμού ανανέωσης των γραφικών και γραφημάτων. Χαρακτηριστική μορφή του αρχείου ρυθμίσεων είναι η παρακάτω. Για περισσότερες λεπτομέρειες δίνονται στα βοηθήματα.

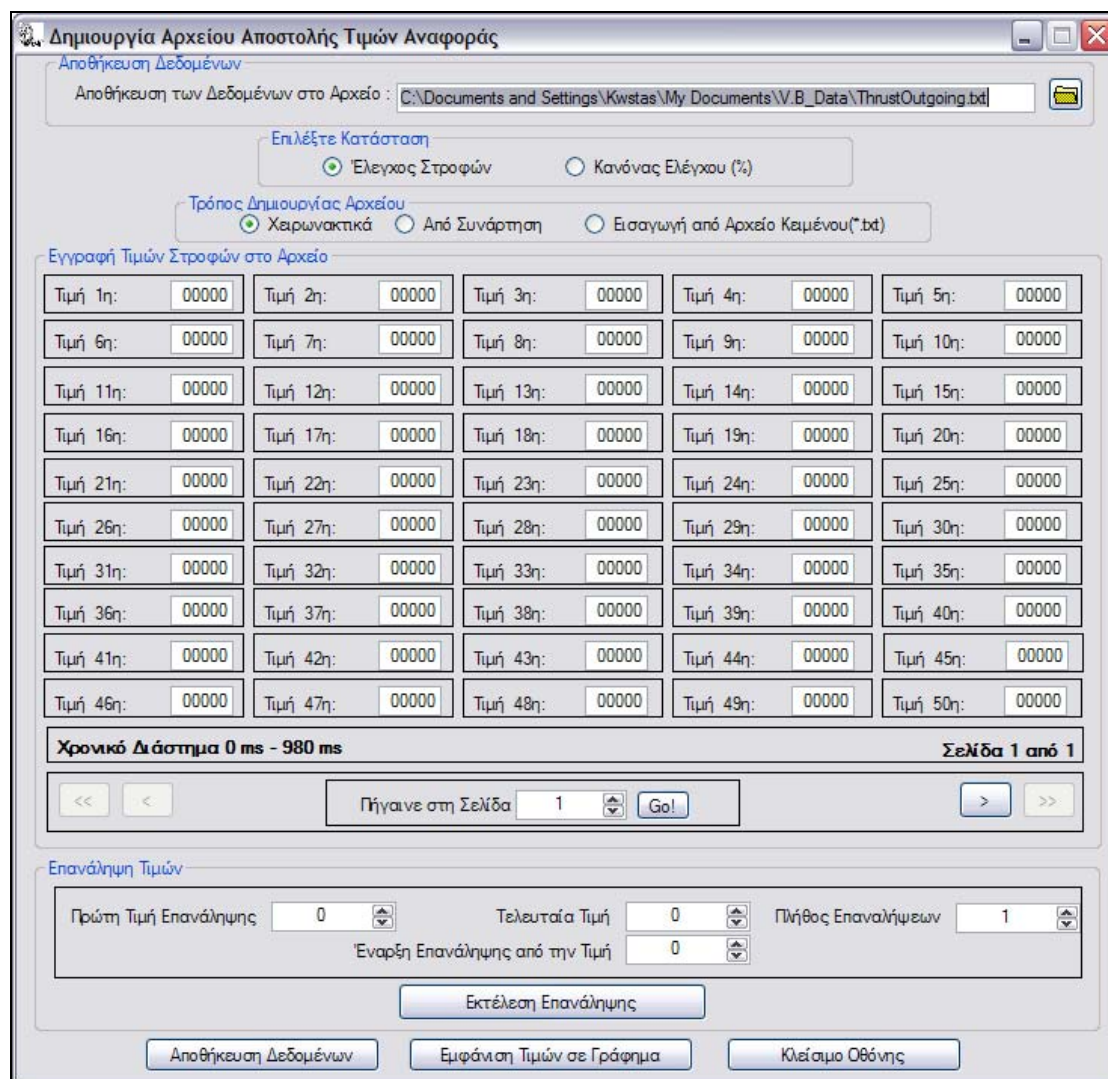
```
[Language Section]
Language = 0
ShowLanguageSelection = 1
[RS232 Communication Section]
Input File = C:\Documents and Settings\Kwstas\My Documents\V.B_Data\SerialInput.txt
Output File = C:\Documents and Settings\Kwstas\My Documents\V.B_Data\SerialOutput.txt
File To Send = C:\Documents and Settings\Kwstas\My Documents\V.B_Data\SerialSendingFile.txt
COM Port = 0
Baud Rate = 7
Parity = 2
Data Bits = 3
Stop Bits = 0
[Thrust Section]
Incoming File = C:\Documents and Settings\Kwstas\My Documents\V.B_Data\ThrustIncoming.txt
Outgoing File = C:\Documents and Settings\Kwstas\My Documents\V.B_Data\ThrustOutgoing.txt
[Graphs Section]
NetGraphs = 0
ExcelGraphs = 1
NetC1NumOfPoints = 500
NetC1HorLines = 1
NetC1HorDiv = 1
NetC1VerticalLine = 0
NetC1TimeVertInterval = 10000
NUDNetC1ConstVerLines = 0
NetC1TypeLines = 0
NetC1Rpm = 0
NetC1SetPointRPM = 0
NetC1PWMLoad = 0
NetC1SetPointPwm = 0
NetC2NumOfPoints = 500
NetC2HorLines = 1
NetC2HorDiv = 1
```

Σχήμα 5.11) Στιγμιότυπο της διαμόρφωσης του αρχείου ρυθμίσεων της εφαρμογής του υπολογιστή.

5.16.3) Εφαρμογή δημιουργίας αρχείων αποστολής σημείων αναφοράς της εφαρμογής Κλίνη Ώσης και Ελέγχου στροφών.

Στο κεφάλαιο 5.15 αναφέρεται ότι ανάμεσα στις απαιτήσεις του προγράμματος είναι η κατασκευή του αρχείου αποστολής, με τα σημεία αναφοράς, τα οποία θα αποσταλούν από την εφαρμογή «Κλίνης Ώσης και Ελέγχου Στροφών». Οι λόγοι για τους οποίους ζητείται η κατασκευή του προγράμματος έχουν εξηγηθεί σε προγενέστερα κεφάλαια «4». Περιληπτικά αναφέρεται ότι χρησιμοποιείται για τη διεξαγωγή πειραμάτων «μετρήσεων» προσφέροντας έτσι τη δυνατότητα επανάληψης της πειραματικής διαδικασίας και τον καθορισμό των τιμών δράσης ελέγχου αυστηρά συναρτήσει του χρόνου. Ο σκοπός της εφαρμογής «**Δημιουργία Αρχείου Αποστολής Τιμών Αναφοράς**» είναι η δημιουργία του αρχείου αποστολής από αυτήν και μόνο, ώστε να ελεγχθούν οι τιμές αναφοράς και να ρυθμιστούν τα αναγκαία δεδομένα και ιδιότητες που πρέπει να πληροί για να αναγνωστεί σωστά και να αποσταλεί από την εφαρμογή «**Κλίνη Ώσης**». Η εμφάνιση της φόρμα αυτής πραγματοποιείται από τον κατάλογο «**Εφαρμογές**» του μενού της κεντρικής φόρμας διαχείρισης.

Η φόρμα «**Δημιουργία Αρχείου Αποστολής Τιμών Αναφοράς**» έχει την παρακάτω μορφή.

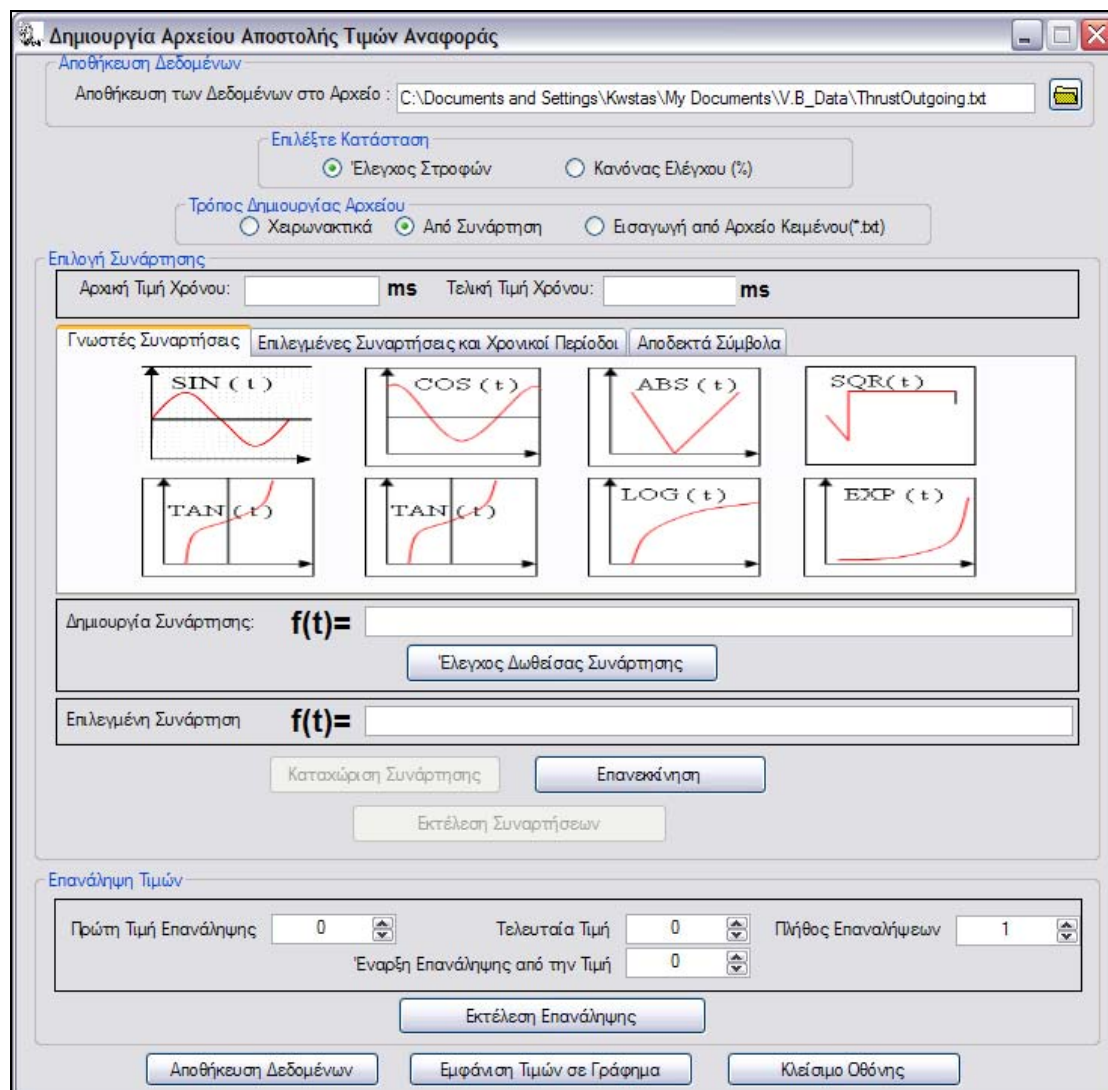


Σχήμα 5.12) Οθόνη λειτουργίας της εφαρμογής Δημιουργία Αρχείου Αποστολής.

Για λόγους βελτιστοποίησης και ολοκληρωμένης λειτουργίας εκτός από τη δυνατότητα εισαγωγής των τιμών αναφοράς απευθείας στα εμφανιζόμενα κελιά μια προς μία δίνεται η δυνατότητα επανάληψης μέρους των εισαγόμενων τιμών. Η επανάληψη προσφέρεται για την επιτάχυνση της εισαγωγής των τιμών, και της δημιουργίας περιοδικών επαναλήψεων των τιμών πχ. Ημιτονοειδή μορφή.

Η εισαγωγή των τιμών στα κελιά προϋποθέτει από το χρήστη τον προϋπολογισμό όλων των επιθυμητών τιμών αναφοράς. Επιπλέον λαμβάνοντας υπόψιν την απαίτηση της απόσταση 20msec μεταξύ τους έχει σαν αποτέλεσμα τη ζήτηση μεγάλου πλήθους εγγραφών στα κελιά ακόμα και για μικρά χρονικά διαστήματα. Πχ. Για ένα λεπτό θα απαιτηθούν $60\text{Sec} * 50\text{values/Sec} = 3000$ εγγραφές. Η απαίτηση της περιόδου των 20ms εξηγείται στο κεφάλαιο 4.7. Αντίθετα αν οι τιμές προκύπτουν από τη δειγματοληψία αναλυτικών μαθηματικών σχέσεων τότε είναι αναγκαία μόνο η γνώση των μαθηματικών σχέσεων και τα χρονικά όρια στα οποία ισχύουν αυτές. Αυτή η παρατήρηση οδήγησε στην κατασκευή και δεύτερου τρόπου εισαγωγής των τιμών με τη χρήση των μαθηματικών σχέσεων. Η δυνατότητα αυτή ενεργοποιείται επιλέγοντας στον πίνακα «**Τρόπος Δημιουργίας Αρχείου**» την επιλογή «**Από Συνάρτηση**». Τα

κελιά εξαφανίζονται και στη θέση τους εμφανίζεται οι διάλογοι εισαγωγής των συναρτήσεων και βοήθειας.

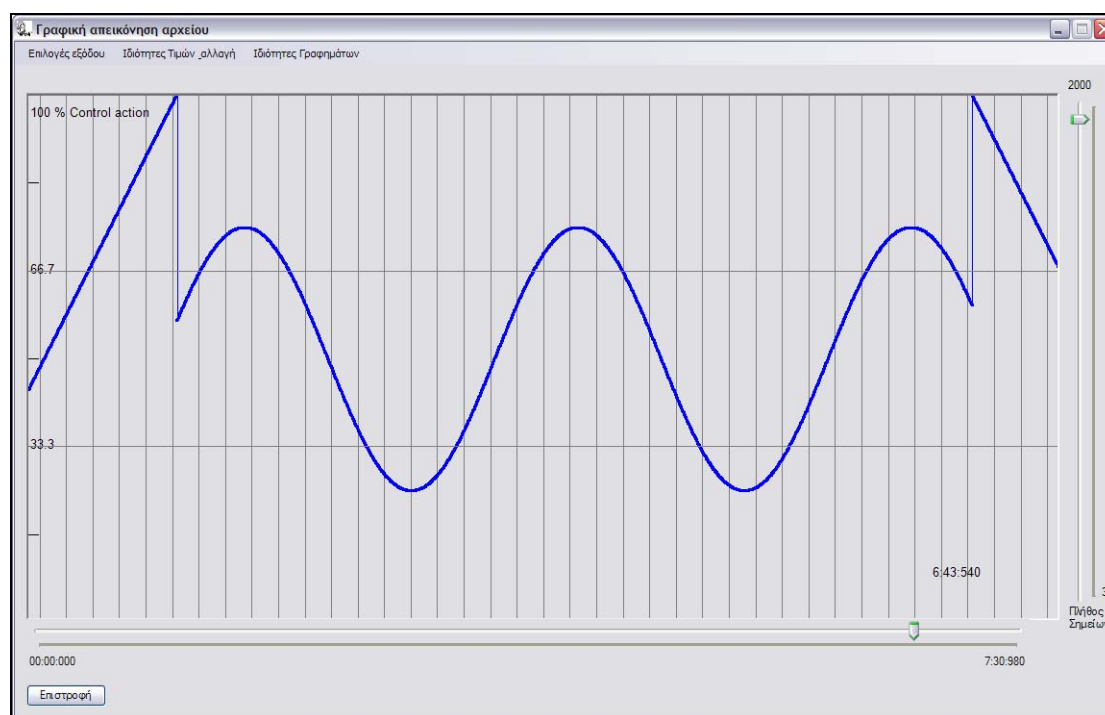


Σχήμα 5.13) Οθόνη δημιουργίας των τιμών του αρχείου αποστολή μέσω συναρτήσεων.

Αναλυτικές πληροφορίες για τη συμπλήρωση και χρήση της φόρμας εισαγωγής των συναρτήσεων δίνεται στο εγχειρίδιο χρήσης. Στην παραπάνω φόρμα φαίνονται τα πεδία εισαγωγής των ορίων των συναρτήσεων σε msec, το πεδίο γραφής της συνάρτησης, το πεδίο ελέγχου της ορθότητας της συνάρτησης και της κατάχωσης. Η καταχώριση έχει την έννοια της επικύρωσης της ισχύος της συνάρτησης στο διάστημα αυτό και το δεσμεύει ώστε να μην είναι δυνατή η εφαρμογή άλλης συνάρτησης εντός του διαστήματος αυτού. Η διαμόρφωση του αρχείου πραγματοποιείται στο τέλος με την επιλογή «**εκτέλεση συναρτήσεων**». Ο υπολογισμός των σημείων αναφοράς πραγματοποιείται από τον υπολογισμό των τιμών των συναρτήσεων με τη δειγματοληψία, ανά 20 msec, των τιμών των συναρτήσεων στα διαστήματα ορισμού τους. Στη συνέχεια μεταφέρονται οι τιμές αυτές στα αντίστοιχα κελιά της προηγούμενης φόρμας εισαγωγής. Τα διαστήματα ή στιγμές στα οποία δεν ορίζεται κάποια συνάρτηση συμπληρώνονται με την τιμή 0.

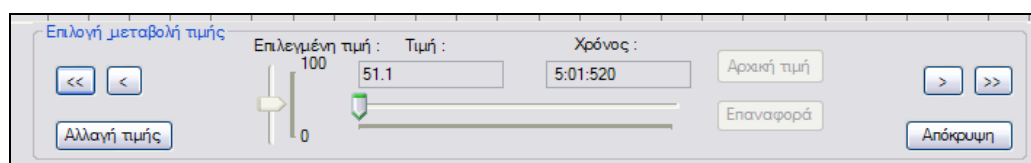
Ο υπολογισμός των τιμών των συναρτήσεων πραγματοποιείται με τη χρήση της γλώσσας «VBScript». Η γλώσσα αυτή είναι προϊόν της Microsoft χρησιμοποιείται για τη γραφή προγραμμάτων σε μορφή κειμένου το οποίο είναι δυνατό να αναγνωρίσει το λειτουργικό των windows και να το εκτελέσει ,εφόσον ο μεταφραστής υπάρχει ήδη ενσωματωμένος στο λειτουργικό. Η χρήση της ξεκίνησε για δικτυακές εφαρμογές τύπου «server-client» για τη μεταφορά της υπολογιστικής διαδικασίας στον client ,απελευθερώνοντας τον server από αυτή, μέσω της μεταφοράς του κώδικα σε μορφή κειμένου. Γρήγορα επεκτάθηκε σε πολλές εφαρμογές εξαιτίας της απλότητας της γραφής της και της μη αναγκαίας χρήσης εξειδικευμένου προγράμματος ανάπτυξης, όπου ένας κειμενογράφος είναι αρκετός. Στο φάκελο «**Επιλεγμένες Συναρτήσεις και Χρονικές Περίοδοι**» εμφανίζονται όλες οι καταχωρημένες συναρτήσεις με τα πεδία τους. Όπως αναφέρθηκε ο υπολογισμός πραγματοποιείται μόνο με την «**Εκτέλεση**» των «**Συναρτήσεων**» οι οποίες εξακολουθώς αποθηκεύονται στα προαναφερόμενα κελιά διαγράφοντας τις προηγούμενες τιμές που περιείχαν. Επομένως για να δούμε τις τιμές που κατέχουν τα κελιά αρκεί να επιλέξουμε από τον πίνακα «**Τρόπος Εισαγωγής**» την επιλογή «**Χειρωνακτικά**» διατηρώντας ταυτόχρονα τη δυνατότητα αλλαγών.

Η παρουσίαση των τιμών στα κελιά εισαγωγής δεν μπορεί να αποδώσει τη γενική εικόνα των τιμών συναρτήσεως του χρόνου. Με σκοπό την εποπτεία των τιμών και την απεικόνιση τους συναρτήσεως του χρόνου σε μεγαλύτερη κλίμακα από ότι μπορεί να δώσει η παρουσίαση στα κελιά δημιουργήθηκε η φόρμα «εμφάνιση τιμών σε γράφημα». Με τη φόρμα αυτή μεταφέρονται όλες οι τιμές των κελιών και σχεδιάζονται πάνω σε ένα γράφημα. Από τα σημεία που ορίζουν οι τιμές των κελιών συναρτήσεως του αντίστοιχου χρόνου ,για κάθε κελί, σχεδιάζεται μία γραμμή η οποία τα ενώνει και αναπαριστά γραφικά την τιμή του σημείου αναφοράς στην LPC2106. Η γραφική αναπαράσταση γίνεται με βάση τη χρήση των τιμών αναφοράς από την LPC2106 ώστε ο χρήστης να έχει καλύτερη αντίληψη της πορείας εκτέλεσης των τιμών του αρχείου που έχει δημιουργήσει.



Σχήμα 5.14) Οθόνη εμφάνισης των τιμών του αρχείου αποστολής της εφαρμογής Δημιουργία Αρχείου Αποστολής.

Εκτός από τη γραφική αναπαράσταση του αρχείου δίνεται η δυνατότητα τροποποίησης των τιμών μεμονωμένα μέσω κατάλληλων χειριστηρίων και εισαγωγής των τιμών. Παρέχεται η δυνατότητα εκτέλεσης διορθώσεων χωρίς τη μεταφορά στην προηγούμενη φόρμα χειρωνακτικής εισαγωγής των τιμών στα κελιά ή στην εισαγωγή από συνάρτηση και επανάληψη της όλης διαδικασίας διορθώνοντας κατάλληλα τις συναρτήσεις. Η πρόσβαση στα χειριστήρια αυτά πραγματοποιείται από τον κατάλογο «**Ιδιότητες τιμών Αλλαγή**» στο εμφανιζόμενο μενού στην κορυφή της φόρμας αυτής. Το επιλεγμένο σημείο προς αλλαγή εμφανίζεται με διαφορετικό χρώμα από τα σχεδιασμένα σημεία των τιμών και από τον πίνακα «επιλογής και αλλαγής των τιμών» τροποποιούμε την τιμή του αντίστοιχου σημείου-αναφοράς.



Σχήμα 5.15) Πίνακας ελέγχων και τροποποίησης των τιμών του αρχείου αποστολής στην οθόνη εμφάνισης των τιμών του αρχείου αποστολής της φόρμας Δημιουργία Αρχείου Αποστολής.

Επειδή ο αριθμός των εμφανιζόμενων σημείων «κελιών» είναι περιορισμένος παρέχονται και χειριστήρια καθοδήγησης των εμφανιζόμενων κελιών με χρονική σειρά «**navigation**» καθώς και τροποποίησης του πλήθους των εμφανιζόμενων σημείων.

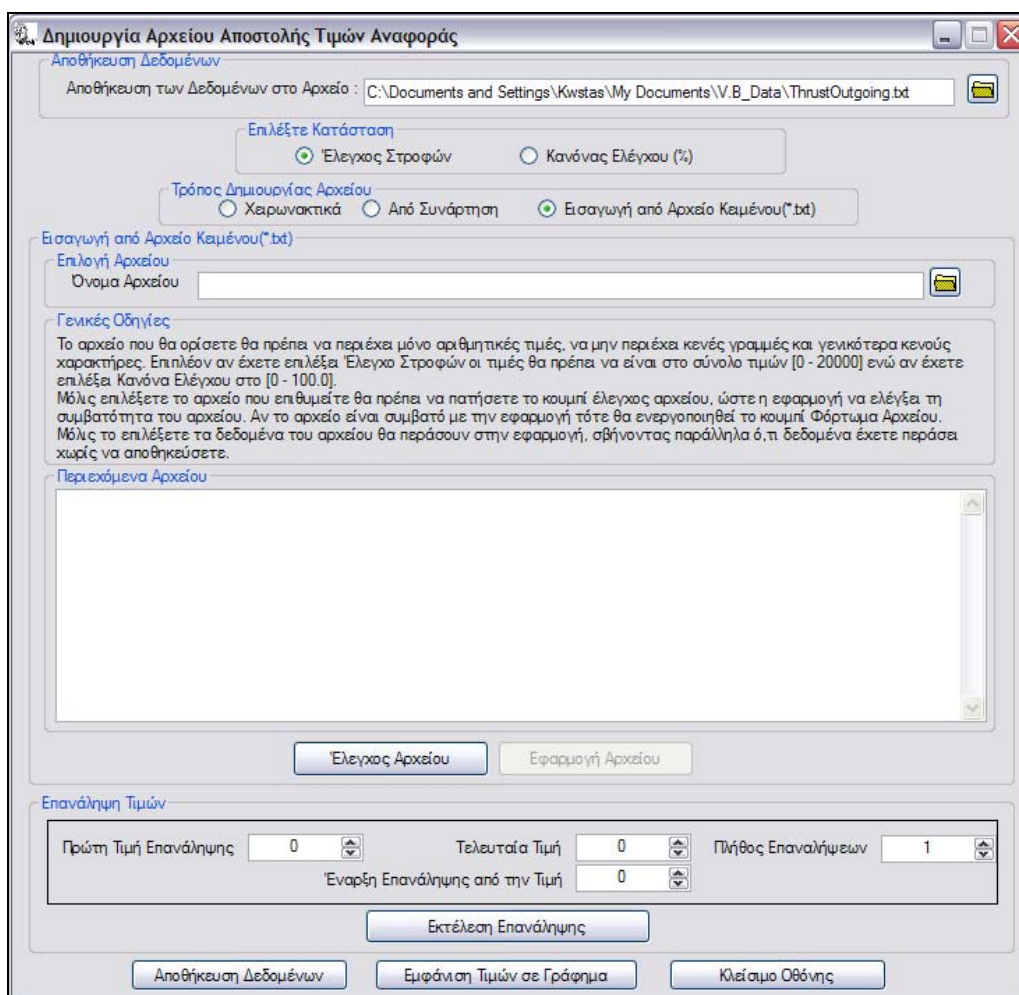
Τέλος παρέχεται η δυνατότητα εφαρμογής των νέων τιμών των κελιών ή απόρριψης των αλλαγών κατά τον τερματισμό της φόρμας «**Γραφικής απεικόνισης των τιμών**».

5.16.4) Εισαγωγή τιμών από αρχείο κειμένου.

Στο κεφάλαιο 5.15 ανάμεσα στις υπόλοιπες απαιτήσεις ζητείται και η δυνατότητα εισαγωγής των τιμών στο αρχείο αποστολής από αρχεία κειμένου τα οποία έχουν διαμορφωθεί από άλλες εφαρμογές. Για να είναι δυνατή η αναγνώριση των περιεχόμενων δεδομένων του αρχείου κειμένου «*.txt» από τη φόρμα «**Δημιουργίας Αρχείου Αποστολής**» ορίζεται μια πρότυπη μορφή η οποία πρέπει να πληρείται ώστε να αναγνωστούν σωστά και να αντιστοιχισθούν σταρέποντα κελιά της εφαρμογής. Η ανάγνωση του αρχείου τιμών πραγματοποιείται από την επιλογή «**Εισαγωγή από Αρχείο**» στον πίνακα «**Τρόπος Δημιουργίας Αρχείου**». Με την επιλογή του εμφανίζονται στη θέση των κελιών οι πίνακες «**Επιλογής Αρχείου**», «**Οδηγιών**», «**Εμφάνισης των τιμών**» καθώς και τα κουμπιά «**Ανάγνωσης του Αρχείου**» και «**Εφαρμογής**» των αναγνωσμένων τιμών. Με το κουμπί επιλογής αρχείου εμφανίζεται ο συνηθής διάλογος ανοίγματος αρχείων των «windows». Στη συνέχεια με το κουμπί «**Ανάγνωσης**» ανοίγεται το αρχείο και εφόσον πληροί την πρότυπη μορφή διαβάζονται οι τιμές και παρουσιάζονται στον πίνακα παρουσίασης. Για την εφαρμογή τους και μεταφορά των τιμών στα κελιά χρησιμοποιείται το κουμπί «**Εφαρμογή αρχείου**».

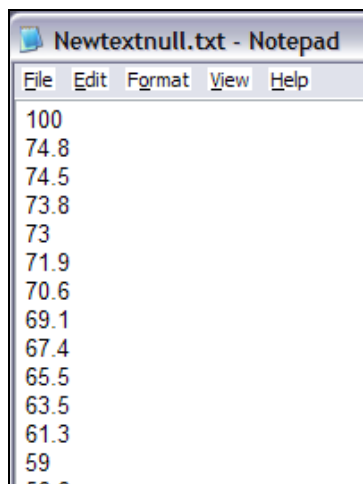
Η πρότυπη μορφή που πρέπει να έχει το αρχείο κειμένου είναι η εξής :

1. Να περιέχει μόνο αριθμητικούς χαρακτήρες.
2. Σε κάθε γραμμή αντιστοιχείται μια και μόνο τιμή αναφοράς.
3. Κάθε γραμμή αντιστοιχείται σε ένα κελί με φόρα αντιστοίχισης τη φορά ανάγνωσης .Δηλαδή η πρώτη γραμμή αντιστοιχείται στο πρώτο κελί ,στιγμή εκκίνησης, η δεύτερη γραμμή στο δεύτερο κελί και ούτω καθ' εξής.
4. Ως χαρακτήρας υποδιαστολής για τα δεκαδικά ψηφία ορίζεται ο χαρακτήρας «.» της τελείας. Ενώ δεν πρέπει να χρησιμοποιείται χαρακτήρας χιλιάδων κ.τ.λ.
5. Αν τιμές αναφοράς αφορούν τη μέθοδο έλεγχου στροφών οφείλουν να είναι ακέραιες με εύρος τιμών από 0 έως 20000.
6. Αν οι τιμές αναφοράς αφορούν τη μέθοδο Κανόνας Ελέγχου θα πρέπει να είναι δεκαδικές και να παίρνουν τιμές από 0.0 έως 100.00.



Σχήμα 5.16) Οθόνη εισαγωγής των τιμών αναφοράς του αρχείου αποστολής από αρχείο κειμένου το οποίο δεν είναι δημιουργημένο μέσω της εφαρμογής Δημιουργία Αρχείου Αποστολής.

Πχ. Ένα τέτοιο αρχείο το οποίο πληροί τις παραπάνω προϋποθέσεις είναι :



Σχήμα 5.17) Στιγμιότυπο της διαμόρφωση του αρχείου κειμένου το οποίο περιέχει τα σημεία αναφοράς το οποία θα εισαχθούν στην εφαρμογή «Δημιουργία Αρχείου Αποστολής» .

5.16.5) Τυποποίηση του αρχείου αποστολής

Το αρχείο αποστολής το οποίο διαμορφώνεται από τη φόρμα δημιουργίας αρχείου προορίζεται να χρησιμοποιηθεί στη φόρμα «**Κλίνη Ώσης και Ελέγχου στροφών**». Επομένως θα πρέπει στο αρχείο εκτός από τις τιμές να περιέχονται όλες οι απαραίτητες πληροφορίες δημιουργίας και η μέθοδος ώστε να μπορέσει να χρησιμοποιηθεί κατάλληλα και για να απαλλάξει το χρήστη από τα να προβεί σε όλες τις κατάλληλες ρυθμίσεις. Επιπροσθέτως για να είναι δυνατή η τροποποίηση ενός τέτοιου αρχείου μέσω της φόρμας «**Δημιουργίας αρχείου**» μετά την γέννηση του, χωρίς να απαιτείται η επαναδημιουργία του, έχουν συμπληρωθεί οι αναγκαίες πληροφορίες.

Για την επίτευξη των παραπάνω σκοπών αποφασίστηκε ότι οι πληροφορίες που είναι απαραίτητες να περιέχονται στο αρχείο είναι :

1. Η μέθοδος λειτουργίας της LPC2106 διαχωρίζεται αν τα σημεία αναφοράς αφορούν τη μέθοδο «**Έλεγχος Στροφών**» ή τον «**Κανόνα ελέγχου**». Με βάση την πληροφορία αυτή η εφαρμογή «**κλίνη ώσης**» ελέγχει τις τιμές που διαβάζονται από το αρχείο αν πληρούν τα όρια, 0 έως 20000 ή 0.0 έως 100.00, πριν σταλούν, ώστε να μη δημιουργηθεί πρόβλημα από την εφαρμογή τους από την LPC2106 .Επιπλέον κατά την ανάγνωση του αρχείου ,από τη φόρμα «**Δημιουργίας Αρχείου Αποστολής**», συμπληρώνεται αυτόματα το κατάλληλο πεδίο επιλογής της κατάστασης υπολογισμού των τιμών στον πίνακα «**Επιλέξτε Κατάσταση**». Έτσι ο χρήστης μπορεί να αντιληφθεί ότι οι τιμές που περιέχονται στο αρχείο αφορούν την κατάσταση «**Έλεγχος στροφών**» ή «**Κανόνας ελέγχου**» και να προβεί σε προσθήκες και διορθώσεις ή στη διαγραφή των υπαρχόντων και δημιουργία νέου αρχείου με το όνομα αυτό.
2. Ο τρόπος δημιουργίας του αρχείου πριν την τελευταία αποθήκευση των τιμών σε αυτό. Η πληροφορία αυτή χρησιμοποιείται μόνο από τη φόρμα

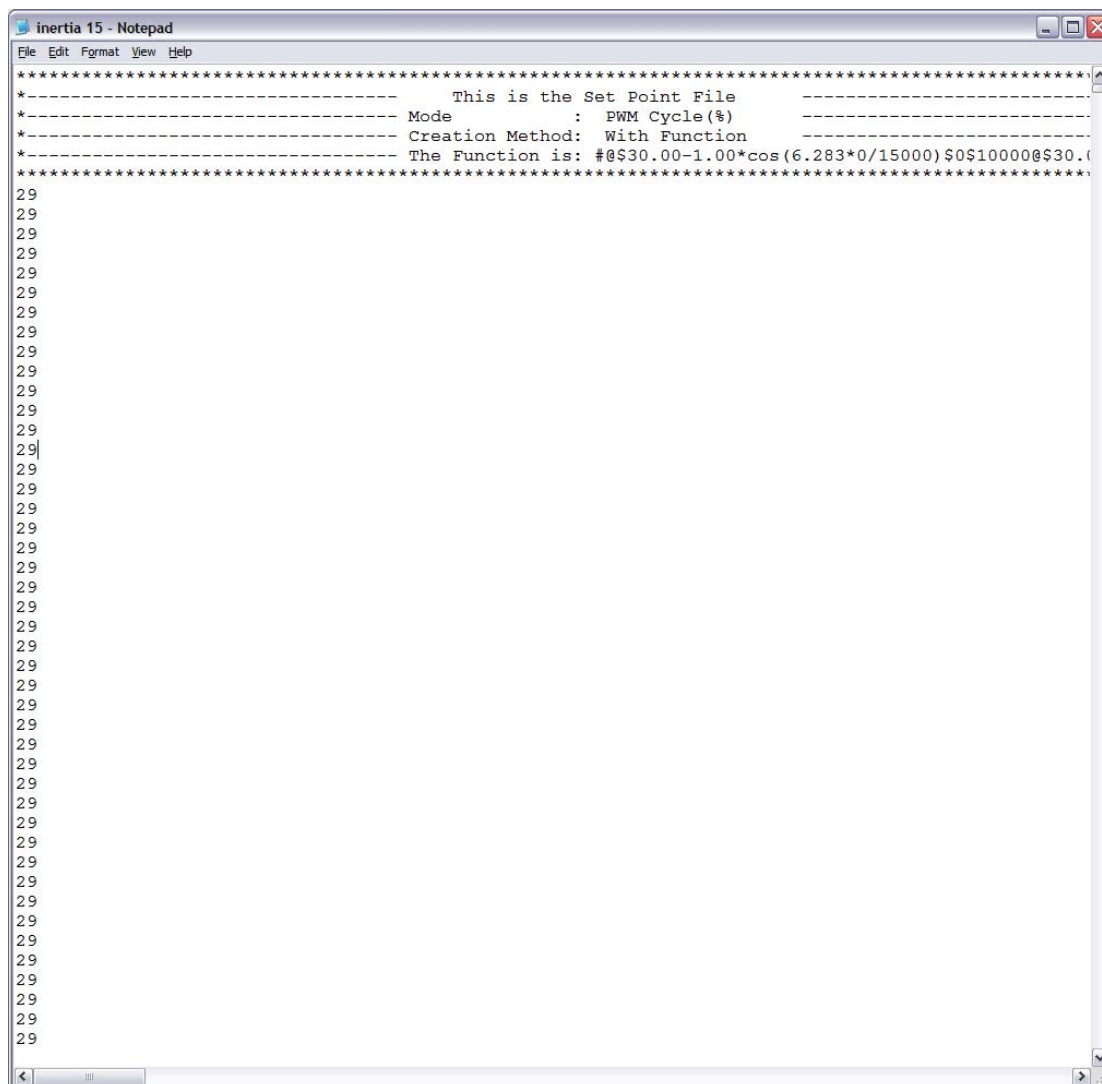
της εφαρμογή «**Δημιουργία αρχείου Αποστολής**». Σκοπός της είναι η πληροφόρηση του χρήστη ότι οι τιμές που περιέχει δημιουργήθηκαν με τη μέθοδο της χειρονακτικής εισαγωγής ή του υπολογισμού τους μέσω συναρτήσεων ή από την εισαγωγή από αρχείο. Με βάση την πληροφορία αυτή μπορεί να προβεί στις τροποποιήσεις που θέλει.

3. Τέλος στην περίπτωση που η μέθοδος δημιουργίας του αρχείου είναι «**από συνάρτηση**» αποθηκεύεται και η αναλυτική μορφή αυτών καθώς και τα όρια χρήσης τους. Ομοίως όπως και στην προηγούμενη παράγραφο η χρήση της πληροφορίας αυτή αφορά μόνο την εφαρμογή «**Δημιουργία αρχείο Αποστολής**» και αποσκοπεί στο να εμφανίσει στο χρήστη τις συναρτήσεις και τα όρια εφαρμογής τους ώστε να προβεί στις κατάλληλες τροποποιήσεις που επιθυμεί. Η εμφάνιση τους πραγματοποιείται όταν ολοκληρωθεί η ανάγνωση του αρχείου και αφού εμφανιστεί ο αντίστοιχος πίνακας εισαγωγής των συναρτήσεων στον φάκελο «**Επιλεγμένες Συναρτήσεις και Χρονικές περιόδου**» .

Η αποθήκευση των παραπάνω πληροφοριών πραγματοποιείται στις πρώτες έξι γραμμές του αρχείου αποστολής ως επικεφαλίδα. Η πρώτη γραμμή συμπληρώνεται με το χαρακτήρα «*» ενώ στη δεύτερη αναγράφεται το λεκτικό «----- This is the Set Point File -----». Στην Τρίτη γραμμή αναγράφεται η σημαντικότερη πληροφορία της μεθόδου λειτουργίας της LPC2106 για τις αποθηκευμένες τιμές. Η γραφή της γραμμής αυτής είναι τυποποιημένη και ορίζεται να περιέχει στα αριστερά το λεκτικό «*----- Mode : » ενώ αμέσως μετά ακολουθούν τα λεκτικά «PWM Cycle(%)» ή «Control RPM» αντίστοιχα για τη μέθοδο «Κανόνα Ελέγχου» και «Έλεγχος Στροφών» αντίστοιχα. Σε καμία περίπτωση δεν πρέπει να περιέχονται στη γραμμή αυτή και τα δυο προηγούμενα λεκτικά. Η τέταρτη γραμμή χρησιμοποιείται για τη γραφή του τρόπου δημιουργίας του αρχείου. Είναι τυποποιημένη για λόγους ανάγνωσης και στα αριστερά γράφεται πάντα το λεκτικό «*----- Creation Method:» και αμέσως μετά ένα από τα λεκτικά «Manually», «With Function» και «Import from File». Στην πέμπτη γραμμή γράφεται η αναλυτική μορφή των συναρτήσεων και τα όρια αυτών, εισάγοντας μπροστά από κάθε συνάρτηση και όριο το χαρακτήρα «&». Κάθε συνάρτηση μαζί με τα όρια τους χωρίζεται από την επόμενη με το χαρακτήρα « @ ».

Πχ
@\$100\$15001\$25000@\$50+25*cos(0.0001*t)\$25001\$50000@\$50+25*cos(0.001*t)
\$50001\$75000@\$50+25*cos(0.01*t)\$75001\$100000#*

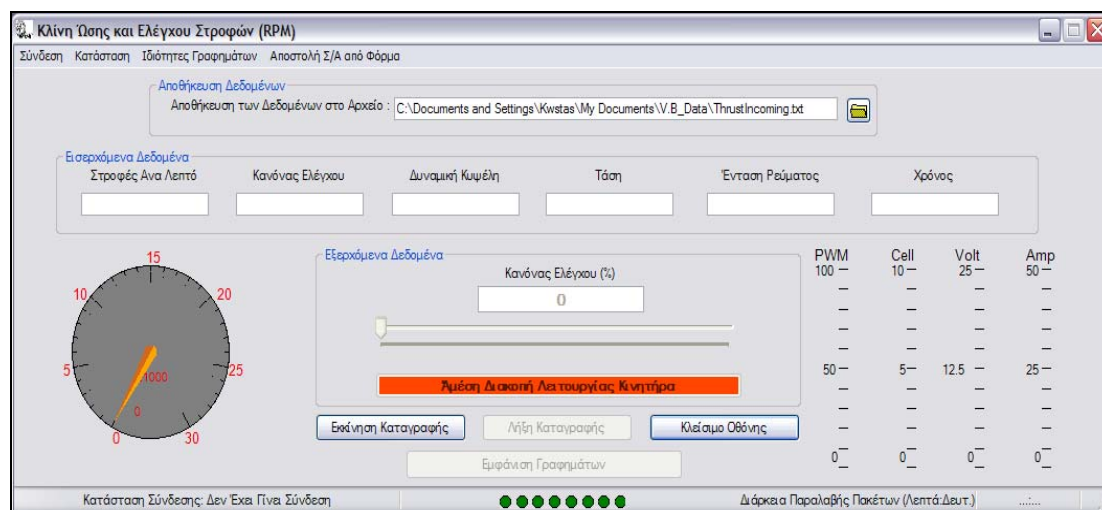
Τέλος η έκτη γραμμή συμπληρώνεται ομοίως από το χαρακτήρα «*» και στις επόμενες γραμμές ακολουθούν οι τιμές των σημείων αναφοράς.



Σχήμα 5.18) Διαμόρφωση του δημιουργημένου αρχείου αποστολής από την εφαρμογή Κλίνη ώσης και Ελέγχου στροφών.

5.16.6) Εφαρμογή Κλίνης Ώσης και Ελέγχου Στροφών.

Ο τελικός σκοπός της εφαρμογής και της πτυχιακής με τη γραφή αυτού του προγράμματος είναι η επικοινωνία με την LPC206 και τη δράση του υπολογιστή ως μονάδα ελέγχου ολόκληρης της πειραματικής διάταξης. Για την καλύτερη δυνατή πληροφόρηση του χρήστη σχετικά με την κατάσταση της εφαρμογής έχουν χρησιμοποιηθεί πλήθος διαφορετικών εργαλείων, γραφικών παραστάσεων και μηνυμάτων πληροφόρησης στη φόρμα της εφαρμογής «**Κλίνη Ώσης και Έλεγχος στροφών**». Επιπλέον προστέθηκαν αυτοματοποιημένες διαδικασίες που απαλλάσσουν το χρήστη από τη συνεχή εξυπηρέτηση επαναλαμβανόμενων διαδικασιών επιτρέποντας του να ασχοληθεί με τον έλεγχο της εγκατάστασης και της πορείας των μετρήσεων.



Σχήμα 5.19) Οθόνη έναρξης της εφαρμογής Κλίνη ώσης και Ελέγχου στροφών.

5.16.7) Αυτοματοποιημένες διαδικασίες και σημαντικές εργασίες.

Στα παρακάτω υποκεφάλαια παρουσιάζονται τα σημαντικότερα χρησιμοποιούμενα εργαλεία, ο αλγόριθμος επικοινωνίας με την LPC2106, και την αυτοματοποίηση διαδικασιών χρονισμού και βασικών ελέγχων που πραγματοποιούνται στην εφαρμογή.

5.16.7.1) Αλγόριθμος επικοινωνίας μέσω της σειριακής θύρας με την LPC2106.

Για την επικοινωνία της εφαρμογής στον υπολογιστή και την LPC2106 πρέπει να χρησιμοποιηθούν τα πρωτόκολλα επικοινωνίας όπως ορίστηκαν στα κεφάλαιο 4.1 & 4.2. Οποιαδήποτε δεδομένα χρειάζεται να αποσταλούν και προς τις δύο κατευθύνσεις μετασχηματίζονται σύμφωνα με το εκάστοτε πρωτόκολλο ενώ ο παραλήπτης τα ανασχηματίζει στην αρχική τους μορφή χρησιμοποιώντας το ίδιο πρωτόκολλο. Έτσι λοιπόν για τη λήψη των δεδομένων η εφαρμογή στον υπολογιστή χρησιμοποιεί το πρωτόκολλο του κεφαλαίου 4.1 ενώ για την αποστολή των σημείων αναφοράς και των εντολών το πρωτόκολλο του κεφαλαίου 4.2 .

Με τη λήψη ενός ολόκληρου πακέτου δεδομένων ,σύμφωνα με το κεφ.4.1 ,δημιουργείται το γεγονός της λήψης δεδομένων στη σειριακή θύρα .Στη συνέχεια εκτελείται η διαδικασία της λήψης δεδομένων «data receive». Στη διαδικασία αυτή εκτός από τον επανασχηματισμό των εισερχόμενων δεδομένων από τη μορφή του πακέτου των bytes εκτελείται και η αποθήκευση αυτών στο αρχείο κειμένου , η ανταπόκριση της εφαρμογής στο bytes ελέγχου και η αποστολή των κατάλληλων εντολών καθώς και η αποθήκευση των εισερχόμενων μετρήσεων των δεδομένων σε ένα προσωρινό πίνακα με σκοπό τη χρήση τους στις γραφικές παραστάσεις.

5.16.7.2) Ανταπόκριση της εφαρμογής στο byte ελέγχου.

Στο κεφάλαιο 4.6 παρουσιάστηκε η δομή του «byte» ελέγχου. Αν στο byte αυτό ζητηθεί η αλλαγή της εκτέλεσης της μεθόδου λειτουργίας της εφαρμογής, σε «έλεγχος στροφών» ή «έλεγχος δράσης PWM», τότε στην εφαρμογή εμφανίζονται τα νέα χειριστήρια για την αντίστοιχη νέα μέθοδο λειτουργίας και στέλνεται μέσω της σειριακής η αντίστοιχη εντολή στην LPC2106 για την αλλαγή. Επίσης αν διαπιστωθεί ασυμφωνία ανάμεσα στη μέθοδο λειτουργίας της εφαρμογής επί του υπολογιστή και της λειτουργίας της LPC2106 τότε αποστέλλεται η αντίστοιχη εντολή στην LPC2106 για να αλλάξει τη μέθοδο λειτουργίας ώστε να γίνει ίδια με του υπολογιστή. Ομοίως ελέγχεται αν είναι κοινή η μέθοδος ανάγνωσης των σημείων αναφοράς, «**ανάγνωση από αρχείο**» ή «**από τη μπάρα εισαγωγής**» στη φόρμα της εφαρμογής, σε αντίθετη περίπτωση στέλνεται η κατάλληλη εντολή στην LPC2106 για αλλαγή της μεθόδου ανάγνωσης ώστε να γίνει όμοια με εκείνη της εφαρμογής.

Τέλος αν ζητηθούν από την κάρτα νέα δεδομένα και εφόσον η μέθοδος ανάγνωσης είναι από αρχείο εκτελείται η διαδικασία ανάγνωσης του αρχείου και αποστολής των τιμών.

5.16.7.3) Αποστολή των επιθυμητών σημείων αναφοράς από τη μπάρα εισαγωγής.

Η εισαγωγή της επιθυμητής τιμής αναφοράς γίνεται είτε μέσω της μπάρας «**Σημείο αναφοράς στροφών**» είτε μέσω της μπάρας «**Κανόνας ελέγχου**». Μετακινώντας τη μπάρα αλλάζει και η τιμή η οποία πρόκειται να σταλεί. Το γεγονός της μετακίνησης της μπάρας στα πλαίσια της εφαρμογής, προκείμενου να διαβαστεί η νέα τιμή και να σταλεί, οδηγούσε σε κορεσμό τον buffer της σειριακής για συνεχείς μεταβολές και στην πρόκληση αστάθειας της LPC2106. Για τη λύση του προβλήματος αυτού επιλέχθηκε η χρήση ενός περιοδικού χρονομέτρου «timer» όπου στο τέλος κάθε περιόδου διαβάζεται η επιθυμητή τιμή από τη μπάρα και στέλνεται μέσω της σειριακής στην κάρτα.

Ο χρήστης της εφαρμογής για να γνωρίζει την τρέχουσα τιμή της μπάρας κάθε φορά που μετακινείται εμφανίζεται η νέα του τιμή στο κουτί κάτω από τον τίτλο «**έλεγχος στροφών**» ή «**κανόνας ελέγχου(%)**». Η τιμή αυτή είναι που πρόκειται να σταλεί στην κάρτα στο τέλος της περιόδου του προαναφερόμενου χρονομετρητή.

5.16.7.4) Περιοδική επανάληψη κρίσιμων διαδικασιών.

Η εφαρμογή στον υπολογιστή και η εκτέλεση της από το λειτουργικό σύστημα των windows έχει αναπτυχθεί υπό το πρίσμα του προγραμματισμού οδηγούμενου από γεγονότα ,βλέπε κεφάλαιο 5.6 . Επομένως η ροή εκτέλεσης της εφαρμογής ανατίθεται πλέον στο χρήστη αυτής μέσω των εντολών του και κατ' επέκταση τα γεγονότα που δημιουργεί με το ποντίκι και το πληκτρολόγιο. Η εκτέλεση διαδικασιών και εντολών περιοδικά δεν συνίσταται η ανάθεση τους στο χρήστη αλλά χρησιμοποιούνται οι λεγόμενοι «timers» οι οποίοι αναλαμβάνουν να δημιουργήσουν αυτοί τα γεγονότα ,αντί του χρήστη, με τα οποία θα εκτελεστούν οι παραπάνω διαδικασίες. Οι «timers» δεν πρέπει να συγχέονται με τη λειτουργία των χρονομετρητών όπως δηλώνει το όνομα τους. Η λειτουργία τους ορίζεται ορθά ως περιοδική γεννήτρια γεγονότων. Δεν έχουν την ικανότητα της άμεσης μέτρησης του χρόνου παρά μόνο τη δημιουργία ενός περιοδικού ή μη γεγονότος σε μία προκαθορισμένη χρονική στιγμή. Τονίζεται ότι το λειτουργικό σύστημα είναι αυτό που ελέγχει το «timer» και το ενημερώνει για την τρέχουσα τιμή του χρόνου του συστήματος. Η ενημέρωση αυτή πραγματοποιείται με τη χρήση ειδικών γεγονότων «ticks» τα οποία καταγράφει ο «timer» και με βάση αυτά υπολογίζει την τιμή του. Δηλαδή τα «ticks» έχουν το ρόλο της εσωτερικής γεννήτριας χρονισμού του «timer» κατά αναλογία με τους «timers» της LPC2106. Κάθε tick αντιστοιχεί σε ένα καθορισμένο χρονικό διάστημα. Η ενημέρωση του «timer» δεν πραγματοποιείται σε πραγματικό χρόνο αλλά όταν το λειτουργικό κρίνει λαμβάνοντας υπόψιν και τα υπόλοιπα γεγονότα που έχει να εκτελέσει. Αποτέλεσμα όλων των παραπάνω είναι η μη ακριβή μέτρηση και επιπλέον ενώ θεωρητικά η ελάχιστη ικανότητα μέτρησής είναι 1 ms στην πράξη κυμαίνεται με ελάχιστη τιμή 50 ms.

Για την εκτέλεση λοιπών περιοδικών ελέγχων και διαδικασιών χρησιμοποιείται ένα πλήθος από «timers», εκ 'των οποίων ο σημαντικότερος ονομάζεται «MyTimer». Η περίοδος του «MyTimer» είναι 100ms και εκκινεί από τη στιγμή που η φόρμα «Ωση και έλεγχος στροφών» εμφανίζεται. Όταν ολοκληρωθεί η περίοδος του «MyTimer» εκτελείται η διαδικασία «MyTimer_Tick» .Στη διαδικασία εκτελείται ο έλεγχος της επικοινωνίας με την LPC2106 και σε περίπτωση καθυστερημένης αποστολής δεδομένων προς τον υπολογιστή ενημερώνεται ο χρήστης εμφανίζοντας μήνυμα για την καθυστέρηση αυτή. Επιπλέον εξετάζεται η συμφωνία της κατάστασης λειτουργίας του υπολογιστή και της LPC2106 μεταξύ των «Έλεγχος Στροφών» ή «Κανόνας Ελέγχου» καθώς και «Αποστολή Τιμών από Αρχείο» ή «Αποστολή τιμών από τη μπάρα». Αν δε συμφωνούν στέλνεται η κατάλληλη εντολή στην LPC2106 ώστε να αλλάξει και να συμφωνήσει με την κατάσταση του υπολογιστή. Τέλος στην περίπτωση που η αποστολή των τιμών από αρχείο έχει ολοκληρωθεί ή δε ζητούνται νέες τιμές για τον έλεγχο ύπαρξης της σειριακής επικοινωνίας στέλνεται η αντίστοιχη εντολή όπως ορίστηκε στο κεφάλαιο 4.4 .

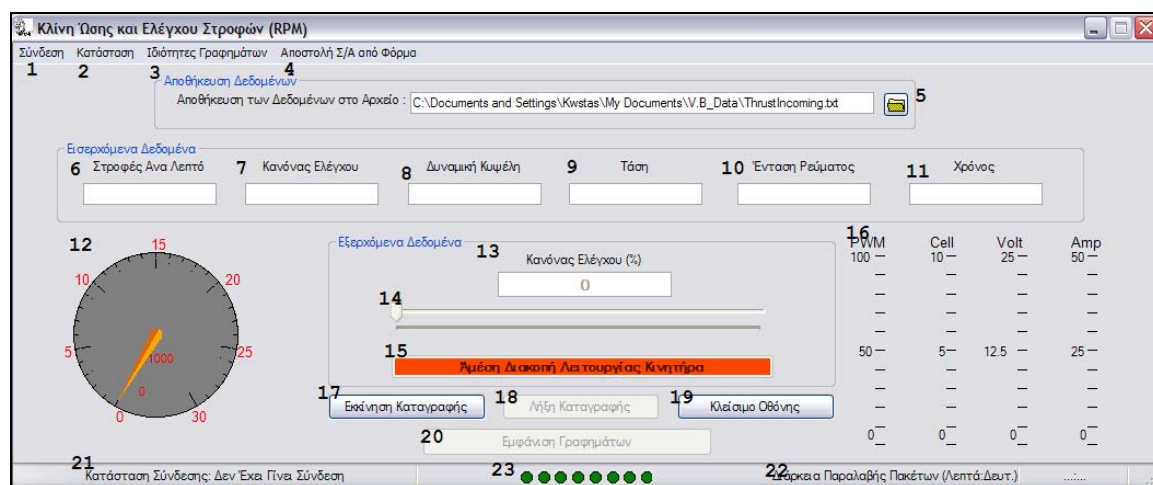
Η χρήση των υπόλοιπων χρονομετρητών αναλώνεται στο χρονισμό της ανανέωσης των γραφικών και πλήθος άλλων λειτουργιών όπως τη δημιουργία «animation» π.χ. στο κουμπί άμεσης διακοπής λειτουργίας κινητήρα.

5.17) Παρουσίαση της φόρμας Κλίνη Ωσης και ελέγχου στροφών.

Παρακάτω ακολουθεί μια εκτεταμένη παρουσίαση της φόρμας ελέγχου και διεξαγωγής των μετρήσεων «Κλίνη Ωσης και Ελέγχου στροφών». Όμως θα

επικεντρωθεί στην παρουσίαση των σημαντικότερων λειτουργιών και ειδικών περιπτώσεων χειρισμού και εμφανιζόμενων μηνυμάτων. Αναλυτική παρουσίαση υπάρχει όπως και για όλη την εφαρμογή στα βοηθήματα στο κεφάλαιο «Εγχειρίδιο χρήσης» καθώς και στη βοήθεια της εφαρμογής.

Η εκκίνηση της εφαρμογής πραγματοποιείται με την επιλογή στον κατάλογο «Εφαρμογές» του μενού του προγράμματος «Πτυχιακή Κωνσταντίνου Καρδάση» της εφαρμογής «Κλίνη Ώσης και Έλεγχου Στροφών». Κατά την εκκίνηση της εφαρμογής διαβάζονται όλες οι αποθηκευμένες ρυθμίσεις από το τρέχον αρχείο ρυθμίσεων ώστε να εκκινήσει σύμφωνα με τις επιλεγμένες ρυθμίσεις. Το παρακάτω παράθυρο εξαφανίζεται κατά την εκκίνηση της «Κλίνη Ώσης Και Ελέγχου Στροφών» με αριθμημένα τα εμφανιζόμενα πεδία επικοινωνίας με το χρήστη .



Σχήμα 5.20) Αρίθμηση και διάταξη των κουμπιών και ελέγχων της οθόνης έναρξης και λειτουργίας της Κλίνη ώσης και Ελέγχου στροφών.

Τα κουμπιά ελέγχου, επικοινωνίας και το μενού σημειώνονται με τους παραπάνω αριθμούς και η λειτουργία τους δίνεται αμέσως παρακάτω.

Σημειώνονται τα πεδία ελέγχου της εφαρμογής .

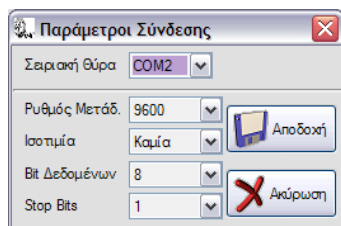
1. Εμφάνιση Φόρμας Παραμέτρων Σύνδεσης , Εκκίνηση και Λήξης.
2. Επιλογή της μεθόδου ελέγχου της εφαρμογής και της Κάρτας ελέγχου Philips .
3. Επιλογή Εμφάνισης και Απόκρυψης Γραφημάτων και Πινάκων Ιδιοτήτων Γραφημάτων .
4. Αποστολή Σ/Α από Αρχείο.
5. Επιλογής Αρχείου αποθήκευσης Εισερχόμενων Δεδομένων από την κάρτα Philips .
6. Εμφάνιση της τιμής μέτρησης των Στροφών ανά λεπτό του Ηλεκτροκινητήρα.
7. Εμφάνιση της τιμής του PWM Duty Cycle σε 100% της δράσης ελέγχου της κάρτας Philips .
8. Εμφάνιση της μετρούμενης Ώσης.
9. Εμφάνιση της εφαρμοζόμενης τάσης στα άκρα τροφοδοσίας του ηλεκτροκινητήρα .

10. Εμφάνιση τρέχουσας τιμής της Έντασης Ρεύματος ,που διαρρέει τον ηλεκτροκινητήρα .
11. Εμφάνιση του χρόνου δειγματοληψίας της κάρτας από τη στιγμή της εκκίνησης και επικοινωνίας με την εφαρμογή .
12. Αναλογικό Στροφόμετρο εμφάνισης της τιμής των στροφών .
13. Εμφάνιση του επιλεγμένου προς αποστολή σημείου αναφοράς των επιθυμητών τροφών ή PWM Duty Cycle σε 100% αντίστοιχα με τη μέθοδο ελέγχου .
14. Μπάρα επιλογής των επιθυμητών στροφών (Σ.Α.Λ) ή PWM Duty Cycle (%)
15. Κουμπί άμεσου τερματισμού της λειτουργίας της Philips .
16. Πίνακας αναλογικής εμφάνισης των τιμών των PWM Duty Cycle ,Ωσης, Τάσης , και Έντασης ρεύματος .
17. Κουμπί εμφάνισης φόρμας παραμέτρων σειριακής σύνδεσης και εκκίνησης της επικοινωνίας με την κάρτα Philips LPC 2106.
18. Κουμπί Λήξης σειριακής επικοινωνίας και αποδέσμευσης της σειριακής θύρας .
19. Κουμπί Τερματισμού εφαρμογής :«Κλίνη Ωσης και Ελέγχου Στροφών» .
20. Κουμπί εμφάνισης και απόκρυψης Γραφημάτων .
21. Εμφάνιση κατάστασης της σύνδεσης και επικοινωνίας με την κάρτα Philips .
22. Εμφάνιση του Χρόνου επικοινωνίας από τη στιγμή της σύνδεσης
23. Γραφική απεικόνιση κατάστασης κάρτας και εφαρμογής .

Εκκίνηση καταγραφή και λήξη επικοινωνίας.

Με την εκκίνηση της εφαρμογής όλα τα κουμπιά και οι λειτουργίες είναι ενεργοποιημένες εκτός από το κουμπί Εμφάνισης Γραφημάτων εφόσον δεν έχουν παραληφθεί δεδομένα και δεν έχει νόημα η ζήτηση γραφήματος. Ο λόγος της ενεργοποίησης όλων των επιλογών και ρυθμίσεων είναι ο χρήστης να τροποποιήσει της ρυθμίσεις της εφαρμογής οι οποίες πραγματοποιήθηκαν σύμφωνα με το αρχείο ρυθμίσεων αν δεν επιθυμεί κάποια από αυτές προσωρινά. Όταν πρόκειται για αλλαγές που θα χρησιμοποιηθούν αρκετές φορές προτιμότερο είναι να γίνεται η δημιουργία ενός νέου αρχείου ρυθμίσεων και η αποθήκευση του εκεί.


Η επικοινωνία της εφαρμογής και της LPC2106 πραγματοποιείται μόνο όταν εκκινηθεί η σειριακή επικοινωνία. Μέχρι τότε το πρόγραμμα αγνοεί όλα τα εισερχόμενα δεδομένα στη σειριακή θύρα και δεν ασκεί κανενός είδους έλεγχο στην LPC2106. Η εκκίνηση της σειριακής επικοινωνίας γίνεται είτε από τον κατάλογο «Σύνδεση» στο μενού της φόρμας είτε απευθείας από το κουμπί «**Εκκίνηση Καταγραφής**». Η λειτουργία και των δύο επιλογών είναι η ίδια εμφανίζοντας το διάλογο αποδοχής ή τροποποίησης των ρυθμίσεων της σειριακής θύρας του υπολογιστή που θα χρησιμοποιηθούν. Με την αποδοχή των ρυθμίσεων εκκινείτε η χρήση της θύρας από την εφαρμογή και τίθεται σε λειτουργία ο αλγόριθμος επικοινωνίας με την LPC2106 με τους κανόνες που έχουν οριστεί στα πρωτόκολλα επικοινωνία στο τέταρτο κεφάλαιο. Κάτω αριστερά στη φόρμα εμφανίζεται η τρέχουσα κατάσταση της σύνδεσης. Αμέσως μόλις ληφθούν τα δεδομένα από τον υπολογιστή εμφανίζονται τα δεδομένα στα κουτιά κειμένου 6 έως 11 .Ενώ ταυτόχρονα ενεργοποιείται το κουμπί εμφάνισης γραφημάτων.



Η λήξη της καταγραφής ισοδυναμεί τον τερματισμό της σειριακής σύνδεσης. Για λόγους ασφαλούς λειτουργία της εγκατάστασης η σειριακή επικοινωνία τερματίζεται μόνο όταν η LPC2106 στείλει την κατάλληλη εντολή ασφαλούς τερματισμού σε αντίθετη περίπτωση η εφαρμογή στέλνει επανειλημμένα την εντολή τερματισμού της LPC2106. Ο τερματισμός γίνεται αντιληπτός από την ενεργοποίηση του κουμπιού **Εκκίνηση Καταγραφής**. Σε περίπτωση που κατά τη διαδικασία τερματισμού της LPC2106 δεν τερματίσει με τον αναμενόμενο τρόπο τότε εμφανίζεται μήνυμα που πληροφορεί το χρήστη για το σφάλμα ούτως ώστε να προβεί σε επανεκκίνηση της κάρτας από το κουμπί «reset» ή διακοπή της τροφοδοσίας της. Μέχρι να αποκατασταθεί ο σωστός τερματισμός της κάρτας το κουμπί εκκίνηση καταγραφής παραμένει απενεργοποιημένο.

Τερματισμός εφαρμογής.

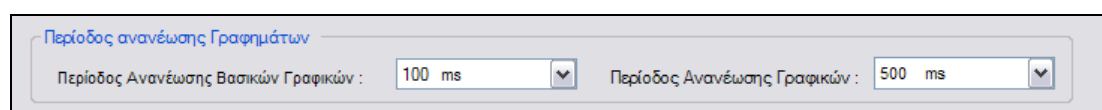
Ο τερματισμός της εφαρμογής δεν προϋποθέτει τη λήξη της σειριακής σύνδεσης ή το σωστό τερματισμό της κάρτας. Οι τους κανόνες σωστού προγραμματισμού ορίζουν ότι η εφαρμογή πρέπει να είναι σε θέση να τερματιστεί από το χρήστη ή το λειτουργικό ανεξάρτητα της κατάστασης της. Χαρακτηριστικές αρνητικές περιπτώσεις είναι προγράμματα που δεν ανταποκρίνονται στις εντολές του χρήστη με αποτέλεσμα υπολογιστής να είναι αδύνατο να λειτουργήσει σωστά. Για λόγους όμως ασφαλούς λειτουργίας κατά τη λήξη της εφαρμογής έχει προστεθεί μία διαδικασία καθυστέρησης του τερματισμού της εφαρμογής, όπου ανιχνεύεται αν η σειριακή θύρα είναι ενεργή και στέλνεται επανειλημμένως η εντολή τερματισμού της κάρτας ώστε να δοθεί το κατάλληλο χρονικό διάστημα να λάβει την εντολή. Στα πλαίσια της ορθής λειτουργίας της εφαρμογής ο τερματισμός πρέπει να αποφεύγεται όσο η επικοινωνία με την κάρτα είναι ενεργή για να μην προκληθούν απρόβλεπτες καταστάσεις.

Ο τερματισμός πραγματοποιείται με τρεις τρόπους πρώτος με το κουμπί «τερματισμού» δεύτερος με το κουμπί κατάργησης της φόρμας ,χαρακτηριστικό  κουμπί όλων των εφαρμογών των windows ,και τρίτος με την επιλογή «Κλείσιμο οθόνης» στον κατάλογο «Σύνδεση» του μενού της φόρμας.

Ρύθμιση ιδιοτήτων γραφημάτων.

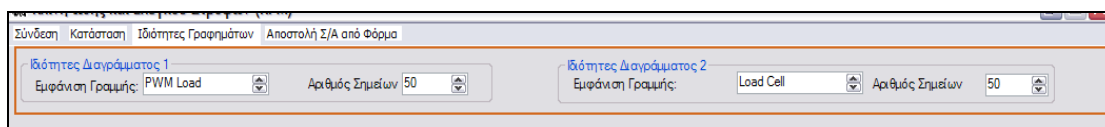
Η ρύθμιση των ιδιοτήτων των γραφημάτων πραγματοποιείται από τον κατάλογο «**Ιδιότητες γραφημάτων**» στο μενού της εφαρμογής, επιλέγοντας την εμφάνιση του πίνακα ρυθμίσεων που περιέχει τις ιδιότητες, του υπό ρύθμιση γραφήματος. Οι ρυθμίσεις που μπορούν να πραγματοποιηθούν είναι :

1. Επιλογή του ρυθμού ανανέωσης των γραφικών παραστάσεων και της ομάδας βασικών γραφικών , στροφόμετρο και κάθετες μπάρες εμφάνισης των τιμών PWM, Load cell, Volt και Ampere.



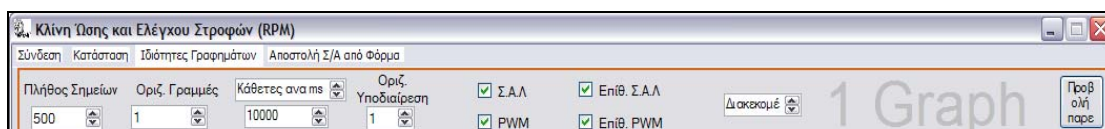
Σχήμα 5.21) Πίνακας ρύθμισης του ρυθμού ανανέωσης των γραφικών επί της οθόνης της κλίνης ώσης .

2. Επιλογή στη χρήση ανάμεσα στο Excel και τη βιβλιοθήκη γραφικών του .Net για τη δημιουργία των γραφικών παραστάσεων.
3. Εμφάνιση του πίνακα ρυθμίσεων των γραφικών παραστάσεων μέσω της χρήσης του Excel. Οι δυνατές ρυθμίσεις είναι επιλογή της εμφανιζόμενης μέτρησης ανά γράφημα και του πλήθους των σημείων για κάθε ένα από αυτά.



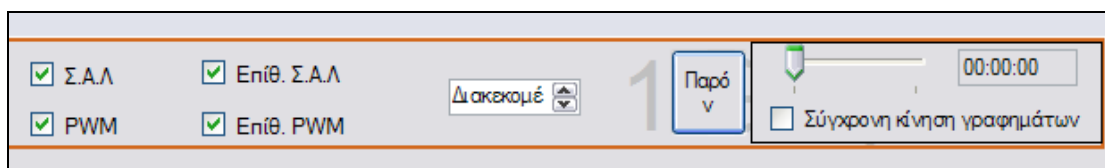
Σχήμα 5.22) Πίνακας ρυθμίσεων των γραφημάτων μέσω Excel.

4. Η χρήση της βιβλιοθήκης γραφικών του «.NET» για τη δημιουργία των γραφημάτων υποστηρίζει ένα μεγάλο αριθμό ρυθμίσεων. Οι οποίες παρουσιάζονται αμέσως παρακάτω.



Σχήμα 5.23) Πίνακας ρυθμίσεων των γραφημάτων μέσω της βιβλιοθήκης του .NET .

- Αναλυτικότερα ρυθμίζεται το πλήθος των τιμών από τα οποία σχεδιάζεται η γραφική παράσταση. Ξεκινώντας τη σχεδίαση από τις νεότερες και πηγαίνοντας προς τις παλαιότερες μέχρι να συμπληρωθεί ο απαιτούμενος αριθμός.
- Ορίζεται το πλήθος των οριζόντιων βοηθητικών γραμμών.
- Οι κάθετες βοηθητικές γραμμές μπορεί να είναι είτε σταθερές στο διάγραμμα είτε ή θέση τους να είναι μεταβλητή και να εξαρτάται από την τιμή του χρόνου ορίζοντας την περίοδο σχεδίασης τους. Η θέση της εμφάνισής τους ορίζεται από τη σχέση $t = n * T$, όπου T η περίοδος σχεδίασης τους και n φυσικός ακέραιος. Δηλαδή οι κάθετες αυτές ορίζουν τα σημεία της γραφικής παράστασης στα οποία η τιμή του χρόνου είναι πολλαπλάσια της ορισμένης περιόδου.
- Οι βοηθητικές γραμμές επί των παραστάσεων μπορεί να είναι είτε διακεκομμένες είτε συνεχής.
- Μπορούν σε κάθε γράφημα να σχεδιαστούν μέχρι τέσσερα δεδομένα τα οποία είναι καθορισμένα για κάθε γραφική παράσταση.
- Τέλος πατώντας το κουμπί «**προβολή παρελθόν**» στον πίνακα ρυθμίσεων εμφανίζεται ο πίνακας σχεδίασης των γραφημάτων από τα εισερχόμενα δεδομένα σε παλαιότερες στιγμές.



Σχήμα 5.24) Πίνακας ελέγχου εμφάνισης των γραφημάτων των δεδομένων σε παλαιότερες στιγμές .

Εμφάνιση γραφημάτων.

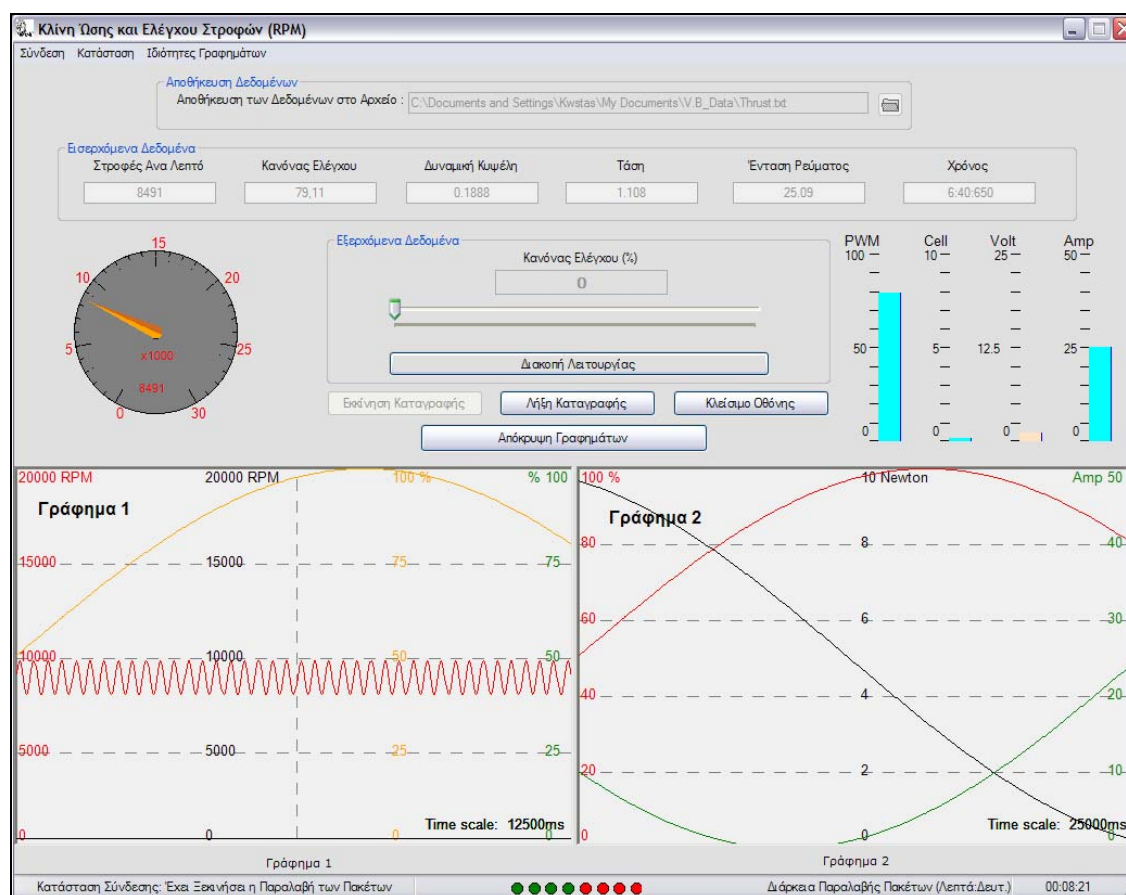
Όταν επιλέγεται η εμφάνιση των γραφημάτων η φόρμα της εφαρμογής κλίνει «Ωσης και Ελέγχου στροφών» αυξάνει αυτόματα το μέγεθός της ώστε να εμφανιστούν οι περιοχές σχεδίασης των γραφικών παραστάσεων. Αντίθετα όταν δεν είναι ενεργοποιημένη η επιλογή της εμφάνισης τους, επανέρχεται αυτόματα στο αρχικό της μέγεθος.

Η ενεργοποίηση των γραφημάτων πραγματοποιείται με δύο τρόπους :

1. Επιλογή του κουμπιού «Εμφάνισης γραφημάτων» επί της φόρμας
2. Επιλογή «Εμφάνιση γραφημάτων» στον κατάλογο «Ιδιότητες γραφημάτων»

Η απενεργοποίηση πραγματοποιείται με όμοιο τρόπο είτε από το κουμπί απόκρυψης γραφημάτων είτε από την επιλογή της απόκρυψης στον κατάλογο ιδιότητες του μενού.

Παρακάτω δίνεται ένα στιγμιότυπο των εμφανιζόμενων γραφικών παραστάσεων.

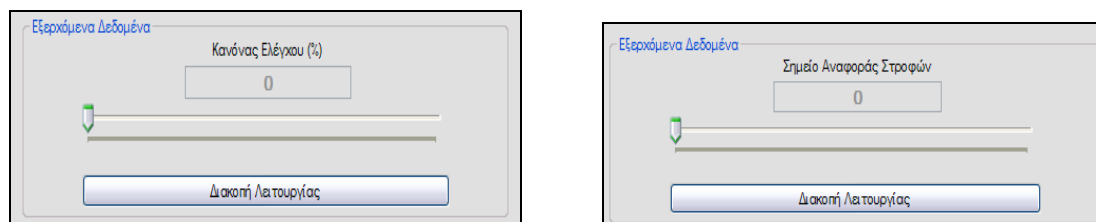


Σχήμα 5.25) Πλήρης οθόνη της εφαρμογής Κλίση ώσης και Ελέγχου στροφών στην οποία εμφανίζονται και τα γραφήματα των εισερχόμενων δεδομένων .

5.18) Αλλαγή κατάστασης λειτουργίας της Κλίνης Ώσης και της LPC2106.

Μια από τις απαιτήσεις που έπρεπε να πληροί η εφαρμογή ήταν η δυνατότητα αλλαγής ανάμεσα στη λειτουργία του «ελέγχου στροφών» και του «κανόνα ελέγχου» κατά τη λειτουργία χωρίς την ανάγκη τερματισμού πρώτα της εγκατάστασης ή τη χρήση διαφορετικών προγραμμάτων για τον υπολογιστή και την LPC2106. Η δυνατότητα αυτή επιτεύχθηκε με την εισαγωγή των εντολών αλλαγής κατάστασης όπως ορίστηκαν στο κεφάλαιο 4.4 .Για να πραγματοποιηθεί η αλλαγή της κατάστασης λειτουργία της εγκατάστασης, ως σύνολο απαιτείται πρώτα να αλλάξει η κατάσταση στη φόρμα Κλίνη Ώσης και, στη συνέχεια στέλνεται μέσω της σειριακή θύρας η κατάλληλη εντολή αλλαγής κατάστασης στην κάρτα.

Η επιλογή της επιθυμητής κατάστασης λειτουργίας γίνεται από τον κατάλογο «Κατάσταση» του μενού της φόρμας. Τονίζεται ότι κατά την εκκίνηση πάντα η κατάσταση εκκίνησης είναι ο «Κανόνας Ελέγχου(%)» για λόγους ασφαλείας της εγκατάστασης. Με την αλλαγή της κατάστασης λειτουργίας εμφανίζεται το κατάλληλο χειριστήριο εισαγωγής των τιμών αναφοράς που στέλνονται στην κάρτα ενώ στην επικεφαλίδα του κουτιού εμφάνισης της τρέχουσας τιμής της μπάρας εισαγωγής, αναγράφεται η κατάσταση λειτουργίας.



Σχήμα 5.26) Μπάρες εισαγωγής του σημείου αναφοράς της δράσης ελέγχου και των Στροφών.

5.19) Άμεση διακοπή λειτουργίας Κινητήρα.

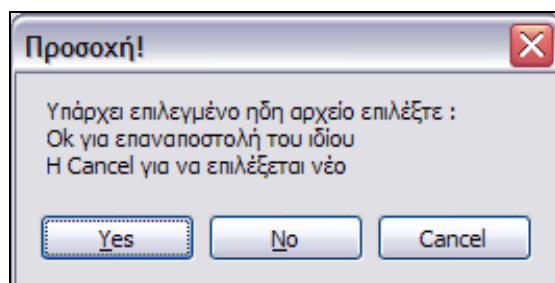
Στα πλαίσια της ορθής λειτουργία της εγκατάστασης υπολογίστηκε και η περίπτωση της πρόκλησης κάποιας απρόβλεπτης κατάστασης και της αναγκαιότητας του άμεσου τερματισμού του κινητήρα. Λόγω της πιθανής σπουδαιότητας του γεγονότος αυτού η εντολή τερματισμού του κινητήρα έχει το σπουδαιότερο ρόλο στην επικοινωνία και για αυτό το λόγο όταν πρόκειται να αποσταλεί διαγράφονται όλα τα δεδομένα στη μνήμη της σειριακής και στη συνέχεια γράφεται στην σειριακή ώστε να σταλεί άμεσα.

Η άμεση διακοπή του κινητήρα γίνεται από την επιλογή του κουμπιού «**Διακοπή λειτουργίας**» της φόρμας. Η διακοπή λειτουργία αφορά τον κινητήρα και όχι την επικοινωνία ή τη λειτουργία της εγκατάστασης. Δίνεται η επιλογή επανεκκίνησης του κινητήρα πατώντας το κουμπί «**Εκκίνησης**» που εμφανίζεται στη θέση του κουμπιού διακοπής. Κατά τη διάρκεια που ισχύει η διακοπή λειτουργίας του κινητήρα το κουμπί «**Εκκίνησης**» αλλάζει περιοδικά το χρώμα του ώστε να τονιστεί η λειτουργία αυτή.

5.20) Αποστολή τιμών αναφοράς από αρχείο.

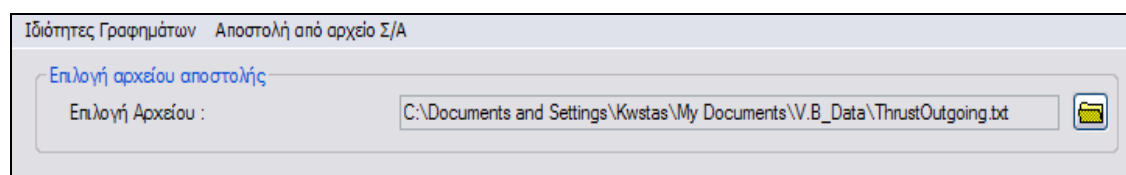
Η αποστολή των τιμών αναφοράς από αρχείο πραγματοποιείται με βάση τον αλγόριθμό όπως παρουσιάστηκε στο κεφάλαιο 4.7. Όμως για τη φόρμα κλίνη ώσης απαιτούνται να πραγματοποιηθούν και ορισμένες ενέργειες από το χρήστη για την ενεργοποίηση της αποστολής. Σημειώνεται ότι η διαδικασία σχεδιάστηκε με γνώμονα την ενεργοποίηση και απενεργοποίηση της κατά τη διάρκεια λειτουργίας χωρίς την απαίτηση διακοπής λειτουργίας της εγκατάστασης. Κάθε φορά που απενεργοποιείται η αποστολή από αρχείο η φόρμα επανέρχεται στην κατάσταση «**Αποστολής τιμών από τη μπάρα εισαγωγής**» και για λόγους ασφαλείας τίθεται ως μέθοδος λειτουργίας ο «**κανόνας ελέγχου (%)**». Επιπλέον δεν τίθεται κάποιος περιορισμός για την απενεργοποίηση της αποστολής του αρχείου. Είναι σχεδιασμένο ώστε να μπορεί να διακοπεί οποιαδήποτε στιγμή επιλεγεί και μπορεί να ενεργοποιηθεί οποιαδήποτε στιγμή. Το επιλεγμένο αρχείο προς αποστολή μπορεί να αποσταλεί όσες φορές επιθυμείται δεν υπάρχει κάποιος περιορισμός. Κάθε φορά που εκκινείτε η αποστολή αρχείου ξεκινά η αποστολή από την αρχή του αρχείου δηλαδή δεν έχει οριστεί κάποιου είδους μνήμη της τελευταία θέσης ανάγνωσης του αρχείου.

Κατά την εκκίνηση της φόρμας «**Κλίνη ώσης**» διαβάζεται από το αρχείο ρυθμίσεων η διεύθυνση του αρχείου αποστολής τιμών αναφοράς και τίθεται το αρχείο αυτό ως το επιλεγμένο αρχείο αποστολής στον αντίστοιχο πίνακα. Η εμφάνιση του παραπάνω πίνακα γίνεται με την επιλογή «**Αποστολή Σ/Α από αρχείο**» στο μενού της φόρμας. Ταυτόχρονα με την εμφάνιση του πίνακα, εφόσον είναι ήδη επιλεγμένο κάποιο αρχείο, εμφανίζεται ο διάλογος επιβεβαίωσης αποστολής του αρχείου και πατώντας αποδοχή «**Yes**» εκκινείτε η αποστολή του αρχείου.



Σχήμα 5.27) Οθόνη μηνύματος επιβεβαίωσης για την αποστολή του αρχείου αποστολής .

Αν επιθυμείται η αποστολή διαφορετικού αρχείου τότε πατώντας το πλήκτρο ακύρωσης «Cancel» παραμένει εμφανής ο πίνακας επιλογής αρχείου αποστολής και επιλέγοντας το κουμπί του διαλόγου ανοίγματος αρχείων των windows ορίζεται το νέο αρχείο προς αποστολή. Αν η επιλογή είναι σωστή τότε τίθεται ως επιλεγμένο αρχείο αποστολής και στη συνέχεια με επανεπιλογή του «**Αποστολή Σ/Α από αρχείο**» και επιλέγοντας «**Yes**» στο παραπάνω μήνυμα αρχίζει αμέσως η αποστολή. Πατώντας το πλήκτρο «**No**» εξαφανίζεται η πίνακας επιλογής αρχείου.



Σχήμα 5.28) Οθόνη εμφάνισης της διεύθυνσης του επιλεγμένου αρχείου αποστολής και πίνακας επιλογής του αρχείου προς αποστολή.

Τέλος σημειώνεται ότι όσο διαρκεί η αποστολή των τιμών από το αρχείου εμφανίζεται στο τίτλο της φόρμας το πρόσθετο λεκτικό «Αποστέλλονται ΣΛΑ από αρχείο» ενώ όταν ολοκληρωθεί εμφανίζεται το λεκτικό «Τέλος αρχείου» .

5.21) Διαμόρφωση του αρχείου αποθήκευσης των εισερχόμενων δεδομένων από την LPC2106.

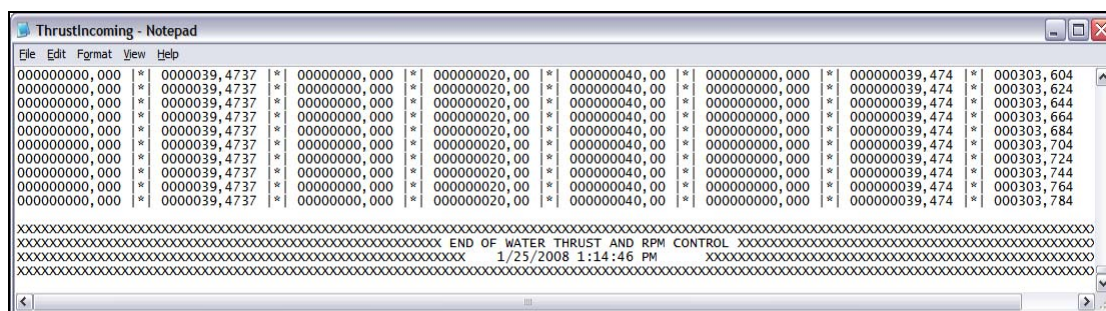
Το αρχείο αποθήκευσης είναι τύπου κειμένου «.txt» .Κατά το άνοιγμα του από την εφαρμογή της «κλίνης ώσης και ελέγχου στροφών» για την αποθήκευση των εισερχόμενων μετρήσεων εγγράφονται ο τίτλος της εφαρμογής καθώς και η ημερομηνία και η ώρα. Ο λόγος είναι για το διαχωρισμό των δεδομένων από παλαιότερες η νεότερες εισαγωγές δεδομένων. Τονίζεται ότι σε κάθε εκκίνηση της εφαρμογής δε διαγράφονται τα περιεχόμενα δεδομένα στο αρχείο αλλά συνεχίζεται από την τελευταία εγγεγραμμένη θέση στο αρχείο. Ιδιαίτερη προσοχή πρέπει να δίνεται στον καθαρισμό του αρχείου από τα δεδομένα πριν από κάθε εκκίνηση της εφαρμογής ώστε να μην περιέχει σημαντικό ποσό δεδομένων. Η συνεχής χρήση χωρίς διαγραφή των παλαιότερων δεδομένων θα οδηγήσει στην αύξηση του μεγέθους του αρχείου ώστε να ξεπεράσει τα όρια που υπάρχουν για κάθε λειτουργικό σύστημα.

Τα δεδομένα αποθηκεύονται κατά στήλες , όπου κάθε στήλη αντιπροσωπεύει ένα και μόνο δεδομένο. Κάθε στήλη διαχωρίζεται από τη γραμματοσειρά « |*| » . Σε κάθε γραμμή αποθηκεύονται τα εισερχόμενα δεδομένα ανά πακέτο από την κάρτα LPC2106 . Στο αρχείο αποθηκεύονται οι εισερχόμενες τιμές της μέτρησης των στροφών , της δράσης ελέγχου σε «%» ,της μετρούμενης ώσης ,της μετρούμενης τάσης ,της μετρούμενης έντασης ρεύματος του κινητήρα , το τρέχων σημείο αναφοράς των στροφών και το τρέχων σημείο αναφοράς της δράσης ελέγχου «PWM» και στην τελευταία στήλη η τιμή του χρόνου σε ms της δειγματοληψίας από την LPC2106. Όλα τα παραπάνω δεδομένα αποθηκεύονται με τη χρήση τυποποίησης τριών δεκαδικών πλην της τρέχουσας δράσης ελέγχου για την οποία χρησιμοποιούνται τέσσερα.

RPM	* Control Reg	* Loadcell	* Voltage	* Amperes (A)	* S.P. RPM	* S.P. Con Reg	* Time
000000000,000	* 0000000,0000	* 00000000,000	* 000000020,00	* 000000040,00	* 000000000,000	* 000000000,000	* 000000,020
000002999,000	* 0000000,0000	* 00000000,000	* 000000020,00	* 000000040,00	* 000000000,000	* 000000000,000	* 000000,040
000002999,000	* 0000000,0000	* 00000000,000	* 000000020,00	* 000000040,00	* 000000000,000	* 000000000,000	* 000000,060
000002999,000	* 0000000,0000	* 00000000,000	* 000000020,00	* 000000040,00	* 000000000,000	* 000000000,000	* 000000,080
000003999,000	* 0000000,0000	* 00000000,000	* 000000020,00	* 000000040,00	* 000000000,000	* 000000000,000	* 000000,100
000003000,000	* 0000000,0000	* 00000000,000	* 000000020,00	* 000000040,00	* 000000000,000	* 000000000,000	* 000000,120
000003000,000	* 0000000,0000	* 00000000,000	* 000000020,00	* 000000040,00	* 000000000,000	* 000000000,000	* 000000,140
000003000,000	* 0000000,0000	* 00000000,000	* 000000020,00	* 000000040,00	* 000000000,000	* 000000000,000	* 000000,160
000003000,000	* 0000000,0000	* 00000000,000	* 000000020,00	* 000000040,00	* 000000000,000	* 000000000,000	* 000000,180
000003000,000	* 0000000,0000	* 00000000,000	* 000000020,00	* 000000040,00	* 000000000,000	* 000000000,000	* 000000,200
000003000,000	* 0000000,0000	* 00000000,000	* 000000020,00	* 000000040,00	* 000000000,000	* 000000000,000	* 000000,220
000003000,000	* 0000000,0000	* 00000000,000	* 000000020,00	* 000000040,00	* 000000000,000	* 000000000,000	* 000000,240
000003000,000	* 0000000,0000	* 00000000,000	* 000000020,00	* 000000040,00	* 000000000,000	* 000000000,000	* 000000,260
000003000,000	* 0000000,0000	* 00000000,000	* 000000020,00	* 000000040,00	* 000000000,000	* 000000000,000	* 000000,280
000003000,000	* 0000000,0000	* 00000000,000	* 000000020,00	* 000000040,00	* 000000000,000	* 000000000,000	* 000000,300
000003000,000	* 0000000,0000	* 00000000,000	* 000000020,00	* 000000040,00	* 000000000,000	* 000000000,000	* 000000,320

Σχήμα 5.29) Στιγμιότυπο του αρχείου αποθήκευσης των εισερχόμενων δεδομένων κατά τις πρώτες εγγραφές.

Κατά τη λήξη της εγγραφής στο αρχείο μετά την τελευταία εγγραφή καταγράφεται ο τίτλος της εφαρμογής με την ένδειξη τέλος καθώς και η ημερομηνία και ώρα λήξης της καταγραφής.

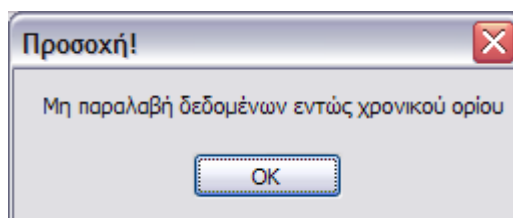


Σχήμα 5.30) Στιγμιότυπο του αρχείου αποθήκευσης των εισερχόμενων δεδομένων κατά τη λήξη της εγγραφής.

5.22) Κρίσιμα μηνύματα της εφαρμογής Κλίνης Ώσης και ο χειρισμός τους.

Κατά τη λειτουργία της εφαρμογής αυτής θα εμφανιστούν αρκετά μηνύματα για την πληροφόρηση του χρήστη είτε για καταστάσεις λάθους της εφαρμογής είτε εσφαλμένης χρήσης είτε απλά για την ενημέρωση του χρήστη. Τα σημαντικότερα από αυτά και η προτεινόμενη αντίδραση του χρήστη παρουσιάζεται στις παρακάτω παραγράφους.

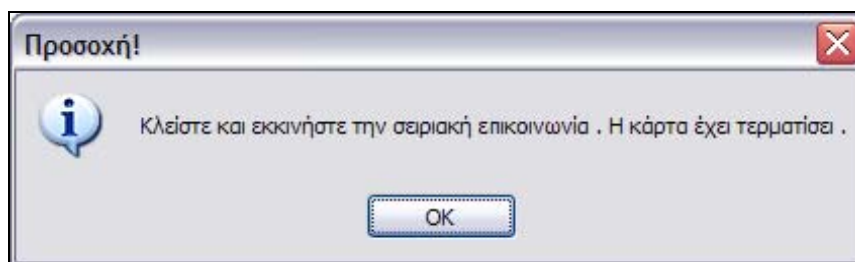
Όταν είναι ενεργή η σύνδεση με τη σειριακή θύρα με την LPC2106 και για κάποιο λόγο διακοπεί ή καθυστερήσει η λήψη νέων δεδομένων τότε εμφανίζεται το παρακάτω μήνυμα με σκοπό την ενημέρωση του χρήστη.



Σχήμα 5.31) Οθόνη μηνύματος μη παραλαβής δεδομένων εντός του χρονικού περιθωρίου.

Προτεινόμενη αντίδραση είναι η επιλογή «OK» ώστε, να ενημερωθεί η εφαρμογή ότι ο χρήστης έχει δει το μήνυμα και να δοθεί παράταση 5 δευτερολέπτων μέχρι τη νέα εμφάνιση του μηνύματος εφόσον δεν έχει αποκατασταθεί η επικοινωνία. Ο χρήστης αν δεν θέλει να του εμφανίζεται το μήνυμα συνέχεια όσο το πρόβλημα παραμένει να **μην** το κλείσει. Όσο είναι το μήνυμα αυτό είναι ενεργό αποκλείεται η πρόσβαση στις λειτουργίες και τα κουμπιά της φόρμα «Κλίνη Ώσης». Δε συνιστάτε να κλείσει το μήνυμα από το κόκκινο κουμπί κατάργησης του μηνύματος διότι απλά θα επανεμφανιστεί αμέσως. Η συνήθης αιτία του προβλήματος είναι η κακή τοποθέτηση του σειριακού καλωδίου ή η αποσύνδεση αυτού.

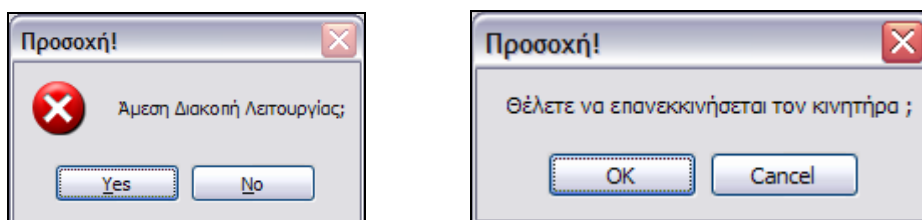
Για λόγους προστασίας και ασφάλειας της εγκατάστασης κατά την προσωρινή διακοπή της επικοινωνίας τίθεται το χρονικό περιθώριο 10 δευτερολέπτων ώστε να αποκατασταθεί η σύνδεση. Αν η επικοινωνία αποκατασταθεί τότε η λειτουργία της εφαρμογής και της εγκατάστασης επανέρχεται στην κανονική λειτουργία. Αν **όχι** τότε η κάρτα θα εκτελέσει τη διαδικασία επανεκκίνησης της με την οποία έρχεται στην αρχική της κατάσταση στην οποία είναι ενεργοποιημένη μόνο η σειριακή επικοινωνία. Σε περίπτωση που αποκατασταθεί η σύνδεση περάν του ορίου των 10 sec και η κάρτα έχει επανεκκίνηση εμφανίζεται το παρακάτω μήνυμα.



Σχήμα 5.32) Οθόνη μηνύματος τερματισμού της επικοινωνίας με την κάρτα LPC2106.

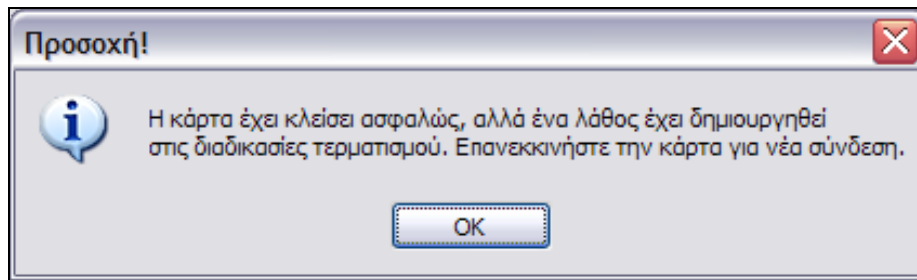
Με την επιλογή «OK» αυτόματα τερματίζεται η σειριακή σύνδεση της εφαρμογής «Κλίνης ώσης» ώστε να επανέλθει στην αρχική της κατάσταση. Για την επανεκκίνηση της σύνδεση με την LPC2106 και του ελέγχου της εγκατάστασης, απλά πραγματοποιήστε νέα σύνδεση όπως παρουσιάστηκε στο κεφάλαιο «**Εκκίνηση καταγραφής**».

Για την αποφυγή κατά λάθος επιλογής του κουμπιού άμεση διακοπής λειτουργίας του κινητήρα εμφανίζεται το μήνυμα επιβεβαίωσης όπου επιλέγοντας «**OK**» ενεργοποιείται η διαδικασία άμεσης διακοπής .Τα ίδια ισχύουν και για την περίπτωση της επανεκκίνησης του κινητήρα.



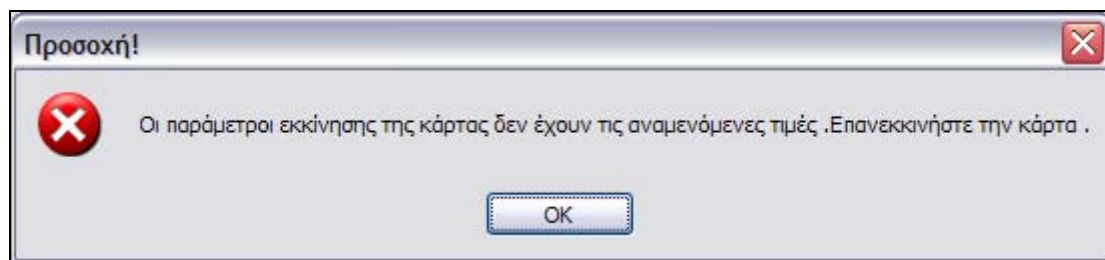
Σχήμα 5.33) Οθόνες μηνυμάτων διακοπής λειτουργίας και επανεκκίνησης του κινητήρα.

Αν κατά την εκτέλεση της διαδικασία επανεκκίνησης της κάρτας παρουσιασθεί κάποιο σφάλμα τότε εμφανίζεται το παρακάτω μήνυμα. Η προτεινόμενη ενέργεια είναι η επανεκκίνηση της κάρτας είτε από το κουμπί reset είτε με τη διακοπή της τροφοδοσία. Ομοίως με παραπάνω όταν εκκινήσει η κάρτα η σύνδεση με τη σειριακή θύρα διακόπτεται ώστε να έρθει και η εφαρμογή στην αρχική της κατάσταση.



Σχήμα 5.34) Οθόνη μηνύματος εσφαλμένου τερματισμού της κάρτας LPC2106.

Αν κατά την εκκίνηση της κάρτας και ενεργοποίησης των λειτουργιών της παρουσιασθεί σφάλμα εμφανίζεται το παρακάτω μήνυμα .Για να εξαφανιστεί το μήνυμα θα πρέπει να εκκινήσετε την κάρτα όπως παραπάνω «hard reset».



Σχήμα 5.35) Οθόνη μηνύματος εσφαλμένου εκκίνησης της κάρτας LPC2106.

6) Θεωρία και Υπολογισμός της Ροπής Αδρανείας του συστήματος πρόωσης.

6.0) Εισαγωγή

Η αδράνεια της εγκατάστασης «water jet» είναι ίση με το άθροισμα της αδρανείας του κινητήρα, της έλικας, του άξονα της και του ύδατος που αυτή συμπαρασύρει κατά την κίνησή της. Η αδράνεια είναι ένα μέγεθος για το οποίο δύσκολα βρίσκονται στοιχεία στη βιβλιογραφία διότι σπάνια υπολογίζεται. Μάλιστα για μια συγκεκριμένη προωστήρια εγκατάσταση η αδράνεια μεταβάλλεται με το χρόνο διότι η ποσότητα του συμπαρασυρόμενου ύδατος από την έλικα δεν είναι σταθερή και επιπλέον ο κινητήρας έχει κινούμενα μέρη. Για την εύρεση της αδρανείας θα εφαρμοστούν δύο βασικές μέθοδοι και παραλλαγές αυτών με σκοπό την εύρεση της βέλτιστης μεθόδου. Δηλαδή της μεθόδου που θα δώσει την ακριβέστερη μέτρηση με βάση τις δυνατότητες των διατιθέμενων πειραματικών διατάξεων του εργαστηρίου.

6.1) Υπολογισμός της ροπής αδρανείας με τη χρήση της αρχής διατήρησης της ορμής.

Γενικά η συνήθης μέθοδος υπολογισμού της αδρανείας είναι η χρήση της αρχής διατήρησης της ορμής σε ένα επιταχυνόμενο ή επιβραδυνόμενο σύστημα. Στα πλοία ο υπολογισμός της αδρανείας εφόσον είναι γνωστά όλα τα κατασκευαστικά στοιχεία τόσο της μηχανής όσο της έλικας και του άξονα μπορεί να επιτευχθεί απευθείας από τα γεωμετρικά στοιχεία του συνολικού συστήματος. Ο μόνος άγνωστος είναι η αδράνεια του συμπαρασυρόμενου ύδατος από την έλικα. Από τα στοιχεία του κινητήρα μπορεί να υπολογιστεί η προσδιδόμενη ροπή στην έλικα και από την καμπύλη της έλικας η απορροφούμενη απ' αυτή ροπή. Η διαφορά των ροπών προς την αδράνεια ισούται με την επιτάχυνση ή την επιβράδυνση της προωστήριας εγκατάστασης του πλοίου.

$$\dot{N} = \frac{Q}{J} \quad (6.1)$$

όπου Q η συνισταμένη των εφαρμοζόμενων ροπών της μηχανής και της έλικας και J η ροπή αδρανείας της εγκατάστασης.

Ολοκληρώνοντας την εξίσωση 6.1 για το χρονικό διάστημα από t_0 έως t_1 προκύπτει :

$$\int_{t_0}^{t_1} \dot{N} dt = \int_{t_0}^{t_1} \frac{Q}{J} dt \Rightarrow N(t_1) - N(t_0) = \frac{1}{J} \cdot \int_{t_0}^{t_1} Q(t) dt \Rightarrow$$
$$J = \frac{\int_{t_0}^{t_1} \Delta Q(t) dt}{N(t) - N(t_0)} \quad (6.2)$$

Στην παραπάνω εξίσωση όλα τα μεγέθη είναι υπολογίσιμα και επομένως μπορεί να προσδιοριστεί το μέτρο της αδρανείας. Στο παραπάνω πρόβλημα προς απλούστευση των παραμέτρων έχει υποτεθεί η ακριβής γνώση της απορροφούμενης και προσδιδόμενης ροπής η οποία στην πράξη είναι αρκετά δύσκολη. Για τους

παραπάνω λόγους προτείνεται η παρακάτω μέθοδος υπολογισμού της ροπής αδρανείας .

6.2) Υπολογισμός ροπής αδρανείας συστήματος πρόωσης με τη μέθοδο της μεταβαλλόμενης ροπής !

Η ροπή αδρανείας εκφράζει μια φυσική ποσότητα η οποία αντιστέκεται στις μεταβολές τις γωνιακής ταχύτητας .Έχει μονάδες ροπής και είναι ουσιαστικά ένας συντελεστής αναλογίας που συνδέει τη μεταβολή της γωνιακής ταχύτητας ενός σώματος με το αλγεβρικό άθροισμα των ροπών των ασκούμενων δυνάμεων επί αυτού

$$.Η μαθηματική διατύπωση είναι \quad J_o \cdot \dot{\omega} = \sum_{i=0}^n T_i \quad (6.3)$$

όπου J_o η ροπή αδρανείας , $\dot{\omega}$ η γωνιακή επιτάχυνση και T_i οι ροπές των ασκούμενων δυνάμεων .

Η σχέση (6.3) έχει μοναδική λύση, διάφορη της μηδενικής μόνο όταν: $\dot{\omega} \neq 0$ &

$$\sum_{i=0}^n T_i \neq 0 \quad . \text{ Άρα η λύση είναι } J_o = \frac{\sum_{i=0}^n T_i}{\dot{\omega}} \quad , \quad (6.4)$$

Δηλαδή όταν είναι γνωστή η στιγμιαία γωνιακή επιτάχυνση και στιγμιαία συνολική ροπή των ασκούμενων δυνάμεων , η ροπή αδρανείας J_o μπορεί να υπολογιστεί απευθείας από τον τύπο της σχέσης 6.4.

Η γνώση των στιγμιαίων τιμών των ροπών και της γωνιακής επιτάχυνσης είναι πρακτικά αδύνατες εφόσον οι μετρήσεις πραγματοποιούνται στο διακριτό χρόνο .Η σχέση (6.1) γράφεται ως :

$$J_o \cdot \dot{\omega}(t) = \sum_{i=0}^n T_i(t) \quad (6.5)$$

Με ολοκλήρωση της παραπάνω σχέσης στο χρονικό διάστημα από t_1 έως t_2 προκύπτει :

$$\int_{t_1}^{t_2} J_o \cdot \dot{\omega}(t) dt = \int_{t_1}^{t_2} \sum_{i=0}^n T_i(t) dt \Leftrightarrow J_o \cdot \int_{t_1}^{t_2} \dot{\omega}(t) dt = \int_{t_1}^{t_2} \sum_{i=0}^n T_i(t) dt \Rightarrow$$

$$J_o = \frac{\int_{t_1}^{t_2} \sum_{i=0}^n T_i(t) dt}{\int_{t_1}^{t_2} \dot{\omega}(t) dt} = \frac{\sum_{i=0}^n \int_{t_1}^{t_2} T_i(t) dt}{\omega(t_2) - \omega(t_1)} \quad (6.6)$$

Από τη σχέση (6.6) προκύπτει η ροπή αδρανείας εφόσον είναι γνωστές οι αναλυτικές εκφράσεις στο πεδίο του συνεχές χρόνου $t \in [t_1, t_2]$ η συνισταμένη ροπή και γωνιακή ταχύτητα. Η αναλυτική όμως έκφραση των ροπών T_i συναρτήσεως μόνο του χρόνου είναι πολύ δύσκολη εξαιτίας της μη δυνατής εφαρμογής γνωστής $\sum T_i(t)$. Η συνισταμένη ροπή αποτελείται από τις επιμέρους ροπές του συστήματος οι οποίες είναι οι $T_{motor}, T_{Load}, T_{friction}$ ροπή κινητήρα, φορτίου και απωλειών αντίστοιχα. Οι ροπές του κινητήρα και των απωλειών είναι γνωστές, προσεγγιστικά βλέπε κεφάλαιο 1, συναρτήσεως της γωνιακής ταχύτητας ω και τάσης ακροδεκτών V_α , του κινητήρα. Δηλαδή $T_{motor}(V_\alpha, \omega), T_{Load}(\omega), T_{friction}(\omega)$ αναλυτική έκφραση των σχέσεων αυτών δίνονται από τις σχέσεις 1.13 και 1.17a ή 1.17b, για τη ροπή του κινητήρα και των απωλειών αντίστοιχα. Άρα εφόσον είναι γνωστή η $\omega(t)$ για $t \in [t_1, t_2]$ οι ροπές γράφονται :

$$T_{motor}(V_\alpha(t), \omega(t)), \quad T_{Load}(\omega(t)), \quad T_{friction}(\omega(t)) \quad (6.7)$$

Πρόκειται δηλαδή για σύνθετες συναρτήσεις των οποίων η εφαρμογή στη σχέση 6.6 θα επιφέρει πρόσθετες δυσκολίες ως προς την επίλυση της. Η $\omega(t)$ πρέπει να προσεγγισθεί από τα αποτελέσματα των μετρήσεων. Λόγω της αδυναμίας μας να εργαστούμε στο πεδίο του χρόνου t θα επιλύσουμε τη σχέση 6.3 απευθείας στο πεδίο της γωνιακής ταχύτητας ω όπου οι ροπές έχουν αναλυτικές εκφράσεις. Η σχέση 6.3 είναι μια διαφορική εξίσωση, της οποίας η μορφή, τάξη και βαθμός εξαρτώνται από την έκφραση των ροπών. Στο σημείο αυτό εξετάζεται η δυνατότητα επίλυσης της σχέσης 6.3 υπό την εφαρμογή φορτίου ή όχι, με σκοπό την εύρεση της βέλτιστης δυνατής λύσης.

6.2.1) Εύρεση της ροπής αδρανείας όταν στο σύστημα ασκείται φορτίο.

Ως γνωστών η ροπή που απαιτείται για την περιστροφή μιας έλικας στο νερό είναι ανάλογη του τετραγώνου της γωνιακής ταχύτητας ω .

$$T_{impeller} = L_{Load} \cdot \omega^2 \quad (6.9)$$

με L_{Load} ορίζεται ως η σταθερά αναλογίας.

Η ροπή των τριβών του αξονικού συστήματος θεωρείται ως γραμμική συνάρτηση του ω , δηλαδή ορίζεται με τον τύπο $T_{friction} = A_f \cdot \omega(t) + B_f$ και η οποία έχει υπολογιστεί στο πρώτο κεφάλαιο. Η αναλυτική έκφραση αυτής δίνεται από τη σχέση 1.17b :

$$T_{friction} = 8 \cdot 10^{-6} \cdot \omega + 0.012 \quad (6.10)$$

Η ροπή του κινητήρα έχει ως βέλτιστη έκφραση τη σχέση 1.13 :

$$T_{motor} = \left(\frac{K_{\tau}}{R_{\alpha}} \right) \cdot V_{\alpha} - \left(\frac{K_e \cdot K_{\tau}}{R_{\alpha}} \right) \cdot \omega$$

Η συνολική ροπής αδράνειας του συζευγμένου συστήματος ,αποτελείται από τις επιμέρους αδράνεις του κινητήρα, προωθητήρα και παρασυρόμενου ύδατος J_{motor} , $J_{waterjet}$, J_{water} .Η ολική ροπή αδράνειας γράφεται :

$$J_o = J_{motor} + J_{waterjet} + J_{water} \quad (6.11)$$

Εφαρμόζοντας την αρχή διατήρησης της στροφορμής, σχέση 6.4, στο συνολικό σύστημα πρόωσης προκύπτει η παρακάτω σχέση :

$$\left(J_{motor} + J_{waterjet} + J_{water} \right) \cdot \dot{\omega} = \sum T = T_{motor} + T_{Load} + T_{friction} \Rightarrow$$

$$J_o \cdot \dot{\omega} = T_{motor} + T_{Load} + T_{friction} \quad (6.12)$$

,όπου T_{motor} η ροπή του κινητήρα , $T_{Load} = T_{impeller}$ η ροπή του φορτίου και $T_{friction}$ η ροπή των απωλειών τριβής των εδράνων κύλισης .Εφαρμόζοντας τις σχέσεις 6.9 , 6.10 , 6.11 στην 6,12 προκύπτει :

$$\left(\frac{K_{\tau}}{R_{\alpha}} \right) \cdot V_{\alpha}(t) - \left(\frac{K_e \cdot K_{\tau}}{R_{\alpha}} \right) \cdot \omega(t) - L_{Load} \cdot \omega(t)^2 - (8 \cdot 10^{-6} \cdot \omega(t) + 0.012) = J_o \cdot \dot{\omega}(t) \Rightarrow$$

$$J_o \cdot \dot{\omega}(t) + L_{Load} \cdot \omega(t)^2 + \left(8 \cdot 10^{-6} + \frac{K_e \cdot K_{\tau}}{R_{\alpha}} \right) \cdot \omega(t) = \left(\frac{K_{\tau}}{R_{\alpha}} \right) \cdot V_{\alpha}(t) - 0.012 \quad (6.13)$$

Η διαφορική εξίσωση 6.13 που προκύπτει είναι 1 τάξης και 2 βαθμού η οποία είναι δύσκολη στην επίλυση της λόγω της παρουσίας ,στο πρώτο μέλος ,του παράγοντα $L_{Load} \cdot \omega(t)^2$. Παύει πλέον το συνολικό σύστημα να είναι γραμμικό.

6.2.2) Εύρεση της ροπής αδρανείας όταν στο σύστημα δεν ασκείται φορτίο.

Επομένως θα εξετάσουμε την περίπτωση κατά την οποία δεν θα εφαρμόζεται φορτίο στο σύστημα .Δηλαδή $T_{Load}=0$. Άρα οι σχέσεις 6.12 & 6.11 γράφονται για την περίπτωση χωρίς φορτίο :

$$J_o = J_{motor} + J_{waterjet}$$

$$J_o \cdot \dot{\omega} = T_{motor} + T_{friction} \quad (6.14)$$

Εφαρμόζοντας τις σχέσεις 6.9 , 6.10 , 6.11 στην 6,14 προκύπτει :

$$\left(\frac{K_{\tau}}{R_{\alpha}}\right) \cdot V_{\alpha}(t) - \left(\frac{K_e \cdot K_{\tau}}{R_{\alpha}}\right) \cdot \omega(t) - \left(8 \cdot 10^{-6} \cdot \omega(t) + 0.012\right) = J_o \cdot \dot{\omega}(t) \Rightarrow$$

$$J_o \cdot \dot{\omega}(t) + \left(8 \cdot 10^{-6} + \frac{K_e \cdot K_{\tau}}{R_{\alpha}}\right) \cdot \omega(t) = \left(\frac{K_{\tau}}{R_{\alpha}}\right) \cdot V_{\alpha}(t) - 0.012 \quad (6.15)$$

Η οποία είναι μια γραμμική διαφορική εξίσωση 1 βαθμού και μπορεί να επιλυθεί με τη χρήση του μετασχηματισμού Laplace .

6.2.3) Εύρεση της ροπής αδρανείας με ημιτονοειδή μεταβολή των στροφών του κινητήρα, όταν στο σύστημα δεν ασκείται φορτίο.

Ως γνωστών «βλέπε Συστήματα αυτομάτου ελέγχου 2003» η απόκριση ενός γραμμικού συστήματος σε αρμονική διέγερση είναι επίσης αρμονική με γνωστό πλάτος ,φάση και συχνότητα τη συχνότητα της διέγερσης. Για την πραγματοποίηση της αρμονικής μεταβολής των στροφών, χρησιμοποιείται ο παλμοτροφοδοτικός μετατροπέας της τάσης σε συνεργασία με την κάρτα LPC2106 .Όπως έχει προαναφερθεί στο πρώτο κεφάλαιο ο ρόλος του είναι η ρύθμιση της εφαρμοζόμενης «τάσης» στον κινητήρα με σκοπό τον τελικό έλεγχο των στροφών λειτουργίας του. Ο ρόλος της LPC2106 είναι να οδηγήσει τον παλμοτροφοδοτικό μετατροπέα με κατάλληλα σήματα ελέγχου στην πραγματοποίηση της ημιτονοειδούς μεταβολής των στροφών του κινητήρα.

Παραδοχές :

- Θεωρείται ότι η τροφοδοσία του συστήματος με τάση και ένταση ρεύματος είναι επαρκείς ώστε να αφαιρεθούν από το σύστημα οι μεταβλητές $e_{\alpha}, I_{τροφ}$. Δηλαδή $e_{\alpha} = const$ και $I_{τροφ} > I_{\alpha}$ σε κάθε χρονική στιγμή .
- Η ρύθμιση της τάσης στους ακροδέκτες του κινητήρα , V_{in} , μπορεί να κυμαίνεται από 0 έως e_{α} volt όπως ορίζεται στο κεφάλαιο 1.7. Επομένως δε μπορούμε να έχουμε απευθείας αρμονική μεταβολή της τάσης ακροδεκτών λόγω του περιορισμού $V_{in} \in [0, e_{\alpha}]$. Η αρμονική μεταβολή της

εφαρμοζόμενης τάσης στον κινητήρα γράφεται ως $\hat{V}_{in} = A \cdot \sin \Omega \cdot t$, με πεδίο τιμών $\hat{V}_{in} \in [-A, A]$, $A \leq e_\alpha$, η οποία είναι μη πραγματοποιήσιμη με τις δυνατότητες του παλμοτροφοδοτικού μετατροπέα. Η άρση του εμποδίου αυτού πραγματοποιείται με τη θεώρηση μίας σύνθετης αρμονικής μεταβολής η οποία αποτελείται από μία σταθερή τάση $V_{in_a} = const$ και της επιθυμητής αρμονικής $\Delta V_{in_b}(t) = A \cdot \sin \Omega \cdot t$. (6.16)

Η σύνθεση των δυο προηγούμενων μεταβολών δίνεται με τη σχέση :

$$V_{in}(t) = V_{in_a} + \Delta V_{in_b}(t), \quad V_{in}(t) \in [V_{in_a} - A, V_{in_a} + A] \quad (6.17)$$

Οι τιμές των σταθερών A , V_{amo} εκλέγονται ώστε να ισχύει :

$$\begin{aligned} [V_{in_a} - A, V_{in_a} + A] \subseteq [0, e_\alpha] &\Leftrightarrow V_{in_a} - A \geq 0 \quad \& \quad V_{in_a} + A \leq e_\alpha \Leftrightarrow \\ \Rightarrow V_{in_a} \geq A \quad \& \quad V_{in_a} \leq e_\alpha - A &\Leftrightarrow A \leq V_{in_a} \leq e_\alpha - A. \end{aligned} \quad (6.18)$$

Επομένως η συνολική απόκριση του συστήματος στη μεταβολή της τάσης του κινητήρα, σχέσης 6.16, είναι ο γραμμικός συνδυασμός των αποκρίσεων, αν εφαρμόζονταν ξεχωριστά $V_{in_a} = const$ και $\Delta V_{in_b}(t) = A \cdot \sin \Omega \cdot t$.

- Ο ρυθμός μεταβολής της τάσης ακροδεκτών του κινητήρα ορίζεται ότι είναι πεπερασμένος και μπορεί να προσεγγισθεί από τη σχέση $\frac{\partial V_{am}}{\partial t} = \frac{\Delta V_{am}}{\Delta t}$

$V_{am} = e_\alpha (D_{utyycle})$, $D_{utyycle} = \frac{T_{Dutyycle}}{T_{PWM}}$, $V_{am}(t) = V_{amo} + \Delta V_{am}(t)$, e_α : τάση τροφοδοσίας του παλμό-τροφοδοτικού ρυθμιστή τάσης.

- Η μεταβολή της γωνιακής ταχύτητας $\dot{\omega} = \frac{\partial \omega}{\partial t} = \frac{\Delta \omega}{\Delta t}$ είναι πεπερασμένη και η γωνιακή ταχύτητα μπορεί να προσεγγισθεί από τη σχέση : $\omega(t) = \omega_o + \Delta \omega(t)$.

Επομένως λόγω της παραπάνω απαίτησης η εφαρμοζόμενη τάση ακροδεκτών του κινητήρα δίνεται από την τροποποιημένη ημιτονοειδή συνάρτησης της σχέσης 6.17 και η απόκριση της γωνιακής ταχύτητας του κινητήρα δίνεται από τη σχέση $\omega = \omega_o + \Delta \omega$.

Ισχύει ότι $J_o \cdot \dot{\omega} = \sum_{i=0}^n T_i$ όπου J_o η ροπή αδρανείας του συστήματος, $\dot{\omega}$ η γωνιακή επιτάχυνση και T_i οι ροπές των εφαρμοζόμενων δυνάμεων επί του συστήματος. Στο σύστημα οι ασκούμενες ροπές είναι αυτή του κινητήρα και εκείνη των δυνάμεων τριβής. Επομένως το συνολικό άθροισμα των ροπών είναι $\sum T = T_{motor} - T_{friction}$, όπου T_{motor} η ροπή που αναπτύσσει ο κινητήρας και $T_{friction}$ η συνολική ροπή αντίστασης η οποία αναπτύσσεται στα έδρανα κύλισης του κινητήρα και προωθητήρα ανάδρασης.

$$V_a(t) = V_{a_o} + \Delta V_a(t) \Leftrightarrow V'_a(t) = \frac{\Delta V_a(t)}{\Delta t} \quad (6.19)$$

$$\omega(t) = \omega_o + \Delta\omega(t) \Leftrightarrow \omega'(t) = \frac{\Delta\omega(t)}{\Delta t} \quad (6.20)$$

Η αναπτυσσόμενη ροπή του κινητήρα για την παραπάνω μεταβολή της τάσης ακροδεκτών του κινητήρα σχέση 6.19 θεωρείται ότι προσεγγίζεται από τη σχέση :

$$T_{motor}(V_\alpha, \omega) = T_{motor}(V_{\alpha_o}, \omega_o) + \frac{\partial T_{motor}(V_\alpha, \omega)}{\partial V_\alpha} \Delta V_\alpha(t) + \frac{\partial T_{motor}(V_\alpha, \omega)}{\partial \omega} \Delta\omega(t) \quad (6.21)$$

Ισχύει ότι $\partial V_\alpha(t) = \Delta V_\alpha(t)$ & $\partial\omega(t) = \Delta\omega(t)$ εκ' των σχέσεων 6.19 και 6.20 και εφαρμόζοντας τα στην παραπάνω σχέση προκύπτει η σχέση της αναπτυσσόμενης ροπής του κινητήρα για την εφαρμοζόμενη τάση της σχέσης 6.17.

$$T_{motor}(V_\alpha, \omega) = T_{motor}(V_{\alpha_o}, \omega_o) + \frac{\partial T_{motor}(V_\alpha, \omega)}{\partial V_\alpha} \Delta V_\alpha(t) + \frac{\partial T_{motor}(V_\alpha, \omega)}{\partial \omega} \Delta\omega(t) \quad (6.22)$$

$$\text{Έστω ότι ο λόγοι γράφονται } \frac{\partial T_{motor}(V_\alpha, \omega)}{\partial V_\alpha} = K_V, \frac{\partial T_{motor}(V_\alpha, \omega)}{\partial \omega} = K_\omega$$

$$T_{motor}(V_\alpha, \omega) = T_{motor}(V_{\alpha_o}, \omega_o) + K_V \cdot \Delta V_\alpha + K_\omega \cdot \Delta\omega \quad (6.23)$$

Εφαρμόζοντας στη σχέση 6.15 τις εκφράσεις των ροπών του κινητήρα, σχέση 6.23, και των απωλειών τριβής, σχέση 1.17a, προκύπτει η τελική σχέση που συνδέει τη ροπή αδρανείας με τη συνισταμένη των εφαρμοζόμενων ροπών.

$$J_o \cdot \dot{\omega} = \sum M = T_{motor} - T_{friction} \Leftrightarrow$$

$$J_o \cdot \dot{\omega} = T_{motor}(V_{\alpha_o}, \omega_o) + K_V \cdot \Delta V_\alpha + K_\omega \cdot \Delta\omega - (A_f \cdot \omega(t) + B_f) \quad (6.24)$$

Στην 6.24 εφαρμόζεται η έκφραση της γωνιακής ταχύτητας της σχέσης 6.20 .

$$J_o \cdot \dot{\omega} = T_{motor}(V_{\alpha_o}, \omega_o) + K_V \cdot \Delta V_\alpha + K_\omega \cdot \Delta\omega - (A_f \cdot (\omega_o + \Delta\omega) + B_f) \quad (6.25)$$

Θεωρούμαι ότι οι μεταβολές της γωνιακής ταχύτητας είναι πεπερασμένες και πριν από κάθε μεταβολή το σύστημα κινητήρα-προωθητήρα βρίσκεται σε κατάσταση ισορροπίας .Δηλαδή η γωνιακή ταχύτητα , ω , είναι σταθερή και ίση με ω_o . Στην κατάσταση ισορροπίας ισχύει $\dot{\omega} = 0$ και επομένως :

$$J_o \cdot \dot{\omega} = 0 = \sum T = T_{motor} - T_{friction} \Rightarrow T_{motor} = T_{friction} \Rightarrow$$

$$T_{motor}(V_{\alpha_o}, \omega_o) = A_f \cdot \omega_o + B_f \quad (6.26)$$

Επομένως η σχέση 6.25 με εφαρμογή της σχ.6.26 γράφεται :

$$J_o \cdot \dot{\omega} = K_V \cdot \Delta V_\alpha + K_\omega \cdot \Delta\omega - A_f \cdot \Delta\omega$$

$$J_o \cdot \dot{\omega} + (A_f - K_\omega) \cdot \Delta\omega = K_V \cdot \Delta V_\alpha \quad (6.27)$$

όπου $\dot{\omega}(t)$, $\Delta\omega(t)$, $\Delta V_\alpha(t)$ και $J_o, A_f, K_\omega, K_V = \text{Constant}$

Η τελευταία εξίσωση 6.27 είναι γραμμική διαφορική εξίσωση πρώτου βαθμού και για την επίλυση της χρησιμοποιώ το μετασχηματισμό Laplace. Ο Laplace μετασχηματισμός των σχέσεων 6.19 και 6.20 είναι :

$$\omega(t) = \omega_o + \Delta\omega(t) \stackrel{L}{=} \omega(s) = \frac{\omega_o}{s} + \Delta\omega(s) \quad \& \quad \Delta\omega(t) \stackrel{L}{=} \Delta\omega(s) \quad (6.28)$$

$$\dot{\omega}(t) = s \cdot \omega(s) - \omega_o = s \cdot \left(\frac{\omega_o}{s} + \Delta\omega(s) \right) - \omega_o = s \cdot \frac{\omega_o}{s} + s \cdot \Delta\omega(s) - \omega_o = s \cdot \Delta\omega(s) \quad (6.29)$$

$$V_\alpha(t) = V_{\alpha_o} + \Delta V_\alpha(t) \stackrel{L}{=} V_\alpha(s) = \frac{V_{\alpha_o}}{s} + \Delta V_\alpha(s) \quad \& \quad \Delta V_\alpha(t) \stackrel{L}{=} \Delta V_\alpha(s) \quad (6.30)$$

Από εφαρμογή των παραπάνω σχέσεων στη μετασχηματισμένη κατά Laplace εξίσωση 6.27 προκύπτει η συνάρτηση μεταφοράς του συστήματος πρόωσης χωρίς φορτίο:

$$J_o \cdot s \cdot \Delta\omega(s) + (A_f - K_\omega) \cdot \Delta\omega(s) = K_V \cdot \Delta V_\alpha(s) \quad (6.31)$$

$$(J_o \cdot s + A_f - K_\omega) \cdot \Delta\omega(s) = K_V \cdot \Delta V_\alpha(s)$$

$$\frac{\Delta\omega(s)}{\Delta V_\alpha(s)} = \frac{K_V}{J_o \cdot s + A_f - K_\omega} = H(s) \quad (6.32)$$

Όπου τα $\Delta\omega(s)$ & $\Delta V_\alpha(s)$ εκφράζουν τη πεπερασμένη μεταβολή της γωνιακής ταχύτητας και της εφαρμοζόμενης τάσης στους ακροδέκτες του ηλεκτρικού κινητήρα κοντά στο επιλεγμένο σημείο γραμμικοποίησης του συστήματος πρόωσης. Θεωρώ το σύστημα $\Delta V_\alpha(s) \rightarrow H(s) \rightarrow \Delta\omega(s)$. Σύμφωνα με τη θεωρία γραμμικών συστημάτων βλέπε Συστήματα αυτομάτου ελέγχου Ξηρός 2003 κεφ.2.4 η απόκριση ,σε αρμονική διέγερση $\Delta V_\alpha(j \cdot \Omega) = \sin(j \cdot \Omega)$, ενός γραμμικού συστήματος, $H(s)$, είναι επίσης αρμονική ,και δίνεται από τη σχέση:

$$\Delta\omega(j \cdot \Omega) = |H(j \cdot \Omega)| \cdot \sin(j \cdot \Omega + \phi) \quad (6.33)$$

, όπου Ω γωνιακή συχνότητα της διέγερσης και $\phi = \angle H(j \cdot \Omega)$.

Εφόσον είναι δυνατό να υπολογιστεί η γωνία φάσης ϕ τότε μπορούμε να υπολογίσουμε τη ροπή αδρανείας J_o . Η 6.32 γράφεται :

$$H(s) = \frac{K_V}{J_o \cdot s + A_f - K_\omega} \stackrel{s=j \cdot \Omega}{\Rightarrow} H(j \cdot \Omega) = \frac{K_V}{J_o \cdot j \cdot \Omega + A_f - K_\omega} \Leftrightarrow$$

$$H(j \cdot \Omega) = \frac{K_V \cdot (A_f - K_\omega)}{(J_o \cdot \Omega)^2 + (A_f - K_\omega)^2} + \frac{-K_V \cdot J_o \cdot \Omega}{(J_o \cdot \Omega)^2 + (A_f - K_\omega)^2} \cdot j \quad (6.34)$$

$$\phi = \angle \frac{K_V \cdot (A_f - K_\omega)}{(J_o \cdot \Omega)^2 + (A_f - K_\omega)^2} + \frac{-K_V \cdot J_o \cdot \Omega}{(J_o \cdot \Omega)^2 + (A_f - K_\omega)^2} \cdot j \Leftrightarrow$$

$$\phi = \arctan \frac{-K_V \cdot J_o \cdot \Omega}{K_V \cdot (A_f - K_\omega)} = \arctan \frac{-J_o \cdot \Omega}{a_1 - K_\omega} \Rightarrow \tan \phi = \frac{-J_o \cdot \Omega}{A_f - K_\omega} \Rightarrow$$

$$\Rightarrow J_o = \frac{-1}{\Omega} \cdot (A_f - K_\omega) \cdot \tan \phi \quad (6.35)$$

Η σχέση 6.17 μετασχηματίζεται κατά Laplace:

$$V_a = V_{in}(t) = V_{in_a} + \Delta V_{in_b}(t) \stackrel{L}{=} V_{in}(s) = \frac{V_{in_a}}{s} + \Delta V_{in_b}(s) \quad (6.36)$$

$$\& \Delta V_{in_b}(t) = A \cdot \sin \Omega \cdot t \stackrel{L}{=} \Delta V_{in_b}(s) = A \cdot \frac{\Omega^2}{s^2 + \Omega^2}$$

Εφαρμόζοντας τη σχέση 6.36 για τη μεταβολή της τάσης των ακροδεκτών του κινητήρα στη συνάρτηση 6.32 .

$$\omega(s) = H(s) \cdot \left[\frac{V_{in_a}}{s} + \Delta V_{in_b}(s) \right] = H(s) \cdot \frac{V_{in_a}}{s} + H(s) \cdot \Delta V_{in_b}(s)$$

Ο όρος $H(s) \cdot \frac{V_{in_a}}{s}$ εφόσον η συνάρτηση μεταφοράς είναι γραμμική και $V_{in_a} = Const$ τότε η απόκριση, γωνιακή ταχύτητα ω , τείνει στη σταθερή τιμή.

$$\lim_{t \rightarrow \infty} \omega(t) = \lim_{s \rightarrow 0} s \cdot \omega(s) \Rightarrow$$

$$\lim_{s \rightarrow 0} (s \cdot H(s) \cdot \frac{V_{in_a}}{s}) = \lim_{s \rightarrow 0} \left(\cancel{s} \cdot \frac{K_V}{J_o \cdot 0 + A_f - K_\omega} \cdot \frac{V_{in_a}}{\cancel{s}} \right) = \frac{K_V \cdot V_{in_a}}{A_f - K_\omega} = Const \quad (6.37)$$

Δηλαδή η απόκριση του συστήματος πρόωσης στη σταθερή τάση V_{in_a} τείνει σε μια σταθερή τιμή με τη πάροδο του χρόνου .

Επειδή το σύστημα είναι γραμμικό ισχύει η αρχή της γραμμικής υπέρθεσης των αποκρίσεων. Επομένως η απόκριση του συστήματος πρόωσης δίνεται από την

άθροιση των αποκρίσεων του ξεχωριστά για τη σταθερή τιμή της τάσης V_{in_a} και την ημιτονοειδή μεταβολή ΔV_{in_b} .

Η απόκριση του συστήματος πρόωσης στην αρμονική μεταβολή της τάσης ακροδεκτών της σύμφωνα με τη σχέση ΔV_{in_b} χωρίς την παρουσία του όρου V_{in_a} είναι επίσης αρμονική βλέπε Συστήματα αυτομάτου ελέγχου Ξηρός 2003 κεφ.2.4 . Άρα η νέα απόκριση του συστήματος στη διέγερση $\Delta V_{in_b}(j \cdot \Omega) = A \cdot \sin(j \cdot \Omega)$ θα δίνεται από τη σχέση :

$$\Delta \omega(j \cdot \Omega) = A \cdot |H(j \cdot \Omega)| \cdot \sin(j \cdot \Omega + \phi) \Leftrightarrow \Delta \omega(j \cdot \Omega) = B \cdot \sin(j \cdot \Omega + \phi) \quad (6.38)$$

$$, B = A \cdot |H(j \cdot \Omega)| \quad (6.39)$$

B το πλάτος της αρμονικής απόκρισης του συστήματος.

Η απόκριση του συστήματος πρόωσης στη συνολική τάση της σχέσης 6.17 δίνεται από την άθροιση των επιμέρους αποκρίσεων 6.37 & 6.38 .

$$\omega(t) = \frac{K_V \cdot V_{in_a}}{A_f - K_\omega} + B \cdot \sin(j \cdot \Omega + \phi) \Leftrightarrow \quad (6.40)$$

$$\left. \begin{aligned} \omega_{Max}(t) &= \frac{\sin(j \cdot \Omega + \phi)=1 K_V \cdot V_{in_a}}{A_f - K_\omega} + B \cdot 1 \\ \omega_{Min}(t) &= \frac{\sin(j \cdot \Omega + \phi)=-1 K_V \cdot V_{in_a}}{A_f - K_\omega} + B \cdot (-1) \end{aligned} \right\} \Rightarrow$$

$$\Rightarrow \omega_{Max}(t) - \omega_{Min}(t) = 2 \cdot B \quad (6.41)$$

Η παραπάνω σχέση 6.39 γράφεται $\frac{B}{A} = |H(j \cdot \Omega)|$. Η σχέση αυτή εκφράζει ότι ο λόγος του πλάτους της απόκρισης προς το πλάτος της διέγερσης ισούται με το μέτρο της συνάρτησης μεταφοράς του συστήματος σε αρμονική διέγερση .Η συνάρτηση μεταφοράς όμως έχει ως μοναδική άγνωστη παράμετρο τη ροπή αδράνειας .

$$\text{Για } s=j \cdot \Omega \text{ η } H(s) \text{ γράφεται } H(j \cdot \Omega) = \frac{K_V}{J_o \cdot j \cdot \Omega + A_f - K_\omega} \quad (6.42)$$

$$\frac{B}{A} = |H(j \cdot \Omega)| = \left| \frac{K_V}{J_o \cdot j \cdot \Omega + A_f - K_\omega} \right| = \frac{|K_V|}{|J_o \cdot j \cdot \Omega + A_f - K_\omega|} \Rightarrow$$

$$\frac{B}{A} = |H(j \cdot \Omega)| = \frac{K_V}{\sqrt{(J_o \cdot \Omega)^2 + (A_f - K_\omega)^2}} \Leftrightarrow \quad (6.43)$$

$$\left(\frac{B}{A}\right)^2 = \frac{K_V^2}{(J_o \cdot \Omega)^2 + (A_f - K_\omega)^2} \Leftrightarrow (J_o \cdot \Omega)^2 + (A_f - K_\omega)^2 = \frac{K_V^2}{\left(\frac{B}{A}\right)^2} \Rightarrow \quad (6.44)$$

$$(J_o \cdot \Omega)^2 = \frac{K_V^2}{\left(\frac{B}{A}\right)^2} - (A_f - K_\omega)^2 \Rightarrow J_o \cdot \Omega = \sqrt{\frac{K_V^2}{\left(\frac{B}{A}\right)^2} - (A_f - K_\omega)^2} \Rightarrow$$

$$\Rightarrow J_o = \frac{1}{\Omega} \cdot \sqrt{\frac{K_V^2}{\left(\frac{B}{A}\right)^2} - (A_f - K_\omega)^2} \quad (6.45)$$

Από την τελευταία σχέση είναι εύκολο να υπολογιστεί η ροπή αδρανείας του συστήματος πρόωσης χωρίς φορτίο. Όλοι οι όροι και συντελεστές της σχέσης είναι γνωστοί.

Στο πρώτο κεφάλαιο έχει υπολογιστεί ότι η ροπή που αναπτύσσει ο κινητήρας, συναρτήσει της τάσης τροφοδοσίας και των στροφών, προσεγγίζεται από τη σχέση 1.13.:

$$T_{motor} = \left(\frac{K_\tau}{R_\alpha}\right) \cdot V_\alpha - \left(\frac{K_e \cdot K_\tau}{R_\alpha}\right) \cdot \omega$$

Στην παραπάνω σχέση τα V_α ω γράφονται σύμφωνα με τις σχέσεις 6.19 και 6.20.

$$T_{motor} = \left(\frac{K_\tau}{R_\alpha}\right) \cdot (V_{\alpha_o} + \Delta V_\alpha(t)) - \left(\frac{K_e \cdot K_\tau}{R_\alpha}\right) \cdot (\omega_{\alpha_o} + \Delta \omega_\alpha(t)) \Leftrightarrow$$

$$T_{motor} = \left(\frac{K_\tau}{R_\alpha}\right) \cdot V_{\alpha_o} - \left(\frac{K_e \cdot K_\tau}{R_\alpha}\right) \cdot \omega_{\alpha_o} + \left(\frac{K_\tau}{R_\alpha}\right) \cdot \Delta V_\alpha(t) - \left(\frac{K_e \cdot K_\tau}{R_\alpha}\right) \cdot \Delta \omega_\alpha(t) \Leftrightarrow$$

$$T_{motor} = T_{motor}(V_{\alpha_o}, \omega_{\alpha_o}) + \left(\frac{K_\tau}{R_\alpha}\right) \cdot \Delta V_\alpha(t) - \left(\frac{K_e \cdot K_\tau}{R_\alpha}\right) \cdot \Delta \omega_\alpha(t) \quad (6.46)$$

Κατά την προηγούμενη ανάλυση της ροπής του κινητήρα είχαμε καταλήξει στη σχέση 6.23 στην οποία ορίστηκαν οι συντελεστές K_V & K_ω .

$$T_{motor}(V_\alpha, \omega) = T_{motor}(V_{\alpha_o}, \omega_{\alpha_o}) + K_V \cdot \Delta V_\alpha + K_\omega \cdot \Delta \omega \quad (6.47)$$

Συγκρίνοντας τις δύο σχέσεις 6.46 και 6.47 προκύπτουν οι παρακάτω αντιστοιχίες για τους συντελεστές K_V & K_ω .

$$\frac{\partial T_{motor}}{\partial V_\alpha} = \frac{K_\tau}{R_\alpha} = K_V \quad (6.48)$$

$$\frac{\partial T_{motor}}{\partial \omega} = (-1) \cdot \frac{K_e \cdot K_\tau}{R_\alpha} = K_\omega \quad (6.49)$$

Συμπερασματικά καταλήγουμε ότι όλες οι παράμετροι της σχέσης 6.45 είναι γνωστές εκτός από τη ροπή αδρανείας J_o . Επομένως είναι δυνατή η εύρεση τις ροπής μετρώντας το πλάτος της αποκρίσεως.

6.3) Σχόλια επί της πειραματικής διάταξης και τις παραμέτρους της διεξαγωγής των μετρήσεων .

Η πειραματική μέτρηση της ροπής αδρανείας θα πραγματοποιηθεί για διαφορετικά πλάτη A καθώς και διαφορετικές συχνότητες Ω της ημιτονοειδούς διέγερσης της τάσης. Ορίστηκε προηγουμένως ότι η εφαρμοζόμενη ημιτονοειδής μεταβολή της τάσης δίνεται από τη σχέση 6.17 όπως εξηγήθηκε στο κεφάλαιο 6.2.3 . Επομένως η τάση γράφεται :

$$V_{in}(t) = V_{in_a} + \Delta V_{in_b}(t) = V_{in_a} + A \cdot \sin \Omega \cdot t \quad (6.50)$$

Οι τιμές των A & Ω θα πρέπει να πληρούν τις προϋποθέσεις της ανάλυσης του συστήματος και οι οποίες έχουν προαναφερθεί . Η τιμή του πλάτους A και της συχνότητας Ω θα πρέπει να είναι τέτοιες ώστε να μην επάγουν σημαντικές τιμές γωνιακής επιτάχυνσης $\dot{\omega}$, ώστε το διαφορικό $\frac{\partial \omega}{\partial t}$ να μπορεί να προσεγγιστεί με τις διαφορές $\frac{\Delta \omega}{\Delta t}$ και $\frac{\partial V_{am}}{\partial t} = \frac{\Delta V_{am}}{\Delta t}$ ώστε να ισχύουν οι σχέσεις για το γραμμικό πρόβλημα.

Οι τιμές του Duty Cycle του παλμού « PWM » ρύθμισης της τάσης του κινητήρα υπολογίζονται με τη βοήθεια της εφαρμογής «**Δημιουργία αρχείου αποστολή τιμών αναφοράς**» και αποθηκεύονται στο αρχείο αποστολής της εφαρμογή «**Κλίνης ώσης και ελέγχου στροφών**». Οι υπολογιζόμενες τιμές του Duty Cycle έχουν ακρίβεια ενός εκατοστού.

6.4) Εφαρμογή του Παλμοτροφοδοτικού μετατροπέα στον ηλεκτρικό κινητήρα και προβλήματα κατά την εκτέλεση του πειράματος μέτρησης της ροπής αδρανείας.

Ο παλμοτροφοδοτικός μετατροπέας τάσης ελέγχεται στην πραγματικότητα από ένα μικροεπεξεργαστή ο οποίος ελέγχει τον εισερχόμενο παλμό από την κάρτα και το μεταφράζει κατάλληλα σε παλμό ισχύος προς τον κινητήρα με διαφορετικά χαρακτηριστικά. Το τροφοδοτικό αυτό επειδή είναι σχεδιασμένο να χρησιμοποιείται σε τηλεκατευθυνόμενα μοντέλα και να συνεργάζεται με τηλεκατευθύνσεις, το πρόγραμμα το οποίο υλοποιείται στο επεξεργαστή του και η εν' γένει συμπεριφορά του δημιουργούσε προβλήματα, όταν συνδεόταν με την κάρτα LPC2106. Κατά την ανίχνευση αυτών των προβλημάτων και στην προσπάθεια επίλυσης τους διαπιστώθηκε ότι φέρει τη δυνατότητα «προγραμματισμού» .Γενικά όλοι οι σύγχρονοι χρησιμοποιούμενοι ψηφιακοί ελεγκτές στα νέα τηλεκατευθυνόμενα πριν

τη χρήσης τους για πρώτη φορά συνίσταται η ρύθμιση τους είτε μέσω της τηλεκατεύθυνσης είτε μέσω ενός υπολογιστή. Οι συνηθέστερες ρυθμίσεις είναι η δυνατότητα χρήσεως φρένου «Brake» ,σταματήματος του κινητήρα όταν το επίπεδο τάσης των μπαταριών τροφοδοσίας πέσει κάτω από ένα όριο ,επιλογής του τύπου του κινητήρα ,ορισμό του αριθμού των μπαταριών κ.λ.π. Στη διάθεσή μας δεν υπήρχε κάποιο εγχειρίδιο χρήσης με αποτέλεσμα τη μη γνώση και καθορισμό των παραμέτρων λειτουργίας του.

Η έλλειψη γνώσης και η αδυναμία ρύθμισης των παραμέτρων λειτουργίας του εισήγαγαν τα παρακάτω προβλήματα για την ολοκλήρωση της εργαστηριακής διάταξης .Αρχικά το πρώτο πρόβλημα που παρουσιάστηκε ήταν η ρύθμιση του σημείου αναφοράς της μηδενικής δράσης ελέγχου. Από πειράματα διαπιστώθηκε ότι το σημείο αυτό ορίζονταν τη στιγμή της λήψης του πρώτου παλμού «P.W.M» από την κάρτα .Ο αρχικός παλμός χρησιμοποιείται ως τιμή αναφοράς της ένδειξης μηδενικής δράσης ελέγχου «γκάζι» .Ο επεξεργαστής κάθε φορά που εισέρχονταν νέος παλμός συνέκρινε την τιμή του πλάτους «Duty Cycle» του παλμού και αφαιρώντας τη νέα τιμή από τη μηδενική τιμή ,υπολόγιζε την τιμή της νέας δράσης ελέγχου που θα εφαρμοστεί στον κινητήρα. Η τιμή της μηδενικής δράσης ελέγχου διατηρείτο σταθερή όσο βρισκόταν εν λειτουργία ο επεξεργαστής. Σε κάθε νέα εκκίνηση χανόταν η τιμή αυτή και πραγματοποιείται αυτόματα η διαδικασία της ρύθμισης από την αρχή.

Τα παραπάνω πρόβλημα ρύθμισης αν και βρέθηκε τρόπος επίλυσης τους οδήγησαν αργότερα σε νέα προβλήματα κατά τη φόρτιση του κινητήρα με πλήρες φορτίο .Το τροφοδοτικό της εγκατάστασης ,με πλήρη φόρτιση του κινητήρα, παρουσίαζε έντονη πτώση τάσης από τα 24 στο επίπεδο των 17 Volt. Η πτώση τάσης αυτή από τα 24 Volt στο 0 στα 17 στο 100% του «Duty Cycle» οδηγούσε τον επεξεργαστή στη διακοπή του κινητήρα και της λειτουργίας του ,ως ασφάλεια λόγω μειωμένης ισχύος μπαταριών, και την εκ' νέου επανεκκίνησης του εφόσον πλέον επανέρχονταν η τάση στο τροφοδοτικό στα επίπεδα των 24volt. Το αποτέλεσμα του άμεσου τερματισμού και επανεκκίνησης ήταν ο ορισμός του χρησιμοποιούμενου παλμού ως το νέο σημείο αναφοράς μηδενικής δράσης ελέγχου. Αυτό είχε ως συνέπεια την απορύθμιση του ελεγκτή και μη σωστή λειτουργία του σε σχέση με την αναμενόμενη σχεδιασμένη συμπεριφορά του.

Εν συνεχεία τρίτο πρόβλημα ήταν ο ορισμός των παραμέτρων του παλμού ,«PWM» της LPC2106, με τον οποίο θα οδηγείται ο παλμοτροφοδοτικός μετατροπέας. Πρωταρχικά έχει οριστεί ότι η περίοδος του παλμού ,όπως έχει προαναφερθεί, έπρεπε να είναι 20 msec ενώ το εύρος του Duty Cycle έπρεπε να κυμαίνεται από 1 έως 2 msec .Στο 1 msec αντιστοιχεί το 0% ενώ στο 2msec το 100% του «Duty Cycle» του παλμού τροφοδοσίας του κινητήρα. Η θεώρηση αυτή λειτουργούσε για το χειρισμό των σερβομηχανισμών αλλά όχι και για τον παλμοτροφοδοτικό μετατροπέα. Στη συνέχεια εξετάστηκε ο παλμός που παράγεται από το δέκτη μιας τηλεκατευθύνσεως και βρέθηκε το εύρος λειτουργία δεν ήταν 1~2 msec αλλά από 1.040 msec έως 1.840 msec για 0 έως 100% «Duty Cycle» ,ενώ η περίοδος ήταν η ίδια. Εφαρμόζοντας τις νέες τιμές πάλι παρουσιάστηκαν τα ίδια προβλήματα. Εκτελώντας πολλές δοκιμές και με τη χρήση του TecTronix TDS1000 παλμογράφου βρέθηκε τελικά ότι κατά την επικοινωνία με την κάρτα για άγνωστο λόγω το παραπάνω πεδίο αντιστοιχίας 0 έως 100% του «Duty Cycle» δεν ήταν 1,040 έως 1,840 msec αλλά από 1,040 έως 1,240 msec.

6.5) Νέος αναλογικός Παλμοτροφοδοτικός μετατροπέας τάσης.

Τα παραπάνω προβλήματα οδήγησαν στην απόφαση της κατασκευής εξ' αρχής νέου παλμοτροφοδοτικού μετατροπέα με όμοια χαρακτηριστικά λειτουργία όπως είχαν αναφερθεί στο πρώτο κεφάλαιο.

Τα βασικά χαρακτηριστικά του παλμού τροφοδοσίας του κινητήρα θα παραμείνουν τα ίδια .Δηλαδή παλμός τύπου «PWM» με περίοδο 0.5 mSec και ύψος παλμού ίσο με τη τάση τροφοδοσία από τροφοδοτικό της εγκατάστασης περίπου 24 Volt. Τη δημιουργία αυτού του παλμού ισχύος θα αναλάβουν τέσσερα εν παραλλήλω NPN Mosfet τα οποία θα κλείνουν το κύκλωμα τροφοδοσίας του κινητήρα στη γραμμή της γείωσης του κινητήρα και του τροφοδοτικού. Τα «Mosfet» οδηγούνται από ένα διπλό συγκριτή ο οποίος αναλαμβάνει να μεταφράσει το εισερχόμενο σήμα από την κάρτα κατάλληλα και να ανοίξει ή να κλείσει τη γραμμή της γείωσης του κίνητρα με τη γείωση του τροφοδοτικού. Σημειώνεται ότι η θετική γραμμή του τροφοδοτικού και του κινητήρα είναι μόνιμα συνδεδεμένες διαμέσου του νέου παλμοτροφοδοτικού.

Ο δημιουργούμενος παλμός από την κάρτα δεν είναι δυνατό να οδηγήσει απευθείας τα «Mosfett» καθώς το ύψος του παραγόμενου παλμού είναι 3.3Volt ενώ τα Mosfett» για να άγουν τη τάση απαιτούν 4 Volt.Για το λόγω αυτό παρεμβάλλεται ένας συγκριτής ο οποίος συγκρίνει το εισερχόμενο παλμό με μία τάση αναφοράς .Αν ο παλμός ξεπεράσει την τάση αυτή τότε εφαρμόζεται τάση +5 volt στην είσοδο «Gate» των Mosfet και άγουν ενώ όταν είναι χαμηλότερα εφαρμόζεται μηδενική τάση με αποτέλεσμα τη μη αγωγή του ρεύματος προς τη γείωση.

Στον παρακάτω πίνακα δίνονται τα υλικά που χρησιμοποιήθηκαν και τιμές τους.

Όνομα	Τιμή	Περιγραφή	Ποσότητα
R1	10 KΩ	Trimmer «linear»	1
R2	2.7 KΩ	Resistor ¼ watt	1
R3	1 KΩ	Resistor ¼ watt	1
R4	1MΩ	Resistor ¼ watt	1
R5	10MΩ	Resistor ¼ watt	1
R6~R9	100Ω	Resistor ¼ watt	4
R10	100 KΩ	Resistor ¼ watt	1
D1	1N4148	Diode	1
D2	1N4004	Diode	1
Led1	any	Led	1
S1	3A 220 Volt	Soft Switch On-Off	1
Q1~Q4	IRFZ44	NPN Mosfet transistor	4
IC2	LM393	Dual Comparator	1

Πίνακας 6.1) Χρησιμοποιούμενα υλικά για την κατασκευή του παλμοτροφοδοτικού μετατροπέα.

Στη συνέχεια παρουσιάζεται το σχεδιάγραμμα του παλμοτροφοδοτικού μετατροπέα.

Ο σχεδιασμός του νέου παλμοτροφοδοτικού έγινε με κύριο γνώμονα να διατηρηθούν οι ίδιες αρχές λειτουργίας με τον ήδη παρουσιασμένο στο πρώτο κεφάλαιο.

Συγκεκριμένα διατηρήθηκε η συνεχής μόνιμη τροφοδοσία της θετική γραμμής του κινητήρα με την τροφοδοσία. Η ρύθμιση της αγωγής τους ρεύματος προς τον κινητήρα πραγματοποιείται στη γραμμή της γείωσης. Ο παλμός εισέρχεται στο κύκλωμα του παλμοτροφοδοτικού μέσω της διόδου D1. Η διόδος χρησιμοποιείται για την προστασία της LPC2106 σε περίπτωση που στην ίδια γραμμή παρουσιαστεί αντίθετη τάση προκαλώντας βλάβη στον επεξεργαστή, αφού η κατεύθυνση για τον παλμό PWM της LPC2106 ορίζεται ως έξοδος. Έν' συνεχεία ο παλμός διέρχεται από την αντίσταση R10 στο θετικό Pin 3 του συγκριτή LM393. Οι αντιστάσεις σε σειρά R10 & R5 έχουν το ρόλο διαιρέτη τάσης. Το σήμα που φθάνει στο συγκριτή δίνεται από τη σχέση :

$$V_3^{IC2A} = V_{PWM_in} \cdot \frac{R5}{R5 + R10} = V_{PWM_in} \cdot \frac{10M}{10M + 100K} \Rightarrow$$

$$V_3^{IC2A} \approx 0.91 \cdot V_{PWM_in} \quad (6.51)$$

Με τη γραμμική ρυθμιζόμενη αντίσταση ρυθμίζεται η εφαρμοζόμενη τάση αναφοράς στο αρνητικό «Pin» του συγκριτή. Η τάση αυτή ορίζεται να είναι μικρότερη της μέγιστης τιμής του που εισέρχεται στο θετικό «Pin». Δηλαδή :

$$V_2^{IC2A} < V_3^{IC2A} = 0.91 \cdot V_{PWM_in} = 0.91 \cdot 3.3 = 3V_{olt} \quad (6.52)$$

Ο πυκνωτής C1 του κυκλώματος χρησιμοποιείται στο για τη σταθεροποίηση της τάσης τροφοδοσίας του συγκριτή. Ο LM393 χρησιμοποιεί στην έξοδο του λογική «open drain». Η μέθοδος αυτή εφαρμόζεται για την επικοινωνία συστημάτων τα οποία χρησιμοποιούν διαφορετικά επίπεδα τάση στα σήματα τους. Όταν το αποτέλεσμα της σύγκρισης είναι θετικό η έξοδος είναι ανοιχτή. Όταν είναι αρνητικό τότε η έξοδος κλείνει προς τη γείωση της τροφοδοσίας του συγκριτή. Η δημιουργία του σήματος, σε αυτό τον τύπο λειτουργίας, πραγματοποιείται με την εφαρμογή μιας τάσης στην έξοδο του συγκριτή ίσου ύψους με την επιθυμητή τιμή που θέλουμε να παράγει ως έξοδο όταν το αποτέλεσμα της σύγκρισης είναι θετικό. Η τάση αυτή εφαρμόζεται μέσω των αντιστάσεων R1 και R4. Σε σειρά λειτουργούν ως διαιρέτης τάσης με αποτέλεσμα η εφαρμοζόμενη τάση στην έξοδο του συγκριτή να είναι :

$$V_1^{IC2A} = V_{CC} \cdot \frac{R4}{R4 + R3} \stackrel{V_{CC}=5}{=} V_{CC} \cdot \frac{1M}{1M + 1K} \Rightarrow$$

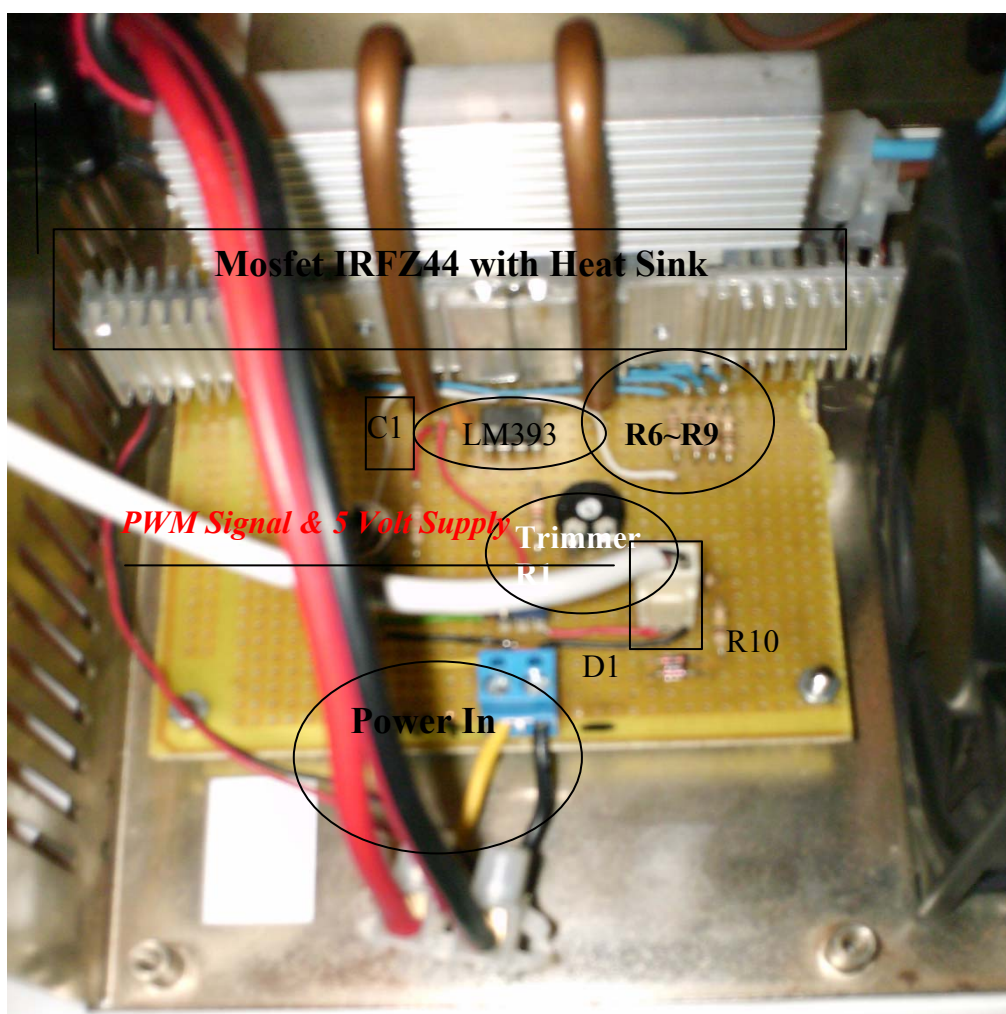
$$V_1^{IC2A} \approx 0.999 \cdot 5 \approx 5V_{olt}$$

Όταν το αποτέλεσμα της σύγκρισης είναι αρνητικό εφαρμόζεται στη έξοδο η γείωση με αποτέλεσμα να δημιουργείται «βραχυκύκλωμα» και η πτώση της τάση στην αντίσταση R3 είναι ίση με 5 Volt. Έτσι τελικά στην έξοδο η τιμή της τάσης

$$\text{είναι } V_1^{IC2A} = 0 \quad (6.53)$$

Το αποτέλεσμα της εισόδου του σήματος στο συγκριτή είναι η εξαγωγή όμοιου παλμού ίδιας περιόδου και «DutyCycle» αλλά πλάτους 5 Volt.

Το σήμα από την έξοδο του συγκριτή οδηγείται μέσω των αντιστάσεων R6~R9 στο «Pin» «Gate» των «Mosfet» Q1~Q4. Όταν το σήμα έχει τιμή μεγαλύτερη από 4 volt τα Q1~4 κλείνουν το κύκλωμα τροφοδοσίας του κινητήρα στη γείωση ενώ όταν είναι 0 μηδέν ανοίγουν το κύκλωμα. Οι αντιστάσεις R6~R9 χρησιμοποιούνται για την προφύλαξη του συγκριτή από ανεπιθύμητα μεγάλα ρεύματα σε περίπτωση αστοχίας. Επιπλέον αυξάνεται η αντίσταση της γραμμής ώστε να μην παίζει σημαντικό ρόλο το μήκος της γραμμής από την έξοδο του συγκριτή μέχρι την πύλη «Gate» των «mosfet» με σκοπό να ανοίγουν ταυτόχρονα. Τα τρανζίστορ ισχύος ή «mosfet» εφαρμόζονται πάνω σε κατάλληλη ψήκτρα ώστε να μειωθεί η παραγόμενη θερμότητα από τη λειτουργία τους μέσω του αέρα.



Εικόνα 6.1) Φωτογραφία του ολοκληρωμένου παλμοτροφοδοτικού μετατροπέα ο οποίος έχει τοποθετηθεί σε κατάλληλο κουτί, το οποίο φέρει ανεμιστήρα εξαγωγής της παραγόμενης θερμότητας.

6.6) Συνεργασία παλμοτροφοδοτικού μετατροπέα και συστήματος πρόωσης.

Στο πρώτο κεφάλαιο παρουσιάστηκε ως εισαγωγή η επίδραση της παρουσίας του παλμοτροφοδοτικού μετατροπέα στην τροφοδοσία του κινητήρα. Η ανάλυση η οποία πραγματοποιήθηκε αφορούσε την επίδρασή και λειτουργία ως προς το αντίστοιχο ισοδύναμο ηλεκτρικό κύκλωμα του κινητήρα. Οι σχέσεις που εξήχθησαν αφορούσαν τα μεγέθη τα οποία μπορούσαμε να μετρήσουμε με τα όργανα του εργαστηρίου, βολτόμετρο και αμπερόμετρο.

Ο παλμοτροφοδοτικός μετατροπέας, με αυτά τα χαρακτηριστικά, στην πραγματικότητα δεν ρυθμίζει την εφαρμοζόμενη τάση αλλά την παραγόμενη ροπή. Ο όρος ρυθμιστής στροφών ή ρυθμιστής τάσης εκφράζει το σκοπό για τον οποίο χρησιμοποιείται αλλά όχι την πραγματική του λειτουργία. Είναι γνωστό ότι ο όρος μόνιμη κατάσταση ορίζεται ως την κατάσταση στην οποία ο κινητήρα στρέφει με σταθερές στροφές υπό σταθερό «Duty cycle» του παλμού τροφοδοσίας. Στη μόνιμη κατάσταση δηλαδή ισχύει ότι κατά τη διάρκεια ενός παλμού τροφοδοσίας η ροπή που παράγεται κατά τη φάση που η τιμή του παλμού είναι μέγιστη «On» πρέπει να συντηρεί την αντιτιθέμενη ροπή του φορτίου και των τριβών σε όλη την υπόλοιπη διάρκεια του. Η προηγούμενη έκφραση απορρέει από την απαίτηση της αρχής διατήρησης της ενέργειας και συνοψίζεται στην παρακάτω σχέση 6.54.

Χρησιμοποιώντας την έκφραση για την παραγόμενη ροπή του κινητήρα 1.13 όπου τα K_V, K_ω ορίζονται από τις σχέσεις 6.48 & 6.49 ενώ $T_f(t), T_L(t)$ εκφράζουν τη ροπή των τριβών και του φορτίου αντίστοιχα.

$$\int_0^{t_{On}} ((K_V \cdot V_\alpha + K_\omega \cdot \omega) \cdot \omega) dt + \int_0^{T_s} ((T_f(t) + T_L(t)) \cdot \omega) dt = 0 \quad (6.54)$$

Εφόσον το σύστημα εξετάζεται στη μόνιμη κατάσταση τότε η ταχύτητα ω θεωρείται σταθερή και επομένως έχουμε .

$$\begin{aligned} \cancel{\omega} \int_0^{t_{On}} (K_V \cdot V_\alpha + K_\omega \cdot \omega) dt + \cancel{\omega} \int_0^{T_s} (T_f(t) + T_L(t)) dt = 0 \Rightarrow \\ \int_0^{t_{On}} (K_V \cdot V_\alpha + K_\omega \cdot \omega) dt + \int_0^{T_s} (T_f(t) + T_L(t)) dt = 0 \end{aligned} \quad (6.55)$$

Η τελευταία σχέση γράφεται όταν στον κινητήρα δεν εφαρμόζεται φορτίο , $T_L(t)=0$, και χρησιμοποιώντας τη σχέση 1.17a για τη ροπή των τριβών.

$$\begin{aligned} \int_0^{t_{On}} (K_V \cdot V_\alpha - K_\omega \cdot \omega) dt - \int_0^{T_s} (A_f \cdot \omega + B_f) dt = 0 \Rightarrow \\ (K_V \cdot V_\alpha - K_\omega \cdot \omega) \cdot t_{On} - (A_f \cdot \omega + B_f) \cdot T_s = 0 \Rightarrow \end{aligned}$$

$$(K_V \cdot V_\alpha - K_\omega \cdot \omega) \cdot \frac{t_{On}}{T_s} = (A_f \cdot \omega + B_f) \quad (6.56)$$

Ορίζεται ο λόγος $\frac{t_{On}}{T_s}$ ως το «Duty Cycle» και συμβολίζεται με το D.

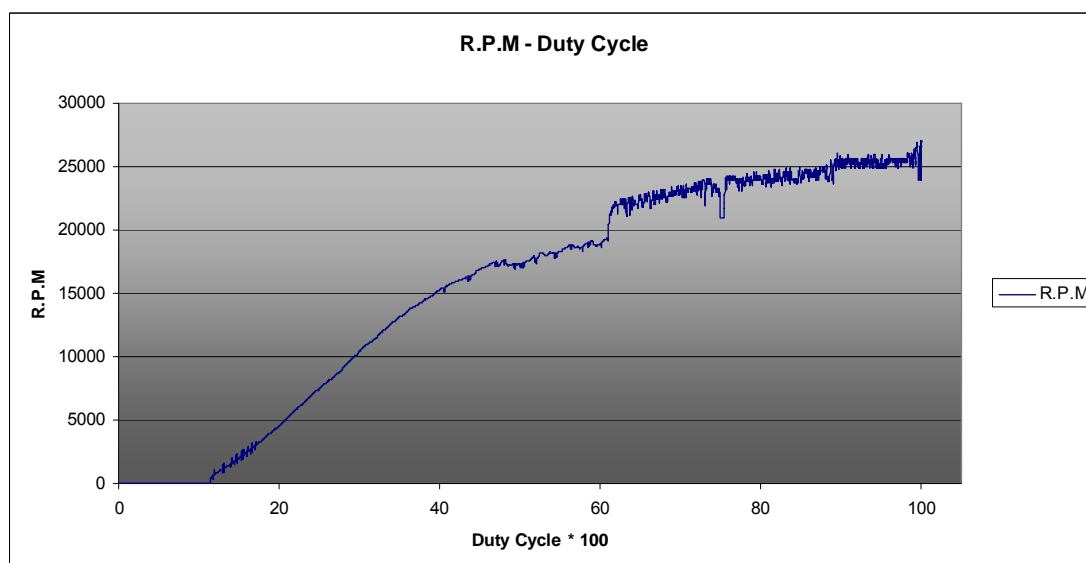
$$(K_V \cdot V_\alpha - K_\omega \cdot \omega) \cdot D = (A_f \cdot \omega + B_f) \Rightarrow$$

$$K_V \cdot V_\alpha \cdot D - B_f = (A_f + K_\omega \cdot D) \cdot \omega \Rightarrow$$

$$\omega = \frac{K_V \cdot V_\alpha \cdot D - B_f}{A_f + K_\omega \cdot D} \quad (6.57)$$

Από την τελευταία σχέση φαίνεται πλέον ότι δεν ισχύει η γραμμικότητα μεταξύ της δράσης ελέγχου και των στροφών. Επιπλέον όταν εφαρμόζεται αυτού του είδους έλεγχος στην τάση τροφοδοσίας επειδή η τάση δεν είναι συνεχής εμφανίζεται ο παράγοντας της επίδρασης των μόνιμων μαγνητών στη ροπή εκκίνησης. Ο κινητήρας εκτός από τη ροπή των τριβών για να εκκινήσει πρέπει να αναπτύξει και μία επιπλέον ροπή ώστε να υπερνικήσει την αντίθετη ροπή που ασκούν οι μαγνήτες όταν τα πηνία του κινητήρα δεν διαρρέονται από ρεύμα. Η ενέργεια αυτή χάνεται και εκφράζει την ροπή που απαιτείται για να συντηρηθεί η περιστροφή του κινητήρα. Ουσιαστικά επιδρά στο ενεργό Duty Cycle που ασκείται στον κινητήρα. Δηλαδή το $D' = D - a$.

Για τη διαπίστωση των παραπάνω υποθέσεων πραγματοποιήθηκε μέτρηση με την οποία μεταβλήθηκε το D από το 0% στο 100% και μετρήθηκε η ταχύτητα του κινητήρα. Τα αποτελέσματα παρουσιάζονται στο παρακάτω γράφημα.



Σχ 6.2) Καμπύλη Στροφών – DutyCycle υπό κενό.

Παρατηρείται ότι δεν ισχύει η γραμμική εξάρτηση D και στροφών και επιπλέον μετά τις 19000 εντοπίζονται έντονα φαινόμενα αύξησης της τριβής καθώς και

απώλειας τις γραμμικής εξάρτησης της από τις στροφές. Ο σχεδιασμός του κινητήρα και του προωθητήρα είναι για μέγιστο όριο λειτουργίας τις 19000 στροφές.

Μεταξύ του 20% και 40% του Duty Cycle και των στροφών παρατηρείται ότι ισχύει μια σχεδόν γραμμική σχέση. Εξετάζοντας τη σχέση 6.57 και υποθέτοντας ότι το D μεταβάλλεται ελάχιστα μπορεί να υποθεθεί ότι η τιμή του παρανομαστή παραμένει σταθερή και η σχέση να γραφεί:

$$\omega(D) = \frac{K_V \cdot V_\alpha \cdot D}{A_f + K_\omega \cdot D} - \frac{B_f}{A_f + K_\omega \cdot D} = \frac{K_V \cdot V_\alpha}{A_f + K_\omega \cdot D} D - \frac{B_f}{A_f + K_\omega \cdot D} \quad (6.58)$$

Από τις σχέσεις 1.13 και 1.17a που ισχύουν για την εφαρμογή στον κινητήρα σταθερή τάση έχουμε υπό κενό φορτίο την ισορροπία των ροπών κινητήρα και τριβών.

$$\begin{aligned} T_{motor} &= K_V \cdot V_\alpha - K_\omega \cdot \omega = T_f = A_f \cdot \omega + B_f \Rightarrow \\ \omega &= \frac{K_V \cdot V_\alpha - B_f}{K_\omega + A_f} \end{aligned} \quad (6.59)$$

Εξετάζοντας την ομοιότητα των σχέσεων 6.58 και 6.59 προκύπτει ότι για την περιοχή του Duty Cycle 20εως 40% η ροπή του κινητήρα θα μπορούσε να γραφεί :

$$T_{motor}^D = K_D \cdot D - K_\omega^D \cdot \omega \quad (6.60)$$

Ομοίως με την παραπάνω εξίσωση των ροπών σε μόνιμη κατάσταση υπό κενό φορτίο προκύπτει αντίστοιχη σχέση

$$\omega = \frac{K_D \cdot D - B_f}{K_\omega^D + A_f} \quad (6.70)$$

Σημειώνεται ότι οι συντελεστές δεν εκφράζουν φυσικά χαρακτηριστικά του κινητήρα αλλά προκύπτουν από την ανάγκη γραμμικοποίησης του προβλήματος. Οι τιμές των συντελεστών αυτών εξαρτώνται από τη συχνότητα του χρησιμοποιούμενου παλμού και την τάση που εφαρμόζεται στο παλμοτροφοδοτικό μετατροπέα. Σε κάθε πείραμα πρέπει να υπολογίζονται οι τιμές των συντελεστών αυτών .

Η ροπή αδρανείας J του συστήματος πρόωσης θα προκύψει από τη σχέση 6.45 στην οποία θα αντικατασταθούν οι όροι K_V, K_ω από τους αντίστοιχους όρους K_D, K_ω^D και θα προκύψει η παρακάτω σχέση :

$$J_o = \frac{1}{\Omega} \cdot \frac{\sqrt{\frac{K_D^2}{\left(\frac{B}{D_o}\right)^2} - (A_f + K_\omega^D)^2}}{\left(\frac{B}{D_o}\right)} \quad (6.71)$$

Με D_o ορίζεται το πλάτος της ημιτονοειδούς μεταβολής της δράσης ελέγχου και ο όροι K_D, K_ω^D δίνονται από τη σύγκριση της σχέσης 6.70 με τη σχέση $\omega(D)$ που προκύπτει από τις πειραματικές μετρήσεις.

6.7) Διεξαγωγή μετρήσεων και αποτελέσματα

Οι μετρήσεις θα πραγματοποιηθούν μεταξύ του 20 και 40 % του duty cycle και το πλάτος της ημιτονοειδούς μεταβολής κυμαίνεται μεταξύ των τιμών 1 έως 6 % ώστε η ροπή του κινητήρα να μπορεί να θεωρηθεί γραμμική σε σχέση με τις στροφές. Πραγματοποιήθηκαν τρεις μετρήσεις στις οποίες σε όλες διατηρούνταν το πλάτος της ημιτονοειδούς μεταβολής της δράσης ελέγχου «PWM» σταθερό ενώ μεταβάλλονταν το Ω του ημίτονου. Η συνάρτηση που εκφράζει τη μεταβολή της δράσης ελέγχου, είναι όμοια με τη σχέση που εκφράζει τη μεταβολή της τάσης στον κινητήρα στις σχέσεις 6.16 & 6.17 και γράφεται αντίστοιχα :

$$D(t) = D_a + \Delta D(t) \quad , \quad \Delta D(t) = D_o \cdot \sin(\Omega \cdot t) \quad (6.72)$$

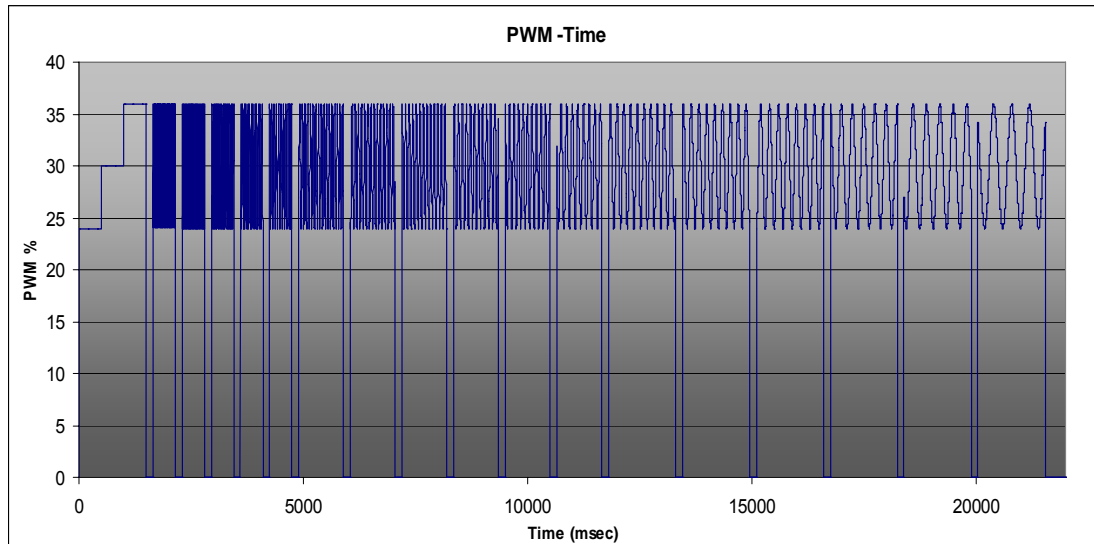
Για την εκτέλεση της ημιτονοειδούς μεταβολής στη δράση ελέγχου χρησιμοποιήθηκε η εφαρμογή «**Δημιουργία αρχείου αποστολής τιμών αναφοράς**» του προγράμματος «**Πτυχιακή Καρδιάση Κωνσταντίνου**». Σε κάθε αρχείο εκτελούνταν 20 διαφορετικές ημιτονοειδείς συναρτήσεις ο οποίες διέφεραν μεταξύ τους ως προς την τιμή του Ω .

Στον παρακάτω πίνακα δίνονται οι εκτελούμενες συναρτήσεις στο αρχείο της μέτρησης της αδράνειας.

A/A	D_a %	$\Delta D(t) = D_o \cdot \sin(\Omega \cdot t)$	Χρόνος έναρξης Sec	Χρόνος λήξης Sec
1	30	-6	0	10
2	30	0	10	20
3	30	+6	20	30
4	30	+6*cos(((2*π*/0.3) t)	33	43
5	30	+6*cos(((2*π*/0.5) t)	46	56
6	30	+6*cos(((2*π*/0.6) t)	59	69
7	30	+6*cos(((2*π*/0.8) t)	72	82
8	30	+6*cos(((2*π*/1) t)	85	95
9	30	+6*cos(((2*π*/1.2) t)	98	118
10	30	+6*cos(((2*π*/1.4) t)	121	141
11	30	+6*cos(((2*π*/1.6) t)	144	164
12	30	+6*cos(((2*π*/1.8) t)	167	187
13	30	+6*cos(((2*π*/2) t)	190	210
14	30	+6*cos(((2*π*/2.5) t)	213	233
15	30	+6*cos(((2*π*/3) t)	236	266
16	30	+6*cos(((2*π*/3.5) t)	269	299
17	30	+6*cos(((2*π*/4) t)	302	332
18	30	+6*cos(((2*π*/5) t)	335	365
19	30	+6*cos(((2*π*/6) t)	368	398
20	30	+6*cos(((2*π*/8) t)	401	431

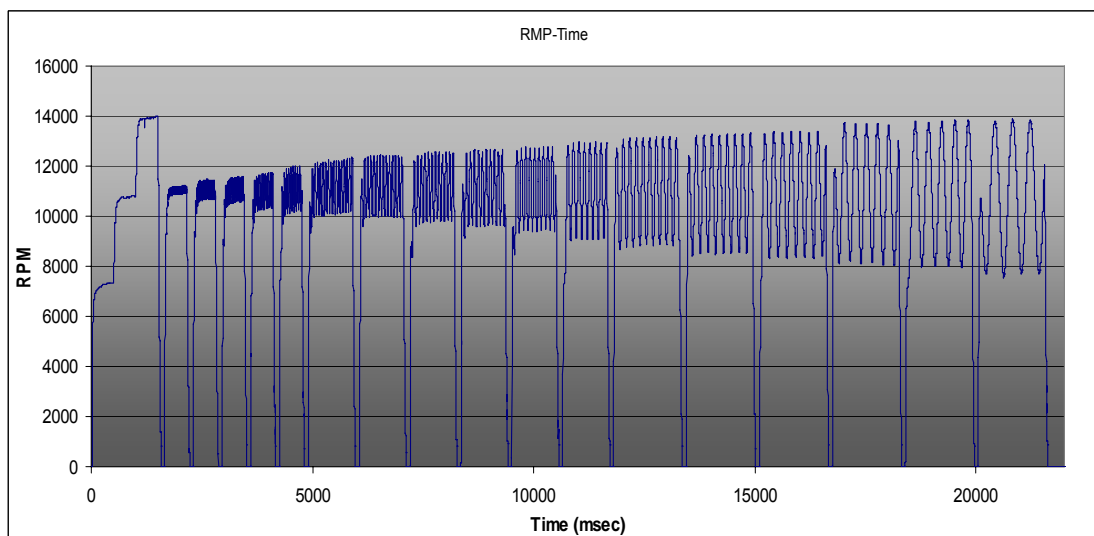
Πίνακας 6.2) Οι εκτελούμενες συναρτήσεις για τον υπολογισμό της ροπής αδρανείας.

Οι τιμές του αρχείου που σχηματίζεται από τις συναρτήσεις του παραπάνω πίνακα παρουσιάζονται στο παρακάτω διάγραμμα .



Σχ 6.3) Τιμές αρχείου αποστολής για τη μέτρηση της ροπής αδρανείας

Η απόκριση του κινητήρα κατά την εκτέλεση του πειράματος δίνεται στο παρακάτω διάγραμμα .



Σχ 6.4) Σχεδιάγραμμα της απόκρισης των στροφών του κινητήρα στη μεταβολής της δράσης ελέγχου του σχ.6.3

Τα αποτελέσματα της παραπάνω μέτρησης παρουσιάζονται στον παρακάτω πίνακα.

Από τις τρεις πρώτες συναρτήσεις υπολογίζεται η ευθεία της συνάρτησης

$$\omega = \frac{K_D \cdot D - B_f}{K_\omega^D + A_f} \text{ και από εκεί οι όροι } K_D, K_\omega^D .$$

A/A	$D = const$ Duty Cycle*100)	RPM
1	24	7328
2	30	10782
3	36	13969

Πίνακας 6.3) Μετρήσεις της ταχύτητας του κινητήρα για τις τρεις πρώτες συναρτήσεις.

Όποτε υπολογίζονται οι όροι $K_D = 0,001038032$ [Nm/(DutyCycle*100)] ,

$$K_{\omega}^D = 9,76525E-06$$
 [Nm/(rad/sec)]. (6.73)

A/A	B (rad/sec)	D_o (Duty Cycle*100)	Ω (rad/Sec)	$J_o = \frac{1}{\Omega} \cdot \sqrt{\frac{K_D^2}{\left(\frac{B}{D_o}\right)^2} - (A_f + K_{\omega}^D)^2}$
4	16,703	5,969	20,944	1,7690E-05
5	41,853	5,969	12,566	1,1695E-05
6	52,971	5,969	10,472	1,1040E-05
7	74,107	5,969	7,854	1,0402E-05
8	94,038	5,969	6,283	1,0097E-05
9	112,469	5,969	5,236	9,9588E-06
10	128,481	5,969	4,488	9,9890E-06
11	145,477	5,969	3,927	9,8564E-06
12	158,158	5,969	3,491	1,0002E-05
13	174,002	5,969	3,142	9,8224E-06
14	201,638	5,969	2,513	9,9750E-06
15	227,818	5,969	2,094	9,8314E-06
16	249,338	5,969	1,795	9,6776E-06
17	263,265	5,969	1,571	9,8260E-06
18	283,351	5,969	1,257	1,0144E-05
19	305,978	5,969	1,047	9,2780E-06
20	322,999	5,969	0,785	9,2100E-06

Πίνακας 6.4) Μετρήσεων της ταχύτητας του κινητήρα και υπολογισμός της ροπής αδρανείας για τις συναρτήσεις από 4 έως 20 της πρώτης μέτρησης .

Οι μετρήσεις 4 έως και 8 θα αποκλειστούν διότι η χρησιμοποιούμενη ταχύτητα ή συχνότητα Ω είναι μικρή οπότε η ημιτονοειδής μεταβολή δεν προσεγγίζετε ικανοποιητικά .Επομένως η ροπή αδρανείας θα προκύψει από τη μέση τιμή των υπολογισμένων ροπών για τις σχέσεις 9 έως 20 .

$$\bar{J}_o = \frac{\sum_{i=9}^{20} J_o^i}{11} = 9,798E-06$$
 (6.74)

και η διακύμανση είναι :

$$VAR(J_o) = \frac{\sum_{i=9}^{20} (J_o^i - \bar{J}_o)^2}{n-1} = 5,150E-14$$
 (6.75)

Εκτελώντας και τις υπόλοιπες μετρήσεις στις οποίες χρησιμοποιήθηκαν οι ίδιες συναρτήσεις χρόνοι και σειρά εκτέλεσης των συναρτήσεων με διαφοροποίηση ως προς το πλάτος βρέθηκε ότι η μέση υπολογιζόμενη τιμή της ροπής αδρανείας κυμαινόταν από την τιμή $8,259E-05$ για $D_o = 2$ και $9,101E-06$ $D_o = 3$.

Η διακύμανση αυτή της ροπής αδρανείας οφείλεται στο ότι οι συνθήκες εκτέλεσης του πειράματος δεν είναι ιδανικές και επιπλέον ο γραμμικός χαρακτήρας της απόκρισης των στροφών στη μεταβολή της δράσης ελέγχου δεν είναι ακριβής. Τονίζεται ότι η γραμμικοποίηση της περιστροφικής ταχύτητας συναρτήσει της δράσης ελέγχου πραγματοποιείται μόνο για την εύρεση της ροπής αδρανείας και ειδικότερα την εκτίμηση της τάξης της τιμής. Η μέτρηση δεν είναι εύκολη καθώς η χρήση του τροφοδοτικού, μετασχηματιστή, δεν προσφέρει σταθερή τάση τροφοδοσίας. Σε συνδυασμό με την πολύ μικρή αδράνεια του συστήματος πρόωσης καθώς και την υψηλή ροπή που αναπτύσσει ο κινητήρας έχει σα συνέπεια η μεταβολή της τάσης τροφοδοσίας να εισάγει σημαντικό θόρυβο στις μετρήσεις. Ωστόσο παρόλες τις δυσκολίες μπορεί να εξαχθεί ένα γενικό συμπέρασμα ότι το μέτρο της ροπής αδρανείας του συστήματος πρόωσης υπό κενό φορτίο είναι της τάξης των 10^{-5} Nm.

Για την επίλυση των παραπάνω προβλημάτων προτείνεται η χρήση συστοιχίας μπαταριών με τάση μεγαλύτερη ή ίση των 16 Volt η, η οποία θα προσφέρει σταθερότερο περιβάλλον για τη διεξαγωγή της μέτρησης. Επιπλέον ένα σημαντικό συμπέρασμα στο οποίο καταλήγουμε είναι ότι η συχνότητα λειτουργίας του παλμοτροφοδοτικού μετατροπέα είναι πολύ μικρή για το συγκεκριμένο κινητήρα, με πολύ μικρή ροπή αδρανείας, ώστε να θεωρηθεί ως μετατροπέας τάσης. Συχνότητες μεγαλύτερες των 20 KHz μπορεί να θεωρηθεί ότι απαιτούνται. Τέτοιες συχνότητες δεν κατέστη δυνατόν να χρησιμοποιηθούν διότι δεν το επέτρεπε ο εξοπλισμός.

7) Μέτρηση ώσης προωθητήρα ανάδρασης.

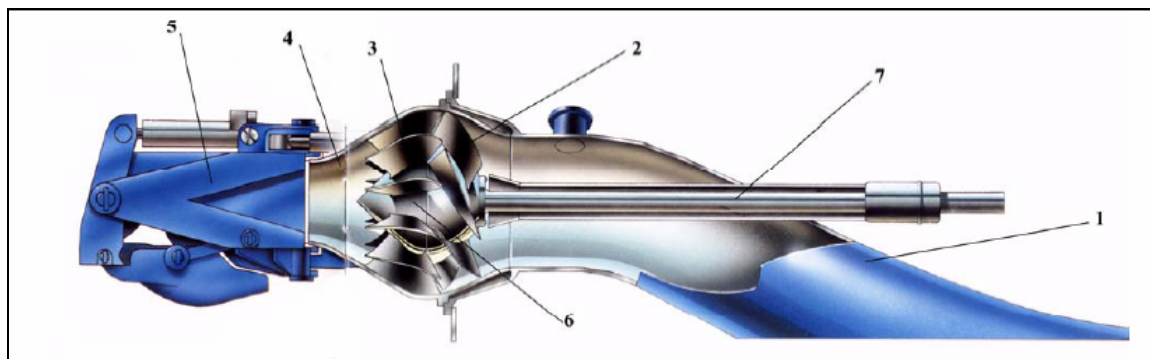
7.0) ΣΥΝΟΨΗ.

Στο κεφάλαιο αυτό πραγματοποιείται η μέτρηση της ώσης του δοκιμαζόμενου συστήματος προώσεως. Για τον έλεγχο και τη διεξαγωγή της μέτρησης χρησιμοποιούνται οι ηλεκτρονικές διατάξεις που παρουσιάστηκαν στα προηγούμενα κεφάλαια. Επίσης στον υπολογιστή η εφαρμογή «Πτυχιακή Καρδάση Κωνσταντίνου» χρησιμοποιείται με σκοπό την οδήγηση των παραπάνω ηλεκτρονικών διατάξεων καθώς επίσης της αποθήκευσης και της επεξεργασίας των μετρήσεων.

Οι μετρήσεις πραγματοποιήθηκαν με τη χρήση του ελέγχου του κινητήρα και των δύο δυνατών μεθόδων ,δηλαδή του ελέγχου του κανόνα και του ελέγχου στροφών. Στο τέλος παρουσιάζονται τα αποτελέσματα των μετρήσεων ώσης σε μορφή γραφικών παραστάσεων τόσο ως προς τον την τιμή του χρόνου όσο ως προς την περιστροφική ταχύτητα του προωθητήρα . Επίσης το κεφάλαιο κλείνει με την παρουσίαση της αποδιδόμενης ισχύος του προωθητήρα ανάδρασης σε γραφική μορφή ισχύος στροφών λειτουργίας .

7.1) Περιγραφή Προωθητήρα Ανάδρασης (Waterjet).

Στο παρακάτω σχήμα παρατίθεται μία απεικόνιση ενός waterjet με αριθμημένα τα κυριότερα μέρη του.



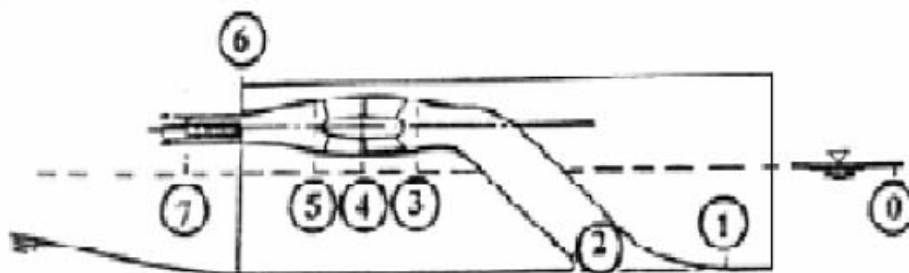
Σχήμα 1.1: Κύρια μέρη προωθητήρα ανάδρασης.

1. Εισαγωγή (flush type)
2. Πτερωτή (μικτής ροής)
3. Στάτης
4. Ακροφύσιο
5. Σύστημα πηδαλιουχίας /αναστροφής
6. Πλήμνη
7. Άξονας

Το νερό μέσω της εισαγωγής (1) του waterjet οδηγείται στην πτερωτή (2), η οποία προσφέρει ενέργεια αυξάνοντας απότομα την πίεση και προσδίδοντας

περιστροφική αδράνεια στο νερό. Ο στάτορας (3) ευθυγραμμίζοντας τη ροή αυξάνει περαιτέρω την ενέργεια εντατικής κατάστασης του νερού, η οποία κατά την εκτόνωση από το ακροφύσιο εξόδου (4) μετατρέπεται σε κινητική ενέργεια.

Για τη μελέτη του συστήματος ανάδρασης, γίνεται αρίθμηση των σημείων-σταθμών του συστήματος κατά I.T.T.C. όπως αυτά παρουσιάζονται στο παρακάτω σχήμα.



Σχήμα 1-2 : Σταθμοί προωθητήρα ανάδρασης κατά I.T.T.C.

Σταθμός 0 : Αδιατάρακτη ροή μακριά από το σύστημα. Αν το σύστημα κινείται με ταχύτητα V_s , η ταχύτητα του ρευστού V_0 ως προς παρατηρητή επί του συστήματος είναι $V_0 = V_s$.

Σταθμός 1 : Μπροστά από την εισαγωγή και πριν αρχίσουν να εισάγονται απώλειες. Η ταχύτητα του ρευστού σε αυτόν το σταθμό δίνεται από τη σχέση $V_1 = V_0 \cdot (1 - W)$. Επομένως το ποσοστό ομόρου ορίζεται από τη σχέση :

$$W = \frac{V_0 - V_1}{V_0} \quad (7.1)$$

Σταθμός 2 : Αμέσως μετά την είσοδο του αγωγού της εισαγωγής.

Σταθμός 3 : Πριν την περωτή.

Σταθμός 4 : Μεταξύ περωτής και στάτορα.

Σταθμός 5 : Μετά το στάτορα .

Σταθμός 6 : Έξοδος ακροφύσιου.

Σταθμός 7 : Σταθμός όπου η δέσμη του νερού εξέρχεται και η πίεση εξισώνεται με την ατμοσφαιρική. Συνήθως ο σταθμός 6 ταυτίζεται με το σταθμό 7.

7.2) Υπολογισμός εξερχόμενης Ισχύος Προωθητήρα Ανάδρασης.

Διευκρινίζεται ότι η ροή του νερού μελετάται στην ειδική περίπτωση της στατικής έλξης. Δηλαδή ισχύει $V_s = V_0 = 0$, επομένως και $V_1 = 0$. Το ποσοστό ομόρου W , το οποίο δίνεται από τη σχέση 7.1, σε αυτήν την περίπτωση δεν ορίζεται. Στην περίπτωση λοιπόν της στατικής έλξης, οι γραμμές ροής στην είσοδο της αναρρόφησης παρουσιάζουν συμμετρική κατανομή. Σχηματίζεται καταβόθρα με συμμετρικό οριακό στρώμα γύρω από την εισαγωγή, αφού δεν υπάρχει οριακό στρώμα λόγω ταχύτητας του συστήματος

• Ορισμοί Συμβόλων

Στη συνέχεια ορίζονται τα μεγέθη τα οποία πρόκειται να χρησιμοποιηθούν στην ενεργειακή ανάλυση του προωθητήρα ανάδρασης .

A_j : Επιφάνεια σταθμού j [m^2], (διατομή ροής).

E_j : Κινητική ενέργεια ανά μονάδα μάζας του ρευστού στο σταθμό j [$Nt \cdot m / Kgr$].

Ορίζεται από τη σχέση $E_j := \frac{1}{2} V_j^2$ (7.2)

M_j : Ροή ορμής στο σταθμό j [Nt]. Ορίζεται από τη σχέση : $M_j := \dot{m} \cdot V_j$ (7.3)

P_j : Ενέργεια εντατικής κατάστασης ανά μονάδα όγκου (πίεση) του ρευστού στο σταθμό [$\frac{Nt}{m^2}$]

V_j : Μέση ταχύτητα του ρευστού στο σταθμό j [$\frac{m}{s}$]

h_j : Υψομετρική απόσταση της ισάλου από το σταθμό j [m]

ρ : Πυκνότητα του ρευστού [$\frac{Kgr}{m^3}$]

Q : παροχή όγκου του ρευστού [$\frac{m^3}{s}$]. Ορίζεται από τη σχέση [$Q = A_l \cdot V_j$] (7.4)

\dot{m} : Παροχή μάζας του ρευστού [$\frac{Kgr}{s}$]. Ορίζεται από τη σχέση $\dot{m} = \rho \cdot Q$ (7.5)

Η παροχή σύμφωνα με το θεώρημα της συνέχειας και την υπόθεση της μη συμπιεστότητας του ρευστού, είναι σταθερή σε όλο το μήκος του αγωγού, γι' αυτό τα μεγέθη \dot{m} και Q δε φέρουν δείκτες.

Η ώση T που παράγει ο προωθητήρας ανάδρασης (Waterjet) ισούται με τη μεταβολή της ροής ορμής ΔM_{1-7} (Lewis EV, 1988; Πολίτης ΓΚ, 2004). Αυτό είναι άμεση συνέπεια της εφαρμογής του νόμου του Νεύτωνα $T = \frac{d}{dt}(m \cdot V)$ μεταξύ των σταθμών 1 και 7. Έτσι λοιπόν ισχύει :

$$T = \Delta M_{1-7} = \dot{m} \cdot (V_7 - V_1) \quad (7.6)$$

Επειδή όπως έχει αναφερθεί, μελετάται το σύστημα σε στατική έλξη, ισχύει $V_1 = 0$ και η εξίσωση (7.6) γράφεται :

$$T = \dot{m} \cdot V_7 = \rho \cdot Q \cdot V_7 = \rho \cdot Q \cdot \left(\frac{Q}{A_7}\right) \Rightarrow$$

$$T = \rho \cdot \frac{Q^2}{A_7} \quad (7.7)$$

Η λύση την εξίσωση (7.7) ως προς την παροχή όγκου του ύδατος προκύπτει:

$$Q = \sqrt{\frac{A_7 \cdot T}{\rho}} \quad (7.8)$$

Στη συνέχεια ορίζεται η ίσαλος γραμμή ως γραμμή αναφοράς του δυναμικού πεδίου. Εφαρμόζεται η εξίσωση Bernoulli στους σταθμούς 1 και 7, μετασχηματισμένη σε όρους ενέργειας ανά μονάδα μάζας. Έτσι λοιπόν η ενέργεια ανά μονάδα μάζας στην είσοδο του αγωγού (σταθμός 1) προκύπτει από τη σχέση :

$$E_1 = \frac{1}{2} \cdot V_1^2 + \frac{p_1}{\rho} - g \cdot h_1 \quad (7.9)$$

Θεωρώντας ότι η πίεση στο σταθμό 1 είναι ίση με την υδροστατική πίεση $\rho \cdot g \cdot h_1$, προκύπτει $E_1 = \frac{1}{2} \cdot V_1^2$. Με την παραδοχή όμως που ισχύει εξ αρχής $V_1 = 0$, προκύπτει τελικά $E_1 = 0$.

Η αντίστοιχη έκφραση της ενέργειας ανά μονάδα μάζας για το σταθμό 7, που είναι η ενέργεια ανά μονάδα μάζας που εξέρχεται από το σύστημα, δίνεται από την παρακάτω σχέση :

$$E_{out} = E_7 = \frac{1}{2} \cdot V_7^2 + \frac{p_7}{\rho} - g \cdot h_7 \quad (7.11)$$

Θεωρώντας ότι στην έξοδο του ακροφυσίου η πίεση του νερού είναι ίση με την υδροστατική πίεση, προκύπτει $\rho \cdot g \cdot h_7$. Άρα τελικά έχουμε για την εξερχόμενη ενέργεια ισχύει :

$$E_{out} = E_7 = \frac{1}{2} \cdot V_7^2 \quad (7.12)$$

Από την εφαρμογή της σχέσης (7.4), στην εξίσωση (7.12) προκύπτει ότι :

$$E_{out} = \frac{1}{2} \cdot \left(\frac{Q}{A_7}\right)^2 \quad (7.13)$$

Εισάγοντας στη σχέση (7.13) την εξίσωση 7.8, καταλήγουμε :

$$E_{out} = E_7 = \frac{1}{2} \cdot \frac{Q^2}{A_7^2} = \frac{1}{2} \cdot \frac{\left[\sqrt{\frac{A_7 \cdot T}{\rho}}\right]^2}{A_7^2} = \frac{1}{2} \cdot \frac{T}{\rho \cdot A_7} \quad (7.14)$$

Η στιγμιαία ισχύς που εξέρχεται από το σύστημα πρόωσης δίνεται από τη σχέση :

$$P_{out} = \dot{m} \cdot E_{out} = \rho \cdot Q \cdot \left[\left(\frac{1}{2} \cdot \frac{Q^2}{A_7^2}\right)\right] \Rightarrow$$

$$P_{out} = \frac{1}{2} \cdot \rho \cdot \frac{Q^3}{A_7^2} \quad (7.15)$$

Η εξίσωση 7.15 τελικά λαμβάνει την εξής μορφή με τη χρήση της σχέσης 7.8:

$$P_{out} = \frac{1}{2} \cdot \rho \cdot \frac{Q^3}{A_7^2} = \frac{1}{2} \cdot \rho \cdot \frac{\left[\sqrt{\frac{A_7 \cdot T}{\rho}} \right]^3}{A_7^2} \Rightarrow$$

$$P_{out} = \frac{T^{\frac{3}{2}}}{2 \cdot \sqrt{\rho \cdot A_7}} \quad (7.16)$$

Στην παραπάνω σχέση ο μόνος άγνωστος είναι η παραγόμενη ωστική δύναμη T . Η ωστική δύναμη εν γένει θεωρείται ότι είναι μία συνάρτηση των στροφών $n(\text{rpm})$ του άξονα της πτερωτής, οπότε ισχύει $T(n)$. Αν F είναι δύναμη, την οποία καταγράφει η δυναμοκυψέλη τότε ισχύει ότι $F(n) = T(n)$. Χρησιμοποιώντας στη συνέχεια την εργαστηριακή εγκατάσταση προσδιορίζεται η καμπύλη. Άρα η τελική μορφή της σχέσης υπολογισμού της ισχύος εξόδου είναι :

$$P_{Out}(N) \stackrel{F=T}{=} \frac{T(N)^{\frac{3}{2}}}{2 \cdot \sqrt{\rho \cdot A_7}} (\text{watt}) \quad (7.17)$$

Όπου η δύναμη $F(N)$ που καταγράφεται έχει μονάδες «Kgf» ή «Kp» και για να χρησιμοποιηθούν στην παραπάνω σχέση μετατρέπονται σε Newton. Η επιφάνεια A_7 σε m^2 .

7.3) Παρουσίαση των παραμέτρων διεξαγωγής των μετρήσεων.

Για τη μέτρηση της ώσης χρησιμοποιούνται και οι δύο μέθοδοι ελέγχου του κινητήρα ,δράσης ελέγχου και έλεγχος στροφών .Χρησιμοποιείται όλο το δυνατόν εύρος τιμών αναφοράς. Για τη δημιουργία του αρχείου των τιμών αναφοράς θα χρησιμοποιηθεί η εφαρμογή «**Δημιουργία αρχείου αποστολής τιμών αναφοράς**» του προγράμματος «**Πτυχιακή Καρδίαση Κωνσταντίνου**».

Εξαγωγή του αρχείου αποστολής με τη χρήση της μεθόδου «Κανόνας Ελέγχου».

Για τη δημιουργία του αρχείου χρησιμοποιήθηκε ο τρόπος κατασκευής «**Εισαγωγή τιμών από συνάρτηση**» . Επιλέχθηκε η γραμμική μεταβολή των τιμών του κανόνα ελέγχου από το 0 έως 100% με χρονικό διάστημα αύξησης τα 150 sec .Στη συνέχεια παραμένει στο 100% για 20 δευτερόλεπτα και κατόπιν μειώνεται πάλι γραμμικά από 100% έως 0% με διάστημα μείωσης ίσο με το πρώτο 150 δευτερόλεπτα. Η εύρεση της ώσης με μειούμενη δράση ελέγχου χρησιμοποιείται για τη διαπίστωση τυχόν διαφορετικής απόκρισης του συστήματος προώσεως κατά τη μείωση του κανόνα ελέγχου.

Συγκεντρωτικά η εκτελούμενες συναρτήσεις δίνονται στον παρακάτω πίνακα.

A/A	Χρονικό διάστημα εφαρμογής συνάρτησης	Συνάρτηση
1	0 ~ 150 Sec	$100.00 \cdot t / 150.000$
2	150 ~ 170 Sec	100
3	170 ~ 320 Sec	$100.00 - (100.00 \cdot (t - 170.000)) / 150.000$

7.1) Πίνακας εκτελούμενων συναρτήσεων με χρήση της μεθόδου δράσης ελέγχου για τη διεξαγωγή του πειράματος ώσης.

Εξαγωγή του αρχείου αποστολής με τη χρήση της μεθόδου «Έλεγχος Στροφών».

Για τη δημιουργία του αρχείου χρησιμοποιήθηκε ο τρόπος κατασκευής «Εισαγωγή τιμών από συνάρτηση», ομοίως με την προηγούμενη διαδικασία. Επιλέχθηκε η γραμμική μεταβολή των τιμών των στροφών από το 0 έως 16000 με χρονικό διάστημα αύξησης τα 150 sec. Στη συνέχεια παραμένει στις 16000 για 20 δευτερόλεπτα και έπειτα μειώνεται πάλι γραμμικά από 16000 έως 0 με διάστημα μείωσης ίσο με το πρώτο 150 δευτερόλεπτα. Η εύρεση της ώσης με μειούμενη την περιστροφική ταχύτητα χρησιμοποιείται για τη διαπίστωση τυχόν διαφορετικής απόκρισης του συστήματος προώσεως κατά τη μείωση των στροφών.

Συγκεντρωτικά οι εκτελούμενες συναρτήσεις δίνονται στον παρακάτω πίνακα.

A/A	Χρονικό διάστημα εφαρμογής συνάρτησης	Συνάρτηση
1	0 ~ 150 Sec	$16000 \cdot t / 150.000$
2	150 ~ 170 Sec	16000
3	170 ~ 320 Sec	$16000 - (16000 \cdot (t - 170.000)) / 150.000$

7.2) Πίνακας εκτελούμενων συναρτήσεων με χρήση της μεθόδου «Έλεγχου Στροφών» για τη διεξαγωγή του πειράματος ώσης.

Από τη φόρμα επαλήθευσης και γραφικής απεικόνισης του αρχείου αποστολής στην εφαρμογή «Δημιουργία αρχείου αποστολής τιμών αναφοράς» εμφανίζονται τα παρακάτω στιγμιότυπα των δημιουργημένων αρχείων.



Σχ 7.1) Αύξηση της δράσης ελέγχου από 0 ~ 100% του αρχείου αποστολής. Σχ 7.2) Μείωση της δράσης ελέγχου από 100% ~ 0 του αρχείου αποστολής.



Σχ 7.3) Αύξηση της δράσης ελέγχου από 0 ~ 16000 του αρχείου αποστολής. Σχ 7.4) Αύξηση της δράσης ελέγχου από 16000 ~ 0 του αρχείου αποστολής.

- Σημειώσεις και παρατηρήσεις επί των ρυθμίσεων της εργαστηριακής διάταξης.

Πρώτον κατά την υλοποίηση και εκτέλεση του πειράματος μέτρησης της ώσης παρατηρήθηκε ότι η βέλτιστη στάθμη γεμίσματος της δεξαμενή νερού είναι μεταξύ του $\frac{1}{4}$ και $\frac{3}{4}$ του ύψους κάλυψης της οπής εξαγωγής του ύδατος από τον προωθητήρα ανάδρασης. Αν η στάθμη είναι χαμηλότερη από το $\frac{1}{4}$ τότε ο προωθητήρας δεν ήταν δυνατό να αναρροφήσει το νερό ενώ μεγαλύτερη στάθμη από $\frac{3}{4}$ είχε ως αποτέλεσμα την εμφάνιση έντονων κυματισμών κατά την επίτευξη υψηλών στροφών λειτουργίας του κινητήρα. Οι κυματισμοί αλληλοεπιδρούσαν με το κουτί στο οποίο είναι προσαρμοσμένο το σύστημα προώσεως, με αποτέλεσμα την έντονη μετακίνηση του, από τη μονάδα δυναμό-μέτρησης του «LoadCell». Η απώλεια της επαφής μεταξύ τους είχε ως αποτέλεσμα τη μετακίνηση αρχικά μακριά από το Loadcell και στη συνέχεια εκτελώντας την αντίθετη κίνηση την κρούση επάνω σε αυτό δίνοντας ακραίες τιμές.

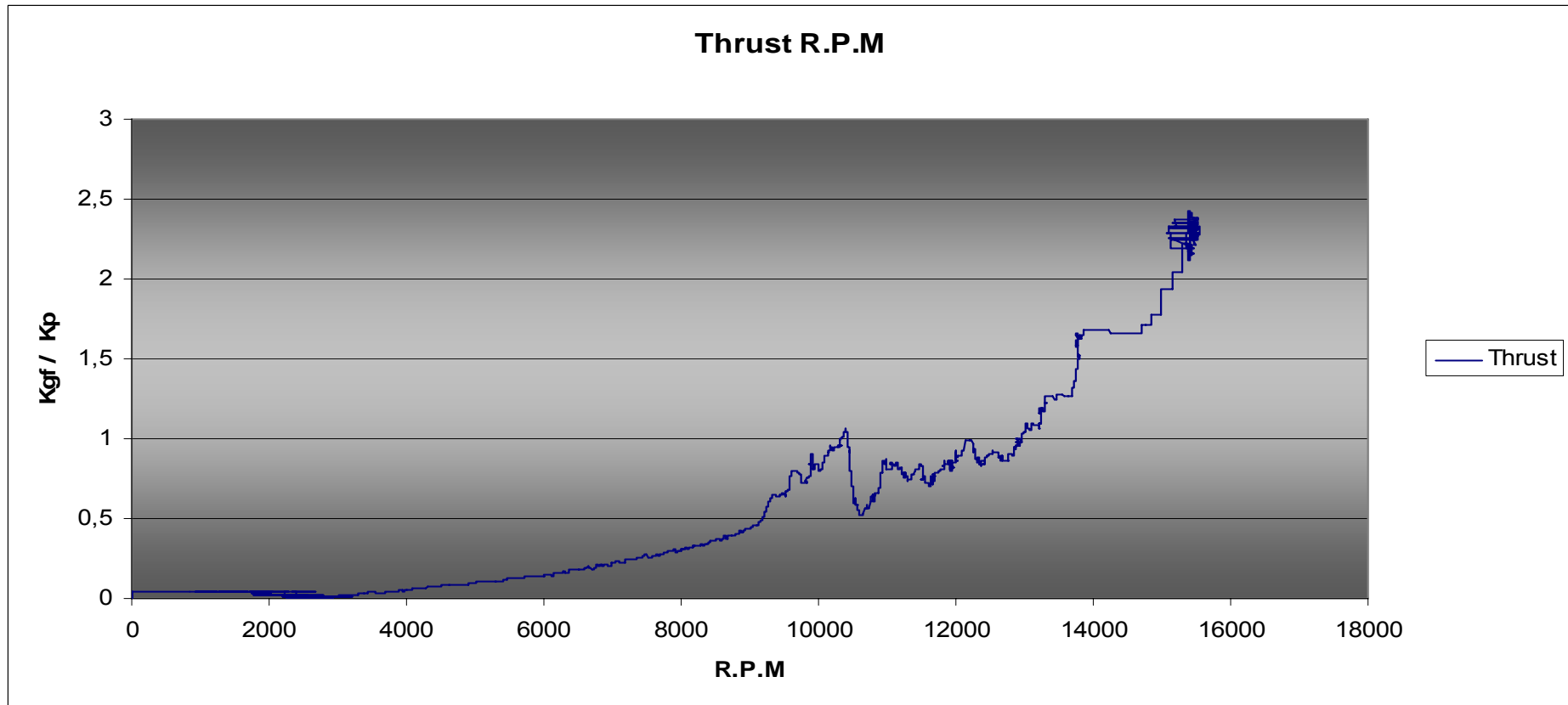
Δεύτερον κατά τη διεξαγωγή των μετρήσεων με τη χρήση της στάθμης του ύδατος εντός του προαναφερθέντος σωστού πεδίου, παρατηρήθηκε ότι προσθέτοντας βάρος εντός του κουτιού, που περιέχει τον προωθητήρα ανάδρασης, η επίδραση των κυματισμών στις μετρήσεις ήταν μειωμένη. Ο λόγος για την παραπάνω επίδραση έγκειται στο γεγονός ότι ο όγκος του κουτιού είναι αρκετά μεγάλος με αποτέλεσμα στη χρησιμοποιούμενη στάθμη η άνωση να εξισορροπεί κατά ένα μεγάλο ποσοστό το βάρος της μεταλλικής κατασκευής με αποτέλεσμα οι κυματισμοί να ανασηκώνουν και να μετακινούν το φορείο. Όμως η αύξηση του βάρους είχε σαν αρνητικό αποτέλεσμα τη δημιουργία σημαντικής τριβής στα έδρανα κύλισης. Το αποτέλεσμα της τριβής αυτής είναι ότι κατά τη μείωση των στροφών του κινητήρα παρατηρείται μια εναπομένουσα δύναμη θλίψης στο load cell η οποία επιδρά στη μέτρηση. Τέλος κατά τη διεξαγωγή των μετρήσεων παρατηρήθηκε έντονη πτώση της τάσης του τροφοδοτικού προς το παλμοτροφοδοτικό μετατροπέα. Σημειώνεται ότι κατά την εφαρμογή δράσης ελέγχου ίση με 100% η τάση του τροφοδοτικού έπεσε στα 17 Volt. Η πτώση τάση αυτή είναι ιδιαίτερα εμφανής κατά σε όλο το εύρος των στροφών και αυξάνεται με την αύξηση του φορτίου του κινητήρα τείνοντας στη μέγιστη τιμή της στις μέγιστες στροφές.

7.4) Παρουσίαση των αποτελεσμάτων από τη διεξαγωγή των μετρήσεων.

Από τη μέτρηση της ώσης θα εξαχθεί η καμπύλη ισχύος στροφών με τη χρήση της σχέσης 7.17. Σημειώνεται ότι η επιφάνεια A_7 είναι ίση με $A_7 = \pi * R^2 = \pi * (0,016)^2 = 8,0424 \cdot 10^{-4} m^2$, όπου $R=0.016$ m είναι η διάμετρος του σωλήνα εξόδου προωθητήρα ενώ η χρησιμοποιούμενη πυκνότητα του νερού είναι $\rho = \frac{1000Kg}{m^3}$. Τα κατασκευαστικά χαρακτηριστικά δίνονται στα βοηθήματα.

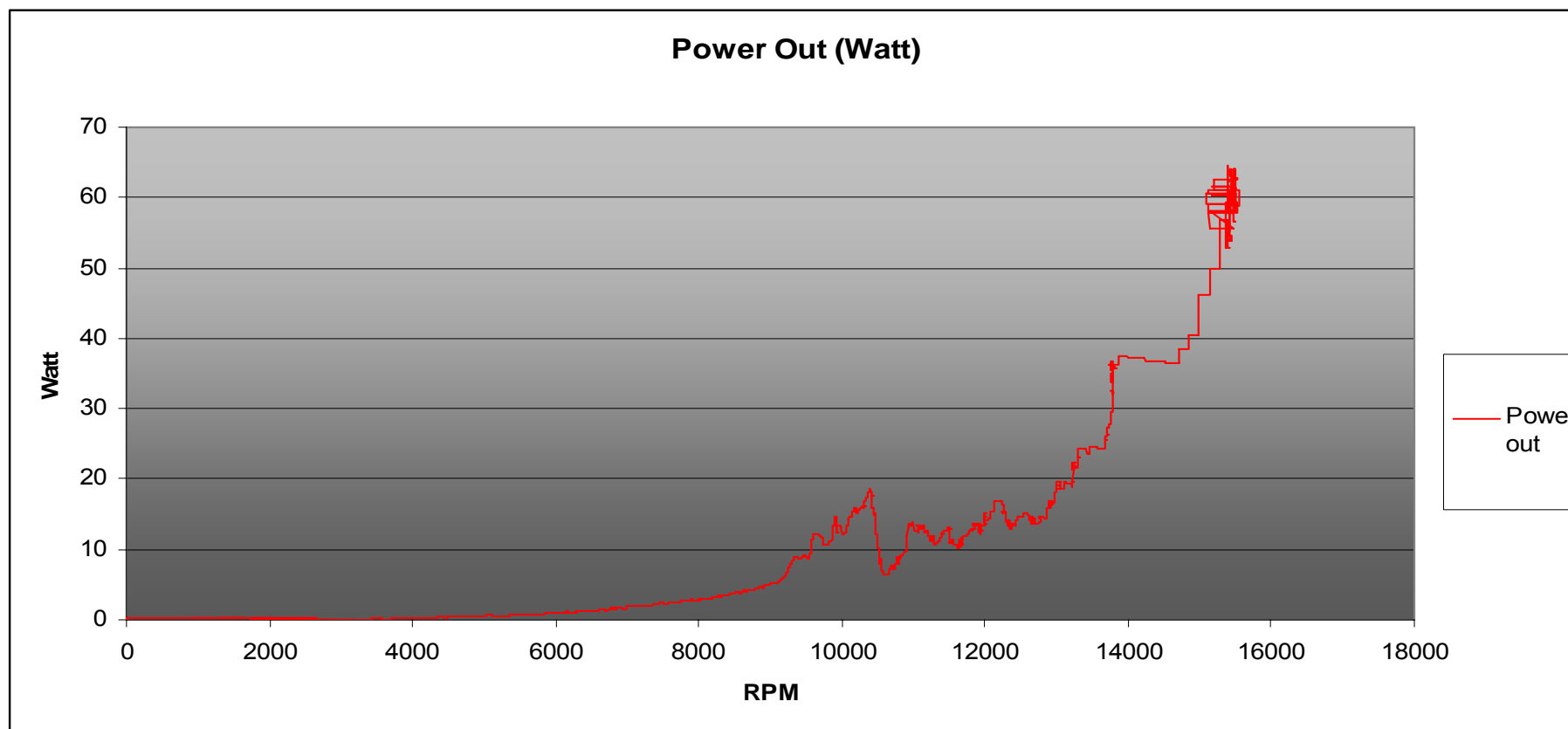
Επομένως η σχέση 7.17 γράφεται:

$$P_{Out}(N) = \frac{F(N)^{\frac{3}{2}} \cdot \left(\frac{Kg \cdot m}{s^2}\right)^{\frac{3}{2}}}{2 \cdot \sqrt{1000 \cdot \pi \cdot 0,016^2}} \cdot \frac{1}{\sqrt{\frac{kg}{m^3} m^2}} \text{ (watt)} \quad (7.18)$$



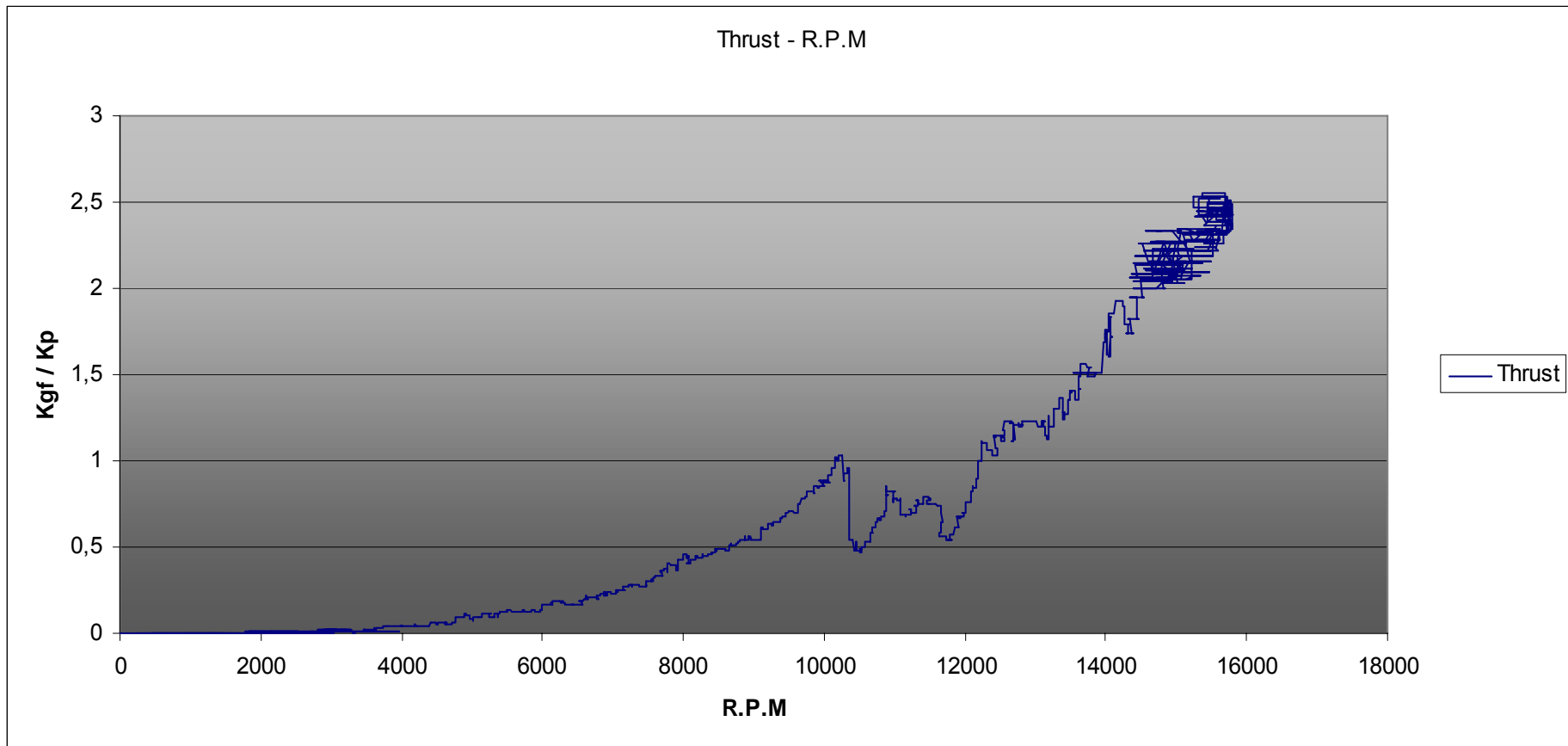
Σχ 7.5) Διάγραμμα ώσης - στροφών με τη χρήση του αρχείου της μεθόδου «Κανόνας ελέγχου»

Το παραπάνω διάγραμμα μετασχηματίζεται με τη χρήση της σχέσης 7.18 στο διάγραμμα Εξαγόμενης Ισχύος και στροφών έλικας και δίνεται στο παρακάτω σχήμα.



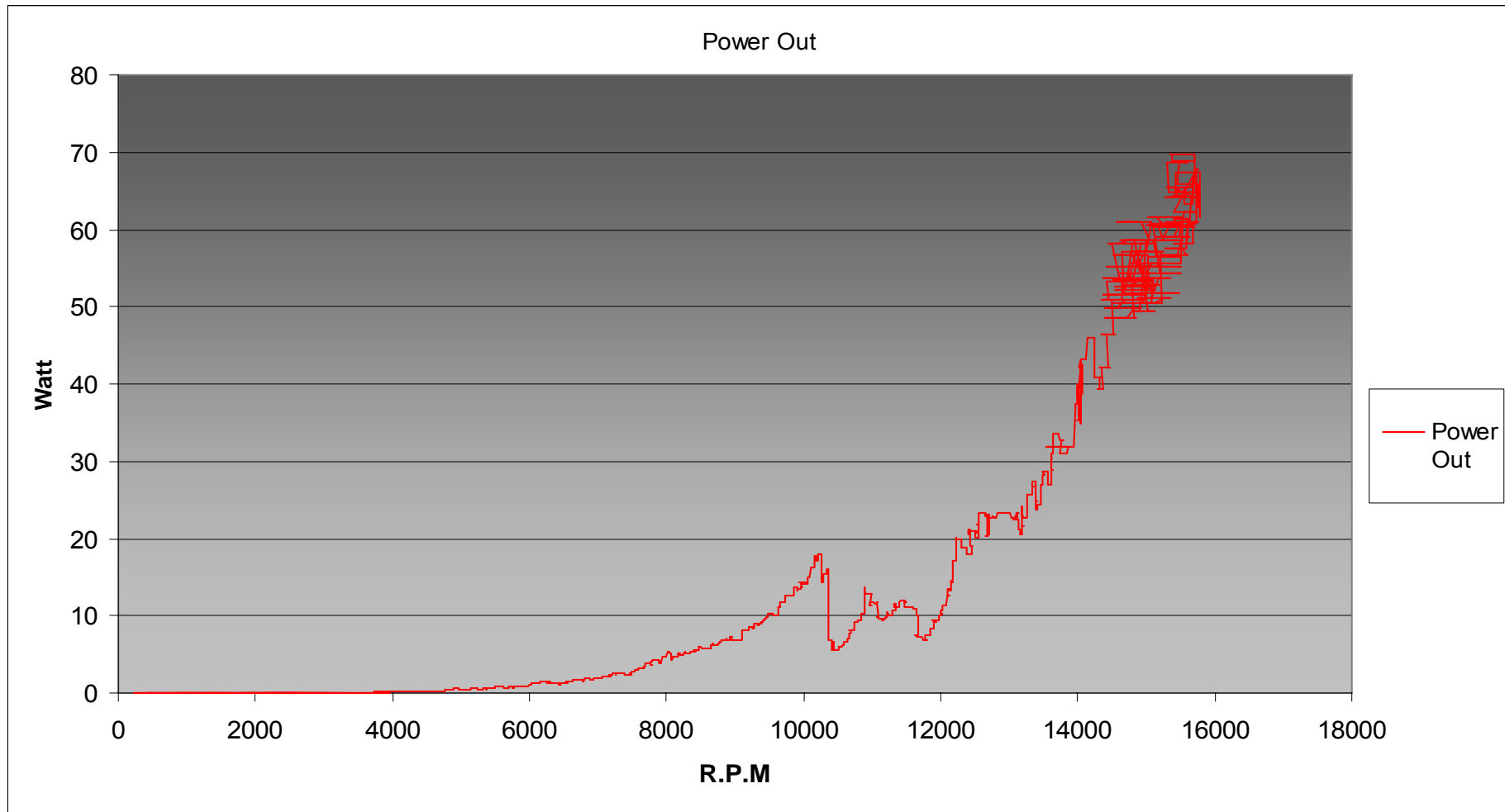
Σχ. 7.6) Διάγραμμα Ισχύος - στροφών με τη χρήση του αρχείου της μεθόδου «Κανόνας ελέγχου»

Στα παρακάτω διάγραμμα παρουσιάζεται τα αποτελέσματα των μετρήσεων κατά τη διεξαγωγή της μέτρησης με την εκτέλεση του αρχείο το οποίο δημιουργείται με τη μέθοδο του ελέγχου στροφών .



Σχ 7.7) Διάγραμμα ώσης - στροφών με τη χρήση του αρχείου της μεθόδου «Έλεγχος τροφών»

Ομοίως με χρήση της σχέσης 7.18 προκύπτει το παρακάτω διάγραμμα Ισχύος – Στροφών.



Σχ 7.8) Διάγραμμα Ισχύος - στροφών με τη χρήση του αρχείου της μεθόδου «Έλεγχος Στροφών».

7.5) Σχολιασμός των αποτελεσμάτων της μέτρησης ώσης.

Παρατηρώντας τα διαγράμματα 7.5 και 7.6 διακρίνονται εύκολα 3 περιοχές στις οποίες έχουμε έντονη αλλαγή στα χαρακτηριστικά της καμπύλης. Η πρώτη περιοχή διακρίνεται στο διάστημα ταχυτήτων λειτουργίας 0 έως περίπου 10500 Σ.Α.Λ «Στροφών ανά Λεπτό». Στην περιοχή αυτή παρατηρείτε ότι η καμπύλη ώσης έχει τετραγωνική μορφή. Επαναλαμβάνοντας την εκτέλεση του πειράματος παρατηρήθηκε επανάληψη της ίδιας μορφής καμπυλών και ίδιες περιοχές διαφορετικής μορφής.

Δεύτερη περιοχή είναι μεταξύ των στροφών 10500 και 12000 στην οποία παρατηρείτε έντονη μείωση της ώσης με αύξηση των στροφών. Το «παράδοξο» αυτό στην πραγματικότητα οφείλεται στην αλλαγή του είδους της ροής εντός του προωθητήρα και στην έλικα κατά την οποία δημιουργείται σπηλαίωση. Στην περιοχή αυτή Σ.Α.Λ παρατηρείται και έντονη αλλαγή της μορφής του εξαγόμενου ύδατος το οποίο πλέον παρουσιάζεται με έντονες φυσαλίδες.

Τρίτη περιοχή ορίζεται αυτή από τις 12000 έως τις 15500 Σ.Α.Λ στην οποία παρατηρείται πλέον η αύξηση της ώσης με εντονότερη κλίση από αυτή της πρώτης περιοχής. Στην περιοχή αυτή των στροφών πλέον έχει επιτευχθεί πλήρης σπηλαίωση και είναι δυνατή η ανάπτυξη ώσης.

Σημειώνεται ότι όταν ο κινητήρας λειτουργούσε στην περιοχή των στροφών 12000 μέχρι 15000 αναπτύσσονταν έντονος κυματισμός ο οποίος επηρέαζε τη μέτρηση της ώσης αλληλεπιδρώντας με το κουτί του συστήματος πρόωσης. Η αλληλεπίδραση αυτή είναι εμφανής στις μετρήσεις με τη σχετική έντονη διακύμανση των μετρήσεων σε σχέση με την πρώτη περιοχή Σ.Α.Λ των μετρήσεων από 0 έως 10000. Επίσης η διακύμανση των μετρήσεων αποδίδεται και στην υπέρ τάχυνση του κινητήρα κατά την αναρρόφηση φυσαλίδων αέρα οι οποίες συσσωρεύονται κοντά στην εισαγωγή του προωθητήρα και όταν ξεπεράσουν έναν μέγεθος τότε αναρροφούνται με αποτέλεσμα το «ξενέρισμα της έλικας και την απότομη απώλεια του φορτίου και τελικά τη στιγμιαία μείωση της ώσης.

Η τελική εξαγόμενη ισχύς του προωθητήρα μπορεί να θεωρηθεί μικρή όμως πρέπει να σημειωθεί ότι ο βαθμός απόδοσης το συστημάτων πρόωσης αυτού του τύπου αυξάνει με την αύξηση της ταχύτητας εισαγωγής του ύδατος. Το πείραμα και η μέτρηση διεξάγεται με μηδενική ταχύτητα εισαγωγής του ύδατος και αναμένεται μικρός βαθμός απόδοσης και τελικής ωφέλιμης εξαγόμενης ισχύς.

8) Ανακεφαλαίωση συμπεράσματα και προτάσεις επί της πτυχιακής.

8.0) Σύνοψη.

Καθώς η πτυχιακή αυτή περιλαμβάνει πολλούς διαφορετικούς τομείς στους οποίους επεκτείνεται κρίνεται αναγκαίο στο τελευταίο αυτό κεφάλαιο να γίνει μια συνοπτική παρουσίαση όλων όσων γράφθηκαν ώστε να υπενθυμιστούν και να αλληλοσυμπληρωθούν τα χρησιμοποιούμενα αποτελέσματα και συμπεράσματα κάθε κεφαλαίου για τη σύνθεση της ολοκληρωμένης εικόνας της πτυχιακής. Στη συνέχεια παρουσιάζονται τα προβλήματα που παρουσιάστηκαν κατά τη διάρκεια εκπόνησης αυτής της πτυχιακής και οι λύσεις που επιλέχθηκαν. Τέλος αναφέρονται συνοπτικά και κρίνονται τα αποτελέσματα των μετρήσεων ενώ επιπλέον παρουσιάζονται οι προτάσεις για την περαιτέρω ανάπτυξη και τους τομείς εφαρμογής της πτυχιακής αυτής.

8.1) Συνοπτική παρουσίαση των κεφαλαίων της πτυχιακής

Στο πρώτο κεφάλαιο πραγματοποιήθηκε μια σύντομη παρουσίαση της **δεξαμενής** στην οποία πραγματοποιήθηκαν τα πειράματα. Για τις ανάγκες των πειραμάτων μοντελοποιήθηκε ο **ηλεκτρικός κινητήρας** και εξήχθησαν τα μέτρα των συντελεστών K_e, R_a . Η μορφή της μοντελοποίησης αυτή συναντάτε στην πλειονότητα της βιβλιογραφίας και αποτελεί συνήθη πρακτική για του κινητήρες συνεχούς τάσης. Προσφέρει μια ικανοποιητική προσέγγιση των ηλεκτρικών χαρακτηριστικών του κινητήρα. Από τους παραπάνω συντελεστές υπολογίστηκε η σχέση της αναπτυσσόμενης ροπής του κινητήρα συναρτήσει της τάσης τροφοδοσίας του, και της περιστροφικής του ταχύτητας ενώ παράλληλα επέτρεψε τον υπολογισμό των απωλειών τριβής του συνολικού συστήματος πρόωσης. Η γνώση των σχέσεων αυτών είναι απαραίτητη για την υλοποίηση του πειράματος μέτρησης της ροπής αδρανείας όπως γίνεται αντιληπτό στο έκτο κεφάλαιο. Στη συνέχεια παρουσιάστηκαν και αναπτύχθηκαν οι βασικές σχέσεις λειτουργίας του, συστήματος τροφοδοσίας της συνεχούς τάσης της εγκατάστασης. Η κατανόηση της λειτουργίας της μη ρυθμιζόμενης τροφοδοσίας «μετασχηματιστής» βοηθάει στην αντίληψη της επιρροής της στο κινητήρα και της συμπεριφοράς του στα πειράματα. Εξίσου σημαντική είναι η πλήρης κατανόηση της λειτουργίας του **παλμοτροφοδοτικού μετατροπέα** με τον οποίο γίνεται δυνατός ο έλεγχος του κινητήρα χρησιμοποιώντας μια «σταθερή» τάση. Αυτή η μέθοδος ρύθμισης, συναντάτε συχνά στη βιβλιογραφία και στην κοινή καθημερινή πράξη όπου απαιτείται η μετατροπή της τιμής μια συνεχούς τάσης ή η μετατροπή της σε εναλλασσόμενη «Inverter». Επίσης υπολογίζονται οι βασικές σχέσεις απόκρισης του κινητήρα και του αντίστοιχου ηλεκτρικού ισοδύναμου κυκλώματος όταν εφαρμόζεται ο μετατροπέας. Από τις σχέσεις αυτές κρίνεται στο κεφάλαιο έξι αν τα χαρακτηριστικά λειτουργίας του μετατροπέα ταιριάζουν με αυτά του συστήματος πρόωσης. Τέλος παρουσιάστηκε ο χρησιμοποιούμενος **υπερύθρος φωτοανιχνευτής** των περιστροφών του κινητήρα. Αξιοσημείωτη είναι η χρήση ενός φυσικού χαρακτηριστικού «οπής», του συνδέσμου κινητήρα και προωθητήρα. Επιπλέον για την ορθή μέτρησης των στροφών αναπτύχθηκε μια ηλεκτρονική διάταξη με σκοπό το φιλτράρισμα του εξαγόμενου παλμού από το φωτοανιχνευτή. Το **φίλτρο** αυτό δεν χρησιμοποιεί την κοινή πρακτική φιλτραρίσματος των ανεπιθύμητων συχνοτήτων εντός του σήματος, μέσω κυκλώματος «RC» αλλά

απορρίπτει όλα τα σήματα όταν το πλάτος τους δεν ξεπερνά μία ορισμένη τιμή. Με το μέθοδο αυτή μπορούν να διέλθουν όλα τα σήματα ανεξαρτήτως της συχνότητας τους εξαλείφοντας την ανάγκη ρύθμισης του φίλτρου. Έτσι μπορεί να χρησιμοποιηθεί σε οποιοδήποτε σήμα αρκεί το πλάτους του να ικανοποιεί της απαιτήσεις του φίλτρου .

Στο δεύτερο κεφάλαιο για λόγους οι οποίοι εξηγούνται στην εισαγωγή και γίνονται κατανοητοί πλήρως στο κεφάλαιο έξη και επτά αναπτύσσεται εν συντομία η **θεωρία των ελεγκτών τριών όρων** στα συνεχή γραμμικά συστήματα. Η θεωρία δε φέρει κάτι νέο απλά αποτελεί το συνδυαστικό κρίκο ανάμεσα στον ελεγκτή και την υλοποίηση του στο πεδίο του διακριτού χρόνου στο μικροεπεξεργαστή. Δεν εξετάζεται η ευστάθεια του διότι δεν ανήκει στο σκοπό της πτυχιακής αυτής και οι τιμές των όρων του προσδιορίζονται με τη διαδικασία δοκιμών. Στη συνέχεια με βάση τη σχέση του ελεγκτή στο συνεχή χρόνο εξάγονται οι σχέσεις του ελεγκτή που θα υλοποιηθούν στο μικροεπεξεργαστή. Χωρίς τη γνώση των παραπάνω βασικών σχέσεων ο αλγόριθμος υλοποιήσεις του ελεγκτή δε θα ήταν πλήρως κατανοητός.

Στο τρίτο κεφάλαιο γίνεται εκτεταμένη αναφορά στα χαρακτηριστικά του χρησιμοποιημένου **μικροεπεξεργαστή** και των δυνατοτήτων που παρέχει. Στη συνέχεια για τη λειτουργία και χρήση της **LPC2106** απαιτούνται να πραγματοποιηθούν και να υπολογιστούν ορισμένες βασικές **ρυθμίσεις** της με γνώμονα τις ανάγκες και τους σκοπούς της πτυχιακής. Στο κεφάλαιο αυτό εξηγούνται αναλυτικά ο τρόπος ρύθμισης και λειτουργίας των καταχωρητών «registers» του επεξεργαστή. Επίσης παρουσιάζονται σε βάθος το σκεπτικό για την εκλογή των ρυθμίσεων ξεχωριστά της κάθε χρησιμοποιούμενης περιφερειακής διάταξης του επεξεργαστή. Τονίζεται ότι το πρόγραμμα σχεδιάστηκε εξ αρχής ώστε να αξιοποιεί πλήρως τη δυνατότητα της χρήσης των διακοπών ροής του προγράμματος για την εξυπηρέτηση των αναγκών λειτουργιών των περιφερειακών συσκευών του επεξεργαστή. Το αποτέλεσμα είναι η δημιουργία ενός ευέλικτου προγράμματος στο οποίο, όλες οι διαδικασίες επικοινωνίας ,ο έλεγχος του παλμοτροφοδοτικού μετατροπέα και ο ελεγκτής είναι πλήρως ανεξάρτητες μεταξύ τους. Έτσι η ροή του προγράμματος δεν είναι προδιαγεγραμμένη αυστηρά και η εκτέλεση κάθε διαδικασίας πραγματοποιείται μόνο όταν αυτή είναι απαραίτητη. Τελικά επιτυγχάνεται η βέλτιστη δυνατή οικονομία της επεξεργαστικής ισχύος και η **σταθερότητα** του κώδικα. Ωστόσο η ευελιξία της ροής του προγράμματος εισάγει αταξία στην εκτέλεση των διακοπών με αποτέλεσμα την απροσδιοριστία της ροής του προγράμματος. Η αταξία αυτή επιλύεται με την ιεράρχηση των αιτημάτων βάση της σπουδαιότητας τους ώστε όταν εκκρεμούν ταυτόχρονα δύο ή περισσότερα αιτήματα διακοπής να κατευθύνεται η ροή στην εκτέλεση της διακοπής με τη μεγαλύτερη τιμή. Επιπλέον για κάθε βασική πηγή αιτήματος διακοπής ,που είναι οι περιφερειακές λειτουργίες του επεξεργαστή, ορίζονται και εσωτερικές αυτών. Έτσι λοιπόν για την ορθή εκτέλεση των εσωτερικών διακοπών ορίστηκε η εσωτερική προτεραιότητα. Η προτεραιότητα αυτή δεν ορίζεται από κάποια τιμή αλλά από τη σειρά γραφής των εκτελούμενων συναρτήσεων των εσωτερικών διακοπών εντός της συνάρτησης που εκτελείται κατά τη διακοπή της αντίστοιχης περιφερειακής συσκευής «**Interrupt Service Routine**». Κατά την παρουσίαση των περιφερειακών λειτουργιών του επεξεργαστή εκτός από τις ρυθμίσεις τους δίνεται και η κατανομή των εσωτερικών προτεραιοτήτων αυτών. Εν' κατακλείδι παρουσιάζεται η κατανομή των προτεραιοτήτων των διακοπών εσωτερικών και μη σε μορφή σχεδιαγράμματος.

Στο τέταρτο κεφάλαιο για την επίτευξη της λειτουργίας της εγκατάστασης, αναπτύσσονται υπό τη μορφή αλγορίθμων ορισμένες διαδικασίες. Ο λόγος της παρουσίας τους στο επίπεδο αλγορίθμων, είναι η κατανόηση από τον αναγνώστη της λειτουργίας των συναρτήσεων αυτών, χωρίς να είναι απαραίτητη η γνώση της γλώσσας προγραμματισμού στην οποία είναι γραμμένες. Για την αποστολή των δεδομένων από και προς τον υπολογιστή δημιουργήθηκαν δυο πρωτόκολλα. Με σκοπό την εξοικονόμηση των μεταφερόμενων δεδομένων δεν χρησιμοποιήθηκε η συνήθης πρακτική της αποστολής χαρακτήρων αλλά τα δεδομένα ομαδοποιούνται σύμφωνα με τα πρωτόκολλα. Στη συνέχεια στέλνεται η δυαδική μορφή αυτής της ομάδας χωρισμένη σε σειρά από «Bytes». Η μέθοδος αυτή κατάφερε να μειώσει τον αριθμό των διακινούμενων Bytes δια της σειριακής. Εξακολούθως παρουσιάζεται ο αλγόριθμος λήψης των μετρήσεων του συστήματος μέτρησης ώσης «Load cell». Για την ανάγκη ελέγχου της εγκατάστασης μέσω του υπολογιστή ορίστηκαν εντολές οι οποίες αναγνωρίζονται από τον κώδικά της LPC2106 και παρουσιάζονται υπό μορφή πίνακα μαζί με τη λειτουργία κάθε εντολής. Στη συνέχεια ακολουθεί ο αλγόριθμος υπολογισμού του χρόνου λήψης των μετρήσεων που στέλνεται από την LPC2106 στον υπολογιστή μαζί με τις μετρήσεις. Τελικά κλείνει το κεφάλαιο με τον αλγόριθμο μέτρησης των στροφών του κινητήρα. Χρησιμοποιείται μια σύνθετη μέθοδος στην οποία τόσο οι μετρούμενες περιστροφές του κινητήρα όσο και η διάρκεια της μέτρησης δεν είναι σταθερές. Με τον τρόπο αυτό επιτυγχάνεται ακριβέστερη μέτρηση της ταχύτητας, μικρότερη της μιας στροφής ανά λεπτό. Αξίζει να σημειωθεί ότι για τη σταθεροποίηση της μέτρησης καθώς και μείωση της μικρότερης δυνατής ταχύτητας που μπορεί να μετρηθεί χρησιμοποιείται για τον υπολογισμό η μέθοδος του κινητού μέσου όρου. Στη μέθοδο αυτή χρησιμοποιείται ο μέσος όρος n παλαιότερων μετρήσεων με σκοπό την εξομάλυνση της μέτρησης από την επίδραση του τυχόν θορύβου. Η άθροιση των παλαιότερων μετρήσεων έχει σαν συνέπεια της αύξησης του διαστήματος δειγματοληψίας σε $n+1$ με αποτέλεσμα στη μείωση της ελάχιστης μετρούμενης ταχύτητας.

Στο πέμπτο κεφάλαιο παρουσιάζεται η εφαρμογή που αναπτύχθηκε και γράφηκε για τη χρήση στο υπολογιστή υπό το λειτουργικό των «Windows». Βασικός σκοπός αυτού του προγράμματος είναι ο έλεγχος της εγκατάστασης, διεξαγωγής και λήψης των μετρήσεων των πειραμάτων καθώς και της δημιουργίας των αρχείων τιμών τα οποία στέλνονται στην LPC2106 από το πρόγραμμα διεξαγωγής των μετρήσεων. Η αποστολή των τιμών αναφοράς από αρχείο αποσκοπεί στη διεξαγωγή των μετρήσεων με ακριβή χρονική σειρά και στη δυνατότητα επανάληψης τους. Η εφαρμογή που εκτελείται στον υπολογιστή είναι γραμμένη σε «Visual Basic». Για το λόγο αυτό παρουσιάζονται αρχικά οι αρχές προγραμματισμού του συγκεκριμένου περιβάλλοντος ώστε ο αναγνώστης του κώδικα να είναι σε θέση να τον αντιληφθεί. Τέλος πραγματοποιείται μια πλήρης βασική παρουσίαση της εφαρμογής στον υπολογιστή τελειώνοντας με την ερμηνεία των εμφανιζόμενων μηνύματα κατά τη χρήση της εφαρμογής.

Στο έκτο κεφάλαιο αναπτύσσεται η θεωρία της μέτρησης της ροπής αδράνειας του συνολικού συστήματος προώσεως. Πιο συγκεκριμένα για το σκοπό της μέτρησης της αδράνειας αναπτύσσονται δύο μέθοδοι. Πρώτη είναι η συνήθης πρακτική που συναντάται στη βιβλιογραφία με την εφαρμογή γνωστής ροπής στον κινητήρα και εκ' της επιτάχυνσης του την εύρεση αυτής. Δεύτερη αναπτύσσεται η μέθοδος της εφαρμογής ημιτονοειδούς μεταβαλλόμενης ροπής στον κινητήρα και από το πλάτος της ημιτονοειδούς μεταβαλλόμενης περιστροφικής ταχύτητας υπολογίζεται η ροπή

αδρανείας. Αξίζει να σημειωθεί ότι η δεύτερη μεθοδολογία δε συναντάτε στη βιβλιογραφία και αναπτύχθηκε για τους σκοπούς της πτυχιακής. Δε χρειάζεται η ακριβής γνώση των χρονικών τιμών της εφαρμοζόμενης ροπής παρά μόνο τα πλάτη της μεταβολής της και της μεταβολής των στροφών. Στη συνέχεια παρουσιάζονται τα προβλήματα της χρήσης του υπάρχοντος παλμοτροφοδοτικού μετατροπέα για τη διεξαγωγή της μέτρησης της αδράνειας. Λόγω των προβλημάτων αυτών σχεδιάστηκε και υλοποιήθηκε εκ' νέου νέος παλμοτροφοδοτικός μετατροπέας οποίος και παρουσιάστηκε. Λόγω των χρησιμοποιούμενων ρυθμίσεων του παλμοτροφοδοτικού και των προηγούμενων προβλημάτων τροποποιήθηκαν κατάλληλα οι σχέσεις υπολογισμού της ροπής αδράνειας της μεθόδου ημιτονοειδούς μεταβολής της ασκούμενης ροπής ώστε να ληφθεί υπόψη η επίδραση της μη ιδανικής λειτουργίας του παλμοτροφοδοτικού μετατροπέα. Η μετατροπή των σχέσεων αυτών είχε σαν αποτέλεσμα τη μείωση της ακρίβειας τους με αποτέλεσμα την εκτίμηση μόνο της τάξης της ροπής αδράνειας. Στη συνέχεια παρουσιάζεται η εκτέλεση του πειράματος μέτρησης καθώς και το αποτέλεσμα της μέτρησης.

Στο έβδομο κεφάλαιο εκτελείται η μέτρηση της ώσης του συστήματος πρόωσης. Εισαγωγικά παρουσιάζεται η θεωρία λειτουργίας των προωθητήρων ανάδρασης και εφαρμόζονται οι παραγόμενες σχέσεις για το χρησιμοποιούμενο προωθητήρα. Τελικά εξάγεται και η τελική σχέση που εκφράζει την εξαγόμενη ισχύ ώσης του προωθητήρα συναρτήσει της μετρούμενη ώσης. Στη συνέχεια παρουσιάζονται οι παράμετροι εκτέλεσης των μετρήσεων οι οποίες διεξάγονται με τη χρήση και των δύο μεθόδων ελέγχου του κινητήρα, κανόνας ελέγχου και έλεγχος στροφών. Τα αποτελέσματα των μετρήσεων παρουσιάζονται σε μορφή διαγραμμάτων ώσης συναρτήσει των στροφών ενώ επίσης με τη χρήση της σχέσης που αναπτύχθηκε στην αρχή του κεφαλαίου προκύπτουν τα διαγράμματα ισχύος ώσης - στροφών.

8.2) Παρουσίαση των προβλημάτων που παρουσιάστηκαν.

Στις προηγούμενες παραγράφους πραγματοποιήθηκε μια σύντομη περιγραφή όλων όσων έγιναν για την ολοκλήρωση της πτυχιακής αυτής. Όπως είναι φανερό τα πεδία που χρειάστηκε να επεκταθεί είναι ποικίλα με αποτέλεσμα και τα προβλήματα που παρουσιάστηκαν να είναι εξίσου πολλά. Μελετώντας τα μπορούν να χωριστούν σε τρεις κύριες κατηγορίες με πρώτη την κατηγορία του προγραμματισμού, δεύτερη αυτή των ηλεκτρονικών διατάξεων και τρίτη αυτή της διεξαγωγής των μετρήσεων.

Η πρώτη κατηγορία αφορά τον προγραμματισμό τόσο της κάρτας LPC2106 όσο και τη δημιουργίας της εφαρμογής στο υπολογιστή. Αναπτύσσοντας δύο προγράμματα ταυτόχρονα σε δύο διαφορετικές γλώσσες προγραμματισμού εισήγαγε πρόσθετες δυσκολίες καθώς απαιτούσε την ταυτόχρονη γνώση και χειρισμό και των δύο. Επιπλέον η απαίτηση της επικοινωνίας των δυο προγραμμάτων είχε σα συνέπεια την παράλληλη ανάπτυξης των δύο εφαρμογών, καθώς δε θα ήταν δυνατή η διαδικασία εκσφαλμάτωσης «debugging» σε κάθε στάδιο ανάπτυξης. Το αποτέλεσμα ήταν σημαντικές διαφοροποιήσεις στον κώδικα, όπως η αποστολή των τιμών αναφοράς της δράσης ελέγχου ή των στροφών από αρχείο, να απαιτεί σημαντικό χρόνο ανάπτυξης και εκσφαλμάτωσης καθώς έπρεπε να γραφεί ο κώδικας και για τα δύο προγράμματα και στη συνέχεια να βρεθούν τυχόν λάθη.

Στο σημείο αυτό πρέπει να αναφερθεί ότι ο σχεδιασμός του προγράμματος για τον υπολογιστή και γενικώς όλης της εργαστηριακής διάταξης έγινε με σκοπό τη χρήση του ως εργαστηριακό υλικό εκτέλεσης πειραμάτων και παρουσίασης στο πλαίσιο μαθημάτων. Ο παραπάνω σκοπός οδήγησε στην ανάπτυξη της εφαρμογής του υπολογιστή, όσο το δυνατόν περισσότερο, παρέχοντας στο φοιτητή που θα διαχειρίζεται την εργαστηριακή διάταξη τη μεγαλύτερη ευελιξία καθώς και εργαλεία για την επίτευξη του στόχου του. Ο κώδικας επεκτάθηκε τόσο που πλέον ξεπέρασε της **10.000 γραμμές** με σημαντικό αντίκτυπο στο χρόνο ανάπτυξης. Η εφαρμογή είναι γραμμένη εζ' αρχής και δεν αποτελεί την επέκταση κάποιας προηγούμενης. Επιπλέον η δημιουργία γραφικών παραστάσεων καθώς και η δημιουργία αρχείων τιμών, αποτελούν τμήματα εμπορικών εφαρμογών κλειστού κώδικα με σημαντική χρηματική αξία. Η ανάπτυξη των γραφικών έγινε από την αρχή χωρίς την παραμικρή ιδέα πως μπορεί να γίνει αυτό και επιπλέον κατέστη δυνατή η χρήση μικρής επεξεργαστικής ισχύος. Φυσικά όταν δημιουργείται κάτι νέο τα προβλήματα που παρουσιάζονται είναι πολλά δε μπορούν να παρουσιαστούν καθώς θα έπρεπε να γραφούν πολλές σελίδες. Ωστόσο ο κώδικας στα πλαίσια της πτυχιακής είναι ανοικτός και η οποιαδήποτε μπορεί να το χρησιμοποιήσει ως βάση εκκίνησης.

Ο προγραμματισμός της **LPC2106** αντίστοιχα είχε και αυτός τα δικά του προβλήματα . Το σημαντικότερο όμως ήταν η εξαγωγή του πηγαίου κώδικα μηχανής και εκτέλεση του στην LPC2106. Όπως και για την εφαρμογή στον υπολογιστή έτσι και εδώ η γραφή του προγράμματος της κάρτας πραγματοποιήθηκε εκ 'του μηδενός . Η χρήση του επεξεργαστή αυτού δε συναντάτε σε παρόμοια προγράμματα ή στο πεδίο του αυτόματου έλεγχου .Για τη γραφή του προγράμματος χρειάστηκε η γνώση σε βάθος και κατανόηση της γλώσσας **C++** με σκοπό τη δημιουργία σύνθετων δομών δεδομένων και συναρτήσεων, για το πέρασμα των σκοπέλων που έθεταν οι εν' γενεί αδυναμίες της χρησιμοποιούμενης εφαρμογής γραφής του κώδικα .Η λειτουργία του μεταφραστή αυτού δεν ήταν εξειδικευμένη για τέτοιες εφαρμογές και επιπροσθέτως πρόκειται για μια εμπορική περιορισμένη έκδοση. Αυτή η έλλειψη πληρότητας θέτει περιορισμό στα διαθέσιμα εργαλεία και αφετέρου οριοθετεί το μέγεθος τους εκτελέσιμου κώδικα ίσο με **32KB** ,το οποίο αποδείχθει αρκετά μικρό. Για το λόγο αυτό χρειάστηκαν αναρίθμητες συνεχείς αλλαγές στον κώδικα μέχρι να καταστεί δυνατή η έκδοση ενός σταθερού εκτελέσιμου κώδικα. Σημειώνεται ότι ενώ ο κώδικας εκτελείτο σωστά στον επεξεργαστή με τη χρήση του συστήματος εκσφαλμάτωσης «JLink» ο αντίστοιχος δημιουργημένος εκτελέσιμος πηγαίος κώδικας παρουσίαζε απρόβλεπτα σφάλματα .Αξίζει να σημειωθεί ότι η σχεδίαση της ροής του προγράμματος έγινε με βάση την ευελιξία που προσέφεραν οι διακόπτες ροής ενώ οι ρυθμίσεις της κάρτας έγιναν εξαρχής. Επιπλέον η σχεδίαση του προγράμματος μέτρησης των στροφών προσέφερε υψηλή ακρίβεια ενώ τα πρωτόκολλα επικοινωνιών σημαντική οικονομία και ασφάλεια των μεταφερόμενων δεδομένων. Τέλος η κατανομή των προτεραιοτήτων των διακοπών «**Interrupts**» με σκοπό την εύρεση του βέλτιστου συνδυασμού απόδοσης και ασφάλειας απαίτησε νέες ιδέες και πολλές ώρες δοκιμών και αλλαγών στον κώδικα.

<i>Προβλήματα προγραμματισμού</i>	
<i>Πρόβλημα</i>	<i>Λύση</i>
<i>Δύο διαφορετικές γλώσσες προγραμματισμού.</i>	<i>Παράλληλη ανάπτυξη των δύο εφαρμογών.</i>
<i>Ιδιαίτερα εκτεταμένη εφαρμογή επί του υπολογιστή μεγαλύτερη των</i>	<i>Απαιτήθηκε ιδιαίτερα σημαντικός χρόνος για την ανάπτυξη του.</i>

<i>10.000 γραμμών κώδικα.</i>	
<i>Αδυναμία εξαγωγείς σωστού εκτελέσιμου κώδικα από το πρόγραμμα IAR Embedded Workbench IDE</i>	<i>Βελτιστοποίηση της γραφής και δοκιμές μέχρι να επιλυθεί.</i>
<i>Ευελιξία του προγράμματος της LPC2106.</i>	<i>Χρήση «Interrupts» του επεξεργαστή.</i>

Πίνακας 8.1) Προβλήματα και λύσεις στον προγραμματισμό.

Στη δεύτερη κατηγορία ενσωματώνονται όλα τα προβλήματα που αφορούν τις ηλεκτρονικές διατάξεις του υπολογιστή, της κάρτας και των λοιπών μετρήσεων. Ένα σημαντικό πρόβλημα ήταν η χρήση απαριθμητών καλωδίων για την επικοινωνία των συστημάτων τα οποία καθιστούσαν δυσχερή τη χρήση τους και πολλές φορές δημιουργούσαν προβλήματα και κινδύνους καταστροφής των ηλεκτρονικών λόγω βραχυκυκλωμάτων. Το πρόβλημα αυτό λύθηκε με την κατασκευή ιδίων καλωδίων και ενσωμάτωσης των ηλεκτρονικών διατάξεων εντός κουτιών προφύλαξης. Εν' συνεχεία απαιτήθηκε ο σχεδιασμός και κατασκευή **φίλτρου** για το σήμα του **υπέρυθρου φωτοανιχνευτή**. Επιπλέον για τη **μέτρηση των στροφών** απαιτήθηκε η χρήση διάταξης μετρητή και συστήματος μετάδοσης της τιμής του στην κάρτα. Το πρόβλημα ήταν η έλλειψη του μετρητή «Counter». Προσπελάστηκε με την ανάπτυξη προγραμματιστικού μετρητή και τη χρήση λειτουργίας καταγραφής της στιγμής ανόδου του σήματος. Αυτή η προσέγγιση αναπτύχθηκε εξ' αρχή στα πλαίσια της πτυχιακής και είναι πρωτότυπη. Ενώ η **ακρίβεια** που επιτυγχάνεται είναι εξαιρετικά μεγάλη, **δέκατα στροφών** ανά λεπτό. Σημαντικό πρόβλημα που εμφανίστηκε ήταν η λειτουργία του παλμοτροφοδοτικού μετατροπέα. Ο μετατροπέας αυτός προερχόταν από τηλεκατεύθυνση οποίος χρησιμοποιούσε ένα ψηφιακό επεξεργαστή για την ανάλυση του σήματος της και εν' συνεχεία της δημιουργία παλμού τροφοδοσίας. Επιπλέον διέθετε τη δυνατότητα προγραμματισμού. Όμως το γεγονός της έλλειψης του εγχειριδίου προγραμματισμού και αφετέρου ότι ήταν σχεδιασμένος να λειτουργεί υπό την τροφοδοσία συστοιχίας μπαταριών καθιστούσε τον έλεγχο του δυσχερή. Έτσι σχεδιάστηκε **νέος παλμοτροφοδοτικός μετατροπέας** ο οποίος οδηγείτε πλέον απευθείας από το δημιουργούμενο παλμό «PWM» του επεξεργαστή της κάρτας. Αποτέλεσμα εκτός τις επίλυσης του παραπάνω προβλήματος είναι και ο έλεγχος των χαρακτηριστικών του παλμού ισχύς, τη συχνότητα του οποίου τη μεταβάλλουμε κατά βούληση και δεύτερον η εξάλειψη της ανάγκης **υδρόψυξης** του παλμοτροφοδοτικού.

Προβλήματα ηλεκτρονικών διατάξεων	
Προβλήματα	Λύσεις
<i>Έλλειψη ολοκληρωμένου μετρητή «Counter»</i>	<i>Δημιουργία προγραμματιστικού μετρητή</i>
<i>Θόρυβος στο σήμα του φωτοανιχνευτή.</i>	<i>Κατασκευή φίλτρου</i>
<i>Αδυναμία ελέγχου του παλμοτροφοδοτικού μετατροπέα, με μικροεπεξεργαστή, της τηλεκατεύθυνσης.</i>	<i>Κατασκευή νέου αναλογικού παλμοτροφοδοτικού μετατροπέα, αποδοτικότερου, ο οποίος οδηγείται απευθείας από τον επεξεργαστή της LPC2106.</i>
<i>Υδρόψυξη παλμοτροφοδοτικού μετατροπέα τηλεκατεύθυνσης.</i>	<i>Ο νέος μετατροπέας σχεδιάστηκε ώστε να χρησιμοποιεί τη ψύξη με αέρα και κατάλληλο ανεμιστήρα.</i>

<i>Έκθεση των ηλεκτρονικών στο περιβάλλον του εργαστηρίου και πλήθος χρησιμοποιούμενων καλωδίων</i>	<i>Ενοωμάτωση των ηλεκτρονικών εντός κουτιών προφύλαξης, κατασκευή ιδιών καλωδίων και οργάνωσής τους.</i>
---	---

Πίνακας 8.2) Προβλήματα και λύσεις ηλεκτρονικών διατάξεων.

Στην τρίτη κατηγορία εντάσσονται τα προβλήματα που παρουσιάστηκαν κατά τη διεξαγωγή των πειραμάτων. Βασικότερο πρόβλημα όλων είναι η παραγωγή κυματισμών οι οποίοι επηρέαζαν το αποτέλεσμα της μέτρησης της ώσης. Αν και ήταν αναμενόμενα ήδη κατά τα πρώτα στάδια εκπόνησης της πτυχιακής είχαν αναπτυχθεί ιδέες για τη μείωση τους, ωστόσο λόγω των απρόβλεπτων εξελίξεων σε σχέση με την ολοκλήρωση της πτυχιακής δεν κατέστη δυνατό να υλοποιηθούν. Τέτοιες λύσεις ήταν χαρακτηριστικά η εφαρμογή κατάλληλων πορωδών υλικών στην επιφάνεια πρόσκρουσης της ροής στη δεξαμενή καθώς και διατάξεων μείωσης του κυματισμού και εξομάλυνσης της ροής προς τον προωθητήρα. Επιπλέον στο σχεδιασμό αυτό προβλέπονταν και η σχεδίαση εκ νέου του υδατοστεγούς κουτιού στο οποίο τοποθετείται το σύστημα πρόωσης από σύνθετα υλικά ενώ το σχήμα του θα ήταν υδροδυναμικό τείνοντας σε μορφή γάστρας. Για τη λήψη καλύτερων μετρήσεων διεξήχθησαν αρκετές δοκιμές και βρέθηκε ότι η βέλτιστη μέτρηση λαμβάνεται όταν η δεξαμενή πληρωθεί με νερό ώστε να καλύπτει από το $\frac{1}{4}$ έως $\frac{3}{4}$ του στομίου εξαγωγής του ύδατος από τον προωθητήρα. Δεύτερο σημαντικό πρόβλημα που παρουσιάστηκε είναι η αναμενόμενη μείωση της τάσης του τροφοδοτικού. Το πρόβλημα αυτό μπορούσε να λυθεί με τη χρήση κατάλληλης συστοιχίας μπαταριών με συνολική τάση μεγαλύτερης των 16 Volt που όμως δεν ήταν στη διάθεση μου στο εργαστήριο. Μία δεύτερη λύση ήταν ο επανασχεδιασμός του τροφοδοτικού και της σχεδίασης συστήματος ρύθμισης της τάσης. Με μία πρόχειρη μελέτη των ηλεκτρονικών που θα απαιτούνταν υπολογίσθηκε το τελικό κόστος υπερέβαινε τις δυνατότητες που είχα και το χρόνο για την ανάπτυξη του. Άλλωστε δεν ήμουν υποχρεωμένος ούτε για τις έως μέχρι τώρα σχεδιασμένες ηλεκτρονικές διατάξεις του φίλτρου και του νέου παλμοτροφοδοτικού και τις τοποθετήσεις όλων σε κουτιά. Δεν εξετάζομαι στις γνώσεις επί των ηλεκτρονικών και του σχεδιασμού πλακετών, αντικείμενα άλλων σχολών, που θα έπρεπε να μου διατεθούν έτοιμα χωρίς να χάσω πολύτιμο χρόνο για την κατασκευή τους.

Προβλήματα Μετρήσεων	
Προβλήματα	Λύσεις
Κυματισμοί	Εύρεση της βέλτιστης στάθμης με σκοπό την μείωση των κυματισμών.
Πτώση τάση τροφοδοσίας	Χρήση συστοιχιών μπαταριών

Πίνακας 8.3) Προβλήματα μετρήσεων και λύσεις.

8.3) Στόχοι

Εξαρχής είχε τεθεί ως στόχος της πτυχιακής εργασίας αυτής, ο σχεδιασμός και η ανάπτυξη μιας διάταξης που θα επέτρεπε τη μέτρησης της ώσης για διάφορα συστήματα πρόωσης και κινητήρων. Η επίτευξη αυτού απαιτήσε τον καταμερισμό σε μικρότερους αυτοτελείς στόχους οι οποίοι δίνονται παρακάτω.

- ✚ Σχεδιασμός και υλοποίηση του συστήματος δεξαμενής, φορείου και εδράνων κύλισης.
- ✚ Προγραμματισμός ηλεκτρονικού μικροεπεξεργαστή.

- ✚ Ανάπτυξη προγράμματος σε ηλεκτρονικό υπολογιστή υπό το λειτουργικό των Windows
- ✚ Υλοποίηση ελεγκτή στροφών στο μικροεπεξεργαστή.
- ✚ Μέτρηση ροπής αδρανείας.
- ✚ Μέτρηση της ώσης.
- ✚ Καταγραφή των μετρούμενων μεγεθών.

Οι λέξεις κλειδιά που χαρακτηρίζουν την πτυχιακή είναι :

- ✓ Σχεδιασμός
- ✓ Προγραμματισμός
- ✓ Μικροεπεξεργαστής
- ✓ **Ροπή αδρανείας**
- ✓ Ελεγκτής στροφών
- ✓ Ρυθμιστής τάσης
- ✓ Φίλτρο σήματος
- ✓ Μετατροπέας τάσης
- ✓ Ωση
- ✓ Μέτρηση Στροφών

8.4) Επίτευξη των στόχων – Αποτελέσματα.

Κάθε ένας από τους παραπάνω επιμέρους στόχους έχει ιδιαίτερη αξία στην ολοκλήρωση της εργασίας. Σε κάθε ένα χρειάστηκε να αναπτυχθούν πρωτότυπες ιδέες για την αντιμετώπιση των προβλημάτων που παρουσιάστηκαν. Αξιοσημείωτο είναι ότι η αδυναμία στην εφαρμογή των μεθόδων μέτρησης της ροπής αδρανείας, που συναντήθηκαν στην βιβλιογραφία, οδήγησε στη διερεύνηση και ανάπτυξη μιας νέας μεθοδολογίας στην οποία δεν απαιτείται η ακριβής μέτρηση των μεγεθών. Απαιτεί μόνο τη γνώση των εφαρμοζόμενων πλατών της ροπής και των στροφών. Η μεταβολή της ροπής δε χρειάζεται να είναι ιδιαίτερα ακριβής αλλά αρκούν 10 διακριτές διαφορετικές τιμές ώστε να προσεγγισθεί ικανοποιητικά. Φυσικά όσο αυξάνει ο αριθμός αυτός τόσο μειώνεται το σφάλμα στην μέτρησης της ροπής αδρανείας.

- ✓ Μεριμνήθηκε ο σχεδιασμός της δεξαμενής και του φορείου έτσι ώστε να προσφέρει δυνατότητα προσαρμογής διάφορων τύπων κινητήρων και συστημάτων πρόωσης. Η χρήση **πολυεστερικών** υλικών και **ανοξείδωτου χάλυβα** δίνουν την απαραίτητη αντοχή για την εφαρμογή και μελέτη πολύ ισχυρότερων συστημάτων πρόωσης.
- ✓ Ο προγραμματισμός του μικροεπεξεργαστή πραγματοποιήθηκε με την χρήση των διακοπών ροής. Η λειτουργία του μικροεπεξεργαστή είναι πλέον **ευέλικτη** «robust» και η ροή του προγράμματος εξελίσσεται σύμφωνα με τα γεγονότα διακοπής αυξάνοντας την ευστάθεια του.
- ✓ Η **εφαρμογή επί του υπολογιστή** προσφέρει ένα εύκολο οπτικό περιβάλλον διαχείρισης της εγκατάστασης μέτρησης. Εκτός από τα εργαλεία χειρισμού της, προσφέρει τη δυνατότητα **αποθήκευση των μετρήσεων** και την εκτέλεση πειραμάτων μέτρησης των οποίων η εξέλιξη έχει σχεδιαστεί και αποθηκευθεί σε κατάλληλο αρχείο. Επίσης η

ενσωμάτωση των γραφικών παραστάσεων βοηθάει στην ρύθμιση ελεγκτών και την παρατήρηση της πορείας του πειράματος.

- ✓ Κατασκευάστηκε ελεγκτής τριών όρων που βοήθησε αποφασιστικά στην επιτυχή μέτρηση της ώσης.
- ✓ Η αδυναμία μέτρηση της ροπής αδρανείας οδήγησε στην ανάπτυξη μιας εύκολης μεθοδολογίας. Η ροπή αδρανείας είναι απαραίτητη για την εξεύρεση των «βέλτιστων» τιμών των όρων του ελεγκτή.

8.5) Προτάσεις για περαιτέρω έρευνα.

Ο σχεδιασμός όλου του προγράμματος και εν γένει της εγκατάστασης έχει γίνει για τη χρήση της ως ολοκληρωμένου και αυτοματοποιημένου πειραματικού συστήματος μετρήσεων. Η σχεδιασμένη ευελιξία της εγκατάστασης και των προγραμμάτων επιτρέπει τη χρήση της σε διάφορους τομείς. Πριν όμως παρουσιαστούν οι προτάσεις για περαιτέρω ερευνά αξίζει να σημειωθεί ότι στην εφαρμογή στον υπολογιστή έχει προβλεφθεί η διεξαγωγή των μετρήσεων τάσης τροφοδοσίας και έντασης ρεύματος. Για την πλήρη αξιοποίηση της εγκατάστασης προτείνεται η χρήση μετατροπέα αναλογικού σήματος σε ψηφιακό «A/D» ώστε να μπορέσουν να μετρηθούν τα παραπάνω μεγέθη αφού η κάρτα δε διαθέτει τη δυνατότητα αυτή. Προτείνεται η χρήση του A/D ολοκληρωμένου AD7767 της «Analog Devices» το οποίο διαθέτει S.P.I «Serial Peripheral Interface» και 24Bit ανάλυση.

Η εγκατάσταση ως έχει μπορεί να αξιοποιηθεί στα πλαίσια μαθημάτων για τη διεξαγωγή μετρήσεων ώσης και την επίδειξη της επίδρασης της σπηλαίωσης στην ώση της έλικας. Επιπλέον μπορεί να χρησιμοποιηθεί για την εύρεση της ροπής αδρανείας κινητήρων και συζευγμένων συστημάτων με αυτούς, σύμφωνα με την εφαρμογή της θεωρίας στο έκτο κεφάλαιο. Δίνεται η δυνατότητα ανάπτυξης πολυπλοκότερων ελεγκτών εφόσον άμεσα κατά την εκτέλεση των πειραμάτων με τη χρήση των γραφικών μπορεί να διαπιστωθεί η συμπεριφοράς τους. Η επανάληψη των μετρήσεων μπορεί να χρησιμοποιηθεί για την βελτιστοποίηση των ελεγκτών, δηλαδή μπορεί να χρησιμοποιηθεί για τη ρύθμιση «tuning» σε πραγματικές συνθήκες λειτουργίας και όχι υπό εξομοιώσεις σε προγράμματα. Η πειραματική διάταξη σχεδιάστηκε με τέτοιο τρόπο ώστε να είναι δυνατή η εφαρμογή διάφορων κινητήρων και συστημάτων πρόωσης ώστε να πραγματοποιηθούν μετρήσεις είτε ως προς του κινητήρες είτε προς την εύρεση της ώσης. Επιπλέον αν προστεθεί και υλοποιηθεί η μέτρηση της τάσης και της έντασης τότε θα μπορούν να πραγματοποιηθούν μετρήσεις για την απόδοσης των συστημάτων αυτών.

Η ευελιξία που προσφέρει η σχεδίαση των προγραμμάτων επιτρέπει τη χρήση τους και σε άλλου είδους πειράματα. Ένα παράδειγμα είναι η σύνδεση του κινητήρα με γεννήτρια και τη διενέργεια ελέγχου των στροφών της γεννήτριας υπό μεταβαλλόμενο φορτίο. Επεκτείνοντας τα προηγούμενα με μέτρηση της τάση και της έντασης του ρεύματος μπορούν να γίνουν και μετρήσεις του βαθμού απόδοσης της γεννήτριας. Η χρήση του προγράμματος και της εργαστηριακής διάταξης μπορεί να επεκταθεί και στη μελέτη ηλεκτρικών κινητήρων εναλλασσομένου ρεύματος καθώς

και θερμικών κινητήρων. Ο επεξεργαστής διαθέτει την κατάλληλη ισχύ και βιβλιοθήκη με μαθηματικές συναρτήσεις για την εκτίμηση της θέσης του δρομέα κινητήρων εναλλασσομένου με σκοπό του ελέγχου των στροφών του.

Προτείνεται η κατασκευή ενός καλύτερου παλμοτροφοδοτικού μετατροπέα με υψηλότερη συχνότητα παλμού ισχύος που θα επιτρέψει τη διεξαγωγή ακριβέστερων μετρήσεων της ροπής αδράνειας. Με τη χρήση της θεωρίας στο έκτο κεφάλαιο καθιστάτε η μέτρηση της αδράνειας πολύ εύκολη χωρίς να απαιτούνται πολύ ακριβή όργανα μέτρησης του χρόνου, της ροπής και της επιτάχυνσης. Η μέτρηση της αδράνεια ήταν πάντα μια ενδιαφέρουσα πρόκληση και η χρήση της ήταν πάντα σημαντική στην επιστήμη των μηχανικών για τη διερεύνηση της συμπεριφοράς των συστημάτων.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Killian CT. Modern Control Technology: Components and Systems. Thomson Delmar, USA, 2000.

Lewis EV. Principles of Naval Architecture-VolumeII. The Society of Naval Architects and Marine Engineers, USA, 1988.

Ξηρός ΝΙ. Συμπληρωματικές Σημειώσεις Αυτομάτου Ελέγχου για Ναυπηγούς Μηχανολόγους. Εκδόσεις Ε.Μ.Π., Αθήνα, 2004.

Πολίτης ΓΚ. Πρόωση με Ανάδραση-Waterjet Propulsion. Σημειώσεις Μαθήματος : Ειδικά Θέματα Πρόωσης Πλοίου, Αθήνα, 2004.

BRIAN W.KERIGHAN : Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C
ΕΚΔΟΣΗ ΔΕΥΤΕΡΗ 2000

Mohan N, Undeland TM, Robbins WP. Power Electronics: Converters, Applications and Design. John Wiley & Sons, USA, 1989.

C++ The Complete Reference, Fourth Edition, Herbert Schildt

C Θεωρία και Πράξη, Αλέξανδρος Τομαράς

Η Αρχιτεκτονική των υπολογιστών, μια δομημένη προσέγγιση, Andrew S. Tanenbaum

Microsoft Visual Basic .NET Programmer's Cookbook, Matthew MacDonald

Links

1. Τεκμηρίωση για το Visual Studio 2005

<http://msdn2.microsoft.com/el-gr/default.aspx>

2. Τεκμηρίωση Visual Basic 8

<http://msdn2.microsoft.com/en-us/vbasic/default.aspx>

3. Τεκμηρίωση LPC2106 κάρτας.

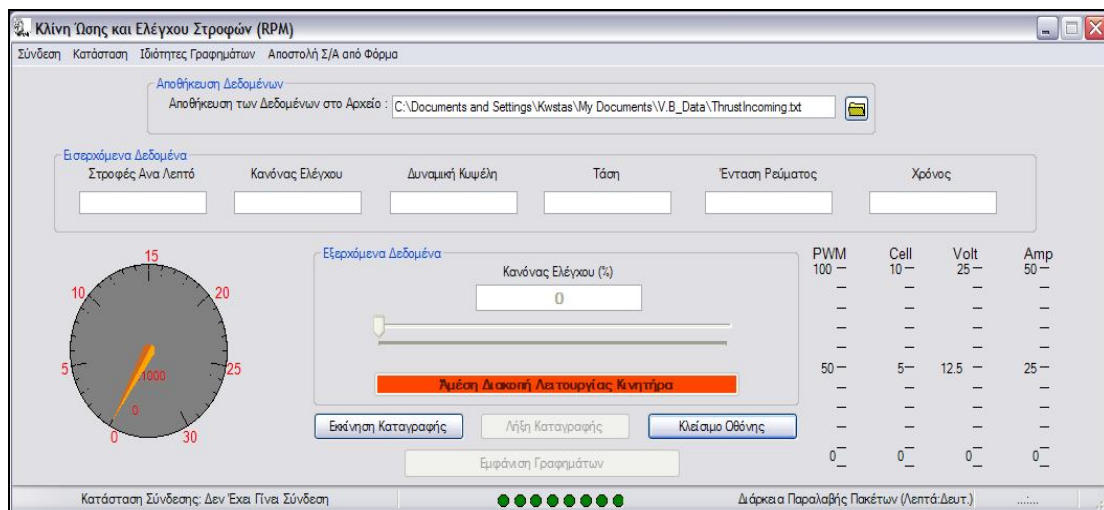
<http://www.iar.com/website1/1.0.1.0/3/1/index.php>

4. Τεκμηρίωση επεξεργαστή ARM7TDMI-S της lpC2106

http://www.nxp.com/pip/LPC2104_2105_2106_6.html

ΠΑΡΑΡΤΗΜΑ Α

ΚΩΔΙΚΑΣ ΠΡΟΓΡΑΜΜΑΤΟΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΣΕ ΓΛΩΣΣΑ VB




```

1 Public Class Evaluator
2
3     Private MyClassCode As New MyPtyxClass
4     Public WhichIsTheLang As Integer
5     Dim scriptControl As New MSScriptControl.ScriptControlClass
6     Private _Formula As String
7     Private _VariablesPreffix As String
8     Private _Operators As ArrayList
9     Private _CorrectedFormula As String
10    Private _FormulaVariables As ArrayList
11    Private _ParseCondition As ParseCondition = ParseCondition.
12    RaiseErroForUndefinedVariable
13
14    Public ReadOnly Property Formula() As String
15        Get
16            Return Me._Formula
17        End Get
18    End Property
19
20    Public ReadOnly Property VariablesPreffix() As String
21        Get
22            Return Me._VariablesPreffix
23        End Get
24    End Property
25
26    Private Sub PreEvaluate(ByVal keyCode As String, ByVal keyValue As String)
27        Me._Formula = Me._Formula.Replace(keyCode.ToUpper, keyValue)
28    End Sub
29
30    Private Sub PreEvaluate(ByVal keyValuesDictionary As Hashtable)
31        For Each keyCode As String In keyValuesDictionary.Keys
32            Me.PreEvaluate(keyCode, keyValuesDictionary(keyCode))
33        Next
34    End Sub
35
36    Public Function Evaluate(ByVal values As Hashtable) As Double
37        Dim tmpFormula As String = Me._Formula
38        For Each var As String In Me._FormulaVariables
39            Select Case var.ToLower
40                Case "sinh", "cosh", "abs", "round", "fix", "sin", "exp", "cos", "int",
41                    "atn", "tan", "sqr", "sgn", "log"
42                Case Else
43                    Dim index As Integer = CInt(var.Replace(Me._VariablesPreffix, ""))
44                    If Not values(index) Is Nothing Then
45                        tmpFormula = tmpFormula.Replace(var, values(index))
46                    Else
47                        If Me._ParseCondition = ParseCondition.
48                            RaiseErroForUndefinedVariable Then
49                            Throw New Exception(MsgBox(MyClassCode.GetStr
50                                (WhichIsTheLang, 246), 0, MyClassCode.GetStr(WhichIsTheLang, 109)))
51                        Else
52                            tmpFormula = tmpFormula.Replace(var, 0)
53                        End If
54                    End If
55                End Select
56            Next
57        Try
58            Return CDbl(scriptControl.Eval(tmpFormula))
59        Catch ex As Exception
60            scriptControl.Error.Clear()
61            Throw New Exception(MsgBox(MyClassCode.GetStr(WhichIsTheLang, 246), 0,
62                MyClassCode.GetStr(WhichIsTheLang, 109)))
63        Return -1
64    End Try
65    End Function
66
67    Public Sub New(ByVal WhichLang As Integer, ByVal formula As String, ByVal
68        variablePreffix As String, ByVal parseCondition As ParseCondition, Optional ByVal
69        preEvaluateKeyValuesDictionary As Hashtable = Nothing)
70        WhichIsTheLang = WhichLang
71        scriptControl.Language = "vbscript"
72        Me._Formula = formula
73        Me._VariablesPreffix = variablePreffix
74        Me._ParseCondition = parseCondition
75        InitializeOperators()

```

```

69     If Not preEvaluateKeyValuesDictionary Is Nothing Then
70         Me.PreEvaluate(preEvaluateKeyValuesDictionary)
71     End If
72     InitializeFormula()
73 End Sub
74
75 Private Sub InitializeOperators()
76     Me._Operators = New ArrayList
77     Me._Operators.Add(CChar("+"))
78     Me._Operators.Add(CChar("-"))
79     Me._Operators.Add(CChar("/"))
80     Me._Operators.Add(CChar("*"))
81     Me._Operators.Add(CChar("^"))
82     Me._Operators.Add(CChar("("))
83     Me._Operators.Add(CChar(")"))
84 End Sub
85
86 Private Sub InitializeFormula()
87     Me._CorrectedFormula = Me._Formula.Replace(" ", "")
88     Me._FormulaVariables = New ArrayList
89     Dim arrOperators() As Char
90     Dim variables() As String = Me._CorrectedFormula.Split(CType(Me._Operators.
91 ToArray(GetType(Char)), Char()))
92     For Each var As String In variables
93         var = var.Trim
94         If var.Length > 0 AndAlso Not IsNumeric(var) Then
95             Me._FormulaVariables.Add(var)
96             If Not IsNumeric(var.Replace(Me._VariablesPrefix, "")) Then
97                 Select Case var.ToLower
98                     Case "sinh", "cosh", "abs", "round", "fix", "sin", "exp", "cos"
99                     , "int", "atn", "tan", "sqr", "sgn", "log"
100                     Case Else
101                         Throw New Exception(MsgBox(MyClassCode.GetStr
102 (WhichIsTheLang, 246), 0, MyClassCode.GetStr(WhichIsTheLang, 109)))
103                     End Select
104                 End If
105             End If
106         Next
107     End Sub
108
109 Public ReadOnly Property Operators() As ArrayList
110     Get
111         Return Me._Operators
112     End Get
113 End Property
114
115 Public ReadOnly Property ParseCondition() As ParseCondition
116     Get
117         Return Me._ParseCondition
118     End Get
119 End Property
120 End Class
121
122 Public Enum ParseCondition
123     RaiseErroForUndefinedVariable
124     DoNotRaiseErroForUndefinedVariable
125 End Enum

```

```

1 Imports System.Text
2 Imports System.IO
3 Imports Microsoft.Win32
4 Imports System.Drawing.Printing
5 Imports system
6
7
8 Public Class Form1
9     Inherits System.Windows.Forms.Form
10
11 #Region " Windows Form Designer generated code "
12
13     Public Sub New()
14         MyBase.New()
15
16         'This call is required by the Windows Form Designer.
17         InitializeComponent()
18
19         'Add any initialization after the InitializeComponent() call
20
21     End Sub
22
23     'Form overrides dispose to clean up the component list.
24     Protected Overrides Sub Dispose(ByVal disposing As Boolean)
25         If disposing Then
26             If Not (components Is Nothing) Then
27                 components.Dispose()
28             End If
29         End If
30         MyBase.Dispose(disposing)
31     End Sub
32
33     'Required by the Windows Form Designer
34     Private components As System.ComponentModel.IContainer
35
36     'NOTE: The following procedure is required by the Windows Form Designer
37     'It can be modified using the Windows Form Designer.
38     'Do not modify it using the code editor.
39     Friend WithEvents MainMenu1 As System.Windows.Forms.MainMenu
40     Friend WithEvents MenuItem1 As System.Windows.Forms.MenuItem
41     Friend WithEvents MenuItem2 As System.Windows.Forms.MenuItem
42     Friend WithEvents MenuItem3 As System.Windows.Forms.MenuItem
43     Friend WithEvents MenuItem4 As System.Windows.Forms.MenuItem
44     Friend WithEvents MenuItem5 As System.Windows.Forms.MenuItem
45     Friend WithEvents ToolBar1 As System.Windows.Forms.ToolBar
46     Friend WithEvents imlTB As System.Windows.Forms.ImageList
47     Friend WithEvents ToolBarButton1 As System.Windows.Forms.ToolBarButton
48     Friend WithEvents ToolBarButton2 As System.Windows.Forms.ToolBarButton
49     Friend WithEvents ToolBarButton3 As System.Windows.Forms.ToolBarButton
50     Friend WithEvents ToolBarButton4 As System.Windows.Forms.ToolBarButton
51     Friend WithEvents sbr As System.Windows.Forms.StatusBar
52     Friend WithEvents sbpComNo As System.Windows.Forms.StatusBarPanel
53     Friend WithEvents sbpSettings As System.Windows.Forms.StatusBarPanel
54     Friend WithEvents sbpStatus As System.Windows.Forms.StatusBarPanel
55     Friend WithEvents tmrRead As System.Windows.Forms.Timer
56     Friend WithEvents TextBox1 As System.Windows.Forms.TextBox
57     Friend WithEvents Button1 As System.Windows.Forms.Button
58     Friend WithEvents LblSaveRecvData As System.Windows.Forms.Label
59     Friend WithEvents TxtSaveRecvData As System.Windows.Forms.TextBox
60     Friend WithEvents BtSaveRecvData As System.Windows.Forms.Button
61     Friend WithEvents BtSaveSentData As System.Windows.Forms.Button
62     Friend WithEvents TxtSaveSentData As System.Windows.Forms.TextBox
63     Friend WithEvents LblSaveSentData As System.Windows.Forms.Label
64     Friend WithEvents GrBxDataSave As System.Windows.Forms.GroupBox
65     Friend WithEvents GrBxDataSend As System.Windows.Forms.GroupBox
66     Friend WithEvents ChkSendChars As System.Windows.Forms.CheckBox
67     Friend WithEvents ChkSendFile As System.Windows.Forms.CheckBox
68     Friend WithEvents TxtDefineFileName As System.Windows.Forms.TextBox
69     Friend WithEvents BtDefineFileName As System.Windows.Forms.Button
70     Friend WithEvents MenuItem7 As System.Windows.Forms.MenuItem
71     Friend WithEvents MenuItem8 As System.Windows.Forms.MenuItem
72     Friend WithEvents MenuItem9 As System.Windows.Forms.MenuItem
73     Friend WithEvents TextBox3 As System.Windows.Forms.TextBox
74     Friend WithEvents LblInputData As System.Windows.Forms.Label
75     Friend WithEvents LblOutputData As System.Windows.Forms.Label

```

```

76 Friend WithEvents MenuItem6 As System.Windows.Forms.MenuItem
77 Friend WithEvents MnConfAllTabs As System.Windows.Forms.MenuItem
78 Friend WithEvents MnConfLanguage As System.Windows.Forms.MenuItem
79 Friend WithEvents MnConfGraphs As System.Windows.Forms.MenuItem
80 Friend WithEvents BtClearInputData As System.Windows.Forms.Button
81 Friend WithEvents BtClearOutputData As System.Windows.Forms.Button
82 Friend WithEvents GrBxPreviewData As System.Windows.Forms.GroupBox
83 Friend WithEvents MnConfSerial As System.Windows.Forms.MenuItem
84 Friend WithEvents MenuItem17 As System.Windows.Forms.MenuItem
85 Friend WithEvents MenuItem18 As System.Windows.Forms.MenuItem
86 Friend WithEvents MenuItem19 As System.Windows.Forms.MenuItem
87 Friend WithEvents MenuItem20 As System.Windows.Forms.MenuItem
88 Friend WithEvents MenuItem21 As System.Windows.Forms.MenuItem
89 Friend WithEvents MenuItem22 As System.Windows.Forms.MenuItem
90 Friend WithEvents MenuItem23 As System.Windows.Forms.MenuItem
91 Friend WithEvents MenuItem24 As System.Windows.Forms.MenuItem
92 Friend WithEvents MenuItem25 As System.Windows.Forms.MenuItem
93 Friend WithEvents MenuItem26 As System.Windows.Forms.MenuItem
94 Friend WithEvents MenuItem27 As System.Windows.Forms.MenuItem
95 Friend WithEvents MITefarmoges As System.Windows.Forms.MenuItem
96 Friend WithEvents TextBox2 As System.Windows.Forms.TextBox
97 Friend WithEvents MySerialPort As System.IO.Ports.SerialPort
98 Friend WithEvents MenuItem29 As System.Windows.Forms.MenuItem
99 Friend WithEvents MenuItem31 As System.Windows.Forms.MenuItem
100 Friend WithEvents MenuItem32 As System.Windows.Forms.MenuItem
101 Friend WithEvents MenuItem33 As System.Windows.Forms.MenuItem
102 Friend WithEvents MnItExit As System.Windows.Forms.MenuItem
103 Friend WithEvents MnConfThrust As System.Windows.Forms.MenuItem
104 Friend WithEvents MenuItem34 As System.Windows.Forms.MenuItem
105 Friend WithEvents MenuItem35 As System.Windows.Forms.MenuItem
106 Friend WithEvents MenuItem36 As System.Windows.Forms.MenuItem
107 Friend WithEvents MenuItem37 As System.Windows.Forms.MenuItem
108 Friend WithEvents PBInitialise As System.Windows.Forms.PictureBox
109 Friend WithEvents PnForVideo As System.Windows.Forms.Panel
110 Friend WithEvents BtVideoStop As System.Windows.Forms.Button
111 Friend WithEvents BtVideoPause As System.Windows.Forms.Button
112 Friend WithEvents BtVideoStart As System.Windows.Forms.Button
113 Friend WithEvents HelpProviderfrmMain As System.Windows.Forms.HelpProvider
114 Friend WithEvents AxWMPVideoPlayer As AxWMPLib.AxWindowsMediaPlayer
115 <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
116     Me.components = New System.ComponentModel.Container
117     Dim resources As System.ComponentModel.ComponentResourceManager = New System.
ComponentModel.ComponentResourceManager(GetType(Form1))
118     Me.MainMenu1 = New System.Windows.Forms.MainMenu(Me.components)
119     Me.MenuItem1 = New System.Windows.Forms.MenuItem
120     Me.MenuItem2 = New System.Windows.Forms.MenuItem
121     Me.MenuItem3 = New System.Windows.Forms.MenuItem
122     Me.MenuItem4 = New System.Windows.Forms.MenuItem
123     Me.MenuItem5 = New System.Windows.Forms.MenuItem
124     Me.MITefarmoges = New System.Windows.Forms.MenuItem
125     Me.MenuItem32 = New System.Windows.Forms.MenuItem
126     Me.MenuItem29 = New System.Windows.Forms.MenuItem
127     Me.MenuItem34 = New System.Windows.Forms.MenuItem
128     Me.MenuItem31 = New System.Windows.Forms.MenuItem
129     Me.MenuItem33 = New System.Windows.Forms.MenuItem
130     Me.MnItExit = New System.Windows.Forms.MenuItem
131     Me.MenuItem6 = New System.Windows.Forms.MenuItem
132     Me.MnConfAllTabs = New System.Windows.Forms.MenuItem
133     Me.MenuItem17 = New System.Windows.Forms.MenuItem
134     Me.MnConfLanguage = New System.Windows.Forms.MenuItem
135     Me.MnConfSerial = New System.Windows.Forms.MenuItem
136     Me.MnConfThrust = New System.Windows.Forms.MenuItem
137     Me.MnConfGraphs = New System.Windows.Forms.MenuItem
138     Me.MenuItem18 = New System.Windows.Forms.MenuItem
139     Me.MenuItem19 = New System.Windows.Forms.MenuItem
140     Me.MenuItem20 = New System.Windows.Forms.MenuItem
141     Me.MenuItem7 = New System.Windows.Forms.MenuItem
142     Me.MenuItem8 = New System.Windows.Forms.MenuItem
143     Me.MenuItem9 = New System.Windows.Forms.MenuItem
144     Me.MenuItem35 = New System.Windows.Forms.MenuItem
145     Me.MenuItem36 = New System.Windows.Forms.MenuItem
146     Me.MenuItem37 = New System.Windows.Forms.MenuItem
147     Me.MenuItem21 = New System.Windows.Forms.MenuItem
148     Me.MenuItem22 = New System.Windows.Forms.MenuItem
149     Me.MenuItem23 = New System.Windows.Forms.MenuItem

```

```

150     Me.MenuItem24 = New System.Windows.Forms.MenuItem
151     Me.MenuItem25 = New System.Windows.Forms.MenuItem
152     Me.MenuItem26 = New System.Windows.Forms.MenuItem
153     Me.MenuItem27 = New System.Windows.Forms.MenuItem
154     Me.ToolBar1 = New System.Windows.Forms.ToolBar
155     Me.ToolBarButton1 = New System.Windows.Forms.ToolBarButton
156     Me.ToolBarButton2 = New System.Windows.Forms.ToolBarButton
157     Me.ToolBarButton3 = New System.Windows.Forms.ToolBarButton
158     Me.ToolBarButton4 = New System.Windows.Forms.ToolBarButton
159     Me.imlTB = New System.Windows.Forms.ImageList(Me.components)
160     Me.sbr = New System.Windows.Forms.StatusBar
161     Me.sbpComNo = New System.Windows.Forms.StatusBarPanel
162     Me.sbpSettings = New System.Windows.Forms.StatusBarPanel
163     Me.sbpStatus = New System.Windows.Forms.StatusBarPanel
164     Me.tmrRead = New System.Windows.Forms.Timer(Me.components)
165     Me.TextBox1 = New System.Windows.Forms.TextBox
166     Me.LblSaveRecvData = New System.Windows.Forms.Label
167     Me.TxtSaveRecvData = New System.Windows.Forms.TextBox
168     Me.TxtSaveSentData = New System.Windows.Forms.TextBox
169     Me.LblSaveSentData = New System.Windows.Forms.Label
170     Me.GrBxDataSave = New System.Windows.Forms.GroupBox
171     Me.GrBxDataSend = New System.Windows.Forms.GroupBox
172     Me.TxtDefineFileName = New System.Windows.Forms.TextBox
173     Me.ChkSendChars = New System.Windows.Forms.CheckBox
174     Me.ChkSendFile = New System.Windows.Forms.CheckBox
175     Me.BtDefineFileName = New System.Windows.Forms.Button
176     Me.TextBox3 = New System.Windows.Forms.TextBox
177     Me.LblInputData = New System.Windows.Forms.Label
178     Me.LblOutputData = New System.Windows.Forms.Label
179     Me.GrBxPreviewData = New System.Windows.Forms.GroupBox
180     Me.BtClearOutputData = New System.Windows.Forms.Button
181     Me.BtClearInputData = New System.Windows.Forms.Button
182     Me.TextBox2 = New System.Windows.Forms.TextBox
183     Me.MySerialPort = New System.IO.Ports.SerialPort(Me.components)
184     Me.BtSaveSentData = New System.Windows.Forms.Button
185     Me.BtSaveRecvData = New System.Windows.Forms.Button
186     Me.Button1 = New System.Windows.Forms.Button
187     Me.PBInitialise = New System.Windows.Forms.PictureBox
188     Me.PnForVideo = New System.Windows.Forms.Panel
189     Me.AxWMPVideoPlayer = New AxWMPLib.AxWindowsMediaPlayer
190     Me.BtVideoStop = New System.Windows.Forms.Button
191     Me.BtVideoPause = New System.Windows.Forms.Button
192     Me.BtVideoStart = New System.Windows.Forms.Button
193     Me.HelpProviderfrmMain = New System.Windows.Forms.HelpProvider
194     CType(Me.sbpComNo, System.ComponentModel.ISupportInitialize).BeginInit()
195     CType(Me.sbpSettings, System.ComponentModel.ISupportInitialize).BeginInit()
196     CType(Me.sbpStatus, System.ComponentModel.ISupportInitialize).BeginInit()
197     Me.GrBxDataSend.SuspendLayout()
198     Me.GrBxPreviewData.SuspendLayout()
199     CType(Me.PBInitialise, System.ComponentModel.ISupportInitialize).BeginInit()
200     Me.PnForVideo.SuspendLayout()
201     CType(Me.AxWMPVideoPlayer, System.ComponentModel.ISupportInitialize).BeginInit()
202     Me.SuspendLayout()
203     '
204     'MainMenu1
205     '
206     Me.MainMenu1.MenuItems.AddRange(New System.Windows.Forms.MenuItem() {Me.
207     MenuItem1, Me.MItEfarmoges, Me.MenuItem6, Me.MenuItem18, Me.MenuItem7, Me.
208     MenuItem21, Me.MenuItem25})
209     '
210     'MenuItem1
211     '
212     Me.MenuItem1.Index = 0
213     Me.MainMenu1.MenuItems.AddRange(New System.Windows.Forms.MenuItem() {Me.
214     MenuItem2, Me.MenuItem3, Me.MenuItem4, Me.MenuItem5})
215     resources.ApplyResources(Me.MenuItem1, "MenuItem1")
216     '
217     'MenuItem2
218     '
219     Me.MenuItem2.Index = 0
220     resources.ApplyResources(Me.MenuItem2, "MenuItem2")
221     '
222     'MenuItem3
223     '

```

```

221     resources.ApplyResources(Me.MenuItem3, "MenuItem3")
222     Me.MenuItem3.Index = 1
223     '
224     'MenuItem4
225     '
226     Me.MenuItem4.Index = 2
227     resources.ApplyResources(Me.MenuItem4, "MenuItem4")
228     '
229     'MenuItem5
230     '
231     Me.MenuItem5.Index = 3
232     resources.ApplyResources(Me.MenuItem5, "MenuItem5")
233     '
234     'MItEfarmoges
235     '
236     Me.MItEfarmoges.Index = 1
237     Me.MItEfarmoges.MenuItems.AddRange(New System.Windows.Forms.MenuItem() {Me.
MenuItems32, Me.MenuItem29, Me.MenuItem34, Me.MenuItem31, Me.MenuItem33, Me.
MnItExit})
238     resources.ApplyResources(Me.MItEfarmoges, "MItEfarmoges")
239     '
240     'MenuItem32
241     '
242     Me.MenuItem32.Index = 0
243     resources.ApplyResources(Me.MenuItem32, "MenuItem32")
244     '
245     'MenuItem29
246     '
247     Me.MenuItem29.Index = 1
248     resources.ApplyResources(Me.MenuItem29, "MenuItem29")
249     '
250     'MenuItem34
251     '
252     Me.MenuItem34.Index = 2
253     resources.ApplyResources(Me.MenuItem34, "MenuItem34")
254     '
255     'MenuItem31
256     '
257     Me.MenuItem31.Index = 3
258     resources.ApplyResources(Me.MenuItem31, "MenuItem31")
259     '
260     'MenuItem33
261     '
262     Me.MenuItem33.Index = 4
263     resources.ApplyResources(Me.MenuItem33, "MenuItem33")
264     '
265     'MnItExit
266     '
267     Me.MnItExit.Index = 5
268     resources.ApplyResources(Me.MnItExit, "MnItExit")
269     '
270     'MenuItem6
271     '
272     Me.MenuItem6.Index = 2
273     Me.MenuItem6.MenuItems.AddRange(New System.Windows.Forms.MenuItem() {Me.
MnConfAllTabs, Me.MenuItem17, Me.MnConfLanguage, Me.MnConfSerial, Me.MnConfThrust,
Me.MnConfGraphs})
274     resources.ApplyResources(Me.MenuItem6, "MenuItem6")
275     '
276     'MnConfAllTabs
277     '
278     Me.MnConfAllTabs.Index = 0
279     resources.ApplyResources(Me.MnConfAllTabs, "MnConfAllTabs")
280     '
281     'MenuItem17
282     '
283     Me.MenuItem17.Index = 1
284     resources.ApplyResources(Me.MenuItem17, "MenuItem17")
285     '
286     'MnConfLanguage
287     '
288     Me.MnConfLanguage.Index = 2
289     resources.ApplyResources(Me.MnConfLanguage, "MnConfLanguage")
290     '
291     'MnConfSerial

```

```

292         '
293         Me.MnConfSerial.Index = 3
294         resources.ApplyResources(Me.MnConfSerial, "MnConfSerial")
295         '
296         'MnConfThrust
297         '
298         Me.MnConfThrust.Index = 4
299         resources.ApplyResources(Me.MnConfThrust, "MnConfThrust")
300         '
301         'MnConfGraphs
302         '
303         Me.MnConfGraphs.Index = 5
304         resources.ApplyResources(Me.MnConfGraphs, "MnConfGraphs")
305         '
306         'MenuItem18
307         '
308         Me.MenuItem18.Index = 3
309         Me.MenuItem18.MenuItems.AddRange(New System.Windows.Forms.MenuItem() {Me.
MenuItem19, Me.MenuItem20})
310         resources.ApplyResources(Me.MenuItem18, "MenuItem18")
311         '
312         'MenuItem19
313         '
314         Me.MenuItem19.Index = 0
315         resources.ApplyResources(Me.MenuItem19, "MenuItem19")
316         '
317         'MenuItem20
318         '
319         Me.MenuItem20.Index = 1
320         resources.ApplyResources(Me.MenuItem20, "MenuItem20")
321         '
322         'MenuItem7
323         '
324         Me.MenuItem7.Index = 4
325         Me.MenuItem7.MenuItems.AddRange(New System.Windows.Forms.MenuItem() {Me.
MenuItem8, Me.MenuItem9, Me.MenuItem35, Me.MenuItem36, Me.MenuItem37})
326         resources.ApplyResources(Me.MenuItem7, "MenuItem7")
327         '
328         'MenuItem8
329         '
330         Me.MenuItem8.Index = 0
331         resources.ApplyResources(Me.MenuItem8, "MenuItem8")
332         '
333         'MenuItem9
334         '
335         Me.MenuItem9.Index = 1
336         resources.ApplyResources(Me.MenuItem9, "MenuItem9")
337         '
338         'MenuItem35
339         '
340         Me.MenuItem35.Index = 2
341         resources.ApplyResources(Me.MenuItem35, "MenuItem35")
342         '
343         'MenuItem36
344         '
345         Me.MenuItem36.Index = 3
346         resources.ApplyResources(Me.MenuItem36, "MenuItem36")
347         '
348         'MenuItem37
349         '
350         Me.MenuItem37.Index = 4
351         resources.ApplyResources(Me.MenuItem37, "MenuItem37")
352         '
353         'MenuItem21
354         '
355         Me.MenuItem21.Index = 5
356         Me.MenuItem21.MenuItems.AddRange(New System.Windows.Forms.MenuItem() {Me.
MenuItem22, Me.MenuItem23, Me.MenuItem24})
357         resources.ApplyResources(Me.MenuItem21, "MenuItem21")
358         '
359         'MenuItem22
360         '
361         Me.MenuItem22.Index = 0
362         resources.ApplyResources(Me.MenuItem22, "MenuItem22")
363         '

```

```

364         'MenuItem23
365         '
366         Me.MenuItem23.Index = 1
367         resources.ApplyResources(Me.MenuItem23, "MenuItem23")
368         '
369         'MenuItem24
370         '
371         Me.MenuItem24.Index = 2
372         resources.ApplyResources(Me.MenuItem24, "MenuItem24")
373         '
374         'MenuItem25
375         '
376         Me.MenuItem25.Index = 6
377         Me.MenuItem25.MenuItems.AddRange(New System.Windows.Forms.MenuItem() {Me.
MenuItem26, Me.MenuItem27})
378         resources.ApplyResources(Me.MenuItem25, "MenuItem25")
379         '
380         'MenuItem26
381         '
382         Me.MenuItem26.Index = 0
383         resources.ApplyResources(Me.MenuItem26, "MenuItem26")
384         '
385         'MenuItem27
386         '
387         Me.MenuItem27.Index = 1
388         resources.ApplyResources(Me.MenuItem27, "MenuItem27")
389         '
390         'ToolBar1
391         '
392         Me.ToolBar1.Buttons.AddRange(New System.Windows.Forms.ToolBarButton() {Me.
ToolBarButton1, Me.ToolBarButton2, Me.ToolBarButton3, Me.ToolBarButton4})
393         resources.ApplyResources(Me.ToolBar1, "ToolBar1")
394         Me.ToolBar1.ImageList = Me.imlTB
395         Me.ToolBar1.Name = "ToolBar1"
396         '
397         'ToolBarButton1
398         '
399         resources.ApplyResources(Me.ToolBarButton1, "ToolBarButton1")
400         '
401         'ToolBarButton2
402         '
403         resources.ApplyResources(Me.ToolBarButton2, "ToolBarButton2")
404         '
405         'ToolBarButton3
406         '
407         resources.ApplyResources(Me.ToolBarButton3, "ToolBarButton3")
408         '
409         'ToolBarButton4
410         '
411         resources.ApplyResources(Me.ToolBarButton4, "ToolBarButton4")
412         '
413         'imlTB
414         '
415         Me.imlTB.ImageStream = CType(resources.GetObject("imlTB.ImageStream"), System.
Windows.Forms.ImageListStreamer)
416         Me.imlTB.TransparentColor = System.Drawing.Color.Transparent
417         Me.imlTB.Images.SetKeyName(0, "")
418         Me.imlTB.Images.SetKeyName(1, "")
419         Me.imlTB.Images.SetKeyName(2, "")
420         '
421         'sbr
422         '
423         resources.ApplyResources(Me.sbr, "sbr")
424         Me.sbr.Name = "sbr"
425         Me.sbr.Panels.AddRange(New System.Windows.Forms.StatusBarPanel() {Me.sbpComNo,
Me.sbpSettings, Me.sbpStatus})
426         Me.sbr.ShowPanels = True
427         '
428         'sbpComNo
429         '
430         resources.ApplyResources(Me.sbpComNo, "sbpComNo")
431         '
432         'sbpSettings
433         '
434         resources.ApplyResources(Me.sbpSettings, "sbpSettings")

```



```

435         '
436         'sbpStatus
437         '
438         resources.ApplyResources(Me.sbpStatus, "sbpStatus")
439         '
440         'TextBox1
441         '
442         Me.TextBox1.AcceptsTab = True
443         resources.ApplyResources(Me.TextBox1, "TextBox1")
444         Me.TextBox1.Name = "TextBox1"
445         '
446         'LblSaveRecvData
447         '
448         resources.ApplyResources(Me.LblSaveRecvData, "LblSaveRecvData")
449         Me.LblSaveRecvData.Name = "LblSaveRecvData"
450         '
451         'TxtSaveRecvData
452         '
453         resources.ApplyResources(Me.TxtSaveRecvData, "TxtSaveRecvData")
454         Me.TxtSaveRecvData.Name = "TxtSaveRecvData"
455         '
456         'TxtSaveSentData
457         '
458         resources.ApplyResources(Me.TxtSaveSentData, "TxtSaveSentData")
459         Me.TxtSaveSentData.Name = "TxtSaveSentData"
460         '
461         'LblSaveSentData
462         '
463         resources.ApplyResources(Me.LblSaveSentData, "LblSaveSentData")
464         Me.LblSaveSentData.Name = "LblSaveSentData"
465         '
466         'GrBxDataSave
467         '
468         resources.ApplyResources(Me.GrBxDataSave, "GrBxDataSave")
469         Me.GrBxDataSave.Name = "GrBxDataSave"
470         Me.GrBxDataSave.TabStop = False
471         '
472         'GrBxDataSend
473         '
474         Me.GrBxDataSend.Controls.Add(Me.TxtDefineFileName)
475         Me.GrBxDataSend.Controls.Add(Me.ChkSendChars)
476         Me.GrBxDataSend.Controls.Add(Me.ChkSendFile)
477         Me.GrBxDataSend.Controls.Add(Me.BtDefineFileName)
478         resources.ApplyResources(Me.GrBxDataSend, "GrBxDataSend")
479         Me.GrBxDataSend.Name = "GrBxDataSend"
480         Me.GrBxDataSend.TabStop = False
481         '
482         'TxtDefineFileName
483         '
484         Me.TxtDefineFileName.AcceptsTab = True
485         resources.ApplyResources(Me.TxtDefineFileName, "TxtDefineFileName")
486         Me.TxtDefineFileName.Name = "TxtDefineFileName"
487         '
488         'ChkSendChars
489         '
490         resources.ApplyResources(Me.ChkSendChars, "ChkSendChars")
491         Me.ChkSendChars.Name = "ChkSendChars"
492         '
493         'ChkSendFile
494         '
495         resources.ApplyResources(Me.ChkSendFile, "ChkSendFile")
496         Me.ChkSendFile.Name = "ChkSendFile"
497         '
498         'BtDefineFileName
499         '
500         resources.ApplyResources(Me.BtDefineFileName, "BtDefineFileName")
501         Me.BtDefineFileName.Name = "BtDefineFileName"
502         '
503         'TextBox3
504         '
505         resources.ApplyResources(Me.TextBox3, "TextBox3")
506         Me.TextBox3.Name = "TextBox3"
507         Me.TextBox3.ReadOnly = True
508         '
509         'LblInputData

```

```

510     '
511     resources.ApplyResources(Me.LblInputData, "LblInputData")
512     Me.LblInputData.Name = "LblInputData"
513     '
514     'LblOutputData
515     '
516     resources.ApplyResources(Me.LblOutputData, "LblOutputData")
517     Me.LblOutputData.Name = "LblOutputData"
518     '
519     'GrBxPreviewData
520     '
521     Me.GrBxPreviewData.Controls.Add(Me.BtClearOutputData)
522     Me.GrBxPreviewData.Controls.Add(Me.BtClearInputData)
523     Me.GrBxPreviewData.Controls.Add(Me.TextBox2)
524     resources.ApplyResources(Me.GrBxPreviewData, "GrBxPreviewData")
525     Me.GrBxPreviewData.Name = "GrBxPreviewData"
526     Me.GrBxPreviewData.TabStop = False
527     '
528     'BtClearOutputData
529     '
530     resources.ApplyResources(Me.BtClearOutputData, "BtClearOutputData")
531     Me.BtClearOutputData.Name = "BtClearOutputData"
532     '
533     'BtClearInputData
534     '
535     resources.ApplyResources(Me.BtClearInputData, "BtClearInputData")
536     Me.BtClearInputData.Name = "BtClearInputData"
537     '
538     'TextBox2
539     '
540     resources.ApplyResources(Me.TextBox2, "TextBox2")
541     Me.TextBox2.Name = "TextBox2"
542     Me.TextBox2.ReadOnly = True
543     '
544     'MySerialPort
545     '
546     '
547     'BtSaveSentData
548     '
549     resources.ApplyResources(Me.BtSaveSentData, "BtSaveSentData")
550     Me.BtSaveSentData.Name = "BtSaveSentData"
551     '
552     'BtSaveRecvData
553     '
554     resources.ApplyResources(Me.BtSaveRecvData, "BtSaveRecvData")
555     Me.BtSaveRecvData.Name = "BtSaveRecvData"
556     '
557     'Button1
558     '
559     resources.ApplyResources(Me.Button1, "Button1")
560     Me.Button1.Name = "Button1"
561     '
562     'PBInitialise
563     '
564     resources.ApplyResources(Me.PBInitialise, "PBInitialise")
565     Me.PBInitialise.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D
566     Me.PBInitialise.Name = "PBInitialise"
567     Me.PBInitialise.TabStop = False
568     '
569     'PnForVideo
570     '
571     Me.PnForVideo.Controls.Add(Me.AxWMPVideoPlayer)
572     Me.PnForVideo.Controls.Add(Me.BtVideoStop)
573     Me.PnForVideo.Controls.Add(Me.BtVideoPause)
574     Me.PnForVideo.Controls.Add(Me.BtVideoStart)
575     resources.ApplyResources(Me.PnForVideo, "PnForVideo")
576     Me.PnForVideo.Name = "PnForVideo"
577     '
578     'AxWMPVideoPlayer
579     '
580     resources.ApplyResources(Me.AxWMPVideoPlayer, "AxWMPVideoPlayer")
581     Me.AxWMPVideoPlayer.Name = "AxWMPVideoPlayer"
582     Me.AxWMPVideoPlayer.OcxState = CType(resources.GetObject("AxWMPVideoPlayer.
OcxState"), System.Windows.Forms.AxHost.State)
583     '

```

```

584         'BtVideoStop
585         '
586         resources.ApplyResources(Me.BtVideoStop, "BtVideoStop")
587         Me.BtVideoStop.Name = "BtVideoStop"
588         Me.BtVideoStop.UseVisualStyleBackColor = True
589         '
590         'BtVideoPause
591         '
592         resources.ApplyResources(Me.BtVideoPause, "BtVideoPause")
593         Me.BtVideoPause.Name = "BtVideoPause"
594         Me.BtVideoPause.UseVisualStyleBackColor = True
595         '
596         'BtVideoStart
597         '
598         resources.ApplyResources(Me.BtVideoStart, "BtVideoStart")
599         Me.BtVideoStart.Name = "BtVideoStart"
600         Me.BtVideoStart.UseVisualStyleBackColor = True
601         '
602         'Form1
603         '
604         resources.ApplyResources(Me, "$this")
605         Me.Controls.Add(Me.PnForVideo)
606         Me.Controls.Add(Me.TextBox1)
607         Me.Controls.Add(Me.LblOutputData)
608         Me.Controls.Add(Me.LblInputData)
609         Me.Controls.Add(Me.TextBox3)
610         Me.Controls.Add(Me.TxtSaveSentData)
611         Me.Controls.Add(Me.TxtSaveRecvData)
612         Me.Controls.Add(Me.BtSaveSentData)
613         Me.Controls.Add(Me.LblSaveSentData)
614         Me.Controls.Add(Me.BtSaveRecvData)
615         Me.Controls.Add(Me.LblSaveRecvData)
616         Me.Controls.Add(Me.Button1)
617         Me.Controls.Add(Me.sbr)
618         Me.Controls.Add(Me.ToolBar1)
619         Me.Controls.Add(Me.GrBxPreviewData)
620         Me.Controls.Add(Me.GrBxDataSend)
621         Me.Controls.Add(Me.GrBxDataSave)
622         Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle
623         Me.MaximizeBox = False
624         Me.Menu = Me.MainMenu1
625         Me.Name = "Form1"
626         CType(Me.sbpComNo, System.ComponentModel.ISupportInitialize).EndInit()
627         CType(Me.sbpSettings, System.ComponentModel.ISupportInitialize).EndInit()
628         CType(Me.sbpStatus, System.ComponentModel.ISupportInitialize).EndInit()
629         Me.GrBxDataSend.ResumeLayout(False)
630         Me.GrBxDataSend.PerformLayout()
631         Me.GrBxPreviewData.ResumeLayout(False)
632         Me.GrBxPreviewData.PerformLayout()
633         CType(Me.PBInitialise, System.ComponentModel.ISupportInitialize).EndInit()
634         Me.PnForVideo.ResumeLayout(False)
635         CType(Me.AxWMPVideoPlayer, System.ComponentModel.ISupportInitialize).EndInit()
636         Me.ResumeLayout(False)
637         Me.PerformLayout()
638
639     End Sub
640
641 #End Region
642     Private MyClassCode As New MyPtyxClass
643     Public MySaveRecv, MySaveSent, MyFileNameSend As String
644     Public SrReceive As IO.StreamWriter
645     Public WhichLang As Integer
646     Public SelectedConfFile As String
647     Public DocumentToPrint As String
648     Public ProblemOccured, WasAlreadyPaused As Boolean
649     Delegate Sub AppendToTextCallback(ByVal UsevbCrLf As Boolean, ByVal WhichTextBox As Integer, ByVal text As String)
650     Public WhichIsTheTextBox As Integer
651     Public WhichIsTheMessage As String
652     Delegate Sub InvokeDelegate()
653     Public ExcelIsInstalled As Boolean
654
655     Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
656         Dim i As Integer

```

```

657         For i = 0 To 9
658             mComs(i) = IsPortAvailable(i + 1)
659         Next
660         MenuItem21.Visible = False
661         ExcelIsInstalled = MyClassCode.VerifyExcelExistance()
662         MakeObjectsNotVisible(True)
663         PlayMyVideo(sender, e)
664
665     End Sub
666
667     Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e As System.
        EventArgs) Handles MenuItem2.Click
668
669         If MyClassCode.IsValidFile(Me.TxtSaveRecvData.Text) And MyClassCode.
        IsValidFile(Me.TxtSaveSentData.Text) Then
670             Dim fOpt As New Options
671
672             If MyClassCode.IsValidFile(SelectedConfFile) Then
673                 fOpt.SetDefaults(SelectedConfFile, WhichLang, True)
674             Else
675                 fOpt.SetDefaults(SelectedConfFile, WhichLang, False)
676             End If
677
678             fOpt.ShowDialog()
679             If fOpt.bGo Then
680                 sbr.Panels(0).Text = fOpt.CmbComPort.SelectedItem
681                 sbr.Panels(1).Text = fOpt.CmbBaudRate.SelectedItem & " - " & fOpt.
        CmbParity.SelectedItem & " - "
682                 & fOpt.CmbDataBits.SelectedItem & " - " & fOpt.CmbStopBits.
        SelectedItem
683
684                 DefineParamsForSerialPortAndOpenIt(fOpt.IntCommPort, fOpt.IntBaudRate,
        fOpt.IntParityBits, fOpt.IntDataBits, fOpt.IntStopBits)
685
686                 If ProblemOccured = False Then
687                     MenuItem2.Enabled = False
688                     MenuItem3.Enabled = True
689                     ToolBar1.Buttons(0).Enabled = False
690                     ToolBar1.Buttons(1).Enabled = True
691                     TextBox1.Enabled = True
692                     TextBox2.Enabled = True
693                     Button1.Enabled = True
694                     ChkSendChars.Enabled = True
695                     ChkSendChars.Checked = True
696                     ChkSendFile.Enabled = True
697                     TxtSaveSentData.Enabled = False
698                     BtSaveSentData.Enabled = False
699                     TxtSaveRecvData.Enabled = False
700                     GrBxDataSend.Focus()
701                     BtSaveRecvData.Enabled = False
702                     Me.BtClearInputData.Enabled = True
703                     Me.BtClearOutputData.Enabled = True
704                     Me.MenuItem6.Enabled = False
705                     Me.MenuItem7.Enabled = False
706                     Me.MenuItem18.Enabled = False
707                     Me.MItEfarmoges.Enabled = False
708                     Me.MenuItem29.Enabled = False
709                     Me.MenuItem31.Enabled = False
710                     MyClassCode.AddFileHeader(Me.TxtSaveRecvData.Text, 1)
711                     MyClassCode.AddFileHeader(Me.TxtSaveSentData.Text, 2)
712                 End If
713             End If
714             fOpt.Close()
715             fOpt = Nothing
716         Else
717             MsgBox(MyClassCode.GetStr(WhichLang, 48))
718         End If
719
720     End Sub
721
722     Private Function IsPortAvailable(ByVal ComPort As Integer) As Boolean
723         ' This function attempts to open the passed Comm Port. If it is
724         ' available, it returns True, else it returns False. To determine
725         ' availability a Try-Catch block is used.
726         Try

```

```

727         MySerialPort.PortName = "COM" & Trim(Str(ComPort))
728         MySerialPort.BaudRate = 9600
729         MySerialPort.Parity = 0
730         MySerialPort.DataBits = 8
731         MySerialPort.StopBits = 1
732
733         MySerialPort.Open()
734         MySerialPort.Close()
735         Return True
736     Catch
737         ' If it gets here, then the attempt to open the Comm Port was unsuccessful
, therefore it doesn't exist
738         Return False
739     End Try
740 End Function
741
742 Private Sub ToolBar1_ButtonClick(ByVal sender As System.Object, ByVal e As System.
Windows.Forms.ToolBarButtonClickEventArgs) Handles ToolBar1.ButtonClick
743     Select Case ToolBar1.Buttons.IndexOf(e.Button)
744         Case 0
745             MenuItem2_Click(MenuItem2, New System.EventArgs)
746         Case 1
747             MenuItem3_Click(MenuItem3, New System.EventArgs)
748         Case Else
749             MenuItem5_Click(MenuItem5, New System.EventArgs)
750     End Select
751 End Sub
752
753 Private Sub MenuItem3_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MenuItem3.Click
754     MyClassCode.AddFileFooter(Me.TxtSaveRecvData.Text, 1)
755     MyClassCode.AddFileFooter(Me.TxtSaveSentData.Text, 2)
756     MySerialPort.Close()
757     MenuItem2.Enabled = True
758     MenuItem3.Enabled = False
759     ToolBar1.Buttons(0).Enabled = True
760     ToolBar1.Buttons(1).Enabled = False
761     TextBox1.Enabled = False
762     TextBox2.Enabled = False
763     Button1.Enabled = False
764     ChkSendChars.Enabled = False
765     ChkSendChars.Checked = False
766     ChkSendFile.Enabled = False
767     ChkSendFile.Checked = False
768     TxtDefineFileName.Enabled = False
769     BtDefineFileName.Enabled = False
770     TxtSaveSentData.Enabled = True
771     BtSaveSentData.Enabled = True
772     TxtSaveRecvData.Enabled = True
773     BtSaveRecvData.Enabled = True
774     If ChkSendFile.Checked = True Then
775         SrReceive.Close()
776     End If
777     Me.TextBox1.Clear()
778     Me.TextBox2.Clear()
779     Me.TextBox3.Clear()
780     Me.TxtDefineFileName.Clear()
781     sbr.Panels(0).Text = ""
782     sbr.Panels(1).Text = ""
783     Me.BtClearInputData.Enabled = False
784     Me.BtClearOutputData.Enabled = False
785     Me.MenuItem6.Enabled = True
786     Me.MenuItem7.Enabled = True
787     Me.MenuItem18.Enabled = True
788     Me.MITefarmoges.Enabled = True
789     Me.MenuItem29.Enabled = True
790     Me.MenuItem31.Enabled = True
791     Me.TextBox1.Focus()
792 End Sub
793
794 Private Sub Form1_Closing(ByVal sender As Object, ByVal e As System.ComponentModel
.CancelEventArgs) Handles MyBase.Closing
795     If MenuItem3.Enabled Then MenuItem3_Click(MenuItem3, New System.EventArgs)
796 End Sub
797

```

```

798 Private Sub MenuItem5_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MenuItem5.Click
799     KillTheBluddyProcess()
800 End Sub
801
802 Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles Button1.Click
803     Dim CheckFile1 As New IO.FileInfo(Me.TxtSaveSentData.Text)
804     Dim CheckFile2 As New IO.FileInfo(Me.TxtSaveRecvData.Text)
805
806     Try
807         If ChkSendChars.Checked Then
808             MySerialPort.Write(Me.TextBox1.Text)
809             WriteMessage(TextBox1.Text, True, 3)
810             If CheckFile1.Exists Then
811                 WriteToFile(2, TxtSaveSentData.Text, TextBox1.Text)
812             Else
813                 If Me.TxtSaveSentData.Text <> "" Then
814                     Me.TxtSaveSentData.Text = MyClassCode.GetStr(WhichLang, 43)
815                 End If
816             End If
817         Else
818
819             Dim CheckFile3 As New IO.FileInfo(Me.TxtDefineFileName.Text)
820             Dim line As String
821             Dim sr As IO.StreamReader = New IO.StreamReader(Me.TxtDefineFileName.
Text)
822             line = sr.ReadLine
823             While line <> Nothing
824                 MySerialPort.Write(line)
825                 WriteMessage(line, True, 3)
826
827                 WriteToFile(2, Me.TxtSaveSentData.Text, line)
828                 line = sr.ReadLine
829             End While
830         End If
831
832         Catch ex As Exception
833             ' Warn the user.
834             MessageBox.Show(MyClassCode.GetStr(WhichLang, 46))
835         Finally
836             TextBox1.Text = ""
837             TextBox1.Focus()
838         End Try
839 End Sub
840
841 ' This subroutine writes a message to the txtStatus TextBox.
842 Private Sub WriteMessage(ByVal message As String)
843     AppendToText(True, 3, message)
844     TextBox3.SelectionStart = TextBox2.Text.Length
845 End Sub
846
847 ' This subroutine writes a message to the txtStatus TextBox and allows
848 ' the line feed to be suppressed.
849 Private Sub WriteMessage(ByVal message As String, ByVal linefeed As Boolean, ByVal
WhichObject As Integer)
850
851     If linefeed Then
852         message += vbCrLf
853     End If
854
855     Select Case WhichObject
856     Case 2
857         AppendToText(False, 2, message)
858     Case 3
859         AppendToText(False, 3, message)
860     End Select
861 End Sub
862
863 Private Sub ChkSendChars_Click(ByVal sender As Object, ByVal e As System.
EventArgs) Handles ChkSendChars.Click
864     If ChkSendChars.Checked = True Then
865         ChkSendFile.Checked = False
866         TxtDefineFileName.Enabled = False
867         BtDefineFileName.Enabled = False

```

```

868         TextBox1.Enabled = True
869         Me.TextBox1.Focus()
870     End If
871 End Sub
872
873 Private Sub ChkSendFile_Click(ByVal sender As Object, ByVal e As System.EventArgs)
874     Handles ChkSendFile.Click
875         If ChkSendFile.Checked = True Then
876             ChkSendChars.Checked = False
877             TextBox1.Text = ""
878             TextBox1.Enabled = False
879             TxtDefineFileName.Enabled = True
880             BtDefineFileName.Enabled = True
881             Me.TxtDefineFileName.Focus()
882         End If
883     End Sub
884
885 Private Sub BtSaveSentData_Click(ByVal sender As System.Object, ByVal e As System.
886     EventArgs) Handles BtSaveSentData.Click
887     Try
888         Dim myStream As IO.Stream
889         Dim openFileDialog1 As New OpenFileDialog
890
891         openFileDialog1.InitialDirectory = "c:\"
892         openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*"
893         openFileDialog1.FilterIndex = 1
894         openFileDialog1.RestoreDirectory = True
895         openFileDialog1.Multiselect = False
896
897         If openFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK And
898         openFileDialog1.FileName.EndsWith(".txt") Then
899             myStream = openFileDialog1.OpenFile()
900             If Not (myStream Is Nothing) Then
901                 TxtSaveSentData.Text = openFileDialog1.FileName
902                 MySaveSent = openFileDialog1.FileName
903                 myStream.Close()
904             End If
905         End If
906         Catch g As Exception
907             MsgBox(g.Message, MsgBoxStyle.Information, )
908             Me.MenuItem3_Click(sender, e)
909         End Try
910     End Sub
911
912 Private Sub BtSaveRecvData_Click(ByVal sender As System.Object, ByVal e As System.
913     EventArgs) Handles BtSaveRecvData.Click
914     Try
915         Dim myStream As IO.Stream
916         Dim openFileDialog1 As New OpenFileDialog
917
918         openFileDialog1.InitialDirectory = "c:\"
919         openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*"
920         openFileDialog1.FilterIndex = 1
921         openFileDialog1.RestoreDirectory = True
922         openFileDialog1.Multiselect = False
923
924         If openFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK And
925         openFileDialog1.FileName.EndsWith(".txt") Then
926             myStream = openFileDialog1.OpenFile()
927             If Not (myStream Is Nothing) Then
928                 TxtSaveRecvData.Text = openFileDialog1.FileName
929                 MySaveRecv = openFileDialog1.FileName
930                 myStream.Close()
931             End If
932         End If
933         Catch g As Exception
934             MsgBox(g.Message, MsgBoxStyle.Information)
935             Me.MenuItem3_Click(sender, e)
936         End Try
937     End Sub
938
939 Private Sub BtDefineFileName_Click(ByVal sender As System.Object, ByVal e As
940     System.EventArgs) Handles BtDefineFileName.Click
941     Try

```

```

937         Dim myStream As IO.Stream
938         Dim openFileDialog1 As New OpenFileDialog
939
940         openFileDialog1.InitialDirectory = "c:\"
941         openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*"
942         openFileDialog1.FilterIndex = 1
943         openFileDialog1.RestoreDirectory = True
944         openFileDialog1.Multiselect = False
945
946         If openFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK And
openFileDialog1.FileName.EndsWith(".txt") Then
947             myStream = openFileDialog1.OpenFile()
948             If Not (myStream Is Nothing) Then
949                 TxtDefineFileName.Text = openFileDialog1.FileName
950                 MyFileNameSend = openFileDialog1.FileName
951                 myStream.Close()
952             End If
953         End If
954     Catch g As Exception
955         MsgBox(g.Message, MsgBoxStyle.Information, )
956     End Try
957 End Sub
958
959
960 Public Sub ChangeLanguage(ByVal WhichLanguage As Integer)
961     'WhichLanguage = 0 for Greek
962     'WhichLanguage = 1 for English
963     'WhichLanguage = 1 for Italy
964     'WhichLanguage = 1 for Spain
965     'WhichLanguage = 1 for French
966
967     ShowSelectedFile(SelectedConfFile, WhichLanguage)
968     WhichLang = WhichLanguage
969
970     MenuItem8.Checked = False
971     MenuItem9.Checked = False
972     MenuItem35.Checked = False
973     MenuItem36.Checked = False
974     MenuItem37.Checked = False
975     Select Case WhichLanguage
976     Case 0
977         MenuItem8.Checked = True
978     Case 1
979         MenuItem9.Checked = True
980     Case 2
981         MenuItem35.Checked = True
982     Case 3
983         MenuItem36.Checked = True
984     Case 4
985         MenuItem37.Checked = True
986     End Select
987
988     Me.GrBxDataSave.Text = MyClassCode.GetStr(WhichLanguage, 0)
989     Me.LblSaveSentData.Text = MyClassCode.GetStr(WhichLanguage, 1)
990     Me.LblSaveRecvData.Text = MyClassCode.GetStr(WhichLanguage, 2)
991     Me.GrBxDataSend.Text = MyClassCode.GetStr(WhichLanguage, 3)
992     Me.ChkSendChars.Text = MyClassCode.GetStr(WhichLanguage, 4)
993     Me.ChkSendFile.Text = MyClassCode.GetStr(WhichLanguage, 5)
994     Me.LblInputData.Text = MyClassCode.GetStr(WhichLanguage, 6)
995     Me.LblOutputData.Text = MyClassCode.GetStr(WhichLanguage, 7)
996     Me.Button1.Text = MyClassCode.GetStr(WhichLanguage, 8)
997     Me.MenuItem1.Text = MyClassCode.GetStr(WhichLanguage, 9)
998     Me.MenuItem2.Text = MyClassCode.GetStr(WhichLanguage, 10)
999     Me.MenuItem3.Text = MyClassCode.GetStr(WhichLanguage, 11)
1000    Me.MenuItem5.Text = MyClassCode.GetStr(WhichLanguage, 12)
1001    Me.MenuItem6.Text = MyClassCode.GetStr(WhichLanguage, 13)
1002    Me.MenuItem7.Text = MyClassCode.GetStr(WhichLanguage, 14)
1003    Me.MenuItem8.Text = MyClassCode.GetStr(WhichLanguage, 15)
1004    Me.MenuItem35.Text = MyClassCode.GetStr(WhichLanguage, 259)
1005    Me.MenuItem36.Text = MyClassCode.GetStr(WhichLanguage, 260)
1006    Me.MenuItem37.Text = MyClassCode.GetStr(WhichLanguage, 261)
1007    Me.MenuItem9.Text = MyClassCode.GetStr(WhichLanguage, 16)
1008    Me.MnConfAllTabs.Text = MyClassCode.GetStr(WhichLanguage, 17)
1009    Me.MnConfLanguage.Text = MyClassCode.GetStr(WhichLanguage, 18)
1010    Me.MnConfGraphs.Text = MyClassCode.GetStr(WhichLanguage, 311)

```



```

1011     Me.MnConfSerial.Text = MyClassCode.GetStr(WhichLanguage, 145)
1012     Me.MnConfThrust.Text = MyClassCode.GetStr(WhichLanguage, 149)
1013     Me.MenuItem18.Text = MyClassCode.GetStr(WhichLanguage, 24)
1014     Me.MenuItem19.Text = MyClassCode.GetStr(WhichLanguage, 25)
1015     Me.MenuItem20.Text = MyClassCode.GetStr(WhichLanguage, 26)
1016     Me.MenuItem21.Text = MyClassCode.GetStr(WhichLanguage, 27)
1017     Me.MenuItem22.Text = MyClassCode.GetStr(WhichLanguage, 28)
1018     Me.MenuItem23.Text = MyClassCode.GetStr(WhichLanguage, 29)
1019     Me.MenuItem24.Text = MyClassCode.GetStr(WhichLanguage, 30)
1020     Me.MenuItem25.Text = MyClassCode.GetStr(WhichLanguage, 31)
1021     Me.MenuItem26.Text = MyClassCode.GetStr(WhichLanguage, 32)
1022     Me.MenuItem27.Text = MyClassCode.GetStr(WhichLanguage, 33)
1023     Me.ToolBarButton1.ToolTipText = MyClassCode.GetStr(WhichLanguage, 34)
1024     Me.ToolBarButton2.ToolTipText = MyClassCode.GetStr(WhichLanguage, 35)
1025     Me.ToolBarButton4.ToolTipText = MyClassCode.GetStr(WhichLanguage, 36)
1026     Me.Text = MyClassCode.GetStr(WhichLanguage, 37)
1027     Me.BtClearInputData.Text = MyClassCode.GetStr(WhichLanguage, 38)
1028     Me.BtClearOutputData.Text = MyClassCode.GetStr(WhichLanguage, 39)
1029     Me.GrBxPreviewData.Text = MyClassCode.GetStr(WhichLanguage, 40)
1030     Me.LblInputData.Text = MyClassCode.GetStr(WhichLanguage, 41)
1031     Me.LblOutputData.Text = MyClassCode.GetStr(WhichLanguage, 42)
1032     Me.MITefarmoges.Text = MyClassCode.GetStr(WhichLanguage, 47)
1033     Me.MenuItem29.Text = MyClassCode.GetStr(WhichLanguage, 145)
1034     Me.MenuItem31.Text = MyClassCode.GetStr(WhichLanguage, 147)
1035     Me.MenuItem32.Text = MyClassCode.GetStr(WhichLanguage, 148)
1036     Me.MnItExit.Text = MyClassCode.GetStr(WhichLanguage, 12)
1037     Me.MenuItem34.Text = MyClassCode.GetStr(WhichLang, 198)
1038     Me.BtVideoStart.Text = MyClassCode.GetStr(WhichLang, 265)
1039     Me.BtVideoStop.Text = MyClassCode.GetStr(WhichLang, 266)
1040     Me.BtVideoPause.Text = MyClassCode.GetStr(WhichLang, 267)
1041 End Sub
1042
1043 Private Sub MenuItem8_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MenuItem8.Click
1044     ChangeLanguage(0)
1045 End Sub
1046
1047 Private Sub MenuItem9_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MenuItem9.Click
1048     ChangeLanguage(1)
1049 End Sub
1050
1051 Private Sub WriteToFile(ByVal WhichApplication As Integer, ByVal MyFileName As
String, ByVal StringToWrite As String)
1052     Dim CheckFile As New IO.FileInfo(MyFileName)
1053     Dim sw As IO.StreamWriter
1054
1055     If CheckFile.Exists Then
1056         sw = CheckFile.AppendText()
1057         sw.WriteLine(StringToWrite)
1058         sw.Close()
1059     End If
1060 End Sub
1061
1062 Private Sub CallParametersForm(ByVal WhichSection As Integer, ByVal WhichLanguage
As Integer, ByVal sender As System.Object, ByVal e As System.EventArgs)
1063     Dim fPar As New FrmParameters
1064     Dim Undopause As Boolean = False
1065     Me.PauseVideo_if_Necessary(Undopause)
1066     fPar.DefaultConfFile = SelectedConfFile
1067     fPar.WhichLang(WhichLanguage)
1068     fPar.WhichEnabled(WhichSection)
1069     fPar.ShowDialog()
1070     fPar.Close()
1071     SelectedConfFile = fPar.DefaultConfFile
1072     ActivateConfFile(fPar.DefaultConfFile)
1073     fPar = Nothing
1074     Me.UndoPause_if_Necesssary(Undopause)
1075 End Sub
1076
1077 Private Sub MnConfAllTabs_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MnConfAllTabs.Click
1078     CallParametersForm(0, WhichLang, sender, e)
1079 End Sub
1080

```

```

1081 Private Sub MnConfSerial_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MnConfSerial.Click
1082     CallParametersForm(2, WhichLang, sender, e)
1083 End Sub
1084
1085 Private Sub MnConfLanguage_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MnConfLanguage.Click
1086     CallParametersForm(1, WhichLang, sender, e)
1087 End Sub
1088
1089 Private Sub MnConfGraphs_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MnConfGraphs.Click
1090     CallParametersForm(4, WhichLang, sender, e)
1091 End Sub
1092
1093 Private Sub MenuItem13_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs)
1094     CallParametersForm(4, WhichLang, sender, e)
1095 End Sub
1096
1097 Private Sub MenuItem14_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs)
1098     CallParametersForm(5, WhichLang, sender, e)
1099 End Sub
1100
1101 Private Sub MenuItem15_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs)
1102     CallParametersForm(6, WhichLang, sender, e)
1103 End Sub
1104
1105 Private Sub BtClearInputData_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtClearInputData.Click
1106     Me.TextBox2.Clear()
1107     If Me.TextBox1.Enabled Then
1108         Me.TextBox1.Focus()
1109     Else
1110         Me.TxtDefineFileName.Focus()
1111     End If
1112 End Sub
1113
1114 Private Sub BtClearOutputData_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtClearOutputData.Click
1115     Me.TextBox3.Clear()
1116     If Me.TextBox1.Enabled Then
1117         Me.TextBox1.Focus()
1118     Else
1119         Me.TxtDefineFileName.Focus()
1120     End If
1121 End Sub
1122 Private Function FindOutInitialLang(ByVal MyFileName As String) As Integer
1123     Try
1124         Dim sr As StreamReader = New StreamReader(MyFileName)
1125         Dim WhichLanguage As Integer
1126         WhichLanguage = Val(sr.ReadLine())
1127
1128         Return WhichLanguage
1129     Catch
1130         ' If it gets here, then the attempt to read confs file was unsuccessful.
1131         Return 0 ' Greek
1132     End Try
1133 End Function
1134
1135 Public Sub ActivateConfFile(ByVal MyFileName As String)
1136     If MyClassCode.IsValidFile(MyFileName) Then
1137         Try
1138             ' Create an instance of StreamReader to read from a file.
1139
1140             ChangeLanguage(Val(MyClassCode.FindParamValue(MyFileName, "Language",
WhichLang)))
1141             TxtSaveRecvData.Text = MyClassCode.FindParamValue(MyFileName, "Input
File", WhichLang)
1142             TxtSaveSentData.Text = MyClassCode.FindParamValue(MyFileName, "Output
File", WhichLang)
1143             TxtDefineFileName.Text = MyClassCode.FindParamValue(MyFileName, "File
To Send", WhichLang)

```

```

1143         GrBxDataSend.Focus ()
1144         TxtSaveRecvData.Enabled = True
1145         BtSaveRecvData.Enabled = True
1146         TxtSaveSentData.Enabled = True
1147         BtSaveSentData.Enabled = True
1148         TxtDefineFileName.Enabled = False
1149         BtDefineFileName.Enabled = False
1150     Catch E As Exception
1151         MsgBox(MyClassCode.GetStr(WhichLang, 44), MsgBoxStyle.Information, 0)
1152     End Try
1153 End If
1154 End Sub
1155
1156 Public Sub DeActivateConfFile ()
1157     ChangeLanguage (0)
1158
1159     SelectedConfFile = ""
1160     TxtSaveSentData.Text = ""
1161     TxtSaveSentData.Enabled = True
1162     BtSaveSentData.Enabled = True
1163
1164     TxtSaveRecvData.Text = ""
1165     TxtSaveRecvData.Enabled = True
1166     BtSaveRecvData.Enabled = True
1167
1168     TxtDefineFileName.Text = ""
1169     sbr.Panels (2).Text = ""
1170
1171     TxtSaveSentData.Focus ()
1172
1173 End Sub
1174
1175 Private Sub MenuItem19_Click (ByVal sender As System.Object, ByVal e As System.
1176     EventArgs) Handles MenuItem19.Click
1177     Dim fSelConfFiles As New SelectConfFiles
1178     Dim Undopause As Boolean = False
1179     Me.PauseVideo_if_Necessary (Undopause)
1180     fSelConfFiles.PrevSelectedConfFile = SelectedConfFile
1181     fSelConfFiles.WhichLang (WhichLang)
1182     fSelConfFiles.ShowDialog ()
1183     If fSelConfFiles.NodeLabel <> "" Then
1184         SelectedConfFile = fSelConfFiles.NodeLabel
1185     End If
1186     fSelConfFiles.Close ()
1187     fSelConfFiles = Nothing
1188
1189     If SelectedConfFile <> "" Then
1190         UpdateRegistry (SelectedConfFile)
1191         ActivateConfFile (SelectedConfFile)
1192         ShowSelectedFile (SelectedConfFile, WhichLang)
1193     End If
1194     Me.UndoPause_if_Necesssary (Undopause)
1195 End Sub
1196
1197 Private Sub ShowSelectedFile (ByVal MyFilename As String, ByVal WhichLang As
1198     Integer)
1199     If MyFilename = "" Then
1200         sbr.Panels (2).Text = ""
1201     Else
1202         sbr.Panels (2).Text = MyClassCode.GetStr (WhichLang, 45) + " " + MyFilename
1203     End If
1204 End Sub
1205
1206 Private Sub MenuItem20_Click (ByVal sender As System.Object, ByVal e As System.
1207     EventArgs) Handles MenuItem20.Click
1208     DeActivateConfFile ()
1209     UpdateRegistry ("c:\windows\myconfs.ini")
1210 End Sub
1211
1212 Private Sub UpdateRegistry (ByVal MySelectedConfFile As String)
1213     Dim key As RegistryKey
1214
1215     key = Registry.LocalMachine.OpenSubKey ("Software\MyPtyx", True)
1216     If key Is Nothing Then
1217         key = Registry.LocalMachine.CreateSubKey ("Software\MyPtyx")

```

```

1215         End If
1216
1217         key.SetValue("MyPtyx", MySelectedConfFile)
1218     End Sub
1219
1220     Public Sub MyPrint(ByVal MyFileName As String)
1221         DocumentToPrint = MyFileName
1222         'Create the document and attach an event handler
1223         Dim MyDoc As New PrintDocument
1224         AddHandler MyDoc.PrintPage, AddressOf MyDoc_PrintPage
1225         'Allow the user to choose a printer and specify other settings
1226         Dim dlgSettings As New PrintDialog
1227         dlgSettings.Document = MyDoc
1228         Dim Result As DialogResult = dlgSettings.ShowDialog()
1229         'if the user clicked ok, print the document
1230         If Result = Windows.Forms.DialogResult.OK Then
1231             'This method returns immediately, before the print job starts
1232             'The PrintPage event will fire asynchronously.
1233             MyDoc.Print()
1234         End If
1235     End Sub
1236
1237     Private Sub MyDoc_PrintPage(ByVal sender As Object, ByVal e As PrintPageEventArgs)
1238         'Define the font
1239         Dim MyFont As New Font("Times New Roman", 12)
1240         Dim MyFont1 As New Font("Times New Roman", 20)
1241         Dim MyFont2 As New Font("Times New Roman", 14)
1242         ' Stop
1243         'Determine the position on page.
1244         'In this case, we read the margin settings
1245         '(although there is nothing that prevents your code from going outside the
margin bounds.)
1246         Dim x As Single = e.MarginBounds.Left
1247         Dim y As Single = e.MarginBounds.Top
1248         'Determine the height of a line (based on the font used).
1249         Dim LineHeight As Single = MyFont.GetHeight(e.Graphics)
1250         Dim sr As StreamReader = New StreamReader(DocumentToPrint)
1251         Dim line As String
1252         line = sr.ReadLine()
1253         If line <> Nothing Or sr.EndOfStream = False Then
1254
1255             y += 100
1256             e.Graphics.DrawString(MyClassCode.GetStr(WhichLang, 112), MyFont1, Brushes
.Black, x - 10, y)
1257
1258             y += 80
1259             e.Graphics.DrawString(MyClassCode.GetStr(WhichLang, 113), MyFont2, Brushes
.Black, e.MarginBounds.Left - 4 + (e.MarginBounds.Right - e.MarginBounds.Left) / 2
, y)
1260
1261             y += 4
1262             While line <> Nothing Or sr.EndOfStream = False
1263                 'Draw the text with a black brush, using the font and coordinates we
have determined.
1264                 e.Graphics.DrawString(line, MyFont, Brushes.Black, x, y)
1265                 'Move down the equivalent spacing of one line
1266                 y += LineHeight
1267                 line = sr.ReadLine()
1268             End While
1269
1270             sr.Close()
1271         Else
1272
1273         End If
1274     End Sub
1275
1276     Private Sub MenuItem22_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MenuItem22.Click
1277         If MyClassCode.IsValidFile(TxtSaveSentData.Text) Then
1278             If CheckIfThereAreData(TxtSaveSentData.Text, WhichLang) Then
1279                 MyPrint(TxtSaveSentData.Text)
1280             End If
1281         Else
1282             FileShowError(TxtSaveSentData.Text)
1283         End If

```

```

1284     End Sub
1285
1286     Public Sub FileShowError(ByVal MyFileName As String)
1287         MsgBox(MyClassCode.GetStr(WhichLang, 107) & MyFileName & _
1288             MyClassCode.GetStr(WhichLang, 108), 0, MyClassCode.GetStr
1289             (WhichLang, 109))
1290     End Sub
1291
1292     Private Sub MenuItem23_Click(ByVal sender As System.Object, ByVal e As System.
1293     EventArgs) Handles MenuItem23.Click
1294         If MyClassCode.IsValidFile(TxtSaveRecvData.Text) Then
1295             If CheckIfThereAreData(TxtSaveRecvData.Text, WhichLang) Then
1296                 MyPrint(TxtSaveRecvData.Text)
1297             End If
1298         Else
1299             FileShowError(TxtSaveRecvData.Text)
1300         End If
1301     End Sub
1302
1303     Private Sub MenuItem24_Click(ByVal sender As System.Object, ByVal e As System.
1304     EventArgs) Handles MenuItem24.Click
1305         If MyClassCode.IsValidFile(TxtDefineFileName.Text) Then
1306             If CheckIfThereAreData(TxtDefineFileName.Text, WhichLang) Then
1307                 MyPrint(TxtDefineFileName.Text)
1308             End If
1309         Else
1310             FileShowError(TxtDefineFileName.Text)
1311         End If
1312     End Sub
1313
1314     Private Function CheckIfThereAreData(ByVal MyFileName As String, ByVal WhichLang
1315     As Integer) As Boolean
1316         Dim sr As StreamReader = New StreamReader(MyFileName)
1317         Dim line As String
1318         Dim status As Boolean
1319
1320         line = sr.ReadLine()
1321         If line = Nothing And sr.EndOfStream = True Then
1322             MsgBox(MyClassCode.GetStr(WhichLang, 110), 0, MyClassCode.GetStr(WhichLang
1323             , 109))
1324             status = False
1325         Else
1326             status = True
1327         End If
1328
1329         sr.Close()
1330         Return status
1331     End Function
1332
1333     Private Sub MenuItem26_Click(ByVal sender As System.Object, ByVal e As System.
1334     EventArgs) Handles MenuItem26.Click
1335         Try
1336             Help.ShowHelp(Me, CurDir() & "\\Help\MyPtyxHelp.chm", HelpNavigator.
1337             TableOfContents)
1338             'Να είναι καλά ο άνθρωπος
1339             'http://www.codeproject.com/dotnet/HelpIntegrationInDotNet.asp?df=100&
1340             forumid=37587&exp=0&select=1325919
1341         Catch
1342             MsgBox(MyClassCode.GetStr(WhichLang, 111), 0, MyClassCode.GetStr(WhichLang
1343             , 109))
1344         End Try
1345     End Sub
1346
1347     Private Sub MenuItem27_Click(ByVal sender As System.Object, ByVal e As System.
1348     EventArgs) Handles MenuItem27.Click
1349         Dim fVersion As New Version
1350         Dim Undopause As Boolean = False
1351         Me.PauseVideo_if_Necessary(Undopause)
1352         fVersion.WhichLang(WhichLang)
1353         fVersion.ShowDialog()
1354         fVersion.Close()
1355         fVersion = Nothing
1356         Me.UndoPause_if_Necesssary(Undopause)
1357     End Sub

```

```

1349
1350 Private Sub KillTheBluddyProcess()
1351     Dim myProcess1 As New Process
1352
1353     myProcess1 = Process.GetCurrentProcess()
1354
1355     myProcess1.Kill()
1356     Me.Close()
1357 End Sub
1358
1359 Private Sub Form1_Closed(ByVal sender As Object, ByVal e As System.EventArgs)
1360     KillTheBluddyProcess()
1361 End Sub
1362
1363 Public Sub DefineParamsForSerialPortAndOpenIt(ByVal ComPort As Integer, ByVal
1364     BaudRate As Integer, ByVal Parity As Integer, ByVal DataBits As Integer, ByVal
1365     StopBits As Integer)
1366     MySerialPort.PortName = "COM" & Trim(Str(ComPort))
1367
1368     MySerialPort.BaudRate = BaudRate
1369     MySerialPort.Parity = Parity
1370     MySerialPort.DataBits = DataBits
1371     MySerialPort.StopBits = StopBits
1372
1373     Try
1374         MySerialPort.Open()
1375         ProblemOccured = False
1376     Catch ex As Exception
1377         ProblemOccured = True
1378         MsgBox(MyClassCode.GetStr(WhichLang, 138), 0, MyClassCode.GetStr(WhichLang
1379 , 109))
1380     End Try
1381 End Sub
1382
1383 Private Sub MySerialPort_DataReceived(ByVal sender As System.Object, ByVal e As
1384     System.IO.Ports.SerialDataReceivedEventArgs) Handles MySerialPort.DataReceived
1385     Dim MyBuffer(12) As Byte
1386     Dim i As Integer
1387
1388     Try
1389         If MySerialPort.BytesToRead >= 12 Then
1390             MySerialPort.Read(MyBuffer, 0, 12)
1391             For i = 0 To 11
1392                 WriteMessage(Asc(Chr(MyBuffer(i))), False, 2)
1393                 If TxtSaveRecvData.Text <> "" Then
1394                     WriteToMyFile(1, TxtSaveRecvData.Text, Asc(Chr(MyBuffer(i))))
1395                 End If
1396             Next i
1397         End If
1398     Catch exc As Exception
1399         ' An exception is raised when there is no information to read.
1400         ' Don't do anything here, just let the exception go.
1401     End Try
1402 End Sub
1403
1404 Private Sub AppendToText(ByVal UsevbCrLf As Boolean, ByVal WhichTextBox As Integer
1405 , ByVal MyText As String)
1406
1407     WhichIsTheTextBox = WhichTextBox
1408     If UsevbCrLf Then
1409         MyText += vbCrLf
1410     End If
1411     WhichIsTheMessage = MyText
1412
1413     Select Case WhichTextBox
1414     Case 2
1415         If Me.TextBox2.InvokeRequired Then
1416             Me.TextBox2.BeginInvoke(New InvokeDelegate(AddressOf
1417 InvokeMethod))
1418         Else
1419             Me.TextBox2.Text += MyText
1420         End If
1421     Case 3
1422         If Me.TextBox3.InvokeRequired Then

```

```

1417             Me.TextBox3.BeginInvoke(New InvokeDelegate(AddressOf
InvokeMethod))
1418         Else
1419             Me.TextBox3.Text += MyText
1420         End If
1421     End Select
1422 End Sub
1423 Public Sub InvokeMethod()
1424     Select Case WhichIsTheTextBox
1425     Case 2
1426         Me.TextBox2.Text += WhichIsTheMessage
1427     Case 3
1428         Me.TextBox3.Text += WhichIsTheMessage
1429     End Select
1430 End Sub
1431
1432 Public Sub MakeObjectsNotVisible(ByVal IsTrue As Boolean)
1433     GrBxDatSave.Visible = Not IsTrue
1434     GrBxDatSend.Visible = Not IsTrue
1435     GrBxPreviewData.Visible = Not IsTrue
1436     LblSaveSentData.Visible = Not IsTrue
1437     LblSaveRecvData.Visible = Not IsTrue
1438     TxtSaveSentData.Visible = Not IsTrue
1439     TxtSaveRecvData.Visible = Not IsTrue
1440     BtSaveSentData.Visible = Not IsTrue
1441     BtSaveRecvData.Visible = Not IsTrue
1442     ChkSendChars.Visible = Not IsTrue
1443     ChkSendFile.Visible = Not IsTrue
1444     TextBox1.Visible = Not IsTrue
1445     TxtDefineFileName.Visible = Not IsTrue
1446     BtDefineFileName.Visible = Not IsTrue
1447     Button1.Visible = Not IsTrue
1448     LblInputData.Visible = Not IsTrue
1449     LblOutputData.Visible = Not IsTrue
1450     TextBox2.Visible = Not IsTrue
1451     TextBox3.Visible = Not IsTrue
1452     BtClearInputData.Visible = Not IsTrue
1453     BtClearOutputData.Visible = Not IsTrue
1454     ToolBar1.Visible = Not IsTrue
1455     sbr.Visible = Not IsTrue
1456     MenuItem1.Visible = Not IsTrue
1457     PnForVideo.Visible = IsTrue
1458     Me.Size = IIf(IsTrue = True, New System.Drawing.Size(923, 572), New System.
Drawing.Size(923, 675))
1459
1460 End Sub
1461
1462 Private Sub MenuItem29_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MenuItem29.Click
1463     Static Dim Undopause As Boolean = False
1464     Me.PauseVideo_if_Necessary(Undopause)
1465
1466     MakeObjectsNotVisible(False)
1467
1468 End Sub
1469
1470 Private Sub MenuItem34_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MenuItem32.Click
1471     Dim Undopause As Boolean = False
1472     If AxWMPVideoPlayer.playState = WMPLib.WMPPlayState.wmppsPaused Then :
Undopause = True : End If
1473     MakeObjectsNotVisible(True)
1474     Me.UndoPause_if_Necesssary(Undopause)
1475 End Sub
1476
1477 Private Sub MnItExit_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MnItExit.Click
1478     KillTheBluddyProcess()
1479 End Sub
1480
1481 Private Sub MenuItem31_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MenuItem31.Click
1482     Dim NumOfProcs As Integer
1483     Dim Undopause As Boolean = False
1484     Me.PauseVideo_if_Necessary(Undopause)

```

```

1485     NumOfProcs = MyClassCode.NumOfExcelProcs()
1486     Dim MyMatrix(NumOfProcs) As Integer
1487     MyMatrix = MyClassCode.FindExcelProcesses()
1488     Dim fThrust As New Thrust
1489     fThrust.ExcelIsInstalled = ExcelIsInstalled
1490     fThrust.NumOfExistingExcelProcs = NumOfProcs
1491     fThrust.RunningExcelProcs = MyMatrix
1492     fThrust.WhichLang(WhichLang)
1493     fThrust.SetDefaults(SelectedConfFile)
1494     fThrust.ShowDialog()
1495     fThrust.Close()
1496     fThrust = Nothing
1497     Me.UndoPause_if_Necesssary(Undopause)
1498
1499 End Sub
1500
1501 Private Sub MnThrust_Click(ByVal sender As System.Object, ByVal e As System.
1502     EventArgs) Handles MnConfThrust.Click
1503     CallParametersForm(3, WhichLang, sender, e)
1504 End Sub
1505
1506 Private Sub MenuItem34_Click_1(ByVal sender As System.Object, ByVal e As System.
1507     EventArgs) Handles MenuItem34.Click
1508     Dim fSpecialFile As New SpecialFile
1509     Dim Undopause As Boolean = False
1510     Me.PauseVideo_if_Necessary(Undopause)
1511     fSpecialFile.WhichLang(WhichLang)
1512     fSpecialFile.SetDefaults(SelectedConfFile)
1513     fSpecialFile.ShowDialog()
1514     fSpecialFile.Close()
1515     fSpecialFile = Nothing
1516     Me.UndoPause_if_Necesssary(Undopause)
1517 End Sub
1518
1519 Private Sub MenuItem35_Click(ByVal sender As System.Object, ByVal e As System.
1520     EventArgs) Handles MenuItem35.Click
1521     ChangeLanguage(2)
1522 End Sub
1523
1524 Private Sub MenuItem36_Click(ByVal sender As System.Object, ByVal e As System.
1525     EventArgs) Handles MenuItem36.Click
1526     ChangeLanguage(3)
1527 End Sub
1528
1529 Private Sub MenuItem37_Click(ByVal sender As System.Object, ByVal e As System.
1530     EventArgs) Handles MenuItem37.Click
1531     ChangeLanguage(4)
1532 End Sub
1533
1534 Public Sub PlayMyVideo(ByVal sender As System.Object, ByVal e As System.EventArgs)
1535     AxWMPVideoPlayer.URL = Application.StartupPath & "\Videos\PtyxVideo.mpg"
1536     BtVideoStart_Click(sender, e)
1537 End Sub
1538
1539 Private Sub BtVideoStart_Click(ByVal sender As System.Object, ByVal e As System.
1540     EventArgs) Handles BtVideoStart.Click, MyBase.Click
1541     AxWMPVideoPlayer.Ctlcontrols.play()
1542     VideoEnabledButtons("Start")
1543 End Sub
1544
1545 Private Sub BtVideoStop_Click(ByVal sender As System.Object, ByVal e As System.
1546     EventArgs) Handles BtVideoStop.Click
1547     AxWMPVideoPlayer.Ctlcontrols.stop()
1548     AxWMPVideoPlayer.Ctlcontrols.previous()
1549     VideoEnabledButtons("Stop")
1550     BtVideoPause.Text = MyClassCode.GetStr(WhichLang, 267)
1551 End Sub
1552
1553 Private Sub BtVideoPause_Click(ByVal sender As System.Object, ByVal e As System.
1554     EventArgs) Handles BtVideoPause.Click
1555     BtVideoPause.Refresh()
1556     If Trim(BtVideoPause.Text) = Trim(MyClassCode.GetStr(WhichLang, 267)) Then
1557         BtVideoPause.Text = MyClassCode.GetStr(WhichLang, 268)
1558         AxWMPVideoPlayer.Ctlcontrols.pause()
1559         VideoEnabledButtons("Pause")

```



```

1552     Else
1553         BtVideoPause.Text = MyClassCode.GetStr(WhichLang, 267)
1554         AxWMPVideoPlayer.Ctlcontrols.play()
1555         VideoEnabledButtons("Start")
1556     End If
1557     BtVideoPause.Refresh()
1558 End Sub
1559
1560 Public Sub VideoEnabledButtons(ByVal WhoIsTheCaller As String)
1561     Select Case WhoIsTheCaller
1562         Case "Start"
1563             BtVideoStart.Enabled = False
1564             BtVideoPause.Enabled = True
1565             BtVideoStop.Enabled = True
1566         Case "Stop"
1567             BtVideoStart.Enabled = True
1568             BtVideoPause.Enabled = False
1569             BtVideoStop.Enabled = False
1570         Case "Pause"
1571             BtVideoStart.Enabled = False
1572             BtVideoPause.Enabled = True
1573             BtVideoStop.Enabled = True
1574     End Select
1575 End Sub
1576
1577 Public Sub DoPauseVideo(ByVal Indeed As Boolean, ByVal sender As System.Object,
1578 ByVal e As System.EventArgs)
1579     'If another form is opened it is not appropriate to have video running so I
1580     pause it and when the form is closed it continues
1581     If BtVideoStart.Enabled = False Then
1582         If Indeed Then
1583             If Trim(BtVideoPause.Text) = Trim(MyClassCode.GetStr(WhichLang, 267))
1584             Then
1585                 BtVideoPause_Click(sender, e)
1586                 WasAlreadyPaused = False
1587             Else
1588                 WasAlreadyPaused = True
1589             End If
1590         Else
1591             BtVideoPause.Text = MyClassCode.GetStr(WhichLang, 268)
1592             BtVideoPause_Click(sender, e)
1593         End If
1594     End If
1595 End Sub
1596 Public Sub PauseVideo_if_Necessary(ByRef Undopause As Boolean)
1597     If AxWMPVideoPlayer.playState = WMPLib.WMPPlayState.wmppsPlaying Then
1598         DoPauseVideo(True, New System.Object, New System.EventArgs)
1599         Undopause = True
1600     End If
1601 End Sub
1602 Public Sub UndoPause_if_Necesssary(ByRef Undopause As Boolean)
1603     If Undopause = True Then : DoPauseVideo(False, New System.Object, New System.
1604     EventArgs) : End If
1605 End Sub
1606 End Class

```

```

1 Imports RS232Tester.SpecialFile
2 Imports System.ComponentModel
3 Imports System.IO
4 Imports System.Timers
5 Imports System.Threading
6
7 Public Class LoadingDataForm
8     Private Const MaximumRPM As Integer = 20000, MaximumPWM As Integer = 10000
9
10    Private MyClassCode As New MyPtyxClass
11    Delegate Sub InvokeDelagate()
12    Private line As String
13    Public Cancel As Boolean = False
14    Public Line3 As String = "", line4 As String = "", Line5 As String = ""
15    Public TheSelectedConfFile, ConfSendDataFile, ConfRepeat, ConfRepeatTimes,
16    SystemDecimalSeperator, ImportFile As String
17    Public WhichIsTheLang, Current50ada, Synolo50adwn, MatrixValues(-1),
18    ImportFromFileMatrix(), MatrixDimension, ImportedFileLines As Integer
19    Public PreviousPercentage As Integer
20
21    Public Sub New()
22        InitializeComponent()
23        Me.LoadFileValuesWorker.WorkerSupportsCancellation = True
24    End Sub
25
26    Private Sub LoadingDataForm_Load(ByVal sender As System.Object, ByVal e As System.
27    EventArgs) Handles MyBase.Load
28        Me.Left = (My.Computer.Screen.WorkingArea.Size.Width - Me.Width) / 2
29        Me.Top = (My.Computer.Screen.WorkingArea.Size.Height - Me.Height) / 2
30        Me.Timer1.Start()
31    End Sub
32
33    Public Sub WhichLang(ByVal WhichLang As Integer)
34        WhichIsTheLang = WhichLang
35
36        Me.Text = MyClassCode.GetStr(WhichIsTheLang, 236)
37        Me.LblFileName.Text = MyClassCode.GetStr(WhichIsTheLang, 81)
38        Me.RTBLoadFileDisplayText.Text = MyClassCode.GetStr(WhichIsTheLang, 262)
39        Me.BtCancelLoadFile.Text = MyClassCode.GetStr(WhichIsTheLang, 264)
40    End Sub
41
42    Public Sub SetDefaults(ByVal SelectedConfFile As String)
43        If MyClassCode.IsValidFile(SelectedConfFile) Then
44            TheSelectedConfFile = SelectedConfFile
45            ActivateConfFile(SelectedConfFile)
46        Else
47            MsgBox("Before proceed to any work select a valid Save File." & ChrW(10) &
48            ChrW(13) & "The selection of file will clear any work at time!" _
49            , MsgBoxStyle.Information, MyClassCode.GetStr(WhichIsTheLang, 109))
50            TheSelectedConfFile = ""
51        End If
52    End Sub
53
54    Private Sub ActivateConfFile(ByVal MyFileName As String)
55        Try
56            If MyClassCode.IsValidFile(MyFileName) Then
57                TxtSaveData.Text = MyFileName
58                ConfSendDataFile = MyFileName
59            Else
60                End If
61            Catch E As Exception
62                ' Do nothing for now
63            End Try
64    End Sub
65
66    Private Sub AdjustAccordingToConfFile(ByVal FileName As String, ByVal sender As
67    System.Object, ByVal e As DoWorkEventArgs) 's
68        Try
69            Dim CheckFile As New IO.FileInfo(FileName)
70            Dim MaxLength As Integer
71            MaxLength = CheckFile.Length
72            Dim sr As System.IO.StreamReader = CheckFile.OpenText()
73
74            Dim CurrentLoadPosition As Integer = 0
75            Dim line As String

```

```

71     Dim met As Integer
72     met = 0
73     MatrixDimension = 0
74     ReDim MatrixValues(MaxLength)
75     Do
76         met += 1
77         line = sr.ReadLine
78
79         If line = Nothing Then
80             Exit Do
81         Else
82             Application.DoEvents()
83             Me.LoadFileValuesWorker.ReportProgress(IIf((CurrentLoadPosition *
100 / MaxLength) > 100, 100, (CurrentLoadPosition * 100 / MaxLength)))
84             CurrentLoadPosition += line.Length + 2
85             If CurrentLoadPosition = MaxLength Then
86                 End If
87             End If
88             Select Case met
89                 Case Is > 6
90                 MatrixValues(met - 7) = IIf(Line3 = "RBStrofes", Val(line), Val
(line) * 100)
91                 MatrixDimension += 1
92                 Case 3
93                     If line.IndexOf("Control RPM") <> -1 Then
94                         Line3 = "RBStrofes"
95                     Else
96                         If line.IndexOf("PWM Cycle(%)" <> -1 Then
97                             Line3 = "RBGkazi"
98                         Else
99                             Line3 = "RBStrofes"
100                            Exit Do
101                            End If
102                        End If
103                    Case 4
104                        If line.IndexOf("Manually") <> -1 Then
105                            line4 = "RBManual"
106                        Else
107                            If line.IndexOf("With Function") <> -1 Then
108                                line4 = "RBFromFunction"
109                            Else
110                                line4 = "RBFromFile"
111                                Exit Do
112                            End If
113                        End If
114                    Case 5
115                        Line5 = line
116                End Select
117
118                If Me.LoadFileValuesWorker.CancellationPending = True Then
119                    e.Cancel = True
120                    Exit Do
121                End If
122                Loop Until line Is Nothing
123                sr.Close()
124                If MatrixDimension * TimeChanges > MyClassCode.MaximumTime Then
125                    MatrixDimension = 0
126                    ReDim MatrixValues(-1)
127                    MsgBox(Me.MyClassCode.GetStr(Me.WhichIsTheLang, 237), MsgBoxStyle.
Information, Me.MyClassCode.GetStr(Me.WhichIsTheLang, 109))
128                Else
129                    ReDim Preserve MatrixValues(MatrixDimension - 1)
130                End If
131            Catch Exp As Exception
132                'Do not set a message here
133                ReDim MatrixValues(-1)
134            End Try
135        End Sub
136
137        Public Sub SetProgressBarValue(ByVal Value As Integer)
138            Me.LoadFileProgressBar.Value = Value
139        End Sub
140
141        Private Sub BtCancelLoadFile_Click(ByVal sender As System.Object, ByVal e As System
.EventArgs) Handles BtCancelLoadFile.Click

```

```

142     If Not LoadFileValuesWorker.CancellationPending And LoadFileValuesWorker.IsBusy
143 = True Then
144         LoadFileValuesWorker.CancelAsync()
145     End If
146     ReDim MatrixValues(-1) ' To satisfy cancel with escape
147 End Sub
148 Private Sub LoadFileValuesWorker_DoWork(ByVal sender As System.Object, ByVal e As
149 System.ComponentModel.DoWorkEventArgs) Handles LoadFileValuesWorker.DoWork
150     Dim worker As BackgroundWorker = CType(sender, BackgroundWorker)
151     AdjustAccordingToConfFile(e.Argument, worker, e)
152 End Sub
153 Private Sub LoadFileValuesWorker_ProgressChanged(ByVal sender As System.Object,
154 ByVal e As System.ComponentModel.ProgressChangedEventArgs) Handles
155 LoadFileValuesWorker.ProgressChanged
156     Me.LoadFileProgressBar.Value = e.ProgressPercentage
157     If e.ProgressPercentage <= 1 Or (e.ProgressPercentage Mod 1) = 0 Then
158         If e.ProgressPercentage > PreviousPercentage Or e.ProgressPercentage = 0
159 Then
160             PreviousPercentage = e.ProgressPercentage
161             LblRemainingTime.Text = MyClassCode.GetStr(WhichIsTheLang, 310) & " " &
162 MyClassCode.EstimateRemainingTime(Now, e.ProgressPercentage)
163         End If
164     End If
165 End Sub
166 Private Sub LoadFileValuesWorker_RunWorkerCompleted(ByVal sender As System.Object,
167 ByVal e As System.ComponentModel.RunWorkerCompletedEventArgs) Handles
168 LoadFileValuesWorker.RunWorkerCompleted
169     If (e.Error IsNot Nothing) Then
170         ReDim Me.MatrixValues(-1) '-1
171         ConfSendDataFile = ""
172         Line3 = ""
173         line4 = ""
174         Line5 = ""
175     Else
176         If e.Cancelled Or sender.CancellationPending = True Then
177             ReDim Me.MatrixValues(-1) '-1
178             ConfSendDataFile = ""
179             Line3 = ""
180             line4 = ""
181             Line5 = ""
182         End If
183     End If
184     Me.Hide()
185 End Sub
186 Private Sub Loadfile()
187     LoadFileValuesWorker.RunWorkerAsync(ConfSendDataFile)
188 End Sub
189 Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
190 Handles Timer1.Tick
191     Me.BeginInvoke(New InvokeDelagate(AddressOf Loadfile))
192     Timer1.Stop()
193 End Sub
194 Private Sub LoadingDataForm_FormClosing(ByVal sender As System.Object, ByVal e As
195 System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing
196     If e.CloseReason = CloseReason.UserClosing Then
197         If LoadFileValuesWorker.IsBusy Then
198             LoadFileValuesWorker.CancelAsync()
199             ConfSendDataFile = ""
200             Line3 = ""
201             line4 = ""
202             Line5 = ""
203             ReDim MatrixValues(0)
204         End If
205     End If
206 End Sub
207 Private Sub TxtSaveData_Click(ByVal sender As Object, ByVal e As System.EventArgs)
208 Handles TxtSaveData.Click
209     BtCancelLoadFile.Focus()

```

```

206     End Sub
207
208     Private Sub TxtSaveData_Enter(ByVal sender As Object, ByVal e As System.EventArgs)
209         Handles TxtSaveData.Enter
210             BtCancelLoadFile.Focus()
211     End Sub
212
213     Private Sub RTBLoadFileDisplayText_Click(ByVal sender As Object, ByVal e As System.
214         EventArgs) Handles RTBLoadFileDisplayText.Click
215         BtCancelLoadFile.Focus()
216     End Sub
217
218     Private Sub RTBLoadFileDisplayText_Enter(ByVal sender As Object, ByVal e As System.
219         EventArgs) Handles RTBLoadFileDisplayText.Enter
220         BtCancelLoadFile.Focus()
221     End Sub
222 End Class

```

```

1 Imports System.IO
2 Imports System.Configuration
3
4 Module General
5     Public mComs(9) As Boolean
6 End Module
7
8 Public Class MyPtyxClass
9     Public MaximumTime As Integer = 108000000 ' 30 hours * 3600 secs * 1000 miliseccs
10    Public Literals(4, 350) As String
11    Public InitiatingConnectionwithPc As Boolean = True
12    Public Enum ThrustConnFormClose
13        Connection_Close = 0
14        Form_Close = 1
15        Close_False = 2
16    End Enum
17
18    Public VbClosingState As ThrustConnFormClose = ThrustConnFormClose.Close_False
19    Public Structure CloseState
20        Public Closed As Boolean
21        Public ClosedResetProperly As Boolean
22    End Structure
23    Public IARCardIsClosedProperly As CloseState
24    Public Const Nubmerof_ValidIncomingResetBytePackages As Int16 = 3
25    Public NumberIdofValidResetPackages As Int16 = 0
26    Public IARID() As Byte = {255}
27    '----- end intialazing connection -----
28
29    '-----
30
31    Const PCLKFREQ As Single = 3686400
32    Const Clockspeed As Single = PCLKFREQ / 4
33
34    Const Timereset As Single = 4660338 '((2 ^ 32) * 1000 / Clockspeed)
35
36    Public Structure SYSTEMTIME
37        Public wYear As Int16
38        Public wMonth As Int16
39        Public wDayOfWeek As Int16
40        Public wDay As Int16
41        Public wHour As Int16
42        Public wMinute As Int16
43        Public wSecond As Int16
44        Public wMilliseconds As Int16
45    End Structure
46    Public MyConnectionTime As SYSTEMTIME
47    Public Sub GetTime(ByRef st As SYSTEMTIME)
48        ' Call the native GetSystemTime method
49        ' with the defined structure.
50        'Dim st As New SYSTEMTIME
51        Dim mytime As System.DateTime
52        mytime = Now()
53        st.wHour = Hour(mytime)
54        st.wMinute = Minute(mytime)
55        st.wSecond = Second(mytime)
56
57    End Sub
58    Public MyStartTime, CurrentTime As System.DateTime
59    Public stTimeConnection As String = "00:00:00"
60    Public LastReceivedDataTime As Long = Now.Ticks 'initial value of creation myclass
61    Public msgTimeoutSerial As MessageBox
62    Public msgTimeoutOkPressed As System.Windows.Forms.DialogResult
63    Public messageaccepted As Boolean = True
64    Public EstimateTimeMat(1) As System.DateTime
65
66    Public Sub New()
67        InitializeLiterals()
68    End Sub
69
70    Public Function DeCaptulate(ByVal Sent1 As UInt32, ByVal Sent2 As UInt32, ByVal
71    Sent3 As UInt32, ByVal Sent4 As UInt32) As UInt32
72        Dim Result As UInt32
73        Dim ChangedSent1, ChangedSent2, ChangedSent3, ChangedSent4 As Long

```

```

74         ChangedSent1 = Sent1 << 24
75         ChangedSent2 = Sent2 << 16
76         ChangedSent3 = Sent3 << 8
77         ChangedSent4 = Sent4
78
79         Result = ChangedSent1 Or ChangedSent2 Or ChangedSent3 Or ChangedSent4
80
81         Return Result
82     End Function
83
84     Public Function Captulate(ByVal MyInput As UInt32) As Byte()
85         Dim MyBuffer(4) As Byte
86         MyBuffer(0) = (MyInput << 0) >> 24
87         MyBuffer(1) = (MyInput << 8) >> 24
88         MyBuffer(2) = (MyInput << 16) >> 24
89         MyBuffer(3) = (MyInput << 24) >> 24
90         Return MyBuffer
91     End Function
92
93     Public Function DeCaptulateTime(ByVal Sent1 As UInt32, ByVal Sent2 As UInt32,
ByVal Sent3 As UInt32, ByVal Sent4 As UInt32) As UInt32
94         Dim Result As UInt32
95         Dim ChangedSent1, ChangedSent2, ChangedSent3, ChangedSent4 As Long
96
97         ChangedSent2 = Sent2 << 16
98         ChangedSent3 = Sent3 << 8
99         ChangedSent4 = Sent4
100        Result = ChangedSent2 Or ChangedSent3 Or ChangedSent4
101
102        Result += Sent1 * Timereset
103
104        If Result < PreviousTime Then ' check if time is only rising and if not then
set time as the previous value of time
105            Throw New Exception("dsfgfsdgf")
106
107        End If
108
109        PreviousTime = Result
110        Return Result
111    End Function
112
113    Public Sub DecaptulateStatusConditions(ByVal sent1 As Byte, ByVal sent2 As Byte,
ByRef StatusIAR() As Byte, ByRef Errors() As Boolean)
114
115        StatusIAR(0) = sent1
116        StatusIAR(1) = sent2
117        Dim bit_access As New BitArray(StatusIAR)
118        For i As Int16 = 8 To 15 Step 1
119            Errors(i - 8) = bit_access(i) 'if 0 false, if 1 = true
120        Next
121    End Sub
122
123    Public Sub InitParametersFor_Connection()
124        InitiatingConnectionwithPc = True
125        PreviousTime = 0
126        Me.IARCardIsClosedProperly.ClosedResetProperly = True
127        Me.IARCardIsClosedProperly.Closed = False
128        Me.VbClosingState = ThrustConnFormClose.Close_False
129    End Sub
130    Public Function CheckIfCard_Closed_or_Reseted(ByVal Matrix() As Byte, ByVal
WhichLang As Integer) As Boolean
131        For i As Int16 = 0 To Matrix.Length - 2
132            If Matrix(i) = Matrix(i + 1) Then
133                If i = Matrix.Length - 2 Then
134                    Select Case Matrix(0)
135                        Case 255
136                            Me.IARCardIsClosedProperly.ClosedResetProperly = True
137                            Me.IARCardIsClosedProperly.Closed = True
138                            Return True
139                        Case Else
140                            Me.IARCardIsClosedProperly.ClosedResetProperly = False
141                            Me.IARCardIsClosedProperly.Closed = True
142                            If MsgBox(GetStr(WhichLang, 316), MsgBoxStyle.Information,
GetStr(WhichLang, 109)) = MsgBoxResult.Ok Then 'neo minima
143                                End If

```

```

144             Return True
145         End Select
146     End If
147     Else
148         Return False
149     End If
150 Next
151
152 End Function
153 Public Sub InitialSerialCom(ByRef port As System.IO.Ports.SerialPort, ByVal WhichLang As Integer)
154
155     Static Dim InitByteMat(-1) As Byte
156     ReDim Preserve InitByteMat(InitByteMat.Length)
157     InitByteMat(InitByteMat.Length - 1) = port.ReadByte
158     If InitByteMat.Length = 100 And CheckifMatVal_is_all_the_same(InitByteMat,
255) Then
159         port.Write(SecureTransmit(DeCaptulate(Asc("R"), Asc("E"), Asc("S"), Asc("E
"))), 0, 6)
160         InitiatingConnectionwithPc = False
161         ReDim InitByteMat(-1)
162     ElseIf InitByteMat.Length = 100 Then
163         Select Case CheckifMatVal_is_all_the_same(InitByteMat, 254)
164             Case True
165                 Static msgresult As MsgBoxResult
166                 If (msgresult = MsgBox(GetStr(WhichLang, 317), MsgBoxStyle.
Critical, GetStr(WhichLang, 109))) = MsgBoxResult.Ok Then
167                     port.DiscardInBuffer()
168                 Else
169                     End If
170             Case False
171                 port.Write(SecureTransmit(4294967293), 0, 6) ' reset card has not
been closed it stil transmitting data
172             End Select
173             ReDim InitByteMat(-1)
174         Else
175
176         End If
177         Refresh_Time_of_Recieved_Data_for_Timeout()
178     End Sub
179
180 Private Function CheckifMatVal_is_all_the_same(ByVal MAT() As Byte, ByVal
Checkbyte As Byte) As Boolean
181     Dim equal As Boolean = True
182     Dim i As Int16 = 0
183     Do
184         equal = IIf(MAT(i) = Checkbyte, True, False)
185         i += 1
186         If i = MAT.Length And equal = True Then
187             Return True
188         End If
189     Loop While (equal = True)
190     Return False
191 End Function
192 Public Function SecureTransmit(ByVal Myinput As UInt32) As Byte()
193     Dim MyBuffer(6) As Byte
194     MyBuffer(0) = 42 ' (*)
195     MyBuffer(1) = (Myinput << 0) >> 24
196     MyBuffer(2) = (Myinput << 8) >> 24
197     MyBuffer(3) = (Myinput << 16) >> 24
198     MyBuffer(4) = (Myinput << 24) >> 24
199     MyBuffer(5) = (MyBuffer(1) Xor MyBuffer(2) Xor MyBuffer(3) Xor MyBuffer(4))
200     Return MyBuffer
201 End Function
202
203 Public Function GetStr(ByVal WhichLang As Integer, ByVal WhichLiteral As Integer)
As String
204     Return Literals(WhichLang, WhichLiteral)
205 End Function
206
207 Private Sub InitializeLiterals()
208     Dim Version, DateVersion As String
209
210     Version = "4.0.30"
211     DateVersion = "5/03/2008 16:00"

```



```

212
213      ' Greek ----->
----->
214      Literals(0, 0) = "Καταχώριση Δεδομένων"
215      Literals(0, 1) = "Αποθήκευση των Εξερχόμενων Δεδομένων στο Αρχείο :"
216      Literals(0, 2) = "Αποθήκευση των Εισερχόμενων Δεδομένων στο Αρχείο :"
217      Literals(0, 3) = "Αποστολή Δεδομένων"
218      Literals(0, 4) = "Εισαγωγή Χαρακτηρών προς Αποστολή"
219      Literals(0, 5) = "Αποστολή Δεδομένων από Αρχείο"
220      Literals(0, 6) = "Εισερχόμενα Δεδομένα"
221      Literals(0, 7) = "Εξερχόμενα Δεδομένα"
222      Literals(0, 8) = "Αποστολή"
223      Literals(0, 9) = "&Αρχείο"
224      Literals(0, 10) = "&Σύνδεση..."
225      Literals(0, 11) = "&Αποσύνδεση"
226      Literals(0, 12) = "&Έξοδος"
227      Literals(0, 13) = "Ρυθμίσεις"
228      Literals(0, 14) = "Αλλαγή Γλώσσας"
229      Literals(0, 15) = "Ελληνικά"
230      Literals(0, 16) = "Αγγλικά"
231      Literals(0, 17) = "Όλες οι Παράμετροι"
232      Literals(0, 18) = "Προεπιλεγμένη Γλώσσα"
233      Literals(0, 19) = "Αποθήκευση Εξερχόμενων Δεδομένων"
234      Literals(0, 20) = "Αποθήκευση Εισερχόμενων Δεδομένων"
235      Literals(0, 21) = "Αποστολή Δεδομένων από Αρχείο"
236      Literals(0, 22) = "Παράμετροι Σύνδεσης"
237      Literals(0, 23) = "Αποθήκευση του Αρχείου Ρυθμίσεων"
238      Literals(0, 24) = "Επιλογή Αρχείου Ρυθμίσεων"
239      Literals(0, 25) = "Επιλογή Συγκεκριμένου Αρχείου"
240      Literals(0, 26) = "Αποεπιλογή Επιλεγμένου Αρχείου"
241      Literals(0, 27) = "Εκτυπώσεις"
242      Literals(0, 28) = "Εκτύπωση Εξερχόμενων Δεδομένων"
243      Literals(0, 29) = "Εκτύπωση Εισερχόμενων Δεδομένων"
244      Literals(0, 30) = "Εκτύπωση Δεδομένων που θα Σταλούν από Αρχείο"
245      Literals(0, 31) = "Βοήθεια"
246      Literals(0, 32) = "Εμφάνιση Βοήθειας"
247      Literals(0, 33) = "Έκδοση"
248      Literals(0, 34) = "Πτυχιακή Κωνσταντίνου Καρδάση"
249      Literals(0, 35) = "Αποσύνδεση"
250      Literals(0, 36) = "Έξοδος από την Εφαρμογή"
251      Literals(0, 37) = "Πτυχιακή Κωνσταντίνου Καρδάση"
252      Literals(0, 38) = "Καθαρισμός Εισερχόμενων Δεδομένων"
253      Literals(0, 39) = "Καθαρισμός Εξερχόμενων Δεδομένων"
254      Literals(0, 40) = "Εμφάνιση Δεδομένων"
255      Literals(0, 41) = "Εισερχόμενα Δεδομένα"
256      Literals(0, 42) = "Εξερχόμενα Δεδομένα"
257      Literals(0, 43) = "Δεν Έχει Δηλωθεί Έγκυρο Όνομα Αρχείου"
258      Literals(0, 44) = "Δεν Είναι Δυνατό να Βρεθεί το Αρχείο!"
259      Literals(0, 45) = "Επιλεγμένο Αρχείο Ρυθμίσεων:"
260      Literals(0, 46) = "Δεν Είναι Δυνατή η Εγγραφή Δεδομένων στη Σειριακή Θύρα!"
261      Literals(0, 47) = "Εφαρμογές"
262      Literals(0, 48) = "Θα Πρέπει να Δηλώσετε Έγκυρα Αρχεία Αποθήκευσης
Εισερχόμενων - Εξερχόμενων Δεδομένων Προκειμένου να Προχωρήσετε στη Σύνδεση."
263      Literals(0, 49) = "Εφαρμογές"
264      Literals(0, 50) = "Αποθήκευση Δεδομένων"
265      Literals(0, 51) = "Στροφές ανά Λεπτό"
266      Literals(0, 52) = "Κανόνας Ελέγχου"
267      Literals(0, 53) = "Χρονομετρητής"
268      Literals(0, 54) = "Αποθήκευση των Δεδομένων στο Αρχείο :"
269      Literals(0, 55) = "Εκκίνηση Καταγραφής"
270      Literals(0, 56) = "Λήξη Καταγραφής"
271      Literals(0, 57) = "Πρέπει να ορίσετε πρώτα ένα έγκυρο αρχείο στο οποίο θα
αποθηκευτούν τα δεδομένα!"
272      Literals(0, 58) = "Κλείσιμο Οθόνης"
273      Literals(0, 59) = "Κατάσταση Σύνδεσης: Δεν Έχει Ξεκινήσει η Παραλαβή των
Πακέτων"
274      Literals(0, 60) = "Κατάσταση Σύνδεσης: Έχει Ξεκινήσει η Παραλαβή των Πακέτων"
275      Literals(0, 61) = "Διάρκεια Παραλαβής Πακέτων (Λεπτά:Δευτ.)"
276      Literals(0, 62) = "Κατάσταση Σύνδεσης: Δεν Έχει Γίνει Σύνδεση"
277      Literals(0, 63) = "Η Παραλαβή Πακέτων Έχει Διακοπεί"
278      Literals(0, 64) = "Έχετε Αποσυνδεθεί"
279      Literals(0, 65) = "Έκδοση Προγράμματος :"
280      Literals(0, 66) = "Έκδοση " & Version
281      Literals(0, 67) = "Ημερομηνία Έκδοσης :"
282      Literals(0, 68) = "Δημιουργός :"

```

```

283     Literals(0, 69) = "Κωνσταντίνος Καρδάσης"
284     Literals(0, 70) = "Κλείσιμο"
285     Literals(0, 71) = "Έκδοση Προγράμματος"
286     Literals(0, 72) = "Σειριακή Θύρα"
287     Literals(0, 73) = "Ρυθμός Μετάδ."
288     Literals(0, 74) = "Ισοτιμία"
289     Literals(0, 75) = "Bit Δεδομένων"
290     Literals(0, 76) = "Stop Bits"
291     Literals(0, 77) = "Αποδοχή"
292     Literals(0, 78) = "Ακύρωση"
293     Literals(0, 79) = "Παράμετροι Σύνδεσης"
294     Literals(0, 80) = "Αποθήκευση του Αρχείου Ρυθμίσεων"
295     Literals(0, 81) = "Όνομα Αρχείου"
296     Literals(0, 82) = "Προεπιλεγμένη Γλώσσα"
297     Literals(0, 83) = "Επιλογή Προεπιλεγμένης Γλώσσας"
298     Literals(0, 84) = "Εμφάνιση Επιλογής Γλώσσας στην Εκκίνηση της Εφαρμογής"
299     Literals(0, 85) = "Αποθήκευση Εξερχόμενων Δεδομένων"
300     Literals(0, 86) = "Όνομα Αρχείου"
301     Literals(0, 87) = "Αποθήκευση Εισερχόμενων Δεδομένων"
302     Literals(0, 88) = "Όνομα Αρχείου"
303     Literals(0, 89) = "Αποστολή Δεδομένων από Αρχείο"
304     Literals(0, 90) = "Όνομα Αρχείου"
305     Literals(0, 91) = "Παράμετροι Σύνδεσης"
306     Literals(0, 92) = "Σειριακή Θύρα"
307     Literals(0, 93) = "Ρυθμός Μετάδοσης"
308     Literals(0, 94) = "Ισοτιμία"
309     Literals(0, 95) = "Data bits"
310     Literals(0, 96) = "Stop bits"
311     Literals(0, 97) = "Αποδοχή Αλλαγών"
312     Literals(0, 98) = "Επιστροφή στην Κύρια Οθόνη"
313     Literals(0, 99) = "Ορισμός Παραμέτρων"
314     Literals(0, 100) = "Αποθήκευση Δεδομένων της Εφαρμογής Ροπής Αδρανείας"
315     Literals(0, 101) = "Όνομα Αρχείου"
316     Literals(0, 102) = "Ελληνικά"
317     Literals(0, 103) = "Αγγλικά"
318     Literals(0, 104) = "Αποδοχή"
319     Literals(0, 105) = "Ακύρωση"
320     Literals(0, 106) = "Παρακαλώ, Επιλέξτε Γλώσσα"
321     Literals(0, 107) = "Το Αρχείο "
322     Literals(0, 108) = "δεν είναι Έγκυρο"
323     Literals(0, 109) = "Προσοχή!"
324     Literals(0, 110) = "Δεν Υπάρχουν Δεδομένα Προς Εκτύπωση!"
325     Literals(0, 111) = "Δεν είναι Δυνατή η Εμφάνιση της Βοήθειας!"
326     Literals(0, 112) = "Σχολή Ναυπηγών Μηχανολόγων Μηχανικών"
327     Literals(0, 113) = "ΔΕΔΟΜΕΝΑ"
328     Literals(0, 114) = "Μη Έγκυρη Σειριακή Θύρα!"
329     Literals(0, 115) = "Μη Έγκυρος Ρυθμός Μετάδοσης!"
330     Literals(0, 116) = "Μη Έγκυρη Ισοτιμία!"
331     Literals(0, 117) = "Μη Έγκυρο Πλήθος bits Δεδομένων!"
332     Literals(0, 118) = "Μη Έγκυρο Πλήθος stop bits!"
333     Literals(0, 119) = "Παρουσιάστηκε κάποιο πρόβλημα με το αρχείο ρυθμίσεων"
334     Literals(0, 120) = "Περιττή"
335     Literals(0, 121) = "Άρτια"
336     Literals(0, 122) = "Καμιά"
337     Literals(0, 123) = "Δεν είναι Εφικτό να Δημιουργηθεί το Αρχείο Ρυθμίσεων!"
338     Literals(0, 124) = "Παραλαβή Χρόνων"
339     Literals(0, 125) = "Παραλαβή Διαφορών Χρόνου"
340     Literals(0, 126) = DateVersion
341     Literals(0, 127) = "Το Αρχείο Ρυθμίσεων δεν Μπορούσε να Βρεθεί!"
342     Literals(0, 128) = "Δεν Έχετε Ορίσει το Αρχείο Ρυθμίσεων ή το Αρχείο αυτό δεν
Υπάρχει!"
343     Literals(0, 129) = "Δεν Έχετε Ορίσει το Αρχείο στο Οποίο θα Αποθηκεύονται τα
Δεδομένα που Στέλνεται ή το Αρχείο αυτό δεν Υπάρχει!"
344     Literals(0, 130) = "Δεν Έχετε Ορίσει το Αρχείο στο Οποίο θα Αποθηκεύονται τα
Δεδομένα που Λαμβάνεται ή το Αρχείο αυτό δεν Υπάρχει!"
345     Literals(0, 131) = "Δεν Έχετε Ορίσει το Αρχείο του Οποίου τα Περιεχόμενα
Μπορείται να Στείλνεται ή το Αρχείο αυτό δεν Υπάρχει!"
346     Literals(0, 132) = "Δεν Έχετε Ορίσει το Αρχείο στο Οποίο θα Αποθηκεύονται τα
Δεδομένα της Εφαρμογής Ροπής Αδρανείας!"
347     Literals(0, 133) = "Τα Αρχεία Αποθήκευσης Εισερχόμενων/Εξερχόμενων Δεδομένων
καθώς και Αποστολής Περιεχομένων Πρέπει να είναι Διαφορετικά Μεταξύ τους!"
348     Literals(0, 134) = "Δεν Μπορείτε να Χρησιμοποιήσετε το Αρχείο Ρυθμίσεων για
Αποθήκευση Δεδομένων!"
349     Literals(0, 135) = "Δεν Είναι Δυνατό να Αποθηκευτούν οι Αλλαγές!"
350     Literals(0, 136) = "Δεν Είναι Δυνατό να Χρησιμοποιηθεί το Αρχείο που Επιλέξατε"

```

```

!"
351 Literals(0, 137) = "Το Αρχείο που Επιλέξατε δεν είναι Έγκυρο!"
352 Literals(0, 138) = "Δεν Είναι Δυνατό να Ανοίξει η Σειριακή Θύρα!"
353 Literals(0, 139) = "Εμφάνιση Γραφημάτων"
354 Literals(0, 140) = "Γράφημα 1"
355 Literals(0, 141) = "Γράφημα 2"
356 Literals(0, 142) = "Απόκρυψη Γραφημάτων"
357 Literals(0, 143) = "Μεγένθυση Γραφημάτων"
358 Literals(0, 144) = "Σμίκρυνση Γραφημάτων"
359 Literals(0, 145) = "Σειριακή Επικοινωνία"
360 Literals(0, 146) = "Ροπή Αδρανείας"
361 Literals(0, 147) = "Κλίνη Ωσης"
362 Literals(0, 148) = "Αρχική Φόρμα"
363 Literals(0, 149) = "Κλίνη Ωσης και Ελέγχου Στροφών (RPM)"
364 Literals(0, 150) = "Σύνδεση"
365 Literals(0, 151) = "Κλείσιμο Θρόνης"
366 Literals(0, 152) = "Κατάσταση"
367 Literals(0, 153) = "Κανόνας Ελέγχου (%)"
368 Literals(0, 154) = "Έλεγχος Στροφών"
369 Literals(0, 155) = "Ιδιότητες Γραφημάτων"
370 Literals(0, 156) = "Παραμετροποίηση Γραφημάτων .NET"
371 Literals(0, 157) = "Παράμετροι 1ου Γραφήματος"
372 Literals(0, 158) = "Στροφές Ανα Λεπτό"
373 Literals(0, 159) = "Δυναμική Κυψέλη"
374 Literals(0, 160) = "Ένταση Ρεύματος"
375 Literals(0, 161) = "Χρόνος"
376 Literals(0, 162) = "Άμεση Διακοπή Λειτουργίας Κινητήρα"
377 Literals(0, 163) = "Άμεση Διακοπή Λειτουργίας;"
378 Literals(0, 164) = "Αποθήκευση Δεδομένων της Εφαρμογής Κλίνη Ωσης"
379 Literals(0, 165) = "Δεν Έχετε Ορίσει το Αρχείο στο Οποίο θα Αποθηκεύονται τα
    Δεδομένα της Εφαρμογής Κλίνη Ωσης!"
380 Literals(0, 166) = "Διαθέσιμα Αρχεία Ρυθμίσεων"
381 Literals(0, 167) = "Επιλογή"
382 Literals(0, 168) = "Σημείο Αναφοράς Στροφών"
383 Literals(0, 169) = "Τάση"
384 Literals(0, 170) = "Εμφάνιση Γραφημάτων μέσω .NET"
385 Literals(0, 171) = "Εμφάνιση Γραφημάτων μέσω Excel"
386 Literals(0, 172) = "Παράμετροι 2ου Γραφήματος"
387 Literals(0, 173) = "Παραμετροποίηση Γραφημάτων Excel"
388 Literals(0, 174) = "Πλήθος Σημείων"
389 Literals(0, 175) = "Οριζ. Γραμμές"
390 Literals(0, 176) = "Κάθ. Γραμμές"
391 Literals(0, 177) = "Οριζ. Υποδιαίρεση"
392 Literals(0, 178) = "PWM"
393 Literals(0, 179) = "Δυναμοκυψέλη"
394 Literals(0, 180) = "Τάση"
395 Literals(0, 181) = "Ένταση Ρεύματος"
396 Literals(0, 182) = "Διακεκομμένες Γραμμές"
397 Literals(0, 183) = "Συνεχείς Γραμμές"
398 Literals(0, 184) = "Τύπος Γραμμών"
399 Literals(0, 185) = "Προβολή παρελθόντος"
400 Literals(0, 186) = "Παρόν"
401 Literals(0, 187) = "Σ.Α.Λ"
402 Literals(0, 188) = "Επίθ. Σ.Α.Λ"
403 Literals(0, 189) = "PWM"
404 Literals(0, 190) = "Επίθ. PWM"
405 Literals(0, 191) = "Σύγχρονη κίνηση γραφημάτων"
406 Literals(0, 192) = "Ιδιότητες Διαγράμματος 1"
407 Literals(0, 193) = "Ιδιότητες Διαγράμματος 2"
408 Literals(0, 194) = "Εμφάνιση Γραμμής:"
409 Literals(0, 195) = "Αριθμός Σημείων"
410 Literals(0, 196) = "Απόκρυψη Πλαισίου"
411 Literals(0, 197) = "Δεν Έχετε Ορίσει το Αρχείο από το Οποίο θα Στέλλονται οι
    Τιμές Γκαζιού ή Στροφών!"
412 Literals(0, 198) = "Δημιουργία Αρχείου Αποστολής Τιμών Αναφοράς"
413 Literals(0, 199) = "Επιλέξτε Κατάσταση"
414 Literals(0, 200) = "Τρόπος Δημιουργίας Αρχείου"
415 Literals(0, 201) = "Χειρωνακτικά"
416 Literals(0, 202) = "Από Συνάρτηση"
417 Literals(0, 203) = "Εισαγωγή από Αρχείο Κειμένου(*.txt)"
418 Literals(0, 204) = "Εγγραφή Τιμών Στροφών στο Αρχείο"
419 Literals(0, 205) = "Εγγραφή Τιμών Κανόνα Ελέγχου (%) στο Αρχείο"
420 Literals(0, 206) = "Τιμή "
421 Literals(0, 207) = "Επανάληψη Τιμών"
422 Literals(0, 208) = "Πλήθος Επαναλήψεων"

```

```

423     Literals(0, 209) = "Εμφάνιση Τιμών σε Γράφημα"
424     Literals(0, 210) = "Επιλογή Συνάρτησης"
425     Literals(0, 211) = "Αρχική Τιμή Χρόνου:"
426     Literals(0, 212) = "Τελική Τιμή Χρόνου:"
427     Literals(0, 213) = "Δημιουργία Συνάρτησης:"
428     Literals(0, 214) = "Έλεγχος Δωθείσας Συνάρτησης"
429     Literals(0, 215) = "Επιλεγμένη Συνάρτηση"
430     Literals(0, 216) = "Εμφάνιση Προηγούμενων 50 Τιμών"
431     Literals(0, 217) = "Εμφάνιση Επόμενων 50 Τιμών ή Εισαγωγή Καινούργιας Σελίδας"
432     Literals(0, 218) = "Εμφάνιση Πρώτων 50 Τιμών"
433     Literals(0, 219) = "Εμφάνιση Τελευταίων 50 Τιμών"
434     Literals(0, 220) = "Πρώτη Τιμή Επανάληψης"
435     Literals(0, 221) = "Τελευταία Τιμή Επανάληψης"
436     Literals(0, 222) = "Εκτέλεση Επανάληψης"
437     Literals(0, 223) = "Υπήρξε Σφάλμα στην Είσοδο Δεδομένων, όπως στο Κελί "
438     Literals(0, 224) = "Αποτυχία Αποθήκευσης, Παρουσιάστηκε Κάποιο Σφάλμα κατά την
Αποθήκευση των Δεδομένων στο Αρχείο "
439     Literals(0, 225) = "Χρονικό Διάστημα "
440     Literals(0, 226) = "Πήγαινε στη Σελίδα"
441     Literals(0, 227) = "Σελίδα "
442     Literals(0, 228) = " από "
443     Literals(0, 229) = "Το Επιτρεπόμενο Σύνολο Τιμών Είναι"
444     Literals(0, 230) = "Εναρξη Επανάληψης από την Τιμή"
445     Literals(0, 231) = "Οι Τιμές που Έχετε Δώσει δεν Είναι Έγκυρες"
446     Literals(0, 232) = "Επιλογή Αρχείου"
447     Literals(0, 233) = "Γενικές Οδηγίες"
448     Literals(0, 234) = "Το αρχείο που θα ορίσετε θα πρέπει να περιέχει μόνο
αριθμητικές τιμές, να μην περιέχει κενές γραμμές και " & _
449     "γενικότερα κενούς χαρακτήρες. Επίπλέον αν έχετε επιλέξει
Έλεγχο Στροφών οι τιμές θα πρέπει να είναι στο " & _
450     "σύνολο τιμών [0 - 20000] ενώ αν έχετε επιλέξει Κανόνα
Ελέγχου στο [0 - 100.0]." & vbCrLf & _
451     "Μόλις επιλέξετε το αρχείο που επιθυμείτε θα πρέπει να
πατήσετε το κουμπί έλεγχος αρχείου, ώστε η εφαρμογή να " & _
452     "ελέγξει τη συμβατότητα του αρχείου. Αν το αρχείο είναι
συμβατό με την εφαρμογή τότε θα ενεργοποιηθεί το κουμπί " & _
453     "Φόρτωμα Αρχείου. Μόλις το επιλέξετε τα δεδομένα του
αρχείου θα περάσουν στην εφαρμογή," & _
454     " σβήνοντας παράλληλα ό,τι δεδομένα έχετε περάσει χωρίς να
αποθηκεύσετε."
455     Literals(0, 235) = "Περιεχόμενα Αρχείου"
456     Literals(0, 236) = "Εφαρμογή Αρχείου" !"φόρτωμα Αρχείου"
457     Literals(0, 237) = "Παρουσιάστηκε Κάποιο Σφάλμα κατά τη φόρτωση του Αρχείου!"
458     Literals(0, 238) = "Πλήθος Τιμών στο Αρχείο:"
459     Literals(0, 239) = "Έλεγχος Αρχείου"
460     Literals(0, 241) = "Γνωστές Συναρτήσεις"
461     Literals(0, 242) = "Επιλεγμένες Συναρτήσεις και Χρονικοί Περίοδοι"
462     Literals(0, 243) = "Καταχώριση Συνάρτησης"
463     Literals(0, 244) = "Επανεκκίνηση"
464     Literals(0, 245) = "Αποδεκτά Σύμβολα"
465     Literals(0, 246) = "Η Συνάρτηση που Γράψατε δεν Είναι Έγκυρη. Ενημερωθείτε για
το ποιά Σύμβολα Είναι Αποδεκτά μέσα από την Καρτέλα "Αποδεκτά Σύμβολα"."
466     Literals(0, 247) = "Εκτέλεση Συναρτήσεων"
467     Literals(0, 248) = "Οι Τιμές των Χρόνων που Δώσατε Δεν Είναι Έγκυρες!"
468     Literals(0, 249) = "Έχετε ήδη Ορίσει Άλλη Συνάρτηση σε Αυτό το Χρονικό
Διάστημα!"
469     Literals(0, 250) = "Έχετε Τροποήσει την Επιλεγμένη Συνάρτηση!"
470     Literals(0, 251) = "Εμφανίστηκε Κάποιο Σφάλμα κατά την Εκτέλεση των
Συναρτήσεων!"
471     Literals(0, 252) = "Χρονικές Περίοδοι"
472     Literals(0, 253) = "Συνάρτηση"
473     Literals(0, 254) = "Συναρ/εις"
474     Literals(0, 255) = "Σύμβολα"
475     Literals(0, 256) = "Μεταβλ/ές"
476     Literals(0, 257) = "Δεν Έχετε Ορίσει Κάποια Συνάρτηση!"
477     Literals(0, 258) = "Οι Τιμές που Προκύπτουν από τις Συναρτήσεις Είναι Εκτός
Ορίων!"
478     Literals(0, 259) = "Ιταλικά"
479     Literals(0, 260) = "Ισπανικά"
480     Literals(0, 261) = "Γαλλικά"
481     Literals(0, 262) = " Παρακαλώ περιμένετε τη φόρτωση των δεδομένων." & vbCrLf &
" Αν δεν επιθυμείτε τη φόρτωση των δεδομένων μπορείτε να πατήσετε το πλήκτρο "
"Ακύρωση Εργασίας"."
482     Literals(0, 263) = "Πρόσδος Εργασίας"
483     Literals(0, 264) = "Ακύρωση Εργασίας"

```

```

484     Literals(0, 265) = "Εκκίνηση"
485     Literals(0, 266) = "Λήξη"
486     Literals(0, 267) = "Παύση"
487     Literals(0, 268) = "Συνέχεια"
488     Literals(0, 269) = "Η Μέγιστη Τιμή Χρόνου που μπορείτε να Δώσετε είναι "
489     ' display file sector
490     Literals(0, 270) = "Κάθετες ανα ms"
491     Literals(0, 271) = "Εμφάνιση Σημείων"
492     Literals(0, 272) = "Μη Εμφάνιση Σημείων"
493     Literals(0, 273) = "Εμφάνιση μέσω των τιμών ανά:"
494     Literals(0, 274) = "Εμφανιζόμενα σημεία"
495     Literals(0, 275) = "Επιστροφή στην διαχείριση αρχείου"
496     Literals(0, 276) = "Ιδιότητες Τιμών & αλλαγή"
497     Literals(0, 277) = "Επιλογή & μεταβολή τιμής"
498     Literals(0, 278) = "Επιλεγμένη τιμή :"
499     Literals(0, 279) = "Αλλαγή τιμής :"
500     Literals(0, 280) = "Τιμή :"
501     Literals(0, 281) = "Έξοδος χωρίς αποθήκευση αλλαγών"
502     Literals(0, 282) = "Αποθήκευση αλλαγών και Έξοδος"
503     Literals(0, 283) = "Επιλογή όλων των αρχικών τιμών"
504     Literals(0, 284) = "Επιλογές εξόδου"
505     Literals(0, 285) = "Γραφική απεικόνιση αρχείου"
506     Literals(0, 286) = "Μη έγκυρο αρχικό αρχείο "
507     Literals(0, 287) = "Μη έγκυρο αρχείο"
508     Literals(0, 288) = "Μη αλλαγή"
509     Literals(0, 289) = "Επέλεξε εμφάνιση τιμών μικρότερο του : 2"
510     Literals(0, 290) = "Δεν έχουν πραγματοποιηθεί αλλαγές"
511     Literals(0, 291) = "Αλλαγές έχουν πραγματοποιηθεί,Επιλέξτε Yes για αποστολή
των τιμών στο αρχείο," & Chr(13) & "Επιλέξτε No για απόρριψη των αλλαγών και χρήση
των αρχικών τιμών," & Chr(13) & "Επιλέξτε Cancel για επαναφορά στη φάσμα γραφικών"
"
512     Literals(0, 292) = "Αρχική τιμή"
513     Literals(0, 293) = "Επιλογή όλων"
514     Literals(0, 294) = "Ο τύπος του αρχείου που επιλέξατε δεν είναι έγκυρος" & Chr(
(13) & "Επιλέξτε αρχείο τύπου *.txt"
515     Literals(0, 295) = "Λάθος επιλογή Αρχείου"
516     Literals(0, 296) = "Λάθος κατά την ανάγνωση από αρχείο"
517     Literals(0, 297) = "Τέλος αρχείου "
518     Literals(0, 298) = "Επιλογή αρχείου αποστολής"
519     Literals(0, 299) = "Αποστολή από αρχείο Σ/Α"
520     Literals(0, 300) = "Αποστολή Σ/Α από Φόρμα "
521     Literals(0, 301) = "Περίοδος ανανέωσης Γραφημάτων "
522     Literals(0, 302) = "Κάθε νέα τιμή"
523     Literals(0, 303) = "Περίοδος Ανανέωσης Βασικών Γραφικών : "
524     Literals(0, 304) = "Περίοδος Ανανέωσης Γραφικών : "
525     Literals(0, 305) = "Ανανέωση Γραφικών "
526     Literals(0, 306) = "Το αρχείο δεν είναι κατάλληλα διαμορφωμένο"
527     Literals(0, 307) = "Υπάρχει επιλεγμένο ήδη αρχείο επιλέξτε : " & Chr(13) & "Ok
για επαναποστολή του ιδίου " & Chr(13) & "H Cancel για να επιλέξεται νέο"
528     Literals(0, 308) = "Έχετε ξεπεράσει το Μέγιστο Χρονικό Διάστημα, Δεν Μπορείτε
να Εισάγετε Νέα Σελίδα!"
529     Literals(0, 309) = "Δεν Μπορεί να Εκτελεστεί η Επανάληψη Γιατί Θα Ξεπεράσετε
το Μέγιστο Χρονικό Διάστημα!"
530     Literals(0, 310) = "Η Εργασία θα Ολοκληρωθεί Περίπου σε"
531     Literals(0, 311) = "Γραφήματα"
532     Literals(0, 312) = "Δεν Μπορείτε να Τροποποιήσετε το Βασικό Αρχείο Ρυθμίσεων!"
533     Literals(0, 313) = "Επιλογή Προκαθορισμένου Γραφήματος"
534     Literals(0, 314) = ".NET Γραφήματα"
535     Literals(0, 315) = "Excel Γραφήματα"
536     'thrust additional literals
537     Literals(0, 316) = "Η κάρτα έχει κλείσει ασφαλώς, αλλά ένα λάθος έχει
δημιουργηθεί " & Chr(13) & "στις διαδικασίες τερματισμού. Επανεκκινήστε την κάρτα
για νέα σύνδεση."
538     Literals(0, 317) = "Οι παράμετροι εκκίνησης της κάρτας δεν έχουν τις
αναμενόμενες τιμές .Επανεκκινήστε την κάρτα ."
539     Literals(0, 318) = "Κατεστραμμένο αρχείο ή οι τιμές του δεν είναι μέσα στα
αποδεκτά όρια ."
540     Literals(0, 319) = "Θέλετε να επανεκκινήσει τον κινητήρα ;"
541     Literals(0, 320) = "Κλείστε και εκκινήστε την σειριακή επικοινωνία . Η κάρτα
έχει τερματίσει ."
542     Literals(0, 321) = "Εκκίνηση Κινητήρα"
543     Literals(0, 322) = "Μη παραλαβή δεδομένων εντός χρονικού ορίου"
544     Literals(0, 323) = "Προσοχή ,πρίν συνεχίσει επιλέξεται αρχείο αποθήκευσης ."
& ChrW(10) & ChrW(13) & " Οποιαδήποτε εργασία έως την επιλογή του αρχείου θα χαθεί"
.

```

```

545     Literals(0, 324) = "Αποστέλονται Σ/Α από Αρχείο"
546     ' English -----
-----
547     Literals(1, 0) = "Save Data"
548     Literals(1, 1) = "Save Output Data in File :'"
549     Literals(1, 2) = "Save Input Data in File :'"
550     Literals(1, 3) = "Send Data"
551     Literals(1, 4) = "Type the characters that you wish to send"
552     Literals(1, 5) = "Send the characters that are included in file:"
553     Literals(1, 6) = "Incomming Data"
554     Literals(1, 7) = "Outgoing Data"
555     Literals(1, 8) = "&Send"
556     Literals(1, 9) = "&File"
557     Literals(1, 10) = "&Connect..."
558     Literals(1, 11) = "&Disconnect"
559     Literals(1, 12) = "&Exit"
560     Literals(1, 13) = "Configurations"
561     Literals(1, 14) = "Change Language"
562     Literals(1, 15) = "Greek"
563     Literals(1, 16) = "English"
564     Literals(1, 17) = "All Parameters"
565     Literals(1, 18) = "Default Language"
566     Literals(1, 19) = "Store Outgoing Data"
567     Literals(1, 20) = "Store Incoming Data"
568     Literals(1, 21) = "Send Data from File"
569     Literals(1, 22) = "Connection Parameters"
570     Literals(1, 23) = "Save Configuration File"
571     Literals(1, 24) = "Configuration File"
572     Literals(1, 25) = "Select Configuration File"
573     Literals(1, 26) = "DeSelect Configuration File"
574     Literals(1, 27) = "Print"
575     Literals(1, 28) = "Print Outgoing Data"
576     Literals(1, 29) = "Print Incoming Data"
577     Literals(1, 30) = "Print Data That Will or Were Send By File"
578     Literals(1, 31) = "Help"
579     Literals(1, 32) = "Show Help"
580     Literals(1, 33) = "Version"
581     Literals(1, 34) = "Konstantino's Kardasis Diplo"
582     Literals(1, 35) = "Disconnect"
583     Literals(1, 36) = "Exit the application"
584     Literals(1, 37) = "Konstantino's Kardasis Diplo"
585     Literals(1, 38) = "Clear Input Data"
586     Literals(1, 39) = "Clear Output Data"
587     Literals(1, 40) = "Data Preview"
588     Literals(1, 41) = "Incoming Data"
589     Literals(1, 42) = "Outgoing Data"
590     Literals(1, 43) = "The Filename you Defined was not Valid"
591     Literals(1, 44) = "The File Could not be Found!"
592     Literals(1, 45) = "Selected Configurations File:"
593     Literals(1, 46) = "Unable to Write to COM Port!"
594     Literals(1, 47) = "Applications"
595     Literals(1, 48) = "You Must Define Valid Files for Saving Input - Output Data
In Order to Proceed In Connection."
596     Literals(1, 49) = "Applications"
597     Literals(1, 50) = "Save Data"
598     Literals(1, 51) = "Revolutions per Minute"
599     Literals(1, 52) = "Control Regulation"
600     Literals(1, 53) = "Time"
601     Literals(1, 54) = "Save Data in file :'"
602     Literals(1, 55) = "Start"
603     Literals(1, 56) = "Stop"
604     Literals(1, 57) = "You Must First Define a Valid File Where The Data Will be
Saved!"
605     Literals(1, 58) = "Close"
606     Literals(1, 59) = "Connection Status: Receiving Data Has Not Been Started"
607     Literals(1, 60) = "Connection Status: Receiving Data Has Been Started"
608     Literals(1, 61) = "Receiving Data Duration (Mins:Secs)"
609     Literals(1, 62) = "Connection Status: Connection Has Not Been Established"
610     Literals(1, 63) = "Receiving Data Has Been Stopped"
611     Literals(1, 64) = "You Have Been Disconnected"
612     Literals(1, 65) = "Version :'"
613     Literals(1, 66) = "Ver " & Version
614     Literals(1, 67) = "Release Date :'"
615     Literals(1, 68) = "Created By :'"
616     Literals(1, 69) = "Konstantinos Kardasis"

```

```

617     Literals(1, 70) = "Ok"
618     Literals(1, 71) = "Version"
619     Literals(1, 72) = "COM Port"
620     Literals(1, 73) = "Baud Rate"
621     Literals(1, 74) = "Parity"
622     Literals(1, 75) = "Data Bits"
623     Literals(1, 76) = "Stop Bits"
624     Literals(1, 77) = "OK"
625     Literals(1, 78) = "Cancel"
626     Literals(1, 79) = "Connection Parameters"
627     Literals(1, 80) = "Save Configurations File"
628     Literals(1, 81) = "File Name"
629     Literals(1, 82) = "Default Language"
630     Literals(1, 83) = "Define Default Language"
631     Literals(1, 84) = "Show Define Language on Startup"
632     Literals(1, 85) = "Save Output Data"
633     Literals(1, 86) = "File Name"
634     Literals(1, 87) = "Save Input Data"
635     Literals(1, 88) = "File Name"
636     Literals(1, 89) = "Send Data from File"
637     Literals(1, 90) = "File Name"
638     Literals(1, 91) = "Connection Parameters"
639     Literals(1, 92) = "COM Port"
640     Literals(1, 93) = "Baud rate"
641     Literals(1, 94) = "Parity"
642     Literals(1, 95) = "Data bits"
643     Literals(1, 96) = "Stop bits"
644     Literals(1, 97) = "Accept Changes"
645     Literals(1, 98) = "Back to Main Window"
646     Literals(1, 99) = "Define Parameters"
647     Literals(1, 100) = "Save Data for Application Torque Inertia" 'Inactivity =
inertia ?
648     Literals(1, 101) = "File Name"
649     Literals(1, 102) = "Greek"
650     Literals(1, 103) = "English"
651     Literals(1, 104) = "Ok"
652     Literals(1, 105) = "Cancel"
653     Literals(1, 106) = "Please, Select Language"
654     Literals(1, 107) = "The File "
655     Literals(1, 108) = "is not Valid"
656     Literals(1, 109) = "Attention!"
657     Literals(1, 110) = "There Are no Data to Print!"
658     Literals(1, 111) = "Could not Show Help File!"
659     Literals(1, 112) = "Naval Architecture & Marine Engineering"
660     Literals(1, 113) = "DATA"
661     Literals(1, 114) = "Invalid Com Port!"
662     Literals(1, 115) = "Invalid Baud Rate!"
663     Literals(1, 116) = "Invalid Parity!"
664     Literals(1, 117) = "Invalid Data Bits!"
665     Literals(1, 118) = "Invalid Stop Bits!"
666     Literals(1, 119) = "An Error About the Configurations File Occured!"
667     Literals(1, 120) = "Odd"
668     Literals(1, 121) = "Even"
669     Literals(1, 122) = "None"
670     Literals(1, 123) = "Couldn't Create the Configurations File!"
671     Literals(1, 124) = "Receive Times"
672     Literals(1, 125) = "Receive Dt"
673     Literals(1, 126) = DateVersion
674     Literals(1, 127) = "The configurations file could not be found!"
675     Literals(1, 128) = "You Have not Defined the Configurations File or it Does
not Exist!"
676     Literals(1, 129) = "You Have not Defined the File Where the Sent Data Will be
Saved or it does not Exist!"
677     Literals(1, 130) = "You Have not Defined the File Where the Received Data will
be Saved or it does not Exist!"
678     Literals(1, 131) = "You Have not Defined the File Whose Contents you can Send
or it does not Exist!"
679     Literals(1, 132) = "You Have not Defined the File Where the Data of the
Application Torque Inertia Will be Saved!"
680     Literals(1, 133) = "The Files That you Save Input/Output Data or Whose Data
are to be Send Should not be the Same!"
681     Literals(1, 134) = "You Cannot Use the Configuration's File for Saving Data!"
682     Literals(1, 135) = "Couldn't Save Changes!"
683     Literals(1, 136) = "You Cannot Use The Specific File!"
684     Literals(1, 137) = "The Selected File is not Valid!"

```

```

685     Literals(1, 138) = "Error! Could not Open Com Port!"
686     Literals(1, 139) = "Show Graphs"
687     Literals(1, 140) = "Graph 1"
688     Literals(1, 141) = "Graph 2"
689     Literals(1, 142) = "Hide Graphs"
690     Literals(1, 143) = "Maximize Graphs"
691     Literals(1, 144) = "Minimize Graphs"
692     Literals(1, 145) = "Serial Communication"
693     Literals(1, 146) = "Torque Inertia"
694     Literals(1, 147) = "Water Thrust and Rpm Control"
695     Literals(1, 148) = "Initial Form"
696     Literals(1, 149) = "Water Thrust and Rpm Control"
697     Literals(1, 150) = "Connection"
698     Literals(1, 151) = "Close Form"
699     Literals(1, 152) = "Status"
700     Literals(1, 153) = "PWM Cycle (%)"
701     Literals(1, 154) = "Control RPM"
702     Literals(1, 155) = "Graphs Properties"
703     Literals(1, 156) = "Parametarize .NET Gaphs"
704     Literals(1, 157) = "First Graph's Parameters"
705     Literals(1, 158) = "RPM"
706     Literals(1, 159) = "LoadCell (Kgr)"
707     Literals(1, 160) = "Amperes (A)"
708     Literals(1, 161) = "Time"
709     Literals(1, 162) = "Emergency Stop"
710     Literals(1, 163) = "Immediately Stop Process?"
711     Literals(1, 164) = "Save Data for Application Water Thrust and Rpm Control"
712     Literals(1, 165) = "You Have not Defined the File Where the Data of the
Application Water Thrust and Rpm Control Will be Saved!"
713     Literals(1, 166) = "Available Configurations Files"
714     Literals(1, 167) = "Select"
715     Literals(1, 168) = "Set Point RPM"
716     Literals(1, 169) = "Voltage"
717     Literals(1, 170) = "Show Graphs with .NET"
718     Literals(1, 171) = "Show Graphs with Excel"
719     Literals(1, 172) = "Second Graph's Parameters"
720     Literals(1, 173) = "Parametarize Excel Gaphs"
721     Literals(1, 174) = "Num Of Points"
722     Literals(1, 175) = "Horizontal Lines"
723     Literals(1, 176) = "Vertical Lines"
724     Literals(1, 177) = "Horizontal Div"
725     Literals(1, 178) = "PWM Load"
726     Literals(1, 179) = "LoadCell"
727     Literals(1, 180) = "Voltage"
728     Literals(1, 181) = "Amperes"
729     Literals(1, 182) = "Dashed Lines"
730     Literals(1, 183) = "Solid Lines"
731     Literals(1, 184) = "Type of Lines"
732     Literals(1, 185) = "Show Past"
733     Literals(1, 186) = "Now"
734     Literals(1, 187) = "R.P.M"
735     Literals(1, 188) = "Set R.P.M"
736     Literals(1, 189) = "PWM"
737     Literals(1, 190) = "set PWM"
738     Literals(1, 191) = "Move Synchronous Both Graphs"
739     Literals(1, 192) = "Chart 1 Properties"
740     Literals(1, 193) = "Chart 2 Properties"
741     Literals(1, 194) = "Select Line :"
742     Literals(1, 195) = "Chart Points"
743     Literals(1, 196) = "Hide Panel"
744     Literals(1, 197) = "You Have Not Defined the File from where the Set Point RPM
/PWM Cycle(%) will be Send!"
745     Literals(1, 198) = "Create File to Send Set Points"
746     Literals(1, 199) = "Select Mode"
747     Literals(1, 200) = "Method for File Creation"
748     Literals(1, 201) = "Manually"
749     Literals(1, 202) = "From Function"
750     Literals(1, 203) = "Import from Text File (*.txt)"
751     Literals(1, 204) = "Write Values for RPM:"
752     Literals(1, 205) = "Write Values for PWM Cycle (%):"
753     Literals(1, 206) = "Value "
754     Literals(1, 207) = "Repeat Values"
755     Literals(1, 208) = "Repeat Times:"
756     Literals(1, 209) = "Show Values in Graph"
757     Literals(1, 210) = "Select Function"

```



```

758     Literals(1, 211) = "Initial Time Value:"
759     Literals(1, 212) = "Final Time Value:"
760     Literals(1, 213) = "Create Function:"
761     Literals(1, 214) = "Check Given Function"
762     Literals(1, 215) = "Selected Function"
763     Literals(1, 216) = "Show Previous 50 Values"
764     Literals(1, 217) = "Show Next 50 Values or Add New Page"
765     Literals(1, 218) = "Show First 50 Values"
766     Literals(1, 219) = "Show Last 50 Values"
767     Literals(1, 220) = "First Repeat Value"
768     Literals(1, 221) = "Last Repeat Value"
769     Literals(1, 222) = "Execute Repeat"
770     Literals(1, 223) = "There was an Error in the Input Data, Such as in Cell  "
771     Literals(1, 224) = "Operation Failed, There was an Error While Saving Data in  ⚡
File "
772     Literals(1, 225) = "Time Period "
773     Literals(1, 226) = "Goto Page"
774     Literals(1, 227) = "Page "
775     Literals(1, 228) = " from "
776     Literals(1, 229) = "The Allowed Values are"
777     Literals(1, 230) = "Start Repeat from Value"
778     Literals(1, 231) = "The Values you have Given are Not Valid"
779     Literals(1, 232) = "Select File"
780     Literals(1, 233) = "General Instructions"
781     Literals(1, 234) = "The file that you will define must have only arithmetic  ⚡
values, no empty lines and generally no spaces as well " & _
782     "as if you have chosen Control RPM the values included  ⚡
must be in region [0 - 20000] whereas if you have chosen " & _
783     "PWM Cycle (%) values must be in region [0 - 100.0]." &  ⚡
vbCrLf & _
784     " As soon as you choose the file that you want you must  ⚡
click on the Check File button so as the " & _
785     "application will check the file's compatibility. If the  ⚡
file is compatible, then the Load File button will " & _
786     "be enabled. If you click on it the file's data will pass  ⚡
to the application deleting all previously " & _
787     "inserted but not saved data."
788     Literals(1, 235) = "File Data"
789     Literals(1, 236) = "Apply File" '"Load File"
790     Literals(1, 237) = "There was an Error While Loading File!"
791     Literals(1, 238) = "Total Number of Records in File:"
792     Literals(1, 239) = "Check File"
793     Literals(1, 240) = "You Must First Check the File"
794     Literals(1, 241) = "Known Functions"
795     Literals(1, 242) = "Selected Functions and Time Periods"
796     Literals(1, 243) = "Register Function"
797     Literals(1, 244) = "Reset"
798     Literals(1, 245) = "Acceptable Symbols"
799     Literals(1, 246) = "The Function you Typed is Not Valid. Check Acceptable  ⚡
Symbols Tab to See Which Symbols are Acceptable. "
800     Literals(1, 247) = "Execute Functions"
801     Literals(1, 248) = "The Time Values you Provided are Not Valid!"
802     Literals(1, 249) = "You have Already Given Another Function in this Period!"
803     Literals(1, 250) = "You have Changed Selected Function!"
804     Literals(1, 251) = "There was an Error While Running Validated Functions!"
805     Literals(1, 252) = "Time Periods"
806     Literals(1, 253) = "Function"
807     Literals(1, 254) = "Functions"
808     Literals(1, 255) = "Symbols"
809     Literals(1, 256) = "Variables"
810     Literals(1, 257) = "You Have Not Defined a Function!"
811     Literals(1, 258) = "The Values Coming From the Functions are out of Limits!"
812     Literals(1, 259) = "Italians"
813     Literals(1, 260) = "Spanish"
814     Literals(1, 261) = "France"
815     Literals(1, 262) = " Please wait to load data." & vbCrLf & " If you don't wish  ⚡
to load data you may click on button ""Cancel Task""."
816     Literals(1, 263) = "Task Progress"
817     Literals(1, 264) = "Cancel Task"
818     Literals(1, 265) = "Start"
819     Literals(1, 266) = "Stop"
820     Literals(1, 267) = "Pause"
821     Literals(1, 268) = "Resume"
822     Literals(1, 269) = "The Maximum Time Value you can Declare is "
823     ' display file sector

```

```

824     Literals(1, 270) = "Vertical per ms"
825     Literals(1, 271) = "Display Points"
826     Literals(1, 272) = "No Points Display "
827     Literals(1, 273) = "Mean Values Display per"
828     Literals(1, 274) = "Displayed points"
829     Literals(1, 275) = "Return to file control"
830     Literals(1, 276) = "Access to data Values"
831     Literals(1, 277) = "Selection and access to values"
832     Literals(1, 278) = "Selected value :"
833     Literals(1, 279) = "Enable Modify :"
834     Literals(1, 280) = "value :"
835     Literals(1, 281) = "Exit Without Save Changes"
836     Literals(1, 282) = "Save changes and Exit"
837     Literals(1, 283) = "Reset all values to Oritzinal"
838     Literals(1, 284) = "Select exit function"
839     Literals(1, 285) = "Display File in graphic format"
840     Literals(1, 286) = "Not valid initial file , check your file"
841     Literals(1, 287) = "File not validated "
842     Literals(1, 288) = "Disable modify"
843     Literals(1, 289) = "Select mean value display less than < 2"
844     Literals(1, 290) = "No changes"
845     Literals(1, 291) = "Changes been made,Press Yes to save changes and replace
the values as file,Press No discard all changes and use the initial values,Cancel
to not exit from form"
846     Literals(1, 292) = "Restore value"
847     Literals(1, 293) = "Restore all values"
848     ' end sector disp file
849     Literals(1, 294) = "File is not *.txt type" & Chr(13) & "Select only *.txt
file "
850     Literals(1, 295) = "File Selection Error"
851     Literals(1, 296) = "Error on reading file"
852     Literals(1, 297) = "End of file "
853     Literals(1, 298) = "Select transmitting file"
854     Literals(1, 299) = "Transmit File Setpoints"
855     Literals(1, 300) = "Transmit Controls Setpoints"
856     ' sector graphs timer
857     Literals(1, 301) = "Refresh Time Of Graphics"
858     Literals(1, 302) = "Every new value"
859     Literals(1, 303) = "Basic Graphics Period :"
860     Literals(1, 304) = "Graphics Period :"
861     Literals(1, 305) = " Graphs Timing"
862     Literals(1, 306) = "The file has not been appropriate formatted"
863     Literals(1, 307) = "Is already selected file press :" & Chr(13) & "Ok to
transmit it or" & Chr(13) & " Cancel to select a new one"
864     Literals(1, 308) = "You Have Reached Maximum Time Limit, Therefore you cannot
Insert New Page!"
865     Literals(1, 309) = "Reapet Cannot be Executed Because Maximum Time Limit Will
be Reached!"
866     Literals(1, 310) = "Task Will Be Completed Approximately in"
867     Literals(1, 311) = "Graphs"
868     Literals(1, 312) = "You Cannot Modify The Basic Configuration File!"
869     Literals(1, 313) = "Select Default Graph"
870     Literals(1, 314) = ".NET Graph"
871     Literals(1, 315) = "Excel Graph"
872     'Thrust additional literals
873     Literals(1, 316) = "The card is safely closed but an unexpected error in
closing proceedings .For next connection RESET the card."
874     Literals(1, 317) = "The initiate parameters of card haven't the appropriate
values, please reset card from reset button on it."
875     Literals(1, 318) = "Damaged file or reading values are not in appropriate
bounds."
876     Literals(1, 319) = "Do you want Motor to start?"
877     Literals(1, 320) = "Close serial port and Reconnect.The card is been closed."
878     Literals(1, 321) = "Enable motor"
879     Literals(1, 322) = " Timeout of Recieved Data at Serial port "
880     Literals(1, 323) = "Before proceed to any work select a valid Save File." &
ChrW(10) & ChrW(13) & "The selection of file will clear any work at time!"
881     Literals(1, 324) = "Transmiting File Setpoints"
882
883     ' Italian -----
-----
884     Literals(2, 0) = "risparmi i dati"
885     Literals(2, 1) = "risparmi i dati dell'uscita in lima: "
886     Literals(2, 2) = "risparmi i dati di input in lima: "
887     Literals(2, 3) = "trasmettono i dati"

```

```

888     Literals(2, 4) = "tipo i caratteri che desiderate trasmettere"
889     Literals(2, 5) = "trasmettono i caratteri che sono in incluso in lima: "
890     Literals(2, 6) = "Incomming data"
891     Literals(2, 7) = "dati uscenti"
892     Literals(2, 8) = "&Send"
893     Literals(2, 9) = "&File"
894     Literals(2, 10) = "&Connect..."
895     Literals(2, 11) = "&Disconnect"
896     Literals(2, 12) = "&Exit"
897     Literals(2, 13) = "configurazioni"
898     Literals(2, 14) = "lingua del cambiamento"
899     Literals(2, 15) = "Greco"
900     Literals(2, 16) = "inglesi"
901     Literals(2, 17) = "tutti i parametri"
902     Literals(2, 18) = "lingua di difetto"
903     Literals(2, 19) = "dati uscenti del deposito"
904     Literals(2, 20) = "dati ricevuti del deposito"
905     Literals(2, 21) = "trasmettono i dati dalla lima"
906     Literals(2, 22) = "parametri del collegamento"
907     Literals(2, 23) = "risparmi la lima di configurazione"
908     Literals(2, 24) = "lima di configurazione"
909     Literals(2, 25) = "lima prescelta di configurazione"
910     Literals(2, 26) = "DeSelect la lima di configurazione"
911     Literals(2, 27) = "stampa"
912     Literals(2, 28) = "dati uscenti della stampa"
913     Literals(2, 29) = "dati ricevuti della stampa"
914     Literals(2, 30) = "dati della stampa che o erano trasmettono da File"
915     Literals(2, 31) = "aiuto"
916     Literals(2, 32) = "aiuto di esposizione"
917     Literals(2, 33) = "versione"
918     Literals(2, 34) = "Kardasis Diplo del Konstantino"
919     Literals(2, 35) = "sconnessione"
920     Literals(2, 36) = "uscita l'applicazione"
921     Literals(2, 37) = "Kardasis Diplo del Konstantino"
922     Literals(2, 38) = " dati di input liberi "
923     Literals(2, 39) = "dati liberi dell'uscita "
924     Literals(2, 40) = "previsione di dati"
925     Literals(2, 41) = "dati ricevuti"
926     Literals(2, 42) = "dati uscenti"
927     Literals(2, 43) = "il nome di schedario che avete definito erano non valide"
928     Literals(2, 44) = "la lima non potrebbero essere trovate!"
929     Literals(2, 45) = "lima selezionata di configurazioni: "
930     Literals(2, 46) = "incapace scrivere all'orificio di COM!"
931     Literals(2, 47) = "applicazioni"
932     Literals(2, 48) = "dovete definire le lime valide per i dati dell'ingresso/
uscita di Saving per continuare nel collegamento. "
933     Literals(2, 49) = "applicazioni"
934     Literals(2, 50) = "risparmi i dati"
935     Literals(2, 51) = "giri al minuto"
936     Literals(2, 52) = "regolazione di controllo"
937     Literals(2, 53) = "Tempo"
938     Literals(2, 54) = "risparmi i dati in lima: "
939     Literals(2, 55) = "inizio"
940     Literals(2, 56) = "arresto"
941     Literals(2, 57) = "dovete in primo luogo definire una lima valida in cui i
dati saranno conservati! "
942     Literals(2, 58) = "vicino"
943     Literals(2, 59) = "condizione del collegamento: La ricezione dei dati non è
stata iniziata "
944     Literals(2, 60) = "condizione del collegamento: La ricezione dei dati è stata
iniziata "
945     Literals(2, 61) = "ricevendo durata di dati (minuti: Sec)"
946     Literals(2, 62) = "condizione del collegamento: Il collegamento non è stato
Established "
947     Literals(2, 63) = "ricevendo i dati è stata interrotta"
948     Literals(2, 64) = "siete stati staccati"
949     Literals(2, 65) = "versione: "
950     Literals(2, 66) = "Ver" & Version
951     Literals(2, 67) = "data di emissione: "
952     Literals(2, 68) = "generato vicino: "
953     Literals(2, 69) = "Konstantinos Kardasis"
954     Literals(2, 70) = "giusto"
955     Literals(2, 71) = "versione"
956     Literals(2, 72) = "COM Port"
957     Literals(2, 73) = "tasso di baud"

```

```

958     Literals(2, 74) = "parità"
959     Literals(2, 75) = " punte di dati"
960     Literals(2, 76) = "punte di arresto"
961     Literals(2, 77) = "GIUSTO"
962     Literals(2, 78) = "annullamento"
963     Literals(2, 79) = "parametri del collegamento"
964     Literals(2, 80) = "risparmi la lima di configurazioni"
965     Literals(2, 81) = "nome di archivio"
966     Literals(2, 82) = "lingua di difetto"
967     Literals(2, 83) = "definiscono la lingua di difetto"
968     Literals(2, 84) = "l'esposizione definiscono la lingua sulla partenza"
969     Literals(2, 85) = "risparmi i dati dell'uscita"
970     Literals(2, 86) = "nome di archivio"
971     Literals(2, 87) = "risparmi i dati di input"
972     Literals(2, 88) = "nome di archivio"
973     Literals(2, 89) = "trasmettono i dati dalla lima"
974     Literals(2, 90) = "nome di archivio"
975     Literals(2, 91) = "parametri del collegamento"
976     Literals(2, 92) = "orificio di COM"
977     Literals(2, 93) = "tasso di baud"
978     Literals(2, 94) = "parità"
979     Literals(2, 95) = "punte di dati"
980     Literals(2, 96) = "punte di arresto"
981     Literals(2, 97) = "accettano i cambiamenti"
982     Literals(2, 98) = "di nuovo alla finestra principale"
983     Literals(2, 99) = "definiscono i parametri"
984     Literals(2, 100) = "risparmi i dati per l'applicazione stringono l'inerzia"
985     Literals(2, 101) = "nome di archivio"
986     Literals(2, 102) = "Greco"
987     Literals(2, 103) = "inglesi"
988     Literals(2, 104) = "giusto"
989     Literals(2, 105) = "annullamento"
990     Literals(2, 106) = "prego, lingua prescelta"
991     Literals(2, 107) = "la lima"
992     Literals(2, 108) = "è non valide"
993     Literals(2, 109) = "attenzione!"
994     Literals(2, 110) = "là dati da stampare!"
995     Literals(2, 111) = "non potrebbero mostrare la lima di aiuto!"
996     Literals(2, 112) = "architettura navale & ingegneria navale"
997     Literals(2, 113) = "DATI"
998     Literals(2, 114) = "COM non valida Port!"
999     Literals(2, 115) = "tasso di baud non valido!"
1000    Literals(2, 116) = "parità non valida!"
1001    Literals(2, 117) = "punte di dati non valide!"
1002    Literals(2, 118) = "punte non valide di arresto!"
1003    Literals(2, 119) = "un errore circa la lima di configurazioni hanno accaduto! ✖
"
1004    Literals(2, 120) = "dispari"
1005    Literals(2, 121) = "anche"
1006    Literals(2, 122) = "nessun"
1007    Literals(2, 123) = "non potrebbero generare le configurazioni archiviano!"
1008    Literals(2, 124) = "ricevono i periodi"
1009    Literals(2, 125) = "ricevono il distacco"
1010    Literals(2, 126) = DateVersion
1011    Literals(2, 127) = "la lima di configurazioni non potrebbero essere trovate!"
1012    Literals(2, 128) = "non avete definito la lima di configurazioni o non esiste! ✖
"
1013    Literals(2, 129) = "non avete definito la lima in cui i dati trasmessi fossero ✖
conservati o non esistono! "
1014    Literals(2, 130) = "non avete definito la lima in cui i dati ricevuti fossero ✖
conservati o non esistono! "
1015    Literals(2, 131) = "non avete definito la lima di cui il soddisfare poteste ✖
trasmettere o non esiste! "
1016    Literals(2, 132) = "non avete definito la lima in cui i dati dell'inerzia di ✖
coppia di torsione di applicazione fossero conservati! "
1017    Literals(2, 133) = "le lime che conservate i dati dell'ingresso/uscita o di cui ✖
i dati sono di essere trasmettono non dovrebbero essere le stesse! "
1018    Literals(2, 134) = "non potete usare la lima della configurazione per i dati ✖
di Saving! "
1019    Literals(2, 135) = "non potrebbero conservare i cambiamenti!"
1020    Literals(2, 136) = "non potete usare la lima specifica!"
1021    Literals(2, 137) = "la lima selezionata è non valide!"
1022    Literals(2, 138) = "errore! Non ha potuto aprire COM Port! "
1023    Literals(2, 139) = "grafici di esposizione"
1024    Literals(2, 140) = "grafico 1"

```

```

1025     Literals(2, 141) = "grafico 2"
1026     Literals(2, 142) = "grafici del pellame"
1027     Literals(2, 143) = "elevano i grafici"
1028     Literals(2, 144) = "minimizzano i grafici"
1029     Literals(2, 145) = "comunicazione di serie"
1030     Literals(2, 146) = "inerzia di coppia di torsione"
1031     Literals(2, 147) = "spinta dell'acqua e controllo di RPM"
1032     Literals(2, 148) = "forma iniziale"
1033     Literals(2, 149) = "spinta dell'acqua e controllo di RPM"
1034     Literals(2, 150) = "collegamento"
1035     Literals(2, 151) = "forma vicina"
1036     Literals(2, 152) = "condizione"
1037     Literals(2, 153) = "ciclo di PWM (%)"
1038     Literals(2, 154) = "controllo RPM"
1039     Literals(2, 155) = "proprietà dei grafici"
1040     Literals(2, 156) = "Parametarize .NET Gaphs"
1041     Literals(2, 157) = "parametri del primo grafico"
1042     Literals(2, 158) = "RPM"
1043     Literals(2, 159) = "LoadCell (Kgr)"
1044     Literals(2, 160) = "ampère (A) "
1045     Literals(2, 161) = "Tempo"
1046     Literals(2, 162) = "arresto di emergenza"
1047     Literals(2, 163) = "immediatamente processo di arresto?"
1048     Literals(2, 164) = "risparmi i dati per la spinta dell'acqua di applicazione  ⚡
ed il controllo di RPM "
1049     Literals(2, 165) = "non avete definito la lima in cui i dati dell'acqua di  ⚡
applicazione spingessero e controllo di RPM sarà conservato! "
1050     Literals(2, 166) = "le configurazioni disponibili archivia"
1051     Literals(2, 167) = "prescelto"
1052     Literals(2, 168) = "punto di regolazione RPM"
1053     Literals(2, 169) = "tensione"
1054     Literals(2, 170) = "grafici di esposizione con .NET"
1055     Literals(2, 171) = "grafici di esposizione con l'Excel"
1056     Literals(2, 172) = "secondi parametri del grafico"
1057     Literals(2, 173) = "Excel Gaphs di Parametarize"
1058     Literals(2, 174) = "numerico dei punti"
1059     Literals(2, 175) = "linee orizzontali"
1060     Literals(2, 176) = "linee verticali"
1061     Literals(2, 177) = "divisioni orizzontali"
1062     Literals(2, 178) = "carico di PWM"
1063     Literals(2, 179) = "LoadCell"
1064     Literals(2, 180) = "tensione"
1065     Literals(2, 181) = "ampère"
1066     Literals(2, 182) = "linee tratteggiate"
1067     Literals(2, 183) = "linee continue"
1068     Literals(2, 184) = "tipo di linee"
1069     Literals(2, 185) = "esposizione oltre"
1070     Literals(2, 186) = "ora"
1071     Literals(2, 187) = "R.P.M"
1072     Literals(2, 188) = "hanno regolato R.P.M"
1073     Literals(2, 189) = "PWM"
1074     Literals(2, 190) = "hanno regolato PWM"
1075     Literals(2, 191) = "movimento sincrono entrambi i grafici"
1076     Literals(2, 192) = "proprietà della tabella 1"
1077     Literals(2, 193) = "proprietà della tabella 2"
1078     Literals(2, 194) = "linea prescelta: "
1079     Literals(2, 195) = "punti di tabella"
1080     Literals(2, 196) = "pannello del pellame"
1081     Literals(2, 197) = "non avete definito la lima da dove il ciclo del punto di  ⚡
regolazione RPM/PWM (%) sarà trasmette! "
1082     Literals(2, 198) = "generano la lima per trasmettere i punti di regolazione"
1083     Literals(2, 199) = "modo prescelto"
1084     Literals(2, 200) = "metodo per la creazione della lima"
1085     Literals(2, 201) = "manualmente"
1086     Literals(2, 202) = "dalla funzione"
1087     Literals(2, 203) = "importazione dalla lima di testo (*.txt)"
1088     Literals(2, 204) = "scrivono i valori per il RPM: "
1089     Literals(2, 205) = "scrivono i valori per il ciclo di PWM (%): "
1090     Literals(2, 206) = "valore"
1091     Literals(2, 207) = "valori di ripetizione"
1092     Literals(2, 208) = "tempi di ripetizione: "
1093     Literals(2, 209) = "valori di esposizione nel grafico"
1094     Literals(2, 210) = "funzione prescelta"
1095     Literals(2, 211) = "valore iniziale di Tempo: "
1096     Literals(2, 212) = "valore finale di Tempo: "

```

```

1097     Literals(2, 213) = "generano la funzione: "
1098     Literals(2, 214) = "funzione data controllo"
1099     Literals(2, 215) = "funzione selezionata"
1100     Literals(2, 216) = "valori precedenti di esposizione 50"
1101     Literals(2, 217) = "esposizione dopo 50 valori o aggiungono la nuova pagina"
1102     Literals(2, 218) = "esposizione primi 50 valori"
1103     Literals(2, 219) = "valori dell'ultimo 50 di esposizione"
1104     Literals(2, 220) = "primo valore di ripetizione"
1105     Literals(2, 221) = "ultimo valore di ripetizione"
1106     Literals(2, 222) = "eseguono la ripetizione"
1107     Literals(2, 223) = "là erano un errore nei dati di input, quale in la cellula
"
1108     Literals(2, 224) = "funzionamento vengono a mancare, ci era un errore mentre
dati di risparmio in lima "
1109     Literals(2, 225) = "periodo di tempo"
1110     Literals(2, 226) = "pagina di avanzamento"
1111     Literals(2, 227) = "pagina"
1112     Literals(2, 228) = "da"
1113     Literals(2, 229) = "i valori permessi sono"
1114     Literals(2, 230) = "ripetizione di inizio da valore"
1115     Literals(2, 231) = "i valori che avete dato sono non valide"
1116     Literals(2, 232) = "lima prescelta"
1117     Literals(2, 233) = "istruzioni generali"
1118     Literals(2, 234) = "la lima che definirete devono non avere soltanto i valori
aritmetici, linee vuote e generalmente spazi pure come se scegliate il controllo
RPM i valori inclusi devono essere nella regione [0 - 20000] mentre se voi hanno
scelto i valori del ciclo di PWM (%) devono essere nella regione [0 - 100.0]. " &
vbCrLf & "Non appena scegliete la lima che li desiderate dovete scattare sopra il
tasto della lima del controllo per l'applicazione controllerà la compatibilità
della lima. Se la lima è compatibile, allora la volontà del tasto della lima del
carico è permessa. Se lo scattate sopra i dati della lima passeranno all
'applicazione che cancella precedentemente tutti dati inseriti ma non conservati."
1119     Literals(2, 235) = "Dati della lima"
1120     Literals(2, 236) = "Lima del carico "
1121     Literals(2, 237) = "Ci era un errore mentre lima di carico!," "
1122     Literals(2, 238) = "Numero totale di annotazioni in lima: „"
1123     Literals(2, 239) = "Controlli la lima„"
1124     Literals(2, 240) = "Dovete controllo preliminarare la lima "
1125     Literals(2, 241) = "funzioni conosciute"
1126     Literals(2, 242) = "funzioni e periodi di tempo selezionati"
1127     Literals(2, 243) = "funzione del registro"
1128     Literals(2, 244) = "RESET"
1129     Literals(2, 245) = "simboli accettabili"
1130     Literals(2, 246) = "la funzione che avete scritto è non valide. Controlli la
linguetta accettabile di simboli per vedere quali simboli sono accettabili. "
1131     Literals(2, 247) = "eseguono le funzioni"
1132     Literals(2, 248) = "i valori che di Tempo avete fornito sono non valide!"
1133     Literals(2, 249) = "già avete dato un'altra funzione in questo periodo! "
1134     Literals(2, 250) = "avete cambiato la funzione selezionata!"
1135     Literals(2, 251) = "là erano un errore mentre facevano funzionare le funzioni
convalidate! "
1136     Literals(2, 252) = "periodi di tempo"
1137     Literals(2, 253) = "funzione"
1138     Literals(2, 254) = "funziona"
1139     Literals(2, 255) = "simboli"
1140     Literals(2, 256) = "variabili"
1141     Literals(2, 257) = "non avete definito una funzione!"
1142     Literals(2, 258) = "i valori che vengono dalle funzioni sono dai limiti! "
1143     Literals(2, 259) = "Italiano"
1144     Literals(2, 260) = "Spagnolo"
1145     Literals(2, 261) = "Francese"
1146     Literals(2, 262) = "Attenda prego per caricare i dati." & vbCrLf & "Se non
desiderate caricare i dati potete scattare sopra il "" di operazione dell
'annullamento del "" del tasto. "
1147     Literals(2, 263) = "Progresso di operazione"
1148     Literals(2, 264) = "Annulli l'operazione"
1149     Literals(2, 265) = "Inizi"
1150     Literals(2, 266) = "Arresti"
1151     Literals(2, 267) = "Pausa"
1152     Literals(2, 268) = "Resume"
1153     Literals(2, 269) = "Il valore che massimo di Tempo potete dichiarare è "
1154     Literals(2, 270) = "verticale per la spettrografia di massa"
1155     Literals(2, 271) = "l'esposizione indica"
1156     Literals(2, 272) = "nessun'esposizione dei punti"
1157     Literals(2, 273) = "esposizione di valori medi per"

```

```

1158     Literals(2, 274) = "hanno visualizzato i punti"
1159     Literals(2, 275) = "ritorno a controllo della lima"
1160     Literals(2, 276) = "accesso ai valori di dati"
1161     Literals(2, 277) = "selezione ed accesso ai valori"
1162     Literals(2, 278) = "hanno selezionato il valore: "
1163     Literals(2, 279) = "permettono modificano: "
1164     Literals(2, 280) = "valore: "
1165     Literals(1, 281) = "Exit Without Save Changes"
1166     Literals(2, 282) = "risparmi i cambiamenti e l'uscita"
1167     Literals(2, 283) = "hanno ripristinato tutti i valori a Oritzinal"
1168     Literals(2, 284) = "funzione prescelta dell'uscita"
1169     Literals(2, 285) = "lima di esposizione nella disposizione grafica"
1170     Literals(2, 286) = "la lima iniziale non valida, controllano la vostra lima"
1171     Literals(2, 287) = "lima non convalidata"
1172     Literals(2, 288) = "inabilitano modificano"
1173     Literals(2, 289) = "esposizione prescelta di valore medio di meno che < 2"
1174     Literals(2, 290) = "nessun cambiamenti"
1175     Literals(2, 291) = "cambiamenti fatto, premono sì per conservare i cambiamenti
e sostituire i valori come lima, per non premere scarto tutti i cambiamenti e per
usare i valori iniziali , l'annullamento per non rimuovere dalla forma "
1176     Literals(2, 292) = "valore di Restore"
1177     Literals(1, 293) = "Restore all values"
1178     Literals(2, 294) = "lima non è tipo di *.txt" & Chr(13) & "seleziona soltanto
lima di *.txt "
1179     Literals(2, 295) = "errore di selezione della lima"
1180     Literals(2, 296) = "errore sulla lima della lettura"
1181     Literals(2, 297) = "estremità della lima"
1182     Literals(2, 298) = "selezionano la lima trasmettente"
1183     Literals(2, 299) = "trasmettono i Setpoints della lima"
1184     Literals(2, 300) = "trasmettono i Setpoints di comandi"
1185     Literals(2, 301) = "rinfrescano Periodo dei grafici"
1186     Literals(2, 302) = "ogni nuovo valore"
1187     Literals(2, 303) = "periodo di base dei grafici: "
1188     Literals(2, 304) = "periodo dei grafici: "
1189     Literals(2, 305) = "cronometrare dei grafici"
1190     Literals(2, 306) = "la lima non è stata adatto formattate"
1191     Literals(2, 307) = "è già pressa selezionata della lima: " & Chr(13) & "giusto
trasmetterlo o" & Chr(13) & " annullamento per selezionare un nuovo "
1192     Literals(2, 308) = "avete raggiunto la scadenza Massima, di conseguenza non
potete inserire la nuova pagina! "
1193     Literals(2, 309) = "ripetizione non possono essere eseguite Poiché la scadenza
Massima sarà raggiunta! "
1194     Literals(2, 310) = "l'operazione saranno completate approssimativamente in "
1195     Literals(2, 311) = "grafici,,"
1196     Literals(2, 312) = "non potete modificare la lima di configurazione di base! "
1197     Literals(2, 313) = "selezionano il grafico di difetto "
1198     Literals(2, 314) = "grafico di .NET,,"
1199     Literals(2, 315) = "grafico dell'Excel,,"
1200     'Costanti letterali supplementari di spinta
1201     Literals(2, 316) = "la scheda è sicuro chiuse ma un errore inatteso negli atti
di chiusura. Per collegamento seguente RIPRISTINATO la scheda. "
1202     Literals(2, 317) = "i parametri iniziati della scheda non hanno i valori
adatti, prego hanno ripristinato la scheda dal tasto di risistemazione su esso. "
1203     Literals(2, 318) = "lima danneggiata o valori della lettura non sono in limiti
adatti. "
1204     Literals(2, 319) = "desiderate il motore cominciare? "
1205     Literals(2, 320) = "orificio di serie vicino e ricollegano. La scheda è stata
chiusa. "
1206     Literals(2, 321) = "Permetta il motore"
1207     Literals(2, 322) = "Prespegnimento dei dati ricevuti ad orificio di serie"
1208     Literals(2, 323) = "Prima che proceed a tutto il lavoro selezioni un valido
risparmi la lima. " & ChrW(10) & ChrW(13) & " La selezione della lima eliminerà
tutto il lavoro a tempo! "
1209     Literals(2, 324) = "Setpoints della lima di Transmiting"
1210
1211     ' Spain -----
-----
1212     Literals(3, 0) = "excepto datos"
1213     Literals(3, 1) = "excepto datos de la salida en archivo: "
1214     Literals(3, 2) = "excepto datos de entrada en archivo: "
1215     Literals(3, 3) = "envían datos"
1216     Literals(3, 4) = "tipo los caracteres que usted desea enviar"
1217     Literals(3, 5) = "envían los caracteres que están en incluido en archivo: "
1218     Literals(3, 6) = "datos de Incomming"
1219     Literals(3, 7) = "datos salientes"

```

```

1220     Literals(3, 8) = "&Send"
1221     Literals(3, 9) = "&File"
1222     Literals(3, 10) = "&Connect..."
1223     Literals(3, 11) = "&Disconnect"
1224     Literals(3, 12) = "&Exit"
1225     Literals(3, 13) = "configuraciones"
1226     Literals(3, 14) = "lengua del cambio"
1227     Literals(3, 15) = "Griego"
1228     Literals(3, 16) = "ingleses"
1229     Literals(3, 17) = "todos los parámetros"
1230     Literals(3, 18) = "lengua del defecto"
1231     Literals(3, 19) = "datos salientes del almacén"
1232     Literals(3, 20) = "datos entrantes del almacén"
1233     Literals(3, 21) = "envían datos de archivo"
1234     Literals(3, 22) = "parámetros de la conexión"
1235     Literals(3, 23) = "excepto archivo de la configuración"
1236     Literals(3, 24) = "archivo de la configuración"
1237     Literals(3, 25) = "archivo selecto de la configuración"
1238     Literals(3, 26) = "DeSelect el archivo de la configuración"
1239     Literals(3, 27) = "impresión"
1240     Literals(3, 28) = "datos salientes de la impresión"
1241     Literals(3, 29) = "datos entrantes de la impresión"
1242     Literals(3, 30) = "datos de la impresión que o eran envían por File"
1243     Literals(3, 31) = "ayuda"
1244     Literals(3, 32) = "ayuda de la demostración"
1245     Literals(3, 33) = "versión"
1246     Literals(3, 34) = "Kardasis Diplo de Konstantino"
1247     Literals(3, 35) = "desconexión"
1248     Literals(3, 36) = "salida el uso"
1249     Literals(3, 37) = "Kardasis Diplo de Konstantino"
1250     Literals(3, 38) = "datos de entrada claros"
1251     Literals(3, 39) = "datos claros de la salida"
1252     Literals(3, 40) = "inspección previo de los datos"
1253     Literals(3, 41) = "datos entrantes"
1254     Literals(3, 42) = "datos salientes"
1255     Literals(3, 43) = "el nombre de fichero que usted definió eran inválidos"
1256     Literals(3, 44) = "el archivo no se podrían encontrar!"
1257     Literals(3, 45) = "archivo seleccionado de las configuraciones: "
1258     Literals(3, 46) = "incapaz escribir al puerto de COM!"
1259     Literals(3, 47) = "usos"
1260     Literals(3, 48) = "usted debe definir los archivos válidos para los datos de
la entrada-salida del ahorro para proceder en la conexión. "
1261     Literals(3, 49) = "usos"
1262     Literals(3, 50) = "excepto datos"
1263     Literals(3, 51) = "revoluciones por minuto"
1264     Literals(3, 52) = "regulación del control"
1265     Literals(3, 53) = "Tiempo"
1266     Literals(3, 54) = "excepto datos en archivo: "
1267     Literals(3, 55) = "comienzo"
1268     Literals(3, 56) = "parada"
1269     Literals(3, 57) = "usted debe primero definir un archivo válido donde los
datos serán ahorrados! "
1270     Literals(3, 58) = "cercano"
1271     Literals(3, 59) = "estado de la conexión: La recepción de datos no se ha
comenzado "
1272     Literals(3, 60) = "estado de la conexión: Se ha comenzado la recepción de
datos "
1273     Literals(3, 61) = "recibiendo la duración de los datos (minutos: Secs)"
1274     Literals(3, 62) = "estado de la conexión: La conexión no ha sido Established "
1275     Literals(3, 63) = "recibiendo datos"
1276     Literals(3, 64) = "le han desconectado"
1277     Literals(3, 65) = "versión: "
1278     Literals(3, 66) = "Ver" & Version
1279     Literals(3, 67) = "fecha del lanzamiento: "
1280     Literals(3, 68) = "creado cerca: "
1281     Literals(3, 69) = "Konstantinos Kardasis"
1282     Literals(3, 70) = "aceptable"
1283     Literals(3, 71) = "versión"
1284     Literals(3, 72) = "puerto de COM"
1285     Literals(3, 73) = "velocidad"
1286     Literals(3, 74) = "paridad"
1287     Literals(3, 75) = "bits de datos"
1288     Literals(3, 76) = "pedacitos de parada"
1289     Literals(1, 77) = "OK"
1290     Literals(3, 78) = "cancelación"

```



```

1291     Literals(3, 79) = "parámetros de la conexión"
1292     Literals(3, 80) = "excepto archivo de las configuraciones"
1293     Literals(3, 81) = "nombre del archivo"
1294     Literals(3, 82) = "lengua del defecto"
1295     Literals(3, 83) = "definen lengua del defecto"
1296     Literals(3, 84) = "demostración definen lengua en arranque"
1297     Literals(3, 85) = "excepto datos de la salida"
1298     Literals(3, 86) = "nombre del archivo"
1299     Literals(3, 87) = "excepto datos de entrada"
1300     Literals(3, 88) = "nombre del archivo"
1301     Literals(3, 89) = "envían datos de archivo"
1302     Literals(3, 90) = "nombre del archivo"
1303     Literals(3, 91) = "parámetros de la conexión"
1304     Literals(3, 92) = "puerto de COM"
1305     Literals(3, 93) = "velocidad"
1306     Literals(3, 94) = "paridad"
1307     Literals(3, 95) = "bits de datos"
1308     Literals(3, 96) = "pedacitos de parada"
1309     Literals(3, 97) = "aceptan cambios"
1310     Literals(3, 98) = "de nuevo a ventana principal"
1311     Literals(3, 99) = "definen parámetros"
1312     Literals(3, 100) = "excepto los datos para el uso aprietan la inercia"
1313     Literals(3, 101) = "nombre del archivo"
1314     Literals(3, 102) = "Griego"
1315     Literals(3, 103) = "ingleses"
1316     Literals(3, 104) = "aceptable"
1317     Literals(3, 105) = "cancelación"
1318     Literals(3, 106) = "por favor, lengua selecta"
1319     Literals(3, 107) = "el archivo"
1320     Literals(3, 108) = "son inválidos"
1321     Literals(3, 109) = "atención!"
1322     Literals(3, 110) = "allí no hay datos a imprimir!"
1323     Literals(3, 111) = "no podrían demostrar el archivo de la ayuda!"
1324     Literals(3, 112) = "arquitectura y ingeniería de marina navales"
1325     Literals(3, 113) = "DATOS"
1326     Literals(3, 114) = "COM inválida viran hacia el lado de babor!"
1327     Literals(3, 115) = "velocidad inválida!"
1328     Literals(3, 116) = "paridad inválida!"
1329     Literals(3, 117) = "bits de datos inválidos!"
1330     Literals(3, 118) = "pedacitos de parada inválidos!"
1331     Literals(3, 119) = "un error sobre el archivo de las configuraciones
ocurrieron! "
1332     Literals(3, 120) = "impar"
1333     Literals(3, 121) = "uniforme"
1334     Literals(3, 122) = "ningunos"
1335     Literals(3, 123) = "no podrían crear las configuraciones archivan!"
1336     Literals(3, 124) = "reciben épocas"
1337     Literals(3, 125) = "reciben despegue"
1338     Literals(3, 126) = DateVersion
1339     Literals(3, 127) = "el archivo de las configuraciones no se podrían encontrar!
"
1340     Literals(3, 128) = "usted no ha definido el archivo de las configuraciones o
no existe! "
1341     Literals(3, 129) = "usted no ha definido el archivo donde los datos enviados
serán ahorrados o no existen! "
1342     Literals(3, 130) = "usted no ha definido el archivo donde los datos recibidos
serán ahorrados o no existen! "
1343     Literals(3, 131) = "usted no ha definido el archivo que contenido usted puede
enviar o no existe! "
1344     Literals(3, 132) = "usted no ha definido el archivo donde los datos de la
inercia del esfuerzo de torsión del uso serán ahorrados! "
1345     Literals(3, 133) = "los archivos que usted ahorra datos de la entrada-salida o
que datos son ser envían no deben ser iguales! "
1346     Literals(3, 134) = "usted no puede utilizar el archivo de la configuración
para los datos del ahorro! "
1347     Literals(3, 135) = "no podrían ahorrar cambios!"
1348     Literals(3, 136) = "usted no puede utilizar el archivo específico!"
1349     Literals(3, 137) = "el archivo seleccionado son inválidos!"
1350     Literals(3, 138) = "error! ¡No podía abrir a COM viran hacia el lado de babor!
"
1351     Literals(3, 139) = "gráficos de la demostración"
1352     Literals(3, 140) = "gráfico 1"
1353     Literals(3, 141) = "gráfico 2"
1354     Literals(3, 142) = "gráficos de la piel"
1355     Literals(3, 143) = "maximizan gráficos"

```

```

1356     Literals(3, 144) = "reducen al mínimo gráficos"
1357     Literals(3, 145) = "comunicación serial"
1358     Literals(3, 146) = "inercia del esfuerzo de torsión"
1359     Literals(3, 147) = "empuje del agua y control de la RPM"
1360     Literals(3, 148) = "forma inicial "
1361     Literals(3, 149) = "empuje del agua y control de la RPM"
1362     Literals(3, 150) = "conexión"
1363     Literals(3, 151) = "forma cercana"
1364     Literals(3, 152) = "estado"
1365     Literals(3, 153) = "ciclo de PWM (%)"
1366     Literals(3, 154) = "control RPM"
1367     Literals(3, 155) = "características de los gráficos"
1368     Literals(3, 156) = "Parametarize .NET Gaphs"
1369     Literals(3, 157) = "parámetros primer gráfico"
1370     Literals(3, 158) = "RPM"
1371     Literals(3, 159) = "LoadCell (Kgr)"
1372     Literals(3, 160) = "amperios (a)"
1373     Literals(3, 161) = "Tiempo"
1374     Literals(3, 162) = "parada de emergencia"
1375     Literals(3, 163) = "inmediatamente proceso de la parada?"
1376     Literals(3, 164) = "excepto los datos para el empuje del agua del uso y el
control de la RPM "
1377     Literals(3, 165) = "usted no ha definido el archivo adonde los datos del agua
del uso empujaron y control de la RPM será ahorrada! "
1378     Literals(3, 166) = "configuraciones disponibles archivan"
1379     Literals(3, 167) = "selecto"
1380     Literals(3, 168) = "punto de ajuste RPM"
1381     Literals(3, 169) = "voltaje"
1382     Literals(3, 170) = "gráficos de la demostración con .NET"
1383     Literals(3, 171) = "gráficos de la demostración con Excel"
1384     Literals(3, 172) = "segundos parámetros del gráfico"
1385     Literals(3, 173) = "Excel Gaphs de Parametarize"
1386     Literals(3, 174) = "numérico de puntos"
1387     Literals(3, 175) = "líneas horizontales"
1388     Literals(3, 176) = "líneas verticales"
1389     Literals(3, 177) = "Div horizontales"
1390     Literals(3, 178) = "carga de PWM"
1391     Literals(3, 179) = "LoadCell"
1392     Literals(3, 180) = "voltaje"
1393     Literals(3, 181) = "amperios"
1394     Literals(3, 182) = "líneas discontinuas"
1395     Literals(3, 183) = "líneas llenas"
1396     Literals(3, 184) = "tipo de líneas"
1397     Literals(3, 185) = "demostración más allá de"
1398     Literals(3, 186) = "ahora"
1399     Literals(3, 187) = "R.P.M"
1400     Literals(3, 188) = "fijaron R.P.M"
1401     Literals(3, 189) = "PWM"
1402     Literals(3, 190) = "fijaron PWM"
1403     Literals(3, 191) = "movimiento síncrono ambos gráficos"
1404     Literals(3, 192) = "características de la carta 1"
1405     Literals(3, 193) = "características de la carta 2"
1406     Literals(3, 194) = "línea selecta: "
1407     Literals(3, 195) = "puntos de carta"
1408     Literals(3, 196) = "panel de la piel"
1409     Literals(3, 197) = "usted no ha definido el archivo de donde estará el ciclo
del punto de ajuste RPM/PWM (%) envía! "
1410     Literals(3, 198) = "crean el archivo para enviar puntos de ajuste"
1411     Literals(3, 199) = "modo selecto"
1412     Literals(3, 200) = "método para la creación del archivo"
1413     Literals(3, 201) = "manualmente"
1414     Literals(3, 202) = "de la función"
1415     Literals(3, 203) = "importación del archivo de texto (*.txt)"
1416     Literals(3, 204) = "escriben los valores para la RPM: "
1417     Literals(3, 205) = "escriben los valores para el ciclo de PWM (%): "
1418     Literals(3, 206) = "valor"
1419     Literals(3, 207) = "valores de la repetición"
1420     Literals(3, 208) = "tiempos de la repetición: "
1421     Literals(3, 209) = "valores de la demostración en gráfico"
1422     Literals(3, 210) = "función selecta"
1423     Literals(3, 211) = "valor inicial de Tiempo: "
1424     Literals(3, 212) = "valor final de Tiempo: "
1425     Literals(3, 213) = "crean la función: "
1426     Literals(3, 214) = "función dada cheque"
1427     Literals(3, 215) = "función seleccionada"

```

```

1428     Literals(3, 216) = "valores anteriores de la demostración 50"
1429     Literals(3, 217) = "demostración después 50 valores o agregan la página nueva"
1430     Literals(3, 218) = "demostración primeros 50 valores"
1431     Literals(3, 219) = "valores del último 50 de la demostración"
1432     Literals(3, 220) = "primer valor de la repetición"
1433     Literals(3, 221) = "valor pasado de la repetición"
1434     Literals(3, 222) = "ejecutan la repetición"
1435     Literals(3, 223) = "allí eran un error en los datos de entrada, por ejemplo en
la célula "
1436     Literals(3, 224) = "operación fallaron, había un error mientras que los datos
que ahorran en archivo "
1437     Literals(3, 225) = "período"
1438     Literals(3, 226) = "página Goto"
1439     Literals(3, 227) = "página"
1440     Literals(3, 228) = "de"
1441     Literals(3, 229) = "los valores permitidos son"
1442     Literals(3, 230) = "repetición del comienzo del valor"
1443     Literals(3, 231) = "los valores que usted ha dado son inválidos"
1444     Literals(3, 232) = "archivo selecto"
1445     Literals(3, 233) = "instrucciones generales"
1446     Literals(3, 234) = "el archivo que usted definirá no deben tener solamente
valores aritméticos, ningunas líneas vacías y generalmente ningún espacio también
como si usted ha elegido el control RPM los valores incluidos deben estar en la
región [0 - 20000] mientras que si usted han elegido los valores del ciclo de PWM
(%) deben estar en la región [0 - 100.0]. " & vbCrLf & "Tan pronto como usted
elige el archivo que usted le desea debe chascar encendido el botón del archivo
del cheque para uso comprobará la compatibilidad del archivo. Si el archivo es
compatible, entonces se permita la voluntad del botón del archivo de la carga. Si
usted lo chasca encendido los datos del archivo pasarán al uso que suprime todos
previamente datos insertados pero no ahorrados."
1447     Literals(3, 235) = "Datos del archivo"
1448     Literals(3, 236) = "Archivo de la carga"
1449     Literals(3, 237) = "Había un error mientras que el archivo que carga!"
1450     Literals(3, 238) = "Número total de expedientes en archivo: "
1451     Literals(3, 239) = "Compruebe el archivo"
1452     Literals(3, 240) = "Usted debe primer cheque el archivo"
1453     Literals(3, 241) = "funciones sabidas"
1454     Literals(3, 242) = "funciones y períodos Seleccionados"
1455     Literals(3, 243) = "función del registro"
1456     Literals(3, 244) = "RESET"
1457     Literals(3, 245) = "símbolos aceptables"
1458     Literals(3, 246) = "la función que usted mecanografió son inválidos. Compruebe
la lengüeta aceptable de los símbolos para ver qué símbolos son aceptables. "
1459     Literals(3, 247) = "ejecutan funciones"
1460     Literals(3, 248) = "los valores de Tiempo que usted proporcionó son inválidos!
"
1461     Literals(3, 249) = "usted ha dado ya otra función en este periodo! "
1462     Literals(3, 250) = "usted ha cambiado la función seleccionada!"
1463     Literals(3, 251) = "allí eran un error mientras que funcionaban funciones
validadas! "
1464     Literals(3, 252) = "períodos"
1465     Literals(3, 253) = "función"
1466     Literals(3, 254) = "funcionan"
1467     Literals(3, 255) = "símbolos"
1468     Literals(3, 256) = "variables"
1469     Literals(3, 257) = "usted no ha definido una función!"
1470     Literals(3, 258) = "los valores que vienen de las funciones están fuera de
límites! "
1471     Literals(3, 259) = "Itálica"
1472     Literals(3, 260) = "Español"
1473     Literals(3, 261) = "Francés"
1474     Literals(3, 262) = "Espere por favor para cargar datos." & vbCrLf & "Si usted
no desea cargar datos usted puede chascar encendido el "" de la tarea de la
cancelación del "" del botón. "
1475     Literals(3, 263) = "Progreso de la tarea"
1476     Literals(3, 264) = "Cancele la tarea"
1477     Literals(3, 265) = "Comience"
1478     Literals(3, 266) = "Pare"
1479     Literals(3, 267) = "Pausa"
1480     Literals(3, 268) = "Curriculum vitae"
1481     Literals(3, 269) = "El valor máximo de Tiempo que usted puede declarar es "
1482     Literals(3, 270) = "vertical por ms"
1483     Literals(3, 271) = "exhibición señalan"
1484     Literals(3, 272) = "ninguna exhibición de los puntos"
1485     Literals(3, 273) = "exhibición de los valores medios por"

```

```

1486     Literals(3, 274) = "exhibieron puntos"
1487     Literals(3, 275) = "vuelta al control del archivo"
1488     Literals(3, 276) = "acceso a los valores de los datos"
1489     Literals(3, 277) = "selección y acceso a los valores"
1490     Literals(3, 278) = "seleccionaron valor: "
1491     Literals(3, 279) = "permiten se modifican: "
1492
1493     Literals(3, 280) = "valor: "
1494     Literals(3, 281) = "salida sin excepto cambios"
1495     Literals(3, 282) = "excepto cambios y la salida"
1496     Literals(3, 283) = "reajustaron todos los valores a Oritzinal"
1497     Literals(3, 284) = "función selecta de la salida"
1498     Literals(3, 285) = "archivo de exhibición en formato gráfico"
1499     Literals(3, 286) = "archivo inicial inválido, comprueban su archivo"
1500     Literals(3, 287) = "archivo no validado"
1501     Literals(3, 288) = "inhabilitan se modifican"
1502     Literals(3, 289) = "exhibición selecta del valor medio menos que < 2"
1503     Literals(3, 290) = "ningunos cambios"
1504     Literals(3, 291) = "cambios hecho, presionan sí para ahorrar cambios y para
substituir los valores como archivo, para no presionar ningún descarte todos los
cambios y para utilizar los valores iniciales, la cancelación para no salir de
forma "
1505     Literals(3, 292) = "valor del Restore"
1506     Literals(3, 293) = "Restore todos los valores"
1507     Literals(3, 294) = "archivo no son tipo de *.txt" & Chr(13) & "selecciona
solamente archivo de *.txt "
1508     Literals(3, 295) = "error de la selección del archivo"
1509     Literals(3, 296) = "error en archivo de la lectura"
1510     Literals(3, 297) = "extremo del archivo"
1511     Literals(3, 298) = "seleccionan el archivo que transmite"
1512     Literals(3, 299) = "transmiten Setpoints del archivo"
1513     Literals(3, 300) = "transmiten Setpoints de los controles"
1514     Literals(3, 301) = "restauran Época de gráficos"
1515     Literals(3, 302) = "cada nuevo valor"
1516     Literals(3, 303) = "período básico de los gráficos: "
1517     Literals(3, 304) = "período de los gráficos: "
1518     Literals(3, 305) = "medir el tiempo de los gráficos"
1519     Literals(3, 306) = "el archivo no han sido apropiados ajustados a formato"
1520     Literals(3, 307) = "son ya prensa seleccionada del archivo: " & Chr(13) &
"aceptable transmitir lo o" & Chr(13) & " cancelación para seleccionar un nuevo "
1521     Literals(3, 308) = "usted ha alcanzado el límite de tiempo máximo, por lo
tanto usted no puede insertar la página nueva! "
1522     Literals(3, 309) = "repetición no pueden ser ejecutados porque el límite de
tiempo máximo será alcanzado! "
1523     Literals(3, 310) = "tarea serán terminados aproximadamente en "
1524     Literals(3, 311) = "gráficos"
1525     Literals(3, 312) = "usted no puede modificar el archivo de la configuración
básica! "
1526     Literals(3, 313) = "seleccionan el gráfico del defecto "
1527     Literals(3, 314) = "gráfico de .NET"
1528     Literals(3, 315) = "gráfico del Excel"
1529     'Literales adicionales del empuje
1530     Literals(3, 316) = "latarjeta están cerrados con seguridad solamente un error
inesperado en procedimientos de cierre. Para la conexión siguiente REAJUSTADA la
tarjeta. "
1531     Literals(3, 317) = "los parámetros iniciados de la tarjeta no tienen los
valores apropiados, reajustaron por favor la tarjeta del botón del reajuste en él.
"
1532     Literals(3, 318) = "archivo dañado o los valores de la lectura no están en
límites apropiados. "
1533     Literals(3, 319) = "usted quisiera que el motor comenzara? "
1534     Literals(3, 320) = "puerto serial cercano y vuelven a conectar. La tarjeta
está cerrada. "
1535     Literals(3, 321) = "Permita el motor"
1536     Literals(3, 322) = " Timeout of Recieved Data at Serial port "
1537     Literals(3, 323) = "Antes de que procceed a cualquier trabajo seleccione un
válido excepto archivo. " & ChrW(10) & ChrW(13) & " La selección del archivo
despejará cualquier trabajo en el tiempo! "
1538     Literals(3, 324) = "Setpoints del archivo de Transmiting"
1539
1540
1541     ' France -----
-----
1542     Literals(4, 0) = " économiser des données "
1543     Literals(4, 1) = " économiser des données de rendement dans le dossier : "

```

```

1544     Literals(4, 2) = " économiser des données d'entrée dans le dossier : "
1545     Literals(4, 3) = " envoient des données "
1546     Literals(4, 4) = " type les caractères que vous souhaitez envoyer "
1547     Literals(4, 5) = " envoient les caractères qui sont dans inclus dans le dossier : "
1548     Literals(4, 6) = " données d'Incomming "
1549     Literals(4, 7) = " données sortantes "
1550     Literals(4, 8) = " &Send "
1551     Literals(4, 9) = " &File "
1552     Literals(4, 10) = " &Connect... "
1553     Literals(4, 11) = " &Disconnect "
1554     Literals(4, 12) = " &Exit "
1555     Literals(4, 13) = " configurations "
1556     Literals(4, 14) = " langue de changement "
1557     Literals(4, 15) = " Grec "
1558     Literals(4, 16) = " les anglais "
1559     Literals(4, 17) = " tous les paramètres "
1560     Literals(4, 18) = " langue de défaut "
1561     Literals(4, 19) = " données sortantes de magasin "
1562     Literals(4, 20) = " données entrantes de magasin "
1563     Literals(4, 21) = " envoient des données à partir du dossier "
1564     Literals(4, 22) = " paramètres de raccordement "
1565     Literals(4, 23) = " économiser le dossier de configuration "
1566     Literals(4, 24) = " dossier de configuration "
1567     Literals(4, 25) = " dossier choisi de configuration "
1568     Literals(4, 26) = " ne pas sélectionner le dossier de configuration "
1569     Literals(4, 27) = " copie "
1570     Literals(4, 28) = " données sortantes d'impression "
1571     Literals(4, 29) = " données entrantes d'impression "
1572     Literals(4, 30) = "les données d'impression qui ou étaient envoient par File "
1573     Literals(4, 31) = " aide "
1574     Literals(4, 32) = " aide d'exposition "
1575     Literals(4, 33) = " version "
1576     Literals(4, 34) = " Kardasis Diplo de Konstantino "
1577     Literals(4, 35) = " débranchement "
1578     Literals(4, 36) = " sortie l'application "
1579     Literals(4, 37) = " Kardasis Diplo de Konstantino "
1580     Literals(4, 38) = " données d'entrée claires "
1581     Literals(4, 39) = " données claires de rendement "
1582     Literals(4, 40) = " prévision de données "
1583     Literals(4, 41) = " données entrantes "
1584     Literals(4, 42) = " données sortantes "
1585     Literals(4, 43) = " le nom de fichier que vous avez défini étaient inadmissibles "
1586     Literals(4, 44) = " le dossier ne pourraient pas être trouvées ! "
1587     Literals(4, 45) = " dossier choisi de configurations : "
1588     Literals(4, 46) = " incapable d'écrire au port de COM ! "
1589     Literals(4, 47) = " applications "
1590     Literals(4, 48) = " vous devez définir les dossiers valides pour des données d'entrée-sortie d'économie afin de procéder dans le raccordement. "
1591     Literals(4, 49) = " applications "
1592     Literals(4, 50) = " économiser des données "
1593     Literals(4, 51) = " révolutions par minute "
1594     Literals(4, 52) = " règlement de commande "
1595     Literals(4, 53) = " Temps "
1596     Literals(4, 54) = " économiser des données dans le dossier : "
1597     Literals(4, 55) = " début "
1598     Literals(4, 56) = " arrêt "
1599     Literals(4, 57) = " vous devez d'abord définir un dossier valide où les données seront sauvées ! "
1600     Literals(4, 58) = " étroit "
1601     Literals(4, 59) = " statut de raccordement : La réception des données n'a pas été commencée "
1602     Literals(4, 60) = " statut de raccordement : La réception des données a été commencée "
1603     Literals(4, 61) = " recevant la durée de données (minutes : Sec) "
1604     Literals(4, 62) = " statut de raccordement : Le raccordement n'a pas été Establised "
1605     Literals(4, 63) = " recevant des données a été arrêtées "
1606     Literals(4, 64) = " vous avez été débranché "
1607     Literals(4, 65) = " version : "
1608     Literals(4, 66) = " Ver et version"
1609     Literals(4, 67) = " date de dégagement : "
1610     Literals(4, 68) = " créé par : "
1611     Literals(4, 69) = " Konstantinos Kardasis "

```

```

1612     Literals(4, 70) = " correct "
1613     Literals(4, 71) = " version "
1614     Literals(4, 72) = " port de COM "
1615     Literals(4, 73) = " vitesse baud "
1616     Literals(4, 74) = " parité "
1617     Literals(4, 75) = " bits d'informations "
1618     Literals(4, 76) = " bits d'arrêt "
1619     Literals(4, 77) = " CORRECT "
1620     Literals(4, 78) = " annulation "
1621     Literals(4, 79) = " paramètres de raccordement "
1622     Literals(4, 80) = " économiser le dossier de configurations "
1623     Literals(4, 81) = " nom de fichier "
1624     Literals(4, 82) = " langue de défaut "
1625     Literals(4, 83) = " définissent la langue de défaut "
1626     Literals(4, 84) = "la exposition définissent la langue sur le démarrage "
1627     Literals(4, 85) = " économiser des données de rendement "
1628     Literals(4, 86) = " nom de fichier "
1629     Literals(4, 87) = " économiser des données d'entrée "
1630     Literals(4, 88) = " nom de fichier "
1631     Literals(4, 89) = " envoient des données à partir du dossier "
1632     Literals(4, 90) = " nom de fichier "
1633     Literals(4, 91) = " paramètres de raccordement "
1634     Literals(4, 92) = " port de COM "
1635     Literals(4, 93) = " vitesse baud "
1636     Literals(4, 94) = " parité "
1637     Literals(4, 95) = " bits d'informations "
1638     Literals(4, 96) = " bits d'arrêt "
1639     Literals(4, 97) = " acceptent des changements "
1640     Literals(4, 98) = " de nouveau à la fenêtre principale "
1641     Literals(4, 99) = " définissent des paramètres "
1642     Literals(4, 100) = " économiser des données pour l'application serrent à la
    clé dynamométrique l'inertie " 'inactivité = inertie ?
1643     Literals(4, 101) = " nom de fichier "
1644     Literals(4, 102) = " Grec "
1645     Literals(4, 103) = "les anglais "
1646     Literals(4, 104) = " correct "
1647     Literals(4, 105) = " annulation "
1648     Literals(4, 106) = " svp, langue choisie "
1649     Literals(4, 107) = " le dossier "
1650     Literals(4, 108) = " est inadmissible "
1651     Literals(4, 109) = " attention ! "
1652     Literals(4, 110) = " là ne sont aucune donnée à imprimer ! "
1653     Literals(4, 111) = " ne pourraient pas montrer le dossier d'aide ! "
1654     Literals(4, 112) = " architecture navale et technologie marine "
1655     Literals(4, 113) = " DONNÉES "
1656     Literals(4, 114) = " COM inadmissible mettent en communication ! "
1657     Literals(4, 115) = " vitesse baud inadmissible ! "
1658     Literals(4, 116) = " parité inadmissible ! "
1659     Literals(4, 117) = " bits d'informations inadmissibles ! "
1660     Literals(4, 118) = " bits d'arrêt inadmissibles ! "
1661     Literals(4, 119) = " une erreur au sujet du dossier de configurations se sont
    produites ! "
1662     Literals(4, 120) = " impair "
1663     Literals(4, 121) = " même "
1664     Literals(4, 122) = " aucun "
1665     Literals(4, 123) = " ne pourraient pas créer le dossier de configurations ! "
1666     Literals(4, 124) = " reçoivent des périodes "
1667     Literals(4, 125) = " reçoivent le décollement "
1668     Literals(4, 126) = DateVersion
1669     Literals(4, 127) = " le dossier de configurations ne pourraient pas être
    trouvées ! "
1670     Literals(4, 128) = " vous n'avez pas défini le dossier de configurations ou il
    n'existe pas ! "
1671     Literals(4, 129) = " vous n'avez pas défini le dossier où les données envoyées
    seront sauvées ou elles n'existent pas ! "
1672     Literals(4, 130) = " vous n'avez pas défini le dossier où les données reçues
    seront sauvées ou elles n'existent pas ! "
1673     Literals(4, 131) = " vous n'avez pas défini le dossier dont le contenu vous
    pouvez envoyer ou il n'existe pas ! "
1674     Literals(4, 132) = " vous n'avez pas défini le dossier où les données de l
    'inertie de couple d'application seront sauvées ! "
1675     Literals(4, 133) = " les dossiers que vous sauvez des données d'entrée-sortie
    ou dont les données sont d'être envoient ne devraient pas être identiques ! "
1676     Literals(4, 134) = " vous ne pouvez pas employer le dossier de la
    configuration pour des données d'économie ! "

```

```

1677     Literals(4, 135) = " ne pourraient pas sauver des changements ! "
1678     Literals(4, 136) = " vous ne pouvez pas employer le dossier spécifique ! "
1679     Literals(4, 137) = " le dossier choisi est inadmissible ! "
1680     Literals(4, 138) = " erreur ! N'a pas pu ouvrir COM mettent en communication ! ⚡
"
1681     Literals(4, 139) = " graphiques d'exposition "
1682     Literals(4, 140) = " graphique 1 "
1683     Literals(4, 141) = " graphique 2 "
1684     Literals(4, 142) = " graphiques de peau "
1685     Literals(4, 143) = " maximisent des graphiques "
1686     Literals(4, 144) = " réduisent au minimum des graphiques "
1687     Literals(4, 145) = " communication périodique "
1688     Literals(4, 146) = " inertie de couple "
1689     Literals(4, 147) = " poussée de l'eau et commande de T/MN "
1690     Literals(4, 148) = " forme initiale "
1691     Literals(4, 149) = " poussée de l'eau et commande de T/MN "
1692     Literals(4, 150) = " raccordement "
1693     Literals(4, 151) = " forme étroite "
1694     Literals(4, 152) = " statut "
1695     Literals(4, 153) = " cycle de PWM (%) "
1696     Literals(4, 154) = " commande T/MN "
1697     Literals(4, 155) = " propriétés de graphiques "
1698     Literals(4, 156) = " Parametarize .NET Gaphs "
1699     Literals(4, 157) = " les paramètres premier graphique "
1700     Literals(4, 158) = " T/MN "
1701     Literals(4, 159) = " LoadCell (Kgr) "
1702     Literals(4, 160) = " ampères (a) "
1703     Literals(4, 161) = " Temps "
1704     Literals(4, 162) = " arrêt d'urgence "
1705     Literals(4, 163) = " immédiatement procédé d'arrêt ? "
1706     Literals(4, 164) = " économiser des données pour la poussée de l'eau d
'application et la commande de T/MN " ⚡
1707     Literals(4, 165) = " vous n'avez pas défini le dossier où les données de l'eau
d'application ont poussé et commande de T/MN sera sauvé ! " ⚡
1708     Literals(4, 166) = " les configurations disponibles classe "
1709     Literals(4, 167) = " choisi "
1710     Literals(4, 168) = " point de réglage T/MN "
1711     Literals(4, 169) = " tension "
1712     Literals(4, 170) = " graphiques d'exposition avec .NET "
1713     Literals(4, 171) = " graphiques d'exposition avec l'Excel "
1714     Literals(4, 172) = " deuxièmes paramètres du graphique "
1715     Literals(4, 173) = " Excel Gaphs de Parametarize "
1716     Literals(4, 174) = " numérique des points "
1717     Literals(4, 175) = " traits horizontaux "
1718     Literals(4, 176) = " lignes verticales "
1719     Literals(4, 177) = " divisions horizontales "
1720     Literals(4, 178) = " charge de PWM "
1721     Literals(4, 179) = " LoadCell "
1722     Literals(4, 180) = " tension "
1723     Literals(4, 181) = " ampères "
1724     Literals(4, 182) = " lignes tirées "
1725     Literals(4, 183) = " lignes solides "
1726     Literals(4, 184) = " type de lignes "
1727     Literals(4, 185) = " exposition après "
1728     Literals(4, 186) = " maintenant "
1729     Literals(4, 187) = " R.P.M "
1730     Literals(4, 188) = " ont placé R.P.M "
1731     Literals(4, 189) = " PWM "
1732     Literals(4, 190) = " ont placé PWM "
1733     Literals(4, 191) = " mouvement synchrone les deux graphiques "
1734     Literals(4, 192) = " propriétés du diagramme 1 "
1735     Literals(4, 193) = " propriétés du diagramme 2 "
1736     Literals(4, 194) = " ligne choisie : "
1737     Literals(4, 195) = " points de diagramme "
1738     Literals(4, 196) = " panneau de peau "
1739     Literals(4, 197) = " vous n'avez pas défini le dossier d'où le cycle du point
de réglage RPM/PWM (%) sera envoient ! " ⚡
1740     Literals(4, 198) = " créent le dossier pour envoyer des points de réglage "
1741     Literals(4, 199) = " mode choisi "
1742     Literals(4, 200) = " méthode pour la création de dossier "
1743     Literals(4, 201) = " manuellement "
1744     Literals(4, 202) = " de fonction "
1745     Literals(4, 203) = " importation à partir du dossier des textes (*.txt) "
1746     Literals(4, 204) = " écrivent des valeurs pour le T/MN : "
1747     Literals(4, 206) = " valeur "

```

```

1748     Literals(4, 207) = " valeurs de répétition "
1749     Literals(4, 208) = " temps de répétition : "
1750     Literals(4, 209) = " valeurs d'exposition dans le graphique "
1751     Literals(4, 210) = " fonction choisie "
1752     Literals(4, 211) = " valeur initiale de Temps : "
1753     Literals(4, 212) = " valeur finale de Temps : "
1754     Literals(4, 213) = " créent la fonction : "
1755     Literals(4, 214) = " fonction donnée par contrôle "
1756     Literals(4, 215) = " fonction choisie "
1757     Literals(4, 216) = " valeurs précédentes d'exposition 50 "
1758     Literals(4, 217) = " exposition après 50 valeurs ou ajoutent la nouvelle
page "
1759     Literals(4, 218) = " exposition 50 premières valeurs "
1760     Literals(4, 219) = " valeurs du bout 50 d'exposition "
1761     Literals(4, 220) = " première valeur de répétition "
1762     Literals(4, 221) = " dernière valeur de répétition "
1763     Literals(4, 222) = " exécutent la répétition "
1764     Literals(4, 223) = " il y avait une erreur dans les données d'entrée, comme en
la cellule "
1765     Literals(4, 224) = " opération échouent, il y avaient une erreur tandis que
les données économisantes dans le dossier "
1766     Literals(4, 225) = " période de temps "
1767     Literals(4, 226) = " page Goto "
1768     Literals(4, 227) = " page "
1769     Literals(4, 228) = " de "
1770     Literals(4, 229) = " les valeurs permises sont "
1771     Literals(4, 230) = " répétition de début de valeur "
1772     Literals(4, 231) = " les valeurs que vous avez données sont inadmissibles "
1773     Literals(4, 232) = " dossier choisi "
1774     Literals(4, 233) = " instructions générales "
1775     Literals(4, 234) = " le dossier que vous définirez doivent n'avoir seulement
les valeurs arithmétiques, aucune ligne vide et généralement aucun espace aussi
bien et comme si vous avez choisi la commande T/MN les valeurs incluses doivent
être dans la région [0 - 20000] tandis que si vous ont choisi et _ les valeurs de
cycle de PWM (%) doivent être dans la région [0 - 100.0]. " & vbCrLf & " Dès que
vous choisirez le dossier que vous voulez devez cliquer sur le bouton de
dossier de contrôle pour application vérifiera la compatibilité du dossier. Si le
dossier est compatible, puis la volonté de bouton de dossier de charge soit
permise. Si vous cliquez là-dessus les données du dossier passeront à l
'application supprimant tous précédemment et données insérées mais non sauvées. "
1776     Literals(4, 235) = " données de dossier "
1777     Literals(4, 236) = " dossier de charge "
1778     Literals(4, 237) = " il y avait une erreur tandis que le dossier de chargement
! "
1779     Literals(4, 238) = " nombre total des disques dans le dossier : "
1780     Literals(4, 239) = " dossier de contrôle "
1781     Literals(4, 240) = " vous devez le premier contrôle le dossier "
1782     Literals(4, 241) = " fonctions connues "
1783     Literals(4, 242) = " fonctions et périodes de temps choisies "
1784     Literals(4, 243) = " fonction de registre "
1785     Literals(4, 244) = "RESET"
1786     Literals(4, 245) = " symboles acceptables "
1787     Literals(4, 246) = " la fonction que vous avez dactylographiée est
inadmissible. Vérifiez l'étiquette acceptable de symboles pour voir quels symboles
sont acceptables. "
1788     Literals(4, 247) = " exécutent des fonctions "
1789     Literals(4, 248) = " les valeurs de Temps que vous avez fournies sont
inadmissibles ! "
1790     Literals(4, 249) = " vous avez déjà donné une autre fonction dans cette
période ! "
1791     Literals(4, 250) = " vous avez changé la fonction choisie ! "
1792     Literals(4, 251) = " il y avait une erreur tout en courant des fonctions
validées ! "
1793     Literals(4, 252) = " périodes de temps "
1794     Literals(4, 254) = " fonctionne "
1795     Literals(4, 255) = " symboles "
1796     Literals(4, 256) = " variables "
1797     Literals(4, 257) = " vous n'avez pas défini une fonction ! "
1798     Literals(4, 258) = " les valeurs venant des fonctions sont hors des limites !
"
1799     Literals(4, 259) = "Italien"
1800     Literals(4, 260) = "Espagnol"
1801     Literals(4, 261) = "Français"
1802     Literals(4, 262) = " Attendez svp pour charger des données. " & vbCrLf & " Si
vous ne souhaitez pas charger des données vous pouvez cliquer sur l'annulation de

```



```

1803     "" de bouton chargez le "" . "
1804     Literals(4, 263) = " Chargez le progrès "
1805     Literals(4, 264) = " Décommandez chargent "
1806     Literals(4, 265) = " Commencez "
1807     Literals(4, 266) = " Arrêtez "
1808     Literals(4, 267) = " Pause "
1809     Literals(4, 268) = " Résumé "
1810     Literals(4, 269) = " La valeur maximum de Temps que vous pouvez déclarer est "
1811     Literals(4, 270) = " verticale par mme. "
1812     Literals(4, 271) = " affichage se dirige "
1813     Literals(4, 272) = " aucun affichage de points "
1814     Literals(4, 273) = " affichage de valeurs moyennes par "
1815     Literals(4, 274) = " ont montré des points "
1816     Literals(4, 275) = " retour à la commande de dossier "
1817     Literals(4, 276) = " Access aux valeurs de données "
1818     Literals(4, 277) = " choix et accès aux valeurs "
1819     Literals(4, 278) = " ont choisi la valeur : "
1820     Literals(4, 279) = " permettent modifient : "
1821     Literals(4, 280) = " valeur : "
1822     Literals(4, 281) = " sortie sans économiser changements "
1823     Literals(4, 282) = " économiser les changements et la sortie "
1824     Literals(4, 283) = " ont remis à zéro toutes les valeurs à Oritzinal "
1825     Literals(4, 284) = " fonction choisie de sortie "
1826     Literals(4, 285) = " fichier graphique dans le format graphique "
1827     Literals(4, 286) = " le dossier initial inadmissible, vérifient votre dossier "
1828     Literals(4, 287) = " dossier non validé "
1829     Literals(4, 288) = " neutralisent modifient "
1830     Literals(4, 289) = " affichage choisi de valeur moyenne moins que < 2"
1831     Literals(4, 290) = " aucun changement "
1832     Literals(4, 291) = " changements fait, serrent oui pour sauver des changements
1833     et pour remplacer les valeurs comme dossier, pour ne serrer aucun écart tous les
1834     changements et pour employer les valeurs initiales , l'annulation pour ne pas
1835     sortir de la forme "
1836     Literals(4, 292) = " valeur de restauration "
1837     Literals(4, 293) = " restauration toutes les valeurs "
1838     Literals(4, 294) = " dossier n'est pas type de *.txt " & Chr(13) &
1839     "" choisissent seulement dossier de *.txt "
1840     Literals(4, 295) = " erreur de choix de dossier "
1841     Literals(4, 296) = " erreur sur le dossier de lecture "
1842     Literals(4, 297) = " extrémité de dossier "
1843     Literals(4, 298) = " choisissent le dossier de transmission "
1844     Literals(4, 299) = " transmettent des Setpoints de dossier "
1845     Literals(4, 300) = " transmettent des Setpoints de commandes "
1846     Literals(4, 301) = " régènèrent Période des graphiques "
1847     Literals(4, 302) = " chaque nouvelle valeur "
1848     Literals(4, 303) = " période de base de graphiques : "
1849     Literals(4, 304) = " période de graphiques : "
1850     Literals(4, 305) = " synchronisation de graphiques "
1851     Literals(4, 306) = " le dossier n'a pas été appropriée composées "
1852     Literals(4, 307) = " est déjà pression choisie de dossier : " & Chr(13) &
1853     "" correct pour transmettre lui ou " & Chr(13) & " annulation pour choisir un neuf
1854     "
1855     Literals(4, 308) = " vous avez atteint le délai Maximum, par conséquent vous
1856     ne pouvez pas insérer la nouvelle page ! "
1857     Literals(4, 309) = " répétition ne peuvent pas être exécutées puisque le délai
1858     Maximum sera atteint ! "
1859     Literals(4, 310) = " chargent seront accomplies approximativement dans "
1860     Literals(4, 311) = " graphiques »"
1861     Literals(4, 312) = " vous ne pouvez pas modifier le dossier de configuration
1862     de base ! "
1863     Literals(4, 313) = " choisissent le graphique de défaut "
1864     Literals(4, 314) = " graphique de .NET »"
1865     Literals(4, 315) = " graphique d'Excel »"
1866     'Coquilles additionnelles de poussée
1867     Literals(4, 316) = " la carte est sans risque clôturées mais une erreur
1868     inattendue dans des démarches de fermeture. Pour le prochain raccordement REMIS À
1869     ZÉRO la carte. "
1870     Literals(4, 317) = " les paramètres initiés de la carte n'ont pas les valeurs
1871     appropriées, ont remis à zéro la carte du bouton de remise là-dessus svp. "
1872     Literals(4, 318) = " ont endommagé le dossier ou les valeurs de lecture ne
1873     sont pas en limites appropriées. "
1874     Literals(4, 319) = " voulez-vous que le moteur démarre-t-il ? "
1875     Literals(4, 320) = " porte série étroite et rebranchent. La carte est fermée.

```

```

1862     Literals(4, 321) = " Permettez le moteur "
1863     Literals(4, 322) = " Arrêt des données reçues à la porte série "
1864     Literals(4, 323) = " Avant que proceed à n'importe quel travail choisissez un
        valide économiser le dossier. " & ChrW(10) & ChrW(13) & " Le choix du dossier
        dégagera n'importe quel travail au temps ! "
1865     Literals(4, 324) = " Setpoints de dossier de Transmitting "
1866 End Sub
1867
1868 Public Function IsValidFile(ByVal MyFileName As String) As Boolean
1869     If MyFileName = "" Then
1870         Return False
1871     Else
1872         Dim CheckFile As New FileInfo(MyFileName)
1873         If CheckFile.Exists Then
1874             Return True
1875         Else
1876             Return False
1877         End If
1878     End If
1879 End Function
1880
1881 Public Function ConvertMilsToSecsAndMils(ByVal Mils As Long) As String
1882     Dim StringToReturn As String
1883
1884     StringToReturn = Format(Int(Mils / 1000), "000000") & "," & Format(Int(Mils -
1885     1000 * Int(Mils / 1000)), "000")
1886
1887     Return StringToReturn
1888 End Function
1889
1890 Public Function ConvertMilsToMinsAndSecsAndMils(ByVal Mils As Long) As String
1891     Dim StringToReturn As String
1892
1893     StringToReturn = Str(Int(Mils / 60000)) & ":" &
1894     IIf(Int((Mils - 60000 * Int(Mils / 60000)) / 1000) < 10, "0",
1895     "") & _
1896     Trim(Str(Int((Mils - 60000 * Int(Mils / 60000)) / 1000))) & "
1897     ":" & _
1898     IIf(Mils - 60000 * Int(Mils / 60000) - 1000 * Int((Mils -
1899     60000 * Int(Mils / 60000)) / 1000) < 10, "00",
1900     IIf(Mils - 60000 * Int(Mils / 60000) - 1000 * Int((Mils -
1901     60000 * Int(Mils / 60000)) / 1000) < 100, "0", "")) & _
1902     Trim(Str(Mils - 60000 * Int(Mils / 60000) - 1000 * Int((Mils
1903     - 60000 * Int(Mils / 60000)) / 1000)))
1904     Return StringToReturn
1905 End Function
1906 Public Function ConvertMilstoHoursMintsSecs(ByVal Mils As Long) As String
1907     Dim StringToReturn As String
1908     Dim MilliSeconds As Long = Mils
1909
1910     MyConnectionTime.wHour = Int(MilliSeconds / 3600000)
1911     MyConnectionTime.wMinute = Int((MilliSeconds - (MyConnectionTime.wHour *
1912     3600000)) / 60000)
1913     MyConnectionTime.wSecond = Int((MilliSeconds - (MyConnectionTime.wHour *
1914     3600000) - (MyConnectionTime.wMinute * 60000)) / 1000)
1915     MyConnectionTime.wMilliseconds = Int((MilliSeconds - (MyConnectionTime.wHour *
1916     3600000) - (MyConnectionTime.wMinute * 60000) - (MyConnectionTime.wSecond *
1917     1000)) / 1000)
1918     StringToReturn = Format(MyConnectionTime.wHour, "00") & ":" & Format
1919     (MyConnectionTime.wMinute, "00") & ":" & Format(MyConnectionTime.wSecond, "00") '&
1920     ":" & Str(MyConnectionTime.wMilliseconds)
1921     Return StringToReturn
1922 End Function
1923 Public Sub InitCountTimeofconnection(ByVal time As System.DateTime)
1924     MyStartTime = time
1925     stTimeConnection = "00:00:00"
1926 End Sub
1927 Public Sub CountTimeOfConnection(ByVal time As System.DateTime)
1928     CurrentTime = time
1929
1930     Dim MilsofTime As Long = (CurrentTime.Ticks - MyStartTime.Ticks) / (10 ^ 4)
1931     If MyStartTime.Ticks > 0 Then
1932         'Dim stTimeConnection As String = ConvertMilstoHoursMintsSecs(MilsofTime)
1933         stTimeConnection = ConvertMilstoHoursMintsSecs(MilsofTime)

```

```

1923     End If
1924 End Sub
1925
1926 Public Function FindExcelProcesses() As Integer()
1927     Dim Proc, Processes() As Process
1928     Dim i, j, Dimension As Integer
1929
1930     Dimension = NumOfExcelProcs()
1931     Dim Matrix(Dimension) As Integer
1932     i = 0
1933     Processes = Process.GetProcessesByName("EXCEL")
1934     For Each Proc In Processes
1935         Matrix(i) = Proc.Id
1936         i += 1
1937     Next
1938
1939     For j = i To (Dimension - 1)
1940         Matrix(j) = 0
1941     Next
1942
1943     Return Matrix
1944 End Function
1945
1946 Public Function NumOfExcelProcs() As Integer
1947     Dim Processes() As Process
1948
1949     Processes = Process.GetProcessesByName("EXCEL")
1950     Return Processes.Length
1951 End Function
1952
1953 Public Sub AddFileHeader(ByVal FileName As String, ByVal WhichApplication As Integer)
1954     Dim Text2Write As String = ""
1955     Dim CheckFile As New IO.FileInfo(FileName)
1956     Dim sw As StreamWriter
1957     Dim MyString As String = Now()
1958
1959     If CheckFile.Exists Then
1960         Select Case WhichApplication
1961             Case 1 'RS232 RECEIVING DATA
1962                 Text2Write = "---- RS232 RECEIVING DATA ----"
1963             Case 2 'RS232 SENDING DATA
1964                 Text2Write = "----- RS232 SENDING DATA -----"
1965             Case 3
1966                 Text2Write = "----- TORQUE INERTIA -----"
1967             Case 4
1968                 Text2Write = " WATER THRUST AND RPM CONTROL "
1969         End Select
1970
1971         sw = CheckFile.AppendText()
1972         sw.WriteLine("")
1973         sw.WriteLine("*****
1974     sw.WriteLine("*****" &
1975 Text2Write & "*****") '126
1976     sw.WriteLine("*****
1977     sw.WriteLine("*****
1978
1979     If WhichApplication = 4 Then
1980         sw.WriteLine("      RPM      |*| Control Reg |*| Loadcell |*|
Voltage |*| Amperes (A) |*| S.P. RPM |*| S.P. Con Reg |*| Time ")
1981         sw.WriteLine("-----")
1982     End If
1983
1984     sw.Close()
1985     sw = Nothing
1986 Else
1987     ' Inform for missing file

```

```

1988         End If
1989
1990     CheckFile = Nothing
1991 End Sub
1992
1993 Public Sub AddFileFooter(ByVal FileName As String, ByVal WhichApplication As Integer)
1994     Dim Text2Write As String = ""
1995     Dim CheckFile As New IO.FileInfo(FileName)
1996     Dim sw As StreamWriter
1997     Dim MyString As String = Now()
1998
1999     If CheckFile.Exists Then
2000         Select Case WhichApplication
2001             Case 1 'RS232 RECEIVING DATA
2002                 Text2Write = "---- END OF RS232 RECEIVING DATA ----"
2003             Case 2 'RS232 SENDING DATA
2004                 Text2Write = "----- END OF RS232 SENDING DATA -----"
2005             Case 3
2006                 Text2Write = "----- END OF TORQUE INERTIA -----"
2007             Case 4
2008                 Text2Write = " END OF WATER THRUST AND RPM CONTROL "
2009         End Select
2010
2011         sw = CheckFile.AppendText()
2012         sw.WriteLine("")
2013         sw.WriteLine(
2014             "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2015             XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" &
2016             Text2Write & "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" &
2017             sw.WriteLine("XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2018             " & MyString.PadRight(22, " ") & "
2019             XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" &
2020             sw.WriteLine(
2021             "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2022             XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" &
2023             sw.WriteLine("")
2024             sw.Close()
2025             sw = Nothing
2026     Else
2027         ' Error ocured but shall be ignored
2028     End If
2029     CheckFile = Nothing
2030 End Sub
2031
2032 Public Function FindMyExcelProcess(ByVal NumOfExistingExcelProcs As Integer, ByVal RunningExcelProcs() As Integer) As Integer
2033     Dim Proc, Processes() As Process
2034     Dim i As Integer
2035     Dim WasRunning As Boolean
2036
2037     Processes = Process.GetProcessesByName("EXCEL")
2038     For Each Proc In Processes
2039         WasRunning = False
2040         For i = 0 To (NumOfExistingExcelProcs - 1)
2041             If RunningExcelProcs(i) = Proc.Id Then
2042                 WasRunning = True
2043             End If
2044         Next
2045         If WasRunning = False Then
2046             Return Proc.Id
2047         End If
2048     Next
2049     Return 0 ' Should not come to here
2050 End Function
2051
2052 Public Sub KillExcelProc(ByVal ProcessAlreadyClosed As Boolean, ByVal CreatedExcelProcID As Integer)
2053     If Not ProcessAlreadyClosed Then
2054         Dim Proc As Process
2055
2056         If CreatedExcelProcID = 0 Then 'Something went wrong
2057             Exit Sub
2058         End If
2059         Proc = Process.GetProcessById(CreatedExcelProcID)

```

```

2053         Proc.Kill()
2054         ProcessAlreadyClosed = True
2055     End If
2056 End Sub
2057
2058 Public Function VerifyExcelExistance() As Boolean
2059     ' This function is used to identify if excel is already installed and returns
yes if there is and no otherwise
2060     Dim ExcelExists As Boolean
2061     Dim NumOfExistingExcelProcs, CreatedExcelProcID As Integer
2062     NumOfExistingExcelProcs = NumOfExcelProcs()
2063     Dim RunningExcelProcs(NumOfExistingExcelProcs) As Integer
2064     RunningExcelProcs = FindExcelProcesses()
2065
2066     Try
2067         Dim app As New Microsoft.Office.Interop.Excel.Application
2068         Dim Doc As Microsoft.Office.Interop.Excel.Workbook
2069         Dim sheet As Microsoft.Office.Interop.Excel.Worksheet
2070         Dim Chart1, Chart2 As Microsoft.Office.Interop.Excel.Chart
2071
2072         app.Visible = False
2073         Dim oldCI As System.Globalization.CultureInfo = System.Threading.Thread.
CurrentThread.CurrentCulture
2074         System.Threading.Thread.CurrentCulture = New System.
Globalization.CultureInfo("en-US")
2075         ExcelExists = True
2076         CreatedExcelProcID = FindMyExcelProcess(NumOfExistingExcelProcs,
RunningExcelProcs)
2077         KillExcelProc(False, CreatedExcelProcID)
2078     Catch
2079         ExcelExists = False
2080     End Try
2081
2082     Return ExcelExists
2083 End Function
2084 Public Sub Check_if_Timeout_Serialport(ByVal MilsofTimeout As Long, ByVal
WhichLang As Integer)
2085     If ((Now.Ticks - LastRecevedDataTime) > (MilsofTimeout * (10 ^ 4))) And
messageaccepted = True Then
2086         LastRecevedDataTime = Now.Ticks
2087         messageaccepted = False
2088         msgTimeoutOkPressed = MsgBox(GetStr(WhichLang, 322), MsgBoxStyle.OkOnly,
GetStr(WhichLang, 109))
2089         If msgTimeoutOkPressed = DialogResult.OK Then
2090             messageaccepted = True
2091             Refresh_Time_of_Receved_Data_for_Timeout() ' dellay next Show of
message
2092         End If
2093     ElseIf messageaccepted = False And Not msgTimeoutOkPressed = DialogResult.OK
Then
2094     End If
2095 End Sub
2096 Public Sub Refresh_Time_of_Receved_Data_for_Timeout()
2097     LastRecevedDataTime = Now.Ticks
2098 End Sub
2099 Public Sub SetPointFileHeader(ByRef sw As IO.StreamWriter, ByVal IsRPM As Boolean,
ByVal WhichMethod As String, ByVal WhichFunction As String)
2100     If WhichFunction.Length > 5000 Then
2101         ' txt's file line length may maximum be 5000 chars so:
2102         WhichFunction = ""
2103     End If
2104     Dim DashesForFunction As New String("-"c, IIf(Trim(WhichFunction).Length <= 19
, 34, IIf(Trim(WhichFunction).Length < 53, 53 - Trim(WhichFunction).Length, 0)))
2105     Dim EmptysForFunction As New String(" "c, IIf(Trim(WhichFunction).Length >= 19
, 0, 19 - WhichFunction.Length))
2106     Dim Dashes As New String("-"c, 34)
2107     Dim AndTheFunctionIs As New String(" "c, 19)
2108     Dim Asterisks As New String("*"c, 107)
2109     Dim MyHeaderText As String = "          This is the Set Point File          "
2110     Dim AndTheMethodIs As String = ""
2111
2112     Select Case WhichMethod
2113     Case "0"
2114         AndTheMethodIs = " Manually          "
2115     Case "1"

```

```

2116         AndTheMethodIs = " With Function      "
2117         Case "2"
2118             AndTheMethodIs = " Import from File  "
2119     End Select
2120
2121     sw.WriteLine(Asterisks)
2122     sw.WriteLine("*" & Dashes & MyHeaderText & Dashes & "*")
2123     sw.WriteLine("*" & Dashes & " Mode           : " & IIf(IsRPM, " Control RPM   "
2124     ", " PWM Cycle(%)           ") & Dashes & "*")
2125     sw.WriteLine("*" & Dashes & " Creation Method: " & AndTheMethodIs & Dashes &
2126     "*")
2127     sw.WriteLine("*" & Dashes & " The Function is: " & WhichFunction &
2128     "EmptysForFunction & DashesForFunction & "*")
2129     sw.WriteLine(Asterisks)
2130 End Sub
2131
2132 Public Sub KillMeIfMySisterIsRunning()
2133     'This subroutine checks if there is another application already running and
2134     'terminates the new one
2135     Dim Proc(), MyProc As Process, ModuleName, ProcName As String
2136
2137     MyProc = Process.GetCurrentProcess
2138     ModuleName = Process.GetCurrentProcess.MainModule.ModuleName
2139     ProcName = System.IO.Path.GetFileNameWithoutExtension(ModuleName)
2140
2141     Proc = Process.GetProcessesByName(ProcName)
2142     If Proc.Length > 1 Then
2143         MyProc.Kill()
2144     End If
2145 End Sub
2146
2147 Public Function EstimateRemainingTime(ByVal now As System.DateTime, ByVal
2148 Percentage As Int16) As String
2149
2150     Select Case Percentage
2151     Case Is >= 1
2152         EstimateTimeMat(1) = now
2153         Dim RemainingMils As Int64 = (((100 - Percentage) / Percentage) *
2154 (EstimateTimeMat(1).Ticks - EstimateTimeMat(0).Ticks)) * 0.0001
2155
2156         Return Me.ConvertMilstoHoursMintsSecs(RemainingMils)
2157     Case 0
2158         EstimateTimeMat(0) = now
2159         Return "--:--:--"
2160     End Select
2161     Return ""
2162 End Function
2163
2164 Public Sub CreateAndDeleteFileIfExists(ByVal MyFileName As String)
2165     Dim CheckFile As New FileInfo(MyFileName)
2166
2167     If CheckFile.Exists Then
2168         CheckFile.Delete()
2169     End If
2170     CheckFile.Create()
2171 End Sub
2172
2173 Public Function FindParamValue(ByVal MyConfFile As String, ByVal MyParam As String
2174 , ByVal WhichIsTheLang As Integer) As String
2175     Dim MyString As String = "error", line As String, MyPosition As Integer
2176     Dim WasFound As Boolean = False
2177
2178     If IsValidFile(MyConfFile) Then
2179         Try
2180             Dim sr As IO.StreamReader = New IO.StreamReader(MyConfFile)
2181
2182             line = sr.ReadLine()
2183             Do While (Not (line Is Nothing)) And WasFound = False
2184                 MyPosition = line.IndexOf("=")
2185                 If MyPosition <> -1 Then
2186                     If Trim(line.Substring(vbTab.Length, MyPosition - 1)) = Trim
2187 (MyParam) Then
2188                         WasFound = True
2189                         MyString = Trim(line.Substring(MyPosition + 1, Trim(line).
2190 Length - MyPosition - 1))

```

```

2182             End If
2183         End If
2184         line = sr.ReadLine()
2185     Loop
2186     sr.Close()
2187     Catch E As Exception
2188         MsgBox(GetStr(WhichIsTheLang, 127), MsgBoxStyle.Information, GetStr
(WhichIsTheLang, 295))
2189         MyString = "error"
2190     End Try
2191     Else
2192         MsgBox(GetStr(WhichIsTheLang, 127), MsgBoxStyle.Information, GetStr
(WhichIsTheLang, 295))
2193     End If
2194     Return MyString
2195 End Function
2196
2197 Public Sub ToOpenAFileDialog(ByRef MyObject As TextBox, ByVal WhichIsTheLang As
Integer, ByVal InitialDirectory As String)
2198     Dim myStream As Stream
2199     Dim openFileDialog1 As New OpenFileDialog
2200     openFileDialog1.InitialDirectory = InitialDirectory
2201     openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*"
2202     openFileDialog1.FilterIndex = 1
2203     openFileDialog1.RestoreDirectory = True
2204     openFileDialog1.Multiselect = False
2205
2206     If openFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK And
openFileDialog1.FileName.EndsWith(".txt") Then
2207         Try
2208             myStream = openFileDialog1.OpenFile()
2209             If Not (myStream Is Nothing) Then
2210                 MyObject.Text = openFileDialog1.FileName
2211                 myStream.Close()
2212             End If
2213         Catch
2214             MsgBox(GetStr(WhichIsTheLang, 136), 0, GetStr(WhichIsTheLang, 109))
2215         End Try
2216     End If
2217 End Sub
2218
2219 Public Sub CheckAndRepairConfFileForConsistency(ByVal MyFileName As String)
2220
2221 End Sub
2222 End Class
2223

```

```

1 Imports System.Text
2 Imports System.IO
3
4 Public Class Options
5     Inherits System.Windows.Forms.Form
6
7     Private MyClassCode As New MyPtyxClass
8     Public WhichIsTheLang As Integer
9     Public IntCommPort, IntBaudRate, IntParityBits, IntDataBits, IntStopBits As Integer
10    Public bGo As Boolean
11
12 #Region " Windows Form Designer generated code "
13
14    Public Sub New()
15        MyBase.New()
16
17        'This call is required by the Windows Form Designer.
18        InitializeComponent()
19
20        'Add any initialization after the InitializeComponent() call
21
22    End Sub
23
24    'Form overrides dispose to clean up the component list.
25    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
26        If disposing Then
27            If Not (components Is Nothing) Then
28                components.Dispose()
29            End If
30        End If
31        MyBase.Dispose(disposing)
32    End Sub
33
34    'Required by the Windows Form Designer
35    Private components As System.ComponentModel.IContainer
36
37    'NOTE: The following procedure is required by the Windows Form Designer
38    'It can be modified using the Windows Form Designer.
39    'Do not modify it using the code editor.
40    Friend WithEvents GroupBox1 As System.Windows.Forms.GroupBox
41    Friend WithEvents LblComPort As System.Windows.Forms.Label
42    Friend WithEvents LblBaudRate As System.Windows.Forms.Label
43    Friend WithEvents LblParity As System.Windows.Forms.Label
44    Friend WithEvents LblDataBits As System.Windows.Forms.Label
45    Friend WithEvents LblStopBits As System.Windows.Forms.Label
46    Friend WithEvents CmbComPort As System.Windows.Forms.ComboBox
47    Friend WithEvents CmbBaudRate As System.Windows.Forms.ComboBox
48    Friend WithEvents CmbDataBits As System.Windows.Forms.ComboBox
49    Friend WithEvents CmbParity As System.Windows.Forms.ComboBox
50    Friend WithEvents CmbStopBits As System.Windows.Forms.ComboBox
51    Friend WithEvents BtOk As System.Windows.Forms.Button
52    Friend WithEvents BtCancel As System.Windows.Forms.Button
53    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
54        Dim resources As System.ComponentModel.ResourceManager = New System.
55        ComponentModel.ComponentResourceManager(GetType(Options))
56        Me.LblComPort = New System.Windows.Forms.Label
57        Me.CmbComPort = New System.Windows.Forms.ComboBox
58        Me.CmbBaudRate = New System.Windows.Forms.ComboBox
59        Me.LblBaudRate = New System.Windows.Forms.Label
60        Me.GroupBox1 = New System.Windows.Forms.GroupBox
61        Me.CmbDataBits = New System.Windows.Forms.ComboBox
62        Me.LblParity = New System.Windows.Forms.Label
63        Me.CmbParity = New System.Windows.Forms.ComboBox
64        Me.LblDataBits = New System.Windows.Forms.Label
65        Me.CmbStopBits = New System.Windows.Forms.ComboBox
66        Me.LblStopBits = New System.Windows.Forms.Label
67        Me.BtOk = New System.Windows.Forms.Button
68        Me.BtCancel = New System.Windows.Forms.Button
69        Me.SuspendLayout()
70        '
71        'LblComPort
72        '
73        Me.LblComPort.AutoSize = True
74        Me.LblComPort.Location = New System.Drawing.Point(7, 8)
75        Me.LblComPort.Name = "LblComPort"

```



```

75     Me.LblComPort.Size = New System.Drawing.Size(53, 13)
76     Me.LblComPort.TabIndex = 0
77     Me.LblComPort.Text = "COM Port"
78     '
79     'CmbComPort
80     '
81     Me.CmbComPort.Anchor = CType((System.Windows.Forms.AnchorStyles.Top Or System.
Windows.Forms.AnchorStyles.Right), System.Windows.Forms.AnchorStyles)
82     Me.CmbComPort.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList
83     Me.CmbComPort.Items.AddRange(New Object() {"COM1", "COM2", "COM3", "COM4",
"COM5", "COM6", "COM7", "COM8", "COM9", "COM10"})
84     Me.CmbComPort.Location = New System.Drawing.Point(92, 5)
85     Me.CmbComPort.Name = "CmbComPort"
86     Me.CmbComPort.Size = New System.Drawing.Size(66, 21)
87     Me.CmbComPort.TabIndex = 1
88     '
89     'CmbBaudRate
90     '
91     Me.CmbBaudRate.Anchor = System.Windows.Forms.AnchorStyles.Right
92     Me.CmbBaudRate.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList
93     Me.CmbBaudRate.Items.AddRange(New Object() {"110", "300", "1200", "2400", "4800",
", "9600", "19200", "38400", "57600", "115200", "460800", "921600"})
94     Me.CmbBaudRate.Location = New System.Drawing.Point(92, 40)
95     Me.CmbBaudRate.Name = "CmbBaudRate"
96     Me.CmbBaudRate.Size = New System.Drawing.Size(74, 21)
97     Me.CmbBaudRate.TabIndex = 3
98     '
99     'LblBaudRate
100    '
101    Me.LblBaudRate.AutoSize = True
102    Me.LblBaudRate.Location = New System.Drawing.Point(7, 43)
103    Me.LblBaudRate.Name = "LblBaudRate"
104    Me.LblBaudRate.Size = New System.Drawing.Size(53, 13)
105    Me.LblBaudRate.TabIndex = 2
106    Me.LblBaudRate.Text = "Baud rate"
107    '
108    'GroupBox1
109    '
110    Me.GroupBox1.Location = New System.Drawing.Point(1, 29)
111    Me.GroupBox1.Name = "GroupBox1"
112    Me.GroupBox1.Size = New System.Drawing.Size(266, 8)
113    Me.GroupBox1.TabIndex = 4
114    Me.GroupBox1.TabStop = False
115    '
116    'CmbDataBits
117    '
118    Me.CmbDataBits.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList
119    Me.CmbDataBits.Items.AddRange(New Object() {"5", "6", "7", "8"})
120    Me.CmbDataBits.Location = New System.Drawing.Point(92, 91)
121    Me.CmbDataBits.Name = "CmbDataBits"
122    Me.CmbDataBits.Size = New System.Drawing.Size(74, 21)
123    Me.CmbDataBits.TabIndex = 8
124    '
125    'LblParity
126    '
127    Me.LblParity.AutoSize = True
128    Me.LblParity.Location = New System.Drawing.Point(7, 68)
129    Me.LblParity.Name = "LblParity"
130    Me.LblParity.Size = New System.Drawing.Size(33, 13)
131    Me.LblParity.TabIndex = 5
132    Me.LblParity.Text = "Parity"
133    '
134    'CmbParity
135    '
136    Me.CmbParity.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList
137    Me.CmbParity.Items.AddRange(New Object() {"Even", "Odd", "None", "Mark"})
138    Me.CmbParity.Location = New System.Drawing.Point(92, 65)
139    Me.CmbParity.Name = "CmbParity"
140    Me.CmbParity.Size = New System.Drawing.Size(74, 21)
141    Me.CmbParity.TabIndex = 6
142    '
143    'LblDataBits
144    '
145    Me.LblDataBits.AutoSize = True
146    Me.LblDataBits.Location = New System.Drawing.Point(7, 94)

```

```

147     Me.LblDataBits.Name = "LblDataBits"
148     Me.LblDataBits.Size = New System.Drawing.Size(49, 13)
149     Me.LblDataBits.TabIndex = 7
150     Me.LblDataBits.Text = "Data bits"
151     '
152     'CmbStopBits
153     '
154     Me.CmbStopBits.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList
155     Me.CmbStopBits.Items.AddRange(New Object() {"1", "2"})
156     Me.CmbStopBits.Location = New System.Drawing.Point(92, 116)
157     Me.CmbStopBits.Name = "CmbStopBits"
158     Me.CmbStopBits.Size = New System.Drawing.Size(74, 21)
159     Me.CmbStopBits.TabIndex = 10
160     '
161     'LblStopBits
162     '
163     Me.LblStopBits.AutoSize = True
164     Me.LblStopBits.Location = New System.Drawing.Point(7, 119)
165     Me.LblStopBits.Name = "LblStopBits"
166     Me.LblStopBits.Size = New System.Drawing.Size(48, 13)
167     Me.LblStopBits.TabIndex = 9
168     Me.LblStopBits.Text = "Stop bits"
169     '
170     'BtOk
171     '
172     Me.BtOk.Image = CType(resources.GetObject("BtOk.Image"), System.Drawing.Image)
173     Me.BtOk.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft
174     Me.BtOk.Location = New System.Drawing.Point(168, 46)
175     Me.BtOk.Name = "BtOk"
176     Me.BtOk.Size = New System.Drawing.Size(88, 39)
177     Me.BtOk.TabIndex = 11
178     Me.BtOk.Text = "OK"
179     Me.BtOk.TextAlign = System.Drawing.ContentAlignment.MiddleRight
180     '
181     'BtCancel
182     '
183     Me.BtCancel.DialogResult = System.Windows.Forms.DialogResult.Cancel
184     Me.BtCancel.Image = CType(resources.GetObject("BtCancel.Image"), System.Drawing
185     .Image)
186     Me.BtCancel.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft
187     Me.BtCancel.Location = New System.Drawing.Point(168, 97)
188     Me.BtCancel.Name = "BtCancel"
189     Me.BtCancel.Size = New System.Drawing.Size(88, 39)
190     Me.BtCancel.TabIndex = 12
191     Me.BtCancel.Text = "Cancel"
192     Me.BtCancel.TextAlign = System.Drawing.ContentAlignment.MiddleRight
193     '
194     'Options
195     '
196     Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
197     Me.CancelButton = Me.BtCancel
198     Me.ClientSize = New System.Drawing.Size(259, 144)
199     Me.Controls.Add(Me.BtOk)
200     Me.Controls.Add(Me.CmbStopBits)
201     Me.Controls.Add(Me.LblStopBits)
202     Me.Controls.Add(Me.CmbParity)
203     Me.Controls.Add(Me.LblDataBits)
204     Me.Controls.Add(Me.CmbDataBits)
205     Me.Controls.Add(Me.LblParity)
206     Me.Controls.Add(Me.GroupBox1)
207     Me.Controls.Add(Me.CmbBaudRate)
208     Me.Controls.Add(Me.LblBaudRate)
209     Me.Controls.Add(Me.CmbComPort)
210     Me.Controls.Add(Me.LblComPort)
211     Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle
212     Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)
213     Me.MaximizeBox = False
214     Me.MinimizeBox = False
215     Me.Name = "Options"
216     Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent
217     Me.Text = "Options"
218     Me.ResumeLayout(False)
219     Me.PerformLayout()
220

```

```

221     End Sub
222
223 #End Region
224
225     Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles BtCancel.Click
226         bGo = False
227         Me.Hide()
228     End Sub
229
230     Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles BtOk.Click
231         If CmbComPort.SelectedIndex = -1 Then
232             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 114), MsgBoxStyle.Exclamation,
MyClassCode.GetStr(WhichIsTheLang, 109))
233             CmbComPort.Focus()
234             Exit Sub
235         End If
236         If CmbBaudRate.SelectedIndex = -1 Then
237             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 115), MsgBoxStyle.Exclamation,
MyClassCode.GetStr(WhichIsTheLang, 109))
238             CmbBaudRate.Focus()
239             Exit Sub
240         End If
241         If CmbParity.SelectedIndex = -1 Then
242             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 116), MsgBoxStyle.Exclamation,
MyClassCode.GetStr(WhichIsTheLang, 109))
243             CmbParity.Focus()
244             Exit Sub
245         End If
246         If CmbDataBits.SelectedIndex = -1 Then
247             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 117), MsgBoxStyle.Exclamation,
MyClassCode.GetStr(WhichIsTheLang, 109))
248             CmbDataBits.Focus()
249             Exit Sub
250         End If
251         If CmbStopBits.SelectedIndex = -1 Then
252             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 118), MsgBoxStyle.Exclamation,
MyClassCode.GetStr(WhichIsTheLang, 109))
253             CmbStopBits.Focus()
254             Exit Sub
255         End If
256         bGo = True
257         FillPubs()
258         Me.Hide()
259     End Sub
260
261     Private Sub CheckConfsDefaults(ByVal MyFileName As String, ByVal WhichLanguage As
Integer)
262         Try : Dim CheckFile As New FileInfo(MyFileName)
263
264
265             If (Not CheckFile.Exists) Then
266                 CreateConfsFile("c:\windows\MyConfs.ini")
267             End If
268
269             CmbComPort.SelectedIndex = Val(MyClassCode.FindParamValue(MyFileName, "COM
Port", WhichLanguage))
270             CmbBaudRate.SelectedIndex = Val(MyClassCode.FindParamValue(MyFileName,
"Baud Rate", WhichLanguage))
271             CmbParity.SelectedIndex = Val(MyClassCode.FindParamValue(MyFileName,
"Parity", WhichLanguage))
272             CmbDataBits.SelectedIndex = Val(MyClassCode.FindParamValue(MyFileName,
>Data Bits", WhichLanguage))
273             CmbStopBits.SelectedIndex = Val(MyClassCode.FindParamValue(MyFileName,
"Stop Bits", WhichLanguage))
274         Catch E As Exception
275             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 119), 0)
276         End Try
277     End Sub
278     Public Sub CreateConfsFile(ByVal MyFileName As String)
279
280         Dim CheckFile As New FileInfo(MyFileName)
281
282         Try

```

```

283         If (Not CheckFile.Exists) Then
284             Dim fParams As New FrmParameters
285             fParams.CreateConfsFile(MyFileName)
286         End If
287     Catch E As Exception
288         MsgBox(MyClassCode.GetStr(WhichIsTheLang, 123), 0)
289     End Try
290 End Sub
291
292 Public Sub WhichLang(ByVal WhichLanguage As Integer)
293     WhichIsTheLang = WhichLanguage
294
295     LblComPort.Text = MyClassCode.GetStr(WhichLanguage, 72)
296     LblBaudRate.Text = MyClassCode.GetStr(WhichLanguage, 73)
297     LblParity.Text = MyClassCode.GetStr(WhichLanguage, 74)
298     LblDataBits.Text = MyClassCode.GetStr(WhichLanguage, 75)
299     LblStopBits.Text = MyClassCode.GetStr(WhichLanguage, 76)
300     BtOk.Text = MyClassCode.GetStr(WhichLanguage, 77)
301     BtCancel.Text = MyClassCode.GetStr(WhichLanguage, 78)
302     Me.Text = MyClassCode.GetStr(WhichLanguage, 79)
303
304     ComboContents(WhichLanguage)
305 End Sub
306
307 Public Sub SetDefaults(ByVal MyFileName As String, ByVal WhichLanguage As Integer,
308 ByVal UseDefaultConfFile As Boolean)
309     AppropriatelyFillComCombo()
310     WhichLang(WhichLanguage)
311     If UseDefaultConfFile Then
312         CheckConfsDefaults(MyFileName, WhichLanguage)
313     End If
314 End Sub
315 Public Sub ComboContents(ByVal WhichLanguage As Integer)
316     CmbParity.Items.Clear()
317
318     CmbParity.Items.Add(MyClassCode.GetStr(WhichLanguage, 120))
319     CmbParity.Items.Add(MyClassCode.GetStr(WhichLanguage, 121))
320     CmbParity.Items.Add(MyClassCode.GetStr(WhichLanguage, 122))
321     CmbParity.Items.Add("Mark")
322
323     CmbParity.SelectionStart = 1
324 End Sub
325 Public Sub AppropriatelyFillComCombo()
326     Dim i As Integer
327
328     Me.CmbComPort.Items.Clear()
329     For i = 0 To 9
330         If mComs(i) Then
331             Me.CmbComPort.Items.Add("COM" & CStr(i + 1))
332         End If
333     Next
334 End Sub
335
336 Public Sub FillPubs()
337     Dim StrPort As String
338
339     StrPort = Trim(CmbComPort.Text)
340     IntCommPort = Microsoft.VisualBasic.Right(StrPort, 1)
341
342     Select Case CmbBaudRate.SelectedIndex
343     Case 0
344         IntBaudRate = 110
345     Case 1
346         IntBaudRate = 300
347     Case 2
348         IntBaudRate = 1200
349     Case 3
350         IntBaudRate = 2400
351     Case 4
352         IntBaudRate = 4800
353     Case 5
354         IntBaudRate = 9600
355     Case 6
356         IntBaudRate = 19200

```

```

357         Case 7
358             IntBaudRate = 38400
359         Case 8
360             IntBaudRate = 57600
361         Case 9
362             IntBaudRate = 115200
363         Case 10
364             IntBaudRate = 460800
365         Case 11
366             IntBaudRate = 921600
367     End Select
368
369     Select Case CmbParity.SelectedIndex
370     Case 0
371         IntParityBits = 2
372     Case 1
373         IntParityBits = 1
374     Case 2
375         IntParityBits = 0
376     Case 3
377         IntParityBits = 3
378     End Select
379
380     Select Case CmbDataBits.SelectedIndex
381     Case 0
382         IntDataBits = 5
383     Case 1
384         IntDataBits = 6
385     Case 2
386         IntDataBits = 7
387     Case 3
388         IntDataBits = 8
389     End Select
390
391     Select Case CmbStopBits.SelectedIndex
392     Case 0
393         IntStopBits = 1
394     Case 1
395         IntStopBits = 2
396     End Select
397     End Sub
398 End Class
399

```

```

1 Imports System.Text
2 Imports System.IO
3
4 Public Class FrmParameters
5     Inherits System.Windows.Forms.Form
6
7 #Region " Windows Form Designer generated code "
8
9     Public Sub New()
10        MyBase.New()
11
12        'This call is required by the Windows Form Designer.
13        InitializeComponent()
14
15        'Add any initialization after the InitializeComponent() call
16
17    End Sub
18
19    'Form overrides dispose to clean up the component list.
20    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
21        If disposing Then
22            If Not (components Is Nothing) Then
23                components.Dispose()
24            End If
25        End If
26        MyBase.Dispose(disposing)
27    End Sub
28
29    'Required by the Windows Form Designer
30    Private components As System.ComponentModel.IContainer
31
32    'NOTE: The following procedure is required by the Windows Form Designer
33    'It can be modified using the Windows Form Designer.
34    'Do not modify it using the code editor.
35    Friend WithEvents BtSaveSentData As System.Windows.Forms.Button
36    Friend WithEvents TxtSaveSentData As System.Windows.Forms.TextBox
37    Friend WithEvents BtSaveRecvData As System.Windows.Forms.Button
38    Friend WithEvents TxtSaveRecvData As System.Windows.Forms.TextBox
39    Friend WithEvents LblDefaultLanguage As System.Windows.Forms.Label
40    Friend WithEvents ChkShowSelectLang As System.Windows.Forms.CheckBox
41    Friend WithEvents LblOutputFileName As System.Windows.Forms.Label
42    Friend WithEvents LblInputFileName As System.Windows.Forms.Label
43    Friend WithEvents BtSave As System.Windows.Forms.Button
44    Friend WithEvents BtBack As System.Windows.Forms.Button
45    Friend WithEvents BtSendFile As System.Windows.Forms.Button
46    Friend WithEvents TxtSendFile As System.Windows.Forms.TextBox
47    Friend WithEvents LblSendFile As System.Windows.Forms.Label
48    Friend WithEvents CmbLangs As System.Windows.Forms.ComboBox
49    Friend WithEvents GrBLangs As System.Windows.Forms.GroupBox
50    Friend WithEvents GrBSaveOutputData As System.Windows.Forms.GroupBox
51    Friend WithEvents GrBSaveInputData As System.Windows.Forms.GroupBox
52    Friend WithEvents GrBSendFile As System.Windows.Forms.GroupBox
53    Friend WithEvents GrBConParams As System.Windows.Forms.GroupBox
54    Friend WithEvents BtConfFile As System.Windows.Forms.Button
55    Friend WithEvents TxtConfFile As System.Windows.Forms.TextBox
56    Friend WithEvents LblConfFile As System.Windows.Forms.Label
57    Friend WithEvents GrBSaveConfFile As System.Windows.Forms.GroupBox
58    Friend WithEvents CmbComPort As System.Windows.Forms.ComboBox
59    Friend WithEvents LblComPort As System.Windows.Forms.Label
60    Friend WithEvents LblBaudRate As System.Windows.Forms.Label
61    Friend WithEvents LblParity As System.Windows.Forms.Label
62    Friend WithEvents LblStopBits As System.Windows.Forms.Label
63    Friend WithEvents LblDataBits As System.Windows.Forms.Label
64    Friend WithEvents CmbStopBits As System.Windows.Forms.ComboBox
65    Friend WithEvents CmbParity As System.Windows.Forms.ComboBox
66    Friend WithEvents CmbDataBits As System.Windows.Forms.ComboBox
67    Friend WithEvents GrBThrust As System.Windows.Forms.GroupBox
68    Friend WithEvents BtThrustIncoming As System.Windows.Forms.Button
69    Friend WithEvents TxtThrustIncoming As System.Windows.Forms.TextBox
70    Friend WithEvents LblThrustIncoming As System.Windows.Forms.Label
71    Friend WithEvents BtThrustOutgoing As System.Windows.Forms.Button
72    Friend WithEvents TxtThrustOutgoing As System.Windows.Forms.TextBox
73    Friend WithEvents LblThrustOutgoing As System.Windows.Forms.Label
74    Friend WithEvents TCMain As System.Windows.Forms.TabControl
75    Friend WithEvents TPSerialCommunication As System.Windows.Forms.TabPage

```

```

76 Friend WithEvents TPThrust As System.Windows.Forms.TabPage
77 Friend WithEvents TPLanguage As System.Windows.Forms.TabPage
78 Friend WithEvents TPGraphs As System.Windows.Forms.TabPage
79 Friend WithEvents TabControl1 As System.Windows.Forms.TabControl
80 Friend WithEvents TPNetGraphs As System.Windows.Forms.TabPage
81 Friend WithEvents TPExcelGraphs As System.Windows.Forms.TabPage
82 Friend WithEvents TPGraphRefreshRate As System.Windows.Forms.TabPage
83 Friend WithEvents PanelChartProperties As System.Windows.Forms.Panel
84 Friend WithEvents GbExcelChart2Props As System.Windows.Forms.GroupBox
85 Friend WithEvents LblExcelC2SelectLine As System.Windows.Forms.Label
86 Friend WithEvents LbExcelC2ChartPoints As System.Windows.Forms.Label
87 Friend WithEvents NUDExcelC2ChartPoints As System.Windows.Forms.NumericUpDown
88 Friend WithEvents DUBExcelC2SelectLine As System.Windows.Forms.DomainUpDown
89 Friend WithEvents GbExcelChart1Props As System.Windows.Forms.GroupBox
90 Friend WithEvents LblExcelC1SelectLine As System.Windows.Forms.Label
91 Friend WithEvents LbExcelC1ChartPoints As System.Windows.Forms.Label
92 Friend WithEvents NUDExcelC1ChartPoints As System.Windows.Forms.NumericUpDown
93 Friend WithEvents DUBExcelC1SelectLine As System.Windows.Forms.DomainUpDown
94 Friend WithEvents GrBRefreshRateofGraphics As System.Windows.Forms.GroupBox
95 Friend WithEvents LbGraphicsRate As System.Windows.Forms.Label
96 Friend WithEvents LbBasicGraphisRate As System.Windows.Forms.Label
97 Friend WithEvents CmbGraphicsRate As System.Windows.Forms.ComboBox
98 Friend WithEvents CmbBasicGraphisRate As System.Windows.Forms.ComboBox
99 Friend WithEvents GBNetGraphProps2 As System.Windows.Forms.GroupBox
100 Friend WithEvents NUDNetC2TimeVertInterval As System.Windows.Forms.NumericUpDown
101 Friend WithEvents DUDNetC2VerticalLine As System.Windows.Forms.DomainUpDown
102 Friend WithEvents DudNetC2TypeLines As System.Windows.Forms.DomainUpDown
103 Friend WithEvents LblNetC2HorlDiv As System.Windows.Forms.Label
104 Friend WithEvents ChkNetC2Ampere As System.Windows.Forms.CheckBox
105 Friend WithEvents ChkNetC2Voltage As System.Windows.Forms.CheckBox
106 Friend WithEvents ChkNetC2LoadCell As System.Windows.Forms.CheckBox
107 Friend WithEvents ChkNetC2PWMLoad As System.Windows.Forms.CheckBox
108 Friend WithEvents LblNetC2HorLines As System.Windows.Forms.Label
109 Friend WithEvents NUDNetC2HorLines As System.Windows.Forms.NumericUpDown
110 Friend WithEvents LblNetC2NumOfPoints As System.Windows.Forms.Label
111 Friend WithEvents NUD1NetC2NumOfPoints As System.Windows.Forms.NumericUpDown
112 Friend WithEvents GBNetGraphProps1 As System.Windows.Forms.GroupBox
113 Friend WithEvents NUDNetC1TimeVertInterval As System.Windows.Forms.NumericUpDown
114 Friend WithEvents DUDNetC1VerticalLine As System.Windows.Forms.DomainUpDown
115 Friend WithEvents LblNetC1HorlDiv As System.Windows.Forms.Label
116 Friend WithEvents NUDNetC1HorlDiv As System.Windows.Forms.NumericUpDown
117 Friend WithEvents DudNetC1TypeLines As System.Windows.Forms.DomainUpDown
118 Friend WithEvents ChkNetC1SetPointPwm As System.Windows.Forms.CheckBox
119 Friend WithEvents ChkNetC1PWMLoad As System.Windows.Forms.CheckBox
120 Friend WithEvents ChkNetC1SetPointRPM As System.Windows.Forms.CheckBox
121 Friend WithEvents ChkNetC1Rpm As System.Windows.Forms.CheckBox
122 Friend WithEvents LblNetC1HorLines As System.Windows.Forms.Label
123 Friend WithEvents NUDNetC1HorLines As System.Windows.Forms.NumericUpDown
124 Friend WithEvents LblNetC1NumOfPoints As System.Windows.Forms.Label
125 Friend WithEvents NUDNetC1NumOfPoints As System.Windows.Forms.NumericUpDown
126 Friend WithEvents NUD1NetC2HorlDiv As System.Windows.Forms.NumericUpDown
127 Friend WithEvents GBDefaultGraph As System.Windows.Forms.GroupBox
128 Friend WithEvents RBExcelGraph As System.Windows.Forms.RadioButton
129 Friend WithEvents RBNetGraph As System.Windows.Forms.RadioButton
130 Friend WithEvents NUDNetC1ConstVerLines As System.Windows.Forms.NumericUpDown
131 Friend WithEvents NUDNetC2ConstVerLines As System.Windows.Forms.NumericUpDown
132 Friend WithEvents CmbBaudRate As System.Windows.Forms.ComboBox
133 <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
134     Dim resources As System.ComponentModel.ResourceManager = New System.
ComponentModel.ResourceManager(GetType(FrmParameters))
135     Me.GrBLangs = New System.Windows.Forms.GroupBox
136     Me.ChkShowSelectLang = New System.Windows.Forms.CheckBox
137     Me.CmbLangs = New System.Windows.Forms.ComboBox
138     Me.LblDefaultLanguage = New System.Windows.Forms.Label
139     Me.GrBSaveOutputData = New System.Windows.Forms.GroupBox
140     Me.BtSaveSentData = New System.Windows.Forms.Button
141     Me.TxtSaveSentData = New System.Windows.Forms.TextBox
142     Me.LblOutputFileName = New System.Windows.Forms.Label
143     Me.GrBSaveInputData = New System.Windows.Forms.GroupBox
144     Me.LblInputFileName = New System.Windows.Forms.Label
145     Me.BtSaveRecvData = New System.Windows.Forms.Button
146     Me.TxtSaveRecvData = New System.Windows.Forms.TextBox
147     Me.GrBSendFile = New System.Windows.Forms.GroupBox
148     Me.BtSendFile = New System.Windows.Forms.Button
149     Me.TxtSendFile = New System.Windows.Forms.TextBox

```

```

150     Me.LblSendFile = New System.Windows.Forms.Label
151     Me.GrBConParams = New System.Windows.Forms.GroupBox
152     Me.CmbStopBits = New System.Windows.Forms.ComboBox
153     Me.CmbParity = New System.Windows.Forms.ComboBox
154     Me.CmbDataBits = New System.Windows.Forms.ComboBox
155     Me.LblDataBits = New System.Windows.Forms.Label
156     Me.LblStopBits = New System.Windows.Forms.Label
157     Me.LblParity = New System.Windows.Forms.Label
158     Me.LblBaudRate = New System.Windows.Forms.Label
159     Me.LblComPort = New System.Windows.Forms.Label
160     Me.CmbBaudRate = New System.Windows.Forms.ComboBox
161     Me.CmbComPort = New System.Windows.Forms.ComboBox
162     Me.BtSave = New System.Windows.Forms.Button
163     Me.BtBack = New System.Windows.Forms.Button
164     Me.GrBSaveConfFile = New System.Windows.Forms.GroupBox
165     Me.BtConfFile = New System.Windows.Forms.Button
166     Me.TxtConfFile = New System.Windows.Forms.TextBox
167     Me.LblConfFile = New System.Windows.Forms.Label
168     Me.GrBThrust = New System.Windows.Forms.GroupBox
169     Me.BtThrustOutgoing = New System.Windows.Forms.Button
170     Me.TxtThrustOutgoing = New System.Windows.Forms.TextBox
171     Me.LblThrustOutgoing = New System.Windows.Forms.Label
172     Me.BtThrustIncoming = New System.Windows.Forms.Button
173     Me.TxtThrustIncoming = New System.Windows.Forms.TextBox
174     Me.LblThrustIncoming = New System.Windows.Forms.Label
175     Me.TCMain = New System.Windows.Forms.TabControl
176     Me.TPLanguage = New System.Windows.Forms.TabPage
177     Me.TPSerialCommunication = New System.Windows.Forms.TabPage
178     Me.TPThrust = New System.Windows.Forms.TabPage
179     Me.TPGraphs = New System.Windows.Forms.TabPage
180     Me.GBDefaultGraph = New System.Windows.Forms.GroupBox
181     Me.RBExcelGraph = New System.Windows.Forms.RadioButton
182     Me.RBNetGraph = New System.Windows.Forms.RadioButton
183     Me.TabControl1 = New System.Windows.Forms.TabControl
184     Me.TPNetGraphs = New System.Windows.Forms.TabPage
185     Me.GBNetGraphProps2 = New System.Windows.Forms.GroupBox
186     Me.NUD1NetC2HorlDiv = New System.Windows.Forms.NumericUpDown
187     Me.NUDNetC2TimeVertInterval = New System.Windows.Forms.NumericUpDown
188     Me.DUDNetC2VerticalLine = New System.Windows.Forms.DomainUpDown
189     Me.DudNetC2TypeLines = New System.Windows.Forms.DomainUpDown
190     Me.LblNetC2HorlDiv = New System.Windows.Forms.Label
191     Me.ChkNetC2Ampere = New System.Windows.Forms.CheckBox
192     Me.ChkNetC2Voltage = New System.Windows.Forms.CheckBox
193     Me.ChkNetC2LoadCell = New System.Windows.Forms.CheckBox
194     Me.ChkNetC2PWMLoad = New System.Windows.Forms.CheckBox
195     Me.LblNetC2HorLines = New System.Windows.Forms.Label
196     Me.NUDNetC2HorLines = New System.Windows.Forms.NumericUpDown
197     Me.LblNetC2NumOfPoints = New System.Windows.Forms.Label
198     Me.NUD1NetC2NumOfPoints = New System.Windows.Forms.NumericUpDown
199     Me.NUDNetC2ConstVerLines = New System.Windows.Forms.NumericUpDown
200     Me.GBNetGraphProps1 = New System.Windows.Forms.GroupBox
201     Me.DUDNetC1VerticalLine = New System.Windows.Forms.DomainUpDown
202     Me.LblNetC1HorlDiv = New System.Windows.Forms.Label
203     Me.NUDNetC1HorlDiv = New System.Windows.Forms.NumericUpDown
204     Me.DudNetC1TypeLines = New System.Windows.Forms.DomainUpDown
205     Me.ChkNetC1SetPointPwm = New System.Windows.Forms.CheckBox
206     Me.ChkNetC1PWMLoad = New System.Windows.Forms.CheckBox
207     Me.ChkNetC1SetPointRPM = New System.Windows.Forms.CheckBox
208     Me.ChkNetC1Rpm = New System.Windows.Forms.CheckBox
209     Me.LblNetC1HorLines = New System.Windows.Forms.Label
210     Me.NUDNetC1HorLines = New System.Windows.Forms.NumericUpDown
211     Me.LblNetC1NumOfPoints = New System.Windows.Forms.Label
212     Me.NUDNetC1NumOfPoints = New System.Windows.Forms.NumericUpDown
213     Me.NUDNetC1TimeVertInterval = New System.Windows.Forms.NumericUpDown
214     Me.NUDNetC1ConstVerLines = New System.Windows.Forms.NumericUpDown
215     Me.TPExcelGraphs = New System.Windows.Forms.TabPage
216     Me.PanelChartProperties = New System.Windows.Forms.Panel
217     Me.GbExcelChart2Props = New System.Windows.Forms.GroupBox
218     Me.LblExcelC2SelectLine = New System.Windows.Forms.Label
219     Me.LbExcelC2ChartPoints = New System.Windows.Forms.Label
220     Me.NUDEExcelC2ChartPoints = New System.Windows.Forms.NumericUpDown
221     Me.DUBExcelC2SelectLine = New System.Windows.Forms.DomainUpDown
222     Me.GbExcelChart1Props = New System.Windows.Forms.GroupBox
223     Me.LblExcelC1SelectLine = New System.Windows.Forms.Label
224     Me.LbExcelC1ChartPoints = New System.Windows.Forms.Label

```



```

225     Me.NUDEXcelC1ChartPoints = New System.Windows.Forms.NumericUpDown
226     Me.DUBExcelC1SelectLine = New System.Windows.Forms.DomainUpDown
227     Me.TPGGraphRefreshRate = New System.Windows.Forms.TabPage
228     Me.GrBRefreshRateofGraphics = New System.Windows.Forms.GroupBox
229     Me.LbGraphicsRate = New System.Windows.Forms.Label
230     Me.LbBasicGraphisRate = New System.Windows.Forms.Label
231     Me.CmbGraphicsRate = New System.Windows.Forms.ComboBox
232     Me.CmbBasicGraphisRate = New System.Windows.Forms.ComboBox
233     Me.GrBLangs.SuspendLayout()
234     Me.GrBSaveOutputData.SuspendLayout()
235     Me.GrBSaveInputData.SuspendLayout()
236     Me.GrBSendFile.SuspendLayout()
237     Me.GrBConParams.SuspendLayout()
238     Me.GrBSaveConfFile.SuspendLayout()
239     Me.GrBThrust.SuspendLayout()
240     Me.TCMain.SuspendLayout()
241     Me.TPLanguage.SuspendLayout()
242     Me.TPSerialCommunication.SuspendLayout()
243     Me.TPThrust.SuspendLayout()
244     Me.TPGraphs.SuspendLayout()
245     Me.GBDefaultGraph.SuspendLayout()
246     Me.TabControll.SuspendLayout()
247     Me.TPNetGraphs.SuspendLayout()
248     Me.GBNetGraphProps2.SuspendLayout()
249     CType(Me.NUD1NetC2HorlDiv, System.ComponentModel.ISupportInitialize).BeginInit()
250     CType(Me.NUDNetC2TimeVertInterval, System.ComponentModel.ISupportInitialize).
BeginInit()
251     CType(Me.NUDNetC2HorLines, System.ComponentModel.ISupportInitialize).BeginInit()
252     CType(Me.NUD1NetC2NumOfPoints, System.ComponentModel.ISupportInitialize).
BeginInit()
253     CType(Me.NUDNetC2ConstVerLines, System.ComponentModel.ISupportInitialize).
BeginInit()
254     Me.GBNetGraphProps1.SuspendLayout()
255     CType(Me.NUDNetC1HorlDiv, System.ComponentModel.ISupportInitialize).BeginInit()
256     CType(Me.NUDNetC1HorLines, System.ComponentModel.ISupportInitialize).BeginInit()
257     CType(Me.NUDNetC1NumOfPoints, System.ComponentModel.ISupportInitialize).
BeginInit()
258     CType(Me.NUDNetC1TimeVertInterval, System.ComponentModel.ISupportInitialize).
BeginInit()
259     CType(Me.NUDNetC1ConstVerLines, System.ComponentModel.ISupportInitialize).
BeginInit()
260     Me.TPExcelGraphs.SuspendLayout()
261     Me.PanelChartProperties.SuspendLayout()
262     Me.GbExcelChart2Props.SuspendLayout()
263     CType(Me.NUDEXcelC2ChartPoints, System.ComponentModel.ISupportInitialize).
BeginInit()
264     Me.GbExcelChart1Props.SuspendLayout()
265     CType(Me.NUDEXcelC1ChartPoints, System.ComponentModel.ISupportInitialize).
BeginInit()
266     Me.TPGGraphRefreshRate.SuspendLayout()
267     Me.GrBRefreshRateofGraphics.SuspendLayout()
268     Me.SuspendLayout()
269     '
270     'GrBLangs
271     '
272     Me.GrBLangs.Controls.Add(Me.ChkShowSelectLang)
273     Me.GrBLangs.Controls.Add(Me.CmbLangs)
274     Me.GrBLangs.Controls.Add(Me.LblDefaultLanguage)
275     Me.GrBLangs.Font = New System.Drawing.Font("Microsoft Sans Serif", 8.25!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(161,
Byte))
276     Me.GrBLangs.Location = New System.Drawing.Point(26, 18)
277     Me.GrBLangs.Name = "GrBLangs"
278     Me.GrBLangs.Size = New System.Drawing.Size(576, 84)
279     Me.GrBLangs.TabIndex = 0
280     Me.GrBLangs.TabStop = False
281     Me.GrBLangs.Text = "Προεπιλεγμένη Γλώσσα"
282     '
283     'ChkShowSelectLang
284     '
285     Me.ChkShowSelectLang.Location = New System.Drawing.Point(19, 48)

```

```

286     Me.ChkShowSelectLang.Name = "ChkShowSelectLang"
287     Me.ChkShowSelectLang.Size = New System.Drawing.Size(488, 30)
288     Me.ChkShowSelectLang.TabIndex = 2
289     Me.ChkShowSelectLang.Text = "Εμφάνιση Επιλογής Γλώσσας στην Εκκίνηση της Εφαρμογής"
290     '
291     'CmbLangs
292     '
293     Me.CmbLangs.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList
294     Me.CmbLangs.Location = New System.Drawing.Point(269, 16)
295     Me.CmbLangs.Name = "CmbLangs"
296     Me.CmbLangs.Size = New System.Drawing.Size(168, 21)
297     Me.CmbLangs.TabIndex = 1
298     '
299     'LblDefaultLanguage
300     '
301     Me.LblDefaultLanguage.Location = New System.Drawing.Point(16, 16)
302     Me.LblDefaultLanguage.Name = "LblDefaultLanguage"
303     Me.LblDefaultLanguage.Size = New System.Drawing.Size(232, 24)
304     Me.LblDefaultLanguage.TabIndex = 0
305     Me.LblDefaultLanguage.Text = "Επιλογή Προεπιλεγμένης Γλώσσας"
306     '
307     'GrBSaveOutputData
308     '
309     Me.GrBSaveOutputData.Controls.Add(Me.BtSaveSentData)
310     Me.GrBSaveOutputData.Controls.Add(Me.TxtSaveSentData)
311     Me.GrBSaveOutputData.Controls.Add(Me.LblOutputFileName)
312     Me.GrBSaveOutputData.Location = New System.Drawing.Point(52, 95)
313     Me.GrBSaveOutputData.Name = "GrBSaveOutputData"
314     Me.GrBSaveOutputData.Size = New System.Drawing.Size(576, 50)
315     Me.GrBSaveOutputData.TabIndex = 1
316     Me.GrBSaveOutputData.TabStop = False
317     Me.GrBSaveOutputData.Text = "Αποθήκευση Εξερχόμενων Δεδομένων"
318     '
319     'BtSaveSentData
320     '
321     Me.BtSaveSentData.Image = CType(resources.GetObject("BtSaveSentData.Image"), System.Drawing.Image)
322     Me.BtSaveSentData.ImeMode = System.Windows.Forms.ImeMode.NoControl
323     Me.BtSaveSentData.Location = New System.Drawing.Point(524, 16)
324     Me.BtSaveSentData.Name = "BtSaveSentData"
325     Me.BtSaveSentData.Size = New System.Drawing.Size(24, 23)
326     Me.BtSaveSentData.TabIndex = 5
327     '
328     'TxtSaveSentData
329     '
330     Me.TxtSaveSentData.Location = New System.Drawing.Point(184, 13)
331     Me.TxtSaveSentData.Name = "TxtSaveSentData"
332     Me.TxtSaveSentData.Size = New System.Drawing.Size(328, 20)
333     Me.TxtSaveSentData.TabIndex = 4
334     '
335     'LblOutputFileName
336     '
337     Me.LblOutputFileName.Location = New System.Drawing.Point(16, 16)
338     Me.LblOutputFileName.Name = "LblOutputFileName"
339     Me.LblOutputFileName.Size = New System.Drawing.Size(152, 24)
340     Me.LblOutputFileName.TabIndex = 0
341     Me.LblOutputFileName.Text = "Όνομα Αρχείου"
342     '
343     'GrBSaveInputData
344     '
345     Me.GrBSaveInputData.Controls.Add(Me.LblInputFileName)
346     Me.GrBSaveInputData.Controls.Add(Me.BtSaveRecvData)
347     Me.GrBSaveInputData.Controls.Add(Me.TxtSaveRecvData)
348     Me.GrBSaveInputData.Location = New System.Drawing.Point(52, 27)
349     Me.GrBSaveInputData.Name = "GrBSaveInputData"
350     Me.GrBSaveInputData.Size = New System.Drawing.Size(576, 51)
351     Me.GrBSaveInputData.TabIndex = 2
352     Me.GrBSaveInputData.TabStop = False
353     Me.GrBSaveInputData.Text = "Αποθήκευση Εισερχόμενων Δεδομένων"
354     '
355     'LblInputFileName
356     '
357     Me.LblInputFileName.Location = New System.Drawing.Point(20, 16)
358     Me.LblInputFileName.Name = "LblInputFileName"

```

```

359     Me.LblInputFileName.Size = New System.Drawing.Size(152, 24)
360     Me.LblInputFileName.TabIndex = 0
361     Me.LblInputFileName.Text = "Όνομα Αρχείου"
362     '
363     'BtSaveRecvData
364     '
365     Me.BtSaveRecvData.Image = CType(resources.GetObject("BtSaveRecvData.Image"),
System.Drawing.Image)
366     Me.BtSaveRecvData.ImeMode = System.Windows.Forms.ImeMode.NoControl
367     Me.BtSaveRecvData.Location = New System.Drawing.Point(524, 16)
368     Me.BtSaveRecvData.Name = "BtSaveRecvData"
369     Me.BtSaveRecvData.Size = New System.Drawing.Size(24, 23)
370     Me.BtSaveRecvData.TabIndex = 7
371     '
372     'TxtSaveRecvData
373     '
374     Me.TxtSaveRecvData.Location = New System.Drawing.Point(184, 16)
375     Me.TxtSaveRecvData.Name = "TxtSaveRecvData"
376     Me.TxtSaveRecvData.Size = New System.Drawing.Size(328, 20)
377     Me.TxtSaveRecvData.TabIndex = 6
378     '
379     'GrBSendFile
380     '
381     Me.GrBSendFile.Controls.Add(Me.BtSendFile)
382     Me.GrBSendFile.Controls.Add(Me.TxtSendFile)
383     Me.GrBSendFile.Controls.Add(Me.LblSendFile)
384     Me.GrBSendFile.Location = New System.Drawing.Point(52, 160)
385     Me.GrBSendFile.Name = "GrBSendFile"
386     Me.GrBSendFile.Size = New System.Drawing.Size(576, 51)
387     Me.GrBSendFile.TabIndex = 3
388     Me.GrBSendFile.TabStop = False
389     Me.GrBSendFile.Text = "Αποστολή Δεδομένων από Αρχείο"
390     '
391     'BtSendFile
392     '
393     Me.BtSendFile.Image = CType(resources.GetObject("BtSendFile.Image"), System.
Drawing.Image)
394     Me.BtSendFile.ImeMode = System.Windows.Forms.ImeMode.NoControl
395     Me.BtSendFile.Location = New System.Drawing.Point(524, 16)
396     Me.BtSendFile.Name = "BtSendFile"
397     Me.BtSendFile.Size = New System.Drawing.Size(24, 23)
398     Me.BtSendFile.TabIndex = 9
399     '
400     'TxtSendFile
401     '
402     Me.TxtSendFile.Location = New System.Drawing.Point(184, 16)
403     Me.TxtSendFile.Name = "TxtSendFile"
404     Me.TxtSendFile.Size = New System.Drawing.Size(328, 20)
405     Me.TxtSendFile.TabIndex = 8
406     '
407     'LblSendFile
408     '
409     Me.LblSendFile.Location = New System.Drawing.Point(20, 16)
410     Me.LblSendFile.Name = "LblSendFile"
411     Me.LblSendFile.Size = New System.Drawing.Size(144, 24)
412     Me.LblSendFile.TabIndex = 1
413     Me.LblSendFile.Text = "Όνομα Αρχείου"
414     '
415     'GrBConParams
416     '
417     Me.GrBConParams.Controls.Add(Me.CmbStopBits)
418     Me.GrBConParams.Controls.Add(Me.CmbParity)
419     Me.GrBConParams.Controls.Add(Me.CmbDataBits)
420     Me.GrBConParams.Controls.Add(Me.LblDataBits)
421     Me.GrBConParams.Controls.Add(Me.LblStopBits)
422     Me.GrBConParams.Controls.Add(Me.LblParity)
423     Me.GrBConParams.Controls.Add(Me.LblBaudRate)
424     Me.GrBConParams.Controls.Add(Me.LblComPort)
425     Me.GrBConParams.Controls.Add(Me.CmbBaudRate)
426     Me.GrBConParams.Controls.Add(Me.CmbComPort)
427     Me.GrBConParams.Location = New System.Drawing.Point(186, 232)
428     Me.GrBConParams.Name = "GrBConParams"
429     Me.GrBConParams.Size = New System.Drawing.Size(264, 184)
430     Me.GrBConParams.TabIndex = 4
431     Me.GrBConParams.TabStop = False

```

```

432     Me.GrBConParams.Text = "Παράμετροι Σύνδεσης"
433     '
434     'CmbStopBits
435     '
436     Me.CmbStopBits.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList
437     Me.CmbStopBits.Items.AddRange(New Object() {"1", "2"})
438     Me.CmbStopBits.Location = New System.Drawing.Point(152, 152)
439     Me.CmbStopBits.Name = "CmbStopBits"
440     Me.CmbStopBits.Size = New System.Drawing.Size(72, 21)
441     Me.CmbStopBits.TabIndex = 29
442     '
443     'CmbParity
444     '
445     Me.CmbParity.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList
446     Me.CmbParity.Items.AddRange(New Object() {"Even", "Odd", "None", "Mark"})
447     Me.CmbParity.Location = New System.Drawing.Point(152, 88)
448     Me.CmbParity.Name = "CmbParity"
449     Me.CmbParity.Size = New System.Drawing.Size(72, 21)
450     Me.CmbParity.TabIndex = 27
451     '
452     'CmbDataBits
453     '
454     Me.CmbDataBits.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList
455     Me.CmbDataBits.Items.AddRange(New Object() {"5", "6", "7", "8"})
456     Me.CmbDataBits.Location = New System.Drawing.Point(152, 120)
457     Me.CmbDataBits.Name = "CmbDataBits"
458     Me.CmbDataBits.Size = New System.Drawing.Size(72, 21)
459     Me.CmbDataBits.TabIndex = 28
460     '
461     'LblDataBits
462     '
463     Me.LblDataBits.Location = New System.Drawing.Point(8, 120)
464     Me.LblDataBits.Name = "LblDataBits"
465     Me.LblDataBits.Size = New System.Drawing.Size(136, 16)
466     Me.LblDataBits.TabIndex = 25
467     Me.LblDataBits.Text = "Data Bits"
468     '
469     'LblStopBits
470     '
471     Me.LblStopBits.Location = New System.Drawing.Point(8, 152)
472     Me.LblStopBits.Name = "LblStopBits"
473     Me.LblStopBits.Size = New System.Drawing.Size(136, 16)
474     Me.LblStopBits.TabIndex = 24
475     Me.LblStopBits.Text = "Stop Bits"
476     '
477     'LblParity
478     '
479     Me.LblParity.Location = New System.Drawing.Point(8, 88)
480     Me.LblParity.Name = "LblParity"
481     Me.LblParity.Size = New System.Drawing.Size(136, 16)
482     Me.LblParity.TabIndex = 23
483     Me.LblParity.Text = "Parity"
484     '
485     'LblBaudRate
486     '
487     Me.LblBaudRate.Location = New System.Drawing.Point(8, 56)
488     Me.LblBaudRate.Name = "LblBaudRate"
489     Me.LblBaudRate.Size = New System.Drawing.Size(136, 16)
490     Me.LblBaudRate.TabIndex = 22
491     Me.LblBaudRate.Text = "Baud Rate"
492     '
493     'LblComPort
494     '
495     Me.LblComPort.Location = New System.Drawing.Point(8, 24)
496     Me.LblComPort.Name = "LblComPort"
497     Me.LblComPort.Size = New System.Drawing.Size(136, 16)
498     Me.LblComPort.TabIndex = 21
499     Me.LblComPort.Text = "COM Port"
500     '
501     'CmbBaudRate
502     '
503     Me.CmbBaudRate.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList
504     Me.CmbBaudRate.Items.AddRange(New Object() {"110", "300", "1200", "2400",
505     "4800", "9600", "19200", "38400", "57600", "115200", "460800", "921600"})

```

```

506     Me.CmbBaudRate.Name = "CmbBaudRate"
507     Me.CmbBaudRate.Size = New System.Drawing.Size(72, 21)
508     Me.CmbBaudRate.TabIndex = 26
509     '
510     'CmbComPort
511     '
512     Me.CmbComPort.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList
513     Me.CmbComPort.Items.AddRange(New Object() {"COM1", "COM2", "COM3", "COM4"})
514     Me.CmbComPort.Location = New System.Drawing.Point(152, 16)
515     Me.CmbComPort.Name = "CmbComPort"
516     Me.CmbComPort.Size = New System.Drawing.Size(72, 21)
517     Me.CmbComPort.TabIndex = 12
518     '
519     'BtSave
520     '
521     Me.BtSave.Location = New System.Drawing.Point(87, 530)
522     Me.BtSave.Name = "BtSave"
523     Me.BtSave.Size = New System.Drawing.Size(232, 24)
524     Me.BtSave.TabIndex = 5
525     Me.BtSave.Text = "Αποθήκευση Αλλαγών"
526     '
527     'BtBack
528     '
529     Me.BtBack.DialogResult = System.Windows.Forms.DialogResult.Cancel
530     Me.BtBack.Location = New System.Drawing.Point(336, 530)
531     Me.BtBack.Name = "BtBack"
532     Me.BtBack.Size = New System.Drawing.Size(232, 24)
533     Me.BtBack.TabIndex = 6
534     Me.BtBack.Text = "Επιστροφή στην Κύρια Οθόνη"
535     '
536     'GrBSaveConfFile
537     '
538     Me.GrBSaveConfFile.Controls.Add(Me.BtConfFile)
539     Me.GrBSaveConfFile.Controls.Add(Me.TxtConfFile)
540     Me.GrBSaveConfFile.Controls.Add(Me.LblConfFile)
541     Me.GrBSaveConfFile.Location = New System.Drawing.Point(8, 0)
542     Me.GrBSaveConfFile.Name = "GrBSaveConfFile"
543     Me.GrBSaveConfFile.Size = New System.Drawing.Size(576, 45)
544     Me.GrBSaveConfFile.TabIndex = 7
545     Me.GrBSaveConfFile.TabStop = False
546     Me.GrBSaveConfFile.Text = "Αποθήκευση του Αρχείου Ρυθμίσεων"
547     '
548     'BtConfFile
549     '
550     Me.BtConfFile.Image = CType(resources.GetObject("BtConfFile.Image"), System.
Drawing.Image)
551     Me.BtConfFile.ImeMode = System.Windows.Forms.ImeMode.NoControl
552     Me.BtConfFile.Location = New System.Drawing.Point(520, 15)
553     Me.BtConfFile.Name = "BtConfFile"
554     Me.BtConfFile.Size = New System.Drawing.Size(24, 23)
555     Me.BtConfFile.TabIndex = 8
556     '
557     'TxtConfFile
558     '
559     Me.TxtConfFile.Location = New System.Drawing.Point(184, 18)
560     Me.TxtConfFile.Name = "TxtConfFile"
561     Me.TxtConfFile.Size = New System.Drawing.Size(328, 20)
562     Me.TxtConfFile.TabIndex = 7
563     '
564     'LblConfFile
565     '
566     Me.LblConfFile.Location = New System.Drawing.Point(16, 19)
567     Me.LblConfFile.Name = "LblConfFile"
568     Me.LblConfFile.Size = New System.Drawing.Size(152, 19)
569     Me.LblConfFile.TabIndex = 6
570     Me.LblConfFile.Text = "Όνομα Αρχείου"
571     '
572     'GrBThrust
573     '
574     Me.GrBThrust.Controls.Add(Me.BtThrustOutgoing)
575     Me.GrBThrust.Controls.Add(Me.TxtThrustOutgoing)
576     Me.GrBThrust.Controls.Add(Me.LblThrustOutgoing)
577     Me.GrBThrust.Controls.Add(Me.BtThrustIncoming)
578     Me.GrBThrust.Controls.Add(Me.TxtThrustIncoming)
579     Me.GrBThrust.Controls.Add(Me.LblThrustIncoming)

```

```

580     Me.GrBThrust.Location = New System.Drawing.Point(34, 92)
581     Me.GrBThrust.Name = "GrBThrust"
582     Me.GrBThrust.Size = New System.Drawing.Size(576, 86)
583     Me.GrBThrust.TabIndex = 9
584     Me.GrBThrust.TabStop = False
585     Me.GrBThrust.Text = "Αποθήκευση Δεδομένων της Εφαρμογής Κλίνη Ωσης και Ελέγχου
Στροφών (RPM)"
586     '
587     'BtThrustOutgoing
588     '
589     Me.BtThrustOutgoing.Image = CType(resources.GetObject("BtThrustOutgoing.Image"
), System.Drawing.Image)
590     Me.BtThrustOutgoing.ImeMode = System.Windows.Forms.ImeMode.NoControl
591     Me.BtThrustOutgoing.Location = New System.Drawing.Point(524, 47)
592     Me.BtThrustOutgoing.Name = "BtThrustOutgoing"
593     Me.BtThrustOutgoing.Size = New System.Drawing.Size(24, 23)
594     Me.BtThrustOutgoing.TabIndex = 8
595     '
596     'TxtThrustOutgoing
597     '
598     Me.TxtThrustOutgoing.Location = New System.Drawing.Point(184, 50)
599     Me.TxtThrustOutgoing.Name = "TxtThrustOutgoing"
600     Me.TxtThrustOutgoing.Size = New System.Drawing.Size(328, 20)
601     Me.TxtThrustOutgoing.TabIndex = 7
602     '
603     'LblThrustOutgoing
604     '
605     Me.LblThrustOutgoing.Location = New System.Drawing.Point(8, 49)
606     Me.LblThrustOutgoing.Name = "LblThrustOutgoing"
607     Me.LblThrustOutgoing.Size = New System.Drawing.Size(152, 34)
608     Me.LblThrustOutgoing.TabIndex = 6
609     Me.LblThrustOutgoing.Text = "Όνομα Αρχείου (Εξερχόμενων Δεδομένων)"
610     '
611     'BtThrustIncoming
612     '
613     Me.BtThrustIncoming.Image = CType(resources.GetObject("BtThrustIncoming.Image"
), System.Drawing.Image)
614     Me.BtThrustIncoming.ImeMode = System.Windows.Forms.ImeMode.NoControl
615     Me.BtThrustIncoming.Location = New System.Drawing.Point(524, 14)
616     Me.BtThrustIncoming.Name = "BtThrustIncoming"
617     Me.BtThrustIncoming.Size = New System.Drawing.Size(24, 23)
618     Me.BtThrustIncoming.TabIndex = 5
619     '
620     'TxtThrustIncoming
621     '
622     Me.TxtThrustIncoming.Location = New System.Drawing.Point(184, 17)
623     Me.TxtThrustIncoming.Name = "TxtThrustIncoming"
624     Me.TxtThrustIncoming.Size = New System.Drawing.Size(328, 20)
625     Me.TxtThrustIncoming.TabIndex = 4
626     '
627     'LblThrustIncoming
628     '
629     Me.LblThrustIncoming.Location = New System.Drawing.Point(8, 16)
630     Me.LblThrustIncoming.Name = "LblThrustIncoming"
631     Me.LblThrustIncoming.Size = New System.Drawing.Size(170, 33)
632     Me.LblThrustIncoming.TabIndex = 0
633     Me.LblThrustIncoming.Text = "Όνομα Αρχείου (Εισερχόμενων Δεδομένων)"
634     '
635     'TCMain
636     '
637     Me.TCMain.Controls.Add(Me.TPLanguage)
638     Me.TCMain.Controls.Add(Me.TPSerialCommunication)
639     Me.TCMain.Controls.Add(Me.TPThrust)
640     Me.TCMain.Controls.Add(Me.TPGraphs)
641     Me.TCMain.Location = New System.Drawing.Point(8, 63)
642     Me.TCMain.Name = "TCMain"
643     Me.TCMain.SelectedIndex = 0
644     Me.TCMain.Size = New System.Drawing.Size(637, 461)
645     Me.TCMain.TabIndex = 10
646     '
647     'TPLanguage
648     '
649     Me.TPLanguage.Controls.Add(Me.GrBLangs)
650     Me.TPLanguage.Location = New System.Drawing.Point(4, 22)
651     Me.TPLanguage.Name = "TPLanguage"

```

```

652     Me.TPLanguage.Size = New System.Drawing.Size(629, 435)
653     Me.TPLanguage.TabIndex = 2
654     Me.TPLanguage.Text = "Προεπιλεγμένη Γλώσσα"
655     Me.TPLanguage.UseVisualStyleBackColor = True
656     '
657     'TPSerialCommunication
658     '
659     Me.TPSerialCommunication.Controls.Add(Me.GrBSaveOutputData)
660     Me.TPSerialCommunication.Controls.Add(Me.GrBSaveInputData)
661     Me.TPSerialCommunication.Controls.Add(Me.GrBSendFile)
662     Me.TPSerialCommunication.Controls.Add(Me.GrBConParams)
663     Me.TPSerialCommunication.Location = New System.Drawing.Point(4, 22)
664     Me.TPSerialCommunication.Name = "TPSerialCommunication"
665     Me.TPSerialCommunication.Padding = New System.Windows.Forms.Padding(3)
666     Me.TPSerialCommunication.Size = New System.Drawing.Size(629, 435)
667     Me.TPSerialCommunication.TabIndex = 0
668     Me.TPSerialCommunication.Text = "Σειριακή Επικοινωνία"
669     Me.TPSerialCommunication.UseVisualStyleBackColor = True
670     '
671     'TPThrust
672     '
673     Me.TPThrust.Controls.Add(Me.GrBThrust)
674     Me.TPThrust.Location = New System.Drawing.Point(4, 22)
675     Me.TPThrust.Name = "TPThrust"
676     Me.TPThrust.Padding = New System.Windows.Forms.Padding(3)
677     Me.TPThrust.Size = New System.Drawing.Size(629, 435)
678     Me.TPThrust.TabIndex = 1
679     Me.TPThrust.Text = "Κλίνη Ωσης και Ελέγχου Στροφών (RPM)"
680     Me.TPThrust.UseVisualStyleBackColor = True
681     '
682     'TPGraphs
683     '
684     Me.TPGraphs.Controls.Add(Me.GBDefaultGraph)
685     Me.TPGraphs.Controls.Add(Me.TabControl1)
686     Me.TPGraphs.Location = New System.Drawing.Point(4, 22)
687     Me.TPGraphs.Name = "TPGraphs"
688     Me.TPGraphs.Size = New System.Drawing.Size(629, 435)
689     Me.TPGraphs.TabIndex = 3
690     Me.TPGraphs.Text = "Γραφήματα"
691     Me.TPGraphs.UseVisualStyleBackColor = True
692     '
693     'GBDefaultGraph
694     '
695     Me.GBDefaultGraph.Controls.Add(Me.RBExcelGraph)
696     Me.GBDefaultGraph.Controls.Add(Me.RBNetGraph)
697     Me.GBDefaultGraph.Location = New System.Drawing.Point(92, 15)
698     Me.GBDefaultGraph.Name = "GBDefaultGraph"
699     Me.GBDefaultGraph.Size = New System.Drawing.Size(447, 38)
700     Me.GBDefaultGraph.TabIndex = 1
701     Me.GBDefaultGraph.TabStop = False
702     Me.GBDefaultGraph.Text = "Επιλογή Προκαθορισμένου Γραφήματος"
703     '
704     'RBExcelGraph
705     '
706     Me.RBExcelGraph.AutoSize = True
707     Me.RBExcelGraph.Location = New System.Drawing.Point(208, 16)
708     Me.RBExcelGraph.Name = "RBExcelGraph"
709     Me.RBExcelGraph.Size = New System.Drawing.Size(113, 17)
710     Me.RBExcelGraph.TabIndex = 1
711     Me.RBExcelGraph.TabStop = True
712     Me.RBExcelGraph.Text = "Excel Γραφήματα"
713     Me.RBExcelGraph.UseVisualStyleBackColor = True
714     '
715     'RBNetGraph
716     '
717     Me.RBNetGraph.AutoSize = True
718     Me.RBNetGraph.Checked = True
719     Me.RBNetGraph.Location = New System.Drawing.Point(40, 16)
720     Me.RBNetGraph.Name = "RBNetGraph"
721     Me.RBNetGraph.Size = New System.Drawing.Size(112, 17)
722     Me.RBNetGraph.TabIndex = 0
723     Me.RBNetGraph.TabStop = True
724     Me.RBNetGraph.Text = ".NET Γραφήματα"
725     Me.RBNetGraph.UseVisualStyleBackColor = True
726     '

```

```

727         'TabControl1
728         '
729         Me.TabControl1.Controls.Add(Me.TPNetGraphs)
730         Me.TabControl1.Controls.Add(Me.TPExcelGraphs)
731         Me.TabControl1.Controls.Add(Me.TPGraphRefreshRate)
732         Me.TabControl1.Location = New System.Drawing.Point(15, 59)
733         Me.TabControl1.Name = "TabControl1"
734         Me.TabControl1.SelectedIndex = 0
735         Me.TabControl1.Size = New System.Drawing.Size(604, 373)
736         Me.TabControl1.TabIndex = 0
737         '
738         'TPNetGraphs
739         '
740         Me.TPNetGraphs.Controls.Add(Me.GBNetGraphProps2)
741         Me.TPNetGraphs.Controls.Add(Me.GBNetGraphProps1)
742         Me.TPNetGraphs.Location = New System.Drawing.Point(4, 22)
743         Me.TPNetGraphs.Name = "TPNetGraphs"
744         Me.TPNetGraphs.Padding = New System.Windows.Forms.Padding(3)
745         Me.TPNetGraphs.Size = New System.Drawing.Size(596, 347)
746         Me.TPNetGraphs.TabIndex = 0
747         Me.TPNetGraphs.Text = ".NET Γραφήματα"
748         Me.TPNetGraphs.UseVisualStyleBackColor = True
749         '
750         'GBNetGraphProps2
751         '
752         Me.GBNetGraphProps2.Controls.Add(Me.NUD1NetC2HorlDiv)
753         Me.GBNetGraphProps2.Controls.Add(Me.NUDNetC2TimeVertInterval)
754         Me.GBNetGraphProps2.Controls.Add(Me.DUDNetC2VerticalLine)
755         Me.GBNetGraphProps2.Controls.Add(Me.DudNetC2TypeLines)
756         Me.GBNetGraphProps2.Controls.Add(Me.LblNetC2HorlDiv)
757         Me.GBNetGraphProps2.Controls.Add(Me.ChkNetC2Ampere)
758         Me.GBNetGraphProps2.Controls.Add(Me.ChkNetC2Voltage)
759         Me.GBNetGraphProps2.Controls.Add(Me.ChkNetC2LoadCell)
760         Me.GBNetGraphProps2.Controls.Add(Me.ChkNetC2PWMLoad)
761         Me.GBNetGraphProps2.Controls.Add(Me.LblNetC2HorLines)
762         Me.GBNetGraphProps2.Controls.Add(Me.NUDNetC2HorLines)
763         Me.GBNetGraphProps2.Controls.Add(Me.LblNetC2NumOfPoints)
764         Me.GBNetGraphProps2.Controls.Add(Me.NUD1NetC2NumOfPoints)
765         Me.GBNetGraphProps2.Controls.Add(Me.NUDNetC2ConstVerLines)
766         Me.GBNetGraphProps2.Location = New System.Drawing.Point(18, 188)
767         Me.GBNetGraphProps2.Name = "GBNetGraphProps2"
768         Me.GBNetGraphProps2.Size = New System.Drawing.Size(568, 147)
769         Me.GBNetGraphProps2.TabIndex = 58
770         Me.GBNetGraphProps2.TabStop = False
771         Me.GBNetGraphProps2.Text = "Chart2 Properties"
772         '
773         'NUD1NetC2HorlDiv
774         '
775         Me.NUD1NetC2HorlDiv.BackColor = System.Drawing.SystemColors.Window
776         Me.NUD1NetC2HorlDiv.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.0
! , System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(161,
Byte))
777         Me.NUD1NetC2HorlDiv.Location = New System.Drawing.Point(489, 18)
778         Me.NUD1NetC2HorlDiv.Name = "NUD1NetC2HorlDiv"
779         Me.NUD1NetC2HorlDiv.Size = New System.Drawing.Size(46, 21)
780         Me.NUD1NetC2HorlDiv.TabIndex = 80
781         '
782         'NUDNetC2TimeVertInterval
783         '
784         Me.NUDNetC2TimeVertInterval.Location = New System.Drawing.Point(137, 64)
785         Me.NUDNetC2TimeVertInterval.Maximum = New Decimal(New Integer() {100000, 0, 0,
0})
786         Me.NUDNetC2TimeVertInterval.Minimum = New Decimal(New Integer() {20, 0, 0, 0})
787         Me.NUDNetC2TimeVertInterval.Name = "NUDNetC2TimeVertInterval"
788         Me.NUDNetC2TimeVertInterval.Size = New System.Drawing.Size(80, 20)
789         Me.NUDNetC2TimeVertInterval.TabIndex = 79
790         Me.NUDNetC2TimeVertInterval.Value = New Decimal(New Integer() {20, 0, 0, 0})
791         '
792         'DUDNetC2VerticalLine
793         '
794         Me.DUDNetC2VerticalLine.BackColor = System.Drawing.SystemColors.Window
795         Me.DUDNetC2VerticalLine.Location = New System.Drawing.Point(9, 64)
796         Me.DUDNetC2VerticalLine.Name = "DUDNetC2VerticalLine"
797         Me.DUDNetC2VerticalLine.ReadOnly = True
798         Me.DUDNetC2VerticalLine.Size = New System.Drawing.Size(106, 20)

```



```

799         Me.DUDNetC2VerticalLine.TabIndex = 78
800         Me.DUDNetC2VerticalLine.Text = "DomainUpDown1"
801         '
802         'DudNetC2TypeLines
803         '
804         Me.DudNetC2TypeLines.BackColor = System.Drawing.SystemColors.Window
805         Me.DudNetC2TypeLines.Location = New System.Drawing.Point(241, 64)
806         Me.DudNetC2TypeLines.Name = "DudNetC2TypeLines"
807         Me.DudNetC2TypeLines.ReadOnly = True
808         Me.DudNetC2TypeLines.Size = New System.Drawing.Size(86, 20)
809         Me.DudNetC2TypeLines.TabIndex = 77
810         Me.DudNetC2TypeLines.Text = "Type of Lines"
811         '
812         'LblNetC2HorlDiv
813         '
814         Me.LblNetC2HorlDiv.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.0!
, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(161,
Byte))
815         Me.LblNetC2HorlDiv.Location = New System.Drawing.Point(392, 12)
816         Me.LblNetC2HorlDiv.Name = "LblNetC2HorlDiv"
817         Me.LblNetC2HorlDiv.Size = New System.Drawing.Size(96, 30)
818         Me.LblNetC2HorlDiv.TabIndex = 76
819         Me.LblNetC2HorlDiv.Text = "H. Division"
820         Me.LblNetC2HorlDiv.TextAlign = System.Drawing.ContentAlignment.MiddleCenter
821         '
822         'ChkNetC2Ampere
823         '
824         Me.ChkNetC2Ampere.AutoSize = True
825         Me.ChkNetC2Ampere.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.0!
, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(161,
Byte))
826         Me.ChkNetC2Ampere.Location = New System.Drawing.Point(323, 99)
827         Me.ChkNetC2Ampere.Name = "ChkNetC2Ampere"
828         Me.ChkNetC2Ampere.Size = New System.Drawing.Size(69, 19)
829         Me.ChkNetC2Ampere.TabIndex = 75
830         Me.ChkNetC2Ampere.Text = "Ampere"
831         Me.ChkNetC2Ampere.UseVisualStyleBackColor = True
832         '
833         'ChkNetC2Voltage
834         '
835         Me.ChkNetC2Voltage.AutoSize = True
836         Me.ChkNetC2Voltage.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.0!
, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(161,
Byte))
837         Me.ChkNetC2Voltage.Location = New System.Drawing.Point(118, 99)
838         Me.ChkNetC2Voltage.Name = "ChkNetC2Voltage"
839         Me.ChkNetC2Voltage.Size = New System.Drawing.Size(67, 19)
840         Me.ChkNetC2Voltage.TabIndex = 74
841         Me.ChkNetC2Voltage.Text = "Voltage"
842         Me.ChkNetC2Voltage.UseVisualStyleBackColor = True
843         '
844         'ChkNetC2LoadCell
845         '
846         Me.ChkNetC2LoadCell.AutoSize = True
847         Me.ChkNetC2LoadCell.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.0
!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(161,
Byte))
848         Me.ChkNetC2LoadCell.Location = New System.Drawing.Point(211, 99)
849         Me.ChkNetC2LoadCell.Name = "ChkNetC2LoadCell"
850         Me.ChkNetC2LoadCell.Size = New System.Drawing.Size(78, 19)
851         Me.ChkNetC2LoadCell.TabIndex = 72
852         Me.ChkNetC2LoadCell.Text = "LoadCell "
853         Me.ChkNetC2LoadCell.UseVisualStyleBackColor = True
854         '
855         'ChkNetC2PWMLoad
856         '
857         Me.ChkNetC2PWMLoad.AutoSize = True
858         Me.ChkNetC2PWMLoad.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.0!
, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(161,
Byte))
859         Me.ChkNetC2PWMLoad.Location = New System.Drawing.Point(7, 99)
860         Me.ChkNetC2PWMLoad.Name = "ChkNetC2PWMLoad"
861         Me.ChkNetC2PWMLoad.Size = New System.Drawing.Size(82, 19)
862         Me.ChkNetC2PWMLoad.TabIndex = 73
863         Me.ChkNetC2PWMLoad.Text = "PwmLoad"

```

```

864         Me.ChkNetC2PWMLoad.UseVisualStyleBackColor = True
865         '
866         'LblNetC2HorLines
867         '
868         Me.LblNetC2HorLines.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.0!
!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(161,
Byte))
869         Me.LblNetC2HorLines.Location = New System.Drawing.Point(194, 12)
870         Me.LblNetC2HorLines.Name = "LblNetC2HorLines"
871         Me.LblNetC2HorLines.Size = New System.Drawing.Size(96, 30)
872         Me.LblNetC2HorLines.TabIndex = 71
873         Me.LblNetC2HorLines.Text = "Horizontal Lines"
874         Me.LblNetC2HorLines.TextAlign = System.Drawing.ContentAlignment.MiddleCenter
875         '
876         'NUDNetC2HorLines
877         '
878         Me.NUDNetC2HorLines.BackColor = System.Drawing.SystemColors.Window
879         Me.NUDNetC2HorLines.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.0!
!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(161,
Byte))
880         Me.NUDNetC2HorLines.Location = New System.Drawing.Point(307, 18)
881         Me.NUDNetC2HorLines.Name = "NUDNetC2HorLines"
882         Me.NUDNetC2HorLines.Size = New System.Drawing.Size(72, 21)
883         Me.NUDNetC2HorLines.TabIndex = 70
884         '
885         'LblNetC2NumOfPoints
886         '
887         Me.LblNetC2NumOfPoints.Font = New System.Drawing.Font("Microsoft Sans Serif",
9.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType
(161, Byte))
888         Me.LblNetC2NumOfPoints.Location = New System.Drawing.Point(6, 18)
889         Me.LblNetC2NumOfPoints.Name = "LblNetC2NumOfPoints"
890         Me.LblNetC2NumOfPoints.Size = New System.Drawing.Size(96, 30)
891         Me.LblNetC2NumOfPoints.TabIndex = 69
892         Me.LblNetC2NumOfPoints.Text = "NumofPoints"
893         Me.LblNetC2NumOfPoints.TextAlign = System.Drawing.ContentAlignment.
MiddleCenter
894         '
895         'NUD1NetC2NumOfPoints
896         '
897         Me.NUD1NetC2NumOfPoints.BackColor = System.Drawing.SystemColors.Window
898         Me.NUD1NetC2NumOfPoints.Font = New System.Drawing.Font("Microsoft Sans Serif",
9.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType
(161, Byte))
899         Me.NUD1NetC2NumOfPoints.Location = New System.Drawing.Point(113, 18)
900         Me.NUD1NetC2NumOfPoints.Maximum = New Decimal(New Integer() {10000, 0, 0, 0})
901         Me.NUD1NetC2NumOfPoints.Name = "NUD1NetC2NumOfPoints"
902         Me.NUD1NetC2NumOfPoints.Size = New System.Drawing.Size(70, 21)
903         Me.NUD1NetC2NumOfPoints.TabIndex = 68
904         '
905         'NUDNetC2ConstVerLines
906         '
907         Me.NUDNetC2ConstVerLines.Location = New System.Drawing.Point(137, 64)
908         Me.NUDNetC2ConstVerLines.Name = "NUDNetC2ConstVerLines"
909         Me.NUDNetC2ConstVerLines.Size = New System.Drawing.Size(80, 20)
910         Me.NUDNetC2ConstVerLines.TabIndex = 81
911         '
912         'GBNetGraphProps1
913         '
914         Me.GBNetGraphProps1.Controls.Add(Me.DUDNetC1VerticalLine)
915         Me.GBNetGraphProps1.Controls.Add(Me.LblNetC1HorlDiv)
916         Me.GBNetGraphProps1.Controls.Add(Me.NUDNetC1HorlDiv)
917         Me.GBNetGraphProps1.Controls.Add(Me.DudNetC1TypeLines)
918         Me.GBNetGraphProps1.Controls.Add(Me.ChkNetC1SetPointPwm)
919         Me.GBNetGraphProps1.Controls.Add(Me.ChkNetC1PWMLoad)
920         Me.GBNetGraphProps1.Controls.Add(Me.ChkNetC1SetPointRPM)
921         Me.GBNetGraphProps1.Controls.Add(Me.ChkNetC1Rpm)
922         Me.GBNetGraphProps1.Controls.Add(Me.LblNetC1HorLines)
923         Me.GBNetGraphProps1.Controls.Add(Me.NUDNetC1HorLines)
924         Me.GBNetGraphProps1.Controls.Add(Me.LblNetC1NumOfPoints)
925         Me.GBNetGraphProps1.Controls.Add(Me.NUDNetC1NumOfPoints)
926         Me.GBNetGraphProps1.Controls.Add(Me.NUDNetC1TimeVertInterval)
927         Me.GBNetGraphProps1.Controls.Add(Me.NUDNetC1ConstVerLines)
928         Me.GBNetGraphProps1.Location = New System.Drawing.Point(18, 15)
929         Me.GBNetGraphProps1.Name = "GBNetGraphProps1"

```

```

930     Me.GBNetGraphProps1.Size = New System.Drawing.Size(568, 167)
931     Me.GBNetGraphProps1.TabIndex = 57
932     Me.GBNetGraphProps1.TabStop = False
933     Me.GBNetGraphProps1.Text = "Chart1 Properties"
934     '
935     'DUDNetC1VerticalLine
936     '
937     Me.DUDNetC1VerticalLine.BackColor = System.Drawing.SystemColors.Window
938     Me.DUDNetC1VerticalLine.Location = New System.Drawing.Point(9, 80)
939     Me.DUDNetC1VerticalLine.Name = "DUDNetC1VerticalLine"
940     Me.DUDNetC1VerticalLine.ReadOnly = True
941     Me.DUDNetC1VerticalLine.Size = New System.Drawing.Size(106, 20)
942     Me.DUDNetC1VerticalLine.TabIndex = 78
943     Me.DUDNetC1VerticalLine.Text = "DomainUpDown1"
944     '
945     'LblNetC1HorlDiv
946     '
947     Me.LblNetC1HorlDiv.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.0!
, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(161,
Byte))
948     Me.LblNetC1HorlDiv.Location = New System.Drawing.Point(392, 25)
949     Me.LblNetC1HorlDiv.Name = "LblNetC1HorlDiv"
950     Me.LblNetC1HorlDiv.Size = New System.Drawing.Size(96, 30)
951     Me.LblNetC1HorlDiv.TabIndex = 77
952     Me.LblNetC1HorlDiv.Text = "H. Division"
953     Me.LblNetC1HorlDiv.TextAlign = System.Drawing.ContentAlignment.MiddleCenter
954     '
955     'NUDNetC1HorlDiv
956     '
957     Me.NUDNetC1HorlDiv.BackColor = System.Drawing.SystemColors.Window
958     Me.NUDNetC1HorlDiv.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.0!
, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(161,
Byte))
959     Me.NUDNetC1HorlDiv.Location = New System.Drawing.Point(489, 31)
960     Me.NUDNetC1HorlDiv.Name = "NUDNetC1HorlDiv"
961     Me.NUDNetC1HorlDiv.Size = New System.Drawing.Size(46, 21)
962     Me.NUDNetC1HorlDiv.TabIndex = 76
963     '
964     'DudNetC1TypeLines
965     '
966     Me.DudNetC1TypeLines.BackColor = System.Drawing.SystemColors.Window
967     Me.DudNetC1TypeLines.Items.Add("Dushed Lines")
968     Me.DudNetC1TypeLines.Items.Add("Solid Lines")
969     Me.DudNetC1TypeLines.Location = New System.Drawing.Point(241, 80)
970     Me.DudNetC1TypeLines.Name = "DudNetC1TypeLines"
971     Me.DudNetC1TypeLines.ReadOnly = True
972     Me.DudNetC1TypeLines.Size = New System.Drawing.Size(74, 20)
973     Me.DudNetC1TypeLines.TabIndex = 75
974     '
975     'ChkNetC1SetPointPwm
976     '
977     Me.ChkNetC1SetPointPwm.AutoSize = True
978     Me.ChkNetC1SetPointPwm.Font = New System.Drawing.Font("Microsoft Sans Serif",
9.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(
161, Byte))
979     Me.ChkNetC1SetPointPwm.Location = New System.Drawing.Point(295, 129)
980     Me.ChkNetC1SetPointPwm.Name = "ChkNetC1SetPointPwm"
981     Me.ChkNetC1SetPointPwm.Size = New System.Drawing.Size(129, 19)
982     Me.ChkNetC1SetPointPwm.TabIndex = 74
983     Me.ChkNetC1SetPointPwm.Text = "Setpoint Pwn Load"
984     Me.ChkNetC1SetPointPwm.UseVisualStyleBackColor = True
985     '
986     'ChkNetC1PWMLoad
987     '
988     Me.ChkNetC1PWMLoad.AutoSize = True
989     Me.ChkNetC1PWMLoad.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.0!
, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(161,
Byte))
990     Me.ChkNetC1PWMLoad.Location = New System.Drawing.Point(189, 129)
991     Me.ChkNetC1PWMLoad.Name = "ChkNetC1PWMLoad"
992     Me.ChkNetC1PWMLoad.Size = New System.Drawing.Size(85, 19)
993     Me.ChkNetC1PWMLoad.TabIndex = 73
994     Me.ChkNetC1PWMLoad.Text = "Pwm Load"
995     Me.ChkNetC1PWMLoad.UseVisualStyleBackColor = True
996     '

```

```

997         'ChkNetC1SetPointRPM
998         '
999         Me.ChkNetC1SetPointRPM.AutoSize = True
1000        Me.ChkNetC1SetPointRPM.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType
(161, Byte))
1001        Me.ChkNetC1SetPointRPM.Location = New System.Drawing.Point(81, 129)
1002        Me.ChkNetC1SetPointRPM.Name = "ChkNetC1SetPointRPM"
1003        Me.ChkNetC1SetPointRPM.Size = New System.Drawing.Size(102, 19)
1004        Me.ChkNetC1SetPointRPM.TabIndex = 71
1005        Me.ChkNetC1SetPointRPM.Text = "Setpoint RPM"
1006        Me.ChkNetC1SetPointRPM.UseVisualStyleBackColor = True
1007        '
1008        'ChkNetC1Rpm
1009        '
1010        Me.ChkNetC1Rpm.AutoSize = True
1011        Me.ChkNetC1Rpm.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType
(161, Byte))
1012        Me.ChkNetC1Rpm.Location = New System.Drawing.Point(7, 129)
1013        Me.ChkNetC1Rpm.Name = "ChkNetC1Rpm"
1014        Me.ChkNetC1Rpm.Size = New System.Drawing.Size(54, 19)
1015        Me.ChkNetC1Rpm.TabIndex = 72
1016        Me.ChkNetC1Rpm.Text = "RPM"
1017        Me.ChkNetC1Rpm.UseVisualStyleBackColor = True
1018        '
1019        'LblNetC1HorLines
1020        '
1021        Me.LblNetC1HorLines.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType
(161, Byte))
1022        Me.LblNetC1HorLines.Location = New System.Drawing.Point(194, 25)
1023        Me.LblNetC1HorLines.Name = "LblNetC1HorLines"
1024        Me.LblNetC1HorLines.Size = New System.Drawing.Size(96, 30)
1025        Me.LblNetC1HorLines.TabIndex = 70
1026        Me.LblNetC1HorLines.Text = "Horizontal Lines"
1027        Me.LblNetC1HorLines.TextAlign = System.Drawing.ContentAlignment.MiddleCenter
1028        '
1029        'NUDNetC1HorLines
1030        '
1031        Me.NUDNetC1HorLines.BackColor = System.Drawing.SystemColors.Window
1032        Me.NUDNetC1HorLines.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType
(161, Byte))
1033        Me.NUDNetC1HorLines.Location = New System.Drawing.Point(307, 31)
1034        Me.NUDNetC1HorLines.Name = "NUDNetC1HorLines"
1035        Me.NUDNetC1HorLines.Size = New System.Drawing.Size(72, 21)
1036        Me.NUDNetC1HorLines.TabIndex = 69
1037        '
1038        'LblNetC1NumOfPoints
1039        '
1040        Me.LblNetC1NumOfPoints.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType
(161, Byte))
1041        Me.LblNetC1NumOfPoints.Location = New System.Drawing.Point(6, 25)
1042        Me.LblNetC1NumOfPoints.Name = "LblNetC1NumOfPoints"
1043        Me.LblNetC1NumOfPoints.Size = New System.Drawing.Size(96, 30)
1044        Me.LblNetC1NumOfPoints.TabIndex = 68
1045        Me.LblNetC1NumOfPoints.Text = "Num of Points"
1046        Me.LblNetC1NumOfPoints.TextAlign = System.Drawing.ContentAlignment.MiddleCenter
1047        '
1048        'NUDNetC1NumOfPoints
1049        '
1050        Me.NUDNetC1NumOfPoints.BackColor = System.Drawing.SystemColors.Window
1051        Me.NUDNetC1NumOfPoints.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType
(161, Byte))
1052        Me.NUDNetC1NumOfPoints.Location = New System.Drawing.Point(113, 31)
1053        Me.NUDNetC1NumOfPoints.Maximum = New Decimal(New Integer() {10000, 0, 0, 0})
1054        Me.NUDNetC1NumOfPoints.Name = "NUDNetC1NumOfPoints"
1055        Me.NUDNetC1NumOfPoints.Size = New System.Drawing.Size(68, 21)
1056        Me.NUDNetC1NumOfPoints.TabIndex = 67
1057        '
1058        'NUDNetC1TimeVertInterval

```

```

1059     '
1060     Me.NUDNetC1TimeVertInterval.Location = New System.Drawing.Point(137, 80)
1061     Me.NUDNetC1TimeVertInterval.Maximum = New Decimal(New Integer() {100000, 0, 0, 0}
0))
1062     Me.NUDNetC1TimeVertInterval.Minimum = New Decimal(New Integer() {20, 0, 0, 0})
1063     Me.NUDNetC1TimeVertInterval.Name = "NUDNetC1TimeVertInterval"
1064     Me.NUDNetC1TimeVertInterval.Size = New System.Drawing.Size(80, 20)
1065     Me.NUDNetC1TimeVertInterval.TabIndex = 79
1066     Me.NUDNetC1TimeVertInterval.Value = New Decimal(New Integer() {20, 0, 0, 0})
1067     '
1068     'NUDNetC1ConstVerLines
1069     '
1070     Me.NUDNetC1ConstVerLines.Location = New System.Drawing.Point(137, 80)
1071     Me.NUDNetC1ConstVerLines.Name = "NUDNetC1ConstVerLines"
1072     Me.NUDNetC1ConstVerLines.Size = New System.Drawing.Size(80, 20)
1073     Me.NUDNetC1ConstVerLines.TabIndex = 80
1074     '
1075     'TPExcelGraphs
1076     '
1077     Me.TPEExcelGraphs.Controls.Add(Me.PanelChartProperties)
1078     Me.TPEExcelGraphs.Location = New System.Drawing.Point(4, 22)
1079     Me.TPEExcelGraphs.Name = "TPEExcelGraphs"
1080     Me.TPEExcelGraphs.Padding = New System.Windows.Forms.Padding(3)
1081     Me.TPEExcelGraphs.Size = New System.Drawing.Size(596, 347)
1082     Me.TPEExcelGraphs.TabIndex = 1
1083     Me.TPEExcelGraphs.Text = "Excel Γραφήματα"
1084     Me.TPEExcelGraphs.UseVisualStyleBackColor = True
1085     '
1086     'PanelChartProperties
1087     '
1088     Me.PanelChartProperties.Controls.Add(Me.GbExcelChart2Props)
1089     Me.PanelChartProperties.Controls.Add(Me.GbExcelChart1Props)
1090     Me.PanelChartProperties.Location = New System.Drawing.Point(20, 49)
1091     Me.PanelChartProperties.Name = "PanelChartProperties"
1092     Me.PanelChartProperties.Size = New System.Drawing.Size(549, 239)
1093     Me.PanelChartProperties.TabIndex = 57
1094     '
1095     'GbExcelChart2Props
1096     '
1097     Me.GbExcelChart2Props.Controls.Add(Me.LblExcelC2SelectLine)
1098     Me.GbExcelChart2Props.Controls.Add(Me.LbExcelC2ChartPoints)
1099     Me.GbExcelChart2Props.Controls.Add(Me.NUDEExcelC2ChartPoints)
1100     Me.GbExcelChart2Props.Controls.Add(Me.DUBExcelC2SelectLine)
1101     Me.GbExcelChart2Props.Location = New System.Drawing.Point(51, 134)
1102     Me.GbExcelChart2Props.Name = "GbExcelChart2Props"
1103     Me.GbExcelChart2Props.Size = New System.Drawing.Size(448, 40)
1104     Me.GbExcelChart2Props.TabIndex = 1
1105     Me.GbExcelChart2Props.TabStop = False
1106     Me.GbExcelChart2Props.Text = "Chart 2 Properties"
1107     '
1108     'LblExcelC2SelectLine
1109     '
1110     Me.LblExcelC2SelectLine.AutoSize = True
1111     Me.LblExcelC2SelectLine.Location = New System.Drawing.Point(12, 16)
1112     Me.LblExcelC2SelectLine.Name = "LblExcelC2SelectLine"
1113     Me.LblExcelC2SelectLine.Size = New System.Drawing.Size(66, 13)
1114     Me.LblExcelC2SelectLine.TabIndex = 4
1115     Me.LblExcelC2SelectLine.Text = "Select Line :"
1116     '
1117     'LbExcelC2ChartPoints
1118     '
1119     Me.LbExcelC2ChartPoints.AutoSize = True
1120     Me.LbExcelC2ChartPoints.Location = New System.Drawing.Point(270, 16)
1121     Me.LbExcelC2ChartPoints.Name = "LbExcelC2ChartPoints"
1122     Me.LbExcelC2ChartPoints.Size = New System.Drawing.Size(64, 13)
1123     Me.LbExcelC2ChartPoints.TabIndex = 3
1124     Me.LbExcelC2ChartPoints.Text = "Chart Points"
1125     '
1126     'NUDEExcelC2ChartPoints
1127     '
1128     Me.NUDEExcelC2ChartPoints.Location = New System.Drawing.Point(362, 14)
1129     Me.NUDEExcelC2ChartPoints.Minimum = New Decimal(New Integer() {2, 0, 0, 0})
1130     Me.NUDEExcelC2ChartPoints.Name = "NUDEExcelC2ChartPoints"
1131     Me.NUDEExcelC2ChartPoints.Size = New System.Drawing.Size(60, 20)
1132     Me.NUDEExcelC2ChartPoints.TabIndex = 2

```

```

1133 Me.NUDExcelC2ChartPoints.Value = New Decimal(New Integer() {2, 0, 0, 0})
1134 '
1135 'DUBExcelC2SelectLine
1136 '
1137 Me.DUBExcelC2SelectLine.Items.Add("RPM")
1138 Me.DUBExcelC2SelectLine.Items.Add("PWM Load")
1139 Me.DUBExcelC2SelectLine.Items.Add("Load Cell")
1140 Me.DUBExcelC2SelectLine.Items.Add("Amperes")
1141 Me.DUBExcelC2SelectLine.Items.Add("Voltage")
1142 Me.DUBExcelC2SelectLine.Items.Add("Setpoint RPM")
1143 Me.DUBExcelC2SelectLine.Items.Add("Setpoint PWM")
1144 Me.DUBExcelC2SelectLine.Location = New System.Drawing.Point(116, 12)
1145 Me.DUBExcelC2SelectLine.Name = "DUBExcelC2SelectLine"
1146 Me.DUBExcelC2SelectLine.Size = New System.Drawing.Size(120, 20)
1147 Me.DUBExcelC2SelectLine.TabIndex = 1
1148 '
1149 'GbExcelChart1Props
1150 '
1151 Me.GbExcelChart1Props.Controls.Add(Me.LblExcelC1SelectLine)
1152 Me.GbExcelChart1Props.Controls.Add(Me.LbExcelC1ChartPoints)
1153 Me.GbExcelChart1Props.Controls.Add(Me.NUDExcelC1ChartPoints)
1154 Me.GbExcelChart1Props.Controls.Add(Me.DUBExcelC1SelectLine)
1155 Me.GbExcelChart1Props.Location = New System.Drawing.Point(51, 48)
1156 Me.GbExcelChart1Props.Name = "GbExcelChart1Props"
1157 Me.GbExcelChart1Props.Size = New System.Drawing.Size(448, 40)
1158 Me.GbExcelChart1Props.TabIndex = 0
1159 Me.GbExcelChart1Props.TabStop = False
1160 Me.GbExcelChart1Props.Text = "Chart1 Properties"
1161 '
1162 'LblExcelC1SelectLine
1163 '
1164 Me.LblExcelC1SelectLine.AutoSize = True
1165 Me.LblExcelC1SelectLine.Location = New System.Drawing.Point(12, 19)
1166 Me.LblExcelC1SelectLine.Name = "LblExcelC1SelectLine"
1167 Me.LblExcelC1SelectLine.Size = New System.Drawing.Size(66, 13)
1168 Me.LblExcelC1SelectLine.TabIndex = 3
1169 Me.LblExcelC1SelectLine.Text = "Select Line : "
1170 '
1171 'LbExcelC1ChartPoints
1172 '
1173 Me.LbExcelC1ChartPoints.AutoSize = True
1174 Me.LbExcelC1ChartPoints.Location = New System.Drawing.Point(270, 17)
1175 Me.LbExcelC1ChartPoints.Name = "LbExcelC1ChartPoints"
1176 Me.LbExcelC1ChartPoints.Size = New System.Drawing.Size(64, 13)
1177 Me.LbExcelC1ChartPoints.TabIndex = 2
1178 Me.LbExcelC1ChartPoints.Text = "Chart Points"
1179 '
1180 'NUDExcelC1ChartPoints
1181 '
1182 Me.NUDExcelC1ChartPoints.Location = New System.Drawing.Point(362, 14)
1183 Me.NUDExcelC1ChartPoints.Minimum = New Decimal(New Integer() {2, 0, 0, 0})
1184 Me.NUDExcelC1ChartPoints.Name = "NUDExcelC1ChartPoints"
1185 Me.NUDExcelC1ChartPoints.Size = New System.Drawing.Size(63, 20)
1186 Me.NUDExcelC1ChartPoints.TabIndex = 1
1187 Me.NUDExcelC1ChartPoints.Value = New Decimal(New Integer() {2, 0, 0, 0})
1188 '
1189 'DUBExcelC1SelectLine
1190 '
1191 Me.DUBExcelC1SelectLine.Items.Add("RPM")
1192 Me.DUBExcelC1SelectLine.Items.Add("PWM Load")
1193 Me.DUBExcelC1SelectLine.Items.Add("Load Cell")
1194 Me.DUBExcelC1SelectLine.Items.Add("Amperes")
1195 Me.DUBExcelC1SelectLine.Items.Add("Voltage")
1196 Me.DUBExcelC1SelectLine.Location = New System.Drawing.Point(116, 14)
1197 Me.DUBExcelC1SelectLine.Name = "DUBExcelC1SelectLine"
1198 Me.DUBExcelC1SelectLine.Size = New System.Drawing.Size(120, 20)
1199 Me.DUBExcelC1SelectLine.TabIndex = 0
1200 '
1201 'TPGraphRefreshRate
1202 '
1203 Me.TPGraphRefreshRate.Controls.Add(Me.GrBRefreshRateofGraphics)
1204 Me.TPGraphRefreshRate.Location = New System.Drawing.Point(4, 22)
1205 Me.TPGraphRefreshRate.Name = "TPGraphRefreshRate"
1206 Me.TPGraphRefreshRate.Size = New System.Drawing.Size(596, 347)
1207 Me.TPGraphRefreshRate.TabIndex = 2

```

```

1208     Me.TPGraphRefreshRate.Text = "Ρυθμός Ανανέωσης Γραφημάτων"
1209     Me.TPGraphRefreshRate.UseVisualStyleBackColor = True
1210     '
1211     'GrBRefreshRateofGraphics
1212     '
1213     Me.GrBRefreshRateofGraphics.Controls.Add(Me.LbGraphicsRate)
1214     Me.GrBRefreshRateofGraphics.Controls.Add(Me.LbBasicGraphisRate)
1215     Me.GrBRefreshRateofGraphics.Controls.Add(Me.CmbGraphicsRate)
1216     Me.GrBRefreshRateofGraphics.Controls.Add(Me.CmbBasicGraphisRate)
1217     Me.GrBRefreshRateofGraphics.Location = New System.Drawing.Point(104, 94)
1218     Me.GrBRefreshRateofGraphics.Name = "GrBRefreshRateofGraphics"
1219     Me.GrBRefreshRateofGraphics.Size = New System.Drawing.Size(339, 143)
1220     Me.GrBRefreshRateofGraphics.TabIndex = 59
1221     Me.GrBRefreshRateofGraphics.TabStop = False
1222     Me.GrBRefreshRateofGraphics.Text = "Refresh Time Of Graphics"
1223     '
1224     'LbGraphicsRate
1225     '
1226     Me.LbGraphicsRate.Location = New System.Drawing.Point(15, 76)
1227     Me.LbGraphicsRate.Name = "LbGraphicsRate"
1228     Me.LbGraphicsRate.Size = New System.Drawing.Size(176, 15)
1229     Me.LbGraphicsRate.TabIndex = 3
1230     Me.LbGraphicsRate.Text = "Graphics Period"
1231     '
1232     'LbBasicGraphisRate
1233     '
1234     Me.LbBasicGraphisRate.Font = New System.Drawing.Font("Arial", 8.25!, System.
1235 Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(161, Byte))
1236     Me.LbBasicGraphisRate.Location = New System.Drawing.Point(15, 27)
1237     Me.LbBasicGraphisRate.Name = "LbBasicGraphisRate"
1238     Me.LbBasicGraphisRate.Size = New System.Drawing.Size(176, 15)
1239     Me.LbBasicGraphisRate.TabIndex = 2
1240     Me.LbBasicGraphisRate.Text = "Basic Graphics Period"
1241     '
1242     'CmbGraphicsRate
1243     '
1244     Me.CmbGraphicsRate.DropDownStyle = System.Windows.Forms.ComboBoxStyle.
1245 DropDownList
1246     Me.CmbGraphicsRate.Font = New System.Drawing.Font("Arial", 8.25!, System.
1247 Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(161, Byte))
1248     Me.CmbGraphicsRate.FormattingEnabled = True
1249     Me.CmbGraphicsRate.Items.AddRange(New Object() {"50      ms", "100      ms",
1250 "200      ms", "500      ms", "1000 ms", "2000 ms", "3        sec", "5        sec", "7
1251 sec", "10       sec"})
1252     Me.CmbGraphicsRate.Location = New System.Drawing.Point(197, 67)
1253     Me.CmbGraphicsRate.Name = "CmbGraphicsRate"
1254     Me.CmbGraphicsRate.Size = New System.Drawing.Size(121, 22)
1255     Me.CmbGraphicsRate.TabIndex = 1
1256     '
1257     'CmbBasicGraphisRate
1258     '
1259     Me.CmbBasicGraphisRate.DropDownStyle = System.Windows.Forms.ComboBoxStyle.
1260 DropDownList
1261     Me.CmbBasicGraphisRate.FormattingEnabled = True
1262     Me.CmbBasicGraphisRate.Items.AddRange(New Object() {"0        ms", "50       ms",
1263 "100      ms", "200      ms", "500      ms", "1000 ms"})
1264     Me.CmbBasicGraphisRate.Location = New System.Drawing.Point(197, 21)
1265     Me.CmbBasicGraphisRate.Name = "CmbBasicGraphisRate"
1266     Me.CmbBasicGraphisRate.Size = New System.Drawing.Size(121, 21)
1267     Me.CmbBasicGraphisRate.TabIndex = 0
1268     '
1269     'FrmParameters
1270     '
1271     Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
1272     Me.CancelButton = Me.BtBack
1273     Me.ClientSize = New System.Drawing.Size(652, 561)
1274     Me.Controls.Add(Me.TCMain)
1275     Me.Controls.Add(Me.GrBSaveConfFile)
1276     Me.Controls.Add(Me.BtBack)
1277     Me.Controls.Add(Me.BtSave)
1278     Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle
1279     Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)
1280     Me.MaximizeBox = False
1281     Me.MinimizeBox = False
1282     Me.Name = "FrmParameters"

```

```

1276     Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent
1277     Me.Text = "My Parameters"
1278     Me.GrBLangs.ResumeLayout(False)
1279     Me.GrBSaveOutputData.ResumeLayout(False)
1280     Me.GrBSaveOutputData.PerformLayout()
1281     Me.GrBSaveInputData.ResumeLayout(False)
1282     Me.GrBSaveInputData.PerformLayout()
1283     Me.GrBSendFile.ResumeLayout(False)
1284     Me.GrBSendFile.PerformLayout()
1285     Me.GrBConParams.ResumeLayout(False)
1286     Me.GrBSaveConfFile.ResumeLayout(False)
1287     Me.GrBSaveConfFile.PerformLayout()
1288     Me.GrBThrust.ResumeLayout(False)
1289     Me.GrBThrust.PerformLayout()
1290     Me.TCMain.ResumeLayout(False)
1291     Me.TPLanguage.ResumeLayout(False)
1292     Me.TPSerialCommunication.ResumeLayout(False)
1293     Me.TPThrust.ResumeLayout(False)
1294     Me.TPGraphs.ResumeLayout(False)
1295     Me.GBDefaultGraph.ResumeLayout(False)
1296     Me.GBDefaultGraph.PerformLayout()
1297     Me.TabControll.ResumeLayout(False)
1298     Me.TPNetGraphs.ResumeLayout(False)
1299     Me.GBNetGraphProps2.ResumeLayout(False)
1300     Me.GBNetGraphProps2.PerformLayout()
1301     CType(Me.NUD1NetC2HorlDiv, System.ComponentModel.ISupportInitialize).EndInit()
1302     CType(Me.NUDNetC2TimeVertInterval, System.ComponentModel.ISupportInitialize).
1303     EndInit()
1304     CType(Me.NUDNetC2HorLines, System.ComponentModel.ISupportInitialize).EndInit()
1305     CType(Me.NUDNetC2NumOfPoints, System.ComponentModel.ISupportInitialize).
1306     EndInit()
1307     CType(Me.NUDNetC2ConstVerLines, System.ComponentModel.ISupportInitialize).
1308     EndInit()
1309     Me.GBNetGraphProps1.ResumeLayout(False)
1310     Me.GBNetGraphProps1.PerformLayout()
1311     CType(Me.NUDNetC1HorlDiv, System.ComponentModel.ISupportInitialize).EndInit()
1312     CType(Me.NUDNetC1HorLines, System.ComponentModel.ISupportInitialize).EndInit()
1313     CType(Me.NUDNetC1NumOfPoints, System.ComponentModel.ISupportInitialize).
1314     EndInit()
1315     CType(Me.NUDNetC1TimeVertInterval, System.ComponentModel.ISupportInitialize).
1316     EndInit()
1317     CType(Me.NUDNetC1ConstVerLines, System.ComponentModel.ISupportInitialize).
1318     EndInit()
1319     Me.TPExcelGraphs.ResumeLayout(False)
1320     Me.PanelChartProperties.ResumeLayout(False)
1321     Me.GbExcelChart2Props.ResumeLayout(False)
1322     Me.GbExcelChart2Props.PerformLayout()
1323     CType(Me.NUDEExcelC2ChartPoints, System.ComponentModel.ISupportInitialize).
1324     EndInit()
1325     Me.GbExcelChart1Props.ResumeLayout(False)
1326     Me.GbExcelChart1Props.PerformLayout()
1327     CType(Me.NUDEExcelC1ChartPoints, System.ComponentModel.ISupportInitialize).
1328     EndInit()
1329     Me.TPGraphRefreshRate.ResumeLayout(False)
1330     Me.GrBRefreshRateofGraphics.ResumeLayout(False)
1331     Me.ResumeLayout(False)
1332
1333 End Sub
1334
1335 #End Region
1336 Private MyClassCode As New MyPtyxClass
1337 Public ConfsComPort, ConfsBaudRate, ConfsParity, ConfsDataBits, ConfsStopBits,
1338 InitialDirectory As String
1339 Public Occured As Boolean
1340 Private WhichIsTheLang As Integer
1341 Private SavedChanges As Boolean
1342 Public DefaultConfFile As String
1343
1344 Private Sub FrmParameters_Load(ByVal sender As System.Object, ByVal e As System.
1345 EventArgs) Handles MyBase.Load
1346     InitialDirectory = "c:\windows\Desktop"
1347     Me.TxtConfFile.Text = DefaultConfFile
1348     AdjustValuesAccordingToConfFile(DefaultConfFile)
1349     ' TPAdraneia.Hide()
1350 End Sub

```



```

1341     Public Sub WhichLang(ByVal WhichLanguage As Integer)
1342         'WhichLanguage = 0 for Greek
1343         'WhichLanguage = 1 for English
1344         WhichIsTheLang = WhichLanguage
1345         FillComboLangs(WhichLanguage)
1346         AppropriatelyFillComCombo()
1347
1348         GrBSaveConfFile.Text = MyClassCode.GetStr(WhichLanguage, 80)
1349         LblConfFile.Text = MyClassCode.GetStr(WhichLanguage, 81)
1350         GrBLangs.Text = MyClassCode.GetStr(WhichLanguage, 82)
1351         LblDefaultLanguage.Text = MyClassCode.GetStr(WhichLanguage, 83)
1352         ChkShowSelectLang.Text = MyClassCode.GetStr(WhichLanguage, 84)
1353         GrBSaveOutputData.Text = MyClassCode.GetStr(WhichLanguage, 85)
1354         LblOutputFileName.Text = MyClassCode.GetStr(WhichLanguage, 86)
1355         GrBSaveInputData.Text = MyClassCode.GetStr(WhichLanguage, 87)
1356         LblInputFileName.Text = MyClassCode.GetStr(WhichLanguage, 88)
1357         GrBSendFile.Text = MyClassCode.GetStr(WhichLanguage, 89)
1358         LblSendFile.Text = MyClassCode.GetStr(WhichLanguage, 90)
1359         GrBConParams.Text = MyClassCode.GetStr(WhichLanguage, 91)
1360         LblComPort.Text = MyClassCode.GetStr(WhichLanguage, 92)
1361         LblBaudRate.Text = MyClassCode.GetStr(WhichLanguage, 93)
1362         LblParity.Text = MyClassCode.GetStr(WhichLanguage, 94)
1363         LblDataBits.Text = MyClassCode.GetStr(WhichLanguage, 95)
1364         LblStopBits.Text = MyClassCode.GetStr(WhichLanguage, 96)
1365         BtSave.Text = MyClassCode.GetStr(WhichLanguage, 97)
1366         BtBack.Text = MyClassCode.GetStr(WhichLanguage, 98)
1367         Me.Text = MyClassCode.GetStr(WhichLanguage, 99)
1368         GrBThrust.Text = MyClassCode.GetStr(WhichLanguage, 164)
1369         LblThrustIncoming.Text = MyClassCode.GetStr(WhichLanguage, 81) & " (" &
MyClassCode.GetStr(WhichLanguage, 6) & ")"
1370         LblThrustOutgoing.Text = MyClassCode.GetStr(WhichLanguage, 81) & " (" &
MyClassCode.GetStr(WhichLanguage, 7) & ")"
1371         TPLanguage.Text = MyClassCode.GetStr(WhichLanguage, 18)
1372         TPSerialCommunication.Text = MyClassCode.GetStr(WhichLanguage, 145)
1373         TPThrust.Text = MyClassCode.GetStr(WhichLanguage, 149)
1374         TPGraphs.Text = MyClassCode.GetStr(WhichLanguage, 311)
1375         GBDefaultGraph.Text = MyClassCode.GetStr(WhichLanguage, 313)
1376         RBNetGraph.Text = MyClassCode.GetStr(WhichLanguage, 314)
1377         RBExcelGraph.Text = MyClassCode.GetStr(WhichLanguage, 315)
1378         Me.GBNetGraphProps1.Text = MyClassCode.GetStr(WhichIsTheLang, 157)
1379         Me.LblNetC1NumOfPoints.Text = MyClassCode.GetStr(WhichIsTheLang, 174)
1380         Me.LblNetC1HorLines.Text = MyClassCode.GetStr(WhichIsTheLang, 175)
1381         Me.LblNetC1HorlDiv.Text = MyClassCode.GetStr(WhichIsTheLang, 177)
1382         Me.ChkNetC1Rpm.Text = MyClassCode.GetStr(WhichIsTheLang, 187)
1383         Me.ChkNetC1SetPointRPM.Text = MyClassCode.GetStr(WhichIsTheLang, 188)
1384         Me.ChkNetC1PWMLoad.Text = MyClassCode.GetStr(WhichIsTheLang, 189)
1385         Me.ChkNetC1SetPointPwm.Text = MyClassCode.GetStr(WhichIsTheLang, 190)
1386         'for graph2 literals
1387         Me.GBNetGraphProps2.Text = MyClassCode.GetStr(WhichIsTheLang, 172)
1388         Me.LblNetC2NumOfPoints.Text = MyClassCode.GetStr(WhichIsTheLang, 174)
1389         Me.LblNetC2HorLines.Text = MyClassCode.GetStr(WhichIsTheLang, 175)
1390         Me.LblNetC2HorlDiv.Text = MyClassCode.GetStr(WhichIsTheLang, 177)
1391         Me.ChkNetC2PWMLoad.Text = MyClassCode.GetStr(WhichIsTheLang, 178)
1392         Me.ChkNetC2LoadCell.Text = MyClassCode.GetStr(WhichIsTheLang, 179)
1393         Me.ChkNetC2Voltage.Text = MyClassCode.GetStr(WhichIsTheLang, 180)
1394         Me.ChkNetC2Ampere.Text = MyClassCode.GetStr(WhichIsTheLang, 181)
1395         ' for chart 1&2 literals
1396         Me.LbBasicGraphisRate.Text = MyClassCode.GetStr(WhichIsTheLang, 303)
1397         Me.LbGraphicsRate.Text = MyClassCode.GetStr(WhichIsTheLang, 304)
1398
1399         Me.GbExcelChart1Props.Text = MyClassCode.GetStr(WhichIsTheLang, 192)
1400         Me.GbExcelChart2Props.Text = MyClassCode.GetStr(WhichIsTheLang, 193)
1401         Me.LblExcelC1SelectLine.Text = MyClassCode.GetStr(WhichIsTheLang, 194)
1402         Me.LblExcelC2SelectLine.Text = MyClassCode.GetStr(WhichIsTheLang, 194)
1403         Me.LbExcelC1ChartPoints.Text = MyClassCode.GetStr(WhichIsTheLang, 195)
1404         Me.LbExcelC2ChartPoints.Text = MyClassCode.GetStr(WhichIsTheLang, 195)
1405         'for refresh rates
1406
1407         Me.TPGraphRefreshRate.Text = MyClassCode.GetStr(WhichIsTheLang, 305)
1408         Me.TPNetGraphs.Text = MyClassCode.GetStr(WhichIsTheLang, 314)
1409         Me.TPExcelGraphs.Text = MyClassCode.GetStr(WhichIsTheLang, 315)
1410
1411         Me.DudNetC1TypeLines.Items.Clear()
1412         Me.DudNetC1TypeLines.Items.Add(MyClassCode.GetStr(WhichIsTheLang, 182)) '
dushed

```

```

1413     Me.DudNetC1TypeLines.Items.Add(MyClassCode.GetStr(WhichIsTheLang, 183)) '
solid
1414     Me.DudNetC1TypeLines.SelectedIndex = 0
1415     Me.DudNetC2TypeLines.Items.Clear()
1416     Me.DudNetC2TypeLines.Items.Add(MyClassCode.GetStr(WhichIsTheLang, 182))
1417     Me.DudNetC2TypeLines.Items.Add(MyClassCode.GetStr(WhichIsTheLang, 183))
1418     Me.DudNetC2TypeLines.SelectedIndex = 0
1419     Me.DUDNetC1VerticalLine.Items.Clear()
1420     Me.DUDNetC1VerticalLine.Items.Add(MyClassCode.GetStr(WhichIsTheLang, 270))
1421     Me.DUDNetC1VerticalLine.Items.Add(MyClassCode.GetStr(WhichIsTheLang, 176))
1422     Me.DUDNetC1VerticalLine.SelectedIndex = 0
1423     Me.DUDNetC2VerticalLine.Items.Clear()
1424     Me.DUDNetC2VerticalLine.Items.Add(MyClassCode.GetStr(WhichIsTheLang, 270))
1425     Me.DUDNetC2VerticalLine.Items.Add(MyClassCode.GetStr(WhichIsTheLang, 176))
1426     Me.DUDNetC2VerticalLine.SelectedIndex = 0
1427     Me.NUDNetC1TimeVertInterval.Minimum = 20
1428     Me.NUDNetC1TimeVertInterval.Maximum = 100000
1429     Me.NUDNetC2TimeVertInterval.Minimum = 20
1430     Me.NUDNetC2TimeVertInterval.Maximum = 100000
1431     Me.CmbBasicGraphisRate.Items(0) = MyClassCode.GetStr(WhichIsTheLang, 302)
1432
1433
1434 End Sub
1435
1436 Private Sub FillComboLangs(ByVal WhichLanguage As Integer)
1437     CmbLangs.Items.Clear()
1438     CmbLangs.Items.Add(MyClassCode.GetStr(WhichIsTheLang, 15))
1439     CmbLangs.Items.Add(MyClassCode.GetStr(WhichIsTheLang, 16))
1440     CmbLangs.Items.Add(MyClassCode.GetStr(WhichIsTheLang, 259))
1441     CmbLangs.Items.Add(MyClassCode.GetStr(WhichIsTheLang, 260))
1442     CmbLangs.Items.Add(MyClassCode.GetStr(WhichIsTheLang, 261))
1443
1444     CmbLangs.SelectionStart = 1
1445 End Sub
1446
1447 Public Sub WhichEnabled(ByVal Section As Integer)
1448     ' 0 -> All sections enabled
1449     ' 1 -> TPLanguage Enabled
1450     ' 2 -> TPSerialCommunication Enabled
1451     ' 3 -> TPAdraneia.Text
1452     ' 4 -> TPThrust
1453     ' 5 -> TPGraphs
1454     SetAllSectionsFalse(True)
1455
1456     Select Case Section
1457     Case 0
1458         SetAllSectionsFalse(False)
1459     Case 1
1460         TPLanguage.Enabled = True
1461         TPLanguage.Refresh()
1462     Case 2
1463         TPSerialCommunication.Enabled = True
1464     Case 3
1465         TPThrust.Enabled = True
1466     Case 4
1467         TPGraphs.Enabled = True
1468     End Select
1469     Me.TCMain.SelectTab(IIf(Section = 0, 0, Section - 1))
1470     TCMain.Refresh()
1471 End Sub
1472
1473 Private Sub SetAllSectionsFalse(ByVal b1 As Boolean)
1474     TPLanguage.Enabled = Not b1
1475     TPSerialCommunication.Enabled = Not b1
1476     TPThrust.Enabled = Not b1
1477     TPGraphs.Enabled = Not b1
1478     TCMain.Refresh()
1479 End Sub
1480
1481 Public Sub AdjustValuesAccordingToConfFile(ByVal MyFileName As String)
1482     Dim line, MyParam, MyValue As String, MyPosition As Integer
1483
1484     If MyClassCode.IsValidFile(IIf(MyFileName = "", "C:\windows\MyConfs.ini",
MyFileName)) Then
1485         Dim CheckFile As New FileInfo(IIf(MyFileName = "", "C:\windows\MyConfs.ini",

```

```

", MyFileName))
1486
1487     Try
1488         If (Not CheckFile.Exists) Then
1489             CreateConfsFile(MyFileName)
1490         End If
1491         Dim If_not_Confile_selected As String = IIf(MyFileName = "", "C:\
windows\MyConfs.ini", MyFileName)
1492         Dim sr As StreamReader = New StreamReader(If_not_Confile_selected)
1493         line = sr.ReadLine()
1494         Do While Not (line Is Nothing)
1495             MyPosition = line.IndexOf("=")
1496             If MyPosition <> -1 Then
1497                 MyParam = Trim(line.Substring(vbTab.Length, MyPosition - 1))
1498                 MyValue = Trim(line.Substring(MyPosition + 1, Trim(line).
Length - MyPosition - 1))
1499                 AssignConfValuesToObjects(MyParam, MyValue)
1500             End If
1501             line = sr.ReadLine()
1502         Loop
1503         sr.Close()
1504     Catch E As Exception
1505         MsgBox(MyClassCode.GetStr(WhichIsTheLang, 127), 0)
1506     End Try
1507 End If
1508 End Sub
1509
1510 Public Sub AssignConfValuesToObjects(ByVal Param As String, ByVal Value As String)
1511     Select Case Param
1512         Case "Language"
1513             CmbLangs.SelectedIndex = Val(Value)
1514         Case "ShowLanguageSelection"
1515             ChkShowSelectLang.Checked = Val(Value)
1516         Case "Input File"
1517             TxtSaveRecvData.Text = Value
1518         Case "Output File"
1519             TxtSaveSentData.Text = Value
1520         Case "File To Send"
1521             TxtSendFile.Text = Value
1522         Case "COM Port"
1523             CmbComPort.SelectedIndex = Val(Value)
1524         Case "Baud Rate"
1525             CmbBaudRate.SelectedIndex = Val(Value)
1526         Case "Parity"
1527             CmbParity.SelectedIndex = Val(Value)
1528         Case "Data Bits"
1529             CmbDataBits.SelectedIndex = Val(Value)
1530         Case "Stop Bits"
1531             CmbStopBits.SelectedIndex = Val(Value)
1532         Case "Incoming File"
1533             TxtThrustIncoming.Text = Value
1534         Case "Outgoing File"
1535             TxtThrustOutgoing.Text = Value
1536         Case "NetGraphs"
1537             RBNetGraph.Checked = IIf(Value = 0, True, False)
1538         Case "ExcelGraphs"
1539             RBExcelGraph.Checked = IIf(Value = 0, True, False)
1540         Case "NetC1NumOfPoints"
1541             NUDNetC1NumOfPoints.Value = Val(Value)
1542         Case "NetC1HorLines"
1543             NUDNetC1HorLines.Value = Val(Value)
1544         Case "NetC1HorlDiv"
1545             NUDNetC1HorlDiv.Value = Val(Value)
1546         Case "NetC1VerticalLine"
1547             DUDNetC1VerticalLine.SelectedIndex = Val(Value)
1548         Case "NetC1TimeVertInterval"
1549             NUDNetC1TimeVertInterval.Value = Val(Value)
1550         Case "NUDNetC1ConstVerLines"
1551             NUDNetC1ConstVerLines.Value = Val(Value)
1552         Case "NetC1TypeLines"
1553             DudNetC1TypeLines.SelectedIndex = Val(Value)
1554         Case "NetC1Rpm"
1555             ChkNetC1Rpm.Checked = IIf(Value = 0, True, False)
1556         Case "NetC1SetPointRPM"
1557             ChkNetC1SetPointRPM.Checked = IIf(Value = 0, True, False)

```

```

1558     Case "NetC1PWMLoad"
1559         ChkNetC1PWMLoad.Checked = IIf(Value = 0, True, False)
1560     Case "NetC1SetPointPwm"
1561         ChkNetC1SetPointPwm.Checked = IIf(Value = 0, True, False)
1562     Case "NetC2NumOfPoints"
1563         NUD1NetC2NumOfPoints.Value = Val(Value)
1564     Case "NetC2HorLines"
1565         NUDNetC2HorLines.Value = Val(Value)
1566     Case "NetC2Hor1Div"
1567         NUD1NetC2Hor1Div.Value = Val(Value)
1568     Case "NetC2VerticalLine"
1569         DUDNetC2VerticalLine.SelectedIndex = Val(Value)
1570     Case "NetC2TimeVertInterval"
1571         NUDNetC2TimeVertInterval.Value = Val(Value)
1572     Case "NUDNetC2ConstVerLines"
1573         NUDNetC2ConstVerLines.Value = Val(Value)
1574     Case "NetC2TypeLines"
1575         DudNetC2TypeLines.SelectedIndex = Val(Value)
1576     Case "NetC2PWMLoad"
1577         ChkNetC2PWMLoad.Checked = IIf(Value = 0, True, False)
1578     Case "NetC2LoadCell"
1579         ChkNetC2LoadCell.Checked = IIf(Value = 0, True, False)
1580     Case "NetC2Voltage"
1581         ChkNetC2Voltage.Checked = IIf(Value = 0, True, False)
1582     Case "NetC2Ampere"
1583         ChkNetC2Ampere.Checked = IIf(Value = 0, True, False)
1584     Case "ExcelC1SelectLine"
1585         DUBExcelC1SelectLine.SelectedIndex = Val(Value)
1586     Case "ExcelC1ChartPoints"
1587         NUDExcelC1ChartPoints.Value = Val(Value)
1588     Case "ExcelC2SelectLine"
1589         DUBExcelC2SelectLine.SelectedIndex = Val(Value)
1590     Case "ExcelC2ChartPoints"
1591         NUDExcelC2ChartPoints.Value = Val(Value)
1592     Case "BasicGraphisRate"
1593         CmbBasicGraphisRate.SelectedIndex = Val(Value)
1594     Case "GraphicsRate"
1595         CmbGraphicsRate.SelectedIndex = Val(Value)
1596     End Select
1597 End Sub
1598
1599 Public Sub CreateConfsFile(ByVal MyFileName As String)
1600
1601     If MyFileName <> "" Then
1602         Dim CheckFile As New FileInfo(MyFileName)
1603
1604         Try
1605             If (Not CheckFile.Exists) Then
1606                 ' Create an instance of StreamReader to write to a file.
1607                 Dim sw As StreamWriter = New StreamWriter(MyFileName)
1608
1609                 sw.WriteLine("[Language Section]")
1610                 sw.WriteLine(vbTab & "Language = 1") ' English
1611                 sw.WriteLine(vbTab & "ShowLanguageSelection = 0") ' Do Show
1612
1613                 Language Selection
1614                 sw.WriteLine("[RS232 Communication Section]")
1615                 sw.WriteLine(vbTab & "Input File = c:\windows\Input.txt")
1616                 sw.WriteLine(vbTab & "Output File = c:\windows\Output.txt")
1617                 sw.WriteLine(vbTab & "File To Send = c:\windows\SendFile.txt")
1618                 sw.WriteLine(vbTab & "COM Port = 0") ' First Available COM Port
1619                 sw.WriteLine(vbTab & "Baud Rate = 5") '9600
1620                 sw.WriteLine(vbTab & "Parity = 2") ' None
1621                 sw.WriteLine(vbTab & "Data Bits = 3") ' 1
1622                 sw.WriteLine(vbTab & "Stop Bits = 0") ' 0
1623                 sw.WriteLine("[Adraneaia Section]")
1624                 sw.WriteLine(vbTab & "Adraneaia File = c:\windows\Adraneaia.txt")
1625                 sw.WriteLine("[Thrust Section]")
1626                 sw.WriteLine(vbTab & "Incoming File = c:\windows\Thrust_Incoming.
1627                 txt")
1628                 sw.WriteLine(vbTab & "Outgoing File = c:\windows\Thrust_Outgoing.
1629                 txt")
1630
1631                 sw.WriteLine("[Graphs Section]")
1632                 sw.WriteLine(vbTab & "NetGraphs = 0")
1633                 sw.WriteLine(vbTab & "ExcelGraphs = 1")
1634                 sw.WriteLine(vbTab & "NetC1NumOfPoints = 100")

```

```

1630         sw.WriteLine(vbTab & "NetC1HorLines = 0")
1631         sw.WriteLine(vbTab & "NetC1HorlDiv = 0")
1632         sw.WriteLine(vbTab & "NetC1VerticalLine = 0")
1633         sw.WriteLine(vbTab & "NetC1TimeVertInterval = 20")
1634         sw.WriteLine(vbTab & "NUDNetC1ConstVerLines = 0")
1635         sw.WriteLine(vbTab & "NetC1TypeLines = 1")
1636         sw.WriteLine(vbTab & "NetC1Rpm = 0")
1637         sw.WriteLine(vbTab & "NetC1SetPointRPM = 0")
1638         sw.WriteLine(vbTab & "NetC1PWMLoad = 0")
1639         sw.WriteLine(vbTab & "NetC1SetPointPwm = 0")
1640         sw.WriteLine(vbTab & "NetC2NumOfPoints = 100")
1641         sw.WriteLine(vbTab & "NetC2HorLines = 0")
1642         sw.WriteLine(vbTab & "NetC2HorlDiv = 0")
1643         sw.WriteLine(vbTab & "NetC2VerticalLine = 0")
1644         sw.WriteLine(vbTab & "NetC2TimeVertInterval = 20")
1645         sw.WriteLine(vbTab & "NUDNetC2ConstVerLines = 0")
1646         sw.WriteLine(vbTab & "NetC2TypeLines = 0")
1647         sw.WriteLine(vbTab & "NetC2PWMLoad = 0")
1648         sw.WriteLine(vbTab & "NetC2Voltage = 0")
1649         sw.WriteLine(vbTab & "NetC2LoadCell = 0")
1650         sw.WriteLine(vbTab & "NetC2Ampere = 0")
1651         sw.WriteLine(vbTab & "ExcelC1SelectLine = 0")
1652         sw.WriteLine(vbTab & "ExcelC1ChartPoints = 10")
1653         sw.WriteLine(vbTab & "ExcelC2SelectLine = 1")
1654         sw.WriteLine(vbTab & "ExcelC2ChartPoints = 10")
1655         sw.WriteLine(vbTab & "BasicGraphisRate = 2")
1656         sw.WriteLine(vbTab & "GraphicsRate = 5")
1657
1658         sw.Close()
1659     End If
1660     'Create required files, if they exist delete them first
1661     MyClassCode.CreateAndDeleteFileIfExists("c:\windows\Output.txt")
1662     MyClassCode.CreateAndDeleteFileIfExists("c:\windows\Input.txt")
1663     MyClassCode.CreateAndDeleteFileIfExists("c:\windows\SendFile.txt")
1664     MyClassCode.CreateAndDeleteFileIfExists("c:\windows\Adraneia.txt")
1665     MyClassCode.CreateAndDeleteFileIfExists("c:\windows\Thrust_Incoming.
1666     txt")
1667     MyClassCode.CreateAndDeleteFileIfExists("c:\windows\Thrust_Outgoing.
1668     txt")
1669     Catch E As Exception
1670         MsgBox(MyClassCode.GetStr(WhichIsTheLang, 127), 0, MyClassCode.GetStr
1671         (WhichIsTheLang, 109))
1672     End Try
1673 End If
1674 End Sub
1675
1676 Private Sub BtBack_Click(ByVal sender As System.Object, ByVal e As System.
1677     EventArgs) Handles BtBack.Click
1678     Me.Hide()
1679 End Sub
1680
1681 Private Sub BtSave_Click(ByVal sender As System.Object, ByVal e As System.
1682     EventArgs) Handles BtSave.Click
1683     SaveChanges(TxtConfFile.Text)
1684     If SavedChanges Then
1685         Occured = False
1686         Me.Hide()
1687     End If
1688 End Sub
1689
1690 Private Sub SaveChanges(ByVal MyFileName As String)
1691     Dim TestString As String = "C:\windows\MyConfs.ini"
1692
1693     If TxtConfFile.Text.ToLower = TestString.ToLower Then
1694         MsgBox(MyClassCode.GetStr(WhichIsTheLang, 312), MsgBoxStyle.Information,
1695         MyClassCode.GetStr(WhichIsTheLang, 109))
1696     Else
1697         Try
1698             SavedChanges = True
1699             Dim CheckFile As New FileInfo(MyFileName)
1700
1701             If CheckFile.Exists Then
1702                 CheckFile.Delete()
1703                 Dim fs As FileStream = CheckFile.Create()
1704                 fs.Close()

```

```

1699         End If
1700
1701         'Can't leave files undeclared
1702         If CheckIfFileExists(TxtConfFile, 128) And CheckIfFileExists
1703             (TxtSaveSentData, 129) And _
1704             CheckIfFileExists(TxtSaveRecvData, 130) And CheckIfFileExists
1705             (TxtSendFile, 131) And _
1706             CheckIfFileExists(TxtThrustOutgoing, 197) And CheckIfFileExists
1707             (TxtThrustIncoming, 165) _
1708             Then
1709             SavedChanges = True
1710         Else
1711             SavedChanges = False
1712         End If
1713
1714         Dim ThrustCheckFile As New FileInfo(Me.TxtThrustOutgoing.Text)
1715
1716         If ThrustCheckFile.Exists Then
1717             ThrustCheckFile.Delete()
1718             Dim fsThrust As FileStream = ThrustCheckFile.Create()
1719             fsThrust.Close()
1720         End If
1721
1722         If (SavedChanges = True) And (Me.TxtThrustIncoming.Text = Me.
1723             TxtSendFile.Text) _
1724             Or (Me.TxtThrustIncoming.Text = Me.TxtSaveSentData.Text) Or (Me.
1725             TxtThrustIncoming.Text = Me.TxtSaveRecvData.Text) _
1726             Or (Me.TxtSendFile.Text = Me.TxtSaveSentData.Text) Or (Me.
1727             TxtSaveRecvData.Text = Me.TxtSendFile.Text) _
1728             Or (Me.TxtSaveSentData.Text = Me.TxtSaveRecvData.Text) _
1729             Or (Me.TxtThrustOutgoing.Text = Me.TxtSaveRecvData.Text) Or (Me.
1730             TxtThrustOutgoing.Text = Me.TxtSaveSentData.Text) _
1731             Or (Me.TxtThrustOutgoing.Text = Me.TxtThrustIncoming.Text) _
1732             Or (Me.TxtThrustOutgoing.Text = Me.TxtSendFile.Text) Then
1733             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 133), 0, MyClassCode.
1734             GetStr(WhichIsTheLang, 109))
1735             SavedChanges = False
1736             Exit Sub
1737         End If
1738
1739         If (Me.TxtSendFile.Text = Me.TxtConfFile.Text) Or (Me.TxtSaveRecvData.
1740             Text = Me.TxtConfFile.Text) Or _
1741             (Me.TxtSaveSentData.Text = Me.TxtConfFile.Text) Or _
1742             (Me.TxtThrustIncoming.Text = Me.TxtConfFile.Text) Or (Me.
1743             TxtThrustOutgoing.Text = Me.TxtConfFile.Text) Then
1744             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 134), 0, MyClassCode.
1745             GetStr(WhichIsTheLang, 109))
1746             SavedChanges = False
1747             Exit Sub
1748         End If
1749
1750         If SavedChanges = True Then
1751             Dim sw As StreamWriter
1752             sw = CheckFile.AppendText()
1753
1754             sw.WriteLine("[Language Section]")
1755             sw.WriteLine(vbTab & "Language = " & Str(Me.CmbLangs.
1756             SelectedIndex))
1757             sw.WriteLine(vbTab & "ShowLanguageSelection = " & IIf
1758             (ChkShowSelectLang.Checked, "1", "0"))
1759             sw.WriteLine("[RS232 Communication Section]")
1760             sw.WriteLine(vbTab & "Input File = " & Me.TxtSaveRecvData.Text)
1761             sw.WriteLine(vbTab & "Output File = " & Me.TxtSaveSentData.Text)
1762             sw.WriteLine(vbTab & "File To Send = " & Me.TxtSendFile.Text)
1763             sw.WriteLine(vbTab & "COM Port = " & Str(Me.CmbComPort.
1764             SelectedIndex))
1765             sw.WriteLine(vbTab & "Baud Rate = " & Str(Me.CmbBaudRate.
1766             SelectedIndex))
1767             sw.WriteLine(vbTab & "Parity = " & Str(Me.CmbParity.
1768             SelectedIndex))
1769             sw.WriteLine(vbTab & "Data Bits = " & Str(Me.CmbDataBits.
1770             SelectedIndex))
1771             sw.WriteLine(vbTab & "Stop Bits = " & Str(Me.CmbStopBits.
1772             SelectedIndex))
1773             sw.WriteLine("[Thrust Section]")

```

```

1756         sw.WriteLine(vbTab & "Incoming File = " & Me.TxtThrustIncoming.
Text)
1757         sw.WriteLine(vbTab & "Outgoing File = " & Me.TxtThrustOutgoing.
Text)
1758         sw.WriteLine("[Graphs Section]")
1759         sw.WriteLine(vbTab & "NetGraphs = " & IIf(RBNetGraph.Checked, "0",
"1"))
1760         sw.WriteLine(vbTab & "ExcelGraphs = " & IIf(RBExcelGraph.Checked,
"0", "1"))
1761         sw.WriteLine(vbTab & "NetC1NumOfPoints = " & Str
(NUDNetC1NumOfPoints.Value))
1762         sw.WriteLine(vbTab & "NetC1HorLines = " & Str(NUDNetC1HorLines.
Value))
1763         sw.WriteLine(vbTab & "NetC1HorlDiv = " & Str(NUDNetC1HorlDiv.
Value))
1764         sw.WriteLine(vbTab & "NetC1VerticalLine = " & Str
(DUDNetC1VerticalLine.SelectedIndex))
1765         sw.WriteLine(vbTab & "NetC1TimeVertInterval = " & Str
(NUDNetC1TimeVertInterval.Value))
1766         sw.WriteLine(vbTab & "NUDNetC1ConstVerLines = " & Str
(NUDNetC1ConstVerLines.Value))
1767         sw.WriteLine(vbTab & "NetC1TypeLines = " & Str(DudNetC1TypeLines.
SelectedIndex))
1768         sw.WriteLine(vbTab & "NetC1Rpm = " & IIf(ChkNetC1Rpm.Checked, "0",
"1"))
1769         sw.WriteLine(vbTab & "NetC1SetPointRPM = " & IIf
(ChkNetC1SetPointRPM.Checked, "0", "1"))
1770         sw.WriteLine(vbTab & "NetC1PWMLoad = " & IIf(ChkNetC1PWMLoad.
Checked, "0", "1"))
1771         sw.WriteLine(vbTab & "NetC1SetPointPwm = " & IIf
(ChkNetC1SetPointPwm.Checked, "0", "1"))
1772         sw.WriteLine(vbTab & "NetC2NumOfPoints = " & Str
(NUD1NetC2NumOfPoints.Value))
1773         sw.WriteLine(vbTab & "NetC2HorLines = " & Str(NUDNetC2HorLines.
Value))
1774         sw.WriteLine(vbTab & "NetC2HorlDiv = " & Str(NUD1NetC2HorlDiv.
Value))
1775         sw.WriteLine(vbTab & "NetC2VerticalLine = " & Str
(DUDNetC2VerticalLine.SelectedIndex))
1776         sw.WriteLine(vbTab & "NetC2TimeVertInterval = " & Str
(NUDNetC2TimeVertInterval.Value))
1777         sw.WriteLine(vbTab & "NUDNetC2ConstVerLines = " & Str
(NUDNetC2ConstVerLines.Value))
1778         sw.WriteLine(vbTab & "NetC2TypeLines = " & Str(DudNetC2TypeLines.
SelectedIndex))
1779         sw.WriteLine(vbTab & "NetC2PWMLoad = " & IIf(ChkNetC2PWMLoad.
Checked, "0", "1"))
1780         sw.WriteLine(vbTab & "NetC2Voltage = " & IIf(ChkNetC2Voltage.
Checked, "0", "1"))
1781         sw.WriteLine(vbTab & "NetC2LoadCell = " & IIf(ChkNetC2LoadCell.
Checked, "0", "1"))
1782         sw.WriteLine(vbTab & "NetC2Ampere = " & IIf(ChkNetC2Ampere.Checked
, "0", "1"))
1783         sw.WriteLine(vbTab & "ExcelC1SelectLine = " & Str
(DUBExcelC1SelectLine.SelectedIndex))
1784         sw.WriteLine(vbTab & "ExcelC1ChartPoints = " & Str
(NUDExcelC1ChartPoints.Value))
1785         sw.WriteLine(vbTab & "ExcelC2SelectLine = " & Str
(DUBExcelC2SelectLine.SelectedIndex))
1786         sw.WriteLine(vbTab & "ExcelC2ChartPoints = " & Str
(NUDExcelC2ChartPoints.Value))
1787         sw.WriteLine(vbTab & "BasicGraphisRate = " & Str
(CmbBasicGraphisRate.SelectedIndex))
1788         sw.WriteLine(vbTab & "GraphicsRate = " & Str(CmbGraphicsRate.
SelectedIndex))
1789
1790         sw.Close()
1791
1792         Dim MyFileName1, line As String
1793         MyFileName1 = "c:\windows\MyConfFiles.ini"
1794         Dim CheckFile1 As New FileInfo(MyFileName1)
1795         Dim EntryExists As Boolean
1796
1797         EntryExists = False
1798         If Not CheckFile1.Exists Then

```

```

1799         Dim sw1 As StreamWriter = New StreamWriter(MyFileName1)
1800         sw1.WriteLine(MyFileName)
1801         sw1.Close()
1802     End If
1803
1804     Dim srl As StreamReader = New StreamReader(MyFileName1)
1805     Do
1806         line = srl.ReadLine()
1807         If line <> Nothing Then
1808             If line.ToLower() = Me.TxtConfFile.Text.ToLower() Then
1809                 EntryExists = True
1810             End If
1811         End If
1812     Loop Until line Is Nothing
1813     srl.Close()
1814
1815     If Not EntryExists Then
1816         Dim sw1 As StreamWriter
1817         sw1 = CheckFile1.AppendText()
1818         sw1.WriteLine("")
1819         sw1.WriteLine(Me.TxtConfFile.Text.ToLower())
1820         sw1.Close()
1821     End If
1822     DefaultConfFile = Me.TxtConfFile.Text
1823
1824     Dim SwThrust As StreamWriter = New StreamWriter(Me.
1825     TxtThrustOutgoing.Text)
1826     'Write Header for Thrust Outgoing Data File
1827     MyClassCode.SetPointFileHeader(SwThrust, True, "0", "")
1828     SwThrust.Close()
1829     End If
1830     Catch E As Exception
1831     ' Let the user know what went wrong.
1832     MsgBox(MyClassCode.GetStr(WhichIsTheLang, 135), 0, MyClassCode.GetStr
1833     (WhichIsTheLang, 109))
1834     End Try
1835 End Sub
1836
1837 Public Function CheckIfFileExists(ByVal MyObject As TextBox, ByVal Message As
1838 Integer) As Boolean
1839     If Not MyClassCode.IsValidFile(MyObject.Text) Then
1840         MsgBox(MyClassCode.GetStr(WhichIsTheLang, Message), 0, MyClassCode.GetStr
1841         (WhichIsTheLang, 109))
1842         Return False
1843     Else
1844         Return True
1845     End If
1846 End Function
1847 Public Sub AppropriatelyFillComCombo()
1848     Dim i As Integer
1849     Me.CmbComPort.Items.Clear()
1850     For i = 0 To 9
1851         If mComs(i) Then
1852             Me.CmbComPort.Items.Add("COM" & CStr(i + 1))
1853         End If
1854     Next
1855 End Sub
1856
1857 Private Sub BtSaveSentData_Click(ByVal sender As System.Object, ByVal e As System.
1858 EventArgs) Handles BtSaveSentData.Click
1859     MyClassCode.ToOpenAFileDialog(TxtSaveSentData, WhichIsTheLang,
1860     InitialDirectory)
1861 End Sub
1862
1863 Private Sub BtSaveRecvData_Click(ByVal sender As System.Object, ByVal e As System.
1864 EventArgs) Handles BtSaveRecvData.Click
1865     MyClassCode.ToOpenAFileDialog(TxtSaveRecvData, WhichIsTheLang,
1866     InitialDirectory)
1867 End Sub
1868
1869 Private Sub BtSendFile_Click(ByVal sender As System.Object, ByVal e As System.
1870 EventArgs) Handles BtSendFile.Click
1871     MyClassCode.ToOpenAFileDialog(TxtSendFile, WhichIsTheLang, InitialDirectory)

```



```

1865     End Sub
1866
1867     Private Sub BtThrust_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles BtThrustIncoming.Click
1868         MyClassCode.ToOpenAFileDialog(TxtThrustIncoming, WhichIsTheLang,
InitialDirectory)
1869     End Sub
1870
1871     Private Sub BtThrustOutgoing_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtThrustOutgoing.Click
1872         MyClassCode.ToOpenAFileDialog(TxtThrustOutgoing, WhichIsTheLang,
InitialDirectory)
1873     End Sub
1874
1875     Private Sub BtConfFile_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles BtConfFile.Click
1876         MyClassCode.ToOpenAFileDialog(TxtConfFile, WhichIsTheLang, InitialDirectory)
1877
1878         If TxtConfFile.Text <> "" And Not MyClassCode.IsValidFile(TxtConfFile.Text)
Then
1879             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 137), 0, MyClassCode.GetStr
(WhichIsTheLang, 109))
1880             TxtConfFile.Focus()
1881         Else
1882             AdjustValuesAccordingToConfFile(TxtConfFile.Text)
1883         End If
1884     End Sub
1885
1886     Private Sub TxtConfFile_Validating(ByVal sender As Object, ByVal e As System.
ComponentModel.CancelEventArgs) Handles TxtConfFile.Validating
1887         If TxtConfFile.Text <> "" And Not MyClassCode.IsValidFile(TxtConfFile.Text)
Then
1888             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 137), 0, MyClassCode.GetStr
(WhichIsTheLang, 109))
1889             TxtConfFile.Focus()
1890         Else
1891             AdjustValuesAccordingToConfFile(TxtConfFile.Text)
1892         End If
1893     End Sub
1894
1895
1896
1897     Private Sub DUDNetC1VerticalLine_SelectedItemChanged(ByVal sender As System.Object
, ByVal e As System.EventArgs) Handles DUDNetC1VerticalLine.SelectedItemChanged
1898         Select Case DUDNetC1VerticalLine.SelectedIndex
1899             Case 0
1900                 Me.NUDNetC1ConstVerLines.Visible = False
1901                 Me.NUDNetC1TimeVertInterval.Visible = True
1902             Case 1
1903                 Me.NUDNetC1ConstVerLines.Visible = True
1904                 Me.NUDNetC1TimeVertInterval.Visible = False
1905         End Select
1906
1907     End Sub
1908
1909     Private Sub DUDNetC2VerticalLine_SelectedItemChanged(ByVal sender As System.Object
, ByVal e As System.EventArgs) Handles DUDNetC2VerticalLine.SelectedItemChanged
1910         Select Case DUDNetC2VerticalLine.SelectedIndex
1911             Case 0
1912                 Me.NUDNetC2ConstVerLines.Visible = False
1913                 Me.NUDNetC2TimeVertInterval.Visible = True
1914             Case 1
1915                 Me.NUDNetC2ConstVerLines.Visible = True
1916                 Me.NUDNetC2TimeVertInterval.Visible = False
1917         End Select
1918
1919     End Sub
1920 End Class

```

```

1 Imports System.Text
2 Imports System.IO
3
4 Public Class SelectConfFiles
5     Inherits System.Windows.Forms.Form
6
7     Private MyClassCode As New MyPtyxClass
8     Public NodeLabel, PrevSelectedConfFile As String
9     Private WhichLanguagePrivate As String
10 #Region " Windows Form Designer generated code "
11
12     Public Sub New()
13         MyBase.New()
14
15         'This call is required by the Windows Form Designer.
16         InitializeComponent()
17
18         'Add any initialization after the InitializeComponent() call
19
20     End Sub
21
22     'Form overrides dispose to clean up the component list.
23     Protected Overrides Sub Dispose(ByVal disposing As Boolean)
24         If disposing Then
25             If Not (components Is Nothing) Then
26                 components.Dispose()
27             End If
28         End If
29         MyBase.Dispose(disposing)
30     End Sub
31
32     'Required by the Windows Form Designer
33     Private components As System.ComponentModel.IContainer
34
35     'NOTE: The following procedure is required by the Windows Form Designer
36     'It can be modified using the Windows Form Designer.
37     'Do not modify it using the code editor.
38     Friend WithEvents TrVConfigFiles As System.Windows.Forms.TreeView
39     Friend WithEvents BtOk As System.Windows.Forms.Button
40     Friend WithEvents BtCancel As System.Windows.Forms.Button
41     <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
42         Dim resources As System.ComponentModel.ComponentResourceManager = New System.
43         ComponentModel.ComponentResourceManager(GetType(SelectConfFiles))
44         Me.TrVConfigFiles = New System.Windows.Forms.TreeView
45         Me.BtOk = New System.Windows.Forms.Button
46         Me.BtCancel = New System.Windows.Forms.Button
47         Me.SuspendLayout()
48         'TrVConfigFiles
49         '
50         Me.TrVConfigFiles.Location = New System.Drawing.Point(8, 0)
51         Me.TrVConfigFiles.Name = "TrVConfigFiles"
52         Me.TrVConfigFiles.Size = New System.Drawing.Size(272, 272)
53         Me.TrVConfigFiles.TabIndex = 0
54         '
55         'BtOk
56         '
57         Me.BtOk.Location = New System.Drawing.Point(24, 280)
58         Me.BtOk.Name = "BtOk"
59         Me.BtOk.Size = New System.Drawing.Size(96, 24)
60         Me.BtOk.TabIndex = 1
61         Me.BtOk.Text = "Επιλογή"
62         '
63         'BtCancel
64         '
65         Me.BtCancel.DialogResult = System.Windows.Forms.DialogResult.Cancel
66         Me.BtCancel.Location = New System.Drawing.Point(152, 280)
67         Me.BtCancel.Name = "BtCancel"
68         Me.BtCancel.Size = New System.Drawing.Size(88, 24)
69         Me.BtCancel.TabIndex = 2
70         Me.BtCancel.Text = "Ακύρωση"
71         '
72         'SelectConfFiles
73         '
74         Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)

```

```

75     Me.CancelButton = Me.BtCancel
76     Me.ClientSize = New System.Drawing.Size(292, 309)
77     Me.Controls.Add(Me.BtCancel)
78     Me.Controls.Add(Me.BtOk)
79     Me.Controls.Add(Me.TrVConfigFiles)
80     Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle
81     Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)
82     Me.MaximizeBox = False
83     Me.MinimizeBox = False
84     Me.Name = "SelectConfFiles"
85     Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent
86     Me.Text = "SelectConfFiles"
87     Me.ResumeLayout(False)
88
89     End Sub
90
91 #End Region
92
93     Public Sub CreateTree()
94         Try
95             Dim MyFileName, line As String
96
97             MyFileName = "c:\windows\MyConfFiles.ini"
98             Dim CheckFile As New FileInfo(MyFileName)
99
100            If Not CheckFile.Exists Then
101                Dim sw As StreamWriter = New StreamWriter(MyFileName)
102                sw.WriteLine("c:\windows\MyConfs.ini")
103                If Trim(PrevSelectedConfFile) <> "c:\windows\MyConfs.ini" Then
104                    sw.WriteLine(PrevSelectedConfFile)
105                End If
106                sw.Close()
107
108            End If
109
110            Dim node As TreeNode
111            node = TrVConfigFiles.Nodes.Add(MyClassCode.GetStr(WhichLanguagePrivate,
166))
112
113            Dim sr As StreamReader = New StreamReader(MyFileName)
114            line = sr.ReadLine()
115            Do While Not line Is Nothing
116                If MyClassCode.IsValidFile(line) Then
117                    If PrevSelectedConfFile = line Then
118                        node.Nodes.Add("wraia", line)
119                    Else
120                        node.Nodes.Add("asxeto", line)
121                    End If
122                End If
123                line = sr.ReadLine()
124            Loop
125            TrVConfigFiles.ExpandAll()
126            If PrevSelectedConfFile <> "" Then
127                Dim temp(1) As System.Windows.Forms.TreeNode
128                temp = node.Nodes.Find("wraia", True)
129                If temp.Length > 0 Then
130                    TrVConfigFiles.SelectedNode = temp(0)
131                End If
132            End If
133        Catch
134            Stop
135        End Try
136    End Sub
137
138    Private Sub TrVConfigFiles_AfterSelect(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.TreeViewEventArgs) Handles TrVConfigFiles.AfterSelect
139        NodeLabel = TrVConfigFiles.SelectedNode.Text
140    End Sub
141
142    Public Sub WhichLang(ByVal WhichLanguage As Integer)
143        WhichLanguagePrivate = WhichLanguage
144        CreateTree()
145        Me.Text = MyClassCode.GetStr(WhichLanguagePrivate, 24)
146        Me.BtOk.Text = MyClassCode.GetStr(WhichLanguagePrivate, 167)
147        Me.BtCancel.Text = MyClassCode.GetStr(WhichLanguagePrivate, 78)

```

```

148     End Sub
149
150     Private Sub BtOk_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) ✓
Handles BtOk.Click
151         If (NodeLabel = MyClassCode.GetStr(WhichLanguagePrivate, 166)) Then
152             MsgBox(MyClassCode.GetStr(WhichLanguagePrivate, 166), 0, MyClassCode.GetStr ✓
(WhichLanguagePrivate, 109))
153         Else
154             Me.Close()
155         End If
156     End Sub
157
158     Private Sub BtCancel_Click(ByVal sender As System.Object, ByVal e As System. ✓
EventArgs) Handles BtCancel.Click
159         NodeLabel = ""
160         Me.Close()
161     End Sub
162
163     Private Sub SelectConfFiles_Closing(ByVal sender As Object, ByVal e As System. ✓
ComponentModel.CancelEventArgs) Handles MyBase.Closing
164         If (NodeLabel = MyClassCode.GetStr(WhichLanguagePrivate, 166)) Then
165             NodeLabel = ""
166         End If
167     End Sub
168 End Class
169

```

```

1 Imports System.IO
2 Imports Microsoft.Win32
3 Imports System.Windows.Forms
4
5 Public Class SelectLanguage
6     Inherits System.Windows.Forms.Form
7     Private MyClassCode As New MyPtyxClass
8     Private DoShowMe As Boolean
9     Public WhichIsTheLang As Integer
10    Friend WithEvents LblItalian As System.Windows.Forms.Label
11    Friend WithEvents PbItalian As System.Windows.Forms.PictureBox
12    Friend WithEvents LblSpanish As System.Windows.Forms.Label
13    Friend WithEvents PBSpanish As System.Windows.Forms.PictureBox
14    Friend WithEvents LblFrance As System.Windows.Forms.Label
15    Friend WithEvents PBFrance As System.Windows.Forms.PictureBox
16    Friend WithEvents RBGreek As System.Windows.Forms.RadioButton
17    Friend WithEvents RBEnglish As System.Windows.Forms.RadioButton
18    Friend WithEvents RBItalian As System.Windows.Forms.RadioButton
19    Friend WithEvents RBSpanish As System.Windows.Forms.RadioButton
20    Friend WithEvents RBFrance As System.Windows.Forms.RadioButton
21    Private DefaultConfFile As String
22
23 #Region " Windows Form Designer generated code "
24
25     Public Sub New()
26         MyBase.New()
27         'This call is required by the Windows Form Designer.
28         InitializeComponent()
29         'Add any initialization after the InitializeComponent() call
30     End Sub
31
32     'Form overrides dispose to clean up the component list.
33     Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
34         If disposing Then
35             If Not (components Is Nothing) Then
36                 components.Dispose()
37             End If
38         End If
39         MyBase.Dispose(disposing)
40     End Sub
41
42     'Required by the Windows Form Designer
43     Private components As System.ComponentModel.IContainer
44     'NOTE: The following procedure is required by the Windows Form Designer
45     'It can be modified using the Windows Form Designer.
46     'Do not modify it using the code editor.
47     Friend WithEvents PBGreek As System.Windows.Forms.PictureBox
48     Friend WithEvents PBEnglish As System.Windows.Forms.PictureBox
49     Friend WithEvents LblGreek As System.Windows.Forms.Label
50     Friend WithEvents LblEnglish As System.Windows.Forms.Label
51     Friend WithEvents BtOk As System.Windows.Forms.Button
52     Friend WithEvents BtCancel As System.Windows.Forms.Button
53     <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
54         Dim resources As System.ComponentModel.ComponentResourceManager = New System.
55         ComponentModel.ComponentResourceManager(GetType(SelectLanguage))
56         Me.PBGreek = New System.Windows.Forms.PictureBox
57         Me.PBEnglish = New System.Windows.Forms.PictureBox
58         Me.LblGreek = New System.Windows.Forms.Label
59         Me.LblEnglish = New System.Windows.Forms.Label
60         Me.BtOk = New System.Windows.Forms.Button
61         Me.BtCancel = New System.Windows.Forms.Button
62         Me.LblItalian = New System.Windows.Forms.Label
63         Me.PbItalian = New System.Windows.Forms.PictureBox
64         Me.LblSpanish = New System.Windows.Forms.Label
65         Me.PBSpanish = New System.Windows.Forms.PictureBox
66         Me.LblFrance = New System.Windows.Forms.Label
67         Me.PBFrance = New System.Windows.Forms.PictureBox
68         Me.RBGreek = New System.Windows.Forms.RadioButton
69         Me.RBEnglish = New System.Windows.Forms.RadioButton
70         Me.RBItalian = New System.Windows.Forms.RadioButton
71         Me.RBSpanish = New System.Windows.Forms.RadioButton
72         Me.RBFrance = New System.Windows.Forms.RadioButton
73         CType(Me.PBGreek, System.ComponentModel.ISupportInitialize).BeginInit()
74         CType(Me.PBEnglish, System.ComponentModel.ISupportInitialize).BeginInit()
75         CType(Me.PbItalian, System.ComponentModel.ISupportInitialize).BeginInit()

```

```

75     CType(Me.PBSpanish, System.ComponentModel.ISupportInitialize).BeginInit()
76     CType(Me.PBFrance, System.ComponentModel.ISupportInitialize).BeginInit()
77     Me.SuspendLayout()
78     '
79     'PBGreek
80     '
81     Me.PBGreek.Cursor = System.Windows.Forms.Cursors.Hand
82     Me.PBGreek.Image = CType(resources.GetObject("PBGreek.Image"), System.Drawing.
Image)
83     Me.PBGreek.Location = New System.Drawing.Point(56, 20)
84     Me.PBGreek.Name = "PBGreek"
85     Me.PBGreek.Size = New System.Drawing.Size(32, 24)
86     Me.PBGreek.TabIndex = 0
87     Me.PBGreek.TabStop = False
88     '
89     'PBEnglish
90     '
91     Me.PBEnglish.Cursor = System.Windows.Forms.Cursors.Hand
92     Me.PBEnglish.Image = CType(resources.GetObject("PBEnglish.Image"), System.
Drawing.Image)
93     Me.PBEnglish.Location = New System.Drawing.Point(56, 51)
94     Me.PBEnglish.Name = "PBEnglish"
95     Me.PBEnglish.Size = New System.Drawing.Size(32, 24)
96     Me.PBEnglish.TabIndex = 1
97     Me.PBEnglish.TabStop = False
98     '
99     'LblGreek
100    '
101    Me.LblGreek.Cursor = System.Windows.Forms.Cursors.Hand
102    Me.LblGreek.Location = New System.Drawing.Point(112, 20)
103    Me.LblGreek.Name = "LblGreek"
104    Me.LblGreek.Size = New System.Drawing.Size(88, 24)
105    Me.LblGreek.TabIndex = 2
106    Me.LblGreek.Text = "Ελληνικά"
107    '
108    'LblEnglish
109    '
110    Me.LblEnglish.Cursor = System.Windows.Forms.Cursors.Hand
111    Me.LblEnglish.Location = New System.Drawing.Point(112, 51)
112    Me.LblEnglish.Name = "LblEnglish"
113    Me.LblEnglish.Size = New System.Drawing.Size(88, 24)
114    Me.LblEnglish.TabIndex = 3
115    Me.LblEnglish.Text = "Αγγλικά"
116    '
117    'BtOk
118    '
119    Me.BtOk.Location = New System.Drawing.Point(3, 187)
120    Me.BtOk.Name = "BtOk"
121    Me.BtOk.Size = New System.Drawing.Size(88, 24)
122    Me.BtOk.TabIndex = 4
123    Me.BtOk.Text = "Αποδοχή"
124    '
125    'BtCancel
126    '
127    Me.BtCancel.DialogResult = System.Windows.Forms.DialogResult.Cancel
128    Me.BtCancel.Location = New System.Drawing.Point(115, 187)
129    Me.BtCancel.Name = "BtCancel"
130    Me.BtCancel.Size = New System.Drawing.Size(88, 24)
131    Me.BtCancel.TabIndex = 5
132    Me.BtCancel.Text = "Ακύρωση"
133    '
134    'LblItalian
135    '
136    Me.LblItalian.Cursor = System.Windows.Forms.Cursors.Hand
137    Me.LblItalian.Location = New System.Drawing.Point(112, 84)
138    Me.LblItalian.Name = "LblItalian"
139    Me.LblItalian.Size = New System.Drawing.Size(88, 24)
140    Me.LblItalian.TabIndex = 9
141    Me.LblItalian.Text = "Ιταλικά"
142    '
143    'PbItalian
144    '
145    Me.PbItalian.Cursor = System.Windows.Forms.Cursors.Hand
146    Me.PbItalian.Image = CType(resources.GetObject("PbItalian.Image"), System.
Drawing.Image)

```

```

147     Me.PbItalian.Location = New System.Drawing.Point(56, 84)
148     Me.PbItalian.Name = "PbItalian"
149     Me.PbItalian.Size = New System.Drawing.Size(32, 24)
150     Me.PbItalian.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage
151     Me.PbItalian.TabIndex = 8
152     Me.PbItalian.TabStop = False
153     '
154     'LblSpanish
155     '
156     Me.LblSpanish.Cursor = System.Windows.Forms.Cursors.Hand
157     Me.LblSpanish.Location = New System.Drawing.Point(112, 117)
158     Me.LblSpanish.Name = "LblSpanish"
159     Me.LblSpanish.Size = New System.Drawing.Size(88, 24)
160     Me.LblSpanish.TabIndex = 12
161     Me.LblSpanish.Text = "Ισπανικά"
162     '
163     'PBSpanish
164     '
165     Me.PBSpanish.Cursor = System.Windows.Forms.Cursors.Hand
166     Me.PBSpanish.Image = CType(resources.GetObject("PBSpanish.Image"), System.
Drawing.Image)
167     Me.PBSpanish.Location = New System.Drawing.Point(56, 117)
168     Me.PBSpanish.Name = "PBSpanish"
169     Me.PBSpanish.Size = New System.Drawing.Size(32, 24)
170     Me.PBSpanish.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage
171     Me.PBSpanish.TabIndex = 11
172     Me.PBSpanish.TabStop = False
173     '
174     'LblFrance
175     '
176     Me.LblFrance.Cursor = System.Windows.Forms.Cursors.Hand
177     Me.LblFrance.Location = New System.Drawing.Point(112, 149)
178     Me.LblFrance.Name = "LblFrance"
179     Me.LblFrance.Size = New System.Drawing.Size(88, 24)
180     Me.LblFrance.TabIndex = 15
181     Me.LblFrance.Text = "Γαλλικά"
182     '
183     'PBFrance
184     '
185     Me.PBFrance.Cursor = System.Windows.Forms.Cursors.Hand
186     Me.PBFrance.Image = CType(resources.GetObject("PBFrance.Image"), System.Drawing
.Image)
187     Me.PBFrance.Location = New System.Drawing.Point(56, 149)
188     Me.PBFrance.Name = "PBFrance"
189     Me.PBFrance.Size = New System.Drawing.Size(32, 24)
190     Me.PBFrance.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage
191     Me.PBFrance.TabIndex = 14
192     Me.PBFrance.TabStop = False
193     '
194     'RBGreek
195     '
196     Me.RBGreek.AutoSize = True
197     Me.RBGreek.Cursor = System.Windows.Forms.Cursors.Hand
198     Me.RBGreek.Location = New System.Drawing.Point(29, 22)
199     Me.RBGreek.Name = "RBGreek"
200     Me.RBGreek.Size = New System.Drawing.Size(14, 13)
201     Me.RBGreek.TabIndex = 16
202     Me.RBGreek.TabStop = True
203     Me.RBGreek.UseVisualStyleBackColor = True
204     '
205     'RBEnglish
206     '
207     Me.RBEnglish.AutoSize = True
208     Me.RBEnglish.Cursor = System.Windows.Forms.Cursors.Hand
209     Me.RBEnglish.Location = New System.Drawing.Point(29, 56)
210     Me.RBEnglish.Name = "RBEnglish"
211     Me.RBEnglish.Size = New System.Drawing.Size(14, 13)
212     Me.RBEnglish.TabIndex = 17
213     Me.RBEnglish.TabStop = True
214     Me.RBEnglish.UseVisualStyleBackColor = True
215     '
216     'RBItalian
217     '
218     Me.RBItalian.AutoSize = True
219     Me.RBItalian.Cursor = System.Windows.Forms.Cursors.Hand

```

```

220     Me.RBItalian.Location = New System.Drawing.Point(29, 88)
221     Me.RBItalian.Name = "RBItalian"
222     Me.RBItalian.Size = New System.Drawing.Size(14, 13)
223     Me.RBItalian.TabIndex = 18
224     Me.RBItalian.TabStop = True
225     Me.RBItalian.UseVisualStyleBackColor = True
226     '
227     'RBSpanish
228     '
229     Me.RBSpanish.AutoSize = True
230     Me.RBSpanish.Cursor = System.Windows.Forms.Cursors.Hand
231     Me.RBSpanish.Location = New System.Drawing.Point(29, 121)
232     Me.RBSpanish.Name = "RBSpanish"
233     Me.RBSpanish.Size = New System.Drawing.Size(14, 13)
234     Me.RBSpanish.TabIndex = 19
235     Me.RBSpanish.TabStop = True
236     Me.RBSpanish.UseVisualStyleBackColor = True
237     '
238     'RBFrance
239     '
240     Me.RBFrance.AutoSize = True
241     Me.RBFrance.Cursor = System.Windows.Forms.Cursors.Hand
242     Me.RBFrance.Location = New System.Drawing.Point(29, 153)
243     Me.RBFrance.Name = "RBFrance"
244     Me.RBFrance.Size = New System.Drawing.Size(14, 13)
245     Me.RBFrance.TabIndex = 20
246     Me.RBFrance.TabStop = True
247     Me.RBFrance.UseVisualStyleBackColor = True
248     '
249     'SelectLanguage
250     '
251     Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
252     Me.CancelButton = Me.BtCancel
253     Me.ClientSize = New System.Drawing.Size(204, 215)
254     Me.Controls.Add(Me.RBFrance)
255     Me.Controls.Add(Me.RBSpanish)
256     Me.Controls.Add(Me.RBItalian)
257     Me.Controls.Add(Me.RBEnglish)
258     Me.Controls.Add(Me.RBGreek)
259     Me.Controls.Add(Me.LblFrance)
260     Me.Controls.Add(Me.PBFrance)
261     Me.Controls.Add(Me.LblSpanish)
262     Me.Controls.Add(Me.PBSpanish)
263     Me.Controls.Add(Me.LblItalian)
264     Me.Controls.Add(Me.PbItalian)
265     Me.Controls.Add(Me.BtCancel)
266     Me.Controls.Add(Me.BtOk)
267     Me.Controls.Add(Me.LblEnglish)
268     Me.Controls.Add(Me.LblGreek)
269     Me.Controls.Add(Me.PBEnglish)
270     Me.Controls.Add(Me.PBGreek)
271     Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle
272     Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)
273     Me.MaximizeBox = False
274     Me.MinimizeBox = False
275     Me.Name = "SelectLanguage"
276     Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen
277     Me.Text = "7"
278     CType(Me.PBGreek, System.ComponentModel.ISupportInitialize).EndInit()
279     CType(Me.PBEnglish, System.ComponentModel.ISupportInitialize).EndInit()
280     CType(Me.PbItalian, System.ComponentModel.ISupportInitialize).EndInit()
281     CType(Me.PBSpanish, System.ComponentModel.ISupportInitialize).EndInit()
282     CType(Me.PBFrance, System.ComponentModel.ISupportInitialize).EndInit()
283     Me.ResumeLayout(False)
284     Me.PerformLayout()
285
286     End Sub
287
288 #End Region
289
290     Private Sub SelectLanguage_Load(ByVal sender As Object, ByVal e As System.
EventArgs) Handles Me.Load
291         MyClassCode.KillMeIfMySisterIsRunning()
292         CheckIfThisShouldBeShown()
293         If Not DoShowMe Then

```



```

294         Me.FormBorderStyle = Windows.Forms.FormBorderStyle.None
295         Me.Width = 0
296         Me.Height = 0
297         Me.ShowInTaskbar = False
298         Me.Hide()
299         Me.Refresh()
300     Else
301         Me.FormBorderStyle = Windows.Forms.FormBorderStyle.FixedSingle
302         Me.Width = 210
303         Me.Height = 245
304         Me.Visible = True
305         Me.RBEnglish.Refresh()
306         Me.RBGreek.Refresh()
307         Me.RBItalian.Refresh()
308         Me.RBSpanish.Refresh()
309         Me.RBFrance.Refresh()
310         Me.BtCancel.Refresh()
311         Me.BtOk.Refresh()
312         Me.LblEnglish.Refresh()
313         Me.LblGreek.Refresh()
314         Me.LblItalian.Refresh()
315         Me.LblSpanish.Refresh()
316         Me.LblFrance.Refresh()
317         Me.PBEnglish.Refresh()
318         Me.PBGreek.Refresh()
319         Me.PbItalian.Refresh()
320         Me.PBSpanish.Refresh()
321         Me.PBFrance.Refresh()
322         Me.StartPosition = FormStartPosition.CenterScreen
323         Me.Refresh()
324     End If
325 End Sub
326
327 Private Sub BtOk_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
328 Handles BtOk.Click
329     Dim fMain As New Form1
330     fMain.Show()
331     fMain.SelectedConfFile = DefaultConfFile
332     fMain.ActivateConfFile(DefaultConfFile)
333     fMain.ChangeLanguage(WhichIsTheLang)
334 End Sub
335
336 Private Sub BtCancel_Click(ByVal sender As System.Object, ByVal e As System.
337 EventArgs) Handles BtCancel.Click
338     Me.Close()
339 End Sub
340 Private Sub ChangeLanguage(ByVal WhichLanguage As Integer)
341     'WhichLanguage = 0 for Greek
342     'WhichLanguage = 1 for English
343     'WhichLanguage = 2 for Italian
344     'WhichLanguage = 3 for Spanish
345     'WhichLanguage = 4 for France
346
347     LblGreek.Text = MyClassCode.GetStr(WhichLanguage, 102)
348     LblEnglish.Text = MyClassCode.GetStr(WhichLanguage, 103)
349     LblItalian.Text = MyClassCode.GetStr(WhichLanguage, 259)
350     LblSpanish.Text = MyClassCode.GetStr(WhichLanguage, 260)
351     LblFrance.Text = MyClassCode.GetStr(WhichLanguage, 261)
352     BtOk.Text = MyClassCode.GetStr(WhichLanguage, 104)
353     BtCancel.Text = MyClassCode.GetStr(WhichLanguage, 105)
354     Me.Text = MyClassCode.GetStr(WhichLanguage, 106)
355 End Sub
356 Private Sub CheckIfThisShouldBeShown()
357     Dim MyFileName, WhichLang, IsShown As String
358
359     Dim key As RegistryKey
360
361     key = Registry.LocalMachine.OpenSubKey("Software\MyPtyx", True)
362     If key Is Nothing Then
363         key = Registry.LocalMachine.CreateSubKey("Software\MyPtyx")
364         key.SetValue("MyPtyx", "c:\windows\myconfs.ini")
365     End If
366

```

```

367     MyFileName = CType(key.GetValue("MyPtyx"), String)
368
369     Dim CheckFile As New FileInfo(MyFileName)
370     If Not CheckFile.Exists Then
371         Dim fParams As New FrmParameters
372         fParams.CreateConfsFile(MyFileName)
373         fParams.Dispose()
374     End If
375
376     If Trim(MyFileName) <> "c:\windows\myconfs.ini" Then
377         Dim fParams As New FrmParameters
378         fParams.CreateConfsFile("c:\windows\myconfs.ini")
379         fParams.Dispose()
380     End If
381
382     WhichLang = MyClassCode.FindParamValue(MyFileName, "Language", 0)
383     IsShown = MyClassCode.FindParamValue(MyFileName, "ShowLanguageSelection", 0)
384     If Val(IsShown) = 0 Then
385         'Form should not be shown
386         Dim fMain As New Form1
387         DoShowMe = False
388         fMain.SelectedConfFile = MyFileName
389         fMain.Show()
390         fMain.ChangeLanguage(Val(WhichLang))
391         fMain.ActivateConfFile(MyFileName)
392     Else
393         DefaultConfFile = MyFileName
394         DoShowMe = True
395         WhichIsTheLang = Val(WhichLang)
396         Select Case Val(WhichLang)
397             Case 0
398                 RBGreek.Checked = True
399             Case 1
400                 RBEnglish.Checked = True
401             Case 2
402                 RBItalian.Checked = True
403             Case 3
404                 RBSpanish.Checked = True
405             Case 4
406                 RBFrance.Checked = True
407         End Select
408     End If
409 End Sub
410
411 Public Sub AndTheChoiceWas(ByVal Whichlang As Integer, ByVal CallerIsNotRadio As Boolean)
412     WhichIsTheLang = Whichlang
413
414     Select Case Whichlang
415         Case 0
416             ChangeLanguage(Whichlang)
417         Case 1
418             ChangeLanguage(Whichlang)
419         Case 2
420             ChangeLanguage(Whichlang)
421         Case 3
422             ChangeLanguage(Whichlang)
423         Case 4
424             ChangeLanguage(Whichlang)
425     End Select
426
427     If CallerIsNotRadio Then
428         Select Case Whichlang
429             Case 0
430                 RBGreek.Checked = True
431             Case 1
432                 RBEnglish.Checked = True
433             Case 2
434                 RBItalian.Checked = True
435             Case 3
436                 RBSpanish.Checked = True
437             Case 4
438                 RBFrance.Checked = True
439         End Select
440     End If

```

```

441     End Sub
442
443     Private Sub PBGreek_Click(ByVal sender As System.Object, ByVal e As System.
444     EventArgs) Handles PBGreek.Click
445         AndTheChoiceWas(0, True)
446     End Sub
447
448     Private Sub LblGreek_Click(ByVal sender As System.Object, ByVal e As System.
449     EventArgs) Handles LblGreek.Click
450         AndTheChoiceWas(0, True)
451     End Sub
452
453     Private Sub PBEnglish_Click(ByVal sender As System.Object, ByVal e As System.
454     EventArgs) Handles PBEnglish.Click
455         AndTheChoiceWas(1, True)
456     End Sub
457
458     Private Sub LblEnglish_Click(ByVal sender As System.Object, ByVal e As System.
459     EventArgs) Handles LblEnglish.Click
460         AndTheChoiceWas(1, True)
461     End Sub
462
463     Private Sub PbItalian_Click(ByVal sender As System.Object, ByVal e As System.
464     EventArgs) Handles PbItalian.Click
465         AndTheChoiceWas(2, True)
466     End Sub
467
468     Private Sub LblItalian_Click(ByVal sender As System.Object, ByVal e As System.
469     EventArgs) Handles LblItalian.Click
470         AndTheChoiceWas(2, True)
471     End Sub
472
473     Private Sub PBSpanish_Click(ByVal sender As System.Object, ByVal e As System.
474     EventArgs) Handles PBSpanish.Click
475         AndTheChoiceWas(3, True)
476     End Sub
477
478     Private Sub LblSpanish_Click(ByVal sender As System.Object, ByVal e As System.
479     EventArgs) Handles LblSpanish.Click
480         AndTheChoiceWas(3, True)
481     End Sub
482
483     Private Sub PBFrance_Click(ByVal sender As System.Object, ByVal e As System.
484     EventArgs) Handles PBFrance.Click
485         AndTheChoiceWas(4, True)
486     End Sub
487
488     Private Sub LblFrance_Click(ByVal sender As System.Object, ByVal e As System.
489     EventArgs) Handles LblFrance.Click
490         AndTheChoiceWas(4, True)
491     End Sub
492
493     Private Sub RBGreek_CheckedChanged(ByVal sender As System.Object, ByVal e As System
494     .EventArgs) Handles RBGreek.CheckedChanged
495         AndTheChoiceWas(0, False)
496     End Sub
497
498     Private Sub RBEEnglish_CheckedChanged(ByVal sender As System.Object, ByVal e As
499     System.EventArgs) Handles RBEEnglish.CheckedChanged
500         AndTheChoiceWas(1, False)
501     End Sub
502
503     Private Sub RBITalian_CheckedChanged(ByVal sender As System.Object, ByVal e As
504     System.EventArgs) Handles RBITalian.CheckedChanged
505         AndTheChoiceWas(2, False)
506     End Sub
507
508     Private Sub RBSpanish_CheckedChanged(ByVal sender As System.Object, ByVal e As
509     System.EventArgs) Handles RBSpanish.CheckedChanged
510         AndTheChoiceWas(3, False)
511     End Sub
512
513     Private Sub RBFrance_CheckedChanged(ByVal sender As System.Object, ByVal e As
514     System.EventArgs) Handles RBFrance.CheckedChanged
515         AndTheChoiceWas(4, False)
516     End Sub

```

```
501     End Sub  
502 End Class  
503
```

```

1 Imports System.IO
2
3 Public Class ShowInGraph
4     Public Enum SelectLine
5         Line1 = 0 '1 'rpm = mode setpoint rpm
6         Line2 = 1 '2 'control action = mode setpoint control action
7     End Enum
8     Private ClosingCallerName As String
9     Private MyClassCode As New MyPtyxClass
10    Public FilePoint() As System.Drawing.Point
11    Public Enum TypeSecondaryLine
12        solid
13        dushed
14    End Enum
15    Public Structure GraphedFileSettings
16        Public NoPoints As Integer
17        Public Line1Legened As String
18        Public Line2Legened As String
19        Public Max1Value As Single
20        Public Max2Value As Single
21        Public value1Demicals As Single
22        Public value2Demicals As Single
23        Public DrawLine As SelectLine
24        Public TimeIntervalOfFileValues As Single
25        Public NoofHorLines As Int32
26        Public NoofHorDivLines As Int32
27        Public NoofVertLines As Int32
28        Public TypeofHorVerLines As TypeSecondaryLine
29        Public PenofVerHorLines As System.Drawing.Pen
30        Public LineCollor As System.Drawing.Pen
31        Public DisplayPoints As Boolean
32        Public PointSize As System.Drawing.Size
33        Public PointBrush As System.Drawing.Brush
34        Public SelectedPointBrush As System.Drawing.Brush
35        Public MeanDiv As Int32
36        Public IndexofLastDispayedPoint As Int32
37        Public TimeLengthhofVertLines As Int32
38        Public ShowTimeVertLine As Boolean
39        Public SelectedPoint As Integer
40    End Structure
41    Public Fileset As GraphedFileSettings
42    Public DislpayedIndex As Integer = 0
43    Public DisplayIndex As Integer = 500
44
45    Public FileValues() As Int32
46    Public ZoomFlag As Boolean = False
47    Public ZoomIndex As Integer = 0
48    Public WhichIsTheLang As Integer
49
50    Public Structure SYSTEMTIME
51        Public wYear As Int16
52        Public wMonth As Int16
53        Public wDayOfWeek As Int16
54        Public wDay As Int16
55        Public wHour As Int16
56        Public wMinute As Int16
57        Public wSecond As Int16
58        Public wMilliseconds As Int16
59    End Structure
60    Private Backup(50) As Int32
61    Public Enum FileMode
62        SetPointRpm = 0
63        SetPointControlAction = 1
64    End Enum
65    Public Mode As FileMode
66    Public msgBacktoFileControlform As System.Windows.Forms.DialogResult
67
68    Public Sub SetDefaults(ByVal MyMatrix() As Int32, ByVal MyMode As Integer)
69        Me.GBGraphProperties.Location = New System.Drawing.Point(180, 600)
70        Mode = MyMode
71        FileValues = MyMatrix
72        Fileset.DisplayPoints = True
73        Fileset.DrawLine = Mode
74        Fileset.Line1Legened = "RPM"
75        Fileset.Line2Legened = " % Control action"

```

```

76     Fileset.LineCollor = Pens.Blue
77     Fileset.Max1Value = 20000
78     Fileset.Max2Value = 10000
79     Fileset.value1Demicals = 1
80     Fileset.value2Demicals = 0.01
81     Fileset.NoofHorDivLines = 2
82     Fileset.NoofHorLines = 2
83     Fileset.NoofVertLines = 1
84     Fileset.NoPoints = 50
85     Fileset.PenofVerHorLines = Pens.DarkGray
86     Fileset.PointSize = New System.Drawing.Size(3, 3)
87     Fileset.PointBrush = Brushes.Blue
88     Fileset.SelectedPointBrush = Brushes.Chocolate
89     Fileset.TimeIntervalOfFileValues = 20
90     Fileset.TimeLengthhofVertLines = 1000
91     Fileset.TypeOfHorVerLines = TypeSecondaryLine.solid
92     Fileset.MeanDiv = 1
93     Fileset.TimeLengthhofVertLines = 100
94     Fileset.ShowTimeVertLine = True
95     Fileset.SelectedPoint = -1
96     Me.DislplayedIndex = FileValues.Length - 1
97     Me.Fileset.IndexofLastDispayedPoint = FileValues.Length - 1
98     Me.TrBTimeindex.Maximum = FileValues.Length - 1
99     Me.TrBTimeindex.Value = FileValues.Length - 1
100    Me.TrBNoPointsDraw.Minimum = 3
101    Me.TrBNoPointsDraw.Maximum = 10000
102    Me.LbMaximumPoints.Text = Me.TrBNoPointsDraw.Maximum.ToString
103    Me.TrBNoPointsDraw.Value = Fileset.NoPoints
104    Me.NUDConstVerticalLines.Minimum = 0
105    Me.NUDConstVerticalLines.Maximum = 100
106    Me.NUDTimeVerticalInterval.Minimum = 20
107    Me.NUDTimeVerticalInterval.Maximum = 100000
108    Me.RBShowTVLines.Checked = True
109    Fileset.TimeLengthhofVertLines = 1000
110    ReDim Backup(FileValues.Length - 1)
111    Array.Copy(FileValues, Backup, FileValues.Length)
112    Me.LbMaximumTime.Text = MyClassCode.ConvertMilsToMinsAndSecsAndMils((FileValues
.Length - 1) * Me.Fileset.TimeIntervalOfFileValues)
113    Me.DUDTypeofLines.Items.Clear()
114    Me.DUDTypeofLines.Items.Add(MyClassCode.GetStr(Me.WhichIsTheLang, 183))
115    Me.DUDTypeofLines.Items.Add(MyClassCode.GetStr(Me.WhichIsTheLang, 182))
116    Me.DUDTypeofLines.SelectedIndex = 0
117    Me.Left = (My.Computer.Screen.WorkingArea.Size.Width - Me.Width) / 2
118    Me.Top = (My.Computer.Screen.WorkingArea.Size.Height - Me.Height) / 2
119    Me.FormBorderStyle = Windows.Forms.FormBorderStyle.FixedSingle
120    Me.MaximizeBox = False
121    Me.StartPosition = FormStartPosition.CenterScreen
122 End Sub
123
124 Private Sub DrawDashedLine(ByVal e As Graphics, ByVal StartPoint As System.Drawing.
Point, ByVal EndPoint As System.Drawing.Point, ByVal pen As Pen, ByVal Pbox As
PictureBox)
125     Dim nextpoint As New System.Drawing.Point(StartPoint.X, StartPoint.Y)
126     Dim NoOfDushed As Integer = 1
127     'Y = aX+b
128     'a = Y2-Y1/X2-X1
129     'b = Y1-aX1
130     Dim a As Single
131     Dim b As Single
132     Dim Fpointdushed As New System.Drawing.Point(StartPoint.X, StartPoint.Y)
133     Dim Spointdushed As New System.Drawing.Point(Fpointdushed.X + 10, a *
Fpointdushed.X + b)
134     If (EndPoint.X - StartPoint.X) = 0 Then
135         Do While Fpointdushed.Y > EndPoint.Y
136             e.DrawLine(pen, Fpointdushed.X, Fpointdushed.Y, Fpointdushed.X,
Fpointdushed.Y + 10)
137             Fpointdushed.Y = Fpointdushed.Y - 20
138         Loop
139     Else
140         a = (EndPoint.Y - StartPoint.Y) / (EndPoint.X - StartPoint.X)
141         b = (StartPoint.Y - a * StartPoint.X)
142         Do While Fpointdushed.X < EndPoint.X
143             Spointdushed.X = Fpointdushed.X + 10
144             Spointdushed.Y = a * Fpointdushed.X + b
145             e.DrawLine(pen, Fpointdushed.X, Fpointdushed.Y, Spointdushed.X,

```

```

Spointdushed.Y)
146         Fpointdushed.X = Fpointdushed.X + 20
147     Loop
148 End If
149 End Sub
150 Private Sub DrawTheFile(ByVal D As System.Drawing.Graphics, ByVal Values() As Int32
, ByVal IndexofLastPoint As Int32, ByVal Settings As GraphedFileSettings, ByVal
Pbox As PictureBox)
151     Try
152         If IndexofLastPoint > 2 Then
153             If Settings.NoPoints < 2 Then
154                 GoTo jumptoend
155             End If
156             Dim Timeid As Integer = IndexofLastPoint
157             Dim pointsoffline(Settings.NoPoints * 2) As System.Drawing.PointF
158             Dim MaxValueFile1 As Int32 = 0
159             Select Case Settings.DrawLine
160                 Case SelectLine.Line1
161                     MaxValueFile1 = Settings.Max1Value
162                 Case SelectLine.Line2
163                     MaxValueFile1 = Settings.Max2Value
164             End Select
165             If Settings.ShowTimeVertLine = True Then
166                 Settings.IndexofLastDispayedPoint = IndexofLastPoint
167                 DrawTimeVertLines(D, Pbox, Values, Settings)
168             End If
169
170             Dim MeanDisplay As Boolean = False
171             If Settings.MeanDiv > 1 And Int(Settings.NoPoints / Settings.MeanDiv) >
= 1 Then
172                 Values = Me.MeanDisplay(Values, IndexofLastPoint, Settings)
173                 IndexofLastPoint = Values.Length - 1
174                 Settings.NoPoints = Values.Length
175                 Settings.TimeIntervalOfFileValues = Settings.
TimeIntervalOfFileValues * Settings.MeanDiv
176                 MeanDisplay = True
177             Else
178                 MeanDisplay = False
179             End If
180             Select Case MeanDisplay
181                 Case False
182                     If IndexofLastPoint >= Settings.NoPoints Then
183                         ReDim pointsoffline(Settings.NoPoints * 2 - 2)
184                         For z As Integer = 0 To Settings.NoPoints - 1
185                             Select Case z
186                                 Case 0
187                                     pointsoffline(z) = New System.Drawing.PointF((z
/ (Settings.NoPoints - 1)) * Pbox.Size.Width _
, Pbox.Size.Height - ((Values(IndexofLastPoint - (Settings.NoPoints - 1) + z) /
MaxValueFile1) * Pbox.Size.Height))
189                                 Case Else
190                                     pointsoffline((z * 2) - 1) = New System.Drawing.
PointF((z / (Settings.NoPoints - 1)) * Pbox.Size.Width _
, pointsoffline((z * 2) - 2).Y)
192                                     pointsoffline(z * 2) = New System.Drawing.PointF
(pointsoffline((z * 2) - 1).X _
, Pbox.Size.Height - ((Values(IndexofLastPoint - (Settings.NoPoints - 1) + z) /
MaxValueFile1) * Pbox.Size.Height))
194                                 End Select
195                             Next z
196
197                 Else
198                     ReDim pointsoffline(IndexofLastPoint * 2)
199                     For z As Integer = 0 To IndexofLastPoint
200                         Select Case z
201                             Case 0
202                                 pointsoffline(z) = New System.Drawing.PointF((z
/ (IndexofLastPoint)) * Pbox.Size.Width _
, Pbox.Size.Height - ((Values(IndexofLastPoint - (IndexofLastPoint) + z) /
MaxValueFile1) * Pbox.Size.Height))
204                             Case Else
205                                 pointsoffline((z * 2) - 1) = New System.Drawing.
PointF((z / (IndexofLastPoint)) * Pbox.Size.Width _
, pointsoffline((z * 2) - 2).Y)
207                                 pointsoffline(z * 2) = New System.Drawing.PointF

```

```

        (pointsoffline((z * 2) - 1).X _
, Pbox.Size.Height - ((Values(IndexofLastPoint - (IndexofLastPoint) + z) /
        MaxValueFile1) * Pbox.Size.Height))
209             End Select
210         Next
211     End If
212     Case True
213         If IndexofLastPoint >= Settings.NoPoints Then
214             ReDim pointsoffline(Settings.NoPoints - 1)
215             For l As Integer = 0 To Settings.NoPoints - 1 Step 1
216                 pointsoffline(l) = New System.Drawing.PointF((l /
217 (Settings.NoPoints - 1)) * Pbox.Size.Width _
                , Pbox.Size.Height - ((Values(IndexofLastPoint
218 - (Settings.NoPoints - 1) + 1) / MaxValueFile1) * Pbox.Size.Height))
                Next
219             Else
220                 ReDim pointsoffline(IndexofLastPoint)
221                 For l As Integer = 0 To IndexofLastPoint Step 1
222                     pointsoffline(l) = New System.Drawing.PointF((l /
223 (Settings.NoPoints - 1)) * Pbox.Size.Width _
                    , Pbox.Size.Height - ((Values(IndexofLastPoint
224 - (Settings.NoPoints - 1) + 1) / MaxValueFile1) * Pbox.Size.Height))
                    Next
225                 End If
226
227             End Select
228             'draw time
229             If Settings.NoofVertLines = 0 Then
230                 Settings.NoofVertLines = 1
231             End If
232
233             If Settings.NoofVertLines > 0 And Settings.ShowTimeVertLine = False
234 Then
235                 Dim timescale As Integer = ((Settings.NoPoints - 1) * Settings.
                TimeIntervalOfFileValues) / Settings.NoofVertLines
                D.DrawString("Time scale: " & Str(timescale) & "ms", New System.
236 Drawing.Font("Arial", 10), Brushes.Black, Pbox.Width - 140, Pbox.Height - 30) '&
                Str(IndexofLastPoint)
                End If
237
238                 D.DrawString(Me.MyClassCode.ConvertMilsToMinsAndSecsAndMils((Timeid) *
                Fileset.TimeIntervalOfFileValues), New System.Drawing.Font("Arial", 10), Brushes.
                Black, Pbox.Width - 100, Pbox.Height - 60)
239                 D.DrawString(Me.MyClassCode.ConvertMilsToMinsAndSecsAndMils(If((Timeid
                - Settings.NoPoints) >= 0, Timeid - Settings.NoPoints, 0) * Fileset.
                TimeIntervalOfFileValues), New System.Drawing.Font("Arial", 10), Brushes.Black, 10,
                Pbox.Height - 60)
240             'draw the line
241             Select Case MeanDisplay
242                 Case False
243                     D.DrawLines(Settings.LineCollor, pointsoffline)
244                 Case True
245                     D.DrawCurve(Settings.LineCollor, pointsoffline)
246             End Select
247
248             '-----draw points-----
249
250             If MeanDisplay = True Then
251                 If Settings.DisplayPoints = True Then
252                     For i As Int32 = 0 To Int((pointsoffline.Length + 1) / 2) - 1
253 Step 1
254                         D.FillRectangle(Settings.SelectedPointBrush, New System.
                Drawing.RectangleF(New System.Drawing.PointF(pointsoffline(i * 2).X - (Settings.
                PointSize.Width / 2) _
                , pointsoffline(i * 2).Y - (Settings.PointSize.Height /
255 2)), Settings.PointSize))
                Next
256                     End If
257                 Else
258                     If Settings.DisplayPoints = True Then
259                         For i As Int32 = 0 To Int((pointsoffline.Length + 1) / 2) - 1
260 Step 1
261                             D.FillRectangle(ReturnPointColor(Settings, Timeid - (
                pointsoffline.Length + 1) / 2) + 1 + i), New System.Drawing.RectangleF(New System.
                Drawing.PointF(pointsoffline(i * 2).X - (Settings.PointSize.Width / 2) _

```



```

261         , pointsofline(i * 2).Y - (Settings.PointSize.Height /
262         2)), ReturnSizeofSelectedPoint(Settings, Timeid - ((pointsofline.Length + 1) / 2) +
263         1 + i))) 'Settings.PointSize
264         Next
265     End If
266     End If
267     jumptoend: End If
268     '----- draw Horizontal lines -----
269     Dim linedisplay As Int16 = 0
270     If Settings.NoofHorLines = 0 Then
271         Settings.NoofHorLines = 1
272     End If
273     If Settings.NoofHorDivLines = 0 Then
274         Settings.NoofHorDivLines = 1
275     End If
276     Dim MaxValueFile As Int32
277     Select Case Settings.DrawLine
278     Case SelectLine.Line1
279         MaxValueFile = Settings.Max1Value
280     Case SelectLine.Line2
281         MaxValueFile = Settings.Max2Value
282     End Select
283     Settings.NoofHorLines = Settings.NoofHorLines + 1
284     For i As Integer = 0 To Settings.NoofHorLines * Settings.NoofHorDivLines
285         Step 1
286         Dim Yvalue As Double = Pbox.Height - (i / (Settings.NoofHorLines *
287         Settings.NoofHorDivLines)) * (Pbox.Height)
288
289         Dim ypoint As New System.Drawing.Point(Pbox.Size.Width, Yvalue)
290         Dim CaseSelectValue As Int16 = linedisplay * Settings.NoofHorDivLines
291         Select Case i
292         Case 0
293             GoTo solidline
294         Case Settings.NoofHorLines * Settings.NoofHorDivLines
295             Select Case Settings.DrawLine
296             Case SelectLine.Line1
297                 D.DrawString(Format((MaxValueFile) *
298                 CorrectDispalyedvalues(Me.Mode, Settings), "#####.#") & (Settings.Line1Legened), _
299                 New System.Drawing.Font("Arial", 10), Brushes.Black,
300                 New System.Drawing.Point(0, ypoint.Y + 10))
301             Case SelectLine.Line2
302                 D.DrawString(Format((MaxValueFile) *
303                 CorrectDispalyedvalues(Me.Mode, Settings), "#####.#") & (Settings.Line2Legened), _
304                 New System.Drawing.Font("Arial", 10), Brushes.Black,
305                 New System.Drawing.Point(0, ypoint.Y + 10))
306             End Select
307             GoTo solidline
308         Case CaseSelectValue
309             D.DrawString(Format(((MaxValueFile * (linedisplay)) / (Settings
310             .NoofHorLines)) * CorrectDispalyedvalues(Me.Mode, Settings), "#####.#"), _
311             New System.Drawing.Font("Arial", 10), Brushes.Black,
312             New System.Drawing.Point(0, ypoint.Y - 10))
313             Select Case Settings.TypeofHorVerLines
314             Case TypeSecondaryLine.dushed
315                 DrawDashedLine(D, New System.Drawing.Point(0, ypoint.Y)
316                 , New System.Drawing.Point(Pbox.Width, ypoint.Y), Pens.Gray, Pbox)
317                 linedisplay += 1
318             Case TypeSecondaryLine.solid
319                 D.DrawLine(Pens.Gray, 0, ypoint.Y, Pbox.Width, ypoint.
320                 Y)
321                 linedisplay += 1
322             End Select
323         Case Else
324             D.DrawLine(Pens.Black, 0, ypoint.Y, 10, ypoint.Y)
325         End Select
326     Next i
327     '-----draw const vertical lines -----
328     -----
329     If Settings.NoofVertLines > 0 And Settings.ShowTimeVertLine = False Then
330         Settings.NoofVertLines = Settings.NoofVertLines + 1
331         For i As Integer = 0 To Settings.NoofVertLines Step 1
332             Dim Xvalue As Single = (i / Settings.NoofVertLines) * (Pbox.Width)

```

```

323         Select Case Settings.TypeofHorVerLines
324             Case TypeSecondaryLine.dushed
325                 If i = 0 Then
326                     D.DrawLine(Pens.Gray, Xvalue, Pbox.Size.Height, Xvalue,
0)
327                 Else
328                     DrawDashedLine(D, New System.Drawing.Point(Xvalue, Pbox
.Size.Height), New System.Drawing.Point(Xvalue, 0), Pens.Gray, Pbox)
329                 End If
330             Case TypeSecondaryLine.solid
331                 D.DrawLine(Pens.Gray, Xvalue, Pbox.Size.Height, Xvalue, 0)
332         End Select
333     Next i
334 End If
335
336     Catch
337     Stop
338 End Try
339
340 End Sub
341
342 Private Function CorrectDispalyedvalues(ByVal mode As FileMode, ByVal sets As
GraphedFileSettings) As Single
343     If mode = FileMode.SetPointControlAction Then
344         Return sets.value2Demicals
345     Else
346         Return sets.value1Demicals
347     End If
348 End Function
349 Private Function ReturnPointColor(ByVal settings As GraphedFileSettings, ByVal
pointindex As Integer) As System.Drawing.Brush
350     If settings.SelectedPoint = pointindex Then
351         Return settings.SelectedPointBrush
352     Else
353         Return settings.PointBrush
354     End If
355 End Function
356 Private Function ReturnSizeofSelectedPoint(ByVal settings As GraphedFileSettings,
ByVal pointindex As Integer) As System.Drawing.Size
357     Select Case pointindex
358         Case settings.SelectedPoint
359             Return New System.Drawing.Size(2 * settings.PointSize.Width, 2 *
settings.PointSize.Height)
360         Case Else
361             Return settings.PointSize
362     End Select
363 End Function
364
365 Private Sub DrawTimeVertLines(ByVal e As System.Drawing.Graphics, ByVal Pbox As
PictureBox, ByVal Values() As Int32, ByRef Settings As GraphedFileSettings)
366     Dim MaxPoints As Integer = Math.Min(Settings.NoPoints - 1, Settings.
IndexofLastDispayedPoint - 1)
367     Dim DrawIntervalTime As Integer = MaxPoints * Settings.TimeIntervalOfFileValues
368     Dim NoofVerLines As Single = DrawIntervalTime / Settings.TimeLengthofVertLines
369     If NoofVerLines >= 50 Then
370         Do While NoofVerLines > 50 And (Settings.TimeLengthofVertLines Mod Settings
.TimeIntervalOfFileValues) <> 0
371             Settings.TimeLengthofVertLines += 1
372             NoofVerLines = DrawIntervalTime / Settings.TimeLengthofVertLines
373         Loop
374     End If
375     For i As Int32 = 0 To MaxPoints
376         If ((Settings.IndexofLastDispayedPoint - i) * Settings.
TimeIntervalOfFileValues Mod Settings.TimeLengthofVertLines) = 0 Then
377             Dim Xvalue As Single = (Pbox.Width) * (1 - (i / (MaxPoints)))
378             Select Case Settings.TypeofHorVerLines
379                 Case TypeSecondaryLine.dushed
380                     DrawDashedLine(e, New System.Drawing.Point(Xvalue, Pbox.Size.
Height), New System.Drawing.Point(Xvalue, 0), Pens.Gray, Pbox)
381                 Case TypeSecondaryLine.solid
382                     e.DrawLine(Pens.Gray, Xvalue, Pbox.Size.Height, Xvalue, 0)
383             End Select
384         End If
385     Next
386

```

```

387 End Sub
388 Private Function MeanDisplay(ByVal Values() As Int32, ByVal IndexofLastPoint As Integer,
389     Dim MeanValues() As Int32
390     ReDim MeanValues(Int(Settings.NoPoints / Settings.MeanDiv))
391
392     If IndexofLastPoint > Settings.NoPoints Then
393         For i As Integer = 0 To Int(Settings.NoPoints / Settings.MeanDiv)
394
395             Dim temp As Integer = 0
396             For k As Integer = 0 To Settings.MeanDiv - 1
397                 If (i * Settings.MeanDiv + k) <= (Settings.NoPoints - 1) Then
398                     temp += Values(IndexofLastPoint - (Settings.NoPoints - 1) + (i
* Settings.MeanDiv + k))
399                 Else
400                     temp += Values(IndexofLastPoint)
401                 End If
402
403             Next
404
405             MeanValues(i) = Int(temp / Settings.MeanDiv)
406         Next
407         Return MeanValues
408     Else
409         ReDim MeanValues(IndexofLastPoint)
410         For k As Integer = IndexofLastPoint To 0 Step -1
411             MeanValues(k) = Values(k)
412         Next
413         Return MeanValues
414     End If
415
416 End Function
417 Private Function FillPointCollor(ByVal IndexDisplayed As Integer, ByVal Pointid As Integer,
418     ByVal PointState() As Boolean, ByVal setts As GraphedFileSettings) As System.Drawing.Brush
419     If PointState(Pointid) = True Then
420         Return setts.SelectedPointBrush
421     Else
422         Return setts.PointBrush
423     End If
424 End Function
425 Private Sub PbFileForm_Paint(ByVal sender As System.Object, ByVal e As System.Windows.Forms.PaintEventArgs)
426     Handles PbFileForm.Paint
427     Me.Fileset.IndexofLastDispayedPoint = DisplayIndex
428     DrawTheFile(e.Graphics, FileValues, DisplayIndex, Me.Fileset, PbFileForm)
429 End Sub
430 Private Sub BtReturnToFileForm_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
431     Handles BtReturnToFileForm.Click
432     If CheckforChanges(Me.FileValues, Me.Backup) Then
433         If Validatevalues(Me.FileValues, Mode) = True Then
434             msgBacktoFileControlform = MsgBox(Me.MyClassCode.GetStr(Me.WhichIsTheLang, 291),
435                 MsgBoxStyle.YesNoCancel)
436             Select Case msgBacktoFileControlform
437                 Case Windows.Forms.DialogResult.Yes
438                     ExodosMeApouhkeyshNevnTimvmToolStripMenuItem_Click (BtReturnToFileForm, e)
439                 Case Windows.Forms.DialogResult.No
440                     TSMIExodosXvrisEfarmoghNeonTimvm_Click(BtReturnToFileForm, e)
441                     Backup.CopyTo(FileValues, 0)
442                     ClosingCallerName = BtReturnToFileForm.Name.ToString
443                     Me.Close()
444                 Case Windows.Forms.DialogResult.Cancel
445             End Select
446         Else
447             End If
448     Else
449         MsgBox(Me.MyClassCode.GetStr(Me.WhichIsTheLang, 290), MsgBoxStyle.OkOnly,
450             MyClassCode.GetStr(Me.WhichIsTheLang, 109))
451         Validatevalues(Me.FileValues, Mode)
452         ClosingCallerName = BtReturnToFileForm.Name.ToString
453         Me.Close()

```

```

453         End If
454
455     End Sub
456
457     Private Sub TrBTimeindex_ValueChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TrBTimeindex.ValueChanged
458         Me.DisplayIndex = TrBTimeindex.Value
459         Me.PbFileForm.Refresh()
460     End Sub
461
462     Private Sub TrBNoPointsDraw_ValueChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TrBNoPointsDraw.ValueChanged
463         Me.Fileset.NoPoints = TrBNoPointsDraw.Value
464         If Me.GBDisplayProp.Visible = True Then
465             Me.GBDisplayProp.Refresh()
466         End If
467         Me.PbFileForm.Refresh()
468     End Sub
469
470     Private Sub ToolStripMenuItem2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripMenuItem2.Click
471         If Me.GBDisplayProp.Visible Then
472             Me.GBDisplayProp.Visible = False
473             Me.TrBTimeindex.Enabled = True
474             Me.TrBTimeindex.Visible = True
475             Me.TrBNoPointsDraw.Enabled = True
476         Else
477             If Me.Fileset.MeanDiv <= 1 Then
478                 TrBSelectPoint.Maximum = Fileset.IndexofLastDispayedPoint
479                 TrBSelectPoint.Minimum = Fileset.IndexofLastDispayedPoint - (Math.Min
(Fileset.NoPoints, Fileset.IndexofLastDispayedPoint + 1)) + 1
480                 TrBSelectPoint.Value = TrBSelectPoint.Minimum
481                 Me.GBDisplayProp.Visible = True
482                 Me.GBGraphProperties.Visible = False
483             Else
484                 MsgBox(MyClassCode.GetStr(Me.WhichIsTheLang, 289), MsgBoxStyle.
Information, 0)
485             End If
486             Me.TrBNoPointsDraw.Enabled = False
487             Me.TrBTimeindex.Enabled = False
488             Me.TrBTimeindex.Visible = False
489         End If
490
491     End Sub
492
493     Private Sub BtCloseSelectValue_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtCloseSelectValue.Click
494         Me.TrBNoPointsDraw.Enabled = True
495         Me.TrBTimeindex.Enabled = True
496         Me.TrBTimeindex.Visible = True
497         Me.GBDisplayProp.Visible = False
498         Fileset.SelectedPoint = -1
499         Me.PbFileForm.Refresh()
500     End Sub
501
502     Private Sub TrBSelectPoint_Scroll(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles TrBSelectPoint.Scroll
503         Fileset.SelectedPoint = TrBSelectPoint.Value
504         Me.MTBValueDisplay.Text = Format(Me.FileValues(TrBSelectPoint.Value) * Me.
CorrectDispalyedvalues(Mode, Me.Fileset), "#####.#")
505         Me.TrBValue.Value = Me.FileValues(TrBSelectPoint.Value)
506         Me.TextBox1.Text = Me.MyClassCode.ConvertMilsToMinsAndSecsAndMils
(TrBSelectPoint.Value * Fileset.TimeIntervalOfFileValues)
507         Me.PbFileForm.Refresh()
508     End Sub
509
510     Private Sub BtAccessToValues_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles BtAccessToValues.Click
511         If TrBValue.Enabled = True Then
512             TrBValue.Enabled = False
513             BtAccessToValues.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 279)
514             BtRestoreValSelectPoint.Enabled = False
515             BtRestoreAllDispPoints.Enabled = False
516         Else
517             TrBValue.Enabled = True

```

```

518         Fileset.MeanDiv = 1
519         Fileset.DisplayPoints = True
520         BtAccessToValues.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 288)
521         BtRestoreValSelectPoint.Enabled = True
522         BtRestoreAllDispPoints.Enabled = True
523     End If
524 End Sub
525
526 Private Sub TrBValue_Scroll(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles TrBValue.Scroll
527     If Me.TrBValue.Enabled = True And Me.Fileset.SelectedPoint >= 0 Then
528         Me.MTBValueDisplay.Text = (TrBValue.Value * Me.CorrectDispalyedvalues(Mode,
Me.Fileset)).ToString
529         FileValues(Me.Fileset.SelectedPoint) = Me.TrBValue.Value
530         Me.PbFileForm.Refresh()
531     End If
532 End Sub
533
534 Private Sub GraphPropertiesToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles GraphPropertiesToolStripMenuItem.Click
535     If Me.GBGraphProperties.Visible = True Then
536         Me.GBGraphProperties.Visible = False
537         Me.TrBTimeindex.Visible = True
538     Else
539         Me.GBGraphProperties.Visible = True
540         Me.GBDisplayProp.Visible = False
541         Me.TrBTimeindex.Visible = False
542     End If
543 End Sub
544
545 Private Sub RBShowConstantVLines_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RBShowConstantVLines.CheckedChanged
546     If RBShowConstantVLines.Checked = True Then
547         NUDConstVerticalLines.Enabled = True
548         NUDConstVerticalLines.Value = Fileset.NoofVertLines
549         Fileset.ShowTimeVertLine = False
550         Me.PbFileForm.Refresh()
551     Else
552         Fileset.ShowTimeVertLine = True
553         NUDConstVerticalLines.Enabled = False
554         NUDConstVerticalLines.Value = Fileset.NoofVertLines
555     End If
556 End Sub
557
558 Private Sub RBShowTVLines_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RBShowTVLines.CheckedChanged
559     If RBShowTVLines.Checked = True Then
560         Fileset.ShowTimeVertLine = True
561         NUDTimeVerticalInterval.Enabled = True
562         NUDTimeVerticalInterval.Value = Fileset.TimeLengthhofVertLines
563         Me.PbFileForm.Refresh()
564     Else
565         Fileset.ShowTimeVertLine = False
566         NUDTimeVerticalInterval.Enabled = False
567         NUDTimeVerticalInterval.Value = Fileset.TimeLengthhofVertLines
568     End If
569 End Sub
570
571 Private Sub NUDTimeVerticalInterval_ValueChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles NUDTimeVerticalInterval.ValueChanged
572     Fileset.TimeLengthhofVertLines = NUDTimeVerticalInterval.Value
573 End Sub
574
575 Private Sub NUDConstVerticalLines_ValueChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles NUDConstVerticalLines.ValueChanged
576     Fileset.NoofVertLines = NUDConstVerticalLines.Value
577 End Sub
578
579 Private Sub RBDisplayPoints_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles RBDisplayPoints.CheckedChanged
580     If Me.RBDisplayPoints.Checked = True Then
581         Fileset.DisplayPoints = True
582     Else
583         Fileset.DisplayPoints = False
584     End If

```

```

585     Me.PbFileForm.Refresh()
586 End Sub
587
588 Private Sub RBNNotDisplayPoints_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RBNNotDisplayPoints.CheckedChanged
589     If RBNNotDisplayPoints.Checked = True Then
590         Fileset.DisplayPoints = False
591     Else
592         Fileset.DisplayPoints = True
593     End If
594     Me.PbFileForm.Refresh()
595 End Sub
596
597 Private Sub BtCloseGrpGraphProp_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtCloseGrpGraphProp.Click
598     Me.GBGraphProperties.Visible = False
599     Me.TrBTimeindex.Visible = True
600     Me.PbFileForm.Refresh()
601 End Sub
602
603 Private Sub NUDMeanDisplay_VisibleChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NUDMeanDisplay.VisibleChanged
604     Me.NUDMeanDisplay.Value = Fileset.MeanDiv
605 End Sub
606
607 Private Sub NUDMeanDisplay_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NUDMeanDisplay.ValueChanged
608     Fileset.MeanDiv = Me.NUDMeanDisplay.Value
609     Me.PbFileForm.Refresh()
610 End Sub
611
612 Private Sub GBDisplayProp_VisibleChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles GBDisplayProp.VisibleChanged
613     If GBDisplayProp.Visible = True Then
614         Me.BtRestoreAllDispPoints.Enabled = False
615         Me.BtRestoreValSelectPoint.Enabled = False
616         Me.BtAccessToValues.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 279)
617         Me.TrBValue.Enabled = False
618         Me.TrBTimeindex.Visible = False
619         Me.TrBValue.Minimum = 0
620         If Mode = FileMode.SetPointRpm Then
621             Me.TrBValue.Maximum = Fileset.Max1Value
622         Else
623             Me.TrBValue.Maximum = Fileset.Max2Value
624         End If
625         Me.Fileset.SelectedPoint = -1
626     Else
627         Me.TrBTimeindex.Visible = True
628     End If
629 End Sub
630
631
632 Private Sub GBGraphProperties_VisibleChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles GBGraphProperties.VisibleChanged
633     If Me.GBGraphProperties.Visible Then
634         Me.TrBTimeindex.Visible = False
635         Me.NUDTimeVerticalInterval.Value = Fileset.TimeLengthofVertLines
636         Me.NUDConstVerticalLines.Value = Fileset.NoofVertLines
637         Me.RBDisplayPoints.Checked = Fileset.DisplayPoints
638         Me.NUDHorLines.Value = Fileset.NoofHorLines
639     Else
640         Me.TrBTimeindex.Visible = True
641         Me.TrBTimeindex.Enabled = True
642         Me.TrBNoPointsDraw.Enabled = True
643     End If
644 End Sub
645
646
647 Private Sub NUDHorLines_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NUDHorLines.ValueChanged
648     Fileset.NoofHorLines = NUDHorLines.Value
649     Me.PbFileForm.Refresh()
650 End Sub
651
652

```

```

653 Private Sub LbMaxValue_VisibleChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles LbMaxValue.VisibleChanged
654     If Mode = FileMode.SetPointRpm Then
655         Me.LbMaxValue.Text = (Fileset.Max1Value * Me.CorrectDispalyedvalues (FileMode.SetPointRpm, Me.Fileset)).ToString
656         Me.LbMinValue.Text = 0
657     Else
658         Me.LbMaxValue.Text = (Fileset.Max2Value * Me.CorrectDispalyedvalues (FileMode.SetPointControlAction, Me.Fileset)).ToString
659         Me.LbMinValue.Text = 0
660     End If
661 End Sub
662
663
664 Private Sub TrBNoPointsDraw_VisibleChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TrBNoPointsDraw.VisibleChanged
665     TrBNoPointsDraw.Value = Me.Fileset.NoPoints
666 End Sub
667 Public Sub WhichLang(ByVal WhichLang As Integer)
668     WhichIsTheLang = WhichLang
669
670     GBGraphProperties.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 155)
671     RBShowConstantVLines.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 176)
672     RBShowTVLines.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 270)
673     RBDisplayPoints.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 271)
674     RBNotDisplayPoints.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 272)
675     LbMeanDisplay.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 273)
676     LbHorlines.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 175)
677     BtCloseGrpGraphProp.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 196)
678     RTBNoOfPoints.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 174)
679     ToolStripMenuItem1.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 275)
680     ToolStripMenuItem2.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 276)
681     GraphPropertiesToolStripMenuItem.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 155)
682     GBDisplayProp.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 277)
683     LbSelectedPoint.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 278)
684     BtAccessToValues.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 279)
685     LbYvalue.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 280)
686     LbTime.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 161) & " : "
687     BtCloseSelectValue.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 196)
688     BtReturnToFileForm.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 275)
689     TSMIExodosXvrisEfarmoghNeonTimvm.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 281)
690     ExodosMeApouhkeyshNevnTimvmToolStripMenuItem.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 282)
691     TSMIEpenaforaTonArxikvnTimon.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 283)
692     ToolStripMenuItem1.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 284)
693     Me.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 285)
694     BtRestoreValSelectPoint.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 292)
695     BtRestoreAllDispPoints.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 293)
696     LbMaximumPoints.Text = Me.TrBNoPointsDraw.Maximum.ToString
697     LbTypeofLines.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 184)
698 End Sub
699
700 Private Sub DUDTypeofLines_SelectedItemChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DUDTypeofLines.SelectedItemChanged
701     Select Case DUDTypeofLines.SelectedIndex
702     Case 0
703         Fileset.TypeofHorVerLines = TypeSecondaryLine.solid
704     Case 1
705         Fileset.TypeofHorVerLines = TypeSecondaryLine.dushed
706     End Select
707 End Sub
708
709 Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
710     If DisplayIndex + 1 < FileValues.Length - 1 Then
711         Me.TrBValue.Enabled = False
712         BtAccessToValues.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 279)
713         Me.DisplayIndex += 1
714         Me.PbFileForm.Refresh()
715         TrBSelectPoint.Maximum = Fileset.IndexofLastDispayedPoint
716         TrBSelectPoint.Minimum = Fileset.IndexofLastDispayedPoint - (Math.Min (Fileset.NoPoints, Fileset.IndexofLastDispayedPoint + 1)) + 1
717     End If

```

```

718     End Sub
719
720     Private Sub BtGotoPreviewsPoint_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtGotoPreviewsPoint.Click
721         If DisplayIndex - 1 > 0 Then
722             Me.TrBValue.Enabled = False
723             BtAccessToValues.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 279)
724             Me.DisplayIndex -= 1
725             Me.PbFileForm.Refresh()
726             TrBSelectPoint.Maximum = Fileset.IndexofLastDispayedPoint
727             TrBSelectPoint.Minimum = Fileset.IndexofLastDispayedPoint - (Math.Min (Fileset.NoPoints, Fileset.IndexofLastDispayedPoint + 1)) + 1
728         End If
729     End Sub
730
731     Private Sub BtMovetoLast_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtMovetoLast.Click
732         Me.TrBValue.Enabled = False
733         BtAccessToValues.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 279)
734         Me.DisplayIndex = Me.FileValues.Length - 1
735         Me.PbFileForm.Refresh()
736         TrBSelectPoint.Maximum = Fileset.IndexofLastDispayedPoint
737         TrBSelectPoint.Minimum = Fileset.IndexofLastDispayedPoint - (Math.Min(Fileset.NoPoints, Fileset.IndexofLastDispayedPoint + 1)) + 1
738     End Sub
739
740     Private Sub BtGotoStartPoint_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtGotoStartPoint.Click
741         Me.TrBValue.Enabled = False
742         BtAccessToValues.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 279)
743         Me.DisplayIndex = Fileset.NoPoints
744         Me.PbFileForm.Refresh()
745         TrBSelectPoint.Maximum = Fileset.IndexofLastDispayedPoint
746         TrBSelectPoint.Minimum = Fileset.IndexofLastDispayedPoint - (Math.Min(Fileset.NoPoints, Fileset.IndexofLastDispayedPoint + 1)) + 1
747     End Sub
748
749     Private Sub BtRestoreAllDispPoints_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtRestoreAllDispPoints.Click
750         Array.Copy(Me.Backup, Me.TrBSelectPoint.Minimum, Me.FileValues, Me.TrBSelectPoint.Minimum, Me.TrBSelectPoint.Maximum - Me.TrBSelectPoint.Minimum + 1)
751         Me.PbFileForm.Refresh()
752     End Sub
753
754     Private Sub BtRestoreValSelectPoint_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtRestoreValSelectPoint.Click
755         If (Fileset.SelectedPoint >= 0) And (Fileset.SelectedPoint <= FileValues.Length - 1) Then
756             FileValues(Fileset.SelectedPoint) = Backup(Fileset.SelectedPoint)
757             Me.PbFileForm.Refresh()
758         End If
759     End Sub
760     Private Function Validatevalues(ByVal matrix() As Integer, ByVal mode As FileMode) As Boolean
761         Dim maxvalue As Integer
762         Select Case mode
763             Case FileMode.SetPointRpm
764                 maxvalue = Fileset.Max1Value
765             Case FileMode.SetPointControlAction
766                 maxvalue = Fileset.Max2Value
767         End Select
768         For i As Integer = 0 To matrix.Length - 1
769             If matrix(i) > maxvalue Then
770                 MsgBox(MyClassCode.GetStr(Me.WhichIsTheLang, 286) & " , " & MyClassCode.GetStr(Me.WhichIsTheLang, 280) & ((FileValues(i) * CorrectDispalyedvalues(mode, Fileset)).ToString) & " " & MyClassCode.GetStr(Me.WhichIsTheLang, 161) & " :" & (i * Fileset.TimeIntervalOfFileValues).ToString, MsgBoxStyle.Information)
771                 Return False
772             End If
773         End If
774         Next
775         Return True
776     End Function
777     Private Function CheckforChanges(ByVal mat1() As Int32, ByVal mat2() As Int32) As Boolean

```



```

778     Try
779         For i As Int32 = 0 To Math.Max(mat1.Length, mat2.Length) - 1
780             Select Case mat1(i).CompareTo(mat2(i)) ' 0 if equal ,>0 if mat1(i)> mat2(i),<0 if mat1(i)<mat2(i)
781                 Case 0
782                 Case Else
783                     Return True
784             End Select
785         Next
786         Return False
787     Catch ex As Exception
788         Return True
789     End Try
790 End Function
791
792 Private Sub TSMIExodosXvrisEfarmoghNeonTimvm_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TSMIExodosXvrisEfarmoghNeonTimvm.Click
793     If Validatevalues(Me.Backup, Mode) = True Then
794         Backup.CopyTo(FileValues, 0)
795         ClosingCallerName = TSMIExodosXvrisEfarmoghNeonTimvm.Name.ToString
796         Me.Close()
797     Else
798         If MsgBox(MyClassCode.GetStr(Me.WhichIsTheLang, 286), MsgBoxStyle.Critical) =
= MsgBoxResult.Ok Then
799             ClosingCallerName = TSMIExodosXvrisEfarmoghNeonTimvm.Name.ToString
800             Me.Close()
801         End If
802     End If
803 End Sub
804
805 Private Sub ExodosMeApouhkeyshNevnTimvmToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ExodosMeApouhkeyshNevnTimvmToolStripMenuItem.Click
806     If Validatevalues(Me.FileValues, Mode) = True Then
807         ClosingCallerName = ExodosMeApouhkeyshNevnTimvmToolStripMenuItem.Name.
ToString
808         Me.Close()
809     Else
810     End If
811 End Sub
812
813 Private Sub TSMIEpenaforaTonArxikvnTimon_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TSMIEpenaforaTonArxikvnTimon.Click
814     If MsgBox(MyClassCode.GetStr(Me.WhichIsTheLang, 283), MsgBoxStyle.OkCancel) =
MsgBoxResult.Ok Then
815         Me.Backup.CopyTo(Me.FileValues, 0)
816         Me.PbFileForm.Refresh()
817     Else
818     End If
819 End Sub
820 Private Sub MenuStrip1_ItemClicked(ByVal sender As System.Object, ByVal e As System
.Windows.Forms.ToolStripItemClickedEventArgs) Handles MenuStrip1.ItemClicked
821 End Sub
822 Private Sub ShowInGraph_Load(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MyBase.Load
823 End Sub
824 Private Sub ShowInGraph_FormClosing(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing
825     If ClosingCallerName = "" Then
826         Me.Backup.CopyTo(Me.FileValues, 0)
827     End If
828 End Sub
829 Private Sub TrBNoPointsDraw_Scroll(ByVal sender As System.Object, ByVal e As System
.EventArgs) Handles TrBNoPointsDraw.Scroll
830 End Sub
831 End Class
832

```

```

1 Imports System.Windows.Forms
2 Imports System.ComponentModel
3 Imports System.Threading
4 Imports System.Timers
5
6 Public Class SpecialFile
7
8     Public Structure ReturnFileValues
9         Public NoError As Boolean
10        Public ValueDispStrg As String
11    End Structure
12    Public Structure MyMatrixLooksLike
13        Public MyFunction As String
14        Public MyStartTime As Int64
15        Public MyStopTime As Int64
16    End Structure
17
18    Public Const MinStrofes As Integer = 0, MinControlAction As Integer = 0,
ControlActionDivider As Integer = 100
19    Public Const MaxStrofes As Integer = 20000, MaxControlAction As Integer = 10000
20    Public Const TimeChanges As Integer = 20
21    Delegate Sub InvokeDelagateCheck(ByVal RdBName As String)
22    Public Line3 As String = "", line4 As String = "", Line5 As String = ""
23    Private CurrentLoadPosition As Int64 = 0
24    Public WhichIsTheLang, Current50ada, Synolo50adwn, MatrixValues(),
ImportFromFileMatrix(), MatrixDimension, ImportedFileLines As Integer
25    Public ValidatedFunctions() As MyMatrixLooksLike, CheckedFunction As String,
ValidatedFunctionsIndex As Integer
26    Public EvaluationOK As Boolean = False, RegistrationOK As Boolean = False
27    Private FileToBeInsertedChecked As Boolean
28    Private MyClassCode As New MyPtyxClass
29    Private TheSelectedConfFile, ConfSendDataFile, ConfRepeat, ConfRepeatTimes,
SystemDecimalSeperator, ImportFile As String
30    Public MyObjectMatrix(49) As MaskedTextBox, PreviousPercentage As Integer = 0
31    Private LoadDelayTimer As New System.Timers.Timer
32
33    Public Sub New()
34        InitializeComponent()
35        SystemDecimalSeperator = FindDecimalSeperator()
36        AddHandler LoadDelayTimer.Elapsed, AddressOf DelayedLoadfile
37        Me.LoadDelayTimer.AutoReset = False
38        Me.LoadDelayTimer.Interval = 500
39        LoadDelayTimer.Start()
40    End Sub
41
42    Private Sub SpecialFile_Load(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MyBase.Load
43        FixToolTips()
44        DeclareAcceptableSymbols()
45        CheckFunctionButtonsEnabledStatus()
46    End Sub
47
48    Public Sub WhichLang(ByVal WhichLang As Integer)
49        WhichIsTheLang = WhichLang
50
51        Me.Text = MyClassCode.GetStr(WhichLang, 198)
52        Me.GrBxDatSave.Text = MyClassCode.GetStr(WhichLang, 50)
53        Me.LblSaveData.Text = MyClassCode.GetStr(WhichLang, 54)
54        Me.GrBSelectMode.Text = MyClassCode.GetStr(WhichLang, 199)
55        Me.RBStrofes.Text = MyClassCode.GetStr(WhichLang, 154)
56        Me.RBGkazi.Text = MyClassCode.GetStr(WhichLang, 153)
57        Me.GrBCreateFileBy.Text = MyClassCode.GetStr(WhichLang, 200)
58        Me.RBManual.Text = MyClassCode.GetStr(WhichLang, 201)
59        Me.RBFromFunction.Text = MyClassCode.GetStr(WhichLang, 202)
60        Me.RBFromFile.Text = MyClassCode.GetStr(WhichLang, 203)
61        Me.GbManual.Text = MyClassCode.GetStr(WhichLang, 204)
62        Me.GBRepeat.Text = MyClassCode.GetStr(WhichLang, 207)
63        Me.LblRepeatTimes.Text = MyClassCode.GetStr(WhichLang, 208)
64        Me.BtSave.Text = MyClassCode.GetStr(WhichLang, 50)
65        Me.BtShowValuesInGraph.Text = MyClassCode.GetStr(WhichLang, 209)
66        Me.GBFromFunction.Text = MyClassCode.GetStr(WhichLang, 210)
67        Me.LblInitialTime.Text = MyClassCode.GetStr(WhichLang, 211)
68        Me.LblFinalTime.Text = MyClassCode.GetStr(WhichLang, 212)
69        Me.LblCreateFunction.Text = MyClassCode.GetStr(WhichLang, 213)
70        Me.BtCheckFunction.Text = MyClassCode.GetStr(WhichLang, 214)

```

```

71     Me.LblSelectedFunction.Text = MyClassCode.GetStr(WhichLang, 215)
72     Me.LblFirstValue.Text = MyClassCode.GetStr(WhichLang, 220)
73     Me.LblLastValue.Text = MyClassCode.GetStr(WhichLang, 221)
74     Me.BtExecuteRepeat.Text = MyClassCode.GetStr(WhichLang, 222)
75     Me.BtClose.Text = MyClassCode.GetStr(WhichLang, 58)
76     Me.LblTimePeriod.Text = MyClassCode.GetStr(WhichLang, 225) & " 0 ms - 980 ms"
77     Me.LblGoToPage.Text = MyClassCode.GetStr(WhichLang, 226)
78     Me.LblPageTadeFromTade.Text = MyClassCode.GetStr(WhichLang, 227) & "1" &
MyClassCode.GetStr(WhichLang, 228) & "1"
79     Me.LblStartRepeatFrom.Text = MyClassCode.GetStr(WhichLang, 230)
80     Me.GBFromFile.Text = MyClassCode.GetStr(WhichLang, 203)
81     Me.GBLoadFile.Text = MyClassCode.GetStr(WhichLang, 232)
82     Me.LblLoadFileName.Text = MyClassCode.GetStr(WhichLang, 81)
83     Me.GbGeneralInstructions.Text = MyClassCode.GetStr(WhichLang, 233)
84     Me.LblGeneralInstruction.Text = MyClassCode.GetStr(WhichLang, 234)
85     Me.GbImportFileData.Text = MyClassCode.GetStr(WhichLang, 235)
86     Me.BtImportFile.Text = MyClassCode.GetStr(WhichLang, 236)
87     Me.BtCheckFile.Text = MyClassCode.GetStr(WhichLang, 239)
88     Me.TPKNknowFunctions.Text = MyClassCode.GetStr(WhichLang, 241)
89     Me.TPSelectedFunctions.Text = MyClassCode.GetStr(WhichLang, 242)
90     Me.BtRegisterFunction.Text = MyClassCode.GetStr(WhichLang, 243)
91     Me.BtReset.Text = MyClassCode.GetStr(WhichLang, 244)
92     Me.TPAcceptableSymbols.Text = MyClassCode.GetStr(WhichLang, 245)
93     Me.GBRunFunction.Text = MyClassCode.GetStr(WhichLang, 247)
94     Me.GBTaskProgress.Text = MyClassCode.GetStr(WhichLang, 263)
95     Me.BtCancelTask.Text = MyClassCode.GetStr(WhichLang, 264)
96
97     BaptiseLabels(Synolo50adwn)
98 End Sub
99
100 Public Sub BaptiseLabels(ByVal MyCounter As Integer)
101     LblValue1.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 1) & IIf(WhichIsTheLang = 0, "η", "") & ":"
102     LblValue2.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 2) & IIf(WhichIsTheLang = 0, "η", "") & ":"
103     LblValue3.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 3) & IIf(WhichIsTheLang = 0, "η", "") & ":"
104     LblValue4.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 4) & IIf(WhichIsTheLang = 0, "η", "") & ":"
105     LblValue5.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 5) & IIf(WhichIsTheLang = 0, "η", "") & ":"
106     LblValue6.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 6) & IIf(WhichIsTheLang = 0, "η", "") & ":"
107     LblValue7.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 7) & IIf(WhichIsTheLang = 0, "η", "") & ":"
108     LblValue8.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 8) & IIf(WhichIsTheLang = 0, "η", "") & ":"
109     LblValue9.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 9) & IIf(WhichIsTheLang = 0, "η", "") & ":"
110     LblValue10.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 10) & IIf(WhichIsTheLang = 0, "η", "") & ":"
111     LblValue11.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 11) & IIf(WhichIsTheLang = 0, "η", "") & ":"
112     LblValue12.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 12) & IIf(WhichIsTheLang = 0, "η", "") & ":"
113     LblValue13.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 13) & IIf(WhichIsTheLang = 0, "η", "") & ":"
114     LblValue14.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 14) & IIf(WhichIsTheLang = 0, "η", "") & ":"
115     LblValue15.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 15) & IIf(WhichIsTheLang = 0, "η", "") & ":"
116     LblValue16.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 16) & IIf(WhichIsTheLang = 0, "η", "") & ":"
117     LblValue17.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 17) & IIf(WhichIsTheLang = 0, "η", "") & ":"
118     LblValue18.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 18) & IIf(WhichIsTheLang = 0, "η", "") & ":"
119     LblValue19.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 19) & IIf(WhichIsTheLang = 0, "η", "") & ":"
120     LblValue20.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 20) & IIf(WhichIsTheLang = 0, "η", "") & ":"
121     LblValue21.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 21) & IIf(WhichIsTheLang = 0, "η", "") & ":"
122     LblValue22.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 22) & IIf(WhichIsTheLang = 0, "η", "") & ":"

```

```

123     LblValue23.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 23) & IIf(WhichIsTheLang = 0, "η", "") & ":"
124     LblValue24.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 24) & IIf(WhichIsTheLang = 0, "η", "") & ":"
125     LblValue25.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 25) & IIf(WhichIsTheLang = 0, "η", "") & ":"
126     LblValue26.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 26) & IIf(WhichIsTheLang = 0, "η", "") & ":"
127     LblValue27.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 27) & IIf(WhichIsTheLang = 0, "η", "") & ":"
128     LblValue28.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 28) & IIf(WhichIsTheLang = 0, "η", "") & ":"
129     LblValue29.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 29) & IIf(WhichIsTheLang = 0, "η", "") & ":"
130     LblValue30.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 30) & IIf(WhichIsTheLang = 0, "η", "") & ":"
131     LblValue31.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 31) & IIf(WhichIsTheLang = 0, "η", "") & ":"
132     LblValue32.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 32) & IIf(WhichIsTheLang = 0, "η", "") & ":"
133     LblValue33.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 33) & IIf(WhichIsTheLang = 0, "η", "") & ":"
134     LblValue34.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 34) & IIf(WhichIsTheLang = 0, "η", "") & ":"
135     LblValue35.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 35) & IIf(WhichIsTheLang = 0, "η", "") & ":"
136     LblValue36.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 36) & IIf(WhichIsTheLang = 0, "η", "") & ":"
137     LblValue37.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 37) & IIf(WhichIsTheLang = 0, "η", "") & ":"
138     LblValue38.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 38) & IIf(WhichIsTheLang = 0, "η", "") & ":"
139     LblValue39.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 39) & IIf(WhichIsTheLang = 0, "η", "") & ":"
140     LblValue40.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 40) & IIf(WhichIsTheLang = 0, "η", "") & ":"
141     LblValue41.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 41) & IIf(WhichIsTheLang = 0, "η", "") & ":"
142     LblValue42.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 42) & IIf(WhichIsTheLang = 0, "η", "") & ":"
143     LblValue43.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 43) & IIf(WhichIsTheLang = 0, "η", "") & ":"
144     LblValue44.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 44) & IIf(WhichIsTheLang = 0, "η", "") & ":"
145     LblValue45.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 45) & IIf(WhichIsTheLang = 0, "η", "") & ":"
146     LblValue46.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 46) & IIf(WhichIsTheLang = 0, "η", "") & ":"
147     LblValue47.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 47) & IIf(WhichIsTheLang = 0, "η", "") & ":"
148     LblValue48.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 48) & IIf(WhichIsTheLang = 0, "η", "") & ":"
149     LblValue49.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 49) & IIf(WhichIsTheLang = 0, "η", "") & ":"
150     LblValue50.Text = MyClassCode.GetStr(WhichIsTheLang, 206) & Str(MyCounter * 50
+ 50) & IIf(WhichIsTheLang = 0, "η", "") & ":"
151 End Sub
152
153 Public Sub SetDefaults(ByVal SelectedConfFile As String)
154     SetInitialFormAttributes()
155
156     If MyClassCode.IsValidFile(SelectedConfFile) Then
157         TheSelectedConfFile = SelectedConfFile
158         ConfSendDataFile = MyClassCode.FindParamValue(SelectedConfFile, "Outgoing
File", WhichIsTheLang)
159         TxtSaveData.Text = ConfSendDataFile
160     Else
161         TheSelectedConfFile = ""
162     End If
163 End Sub
164
165 Public Sub AdjustAccordingToConfFile(ByVal FileName As String, ByVal sender As
System.Object, ByVal e As DoWorkEventArgs) 's
166     If MyClassCode.IsValidFile(FileName) Then
167         Try

```

```

168         Dim sr As IO.StreamReader = New IO.StreamReader(FileName)
169         Dim CheckFile As New IO.FileInfo(FileName)
170
171         Dim MaxLength As Integer = CheckFile.Length
172
173         CurrentLoadPosition = 0
174         Dim line As String
175         Dim met As Integer
176         met = 0
177         MatrixDimension = 0
178         Do
179             met += 1
180             line = sr.ReadLine()
181             If line = Nothing Then
182                 Exit Do
183             End If
184             Select Case met
185                 Case Is > 6
186                     ReDim Preserve MatrixValues(met - 7)
187
188                     MatrixValues(met - 7) = IIf(Line3 = RBStrofes.Name, Val
189 (line), Val(line) * ControlActionDivider)
190                     MatrixDimension += 1
191                 Case 3
192                     If line.IndexOf("Control RPM") <> -1 Then
193                         Line3 = RBStrofes.Name
194                     Else
195                         Line3 = RBGkazi.Name
196                     End If
197                 Case 4
198                     If line.IndexOf("Manually") <> -1 Then
199                         line4 = RBManual.Name
200                     Else
201                         If line.IndexOf("With Function") <> -1 Then
202                             line4 = RBFromFunction.Name
203                         Else
204                             line4 = RBFromFile.Name
205                         End If
206                     End If
207                 Case 5
208                     Line5 = line
209             End Select
210             If Me.LoadFileBackgroundWorker.CancellationPending = True Then
211                 e.Cancel = True
212             End If
213             Loop Until line Is Nothing
214             sr.Close()
215         Catch Exp As Exception
216             ' Do nothing for now
217         End Try
218     End If
219 End Sub
220 Private Sub BtSaveData_Click(ByVal sender As System.Object, ByVal e As System.
221 EventArgs) Handles BtSaveData.Click
222     Try
223         Dim myStream As IO.Stream
224         Dim openFileDialog1 As New OpenFileDialog
225
226         openFileDialog1.InitialDirectory = "c:\Windows\Desktop"
227         openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*"
228         openFileDialog1.FilterIndex = 1
229         openFileDialog1.RestoreDirectory = True
230         openFileDialog1.Multiselect = False
231
232         If openFileDialog1.ShowDialog() = System.Windows.Forms.DialogResult.OK And
233         openFileDialog1.FileName.EndsWith(".txt") Then
234             myStream = openFileDialog1.OpenFile()
235             If Not (myStream Is Nothing) Then

```

```

234         TxtSaveData.Text = openFileDialog1.FileName
235         myStream.Close()
236         Me.BeginInvoke(New InvokeDelagateCheck(AddressOf LoadFile),
    TxtSaveData.Text)
237     End If
238     Else
239         MsgBox(Me.MyClassCode.GetStr(Me.WhichIsTheLang, 294), MsgBoxStyle.
Information, Me.MyClassCode.GetStr(Me.WhichIsTheLang, 295))
240     End If
241     Catch exp As Exception
242         MsgBox(exp.Message, MsgBoxStyle.Information, MyClassCode.GetStr
(WhichIsTheLang, 37))
243     End Try
244 End Sub
245
246 Public Sub FixToolTips()
247     Me.MyToolTip.IsBalloon = True
248     BtPreviousValues.Cursor = Cursors.Hand
249     BtNextValues.Cursor = Cursors.Hand
250     BtFirstValues.Cursor = Cursors.Hand
251     BtLastValues.Cursor = Cursors.Hand
252     MyToolTip.SetToolTip(BtPreviousValues, MyClassCode.GetStr(WhichIsTheLang,
216))
253     MyToolTip.SetToolTip(BtNextValues, MyClassCode.GetStr(WhichIsTheLang, 217))
254     MyToolTip.SetToolTip(BtFirstValues, MyClassCode.GetStr(WhichIsTheLang, 218))
255     MyToolTip.SetToolTip(BtLastValues, MyClassCode.GetStr(WhichIsTheLang, 219))
256 End Sub
257
258 Public Sub SelectGBToShow(ByVal WhichGB As Integer)
259     ' 1 -> manual
260     ' 2 -> from function
261     ' 3 -> import from file
262     ResetGBs()
263     Select Case WhichGB
264     Case 1
265         GbManual.BringToFront()
266         GbManual.Visible = True
267         PlTimePages.BringToFront()
268         PlNavigator.BringToFront()
269         PlGoTo.BringToFront()
270     Case 2
271         GBFromFunction.BringToFront()
272         GBFromFunction.Visible = True
273     Case 3
274         If ImportFile = "" Then
275             BtImportFile.Enabled = False
276         End If
277         GBFromFile.BringToFront()
278         GBFromFile.Visible = True
279     End Select
280 End Sub
281
282 Public Sub ResetGBs()
283     GbManual.SendToBack()
284     GbManual.Visible = False
285     GbManual.Left = 12
286     GbManual.Top = 126
287     GBFromFunction.SendToBack()
288     GBFromFunction.Visible = False
289     GBFromFunction.Left = 12
290     GBFromFunction.Top = 126
291     GBFromFile.SendToBack()
292     GBFromFile.Visible = False
293     GBFromFile.Left = 12
294     GBFromFile.Top = 126
295 End Sub
296
297 Public Sub SetInitialFormAttributes()
298     Me.Width = 737
299     Me.Height = 713
300     Me.Left = (My.Computer.Screen.WorkingArea.Size.Width - Me.Width) / 2
301     Me.Top = (My.Computer.Screen.WorkingArea.Size.Height - Me.Height) / 2
302
303     Me.FormBorderStyle = Windows.Forms.FormBorderStyle.FixedSingle
304     Me.MaximizeBox = False

```

```

305         Me.StartPosition = FormStartPosition.CenterScreen
306
307         MyObjectMatrix(0) = MTxtValue1
308         MyObjectMatrix(1) = MTxtValue2
309         MyObjectMatrix(2) = MTxtValue3
310         MyObjectMatrix(3) = MTxtValue4
311         MyObjectMatrix(4) = MTxtValue5
312         MyObjectMatrix(5) = MTxtValue6
313         MyObjectMatrix(6) = MTxtValue7
314         MyObjectMatrix(7) = MTxtValue8
315         MyObjectMatrix(8) = MTxtValue9
316         MyObjectMatrix(9) = MTxtValue10
317         MyObjectMatrix(10) = MTxtValue11
318         MyObjectMatrix(11) = MTxtValue12
319         MyObjectMatrix(12) = MTxtValue13
320         MyObjectMatrix(13) = MTxtValue14
321         MyObjectMatrix(14) = MTxtValue15
322         MyObjectMatrix(15) = MTxtValue16
323         MyObjectMatrix(16) = MTxtValue17
324         MyObjectMatrix(17) = MTxtValue18
325         MyObjectMatrix(18) = MTxtValue19
326         MyObjectMatrix(19) = MTxtValue20
327         MyObjectMatrix(20) = MTxtValue21
328         MyObjectMatrix(21) = MTxtValue22
329         MyObjectMatrix(22) = MTxtValue23
330         MyObjectMatrix(23) = MTxtValue24
331         MyObjectMatrix(24) = MTxtValue25
332         MyObjectMatrix(25) = MTxtValue26
333         MyObjectMatrix(26) = MTxtValue27
334         MyObjectMatrix(27) = MTxtValue28
335         MyObjectMatrix(28) = MTxtValue29
336         MyObjectMatrix(29) = MTxtValue30
337         MyObjectMatrix(30) = MTxtValue31
338         MyObjectMatrix(31) = MTxtValue32
339         MyObjectMatrix(32) = MTxtValue33
340         MyObjectMatrix(33) = MTxtValue34
341         MyObjectMatrix(34) = MTxtValue35
342         MyObjectMatrix(35) = MTxtValue36
343         MyObjectMatrix(36) = MTxtValue37
344         MyObjectMatrix(37) = MTxtValue38
345         MyObjectMatrix(38) = MTxtValue39
346         MyObjectMatrix(39) = MTxtValue40
347         MyObjectMatrix(40) = MTxtValue41
348         MyObjectMatrix(41) = MTxtValue42
349         MyObjectMatrix(42) = MTxtValue43
350         MyObjectMatrix(43) = MTxtValue44
351         MyObjectMatrix(44) = MTxtValue45
352         MyObjectMatrix(45) = MTxtValue46
353         MyObjectMatrix(46) = MTxtValue47
354         MyObjectMatrix(47) = MTxtValue48
355         MyObjectMatrix(48) = MTxtValue49
356         MyObjectMatrix(49) = MTxtValue50
357     End Sub
358
359     Private Sub RBStrofes_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RBStrofes.CheckedChanged
360         GbManual.Text = MyClassCode.GetStr(WhichIsTheLang, 204)
361         AdjustMask(0)
362     End Sub
363
364     Private Sub RBGkazi_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RBGkazi.CheckedChanged
365         GbManual.Text = MyClassCode.GetStr(WhichIsTheLang, 205)
366         AdjustMask(1)
367     End Sub
368
369     Private Sub RBManual_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RBManual.CheckedChanged
370         SelectGBToShow(1)
371
372     End Sub
373
374     Private Sub RBFromFunction_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles RBFromFunction.CheckedChanged
375         SelectGBToShow(2)

```

```

376     End Sub
377
378     Private Sub RBFFromFile_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RBFFromFile.CheckedChanged
379         SelectGBToShow(3)
380     End Sub
381
382     Private Sub BtSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtSave.Click
383         Dim ErrorExists, i As Integer, MyFunctionString As String, k As Integer
384
385         PopulateMatrixFromTable(Current50ada)
386
387         If TxtSaveData.Text = "" Or (Not MyClassCode.IsValidFile(TxtSaveData.Text)) Then
388             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 57), MsgBoxStyle.Information, MyClassCode.GetStr(WhichIsTheLang, 109))
389             Exit Sub
390         End If
391
392         If RBStrofes.Checked Then
393             ErrorExists = ValidateStrofesValues(MatrixValues, MatrixDimension)
394         Else
395             ErrorExists = ValidateGkaziValues(MatrixValues, MatrixDimension)
396         End If
397
398         If ErrorExists <> -1 Then
399             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 223) & Str(ErrorExists), MsgBoxStyle.Information, MyClassCode.GetStr(WhichIsTheLang, 109))
400             Exit Sub
401         End If
402
403         'All ok, lets write to the file
404         Try
405             Dim CheckFile As New IO.FileInfo(Trim(TxtSaveData.Text))
406
407             ' If file already exists then it shall be deleted and remade
408             If CheckFile.Exists Then
409                 CheckFile.Delete()
410                 Dim fs As IO.FileStream = CheckFile.Create()
411                 fs.Close()
412             End If
413
414             Dim sw As IO.StreamWriter
415             sw = CheckFile.AppendText()
416
417             MyFunctionString = ""
418             If Not (ValidatedFunctions Is Nothing) Then
419                 MyFunctionString &= "#"
420                 For k = 0 To ValidatedFunctions.Length - 1
421                     MyFunctionString += "@$" & ValidatedFunctions(k).MyFunction & "$" & ValidatedFunctions(k).MyStartTime & "$" & ValidatedFunctions(k).MyStopTime
422                 Next
423                 MyFunctionString &= "#"
424             End If
425
426             MyClassCode.SetPointFileHeader(sw, RBStrofes.Checked, IIf(RBManual.Checked, "0", "1"), IIf(RBFFromFunction.Checked, "1", "2")), IIf(RBFFromFunction.Checked, MyFunctionString, "")
427             DisableForProgressBar(True, "FromFunction")
428
429             Dim ProgressBar_ChangeVal As Integer
430             For i = 0 To (Synolo50adwn * 50 - 1)
431                 If RBStrofes.Checked Then
432                     sw.WriteLine(Trim(Str(MatrixValues(i))))
433                 Else
434                     sw.WriteLine(Trim(Str(MatrixValues(i) / ControlActionDivider)))
435                 End If
436                 Me.MyProgressBar.Value = Int(100 * (i / (Synolo50adwn * 50 - 1)))
437
438                 If (ProgressBar_ChangeVal <> MyProgressBar.Value Or ProgressBar_ChangeVal = 0) Then
439                     ProgressBar_ChangeVal = MyProgressBar.Value
440                     Me.LblRemainingTime.Text = MyClassCode.GetStr(WhichIsTheLang, 310)

```



```

442     & " " & MyClassCode.EstimateRemainingTime(Now, MyProgressBar.Value)
443         LblRemainingTime.Refresh()
444     End If
445 Next
446
447     DisableForProgressBar(False, "FromFunction")
448     sw.Close()
449     Catch ex As Exception
450         If Me.MyProgressBar.Visible = True Then : DisableForProgressBar(False,
451 "FromFunction") : End If
452     MsgBox(MyClassCode.GetStr(WhichIsTheLang, 224) & Trim(TxtSaveData.Text),
453 MsgBoxStyle.Information, MyClassCode.GetStr(WhichIsTheLang, 109))
454 End Try
455 End Sub
456
457 Function ValidateStrofesValues(ByVal MyArray() As Integer, ByVal MyDimension As
458 Integer) As Integer
459     'Returns -1 if all ok else the number of the first problematic array item
460     Dim i As Integer
461     For i = 0 To MyDimension - 1
462         If (MyArray(i) < MinStrofes) Or (MyArray(i) > MaxStrofes) Then
463             Return i
464         End If
465     Next
466     Return -1
467 End Function
468
469 Function ValidateGkaziValues(ByVal MyArray() As Integer, ByVal MyDimension As
470 Integer) As Integer
471     ' Attention please: Values for gkazi are multiplied by 100 and then inserted
472 in the array
473     'Returns -1 if all ok else the number of the first problematic array item
474     Dim i As Integer
475     For i = 0 To MyDimension - 1
476         If (MyArray(i) < MinControlAction) Or (MyArray(i) > MaxControlAction) Or
477 (MyArray(i) * ControlActionDivider <> Int(MyArray(i) * ControlActionDivider)) Then
478             Return (i + 1)
479         End If
480     Next
481     Return -1
482 End Function
483
484 Public Sub PopulateMatrixFromTable(ByVal MyPage As Integer)
485     ' Used to update dynamic matrix with with proper values of table (page)
486     Dim i As Integer
487     If MatrixValues.Length = 0 Or MatrixValues Is Nothing Then
488         ReDim MatrixValues(49)
489     End If
490     MyPage -= 1
491     For i = 0 To 49
492         MatrixValues(MyPage * 50 + i) = IIf(RBStrofes.Checked, Val(MyObjectMatrix
493 (i).Text), Val(MyObjectMatrix(i).Text) * ControlActionDivider)
494     Next
495 End Sub
496
497 Public Sub PopulateTableFromMatrix(ByVal MyPage As Integer)
498     ' Used to update table (page) with proper values for dynamic matrix
499     Dim i As Integer, temp As String
500
501     MyPage -= 1
502     BaptiseLabels(MyPage)
503     For i = 0 To 49
504         temp = IIf(RBStrofes.Checked, Trim(Str(MatrixValues(MyPage * 50 + i))).
505 PadLeft(5, "0"c), CustomiseOutput(MatrixValues(MyPage * 50 + i)))
506         MyObjectMatrix(i).Text = temp
507     Next
508 End Sub
509
510 Private Sub BtPreviousValues_Click(ByVal sender As System.Object, ByVal e As
511 System.EventArgs) Handles BtPreviousValues.Click
512     PopulateMatrixFromTable(Current50ada)

```

```

507         Current50ada -= 1
508         CheckPageNavigatorsEnabledStatus(1)
509         PopulateTableFromMatrix(Current50ada)
510     End Sub
511
512     Private Sub BtFirstValues_Click(ByVal sender As System.Object, ByVal e As System.
513     EventArgs) Handles BtFirstValues.Click
514         PopulateMatrixFromTable(Current50ada)
515         Current50ada = 1
516         CheckPageNavigatorsEnabledStatus(0)
517         PopulateTableFromMatrix(1)
518     End Sub
519
520     Private Sub BtLastValues_Click(ByVal sender As System.Object, ByVal e As System.
521     EventArgs) Handles BtLastValues.Click
522         PopulateMatrixFromTable(Current50ada)
523         Current50ada = Synolo50adwn
524         CheckPageNavigatorsEnabledStatus(2)
525         PopulateTableFromMatrix(Current50ada)
526     End Sub
527
528     Private Sub BtNextValues_Click(ByVal sender As System.Object, ByVal e As System.
529     EventArgs) Handles BtNextValues.Click
530         If (Current50ada * 50 * TimeChanges + 1) > MyClassCode.MaximumTime Then
531             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 308), MsgBoxStyle.Information,
532             MyClassCode.GetStr(WhichIsTheLang, 109))
533         Else
534             Current50ada += 1
535
536             PopulateMatrixFromTable(Current50ada - 1)
537             CheckPageNavigatorsEnabledStatus(3)
538             If Synolo50adwn >= Current50ada Then
539                 PopulateTableFromMatrix(Current50ada)
540             Else
541                 Synolo50adwn += 1
542                 MatrixDimension += 50
543                 ReDim Preserve MatrixValues(Synolo50adwn * 50 - 1)
544                 BaptiseLabels(Current50ada)
545             End If
546         End If
547     End Sub
548
549     Public Sub AdjustMask(ByVal WhichMode As Integer)
550         ' WhichMode -> 0 for strofes
551         ' WhichMode -> 1 for gkazi
552         Dim i As Integer
553
554         Select Case WhichMode
555             Case 0
556                 For i = 0 To 49
557                     MyObjectMatrix(i).Mask = "00000"
558                 Next
559             Case 1
560                 For i = 0 To 49
561                     MyObjectMatrix(i).Mask = "00" & SystemDecimalSeperator & "00"
562                 Next
563         End Select
564
565         ResetValues(True)
566     End Sub
567
568     Private Function FindDecimalSeperator() As String
569         'Due to Regional Settings problem with Decimal Seperator we have these...
570         Dim Temp As String
571         TxtMyHidden.Text = 3 / 2
572         Temp = TxtMyHidden.Text
573
574         Return Temp.Substring(1, 1)
575     End Function
576
577     Public Sub CheckPageNavigatorsEnabledStatus(ByVal Caller As Integer)
578         ' BtFirstValues -> 0
579         ' BtPreviousValues -> 1
580         ' BtLastValues -> 2
581         ' BtNextValues -> 3

```

```

578     ' Initialisation    -> 4
579     ' Go Button        -> 5
580     ' Repeat           -> 6
581     ' Import File      -> 7
582
583     ' Must be called after current page index has been updated
584     Select Case Caller
585         Case 0
586             BtFirstValues.Enabled = False
587             BtPreviousValues.Enabled = False
588             BtLastValues.Enabled = True
589             BtNextValues.Enabled = True
590         Case 1
591             BtLastValues.Enabled = True
592             BtNextValues.Enabled = True
593             If Current50ada = 1 Then
594                 BtFirstValues.Enabled = False
595                 BtPreviousValues.Enabled = False
596             Else
597                 BtFirstValues.Enabled = True
598                 BtPreviousValues.Enabled = True
599             End If
600         Case 2
601             BtFirstValues.Enabled = True
602             BtPreviousValues.Enabled = True
603             BtLastValues.Enabled = False
604             BtNextValues.Enabled = True
605         Case 3
606             BtFirstValues.Enabled = True
607             BtPreviousValues.Enabled = True
608             BtLastValues.Enabled = True
609             If Synolo50adwn <= Current50ada Then
610                 BtNextValues.Enabled = True
611                 BtLastValues.Enabled = False
612             End If
613         Case 4
614             BtPreviousValues.Enabled = False
615             BtFirstValues.Enabled = False
616             If (Synolo50adwn > 1) Then
617                 BtLastValues.Enabled = True
618             Else
619                 BtLastValues.Enabled = False
620             End If
621         Case Is >= 5
622             If Current50ada = 1 Then
623                 BtPreviousValues.Enabled = False
624                 BtFirstValues.Enabled = False
625             Else
626                 BtPreviousValues.Enabled = True
627                 BtFirstValues.Enabled = True
628             End If
629
630             If Current50ada >= Synolo50adwn Then
631                 BtLastValues.Enabled = False
632             Else
633                 BtLastValues.Enabled = True
634             End If
635     End Select
636     NUDStartRepeatFromValue.Maximum = Synolo50adwn * 50 + 1
637     NUDFirstValue.Maximum = Synolo50adwn * 50
638     NUDLastValue.Maximum = Synolo50adwn * 50
639     NUDGoToPage.Maximum = Synolo50adwn
640     UpdateTimePageLiterals(Current50ada, Synolo50adwn)
641     If Caller = 1 Or Caller = 2 Or Caller = 3 Then
642         ResetValues(False)
643     End If
644 End Sub
645
646 Public Sub UpdateTimePageLiterals(ByVal CurrentPage As Integer, ByVal TotalPages As Integer)
647     Me.LblTimePeriod.Text = MyClassCode.GetStr(WhichIsTheLang, 225) & Trim(Str(
648     (CurrentPage - 1) * 50 * TimeChanges)) & " ms - " & Trim(Str(CurrentPage * 50 *
649     TimeChanges - TimeChanges)) & " ms"
650     Me.LblPageTadeFromTade.Text = MyClassCode.GetStr(WhichIsTheLang, 227) & Trim
651     (Str(CurrentPage)) & MyClassCode.GetStr(WhichIsTheLang, 228) & Trim(Str(IIf

```

```

(CurrentPage > TotalPages, TotalPages + 1, TotalPages)))
649 End Sub
650
651 Public Sub ValidateEachValue(ByVal sender As Object)
652     Select Case Me.RBStrofes.Checked
653         Case True
654             If Val(sender.text) > MaxStrofes Or Val(sender.text) < MinStrofes Then
655                 sender.BackColor = Color.Red
656                 MsgBox(MyClassCode.GetStr(WhichIsTheLang, 229) & " [" & Str
        (MinStrofes) & " - " & Str(MaxStrofes) & "]", MsgBoxStyle.Information, MyClassCode
        .GetStr(WhichIsTheLang, 109))
657                 sender.focus()
658             Else
659                 sender.BackColor = Color.White
660             End If
661         Case False
662             If Val(sender.text) > 100 Or Val(sender.text) < 0 Then
663                 sender.BackColor = Color.Red
664                 MsgBox(MyClassCode.GetStr(WhichIsTheLang, 229) & " [0.0 - 100.0]",
        MsgBoxStyle.Information, MyClassCode.GetStr(WhichIsTheLang, 109))
665                 sender.focus()
666             Else
667                 sender.BackColor = Color.White
668             End If
669     End Select
670 End Sub
671
672 Public Sub ResetValues(ByVal TotalReset As Boolean)
673     Dim i As Integer
674
675     For i = 0 To 49
676         MyObjectMatrix(i).Text = ""
677     Next
678
679     If TotalReset Then
680         If MatrixDimension > 0 Then
681             For i = 0 To (MatrixDimension - 1)
682                 MatrixValues(i) = 0
683             Next
684         End If
685
686         Current50ada = 1
687         Synolo50adwn = 1
688         MatrixDimension = 0
689         CheckPageNavigatorsEnabledStatus(4)
690     End If
691 End Sub
692
693 Public Function CustomiseOutput(ByVal MyValue As Integer) As String
694     If (MyValue / ControlActionDivider) = Int(MyValue / ControlActionDivider) Then
695         Return (Trim(Str(MyValue / ControlActionDivider)) + ".00").PadLeft(5, "0"
        "c)
696     ElseIf (MyValue / 10) = Int(MyValue / 10) Then
697         Return (Trim(Str(MyValue / ControlActionDivider)) + "0").PadLeft(5, "0"c)
698     Else
699         Return Trim(Str(MyValue / ControlActionDivider)).PadLeft(5, "0"c)
700     End If
701 End Function
702 Private Sub MTxtValue1_Validated(ByVal sender As Object, ByVal e As System.
        EventArgs) Handles MTxtValue1.Validated
703     ValidateEachValue(sender)
704 End Sub
705 Private Sub MTxtValue2_Validated(ByVal sender As Object, ByVal e As System.
        EventArgs) Handles MTxtValue2.Validated
706     ValidateEachValue(sender)
707 End Sub
708 Private Sub MTxtValue3_Validated(ByVal sender As Object, ByVal e As System.
        EventArgs) Handles MTxtValue3.Validated
709     ValidateEachValue(sender)
710 End Sub
711 Private Sub MTxtValue4_Validated(ByVal sender As Object, ByVal e As System.
        EventArgs) Handles MTxtValue4.Validated
712     ValidateEachValue(sender)
713 End Sub
714 Private Sub MTxtValue5_Validated(ByVal sender As Object, ByVal e As System.

```

```

EventArgs) Handles MTxtValue5.Validated
715     ValidateEachValue(sender)
716 End Sub
717 Private Sub MTxtValue6_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue6.Validated
718     ValidateEachValue(sender)
719 End Sub
720 Private Sub MTxtValue7_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue7.Validated
721     ValidateEachValue(sender)
722 End Sub
723 Private Sub MTxtValue8_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue8.Validated
724     ValidateEachValue(sender)
725 End Sub
726 Private Sub MTxtValue9_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue9.Validated
727     ValidateEachValue(sender)
728 End Sub
729 Private Sub MTxtValue10_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue10.Validated
730     ValidateEachValue(sender)
731 End Sub
732 Private Sub MTxtValue11_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue11.Validated
733     ValidateEachValue(sender)
734 End Sub
735 Private Sub MTxtValue12_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue12.Validated
736     ValidateEachValue(sender)
737 End Sub
738 Private Sub MTxtValue13_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue13.Validated
739     ValidateEachValue(sender)
740 End Sub
741 Private Sub MTxtValue14_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue14.Validated
742     ValidateEachValue(sender)
743 End Sub
744 Private Sub MTxtValue15_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue15.Validated
745     ValidateEachValue(sender)
746 End Sub
747 Private Sub MTxtValue16_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue16.Validated
748     ValidateEachValue(sender)
749 End Sub
750 Private Sub MTxtValue17_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue17.Validated
751     ValidateEachValue(sender)
752 End Sub
753 Private Sub MTxtValue18_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue18.Validated
754     ValidateEachValue(sender)
755 End Sub
756 Private Sub MTxtValue19_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue19.Validated
757     ValidateEachValue(sender)
758 End Sub
759 Private Sub MTxtValue20_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue20.Validated
760     ValidateEachValue(sender)
761 End Sub
762 Private Sub MTxtValue21_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue21.Validated
763     ValidateEachValue(sender)
764 End Sub
765 Private Sub MTxtValue22_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue22.Validated
766     ValidateEachValue(sender)
767 End Sub
768 Private Sub MTxtValue23_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue23.Validated
769     ValidateEachValue(sender)
770 End Sub

```

```

771 Private Sub MTxtValue24_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue24.Validated
772     ValidateEachValue(sender)
773 End Sub
774 Private Sub MTxtValue25_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue25.Validated
775     ValidateEachValue(sender)
776 End Sub
777 Private Sub MTxtValue26_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue26.Validated
778     ValidateEachValue(sender)
779 End Sub
780 Private Sub MTxtValue27_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue27.Validated
781     ValidateEachValue(sender)
782 End Sub
783 Private Sub MTxtValue28_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue28.Validated
784     ValidateEachValue(sender)
785 End Sub
786 Private Sub MTxtValue29_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue29.Validated
787     ValidateEachValue(sender)
788 End Sub
789 Private Sub MTxtValue30_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue30.Validated
790     ValidateEachValue(sender)
791 End Sub
792 Private Sub MTxtValue31_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue31.Validated
793     ValidateEachValue(sender)
794 End Sub
795 Private Sub MTxtValue32_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue32.Validated
796     ValidateEachValue(sender)
797 End Sub
798 Private Sub MTxtValue33_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue33.Validated
799     ValidateEachValue(sender)
800 End Sub
801 Private Sub MTxtValue34_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue34.Validated
802     ValidateEachValue(sender)
803 End Sub
804 Private Sub MTxtValue35_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue35.Validated
805     ValidateEachValue(sender)
806 End Sub
807 Private Sub MTxtValue36_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue36.Validated
808     ValidateEachValue(sender)
809 End Sub
810 Private Sub MTxtValue37_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue37.Validated
811     ValidateEachValue(sender)
812 End Sub
813 Private Sub MTxtValue38_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue38.Validated
814     ValidateEachValue(sender)
815 End Sub
816 Private Sub MTxtValue39_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue39.Validated
817     ValidateEachValue(sender)
818 End Sub
819 Private Sub MTxtValue40_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue40.Validated
820     ValidateEachValue(sender)
821 End Sub
822 Private Sub MTxtValue41_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue41.Validated
823     ValidateEachValue(sender)
824 End Sub
825 Private Sub MTxtValue42_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue42.Validated
826     ValidateEachValue(sender)

```

```

827 End Sub
828 Private Sub MTxtValue43_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue43.Validated
829     ValidateEachValue(sender)
830 End Sub
831 Private Sub MTxtValue44_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue44.Validated
832     ValidateEachValue(sender)
833 End Sub
834 Private Sub MTxtValue45_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue45.Validated
835     ValidateEachValue(sender)
836 End Sub
837 Private Sub MTxtValue46_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue46.Validated
838     ValidateEachValue(sender)
839 End Sub
840 Private Sub MTxtValue47_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue47.Validated
841     ValidateEachValue(sender)
842 End Sub
843 Private Sub MTxtValue48_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue48.Validated
844     ValidateEachValue(sender)
845 End Sub
846 Private Sub MTxtValue49_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue49.Validated
847     ValidateEachValue(sender)
848 End Sub
849 Private Sub MTxtValue50_Validated(ByVal sender As Object, ByVal e As System.
EventArgs) Handles MTxtValue50.Validated
850     ValidateEachValue(sender)
851 End Sub
852
853 Private Sub BtGo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles BtGo.Click
854     PopulateMatrixFromTable(Current50ada)
855     Current50ada = NUDGoToPage.Value
856     CheckPageNavigatorsEnabledStatus(5)
857     PopulateTableFromMatrix(Current50ada)
858 End Sub
859
860 Private Sub BtExecuteRepeat_Click(ByVal sender As System.Object, ByVal e As System
.EventArgs) Handles BtExecuteRepeat.Click
861     Try
862
863         Dim i, j, ElemStart, ElemStop, ElemBeginFrom, RepeatTimes As Integer
864
865         CheckPeriods(NUDFirstValue, NUDStartRepeatFromValue.Value, NUDFirstValue.
Value, NUDLastValue.Value, False)
866         If NUDFirstValue.BackColor = Color.Red Or NUDLastValue.BackColor = Color.
Red Or NUDStartRepeatFromValue.BackColor = Color.Red Then
867             Exit Sub
868         End If
869
870         ' Must check if maximum time will be reached
871         If ((NUDLastValue.Value - NUDFirstValue.Value + 1) * NUDRepeatTimes.Value
* TimeChanges + (NUDFirstValue.Value - 1) * TimeChanges) > MyClassCode.MaximumTime
Then
872             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 309), MsgBoxStyle.
Information, MyClassCode.GetStr(WhichIsTheLang, 109))
873             Exit Sub
874         End If
875
876         PopulateMatrixFromTable(Current50ada) ' In case the user has made a change
877         ElemStart = NUDFirstValue.Value
878         ElemStop = NUDLastValue.Value
879         ElemBeginFrom = NUDStartRepeatFromValue.Value
880         RepeatTimes = NUDRepeatTimes.Value
881         CheckAndFixMatrixDimension(ElemBeginFrom, ElemStart, ElemStop,
RepeatTimes)
882         For i = 0 To RepeatTimes - 1
883             For j = 0 To ElemStop - ElemStart
884                 MatrixValues((ElemStop - ElemStart + 1) * i + ElemBeginFrom - 1 +
j) = MatrixValues(ElemStart - 1 + j)

```

```

885         Next
886     Next
887     CheckPageNavigatorsEnabledStatus(6)
888     PopulateTableFromMatrix(Current50ada)
889
890     Catch
891         Stop
892     End Try
893 End Sub
894
895 Public Sub CheckAndFixMatrixDimension(ByVal StartPoint As Integer, ByVal FirstElem
As Integer, ByVal LastElem As Integer, ByVal TotalTimes As Integer)
896     ' Only for repeat action
897     Dim AdditionalElementsRequired, PartialElements, TotalElements As Integer
898
899     PartialElements = LastElem - FirstElem + 1 ' For each repeat PartialElements are
needed
900     TotalElements = PartialElements * TotalTimes ' For full repeat TotalElements are
needed
901     AdditionalElementsRequired = (MatrixDimension - StartPoint + 1) - TotalElements
902     If AdditionalElementsRequired > 0 Then
903         ' No need to do anything, we have space
904     Else
905         AdditionalElementsRequired = Math.Abs(AdditionalElementsRequired)
906         MatrixDimension += AdditionalElementsRequired
907         Synolo50adwn = Int(MatrixDimension / 50) + IIf(MatrixDimension / 50 = Int
(MatrixDimension / 50), 0, 1)
908         ReDim Preserve MatrixValues(Synolo50adwn * 50 - 1)
909     End If
910 End Sub
911
912 Public Function CheckPeriods(ByVal sender As Object, ByVal StartPoint As Integer,
ByVal FirstElem As Integer, ByVal LastElem As Integer, ByVal AllowZeros As
Boolean) As Integer
913     ' Used to check if values entered in 3 Numeric Up Downs are ok
914
915     If AllowZeros And FirstElem = 0 And LastElem = 0 And StartPoint = 0 Then
916         ' Don't check anything
917         NUDFirstValue.BackColor = Color.White
918         NUDLastValue.BackColor = Color.White
919         NUDStartRepeatFromValue.BackColor = Color.White
920     Else
921         If FirstElem > LastElem Or (StartPoint >= FirstElem And StartPoint <=
LastElem) Or FirstElem = 0 Or LastElem = 0 Then
922             NUDFirstValue.BackColor = Color.Red
923             NUDLastValue.BackColor = Color.Red
924             NUDStartRepeatFromValue.BackColor = Color.Red
925             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 231), MsgBoxStyle.
Information, MyClassCode.GetStr(WhichIsTheLang, 109))
926             'sender.Focus()
927         Else
928             NUDFirstValue.BackColor = Color.White
929             NUDLastValue.BackColor = Color.White
930             NUDStartRepeatFromValue.BackColor = Color.White
931         End If
932     End If
933 End Function
934
935 Private Sub NUDFirstValue_Validated(ByVal sender As Object, ByVal e As System.
EventArgs)
936
937 End Sub
938
939 Private Sub NUDLastValue_Validated(ByVal sender As Object, ByVal e As System.
EventArgs)
940
941 End Sub
942
943 Private Sub NUDStartRepeatFromValue_Validated(ByVal sender As Object, ByVal e As
System.EventArgs) Handles NUDStartRepeatFromValue.Validated
944
945 End Sub
946
947 Public Function IsImportedFileOK(ByVal MyFileName As String, ByVal worker As
BackgroundWorker, ByVal e As DoWorkEventArgs) As ReturnFilevalues

```



```

948     Dim ReturnResult As ReturnFilevalues
949     If MyClassCode.IsValidFile(MyFileName) Then
950         Dim CheckFile As New IO.FileInfo(MyFileName)
951         Dim MaxLength As Integer = CheckFile.Length
952         ReDim ImportFromFileMatrix(MaxLength)
953         Dim met As Integer = 0
954         Dim MaxValue As Int64 = CheckFile.Length
955         Dim Position As Int64
956         ReturnResult.NoError = False
957         ReturnResult.ValueDispStrg = ""
958
959         Try
960             If (Not CheckFile.Exists) Then
961                 MsgBox(MyClassCode.GetStr(WhichIsTheLang, 44), MsgBoxStyle.
Information, MyClassCode.GetStr(WhichIsTheLang, 109))
962                 Return ReturnResult
963             End If
964
965             Dim sr As IO.StreamReader = New IO.StreamReader(MyFileName)
966             Dim line As String
967
968             line = sr.ReadLine()
969             If line Is Nothing Then
970                 MsgBox(MyClassCode.GetStr(WhichIsTheLang, 107) & MyFileName & " "
& MyClassCode.GetStr(WhichIsTheLang, 108), MsgBoxStyle.Information, MyClassCode.
GetStr(WhichIsTheLang, 109))
971                 Return ReturnResult
972             End If
973
974             Do
975                 Application.DoEvents()
976                 If IsLineBad(line) Then
977                     MsgBox(MyClassCode.GetStr(WhichIsTheLang, 107) & MyFileName &
" " & MyClassCode.GetStr(WhichIsTheLang, 108), MsgBoxStyle.Information,
MyClassCode.GetStr(WhichIsTheLang, 109))
978                     Return ReturnResult
979                 End If
980                 met += 1
981                 ImportFromFileMatrix(met - 1) = IIf(RBStrofes.Checked, Val(line),
(Val(line) * ControlActionDivider))
982                 Position += line.Length + 2
983                 line = sr.ReadLine()
984                 If Me.LoadFileBackgroundWorker.CancellationPending = True Then
985                     e.Cancel = True
986                     Return ReturnResult
987                 End If
988                 Me.LoadFileBackgroundWorker.ReportProgress(Position * 100 /
MaxValue)
989             Loop Until line Is Nothing
990             ImportedFileLines = met
991             sr.Close()
992             ReDim Preserve ImportFromFileMatrix(met - 1)
993             If ImportedFileLines * TimeChanges > MyClassCode.MaximumTime Then
994                 ReDim MatrixValues(-1)
995                 MsgBox(Me.MyClassCode.GetStr(Me.WhichIsTheLang, 237), MsgBoxStyle.
Information, Me.MyClassCode.GetStr(Me.WhichIsTheLang, 109))
996                 e.Cancel = True
997                 ReturnResult.NoError = False
998                 ReturnResult.ValueDispStrg = ""
999                 Return ReturnResult
1000             End If
1001             Catch Exp As Exception
1002                 MsgBox(MyClassCode.GetStr(WhichIsTheLang, 237), MsgBoxStyle.
Information, MyClassCode.GetStr(WhichIsTheLang, 109))
1003                 Return ReturnResult
1004             End Try
1005
1006             ReturnResult.ValueDispStrg = ShowDataInTextBox(ImportFromFileMatrix,
ImportedFileLines)
1007             ReturnResult.NoError = True
1008             If Not Me.LoadFileBackgroundWorker.CancellationPending Then
1009                 Return ReturnResult
1010             Else
1011                 e.Cancel = True
1012                 ReturnResult.NoError = False

```

```

1013         ReturnResult.ValueDispStrg = ""
1014         Return ReturnResult
1015     End If
1016 Else
1017     e.Cancel = True
1018     ReturnResult.NoError = False
1019     ReturnResult.ValueDispStrg = ""
1020     Return ReturnResult
1021 End If
1022 End Function
1023
1024 Public Function IsLineBad(ByVal line As String) As Boolean
1025     Dim i As Integer
1026
1027     For i = 0 To (line.Length - 1)
1028         If line(i) = "0" Or line(i) = "1" Or line(i) = "2" Or line(i) = "3" Or
line(i) = "4" Or line(i) = "5" Or line(i) = "6" Or line(i) = "7" Or line(i) = "8"
Or line(i) = "9" Or line(i) = "." Or line(i) = "," Then
1029             If line(i) = "." Or line(i) = "," Then
1030                 If i = 0 Then
1031                     Return True
1032                 End If
1033
1034                 If i = line.Length - 1 Then
1035                     Return True
1036                 End If
1037
1038                 If (line(i - 1) = "." Or line(i - 1) = ",") Or (line(i + 1) = "."
Or line(i + 1) = ",") Then
1039                     Return True
1040                 End If
1041             End If
1042             ' Do nothing yet
1043         Else
1044             Return True
1045         End If
1046     Next
1047
1048     'Up to now no problem, time to check values
1049     If RBStrofes.Checked Then
1050         If Val(line) < MinStrofes Or Val(line) > MaxStrofes Or (Val(line) <> Int
(Val(line))) Then
1051             Return True
1052         End If
1053     Else
1054         If Val(line) < 0 Or Val(line) > 100 Then
1055             Return True
1056         End If
1057     End If
1058     ' Passed all checks, time for thanks
1059     Return False
1060 End Function
1061
1062 Public Function ShowDataInTextBox(ByVal MyMatrix() As Integer, ByVal MyDimension
As Integer) As String
1063     Dim i As Integer
1064     Dim Dashes As New String("-"c, 211)
1065     Dim Spaces As New String(" "c, 90)
1066     Dim MyString As String = ""
1067     Dim MyStringBuilder As New System.Text.StringBuilder(MyString)
1068
1069     MyStringBuilder.Append(Dashes & vbCrLf)
1070     MyStringBuilder.Append(Spaces & MyClassCode.GetStr(WhichIsTheLang, 238) & " "
& Trim(Str(MyDimension)) & vbCrLf)
1071     MyStringBuilder.Append(Dashes & vbCrLf)
1072
1073     Dim Count16 As Integer = 0
1074     For i = 0 To (MyDimension - 1)
1075         Count16 += 1
1076         MyStringBuilder.Append(" | " & IIf(RBStrofes.Checked, Trim(Str(MyMatrix
(i))).PadLeft(5, "0"c), CustomiseOutput(MyMatrix(i))))
1077         If Count16 >= 16 Then
1078             Count16 = 0
1079             MyStringBuilder.Append(" | " & vbCrLf)
1080         End If

```

```

1081     Next
1082     MyStringBuilder.Append("|")
1083     Return MyStringBuilder.ToString
1084 End Function
1085
1086 Private Sub BtCheckFile_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles BtCheckFile.Click
1087     If TxtLoadFile.Text = "" Or (Not MyClassCode.IsValidFile(TxtLoadFile.Text))
Then
1088         MsgBox(MyClassCode.GetStr(WhichIsTheLang, 57), MsgBoxStyle.Information,
MyClassCode.GetStr(WhichIsTheLang, 109))
1089         Exit Sub
1090     End If
1091
1092     Try
1093         If Not (LoadFileBackgroundWorker.CancellationPending Or
LoadFileBackgroundWorker.IsBusy) Then
1094             DisableForProgressBar(True, "FromImportFile")
1095             Me.LoadFileBackgroundWorker.RunWorkerAsync(TxtLoadFile.Text)
1096             Me.ProgressBarImportFromFileData.Visible = True
1097         End If
1098     Catch
1099         ' Do nothing for now
1100     End Try
1101 End Sub
1102
1103 Private Sub TxtLoadFile_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TxtLoadFile.TextChanged
1104     BtImportFile.Enabled = False
1105 End Sub
1106
1107 Private Sub BtLoadFile_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles BtLoadFile.Click
1108     Try
1109         Dim myStream As IO.Stream
1110         Dim openFileDialog1 As New OpenFileDialog
1111
1112         openFileDialog1.InitialDirectory = "c:\Windows\Desktop"
1113         openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*"
1114         openFileDialog1.FilterIndex = 1
1115         openFileDialog1.RestoreDirectory = True
1116         openFileDialog1.Multiselect = False
1117
1118         If openFileDialog1.ShowDialog() = System.Windows.Forms.DialogResult.OK And
openFileDialog1.FileName.EndsWith(".txt") Then
1119             myStream = openFileDialog1.OpenFile()
1120             If Not (myStream Is Nothing) Then
1121                 TxtLoadFile.Text = openFileDialog1.FileName
1122                 myStream.Close()
1123             End If
1124         Else
1125             MsgBox(Me.MyClassCode.GetStr(Me.WhichIsTheLang, 294), MsgBoxStyle.
Information, Me.MyClassCode.GetStr(Me.WhichIsTheLang, 295))
1126         End If
1127     Catch exp As Exception
1128         MsgBox(exp.Message, MsgBoxStyle.Information, MyClassCode.GetStr
(WhichIsTheLang, 37))
1129     End Try
1130 End Sub
1131
1132 Private Sub TxtFileData_GotFocus(ByVal sender As Object, ByVal e As System.
EventArgs) Handles TxtFileData.GotFocus
1133     BtCheckFile.Focus()
1134 End Sub
1135
1136 Private Sub BtImportFile_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles BtImportFile.Click
1137     Dim i As Integer
1138     Me.ProgressBarImportFromFileData.Visible = True
1139     If ImportFile = TxtLoadFile.Text And ImportFile <> "" Then
1140         DisableForProgressBar(True, "FromImportFile")
1141         Current50ada = 1
1142         Synolo50adwn = Int(ImportedFileLines / 50) + IIf(ImportedFileLines / 50 =
Int(ImportedFileLines / 50), 0, 1)
1143         MatrixDimension = ImportedFileLines

```

```

1144         CheckPageNavigatorsEnabledStatus("7")
1145         ReDim MatrixValues(Synolo50adwn * 50 - 1)
1146         For i = 0 To (ImportedFileLines - 1)
1147             MatrixValues(i) = ImportFromFileMatrix(i)
1148             Me.ProgressBarImportFromFileData.Value = Int(i * 100 /
ImportFromFileMatrix.Length)
1149         Next
1150         PopulateTableFromMatrix(Current50ada)
1151         BtImportFile.Enabled = False
1152         DisableForProgressBar(False, "FromImportFile")
1153     Else
1154         MsgBox(MyClassCode.GetStr(WhichIsTheLang, 240), MsgBoxStyle.Information,
MyClassCode.GetStr(WhichIsTheLang, 109))
1155         BtImportFile.Enabled = False
1156     End If
1157     Me.ProgressBarImportFromFileData.Visible = False
1158 End Sub
1159
1160 Public Sub FunctionFromPBs(ByVal WhichOne As Integer)
1161     Dim MyString As String = ""
1162
1163     Select Case WhichOne
1164     Case 1
1165         MyString = "sin(t)"
1166     Case 2
1167         MyString = "cos(t)"
1168     Case 5
1169         MyString = "abs(t)"
1170     Case 6
1171         MyString = "tan(t)"
1172     Case 7
1173         MyString = "atan(t)"
1174     Case 8
1175         MyString = "sqr(t)"
1176     Case 9
1177         MyString = "log(t)"
1178     Case 10
1179         MyString = "exp(t)"
1180     End Select
1181
1182     If TxtSelectedFunction.Text <> "" Then
1183         TxtSelectedFunction.Text &= " + " & MyString
1184     Else
1185         TxtSelectedFunction.Text = MyString
1186     End If
1187 End Sub
1188
1189 Private Sub PBFunction1_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles PBFunction1.Click
1190     FunctionFromPBs(1)
1191 End Sub
1192 Private Sub PBFunction2_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles PBFunction2.Click
1193     FunctionFromPBs(2)
1194 End Sub
1195 Private Sub PBFunction3_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs)
1196     FunctionFromPBs(3)
1197 End Sub
1198 Private Sub PBFunction4_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs)
1199     FunctionFromPBs(4)
1200 End Sub
1201 Private Sub PBFunction5_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles PBFunction5.Click
1202     FunctionFromPBs(5)
1203 End Sub
1204 Private Sub PBFunction6_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles PBFunction6.Click
1205     FunctionFromPBs(6)
1206 End Sub
1207 Private Sub PBFunction7_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles PBFunction7.Click
1208     FunctionFromPBs(7)
1209 End Sub

```

```

1210 Private Sub PBFunction8_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles PBFunction8.Click
1211     FunctionFromPBs(8)
1212 End Sub
1213 Private Sub PBFunction9_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles PBFunction9.Click
1214     FunctionFromPBs(9)
1215 End Sub
1216 Private Sub PBFunction10_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles PBFunction10.Click
1217     FunctionFromPBs(10)
1218 End Sub
1219
1220 Private Function WasEvaluateFunctionOK() As Boolean
1221     Dim i As Integer
1222
1223     Try
1224         Dim calculator As New Evaluator(WhichIsTheLang, TxtSelectedFunction.Text.
Replace("t", "P1"), "P", ParseCondition.RaiseErrorForUndefinedVariable)
1225
1226         Dim parameters As New Hashtable()
1227         parameters.Add(1, 2)
1228         Dim matvalue() As Double
1229
1230         'Will run once to see if it will boom or not
1231         i = 1 ' case you have subfunction 1/t
1232
1233         ReDim Preserve matvalue(i + 1)
1234
1235         parameters.Item(1) = i
1236
1237         Dim value As Double = calculator.Evaluate(parameters)
1238
1239         TxtEvaluatedFunction.Text = TxtSelectedFunction.Text
1240         TxtEvaluatedFunction.Refresh()
1241         matvalue(i) = value
1242
1243         Return True
1244     Catch e As Exception
1245         ' No Reason to show a message here, it is shown from evaluate
1246         Return False
1247     End Try
1248 End Function
1249
1250 Private Sub BtCheckFunction_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles BtCheckFunction.Click
1251     If TxtSelectedFunction.Text = "" Then
1252         MsgBox(MyClassCode.GetStr(WhichIsTheLang, 257), MsgBoxStyle.Information,
MyClassCode.GetStr(WhichIsTheLang, 109))
1253     Else
1254         CheckedFunction = TxtSelectedFunction.Text
1255         If WasEvaluateFunctionOK() Then
1256             TxtEvaluatedFunction.Text = TxtSelectedFunction.Text
1257             EvaluationOK = True
1258         Else
1259             TCFUNCTIONS.SelectTab(2)
1260             TxtSelectedFunction.Focus()
1261             EvaluationOK = False
1262         End If
1263         CheckFunctionButtonsEnabledStatus()
1264     End If
1265 End Sub
1266
1267 Public Sub DeclareAcceptableSymbols()
1268     Dim Dashes As New String("-"c, 87)
1269     Dim Spaces As New String(" "c, 10)
1270
1271     TxtAcceptableSymbols.Text = ""
1272     TxtAcceptableSymbols.Text &= "|" & Spaces & MyClassCode.GetStr(WhichIsTheLang,
254) & Spaces & "|" & Spaces & MyClassCode.GetStr(WhichIsTheLang, 255) & Spaces &
1273     "|" & Spaces & MyClassCode.GetStr(WhichIsTheLang, 256) & Spaces & "|" & vbCrLf
1274     TxtAcceptableSymbols.Text &= "|" & Dashes & "|" & vbCrLf
1275
1276     TxtAcceptableSymbols.Text &= "|" & Spaces & " abs() " & Spaces & "|" &

```

```

1277     Spaces & " + " & Spaces & "|" & Spaces & " t " & Spaces & "|" & vbCrLf
        TxtAcceptableSymbols.Text &= "|" & Spaces & " atn() " & Spaces & "|" &
1278     Spaces & " - " & Spaces & "|" & Spaces & " " & Spaces & "|" & vbCrLf
        TxtAcceptableSymbols.Text &= "|" & Spaces & " cos() " & Spaces & "|" &
1279     Spaces & " * " & Spaces & "|" & Spaces & " " & Spaces & "|" & vbCrLf
        TxtAcceptableSymbols.Text &= "|" & Spaces & " exp() " & Spaces & "|" &
1280     Spaces & " / " & Spaces & "|" & Spaces & " " & Spaces & "|" & vbCrLf
        TxtAcceptableSymbols.Text &= "|" & Spaces & " log() " & Spaces & "|" &
1281     Spaces & " ^ " & Spaces & "|" & Spaces & " " & Spaces & "|" & vbCrLf
        TxtAcceptableSymbols.Text &= "|" & Spaces & " sin() " & Spaces & "|" &
1282     Spaces & " ( " & Spaces & "|" & Spaces & " " & Spaces & "|" & vbCrLf
        TxtAcceptableSymbols.Text &= "|" & Spaces & " sqr() " & Spaces & "|" &
1283     Spaces & " ) " & Spaces & "|" & Spaces & " " & Spaces & "|" & vbCrLf
        TxtAcceptableSymbols.Text &= "|" & Spaces & " tan() " & Spaces & "|" &
1284     Spaces & " ) " & Spaces & "|" & Spaces & " " & Spaces & "|" & vbCrLf
1285     End Sub
1286     Private Sub TxtAcceptableSymbols_Click(ByVal sender As Object, ByVal e As System.
        EventArgs) Handles TxtAcceptableSymbols.Click
1287         TxtSelectedFunction.Focus()
1288     End Sub
1289
1290     Private Sub TxtSelectedFunctions_Click(ByVal sender As Object, ByVal e As System.
        EventArgs) Handles TxtSelectedFunctions.Click
1291         TxtSelectedFunction.Focus()
1292     End Sub
1293
1294     Private Sub BtRegisterFunction_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles BtRegisterFunction.Click
1295         Try
1296
1297             Dim i As Integer, StartTime As Integer, StopTime As Integer,
        ValidatedStart As Integer, ValidatedStop As Integer
1298
1299             If CheckedFunction <> TxtSelectedFunction.Text Then
1300                 MsgBox(MyClassCode.GetStr(WhichIsTheLang, 250), MsgBoxStyle.
        Information, MyClassCode.GetStr(WhichIsTheLang, 109))
1301                 CheckFunctionButtonsEnabledStatus()
1302                 Exit Sub
1303             End If
1304
1305             'Check if time given is not integer
1306             If Val(TxtInitialTime.Text) > MyClassCode.MaximumTime Or Val(TxtFinalTime.
        Text) > MyClassCode.MaximumTime Then
1307                 MsgBox(MyClassCode.GetStr(WhichIsTheLang, 269) & Str(MyClassCode.
        MaximumTime), MsgBoxStyle.Information, MyClassCode.GetStr(WhichIsTheLang, 109))
1308                 Exit Sub
1309             End If
1310
1311             StartTime = Val(TxtInitialTime.Text)
1312             StopTime = Val(TxtFinalTime.Text)
1313
1314             If TxtInitialTime.Text = "" Or TxtFinalTime.Text = "" Or StartTime >=
        StopTime Then
1315                 MsgBox(MyClassCode.GetStr(WhichIsTheLang, 248), MsgBoxStyle.
        Information, MyClassCode.GetStr(WhichIsTheLang, 109))
1316                 CheckFunctionButtonsEnabledStatus()
1317                 Exit Sub
1318             End If
1319
1320             ' Check if time periods were already given
1321             If Not (ValidatedFunctions Is Nothing) Then
1322                 For i = 0 To (ValidatedFunctions.Length - 1)
1323                     ValidatedStart = ValidatedFunctions(i).MyStartTime
1324                     ValidatedStop = ValidatedFunctions(i).MyStopTime
1325
1326                     If (StartTime >= ValidatedStart And StartTime <= ValidatedStop) Or
        -
1327                     (StopTime >= ValidatedStart And StopTime <= ValidatedStop) Or
        -
1328                     (StartTime < ValidatedStart And StopTime > ValidatedStop) Then
1329
1330                     TCFunctions.SelectTab(1)
1331                     TxtSelectedFunction.Focus()
1332                     MsgBox(MyClassCode.GetStr(WhichIsTheLang, 249), MsgBoxStyle.
        Information, MyClassCode.GetStr(WhichIsTheLang, 109))

```

```

1333             CheckFunctionButtonsEnabledStatus()
1334             Exit Sub
1335         End If
1336     Next
1337 End If
1338
1339 ' If it gets here values are valid
1340 If ValidatedFunctions Is Nothing Then
1341     ReDim ValidatedFunctions(0)
1342 Else
1343     ReDim Preserve ValidatedFunctions(ValidatedFunctions.Length)
1344 End If
1345 ValidatedFunctions(ValidatedFunctions.Length - 1).MyFunction = Trim
(TxtEvaluatedFunction.Text)
1346 ValidatedFunctions(ValidatedFunctions.Length - 1).MyStartTime = StartTime
1347 ValidatedFunctions(ValidatedFunctions.Length - 1).MyStopTime = StopTime
1348 OrderValidatedFunctionsArray(ValidatedFunctions, ValidatedFunctions.Length
- 1)
1349 CheckFunctionButtonsEnabledStatus()
1350 ShowValidatedFunctions()
1351 Catch
1352     Stop
1353 End Try
1354 End Sub
1355
1356 Public Sub ResetFunctionValues()
1357     TxtInitialTime.Text = ""
1358     TxtFinalTime.Text = ""
1359     TxtSelectedFunction.Text = ""
1360     TxtEvaluatedFunction.Text = ""
1361     TxtSelectedFunctions.Text = ""
1362     ValidatedFunctions = Nothing
1363     ValidatedFunctionsIndex = 0
1364     BtRegisterFunction.Enabled = False
1365     BtRunFunction.Enabled = False
1366 End Sub
1367
1368 Public Sub CheckFunctionButtonsEnabledStatus()
1369     If EvaluationOK Then
1370         BtRegisterFunction.Enabled = True
1371     Else
1372         BtRegisterFunction.Enabled = False
1373     End If
1374
1375     If ValidatedFunctions Is Nothing Then
1376         BtRunFunction.Enabled = False
1377     Else
1378         BtRunFunction.Enabled = True
1379     End If
1380 End Sub
1381
1382 Private Sub BtReset_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles BtReset.Click
1383     ResetFunctionValues()
1384 End Sub
1385
1386 Private Sub RunEvaluatedFunctions(ByVal sender As System.Object, ByVal e As System
.ComponentModel.DoWorkEventArgs)
1387     Dim i As Integer, j As Integer, Results() As Integer, MyMet As Integer = 0
1388     Dim TempStart, MyMod As Integer, MyValidValidatedFunctions() As
MyMatrixLooksLike
1389
1390     Try
1391         ReDim MyValidValidatedFunctions(ValidatedFunctions.Length - 1)
1392         MyValidValidatedFunctions = ValidatedFunctions
1393
1394         CheckAndCorrectTimeConsistency(MyValidValidatedFunctions)
1395         Dim maxmymet As Integer = GetReadyForProgressBar
(MyValidValidatedFunctions)
1396
1397         'Will run once to see if it will boom or not
1398         ReDim Results((MyValidValidatedFunctions(MyValidValidatedFunctions.Length
- 1).MyStopTime / TimeChanges) + 1)
1399         For i = 0 To (MyValidValidatedFunctions.Length - 1)
1400             Dim calculator As New Evaluator(WhichIsTheLang,

```

```

MyValidValidatedFunctions(i).MyFunction.Replace("t", "P1"), "P", ParseCondition.
RaiseErroForUndefinedVariable)
1401         Dim parameters As New Hashtable()
1402         parameters.Add(1, 2)
1403         MyMod = MyValidValidatedFunctions(i).MyStartTime Mod TimeChanges
1404         TempStart = IIf(MyMod = 0, MyValidValidatedFunctions(i).MyStartTime,
MyValidValidatedFunctions(i).MyStartTime - MyMod + TimeChanges)
1405
1406         For j = TempStart To MyValidValidatedFunctions(i).MyStopTime Step
TimeChanges
1407             MyMet += 1
1408             parameters.Item(1) = j
1409             Dim value As Double = calculator.Evaluate(parameters)
1410             Results(MyMet - 1) = IIf(RBStrofes.Checked, Int(value), Int(value
* ControlActionDivider))
1411             If RBStrofes.Checked Then
1412                 If Results(MyMet - 1) < MinStrofes Or Results(MyMet - 1) >
MaxStrofes Then
1413                     MsgBox(MyClassCode.GetStr(WhichIsTheLang, 258),
MsgBoxStyle.Information, MyClassCode.GetStr(WhichIsTheLang, 109))
1414                     Exit Sub
1415                 End If
1416             Else
1417                 If Results(MyMet - 1) < MinControlAction Or Results(MyMet - 1)
> MaxControlAction Then
1418                     MsgBox(MyClassCode.GetStr(WhichIsTheLang, 258),
MsgBoxStyle.Information, MyClassCode.GetStr(WhichIsTheLang, 109))
1419                     Exit Sub
1420                 End If
1421             End If
1422
1423             Me.MyBackgroundWorker.ReportProgress(IIf(Int(MyMet * 100 /
maxmymet) <= 100, Int(MyMet * 100 / maxmymet), 100))
1424             ' End If
1425             If MyBackgroundWorker.CancellationPending = True Then
1426                 e.Cancel = True
1427                 Exit Sub
1428             End If
1429         Next
1430     Next
1431
1432     Current50ada = 1
1433     Synolo50adwn = Int(MyMet / 50) + IIf(MyMet / 50 = Int(ImportedFileLines /
50), 0, 1)
1434     MatrixDimension = MyMet
1435     ReDim MatrixValues(Synolo50adwn * 50 - 1)
1436     For i = 0 To (MyMet - 1)
1437         MatrixValues(i) = Results(i)
1438     Next
1439     Catch exp As Exception
1440         MsgBox(MyClassCode.GetStr(WhichIsTheLang, 251), MsgBoxStyle.Information,
MyClassCode.GetStr(WhichIsTheLang, 109))
1441     End Try
1442 End Sub
1443
1444 Public Sub SortMyArray(ByRef MyMatrix(,) As String)
1445
1446 End Sub
1447
1448 Public Sub ShowValidatedFunctions()
1449     Dim i As Integer
1450     Dim Dashes As New String("-"c, 87)
1451     Dim Spaces As New String(" "c, 15)
1452
1453     TxtSelectedFunctions.Text = "" & vbCrLf
1454     For i = 0 To (ValidatedFunctions.Length - 1)
1455         TxtSelectedFunctions.Text &= MyClassCode.GetStr(WhichIsTheLang, 252) & " :
" & Str(ValidatedFunctions(i).MyStartTime) & "ms - " & _
1456             Str(ValidatedFunctions(i).MyStopTime) & "ms " & _
& MyClassCode.GetStr(WhichIsTheLang, 253) & " f(t) = " & _
1457             Trim(ValidatedFunctions(i).MyFunction) &
vbCrLf
1458     Next
1459 End Sub
1460

```



```

1461 Private Sub BtRunFunction_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles BtRunFunction.Click
1462     Try
1463         If Not Me.MyBackgroundWorker.IsBusy() Then
1464             MyProgressBar.Value = 0
1465             MyProgressBar.Visible = True
1466             DisableForProgressBar(True, "FromFunction")
1467             Me.MyBackgroundWorker.RunWorkerAsync()
1468         End If
1469     Catch
1470         Stop
1471         ' Do nothing
1472     End Try
1473 End Sub
1474
1475 Private Sub BtShowValuesInGraph_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtShowValuesInGraph.Click
1476     Dim fShowInGraph As New ShowInGraph
1477     Dim temp(MatrixValues.Length - 1) As Int32
1478     fShowInGraph.WhichLang(WhichIsTheLang)
1479     If MatrixValues.Length < 50 Then
1480         Dim MatrixValuesImage(49) As Integer
1481         fShowInGraph.SetDefaults(MatrixValuesImage, IIf(RBStrofes.Checked, 0, 1))
1482     Else
1483         fShowInGraph.SetDefaults(MatrixValues, IIf(RBStrofes.Checked, 0, 1))
1484     End If
1485     fShowInGraph.ShowDialog()
1486     MatrixValues = fShowInGraph.FileValues
1487     fShowInGraph.Close()
1488     fShowInGraph = Nothing
1489     PopulateTableFromMatrix(Current50ada)
1490 End Sub
1491
1492 Public Sub OrderValidatedFunctionsArray(ByRef ValidatedFunctions() As
MyMatrixLooksLike, ByVal ValidatedFunctionsIndex As Integer)
1493     Dim i, j As Integer, temp As MyMatrixLooksLike
1494
1495     ' Order Array Data according to start time
1496     For i = 0 To ValidatedFunctionsIndex - 1
1497         If i > 0 Then
1498             For j = i To 1 Step -1
1499                 If ValidatedFunctions(j).MyStartTime < ValidatedFunctions(j - 1).
MyStartTime Then
1500                     temp.MyFunction = ValidatedFunctions(j).MyFunction
1501                     temp.MyStartTime = ValidatedFunctions(j).MyStartTime
1502                     temp.MyStopTime = ValidatedFunctions(j).MyStopTime
1503
1504                     ValidatedFunctions(j).MyFunction = ValidatedFunctions(j - 1).
MyFunction
1505                     ValidatedFunctions(j).MyStartTime = ValidatedFunctions(j - 1).
MyStartTime
1506                     ValidatedFunctions(j).MyStopTime = ValidatedFunctions(j - 1).
MyStopTime
1507
1508                     ValidatedFunctions(j - 1).MyFunction = temp.MyFunction
1509                     ValidatedFunctions(j - 1).MyStartTime = temp.MyStartTime
1510                     ValidatedFunctions(j - 1).MyStopTime = temp.MyStopTime
1511                 End If
1512             Next
1513         End If
1514     Next
1515 End Sub
1516
1517 Public Sub CheckAndCorrectTimeConsistency(ByRef MyValidValidatedFunctions() As
MyMatrixLooksLike)
1518     ' This sub fills not mentioned time with zeros since time is increasing
starting with 0
1519     Dim i As Integer
1520
1521     For i = 0 To MyValidValidatedFunctions.Length - 1
1522         If i = 0 Then
1523             ReDim MyValidValidatedFunctions(0)
1524             If ValidatedFunctions(i).MyStartTime = 0 Then
1525                 MyValidValidatedFunctions(i) = ValidatedFunctions(i)
1526             Else

```

```

1527         MyValidValidatedFunctions(i).MyFunction = "0"
1528         MyValidValidatedFunctions(i).MyStartTime = 0
1529         MyValidValidatedFunctions(i).MyStopTime = ValidatedFunctions(i).
MyStartTime - 1 'TimeChanges
1530
1531         ReDim Preserve MyValidValidatedFunctions(1)
1532         MyValidValidatedFunctions(i + 1) = ValidatedFunctions(i)
1533     End If
1534 Else
1535     ReDim Preserve MyValidValidatedFunctions(MyValidValidatedFunctions.
Length)
1536     If ValidatedFunctions(i).MyStartTime = (ValidatedFunctions(i - 1).
MyStopTime + 1) Then
1537         MyValidValidatedFunctions(MyValidValidatedFunctions.Length - 1) =
ValidatedFunctions(i)
1538     Else
1539         MyValidValidatedFunctions(MyValidValidatedFunctions.Length - 1).
MyFunction = "0"
1540         MyValidValidatedFunctions(MyValidValidatedFunctions.Length - 1).
MyStartTime = ValidatedFunctions(i - 1).MyStopTime + 1
1541         MyValidValidatedFunctions(MyValidValidatedFunctions.Length - 1).
MyStopTime = ValidatedFunctions(i).MyStartTime - 1
1542
1543         ReDim Preserve MyValidValidatedFunctions(MyValidValidatedFunctions
.Length)
1544         MyValidValidatedFunctions(MyValidValidatedFunctions.Length - 1) =
ValidatedFunctions(i)
1545     End If
1546 End If
1547 Next
1548 End Sub
1549
1550 Public Function GetReadyForProgressBar(ByVal MyValidValidatedFunctions() As
MyMatrixLooksLike) As Integer
1551     Dim StartTime, StopTime As Integer
1552
1553     StartTime = MyValidValidatedFunctions(0).MyStartTime
1554     StopTime = MyValidValidatedFunctions(MyValidValidatedFunctions.Length - 1).
MyStopTime
1555
1556     Return Int((StopTime - StartTime) / TimeChanges)
1557 End Function
1558
1559 Private Sub TxtEvaluatedFunction_Click(ByVal sender As Object, ByVal e As System.
EventArgs) Handles TxtEvaluatedFunction.Click
1560     TxtSelectedFunction.Focus()
1561 End Sub
1562
1563 Private Sub TxtInitialTime_Click(ByVal sender As Object, ByVal e As System.
EventArgs) Handles TxtInitialTime.Click
1564     If TxtInitialTime.Text <> "" Then
1565         TxtInitialTime.SelectAll()
1566     End If
1567 End Sub
1568
1569 Private Sub TxtInitialTime_Enter(ByVal sender As Object, ByVal e As System.
EventArgs) Handles TxtInitialTime.Enter
1570     If TxtInitialTime.Text <> "" Then
1571         TxtInitialTime.SelectAll()
1572     End If
1573 End Sub
1574
1575 Private Sub TxtFinalTime_Click(ByVal sender As Object, ByVal e As System.
EventArgs) Handles TxtFinalTime.Click
1576     If TxtFinalTime.Text <> "" Then
1577         TxtFinalTime.SelectAll()
1578     End If
1579 End Sub
1580
1581 Private Sub TxtFinalTime_Enter(ByVal sender As Object, ByVal e As System.
EventArgs) Handles TxtFinalTime.Enter
1582     If TxtFinalTime.Text <> "" Then
1583         TxtFinalTime.SelectAll()
1584     End If
1585 End Sub

```

```

1586
1587 Private Sub TxtSelectedFunction_Click(ByVal sender As Object, ByVal e As System.
EventArgs) Handles TxtSelectedFunction.Click
1588     If TxtSelectedFunction.Text <> "" Then
1589         TxtSelectedFunction.SelectAll()
1590     End If
1591 End Sub
1592 Private Sub TxtSelectedFunction_Enter(ByVal sender As Object, ByVal e As System.
EventArgs) Handles TxtSelectedFunction.Enter
1593     If TxtSelectedFunction.Text <> "" Then
1594         TxtSelectedFunction.SelectAll()
1595     End If
1596 End Sub
1597
1598 Private Sub TxtEvaluatedFunction_Enter(ByVal sender As Object, ByVal e As System.
EventArgs) Handles TxtEvaluatedFunction.Enter
1599     TxtSelectedFunction.Focus()
1600 End Sub
1601
1602 Public Sub RunEvaluation(ByVal worker As BackgroundWorker, ByVal e As
DoWorkEventArgs)
1603     Try
1604         If Not e.Cancel Then
1605             RunEvaluatedFunctions(worker, e)
1606         End If
1607     Catch
1608         e.Cancel = True
1609     End Try
1610 End Sub
1611
1612 Private Sub MyBackgroundWorker_DoWork(ByVal sender As System.Object, ByVal e As
System.ComponentModel.DoWorkEventArgs) Handles MyBackgroundWorker.DoWork
1613     Dim EvalWorker As BackgroundWorker = CType(sender, BackgroundWorker)
1614     RunEvaluation(EvalWorker, e)
1615 End Sub
1616
1617 Private Sub MyBackgroundWorker_ProgressChanged(ByVal sender As System.Object,
ByVal e As System.ComponentModel.ProgressChangedEventArgs) Handles
MyBackgroundWorker.ProgressChanged
1618     Me.MyProgressBar.Value = e.ProgressPercentage
1619     If e.ProgressPercentage <= 1 Or (e.ProgressPercentage Mod 1) = 0 Then
1620         If e.ProgressPercentage <> PreviousPercentage Or e.ProgressPercentage = 0
Then
1621             PreviousPercentage = e.ProgressPercentage
1622             LblRemainingTime.Text = MyClassCode.GetStr(WhichIsTheLang, 310) & " "
& MyClassCode.EstimateRemainingTime(Now, e.ProgressPercentage)
1623         End If
1624     End If
1625 End Sub
1626
1627 Private Sub MyBackgroundWorker_RunWorkerCompleted(ByVal sender As System.Object,
ByVal e As System.ComponentModel.RunWorkerCompletedEventArgs) Handles
MyBackgroundWorker.RunWorkerCompleted
1628     Try
1629         If (e.Error IsNot Nothing) Then
1630             MyProgressBar.Visible = False
1631             MessageBox.Show(e.Error.Message)
1632         ElseIf e.Cancelled Or sender.CancellationPending = True Then
1633             MyProgressBar.Visible = False
1634         Else
1635             CheckPageNavigatorsEnabledStatus("7")
1636             PopulateTableFromMatrix(Current50ada)
1637             MyProgressBar.Visible = False
1638         End If
1639         BtRunFunction.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 247)
1640     Catch
1641         ' Do nothing for now
1642     End Try
1643
1644     DisableForProgressBar(False, "FromFunction")
1645 End Sub
1646
1647 Private Sub LoadFileBackgroundWorker_ProgressChanged(ByVal sender As System.Object
, ByVal e As System.ComponentModel.ProgressChangedEventArgs) Handles
LoadFileBackgroundWorker.ProgressChanged

```

```

1648     Me.ProgressBarImportFromFileData.Value = e.ProgressPercentage
1649     If e.ProgressPercentage <= 1 Or (e.ProgressPercentage Mod 1) = 0 Then
1650         If e.ProgressPercentage > PreviousPercentage Or e.ProgressPercentage = 0
1651     Then
1652         PreviousPercentage = e.ProgressPercentage
1653         LblRemainingTime.Text = MyClassCode.GetStr(WhichIsTheLang, 310) & " "
1654     & MyClassCode.EstimateRemainingTime(Now, e.ProgressPercentage)
1655     End If
1656 End If
1657 End Sub

1658 Private Sub LoadFileBackgroundWorker_RunWorkerCompleted(ByVal sender As System.
1659 Object, ByVal e As System.ComponentModel.RunWorkerCompletedEventArgs) Handles
1660 LoadFileBackgroundWorker.RunWorkerCompleted
1661     Try
1662         If (e.Error IsNot Nothing) Then
1663             BtImportFile.Enabled = False
1664             ImportFile = ""
1665         ElseIf e.Cancelled Or sender.CancellationPending = True Then
1666             BtImportFile.Enabled = False
1667             ImportFile = ""
1668         Else
1669             Dim temp As ReturnFilevalues = e.Result
1670             If temp.NoError = True Then
1671                 BtImportFile.Enabled = True
1672                 ImportFile = TxtLoadFile.Text
1673             Else
1674                 BtImportFile.Enabled = False
1675                 TxtLoadFile.Text = ""
1676                 ImportFile = ""
1677             End If
1678             TxtFileData.Text = temp.ValueDispStrg
1679             Me.ProgressBarImportFromFileData.Value = 0
1680         End If
1681         DisableForProgressBar(False, "FromImportFile")
1682     Catch
1683         MsgBox("xa")
1684         'Do nothing for now
1685     End Try
1686 End Sub

1687 Public Sub Cancelloloadfile()
1688     Me.LoadFileBackgroundWorker.CancelAsync()
1689 End Sub

1690 Private Sub LoadFileBackgroundWorker_DoWork(ByVal sender As System.Object, ByVal e
1691 As System.ComponentModel.DoWorkEventArgs) Handles LoadFileBackgroundWorker.DoWork
1692     Dim workerfile As BackgroundWorker = CType(sender, BackgroundWorker)
1693     e.Result = IsImportedFileOK(e.Argument, workerfile, e)
1694 End Sub

1695 Private Sub CheckRadioButtons(ByVal name As String)
1696     Select Case name
1697     Case RBStrofes.Name
1698         RBStrofes.Checked = True
1699     Case RBGkazi.Name
1700         RBGkazi.Checked = True
1701     Case RBManual.Name
1702         RBManual.Checked = True
1703     Case RBFromFunction.Name
1704         RBFromFunction.Checked = True
1705     Case RBFromFile.Name
1706         RBFromFunction.Checked = True
1707     End Select
1708 End Sub

1709 Private Sub EnumAccording2Line5(ByVal linestext As String)
1710     If line4 = RBFromFunction.Name Then
1711         Dim MyString, Word, Words1(), Words2() As String
1712         Dim Seperators1() As Char = {"@"c}
1713         Dim Seperators2() As Char = {"$"c}
1714         Dim MyMet As Integer = -1
1715         Dim MyStart, MyStop As Integer
1716         MyStart = linestext.IndexOf("#")
1717
1718         If MyStart <> -1 Then
1719             MyStop = linestext.IndexOf("#", MyStart + 1)
1720             MyString = linestext.Substring(MyStart + 1, MyStop - MyStart - 1)

```

```

1718         Words1 = MyString.Split(Seperators1)
1719         For Each Word In Words1
1720             If MyMet <> -1 Then
1721                 Words2 = Word.Split(Seperators2)
1722                 ReDim Preserve ValidatedFunctions(MyMet)
1723                 ValidatedFunctions(MyMet).MyFunction = Words2(1)
1724                 ValidatedFunctions(MyMet).MyStartTime = Val(Words2(2))
1725                 ValidatedFunctions(MyMet).MyStopTime = Val(Words2(3))
1726             End If
1727             MyMet += 1
1728         Next
1729         ShowValidatedFunctions()
1730         TCFunctions.SelectTab(1)
1731     End If
1732 End If
1733 End Sub
1734 Private Sub LoadFile(ByVal File As String)
1735     Try
1736         Dim fLoadFile As New LoadingDataForm
1737         fLoadFile.WhichLang(Me.WhichIsTheLang)
1738         fLoadFile.SetDefaults(File)
1739         fLoadFile.ShowDialog()
1740
1741         If fLoadFile.MatrixValues.Length > 0 Then
1742             Line3 = fLoadFile.Line3
1743             line4 = fLoadFile.line4
1744             Line5 = fLoadFile.Line5
1745         Else
1746             Line3 = "RBStrofes"
1747             line4 = "RBManual"
1748             Line5 = ""
1749         End If
1750         MatrixValues = fLoadFile.MatrixValues
1751         MatrixDimension = MatrixValues.Length
1752         Me.TxtSaveData.Text = fLoadFile.ConfSendDataFile
1753         Me.ConfSendDataFile = fLoadFile.ConfSendDataFile
1754
1755         fLoadFile.Close()
1756
1757         Dim tepmmatrix(MatrixValues.Length - 1) As Integer
1758         If MatrixValues.Length Then : Array.Copy(Me.MatrixValues, tepmmatrix,
MatrixValues.Length) : End If
1759
1760         CheckRadioButtons(Line3)
1761         CheckRadioButtons(line4)
1762
1763         If MatrixValues.Length Then : Array.Copy(tepmmatrix, Me.MatrixValues,
MatrixValues.Length) : End If
1764
1765         EnumAccording2Line5(Line5)
1766         If MatrixValues.Length > 0 Then
1767             If MatrixValues.Length < 50 Then : ReDim Preserve MatrixValues(49)
1768             : Else : ReDim Preserve MatrixValues(IIf(MatrixValues.Length Mod 50 = 0
, MatrixValues.Length, Math.Ceiling(MatrixValues.Length / 50) * 50) - 1) : End If
1769
1770             MatrixDimension = MatrixValues.Length
1771             Synolo50adwn = Int(MatrixDimension / 50) + IIf(MatrixDimension / 50 =
Int(MatrixDimension / 50), 0, 1)
1772             Current50ada = 1
1773
1774             PopulateTableFromMatrix(Current50ada)
1775             CheckPageNavigatorsEnabledStatus(4)
1776         Else
1777             If RBStrofes.Checked = False And RBGkazi.Checked = False Then
1778                 ' If cancel button was pressed very quickly because of the timer
values could not be initialised
1779                 RBStrofes.Checked = True
1780                 RBManual.Checked = True
1781             End If
1782             'case new *.txt file or cancel
1783             ReDim MatrixValues(49)
1784             MatrixDimension = 50
1785             Synolo50adwn = Int(MatrixDimension / 50) + IIf(MatrixDimension / 50 =
Int(MatrixDimension / 50), 0, 1)
1786             Current50ada = 1

```

```

1787         PopulateTableFromMatrix(Current50ada)
1788         CheckPageNavigatorsEnabledStatus(4)
1789     End If
1790     BtRunFunction.Text = MyClassCode.GetStr(Me.WhichIsTheLang, 247)
1791     fLoadFile.Dispose()
1792 Catch
1793     Stop
1794     'fLoadFile.Dispose()
1795 End Try
1796 End Sub
1797
1798 Private Sub DelayedLoadfile(ByVal source As Object, ByVal e As ElapsedEventArgs)
1799     Me.BeginInvoke(New InvokeDelagateCheck(AddressOf LoadFile), ConfSendDataFile)
1800 End Sub
1801
1802 Private Sub BtCancelTask_Click(ByVal sender As System.Object, ByVal e As System.
1803     EventArgs) Handles BtCancelTask.Click
1804     Try
1805         If ProgressBarImportFromFileData.Visible = True And
1806             LoadFileBackgroundWorker.CancellationPending = False And LoadFileBackgroundWorker.
1807             IsBusy Then
1808             LoadFileBackgroundWorker.CancelAsync()
1809             DisableForProgressBar(False, "FromFunction")
1810         End If
1811
1812         If MyProgressBar.Visible = True And MyBackgroundWorker.CancellationPending
1813             = False Then
1814             Me.MyBackgroundWorker.CancelAsync()
1815             DisableForProgressBar(False, "FromImportFile")
1816         End If
1817     Catch
1818         ' Do nothing
1819     End Try
1820 End Sub
1821
1822 Public Sub DisableForProgressBar(ByVal Status As Boolean, ByVal WhoIsTheCaller As
1823     String)
1824     GrBxDataSave.Enabled = Not Status
1825     GrBSelectMode.Enabled = Not Status
1826     GrBCreateFileBy.Enabled = Not Status
1827     GBRepeat.Visible = Not Status
1828     BtSave.Enabled = Not Status
1829     BtShowValuesInGraph.Enabled = Not Status
1830     BtClose.Enabled = Not Status
1831     GBTaskProgress.Visible = Status
1832     GBTaskProgress.Left = 12
1833     GBTaskProgress.Top = 532
1834     BtCancelTask.Visible = Status
1835
1836     Select Case WhoIsTheCaller
1837     Case "FromFunction"
1838         GBFromFunction.Enabled = Not Status
1839         MyProgressBar.Visible = Status
1840         ProgressBarImportFromFileData.Visible = False
1841     Case "FromImportFile"
1842         GBFromFile.Enabled = Not Status
1843         ProgressBarImportFromFileData.Visible = Status
1844         MyProgressBar.Visible = False
1845     End Select
1846
1847     GrBxDataSave.Refresh()
1848     GrBSelectMode.Refresh()
1849     GrBCreateFileBy.Refresh()
1850     GBRepeat.Refresh()
1851     BtSave.Refresh()
1852     BtShowValuesInGraph.Refresh()
1853     BtClose.Refresh()
1854     GBTaskProgress.Refresh()
1855     GBTaskProgress.Refresh()
1856     GBTaskProgress.Refresh()
1857     BtCancelTask.Refresh()
1858     GBFromFunction.Refresh()
1859     MyProgressBar.Refresh()
1860     ProgressBarImportFromFileData.Refresh()

```

```
1856         GBFromFile.Refresh()  
1857     End Sub  
1858  
1859  
1860 End Class
```

```

1
2 Option Strict Off
3 Imports Microsoft.Office.Interop.Excel
4 Imports System.Drawing
5 Imports System.Collections
6 Imports System.ComponentModel
7 Imports System.IO
8 Imports System.Windows.Forms
9 Imports System.Timers
10 Imports Microsoft
11 Imports System
12 Imports System.Threading
13 Imports Microsoft.VisualBasic
14 Imports Microsoft.VisualBasic.Devices
15
16 Public Class Thrust
17     Public WhichIsTheLang As Integer
18     Private MyClassCode As New MyPtyxClass
19     Private ProblemOccured As Boolean
20     Private FlagStartSend As Boolean
21     Public StartIndex As Integer, ExcelIsInstalled As Boolean
22     Private Result1, Result2, Result3, Result4, Result5, Result6, Result7, Result8 As Integer
23     Private Hours, SecsMet, Mins, Secs, mSecs As Integer
24     Public DecimalSeparator, StringToWrite, TheSelectedConfFile As String
25     Delegate Sub InvokeDelegate(ByVal i As Integer, ByVal text As String)
26     Delegate Sub InvokeDelagateGraphs()
27     Delegate Sub InvokeDelagateMode(ByVal bool As Boolean)
28     Delegate Sub InvokeDelegateWrite2File(ByVal String2Write As String)
29     Public Teststart() As Integer = {115, 116, 97, 114, 116}
30     Private MyTemp(5) As Char
31     Private MyTempMet As Integer
32     Private sw As IO.StreamWriter
33     Public FilehasBeenUpdated As Boolean
34     Public ZoomInCursor As New Cursor(CurDir() & "\ICONS\ZoomIn.cur")
35     Public ZoomOutCursor As New Cursor(CurDir() & "\ICONS\ZoomOut.cur")
36     Private DemicalDivider() As UInt16 = {1, 100, 10000, 1000, 100, 1}
37     Private Const ControlActionDivider As Integer = 100
38     Public Const MinStrofes As Integer = 0, MinControlAction As Integer = 0
39     Public Const MaxStrofes As Integer = 20000, MaxControlAction As Integer = 10000
40     Public PbShowAllGraphs, ProcessAlreadyClosed, GraphRunFirstTime As Boolean
41     Public ArraysMet, ArraysReadMet, NumOfExistingExcelProcs As Integer
42     Public CreatedExcelProcID, RunningExcelProcs(NumOfExistingExcelProcs) As Integer
43     Public app As Microsoft.Office.Interop.Excel.Application
44     Public oldCI As System.Globalization.CultureInfo = System.Threading.Thread.
        CurrentThread.CurrentCulture
45     Public Doc As Microsoft.Office.Interop.Excel.Workbook
46     Public sheet As Microsoft.Office.Interop.Excel.Worksheet
47     Public Chart1, Chart2 As Microsoft.Office.Interop.Excel.Chart
48     ' for chart graphs -----
49     Private NoTimeReset As UInt16
50     Public ChartCreationError As Boolean
51     Public image1, image2 As Image
52     Public series1, series2, series3 As Microsoft.Office.Interop.Excel.Series
53     ' ----- chart setting variables
54     Public Structure ChartSettings
55         Public showcharts1 As Boolean
56         Public showcharts2 As Boolean
57         Public Graphrpm As Boolean
58         Public Graphgkazi As Boolean
59         Public Graphloadcell As Boolean
60         Public Graphvoltage As Boolean
61         Public Graphampere As Boolean
62         Public Graphwrpm As Boolean
63         Public Graphwgkazi As Boolean
64         Public NoPointstodisplay As UInt16
65         Public ChartLocation As System.Drawing.Point
66         Public ChartSize As System.Drawing.Size
67     End Structure
68
69     Private Chart1Settings As New ChartSettings
70     Private Chart2Settings As New ChartSettings
71     Private LineLegend() As String = {"RPM", "PWMLoad", "LOADCELL", "Voltage", "Ampere",
        "Setpoint RPM", "Setpoint PWM"}
72     Private MaxValdivider() As Int32 = {1, 10, 10000, 1000, 100, 1, 10}

```



```

73 Private NoofData As UInt32 = 0
74 ' ----- which graphs to display varaiable declaration
75 Public Enum GraphicsDisplay
76     ShowGraphics
77     showChartGraphics
78     ShowNoGraphics
79 End Enum
80 Private Graphs As GraphicsDisplay
81 Public matforworker(100, 2) As Int64
82 Public workerlcompleted As Boolean = True
83 Public IDnumberofChart As Int32
84 '-----RPM METER -----
85 Private g As Graphics
86 Public RPM, pwm, cell, volt, amp As Int32
87 Private ShowPictureBox2 As Boolean
88 Private CenterOfRPMmeter As New System.Drawing.Point
89 Private DiameterRPM As Int16 = 60
90 Private DiameterMin As Int16 = 5
91 Dim pointmat(100, 1) As Integer ' is been using for saving the calculated
points of lines from data and be sended to draw.lines method to draw the lines
92 Private showRpm, showPwmLoad As Boolean
93 Private Legendsof2box() As String = {"---Pwmload", "---Loadcell", "---Volt", "---
Ampere"}
94 Private Unitsdisplayfor2box() As String = {"% ", "Kgf", "Volt", "Amp"}
95 Dim pento2box() As Pen = {Pens.Red, Pens.Black, Pens.Orange, Pens.Green}
96 Dim Brushsesto2box() As Brush = {Brushes.Red, Brushes.Black, Brushes.Orange,
Brushes.Green}
97 Private MaxValueOfLinesData2box() As Integer = {1000, 100000, 25000, 5000}
98 Private displayLineValue2box() As Integer = {100, 10, 25, 50}
99
100 Private Legendsof1box() As String = {"---RPM", "---SetpointRpm", "---PwmLoad", "---
SetpointPwmLoad"}
101 Private Unitsdisplayfor1box() As String = {"RPM ", "RPM", "%", "%"}
102 Dim pentolbox() As Pen = {Pens.Red, Pens.Black, Pens.Orange, Pens.Green}
103 Dim Brushsestolbox() As Brush = {Brushes.Red, Brushes.Black, Brushes.Orange,
Brushes.Green}
104 Private MaxValueOfLinesData1box() As Integer = {20000, 20000, 1000, 1000}
105 Private displayLineValue1box() As Integer = {20000, 20000, 100, 100}
106
107 Private Rmatrix(100000), WRmatrix(100000), Loadcell(100000), Ampere(100000),
Voltage(100000), gkazi(100000) As Integer
108 Private Tmatrix(100000) As UInt32
109 Private Wantedgkazi(100000) As Integer
110 Public Enum typeoffline
111     DushedLine '=0
112     SolidLine '=1
113 End Enum 'typeoffline
114
115 Public Enum Modeofapplication
116     ControledRPM = 0
117     ControlPWMDuty = 1
118     ControlInertia = 2
119 End Enum 'ModeofIAR
120
121 Public IARmode As Modeofapplication
122 Public VBstatus As Modeofapplication
123 Public StatusofIAR(2) As Byte
124 Public Errors(7) As Boolean
125 Public GetSetpointsFromFile As Boolean
126 '----- set points root of
application -----
127 Public Enum SetpointsRoot
128     FormControls = 0
129     File = 1
130 End Enum
131
132 Public IARSetPointRoot As SetpointsRoot
133 Public VBSetPointRoot As SetpointsRoot = SetpointsRoot.FormControls
134 '----- end point of set points root
-----
135 Public Enum MotorCondition
136     MotorRunning = 1
137     MotorStopped = 0
138 End Enum

```

```

139
140     Public IARMotor As MotorCondition = MotorCondition.MotorRunning
141     Public VBMotor As MotorCondition = MotorCondition.MotorRunning
142
143     Public Enum DoSendFrame
144         DoNotSendFrame = 0
145         SendFrame = 1
146     End Enum
147     Public Enum IARModeRequest
148         ControlRPM = 0
149         ControlAction = 1
150     End Enum
151
152     Public Structure GraphSettings
153         Public horizontalLines As Int16
154         Public Verticallines As Int16
155         Public HLinesDivisions As Int16
156         Public TypeofHline As typeofline
157         Public TypeofVline As typeofline
158         Public PenofHVLines As Pen
159         Public ShowHLines As Boolean
160         Public showVlines As Boolean
161         Public GraphLine1 As Boolean
162         Public GraphLine2 As Boolean
163         Public GraphLine3 As Boolean
164         Public GraphLine4 As Boolean
165         Public ShowLegengofLines As Boolean
166         Public Size As System.Drawing.Size
167         Public Location As System.Drawing.Point
168         Public NofPointsofLine As Int16
169         Public TimeIntervOfVertLine As UInt32
170         Public TypeofTVLines As typeofline
171         Public ShowTVLines As Boolean
172
173     End Structure
174     ' ----- end graphsettings -----
175
176     Private graph2settings As GraphSettings
177     Private graph1settings As GraphSettings
178
179     Public Structure barsettings
180         Public width As Integer
181         Public height As Integer
182         Public NofVal As Integer
183         Public brushes() As System.Drawing.Brush
184     End Structure
185     Public MaxDispValue() As Integer = {100, 10, 25, 50}
186     Public MaxValues() As Integer = {1000, 100000, 25000, 5000}
187     Public Barset As barsettings
188     ' ----- Led settings -----
189
190     Public Structure LedSettings
191         Public diameter As Integer
192         Public pen() As System.Drawing.Pen
193         Public Brush() As System.Drawing.Brush
194     End Structure
195     Public Ledset As LedSettings
196     ' ----- testing timer accuracy -----
197     Private Testtimer(1) As System.DateTime ' for system timer usage
198     Private BasicGraphicTimer As New System.Timers.Timer
199     Private AnimateEmergencyStopButtonlighting As New System.Timers.Timer
200
201     Public Structure FileReading
202         Public FileName As String
203         Public line As Integer
204         Public values() As Int32
205         Public Nooflinesread As Integer
206         Public FileisReaded As Boolean
207         Public ReadInitStatus As Boolean
208     End Structure
209     Public FilewithData As FileReading
210     Private sr As IO.StreamReader
211     Public Structure meanvlaues

```

```

211     Public rpm() As Long
212     Public pwm() As Long
213     Public cell() As Long
214     Public volt() As Long
215     Public amp() As Long
216
217 End Structure
218
219 Public MeanBasic As New meanvlaues
220 Private Savecomplete As Boolean
221 Private FileBufferValues() As UInt32
222 Private sendingindex As Integer
223 Private PreviousText_at_ConnStatusBar As String
224 Public Const PWM_Min As Integer = 0
225 Public Const PWM_Duty As Integer = 10000
226 Public RequestDataServed As Boolean = True
227 '-----end read from checked file -----
-----
-----
228
229 Public Sub New()
230     ' This call is required by the Windows Form Designer.
231     InitializeComponent()
232     AddHandler BasicGraphicTimer.Elapsed, AddressOf BasicTimerGraphsEvent '
'OnTimedEvent
233     Me.BasicGraphicTimer.AutoReset = True ' for system timer usage
234     Me.BasicGraphicTimer.Interval = 100 '50 ' for system timer usage
235     AddHandler AnimateEmergencyStopButtonlighting.Elapsed, AddressOf
AnimateEmergencyStopMotorButton
236     Me.AnimateEmergencyStopButtonlighting.AutoReset = True ' for system timer
usage
237     Me.AnimateEmergencyStopButtonlighting.Interval = 500
238     Me.AnimateEmergencyStopButtonlighting.Stop()
239     InitValofMeanBasic()
240     ' Add any initialization after the InitializeComponent() call.
241     DecimalSeperator = FindDecimalSeperator()
242 End Sub
243
244 Private Sub Thrust_Load(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MyBase.Load
245     Me.WindowState = FormWindowState.Normal
246     Me.Size = New System.Drawing.Size(1054, 816)
247     Me.Left = (My.Computer.Screen.WorkingArea.Size.Width - Me.Width) / 2
248     FilewithData.ReadInitStatus = True
249     PopulateCombos()
250     MnShowGraphs.Enabled = False
251     MnHideGraphs.Enabled = False
252     ElegxosStrofwn(False)
253     ConOrDisConsequenses(False)
254     InitialiseGraphs()
255     MnShowExcelGraphs.Enabled = ExcelIsInstalled
256     MnChartProperties.Enabled = ExcelIsInstalled
257     GraphRunFirstTime = True
258     FixToolTips(True, 0)
259     ChangeFormHeight("S")
260     BtShowGraphs.Enabled = False
261     MnShowNETGraphs_Click(sender, e)
262     Me.CreatedExcelProcID = MyClassCode.FindMyExcelProcess(NumOfExistingExcelProcs
, RunningExcelProcs) ' Used to find the created excel proc
263     ProcessAlreadyClosed = False
264     Getinitialvalues()
265
266 End Sub
267
268 Private Sub TrBrOutStrofes_Scroll(ByVal sender As System.Object, ByVal e As System
EventArgs) Handles TrBrOutStrofes.Scroll
269     If MnElegxosStrofwn.Checked Then
270         TxtOutStrofes.Text = TrBrOutStrofes.Value()
271     Else
272         TxtOutStrofes.Text = (TrBrOutStrofes.Value() / ControlActionDivider) & IIf
(TrBrOutStrofes.Value() <> MinControlAction And TrBrOutStrofes.Value() <>
MaxControlAction And (Int(TrBrOutStrofes.Value() / ControlActionDivider) =
(TrBrOutStrofes.Value() / ControlActionDivider)), DecimalSeperator & "00", IIf(Int
(TrBrOutStrofes.Value() / 10) = (TrBrOutStrofes.Value() / 10) And TrBrOutStrofes.
Value < MaxControlAction, "0", ""))

```

```

273     End If
274 End Sub
275
276 Public Sub DefineParamsForSerialPortAndOpenIt(ByVal ComPort As Integer, ByVal
BaudRate As Integer, ByVal Parity As Integer, ByVal DataBits As Integer, ByVal
StopBits As Integer)
277     MySerialPort.PortName = "COM" & Trim(Str(ComPort))
278
279     MySerialPort.BaudRate = BaudRate
280     MySerialPort.Parity = Parity
281     MySerialPort.DataBits = DataBits
282     MySerialPort.StopBits = StopBits
283     MySerialPort.ReceivedBytesThreshold = 1
284     Try
285         MySerialPort.Open()
286         ProblemOccured = False
287     Catch ex As Exception
288         ProblemOccured = True
289         MsgBox(MyClassCode.GetStr(WhichIsTheLang, 138), 0, MyClassCode.GetStr
(WhichIsTheLang, 109))
290     End Try
291 End Sub
292
293 Private Sub MySerialPort_DataReceived(ByVal sender As System.Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles MySerialPort.DataReceived
294     Static Dim Matrix(26) As Byte ' array dimension equal to bytes to read for
proper working
295     Dim k As Integer
296     Try
297         If MyClassCode.InitiatingConnectionwithPc = False Then
298             If FlagStartSend Then
299                 k = MySerialPort.BytesToRead
300                 If k >= MySerialPort.ReceivedBytesThreshold Then '25
301                     MySerialPort.Read(Matrix, 0, 27)
302                     If CheckIfValidData(Matrix, MySerialPort) = True Then
303
304                         Result1 = MyClassCode.DeCaptulate(0, 0, Matrix(2), Matrix
(3))
305                         Result7 = MyClassCode.DeCaptulate(0, 0, Matrix(0), Matrix
(1))
306                         AppendToText(1, Trim(Str(Result1)))
307
308                         Result2 = MyClassCode.DeCaptulate(0, 0, Matrix(6), Matrix
(7))
309                         Result8 = MyClassCode.DeCaptulate(0, 0, Matrix(4), Matrix
(5))
310                         AppendToText(2, Trim(Format(((Result2 - PWM_Min) /
(PWM_Duty / 100)), "#00.00")))
311
312
313                         Result3 = MyClassCode.DeCaptulate(Matrix(8), Matrix(9),
Matrix(10), Matrix(11))
314                         AppendToText(3, Trim(Str(Result3 / 10000)))
315
316                         Result4 = MyClassCode.DeCaptulate(Matrix(12), Matrix(13),
Matrix(14), Matrix(15))
317                         AppendToText(4, Trim(Str(Result4 / 1000)))
318
319                         Result5 = MyClassCode.DeCaptulate(Matrix(16), Matrix(17),
Matrix(18), Matrix(19))
320                         AppendToText(5, Trim(Str(Result5 / 100)))
321
322                         Result6 = MyClassCode.DeCaptulateTime(Matrix(20), Matrix
(21), Matrix(22), Matrix(23))
323                         AppendToText(6, MyClassCode.
ConvertMilsToMinsAndSecsAndMils(Result6))
324
325                         MyClassCode.DecaptulateStatusConditions(Matrix(24), Matrix
(25), StatusofIAR, Errors)
326
327                         StringToWrite = Format(Result1, "00000000.000") & " |*| "
& Format(((Result2 - PWM_Min) / (PWM_Duty / 100)), "000000.0000") & " |*| " &
Format(Result3 / 10000, "00000000.000") & " |*| " & Format(Result4 / 1000,
"000000000.00") & " |*| " & Format(Result5 / 100, "000000000.00") & " |*| " &
Format(Result7, "000000000.000") & " |*| " & Format(((Result8 - PWM_Min) /

```

```

(PWM_Duty / 100)), "000000000.000") & " |*| " & MyClassCode.
ConvertMilsToSecsAndMils(Result6)
328
329         sw.WriteLine(StringToWrite)
330         ServeControlByte(StatusofIAR, Me.VBstatus, Me.IARmode, Me.
MySerialPort)
331         SaveValuesInPubArrays(Result1, Result2, Result3, Result4,
Result5, Result8, Result7, Result6)
332         RpmMeter_refresh(Result1 / 1, Result2, Result3, Result4,
Result5) ' for basic graphics
333         ' -----refresh rpmmeter and bars-----
334         If Me.BasicGraphicTimer.Enabled = False Then
335             Me.BeginInvoke(New InvokeDelagateGraphs(AddressOf
DisplayDefaultGraphics)) '
336             End If
337         ' ----- data time refresh -----
338         MyClassCode.Refresh_Time_of_Recieved_Data_for_Timeout()
339         NoofData += 1
340         End If
341     End If
342     Else
343         CheckStartFlag(StartIndex)
344     End If
345     Else
346         MyClassCode.InitialSerialCom(MySerialPort, Me.WhichIsTheLang)
347     End If
348     Catch exc As Exception
349         ' An exception is raised when there is no information to read.
350     End Try
351 End Sub
352
353
354 Private Function CheckIfValidData(ByRef matrix() As Byte, ByRef myserialport As
System.IO.Ports.SerialPort) As Boolean
355     Dim CheckByte As Byte = 0
356     Dim BufferLessthanTreshold As Boolean = False
357     Dim bufferedbytes As Int16 = 0
358     Do While BufferLessthanTreshold = False
359
360         bufferedbytes = myserialport.BytesToRead
361         CheckByte = 0
362         For i As Int16 = 0 To matrix.Length - 2
363             CheckByte = matrix(i) Xor CheckByte
364         Next
365         If matrix(matrix.Length - 1) = CheckByte And matrix(12) = 4 And matrix(16)
= 5 And matrix(8) = 3 Then '
366             matrix(12) = 0
367             matrix(16) = 0
368             matrix(8) = 0
369             BufferLessthanTreshold = True
370             Return True
371         ElseIf MyClassCode.CheckIfCard_Closed_or_Reseted(matrix, WhichIsTheLang)
Then
372             BufferLessthanTreshold = True
373             Return False
374         ElseIf bufferedbytes > 0 Then
375             For i As Int16 = 0 To matrix.Length - 2 ' last value at index matrix.
Length - 1 which is the xor byte of all ather previous
376                 matrix(i) = matrix(i + 1)
377             Next i
378             matrix(matrix.Length - 1) = myserialport.ReadByte()
379             BufferLessthanTreshold = False 'condinue until you find the correct
bytes
380         ElseIf bufferedbytes <= 0 Then
381             BufferLessthanTreshold = True
382             Return False
383         End If
384     Loop
385 End Function
386 Private Sub Write2File(ByVal Str2Write As String)
387     sw.WriteLine(Str2Write)
388 End Sub
389 Private Sub AppendToText(ByVal WhichTextBox As Integer, ByVal text As String)

```

```

390     Dim WhichIsTheMessage As String = text
391     Dim array(1) As Object
392     array(0) = WhichTextBox
393     array(1) = text
394     Select Case WhichTextBox
395         Case 1 'TxtStrofes
396             If Me.TxtStrofes.InvokeRequired Then
397                 Me.TxtStrofes.BeginInvoke(New InvokeDelegate(AddressOf
InvokeMethod), 1, WhichIsTheMessage)
398             Else
399                 Me.TxtStrofes.Text += text
400             End If
401         Case 2 'TxtGkazi
402             If Me.TxtGkazi.InvokeRequired Then
403                 Me.TxtGkazi.BeginInvoke(New InvokeDelegate(AddressOf InvokeMethod)
, 2, WhichIsTheMessage)
404             Else
405                 Me.TxtGkazi.Text += text
406             End If
407         Case 3 'TxtTime2
408             If Me.TxtLoadCell.InvokeRequired Then
409                 Me.TxtLoadCell.BeginInvoke(New InvokeDelegate(AddressOf
InvokeMethod), 3, WhichIsTheMessage)
410             Else
411                 Me.TxtLoadCell.Text += text
412             End If
413         Case 4 ' TxtVoltage
414             If Me.TxtVoltage.InvokeRequired Then
415                 Me.TxtVoltage.BeginInvoke(New InvokeDelegate(AddressOf
InvokeMethod), 4, WhichIsTheMessage)
416             Else
417                 Me.TxtVoltage.Text += text
418             End If
419
420         Case 5
421             If Me.TxtAmperes.InvokeRequired Then
422                 Me.TxtAmperes.BeginInvoke(New InvokeDelegate(AddressOf
InvokeMethod), 5, WhichIsTheMessage)
423             Else
424                 Me.TxtAmperes.Text += text
425             End If
426         Case 6
427             If Me.TxtTime.InvokeRequired Then
428                 Me.TxtTime.BeginInvoke(New InvokeDelegate(AddressOf InvokeMethod)
, 6, WhichIsTheMessage)
429             Else
430                 Me.TxtTime.Enabled = True
431             End If
432         Case 7
433             If Me.StBStatus.InvokeRequired Then
434                 Me.StBStatus.BeginInvoke(New InvokeDelegate(AddressOf
InvokeMethod), 7, WhichIsTheMessage)
435             Else
436                 Me.StBPCConnStatusLbl.Text = MyClassCode.GetStr(WhichIsTheLang, 60)
437             End If
438         Case 8
439             If Me.BtShowGraphs.InvokeRequired Then
440                 Me.BtShowGraphs.BeginInvoke(New InvokeDelegate(AddressOf
InvokeMethod), 8, WhichIsTheMessage)
441             Else
442                 Me.BtShowGraphs.Enabled = True
443                 TreatMenuGraphs()
444             End If
445         Case 9
446         Case 10
447         Case 11
448         Case 12
449             If Me.PBGraph1.InvokeRequired Then
450                 Me.PBGraph1.BeginInvoke(New InvokeDelegate(AddressOf InvokeMethod)
, 12, WhichIsTheMessage)
451             Else
452                 Me.PBGraph1.Refresh()
453             End If
454         Case 13
455

```

```

456         If Me.PBGraph2.InvokeRequired Then
457             Me.PBGraph2.BeginInvoke(New InvokeDelegate(AddressOf InvokeMethod)
, 13, WhichIsTheMessage)
458         Else
459             Me.PBGraph2.Refresh()
460         End If
461     Case 14
462         BeginInvoke(New InvokeDelegate(AddressOf InvokeMethod), 14,
WhichIsTheMessage)
463     Case 15
464         BeginInvoke(New InvokeDelegate(AddressOf InvokeMethod), 15,
WhichIsTheMessage)
465     Case 16
466         If Me.PbRpmMeter1.InvokeRequired Then
467             Me.PbRpmMeter1.BeginInvoke(New InvokeDelegate(AddressOf
InvokeMethod), 16, WhichIsTheMessage)
468         Else
469             Me.PbRpmMeter1.Refresh()
470         End If
471     Case 17
472         BeginInvoke(New InvokeDelegate(AddressOf InvokeMethod), 17,
WhichIsTheMessage)
473     Case 18
474         BeginInvoke(New InvokeDelegate(AddressOf InvokeMethod), 18,
WhichIsTheMessage)
475     End Select
476
477 End Sub
478
479 Public Sub InvokeMethod(ByVal i As Integer, ByVal text As String)
480     Dim WhichIsTheMessage As String = text
481     Select Case i 'WhichIsTheTextBox
482     Case 1
483         Me.TxtStrofes.Text = WhichIsTheMessage
484         Me.TxtStrofes.Refresh()
485     Case 2
486         Me.TxtGkazi.Text = WhichIsTheMessage
487         Me.TxtGkazi.Refresh()
488     Case 3
489         Me.TxtLoadCell.Text = WhichIsTheMessage
490         Me.TxtLoadCell.Refresh()
491     Case 4
492         Me.TxtVoltage.Text = WhichIsTheMessage
493         Me.TxtVoltage.Refresh()
494     Case 5
495         Me.TxtAmperes.Text = WhichIsTheMessage
496         Me.TxtAmperes.Refresh()
497     Case 6
498         Me.TxtTime.Text = WhichIsTheMessage
499         Me.TxtTime.Refresh()
500     Case 7
501         Me.StBPCConnStatusLbl.Text = MyClassCode.GetStr(WhichIsTheLang, 60)
502     Case 8
503         Me.BtShowGraphs.Enabled = True
504         TreatMenuGraphs()
505     Case 9
506     Case 10
507     Case 11
508     Case 12
509         Me.PBGraph1.BringToFront()
510     Case 13
511         Me.PBGraph2.BringToFront()
512     Case 14
513         ElegxosStrofwn(True)
514     Case 15
515         ElegxosStrofwn(False)
516     Case 16
517         PbRpmMeter1.Refresh()
518     Case 17
519         DisplayDefaultGraphics()
520     Case 18
521         DisplayGraphics()
522     End Select
523 End Sub
524

```

```

525     Public Sub CheckStartFlag(ByRef StartIndex As Integer)
526         Dim Temp As Integer
527
528         Temp = MySerialPort.ReadChar()
529         If Temp = Teststart(StartIndex) Then
530             StartIndex += 1
531             If StartIndex = 5 Then
532                 FlagStartSend = True
533                 AppendToText(7, "")
534                 Me.MySerialPort.ReceivedBytesThreshold = 27
535                 AdjustStBLabels(1)
536                 StartIndex = 0
537                 AppendToText(8, "")
538                 MnShowGraphs.Enabled = True
539
540                 MyClassCode.Refresh_Time_of_Recieved_Data_for_Timeout()
541             End If
542         Else
543             StartIndex = 0
544         End If
545     End Sub
546
547     Public Sub ConOrDisConsequenses(ByVal WillConnect As Boolean)
548         BtStart.Enabled = Not WillConnect
549         MnConnect.Enabled = Not WillConnect
550         BtStop.Enabled = WillConnect
551         MnDisconnect.Enabled = WillConnect
552         BtEmergencyStop.Enabled = WillConnect
553         TrBrOutStrofes.Enabled = WillConnect
554         TxtSaveData.Enabled = Not WillConnect
555         BtSaveData.Enabled = Not WillConnect
556         If WillConnect Then
557             PrepareFileForWriting()
558         End If
559         If Not WillConnect Then
560             Try
561                 sw.Close()
562                 sw = Nothing
563             Catch ex As Exception
564                 'Stop
565             End Try
566         End If
567     End Sub
568
569     Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.
570     EventArgs) Handles BtClose.Click
571         If MySerialPort.IsOpen = True Then
572             MySerialPort.DiscardOutBuffer()
573             For i As Int16 = 0 To 100
574                 MySerialPort.Write(MyClassCode.SecureTransmit(4294967293), 0, 6)
575                 'causes IARcard to reset to initial conditions
576             Next
577         End If
578         MenuItem5_Click(sender, e)
579         MyClassCode.VbClosingState = MyPtyxClass.ThrustConnFormClose.Form_Close
580     End Sub
581
582     Private Sub MnConnect_Click(ByVal sender As System.Object, ByVal e As System.
583     EventArgs) Handles MnConnect.Click
584         If Not MyClassCode.IsValidFile(TxtSaveData.Text) Then
585             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 130), MsgBoxStyle.Information,
586             MyClassCode.GetStr(WhichIsTheLang, 109))
587         Exit Sub
588         End If
589
590         Dim fOpt As New Options
591         fOpt.SetDefaults(TheSelectedConfFile, WhichIsTheLang, True)
592         fOpt.ShowDialog()
593         If fOpt.bGo Then
594             DefineParamsForSerialPortAndOpenIt(fOpt.IntCommPort, fOpt.IntBaudRate,
595             fOpt.IntParityBits, fOpt.IntDataBits, fOpt.IntStopBits)
596             MyClassCode.InitParametersFor_Connection()
597             MyClassCode.InitCountTimeofconnection(New System.DateTime(Now.Ticks))
598             ArraysMet = 0
599         End If
600     End Sub

```



```

595         If ProblemOccured = False Then
596             MySerialPort.DiscardInBuffer()
597             MySerialPort.DiscardOutBuffer()
598             AdjustStBLabels(0)
599             ConOrDisConsequenses(True)
600         Else
601             ConOrDisConsequenses(False)
602         End If
603
604     End If
605     fOpt.Close()
606     fOpt = Nothing
607     If Me.VBSetPointRoot = SetpointsRoot.File Then
608         ResetFileTransitSelection(True) 'will get to here only if the card is
        reseted during file transmission mode control or the connenction closed normally
        after file transmit with out return to normal operation controls mode nad begin
        again the connection with our restart the programm
609     End If
610
611     ResetBufferArrays()
612     Reset_AnimColorOfEmergencyStopMotorButton()
613 End Sub
614
615 Private Sub MnDisconnect_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MnDisconnect.Click
616     AdjustStBLabels(2)
617     ConOrDisConsequenses(False)
618     MySerialPort.DiscardOutBuffer()
619     MySerialPort.DiscardInBuffer()
620     MySerialPort.Close()
621     FlagStartSend = False
622     If FilehasBeenUpdated Then
623         MyClassCode.AddFileFooter(TxtSaveData.Text, 4)
624         FilehasBeenUpdated = False
625     End If
626 End Sub
627 Private Sub MenuItem5_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MnExit.Click
628     If BtStart.Enabled = False Then
629         MnDisconnect_Click(sender, e)
630     End If
631     PrepareFileForClosing()
632     AdjustStBLabels(2)
633     Me.Hide()
634     MyClassCode.KillExcelProc(ProcessAlreadyClosed, CreatedExcelProcID)
635 End Sub
636
637 Private Sub NUDStrofes_KeyPress(ByVal sender As Object, ByVal e As System.Windows.
Forms.KeyPressEventArgs)
638     TrBrOutStrofes.Value() = TxtOutStrofes.Text
639 End Sub
640
641 Private Sub ElegxosStrofwn(ByVal IsEnabled As Boolean)
642     TrBrOutStrofes.Value = 0
643     TxtOutStrofes.Text = 0
644     If IsEnabled Then
645         TrBrOutStrofes.Maximum = MaxStrofes
646         TrBrOutStrofes.LargeChange = 200
647         LblOutStrofes.Text = MyClassCode.GetStr(WhichIsTheLang, 168)
648         MnElegxosGkaziou.Checked = False
649         MnElegxosStrofwn.Checked = True
650         VBstatus = Modeofapplication.ControlledRPM ' importan to be here
651         If FlagStartSend Then
652             MySerialPort.Write(MyClassCode.SecureTransmit(4294967281), 0, 6)
653         End If
654     Else
655         TrBrOutStrofes.Maximum = MaxControlAction
656         TrBrOutStrofes.LargeChange = ControlActionDivider
657         LblOutStrofes.Text = MyClassCode.GetStr(WhichIsTheLang, 52) & " (%)"
658         MnElegxosGkaziou.Checked = True
659         MnElegxosStrofwn.Checked = False
660         VBstatus = Modeofapplication.ControlPWMDuty ' importan to be here
661         If FlagStartSend Then
662             MySerialPort.Write(MyClassCode.SecureTransmit(4294967282), 0, 6)
663         End If

```

```

664         End If
665     End Sub
666
667
668     Private Sub MenuItem32_Click(ByVal sender As System.Object, ByVal e As System.
        EventArgs) Handles MnElegxosGkaziou.Click
669         ElegxosStrofwn(False)
670         If (Me.VBSetPointRoot = SetpointsRoot.File) Then : ResetFileTransitSelection
        (True) : End If
671     End Sub
672
673     Private Sub MenuItem29_Click(ByVal sender As System.Object, ByVal e As System.
        EventArgs) Handles MnElegxosStrofwn.Click
674         ElegxosStrofwn(True)
675         If (Me.VBSetPointRoot = SetpointsRoot.File) Then : ResetFileTransitSelection
        (True) : End If
676     End Sub
677     Private Function FindDecimalSeperator() As String
678         'Due to Regional Settings problem with Decimal Seperator we have these...
679         Dim Temp As String
680         TxtMyHidden.Text = 3 / 2
681         Temp = TxtMyHidden.Text
682
683         Return Temp.Substring(1, 1)
684     End Function
685
686     Private Sub SendImmediateStop()
687         Dim Response, MyEmergencyInt As UInt32
688
689         Response = MsgBox(MyClassCode.GetStr(WhichIsTheLang, 163), MsgBoxStyle.
        DefaultButton1 Or MsgBoxStyle.Critical Or MsgBoxStyle.YesNo, MyClassCode.GetStr
        (WhichIsTheLang, 109))
690         If Response = 6 Then
691             Me.VBMotor = MotorCondition.MotorStopped
692             MySerialPort.DiscardOutBuffer()
693             MyEmergencyInt = 4294967295 'h FFFF
694             MySerialPort.Write(MyClassCode.SecureTransmit(MyEmergencyInt), 0, 6)
695         End If
696     End Sub
697     Private Sub InitiateMotorFromStopState()
698         If MySerialPort.IsOpen = True Then : MySerialPort.Write(MyClassCode.
        SecureTransmit(4294967294), 0, 6)
699         End If
700     End Sub
701     Private Sub SendStopMotorSignal() 'for motor stop senario
702         If MySerialPort.IsOpen = True Then : MySerialPort.Write(MyClassCode.
        SecureTransmit(4294967295), 0, 6)
703         End If
704     End Sub
705     Private Sub BtEmergencyStop_Click(ByVal sender As System.Object, ByVal e As System
        .EventArgs) Handles BtEmergencyStop.Click
706         If Me.VBMotor = MotorCondition.MotorRunning Then
707             Me.BtEmergencyStop.BackColor = Color.OrangeRed
708             SendImmediateStop()
709             Me.TrBrOutStrofes.Value = Me.TrBrOutStrofes.Minimum 'sets the minimum
        value for trackbar
710             Me.TrBrOutStrofes_Scroll(sender, e) ' enumerates txt display
711             Me.TrBrOutStrofes.Enabled = False
712             Me.BtEmergencyStop.Text = MyClassCode.GetStr(WhichIsTheLang, 321)
713             Me.AnimateEmergencyStopButtonlighting.Start()
714         Else
715             If MsgBox(MyClassCode.GetStr(WhichIsTheLang, 319), MsgBoxStyle.OkCancel,
        MyClassCode.GetStr(WhichIsTheLang, 109)) = MsgBoxResult.Ok Then
716                 Me.VBMotor = MotorCondition.MotorRunning
717                 InitiateMotorFromStopState()
718                 Me.BtEmergencyStop.BackColor = Color.OrangeRed
719                 Me.TrBrOutStrofes.Value = Me.TrBrOutStrofes.Minimum 'sets the
        minimum value for trackbar
720                 Me.TrBrOutStrofes_Scroll(sender, e) ' enumerates txt display
721                 Me.TrBrOutStrofes.Enabled = IIf(Me.VBSetPointRoot = SetpointsRoot.
        FormControls, True, False)
722                 Me.BtEmergencyStop.Text = MyClassCode.GetStr(WhichIsTheLang, 162)
723                 Me.AnimateEmergencyStopButtonlighting.Stop()
724             End If
725         End If

```

```

726 End Sub
727
728 Private Sub BtSaveData_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles BtSaveData.Click
729     Try
730         Dim myStream As IO.Stream
731         Dim openFileDialog1 As New OpenFileDialog
732
733         openFileDialog1.InitialDirectory = "c:\Windows\Desktop"
734         openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*"
735         openFileDialog1.FilterIndex = 1
736         openFileDialog1.RestoreDirectory = True
737         openFileDialog1.Multiselect = False
738
739         If openFileDialog1.ShowDialog() = System.Windows.Forms.DialogResult.OK And
openFileDialog1.FileName.EndsWith(".txt") Then
740             myStream = openFileDialog1.OpenFile()
741             If Not (myStream Is Nothing) Then
742                 TxtSaveData.Text = openFileDialog1.FileName
743                 myStream.Close()
744             End If
745         Else
746             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 294), MsgBoxStyle.
Information, MyClassCode.GetStr(WhichIsTheLang, 295))
747         End If
748         Catch exp As Exception
749             MsgBox(exp.Message, MsgBoxStyle.Information, MyClassCode.GetStr
(WhichIsTheLang, 109))
750         End Try
751 End Sub
752
753 Public Sub PrepareFileForWriting()
754     MyClassCode.AddFileHeader(TxtSaveData.Text, 4)
755     Dim CheckFile As New IO.FileInfo(TxtSaveData.Text)
756
757     If CheckFile.Exists Then
758         FilehasBeenUpdated = True
759         sw = CheckFile.AppendText()
760     Else
761         FilehasBeenUpdated = False
762         FilehasBeenUpdated = False
763     End If
764 End Sub
765
766 Public Sub PrepareFileForClosing()
767     Try
768         sw.Close()
769         sw = Nothing
770     Catch ex As Exception
771         ' Do nothing for now
772     End Try
773
774     If FilehasBeenUpdated Then
775         MyClassCode.AddFileFooter(TxtSaveData.Text, 4)
776         FilehasBeenUpdated = False
777     End If
778 End Sub
779
780 Public Sub ActivateConfFile(ByVal MyFileName As String)
781     If MyClassCode.IsValidFile(MyFileName) Then
782         Try
783             TxtSaveData.Text = MyClassCode.FindParamValue(MyFileName, "Incoming
File", WhichIsTheLang)
784             TBReadFromFile.Text = MyClassCode.FindParamValue(MyFileName, "Outgoing
File", WhichIsTheLang)
785             Me.FilewithData.FileName = TBReadFromFile.Text
786         Catch E As Exception
787             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 119), MsgBoxStyle.
Information, MyClassCode.GetStr(WhichIsTheLang, 109))
788         End Try
789     End If
790 End Sub
791
792 Public Sub SetDefaults(ByVal SelectedConfFile As String)
793     If MyClassCode.IsValidFile(SelectedConfFile) Then

```

```

794         TheSelectedConfFile = SelectedConfFile
795         ActivateConfFile(SelectedConfFile)
796     Else
797         TheSelectedConfFile = ""
798     End If
799 End Sub
800
801 Private Sub AdjustStBLabels(ByVal Status As Integer)
802     'Status = 0 : Έγινε Σύνδεση, αλλά δεν έχει αρχίσει η παραλαβή πακέτων
803     'Status = 1 : Έγινε Σύνδεση, και έχει αρχίσει η παραλαβή πακέτων
804     'Status = 2 : Διακόπηκε η παραλαβή πακέτων
805
806     Select Case Status
807         Case 0
808             Me.StBPCnnStatusLbl.Text = MyClassCode.GetStr(WhichIsTheLang, 59)
809         Case 1
810             AppendToText(7, MyClassCode.GetStr(WhichIsTheLang, 60))
811         Case 2
812             Select Case FlagStartSend
813                 Case True
814                     Me.StBPCnnStatusLbl.Text = MyClassCode.GetStr(WhichIsTheLang, 63)
815                 Case False
816                     Me.StBPCnnStatusLbl.Text = MyClassCode.GetStr(WhichIsTheLang, 64)
817             End Select
818         End Select
819 End Sub
820 Public Sub ShowGraphs(ByVal ShowTheGraphs As Boolean)
821     PbShowAllGraphs = ShowTheGraphs 'Needed to enable required code in timer
822     interval
823     TreatMenuGraphs()
824     Select Case ShowTheGraphs
825         Case True
826             Me.BtShowGraphs.Text = MyClassCode.GetStr(WhichIsTheLang, 142)
827             ChangeFormHeight("L")
828             If Graphs = GraphicsDisplay.ShowGraphics Then
829                 PBGraph1.Visible = False
830                 PBGraph2.Visible = False
831             ElseIf Graphs = GraphicsDisplay.showChartGraphics Then
832                 PbGraph1Box1.Visible = False
833                 PbGraph2Box1.Visible = False
834             End If
835         Case False
836             Me.BtShowGraphs.Text = MyClassCode.GetStr(WhichIsTheLang, 139)
837             ChangeFormHeight("S")
838             Me.PBGraph1.Image = Nothing
839             Me.PBGraph2.Image = Nothing
840             Me.PbGraph1Box1.Image = Nothing
841             Me.PbGraph2Box1.Image = Nothing
842         End Select
843 End Sub
844 Public Sub TreatMenuGraphs()
845     Me.MnShowGraphs.Enabled = Not PbShowAllGraphs
846     Me.MnHideGraphs.Enabled = PbShowAllGraphs
847 End Sub
848
849
850 Public Sub CreateGraphs(ByVal matfromworker(,) As Int64)
851     app.Visible = False
852     Dim oldCI As System.Globalization.CultureInfo = System.Threading.Thread.
853     CurrentThread.CurrentCulture
854     System.Threading.Thread.CurrentThread.CurrentCulture = New System.
855     Globalization.CultureInfo("en-US")
856
857     'Format the table headings
858     Try
859         sheet.Range("A1").Value = "Time"
860         sheet.Range("B1").Value = LineLegend(matforworker(0, 1))
861         sheet.Range("C1").Value = LineLegend(matforworker(0, 3))
862         sheet.Range("D1").Value = "Time"
863         sheet.Range("E1").Value = LineLegend(matforworker(0, 2))
864
865     'Populate the sheet with some data

```

```

864         For i As Integer = 1 To Math.Max(Chart1Settings.NoPointstodisplay,           ✎
Chart2Settings.NoPointstodisplay)
865             sheet.Range("A" & (i + 1)).Value = matfromworker(i, 0) / MaxValdivider ✎
(0)
866             sheet.Range("B" & (i + 1)).Value = matfromworker(i, 1) / MaxValdivider ✎
(matfromworker(0, 1))
867             sheet.Range("C" & (i + 1)).Value = matfromworker(i, 3) / MaxValdivider ✎
(matfromworker(0, 3))
868             sheet.Range("D" & (i + 1)).Value = matfromworker(i, 0) / MaxValdivider ✎
(0)
869             sheet.Range("E" & (i + 1)).Value = matfromworker(i, 2) / MaxValdivider ✎
(matfromworker(0, 2))
870         Next
871         'Create and format a new 3D column chart
872         If GraphRunFirstTime = True Then
873             Chart1 = Doc.Charts.Add(, sheet)
874             Chart1.ChartType = XlChartType.xlLine
875
876         End If
877
878         Dim MaxPoint As Integer = Math.Max(Chart1Settings.NoPointstodisplay,           ✎
Chart2Settings.NoPointstodisplay)
879
880         If (Chart1Settings.Graphrpm = True And Chart1Settings.Graphwrpm = True) Or ✎
(Chart1Settings.Graphgkazi = True And Chart1Settings.Graphwgkazi = True) Then
881
882             Chart1.SetSourceData(sheet.Range("B1:C" & (Chart1Settings.           ✎
NoPointstodisplay)))
883
884             series1 = Chart1.SeriesCollection(1)
885             series1.XValues = sheet.Range("A" & (MaxPoint - Chart1Settings.           ✎
NoPointstodisplay + 2), "A" & (MaxPoint + 1))
886             series1.Values = sheet.Range("B" & (MaxPoint - Chart1Settings.           ✎
NoPointstodisplay + 2), "B" & (MaxPoint + 1))
887             series1.Name = sheet.Range("B1").Value.ToString
888
889             series2 = Chart1.SeriesCollection(2)
890             series2.XValues = sheet.Range("A" & (MaxPoint - Chart1Settings.           ✎
NoPointstodisplay + 2), "A" & (MaxPoint + 1))
891             series2.Values = sheet.Range("C" & (MaxPoint - Chart1Settings.           ✎
NoPointstodisplay + 2), "C" & (MaxPoint + 1))
892             series2.Name = sheet.Range("C1").Value.ToString
893             Chart1.HasTitle = False
894         Else
895             Chart1.SetSourceData(sheet.Range("B1:B" & (Chart1Settings.           ✎
NoPointstodisplay)))
896
897             series1 = Chart1.SeriesCollection(1)
898             If Me.Chart1Settings.NoPointstodisplay >= 2 Then
899                 series1.XValues = sheet.Range("A" & (MaxPoint - Chart1Settings.           ✎
NoPointstodisplay + 2), "A" & (MaxPoint + 1))
900                 series1.Values = sheet.Range("B" & (MaxPoint - Chart1Settings.           ✎
NoPointstodisplay + 2), "B" & (MaxPoint + 1))
901                 series1.Name = sheet.Range("B1").Value.ToString
902
903             Chart1.HasTitle = False
904         End If
905     End If
906
907     If GraphRunFirstTime = True Then
908         Chart2 = Doc.Charts.Add(, sheet)
909         Chart2.ChartType = XlChartType.xlLine
910         Chart2.HasTitle = False
911     End If
912     If Me.Chart2Settings.NoPointstodisplay >= 2 Then
913         Chart2.SetSourceData(sheet.Range("E1:E" & (Chart2Settings.           ✎
NoPointstodisplay)))
914         series3 = Chart2.SeriesCollection(1)
915         series3.XValues = sheet.Range("D" & (MaxPoint - Chart2Settings.           ✎
NoPointstodisplay + 2), "D" & (MaxPoint + 1))
916         series3.Values = sheet.Range("E" & (MaxPoint - Chart2Settings.           ✎
NoPointstodisplay + 2), "E" & (MaxPoint + 1))
917         series3.Name = sheet.Range("E1").Value.ToString
918     End If
919     System.Threading.Thread.CurrentThread.CurrentCulture = oldCI

```

```

920
921     Catch
922         ChartCreationError = True
923         BackgroundWorker1.CancelAsync()
924         System.Threading.Thread.CurrentThread.CurrentCulture = oldCI
925     End Try
926 End Sub
927
928 Public Sub DisplayChartatPbox()
929
930     If ChartCreationError = False Then
931         Dim Chart1Image, Chart2Image As Image
932         Dim ScalingRatio1 As Single
933         Dim ScalingRatio2 As Single
934         Dim oldCI As System.Globalization.CultureInfo = System.Threading.Thread.
CurrentThread.CurrentCulture
935         System.Threading.Thread.CurrentThread.CurrentCulture = New System.
Globalization.CultureInfo("en-US")
936
937         Try
938             Chart1.Export("C:\Chart1.gif", "GIF", False)
939             PBGraph1.ImageLocation = "C:\Chart1.gif"
940             Chart1.SetSourceData(sheet.Range("F1:F2"))
941             'Copy the chart to a PictureBox on this form, using the clipboard
942
943             Chart2.Export("C:\Chart2.gif", "GIF", False)
944             Me.PBGraph2.ImageLocation = "C:\Chart2.gif"
945             Chart2.SetSourceData(sheet.Range("F1:F2"))
946         Catch k As Exception
947             ChartCreationError = True
948             BackgroundWorker1.CancelAsync()
949         End Try
950         System.Threading.Thread.CurrentThread.CurrentCulture = oldCI
951
952     End If
953 End Sub
954
955 Public Sub SaveValuesInPubArrays(ByVal Rpm As Integer, ByVal Pwmload As Integer,
ByVal force As Integer, ByVal voltmeter As Integer, ByVal ampermeter As Integer,
ByVal setpointPwm As Integer, ByVal setpointRPM As Integer, ByVal timer As
UInteger)
956
957     Tmatrix(ArrayMet) = timer
958     Rmatrix(ArrayMet) = Rpm
959     WRmatrix(ArrayMet) = setpointRPM
960     Loadcell(ArrayMet) = force
961     Voltage(ArrayMet) = voltmeter
962     Ampere(ArrayMet) = ampermeter
963     gkazi(ArrayMet) = (Pwmload - PWM_Min) / (PWM_Duty / 1000)
964     Wantedgkazi(ArrayMet) = (setpointPwm - PWM_Min) / (PWM_Duty / 1000)
965     ArrayMet += 1
966     If ArrayMet = 100000 Then
967
968         Array.ConstrainedCopy(Tmatrix, 50000, Tmatrix, 0, 50000)
969         Array.ConstrainedCopy(Rmatrix, 50000, Rmatrix, 0, 50000)
970         Array.ConstrainedCopy(WRmatrix, 50000, WRmatrix, 0, 50000)
971         Array.ConstrainedCopy(Loadcell, 50000, Loadcell, 0, 50000)
972         Array.ConstrainedCopy(Voltage, 50000, Voltage, 0, 50000)
973         Array.ConstrainedCopy(Ampere, 50000, Ampere, 0, 50000)
974         Array.ConstrainedCopy(gkazi, 50000, gkazi, 0, 50000)
975         Array.ConstrainedCopy(Wantedgkazi, 50000, Wantedgkazi, 0, 50000)
976         ArrayMet = 50000
977
978     End If
979 End Sub
980
981
982 Public Sub InitialiseGraphs()
983     Try
984         If ExcelIsInstalled = True Then
985             System.Threading.Thread.CurrentThread.CurrentCulture = New System.
Globalization.CultureInfo("en-US")
986             app = New Microsoft.Office.Interop.Excel.Application
987             Doc = app.Workbooks.Add

```

```

988         sheet = Doc.Sheets(1)
989
990     End If
991
992     Catch ex As Exception
993
994 End Try
995
996 End Sub
997
998 Public Sub EnLargeGraphs(ByVal What As Char)
999     ' Possible values are
1000     ' 1 : For Graph1
1001     ' 2 : For Graph2
1002     ' R : For Reset
1003
1004     Select Case What
1005     Case "1" ' Long
1006         ShowObjects(False)
1007         PBGraph1.BringToFront()
1008         PBGraph1.Location = New System.Drawing.Point(0, 0)
1009         PBGraph1.Size = New System.Drawing.Size(Me.Width, Me.Height - Me.
1010 StBStatus.Height - 60)
1011         FixToolTips(False, 0)
1012         PBGraph1.Visible = True
1013         EnableMenu(False)
1014         Me.StBStatus.Visible = True
1015     Case "2" ' Long
1016         ShowObjects(False)
1017         PBGraph2.BringToFront()
1018         PBGraph2.Location = New System.Drawing.Point(0, 0)
1019         PBGraph2.Size = New System.Drawing.Size(Me.Width, Me.Height - Me.
1020 StBStatus.Height - 60)
1021         FixToolTips(False, 0)
1022         PBGraph2.Visible = True
1023         EnableMenu(False)
1024         Me.StBStatus.Visible = True
1025     Case "R" ' Short
1026         ShowObjects(True)
1027         Me.PBGraph1.Location = Me.Chart1Settings.ChartLocation
1028         Me.PBGraph1.Size = Me.Chart1Settings.ChartSize
1029         Me.PBGraph2.Location = Me.Chart2Settings.ChartLocation
1030         Me.PBGraph2.Size = Me.Chart2Settings.ChartSize
1031         FixToolTips(True, 0)
1032         EnableMenu(True)
1033         PbGraph1Box1.Visible = False
1034         PbGraph2Box1.Visible = False
1035
1036 End Select
1037 End Sub
1038
1039 Public Sub ShowObjects(ByVal DoShow As Boolean)
1040     GrBxDataSave.Visible = DoShow
1041     GrBxIncoming.Visible = DoShow
1042     GrBxOutgoing.Visible = DoShow
1043     PBGraph1.Visible = DoShow
1044     LblGraph1.Visible = DoShow
1045     PBGraph2.Visible = DoShow
1046     LblGraph2.Visible = DoShow
1047     PbGraph1Box1.Visible = DoShow
1048     PbGraph2Box1.Visible = DoShow
1049     StBStatus.Visible = DoShow
1050     BtClose.Visible = DoShow
1051     BtShowGraphs.Visible = DoShow
1052
1053 End Sub
1054
1055 Public Sub FixToolTips(ByVal Maximize As Boolean, ByVal Intention As Integer)
1056     Select Case Intention
1057     Case 0
1058         If Maximize Then
1059             MyToolTip.SetToolTip(PBGraph1, MyClassCode.GetStr(WhichIsTheLang,
1060 143))
1061             MyToolTip.SetToolTip(PBGraph2, MyClassCode.GetStr(WhichIsTheLang,
1062 143))
1063             PBGraph1.Cursor = ZoomInCursor

```

```

1059         PBGraph2.Cursor = ZoomInCursor
1060     Else
1061         MyToolTip.SetToolTip(PBGraph1, MyClassCode.GetStr(WhichIsTheLang,
144))
1062         MyToolTip.SetToolTip(PBGraph2, MyClassCode.GetStr(WhichIsTheLang,
144))
1063         PBGraph1.Cursor = ZoomOutCursor
1064         PBGraph2.Cursor = ZoomOutCursor
1065     End If
1066     Case 1
1067         MyToolTip.SetToolTip(PanelPropertiesGraph1, MyClassCode.GetStr
(WhichIsTheLang, 196))
1068         MyToolTip.SetToolTip(PanelPropertiesGraphs2, MyClassCode.GetStr
(WhichIsTheLang, 196))
1069         MyToolTip.SetToolTip(PanelChartProperties, MyClassCode.GetStr
(WhichIsTheLang, 196))
1070     End Select
1071
1072 End Sub
1073
1074 Private Sub PBGraph2_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles PBGraph2.Click
1075     If BtClose.Visible Then
1076         EnLargeGraphs("2")
1077     Else
1078         EnLargeGraphs("R")
1079     End If
1080 End Sub
1081
1082 Private Sub PBGraph1_Click_1(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles PBGraph1.Click
1083     If BtClose.Visible Then
1084         EnLargeGraphs("1")
1085     Else
1086         EnLargeGraphs("R")
1087     End If
1088 End Sub
1089
1090 Public Sub ChangeFormHeight(ByVal What As Char)
1091     Select Case What
1092     Case "S" ' Short
1093         Me.Height = 437
1094         Me.Top = (My.Computer.Screen.WorkingArea.Size.Height - Me.Height) / 2
1095         PBGraph1.Visible = False
1096         PBGraph1.Enabled = False
1097         PBGraph2.Visible = False
1098         PBGraph2.Enabled = False
1099         PbGraph1Box1.Visible = False
1100         PbGraph1Box1.Enabled = False
1101         PbGraph2Box1.Visible = False
1102         PbGraph2Box1.Enabled = False
1103
1104     Case "L" ' Long
1105         Me.Height = 820
1106         Me.Top = (My.Computer.Screen.WorkingArea.Size.Height - Me.Height) / 2
1107         PBGraph1.Visible = True
1108         PBGraph1.Enabled = True
1109         PBGraph2.Visible = True
1110         PBGraph2.Enabled = True
1111         PbGraph1Box1.Visible = True
1112         PbGraph1Box1.Enabled = True
1113         PbGraph2Box1.Visible = True
1114         PbGraph2Box1.Enabled = True
1115
1116     End Select
1117 End Sub
1118
1119 Private Sub BtShowGraphs_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles BtShowGraphs.Click
1120     If PbShowAllGraphs Then
1121         ShowGraphs(False)
1122         If Not FlagStartSend Then
1123             BtShowGraphs.Enabled = False
1124         End If
1125     Else

```



```

1126         ShowGraphs(True)
1127     End If
1128 End Sub
1129
1130 Private Sub BtStart_Click(ByVal sender As System.Object, ByVal e As System.
1131     EventArgs) Handles BtStart.Click
1132     MnConnect_Click(sender, e)
1133 End Sub
1134 Private Sub BtStop_Click(ByVal sender As System.Object, ByVal e As System.
1135     EventArgs) Handles BtStop.Click
1136     If MyClassCode.IARCardIsClosedProperly.Closed = True Then      *** for
1137     Close senario
1138     MnDisconnect_Click(sender, e)
1139     Else
1140     If MySerialPort.IsOpen Then : MySerialPort.Write(MyClassCode.
1141     SecureTransmit(4294967293), 0, 6) 'causes IARcard to reset to initial conditions
1142     End If
1143     End If
1144     MyClassCode.VbClosingState = MyPtyxClass.ThrustConnFormClose.Connection_Close
1145     ChangeModefromSendFile_to_Send_controls()
1146 End Sub
1147 Private Sub createRPMmeter(ByVal e As Graphics, ByVal Pbox As PictureBox)
1148     g = e
1149     g.InterpolationMode() = Drawing2D.InterpolationMode.High
1150     g.CompositingQuality = Drawing2D.CompositingQuality.HighSpeed
1151     'If firsttime = True Then
1152     Dim i As Int16
1153     Dim gonia As Double
1154     Dim formcolol As Color = Me.BackColor
1155     Dim MaxSizeRPMmeter As Single = Math.Min(Pbox.Size.Width - 30, Pbox.Size.
1156     Height - 30)
1157     Dim CenterPoint As New System.Drawing.PointF(Pbox.Width / 2, Pbox.Height / 2)
1158     gonia = ((RPM / 30000) * 300 + 120) * (Math.PI / 180)
1159
1160     g.FillPie(Brushes.Gray, CenterPoint.X - (MaxSizeRPMmeter / 2), CenterPoint.Y -
1161     (MaxSizeRPMmeter / 2), MaxSizeRPMmeter, MaxSizeRPMmeter, 120, 360)
1162     g.DrawPie(Pens.Black, CenterPoint.X - (MaxSizeRPMmeter / 2), CenterPoint.Y -
1163     (MaxSizeRPMmeter / 2), MaxSizeRPMmeter, MaxSizeRPMmeter, 120, 360)
1164
1165     For i = 0 To 30000 Step 1000
1166         Dim gonial As Integer = ((i / 30000) * 300)
1167         Dim textpoint As New System.Drawing.Point(CenterPoint.X + 65 * Math.Cos((
1168     (i / 30000) * 300 + 120) * (Math.PI / 180)), CenterPoint.Y + (MaxSizeRPMmeter / 2)
1169     * Math.Sin((i / 30000) * 300 + 120) * (Math.PI / 180))
1170         Select Case i
1171             Case 0
1172                 textpoint.X = textpoint.X - 10
1173                 textpoint.Y = textpoint.Y + 2
1174
1175                 g.DrawString(Str(i / 1000), New System.Drawing.Font("Arial bolt",
1176     10), Brushes.Red, textpoint)
1177             Case 5000     "This is a diagonal line drawn on the control"
1178                 g.DrawPie(Pens.Black, CenterPoint.X - (MaxSizeRPMmeter / 2),
1179     CenterPoint.Y - (MaxSizeRPMmeter / 2), MaxSizeRPMmeter, MaxSizeRPMmeter, 120,
1180     gonial)
1181                 g.DrawPie(Pens.Gray, CenterPoint.X - (MaxSizeRPMmeter / 2) + 15,
1182     CenterPoint.Y - (MaxSizeRPMmeter / 2) + 15, MaxSizeRPMmeter - 30, MaxSizeRPMmeter
1183     - 30, 120, gonial)
1184                 textpoint.X = textpoint.X - 25
1185                 textpoint.Y = textpoint.Y - 5
1186
1187                 g.DrawString(Str(i / 1000), New System.Drawing.Font("Arial", 10),
1188     Brushes.Red, textpoint)
1189             Case 10000

```

```

1186         g.DrawPie(Pens.Black, CenterPoint.X - (MaxSizeRPMmeter / 2),      ✎
CenterPoint.Y - (MaxSizeRPMmeter / 2), MaxSizeRPMmeter, MaxSizeRPMmeter, 120, ✎
gonial)
1187         g.DrawPie(Pens.Gray, CenterPoint.X - (MaxSizeRPMmeter / 2) + 15, ✎
CenterPoint.Y - (MaxSizeRPMmeter / 2) + 15, MaxSizeRPMmeter - 30, MaxSizeRPMmeter ✎
- 30, 120, gonial)
1188
1189         textpoint.X = textpoint.X - 30
1190         textpoint.Y = textpoint.Y - 10
1191         g.DrawString(Str(i / 1000), New System.Drawing.Font("Arial", 10), ✎
Brushes.Red, textpoint)
1192         Case 15000
1193         g.DrawPie(Pens.Black, CenterPoint.X - (MaxSizeRPMmeter / 2),      ✎
CenterPoint.Y - (MaxSizeRPMmeter / 2), MaxSizeRPMmeter, MaxSizeRPMmeter, 120, ✎
gonial)
1194         g.DrawPie(Pens.Gray, CenterPoint.X - (MaxSizeRPMmeter / 2) + 15, ✎
CenterPoint.Y - (MaxSizeRPMmeter / 2) + 15, MaxSizeRPMmeter - 30, MaxSizeRPMmeter ✎
- 30, 120, gonial)
1195         textpoint.X = textpoint.X - 10
1196         textpoint.Y = textpoint.Y - 15
1197         g.DrawString(Str(i / 1000), New System.Drawing.Font("Arial", 10), ✎
Brushes.Red, textpoint)
1198
1199         Case 20000
1200         g.DrawPie(Pens.Black, CenterPoint.X - (MaxSizeRPMmeter / 2),      ✎
CenterPoint.Y - (MaxSizeRPMmeter / 2), MaxSizeRPMmeter, MaxSizeRPMmeter, 120, ✎
gonial)
1201         g.DrawPie(Pens.Gray, CenterPoint.X - (MaxSizeRPMmeter / 2) + 15, ✎
CenterPoint.Y - (MaxSizeRPMmeter / 2) + 15, MaxSizeRPMmeter - 30, MaxSizeRPMmeter ✎
- 30, 120, gonial)
1202         textpoint.X = textpoint.X + 8
1203         textpoint.Y = textpoint.Y - 10
1204         g.DrawString(Str(i / 1000), New System.Drawing.Font("Arial", 10), ✎
Brushes.Red, textpoint)
1205
1206         Case 25000
1207         g.DrawPie(Pens.Black, CenterPoint.X - (MaxSizeRPMmeter / 2),      ✎
CenterPoint.Y - (MaxSizeRPMmeter / 2), MaxSizeRPMmeter, MaxSizeRPMmeter, 120, ✎
gonial)
1208         g.DrawPie(Pens.Gray, CenterPoint.X - (MaxSizeRPMmeter / 2) + 15, ✎
CenterPoint.Y - (MaxSizeRPMmeter / 2) + 15, MaxSizeRPMmeter - 30, MaxSizeRPMmeter ✎
- 30, 120, gonial)
1209         textpoint.X = textpoint.X + 5
1210         textpoint.Y = textpoint.Y - 5
1211         g.DrawString(Str(i / 1000), New System.Drawing.Font("Arial", 10), ✎
Brushes.Red, textpoint)
1212
1213         Case 30000
1214         g.DrawPie(Pens.Black, CenterPoint.X - (MaxSizeRPMmeter / 2),      ✎
CenterPoint.Y - (MaxSizeRPMmeter / 2), MaxSizeRPMmeter, MaxSizeRPMmeter, 120, ✎
gonial)
1215         g.DrawPie(Pens.Gray, CenterPoint.X - (MaxSizeRPMmeter / 2) + 15, ✎
CenterPoint.Y - (MaxSizeRPMmeter / 2) + 15, MaxSizeRPMmeter - 30, MaxSizeRPMmeter ✎
- 30, 120, gonial)
1216         textpoint.X = textpoint.X - 5
1217         textpoint.Y = textpoint.Y + 2
1218         g.DrawString(Str(i / 1000), New System.Drawing.Font("Arial", 10), ✎
Brushes.Red, textpoint)
1219         Case Else
1220         g.DrawPie(Pens.Black, CenterPoint.X - (MaxSizeRPMmeter / 2),      ✎
CenterPoint.Y - (MaxSizeRPMmeter / 2), MaxSizeRPMmeter, MaxSizeRPMmeter, 120, ✎
gonial)
1221         g.DrawPie(Pens.Gray, CenterPoint.X - (MaxSizeRPMmeter / 2) + 5, ✎
CenterPoint.Y - (MaxSizeRPMmeter / 2) + 5, MaxSizeRPMmeter - 10, MaxSizeRPMmeter - ✎
10, 120, gonial)
1222         End Select
1223         Next i
1224
1225         DiameterRPM = MaxSizeRPMmeter / 2 - 5
1226         DiameterMin = 6
1227         Dim POINT1 As New System.Drawing.Point(DiameterRPM * Math.Cos(gonia) + ✎
CenterPoint.X, DiameterRPM * Math.Sin(gonia) + CenterPoint.Y)
1228         Dim POINT2 As New System.Drawing.Point(DiameterMin * Math.Cos(gonia - (Math.PI ✎
/ 2)) + CenterPoint.X, DiameterMin * Math.Sin(gonia - (Math.PI / 2)) + ✎
CenterPoint.Y)

```

```

1229     Dim POINT3 As New System.Drawing.Point(DiameterMin * Math.Cos(gonia + (Math.PI
    / 2)) + CenterPoint.X, DiameterMin * Math.Sin(gonia + (Math.PI / 2)) +
    CenterPoint.Y)
1230     Dim POINTER As System.Drawing.Point() = {POINT1, POINT2, POINT3}
1231
1232     g.DrawPie(Pens.Black, CenterPoint.X - DiameterMin, CenterPoint.Y - DiameterMin
    , DiameterMin * 2, DiameterMin * 2, 0, 360)
1233     g.FillPie(Brushes.Gray, CenterPoint.X - (MaxSizeRPMmeter / 2) + 15,
    CenterPoint.Y - (MaxSizeRPMmeter / 2) + 15, MaxSizeRPMmeter - 30, MaxSizeRPMmeter
    - 30, 120, 300)
1234     g.DrawString("x1000", New System.Drawing.Font("Arial", 8), Brushes.Red,
    CenterPoint.X - 15, CenterPoint.Y + 15)
1235     g.DrawString(Str(RPM), New System.Drawing.Font("Arial", 8), Brushes.Red,
    CenterPoint.X - 20, CenterPoint.Y + 45)
1236     g.FillPolygon(Brushes.Chocolate, POINTER)
1237     g.FillPie(Brushes.Chocolate, CenterPoint.X - DiameterMin, CenterPoint.Y -
    DiameterMin, DiameterMin * 2, DiameterMin * 2, 0, 360)
1238     Dim pointer1 As System.Drawing.PointF() = {POINT1, POINT2, New System.Drawing.
    Point(CenterPoint.X, CenterPoint.Y)}
1239     g.FillPolygon(Brushes.Orange, pointer1)
1240     Dim angle As Integer = (RPM / 30000) * 300
1241     g.FillPie(Brushes.Chocolate, CenterPoint.X - DiameterMin, CenterPoint.Y -
    DiameterMin, DiameterMin * 2, DiameterMin * 2, 120 + angle, 180)
1242     g.FillPie(Brushes.Orange, CenterPoint.X - DiameterMin, CenterPoint.Y -
    DiameterMin, DiameterMin * 2, DiameterMin * 2, 120 + angle + 180, 180)
1243
1244 End Sub
1245
1246 Private Sub PbRpmMeter1_Paint(ByVal sender As System.Object, ByVal e As System.
    Windows.Forms.PaintEventArgs) Handles PbRpmMeter1.Paint
    creatorRPMmeter(e.Graphics, PbRpmMeter1)
1247
1248 End Sub
1249
1250 Private Sub RpmMeter_refresh(ByVal RpmValue As Integer, ByVal pwm1 As Integer,
    ByVal load1 As Integer, ByVal volt1 As Integer, ByVal ampl As Integer)
1251     If Me.BasicGraphicTimer.Enabled = False Then
1252         RPM = RpmValue
1253         pwm = pwm1
1254         cell = load1
1255         volt = volt1
1256         amp = ampl
1257     Else
1258         SaveMeanvalue(RpmValue, pwm1, load1, volt1, ampl)
1259     End If
1260 End Sub
1261
1262 Private Sub TmrRPMmeter_Tick(ByVal sender As System.Object, ByVal e As System.
    EventArgs) Handles TmrGraphsTimer.Tick
1263
1264     If FlagStartSend = True Then
1265         If Graphs = GraphicsDisplay.ShowGraphics Then
1266             PbGraph1Box1.Refresh()
1267             PbGraph2Box1.Refresh()
1268         Else
1269             Me.DisplayGraphics()
1270         End If
1271     End If
1272
1273 End Sub
1274
1275
1276 Private Sub drawgraphline(ByVal e As Graphics, ByVal TimeMatrix() As UInt32, ByVal
    line1() As Integer, ByVal maxvalue As Integer, ByVal index As Integer, ByVal pen
    As Pen, ByVal graphsetting As GraphSettings, ByVal Pbox As PictureBox, ByVal
    zeropoint As System.Drawing.Point)
1277     Try
1278         If index > 2 Then
1279             If graphsetting.NofPointsofLine < 2 Then
1280
1281                 GoTo jumptoend
1282             End If
1283             ReDim pointmat(graphsetting.NofPointsofLine - 1, 1)
1284
1285             Dim pointsofline() As System.Drawing.Point
1286             If index >= graphsetting.NofPointsofLine Then

```

```

1287         For z As Integer = 0 To graphsetting.NofPointsofLine - 1
1288             pointmat(z, 1) = Pbox.Size.Height - ((line1(index -
(graphsetting.NofPointsofLine - 1) + z) / maxvalue) * Pbox.Size.Height) 'y point
1289             If Not TimeMatrix(index) = TimeMatrix(index - (graphsetting.
NofPointsofLine - 1)) Then 'i can not have dieresi with zero
1290                 pointmat(z, 0) = ((TimeMatrix(index - (graphsetting.
NofPointsofLine - 1) + z) - TimeMatrix(index - (graphsetting.NofPointsofLine -
1)))) / (CLng(TimeMatrix(index)) - CLng(TimeMatrix(index - (graphsetting.
NofPointsofLine - 1)))) * Pbox.Size.Width + zeropoint.X ' X point
1291                 ElseIf 0 Then ' case 0
1292                     pointmat(z, 0) = 0
1293                 End If
1294             Next z
1295
1296         Else
1297             For z As Integer = 0 To index
1298
1299                 pointmat(z, 1) = Pbox.Size.Height - ((line1(z) / maxvalue) *
Pbox.Size.Height) 'y point
1300                 If Not TimeMatrix(index) = TimeMatrix(0) Then
1301                     pointmat(z, 0) = ((TimeMatrix(z) - TimeMatrix(0)) / (CLng
(TimeMatrix(index)) - CLng(TimeMatrix(0)))) * Pbox.Size.Width + zeropoint.X ' X
point
1302                     Else
1303                         pointmat(z, 0) = 0
1304                     End If
1305                 Next z
1306
1307             End If
1308
1309             Dim timeindex As Integer
1310             If index >= graphsetting.NofPointsofLine Then
1311                 ReDim pointsofline(graphsetting.NofPointsofLine - 1)
1312
1313                 For intexP As Integer = 0 To graphsetting.NofPointsofLine - 1
1314                     Dim rpmpoints As New System.Drawing.Point(pointmat(intexP, 0),
pointmat(intexP, 1))
1315
1316                     pointsofline(intexP) = rpmpoints
1317                 Next intexP
1318                 timeindex = index - (graphsetting.NofPointsofLine - 1)
1319             Else
1320                 ReDim pointsofline(index)
1321                 For intexP As Integer = 0 To index
1322                     Dim rpmpoints As New System.Drawing.Point(pointmat(intexP, 0),
pointmat(intexP, 1))
1323
1324                     pointsofline(intexP) = rpmpoints
1325                 Next intexP
1326
1327                 timeindex = 0
1328             End If
1329             If graphsetting.Verticallines = 0 Then
1330                 graphsetting.Verticallines = 1
1331             End If
1332             Dim timescale As Integer = (TimeMatrix(index) - TimeMatrix(timeindex))
/ graphsetting.Verticallines
1333             e.DrawString("Time scale: " & Str(timescale) & "ms", New System.
Drawing.Font("Arial", 10), Brushes.Black, Pbox.Width - 140, Pbox.Height - 30)
1334             e.DrawLines(pen, pointsofline) ' color of rpm line
1335
1336         jumptoend: End If
1337         Catch
1338             Stop
1339         End Try
1340     End Sub
1341 End Sub
1342 Private Sub DrawDashedLine(ByVal e As Graphics, ByVal StartPoint As System.Drawing
.Point, ByVal EndPoint As System.Drawing.Point, ByVal pen As Pen, ByVal Pbox As
PictureBox)
1343     Dim nextpoint As New System.Drawing.Point(StartPoint.X, StartPoint.Y)
1344     Dim NoOfDushed As Integer = 1
1345     'Y = aX+b
1346     'a = Y2-Y1/X2-X1
1347     'b = Y1-aX1

```

```

1348     Dim a As Single
1349     Dim b As Single
1350     Dim Fpointdushed As New System.Drawing.Point(StartPoint.X, StartPoint.Y)
1351     Dim Spointdushed As New System.Drawing.Point(Fpointdushed.X + 10, a *
Fpointdushed.X + b)
1352     If (EndPoint.X - StartPoint.X) = 0 Then
1353         Do While Fpointdushed.Y > EndPoint.Y
1354             e.DrawLine(pen, Fpointdushed.X, Fpointdushed.Y, Fpointdushed.X,
Fpointdushed.Y + 10)
1355             Fpointdushed.Y = Fpointdushed.Y - 20
1356         Loop
1357     Else
1358         a = (EndPoint.Y - StartPoint.Y) / (EndPoint.X - StartPoint.X)
1359         b = (StartPoint.Y - a * StartPoint.X)
1360         Do While Fpointdushed.X < EndPoint.X
1361             Spointdushed.X = Fpointdushed.X + 10
1362             Spointdushed.Y = a * Fpointdushed.X + b
1363             e.DrawLine(pen, Fpointdushed.X, Fpointdushed.Y, Spointdushed.X,
Spointdushed.Y)
1364             Fpointdushed.X = Fpointdushed.X + 20
1365         Loop
1366     End If
1367 End Sub
1368 Private Sub DrawHorizontalLines(ByVal e As Graphics, ByVal numberOfLines As Int16,
ByVal NumberOfDivisions As Int16, ByVal typeoflines As typeofline, ByVal
zeropoint As System.Drawing.Point, ByVal pen As Pen, ByVal Pbox As PictureBox)
1369     Dim dushedpen As Pen = pen
1370     Dim linedisplay As Int16 = 0
1371
1372     If numberOfLines = 0 Then
1373         numberOfLines = 1
1374     End If
1375     If NumberOfDivisions = 0 Then
1376         NumberOfDivisions = 1
1377     End If
1378
1379     For i As Integer = 0 To numberOfLines * NumberOfDivisions Step 1
1380         Dim Yvalue As Double = Pbox.Height - (i / (numberOfLines *
NumberOfDivisions)) * (Pbox.Height)
1381
1382         Dim ypoint As New System.Drawing.Point(Pbox.Size.Width, Yvalue)
1383         Dim startPoint As New System.Drawing.Point(zeropoint.X, ypoint.Y)
1384         Dim stopPoint As New System.Drawing.Point(Pbox.Width, ypoint.Y)
1385         Dim CaseSelectValue As Int16 = linedisplay * NumberOfDivisions
1386         Select Case i
1387             Case 0
1388                 ' draw only inside chart the line dushed
1389                 GoTo solidline
1390             Case numberOfLines * NumberOfDivisions
1391                 GoTo solidline
1392             Case CaseSelectValue
1393                 Select Case typeoflines
1394                     Case typeofline.DashedLine
1395                         DrawDashedLine(e, startPoint, stopPoint, dushedpen, Pbox)
1396                         linedisplay += 1
1397                     Case typeofline.SolidLine
1398                         e.DrawLine(Pens.Gray, zeropoint.X, ypoint.Y, Pbox.Width,
ypoint.Y)
1399                         linedisplay += 1
1400                 End Select
1401             Case Else
1402                 e.DrawLine(Pens.Black, zeropoint.X, ypoint.Y, zeropoint.X + 10,
ypoint.Y)
1403         End Select
1404     Next i
1405 End Sub
1406 Private Sub DrawVerticalLines(ByVal e As Graphics, ByVal numberoflines As Int16,
ByVal zeropoint As System.Drawing.Point, ByVal dushedpen As Pen, ByVal typeoflines
As typeofline, ByVal Pbox As PictureBox)
1407     If numberoflines = 0 Then
1408         numberoflines = 1
1409     End If
1410     For i As Integer = 0 To numberoflines Step 1
1411         Dim Xvalue As Double = (i / numberoflines) * (Pbox.Width)

```

```

1412
1413     Dim Xpoint As New System.Drawing.Point(Xvalue + zeropoint.X, Pbox.Size.
Height)
1414     Dim stoppoint As New System.Drawing.Point(Xpoint.X, 0)
1415     Select Case typeoflines
1416         Case typeofline.DushedLine
1417             If i = 0 Then
1418                 e.DrawLine(dushedpen, Xpoint.X, zeropoint.Y, Xpoint.X, 0)
1419             Else
1420                 DrawDashedLine(e, Xpoint, stoppoint, dushedpen, Pbox)
1421             End If
1422
1423         Case typeofline.SolidLine
1424             e.DrawLine(dushedpen, Xpoint.X, zeropoint.Y, Xpoint.X, 0)
1425     End Select
1426
1427     Next i
1428 End Sub
1429
1430 Private Sub drawvalues4lines(ByVal PWM As Boolean, ByVal maxval1 As Integer, ByVal
Loadcell As Boolean, ByVal maxval2 As Integer, ByVal Voltage As Boolean, ByVal
maxval3 As Integer, ByVal Amperes As Boolean, ByVal maxval4 As Integer, ByVal
NoFLinesDisplay As Int16, ByVal Units() As String, ByVal brush() As Brush, ByVal
font As System.Drawing.Font, ByVal Pbox As PictureBox, ByVal e As Graphics)
1431     Dim f As UInt16 = Math.Abs(PWM + Loadcell + Voltage + Amperes)
1432     Dim pbsize As New System.Drawing.Point(Pbox.Size.Width, Pbox.Size.Height)
1433     Dim zeropoint As New System.Drawing.Point(0, Pbox.Size.Height - 0)
1434     Dim checmatrix(,) As Integer = {{-PWM, maxval1}, {-Loadcell, maxval2}, {-
Voltage, maxval3}, {-Amperes, maxval4}}
1435     Dim HeadLegend() As String '= {"% ", "Newton", "Volt", "Amp"}
1436     HeadLegend = Units
1437     Dim legeng As String = ""
1438     Dim g As Double = 0
1439     Dim correctLastLine As Int16 = 0
1440
1441     If NoFLinesDisplay = 0 Then
1442         NoFLinesDisplay = 1 'case for error
1443     End If
1444
1445     For l As Integer = 0 To 3
1446
1447         If (checmatrix(l, 0) = 1) Then
1448
1449             If g = f - 1 Then ' corect the position at X of strings between the
points upper and bottom of picturebox
1450                 Select Case g
1451                     Case 0 ' for firstline if only 1 line displayed
1452                         correctLastLine = 0
1453                     Case Else 'for last line at any other situation
1454                         correctLastLine = -3 * font.SizeInPoints 'move the x
value of line 3 chars length at left 'correctLastLine = -30 - length - 8
1455                 End Select
1456
1457             End If
1458             For i As Integer = 0 To NoFLinesDisplay Step 1
1459                 Dim TextToDisplay As String = ""
1460                 Dim Yvalue As Double = pbsize.Y - (i / NoFLinesDisplay) * (pbsize.
Y)
1461
1462                 If i = 0 Then 'botom point to display special case at
the bottom
1463                     Yvalue += -10
1464                     legeng = ""
1465                     TextToDisplay = Format(i * checmatrix(l, 1) / NoFLinesDisplay,
"#0.#") & legeng
1466                 ElseIf i = NoFLinesDisplay Then ' Upper point to display
special case at the bottom
1467                     Yvalue += 10
1468                     legeng = HeadLegend(l)
1469
1470                 If g = f - 1 Then ' accesed only for last line
1471                     TextToDisplay = legeng & " " & Format(i * checmatrix(l, 1)
/ NoFLinesDisplay, "#0.#")
1472                     Dim length As Integer = 8 * legeng.Length ' Arial 8
1473                     Select Case g

```

```

1474         Case 0 ' for firstline if only 1 line displayed
1475         correctLastLine = 0 'font.SizeInPoints '-30 -
length - 8
1476         TextToDisplay = Format(i * checmatrix(l, 1) /
NoFLinesDisplay, "#0.#") & " " & legeng
1477         Case Else 'for last line ,on any other case
1478         correctLastLine = -30 - length - 8
1479         End Select
1480
1481     Else
1482         TextToDisplay = Format(i * checmatrix(l, 1) /
NoFLinesDisplay, "#0.#") & " " & legeng
1483         End If
1484
1485     Else
1486         legeng = ""
1487         TextToDisplay = Format(i * checmatrix(l, 1) / NoFLinesDisplay,
"#0.#") & legeng
1488         End If
1489
1490
1491     Dim ypoint As New System.Drawing.Point(Pbox.Size.Width, Yvalue)
1492
1493
1494     Select Case f
1495     Case 1
1496         Select Case l
1497         Case 0
1498             e.DrawString(TextToDisplay, font, brush(l),
zeropoint.X + (g * pbsize.X) + correctLastLine, ypoint.Y - 10)
1499         Case 1
1500             e.DrawString(TextToDisplay, font, brush(l),
zeropoint.X + (g * pbsize.X) + correctLastLine, ypoint.Y - 10)
1501         Case 2
1502             e.DrawString(TextToDisplay, font, brush(l),
zeropoint.X + (g * pbsize.X) + correctLastLine, ypoint.Y - 10)
1503         Case 3
1504             e.DrawString(TextToDisplay, font, brush(l),
zeropoint.X + (g * pbsize.X) + correctLastLine, ypoint.Y - 10)
1505         End Select
1506
1507     Case 2
1508         Select Case l
1509         Case 0
1510             e.DrawString(TextToDisplay, font, brush(l),
zeropoint.X + (g * pbsize.X) + correctLastLine, ypoint.Y - 10)
1511         Case 1
1512             e.DrawString(TextToDisplay, font, brush(l),
zeropoint.X + (g * pbsize.X) + correctLastLine, ypoint.Y - 10)
1513         Case 2
1514             e.DrawString(TextToDisplay, font, brush(l),
zeropoint.X + (g * pbsize.X) + correctLastLine, ypoint.Y - 10)
1515         Case 3
1516             e.DrawString(TextToDisplay, font, brush(l),
zeropoint.X + (g * pbsize.X) + correctLastLine, ypoint.Y - 10)
1517         End Select
1518
1519     Case 3
1520         Select Case l
1521         Case 0
1522             e.DrawString(TextToDisplay, font, brush(l),
zeropoint.X + (g * pbsize.X / 2) + correctLastLine, ypoint.Y - 10)
1523         Case 1
1524             e.DrawString(TextToDisplay, font, brush(l),
zeropoint.X + (g * pbsize.X / 2) + correctLastLine, ypoint.Y - 10)
1525         Case 2
1526             e.DrawString(TextToDisplay, font, brush(l),
zeropoint.X + (g * pbsize.X / 2) + correctLastLine, ypoint.Y - 10)
1527         Case 3
1528             e.DrawString(TextToDisplay, font, brush(l),
zeropoint.X + (g * pbsize.X / 2) + correctLastLine, ypoint.Y - 10)
1529         End Select
1530
1531     Case 4
1532         Select Case l

```

```

1533             Case 0
1534                 e.DrawString(TextToDisplay, font, brush(1),
zeropoint.X + (g * pbsize.X / 3) + correctLastLine, ypoint.Y - 10)
1535             Case 1
1536                 e.DrawString(TextToDisplay, font, brush(1),
zeropoint.X + (g * pbsize.X / 3) + correctLastLine, ypoint.Y - 10)
1537             Case 2
1538                 e.DrawString(TextToDisplay, font, brush(1),
zeropoint.X + (g * pbsize.X / 3) + correctLastLine, ypoint.Y - 10)
1539             Case 3
1540                 e.DrawString(TextToDisplay, font, brush(1),
zeropoint.X + (g * pbsize.X / 3) + correctLastLine, ypoint.Y - 10)
1541             End Select
1542
1543
1544         End Select
1545
1546     Next i
1547     g += 1
1548 End If
1549 Next l
1550 End Sub
1551
1552 Private Sub displayLineLegend(ByVal e As Graphics, ByVal TxtLegend() As String,
ByVal brush() As Brush, ByVal DisplayLegend() As Boolean, ByVal showlegend As
Boolean, ByVal font As System.Drawing.Font, ByVal point2display As System.Drawing.
Point, ByVal sizeoflegend As Size)
1553
1554     If showlegend = False Then
1555         GoTo endpoint
1556     End If
1557     Dim lengthoflegend, Nofmean As Integer
1558     For j As Integer = 0 To TxtLegend.Length - 1
1559         If DisplayLegend(j) Then
1560             lengthoflegend += TxtLegend(j).Length * font.SizeInPoints '* 0.82 '
size in points of all displayed chars
1561             Nofmean += 1
1562         End If
1563     Next
1564     Dim diferense As Integer
1565     If lengthoflegend > sizeoflegend.Width Or Nofmean < 2 Then
1566         sizeoflegend.Width = lengthoflegend + 8
1567         diferense = 0
1568     Else
1569
1570         diferense = sizeoflegend.Width - lengthoflegend 'mean defferance
1571         diferense = diferense / (Nofmean - 1)
1572     End If
1573     Dim rect As New System.Drawing.Rectangle(point2display, sizeoflegend)
1574     e.FillRectangle(Brushes.White, rect)
1575     For i As Integer = 0 To TxtLegend.Length - 1
1576         If DisplayLegend(i) = True Then
1577             e.DrawString(TxtLegend(i), font, brush(i), rect)
1578             rect.X = rect.X + TxtLegend(i).Length * font.SizeInPoints
1579             rect.X = rect.X + diferense
1580         End If
1581     Next
1582 endpoint:
1583 End Sub
1584
1585 Private Sub createGraphsfor2Box(ByVal e As Graphics, ByVal TimeMatrix() As UInt32,
ByVal PWMvalue() As Integer, ByVal load() As Integer, ByVal Volt() As Integer,
ByVal Amp() As Integer, ByVal MaxValueOfLinesData() As Integer, ByVal
displayLineValue() As Integer, ByVal Unitsfordisplay() As String, ByVal
LegendsofLines() As String, ByVal graphsetting As GraphSettings, ByVal pen() As
Pen, ByVal Brusheses() As Brush, ByVal Pbox As PictureBox, ByVal Index As Integer)
1586
1587     Dim k As Graphics = e
1588     k.InterpolationMode = Drawing2D.InterpolationMode.Low
1589     Dim sz As New System.Drawing.Size
1590     Dim zeropoint As New System.Drawing.Point(0, Pbox.Size.Height - 0)
1591
1592     Dim HeadLegendUnits() As String = Unitsfordisplay

```



```

1595
1596     Dim ShowLines() As Boolean = {graphsetting.GraphLine1, graphsetting.GraphLine2
, graphsetting.GraphLine3, graphsetting.GraphLine4}
1597
1598     Dim Pbsize As New System.Drawing.Point(Pbox.Size.Width, Pbox.Size.Height)
1599
1600     ' draw y line horizontal y= constant lines
1601
1602     If graphsetting.ShowHLines = True Then
1603         DrawHorizontalLines(k, graphsetting.horizontalLines, graphsetting.
HLinesDivisions, graphsetting.TypeofHline, zeropoint, graphsetting.PenofHVLines,
Pbox)
1604     End If
1605     'draw x = constant line vertical help lines
1606     drawvalues4lines(ShowLines(0), displayLineValue(0), ShowLines(1),
displayLineValue(1), ShowLines(2), displayLineValue(2), ShowLines(3),
displayLineValue(3), graphsetting.horizontalLines, HeadLegendUnits, Brushses, New
System.Drawing.Font("Arial", 10), Pbox, k)
1607
1608     If graphsetting.ShowTVLines = True Then
1609         DrawTimeVertLines(k, TimeMatrix, Index, graphsetting, graphsetting.
TimeIntervOfVertLine, Pbox)
1610     ElseIf graphsetting.showVlines = True Then
1611         DrawVerticalLines(k, graphsetting.Verticallines, zeropoint, graphsetting.
PenofHVLines, graphsetting.TypeofVline, Pbox)
1612     End If
1613     'convert data matrix to point for graph
1614     If Index > 2 Then
1615         'declare line dimation of line and maxvalue
1616         Dim maxvalueoffline As Integer
1617         Dim Line() As Array = {PWMvalue, load, Volt, Amp}
1618
1619         'draw lines
1620         For i As Integer = 0 To 3
1621             Select Case i
1622                 Case 0
1623                     maxvalueoffline = MaxValueOfLinesData(i)
1624                     'Line = PWMvalue
1625                     If graphsetting.GraphLine1 = True Then
1626                         Else
1627                             GoTo nextline
1628                         End If
1629                 Case 1
1630                     maxvalueoffline = MaxValueOfLinesData(i)
1631                     'Line = load
1632                     If graphsetting.GraphLine2 = True Then
1633                         Else
1634                             GoTo nextline
1635                         End If
1636                 Case 2
1637                     maxvalueoffline = MaxValueOfLinesData(i)
1638                     'Line = Volt
1639                     If graphsetting.GraphLine3 = True Then
1640                         Else
1641                             GoTo nextline
1642                         End If
1643                 Case 3
1644                     maxvalueoffline = MaxValueOfLinesData(i)
1645                     'Line = Amp
1646                     If graphsetting.GraphLine4 = True Then
1647                         Else
1648                             GoTo nextline
1649                         End If
1650             End Select
1651             ' draw y values help line
1652             ReDim pointmat(graphsetting.NofPointsofLine - 1, 1)
1653             drawgraphline(k, TimeMatrix, Line(i), maxvalueoffline, Index, pen(i),
graphsetting, Pbox, zeropoint)
1654
1655 nextline: Next i
1656
1657     Dim point As New System.Drawing.Point(0, Pbsize.Y - 20)
1658     Dim size As Size
1659     size.Width = Pbox.Width
1660     size.Height = 20

```

```

1661         displayLineLegend(k, LegendsofLines, Brusheses, ShowLines, graphsetting.
ShowLegenofLines, New System.Drawing.Font("courier new", 10), point, size)
1662     End If
1663
1664 End Sub
1665 Private Sub DrawTimeVertLines(ByVal e As Graphics, ByVal TimeMatrix() As UInt32,
ByVal Index As Integer, ByVal Graphset As GraphSettings, ByVal VertTimeInterval As
Integer, ByVal Pbox As PictureBox)
1666
1667     If VertTimeInterval >= 1 And Index > 3 Then
1668         Dim Maxpoints As Integer = Math.Min(Index - 1, Graphset.NofPointsofLine)
1669         Dim DrawIntervalTime As Integer = CLng(TimeMatrix(Index)) - CLng
(TimeMatrix(Index - Maxpoints))
1670         Dim NoofVerLines As Single = DrawIntervalTime / VertTimeInterval
1671         If NoofVerLines >= 100 And NoofVerLines > 0.5 Then '100 max number of
lines lines
1672             NoofVerLines = 100
1673             VertTimeInterval = Int(DrawIntervalTime / NoofVerLines)
1674         End If
1675         Dim MilsVsDrawPoints As Single = DrawIntervalTime / Pbox.Width
1676
1677         Dim Milseconds As Integer = ((TimeMatrix(Index) - Milseconds) Mod
VertTimeInterval)
1678
1679         Dim PointOffset As Single = IIf(MilsVsDrawPoints <> 0, Milseconds /
MilsVsDrawPoints, Pbox.Width)
1680         If NoofVerLines >= 0 Then '1
1681             Dim FirstLine As Integer = Pbox.Width - PointOffset
1682             For i As UInt32 = 0 To NoofVerLines
1683                 Dim Xvalue As Single = FirstLine - i * (VertTimeInterval /
MilsVsDrawPoints)
1684                 Select Case Graphset.TypeofTVLines
1685                     Case typeofline.DushedLine
1686                         DrawDashedLine(e, New System.Drawing.Point(CInt(Xvalue),
Pbox.Size.Height), New System.Drawing.Point(CInt(Xvalue), 0), Pens.Gray, Pbox)
1687                     Case typeofline.SolidLine
1688                         e.DrawLine(Pens.Gray, Xvalue, Pbox.Size.Height, Xvalue, 0)
1689                 End Select
1690             Next
1691         End If
1692     End If
1693 End Sub
1694
1695 Private Sub WriteValuesOfLineonPbox(ByVal e As Graphics, ByVal
NumberOfDisplayedValues As Int16, ByVal MaxDisplayedValues As Integer, ByVal
UnitLegend As String, ByVal zeropoint As System.Drawing.Point, ByVal font As
System.Drawing.Font, ByVal Brushes As Brush, ByVal Pbox As PictureBox)
1696
1697     If NumberOfDisplayedValues = 0 Then
1698         NumberOfDisplayedValues = 1
1699     End If
1700
1701     For i As Integer = 0 To NumberOfDisplayedValues
1702
1703         Dim Yvalue As Double = Pbox.Height - (i / NumberOfDisplayedValues) * (Pbox
.Height)
1704         Dim ypoint As New System.Drawing.Point(Pbox.Size.Width, Yvalue)
1705
1706         Select Case i
1707             Case 0
1708                 Dim text2display As String = "0"
1709                 e.DrawString(text2display, font, Brushes, zeropoint.X, ypoint.Y -
1710                 18)
1711             Case NumberOfDisplayedValues
1712                 Dim text2display As String = Format((i / NumberOfDisplayedValues)
* MaxDisplayedValues, "#####.#") & UnitLegend
1713                 e.DrawString(text2display, font, Brushes, zeropoint.X, ypoint.Y)
1714             Case Else
1715                 Dim text2display As String = Format((i / NumberOfDisplayedValues)
* MaxDisplayedValues, "#####.#")
1716                 e.DrawString(text2display, font, Brushes, zeropoint.X, ypoint.Y -
1717                 8)
1718         End Select
1719     End For
1720 End Sub

```

```

1719
1720     Next i
1721 End Sub
1722 Private Sub create2lines_graph(ByVal e As Graphics, ByVal TimeMatrix() As UInt32,
ByVal RpmMatrix() As Integer, ByVal WantedRpm() As Integer, ByVal showRpmgraph As
Boolean, ByVal PWMLoad() As Integer, ByVal WantedPWMLoad() As Integer, ByVal
graphsetting As GraphSettings, ByVal showPwmloadgraph As Boolean, ByVal Pbox As
PictureBox, ByVal Index As Integer)
1723
1724     If showRpmgraph = True Or showPwmloadgraph = True Then
1725         Dim k As Graphics = e
1726         Dim sz As New System.Drawing.Size
1727         Dim zeropoint As New System.Drawing.Point(0, Pbox.Size.Height - 0)
1728         Dim MaximumValues() As Single = {MaxStrofes, MaxControlAction}
1729         Dim pen(,) As Pen = {{pentolbox(0), pentolbox(1)}, {pentolbox(2),
pentolbox(3)}}
1730         Dim BrushRPM() As Brush = {Brushsestolbox(0), Brushsestolbox(1)}
1731         Dim BrushPWM() As Brush = {Brushsestolbox(2), Brushsestolbox(3)}
1732         Dim Brushed() As Array = {BrushRPM, BrushPWM}
1733         Dim MaxDisplayedValues() As Integer = {20000, 100}
1734         Dim UnitDisplay() As String = {" RPM", "% PWM"}
1735         Dim Pbsize As New System.Drawing.Point(Pbox.Size.Width, Pbox.Size.Height)
1736         Dim LegendRpm() As String = {Legendsoflbox(0), Legendsoflbox(1)}
1737         Dim LegendPWM() As String = {Legendsoflbox(2), Legendsoflbox(3)}
1738         Dim LegendString() As Array = {LegendRpm, LegendPWM}
1739         Dim showlegendofLines() As Boolean = {True, True}
1740         ' draw y line
1741
1742         k.DrawLine(Pens.Black, zeropoint.X, Pbox.Size.Height - zeropoint.Y, Pbox.
Size.Width, Pbox.Size.Height - zeropoint.Y)
1743         If showRpmgraph = True Then
1744             DrawHorizontalLines(k, graphsetting.horizontalLines, graphsetting.
HLinesDivisions, graphsetting.TypeofHline, zeropoint, graphsetting.PenofHVLines,
Pbox)
1745             WriteValuesOfLineonPbox(k, graphsetting.horizontalLines,
MaxDisplayedValues(0), UnitDisplay(0), zeropoint, New System.Drawing.Font("Arial",
10), Brushes.Black, Pbox)
1746         Else
1747             DrawHorizontalLines(k, graphsetting.horizontalLines, graphsetting.
HLinesDivisions, graphsetting.TypeofHline, zeropoint, graphsetting.PenofHVLines,
Pbox)
1748             WriteValuesOfLineonPbox(k, graphsetting.horizontalLines,
MaxDisplayedValues(1), UnitDisplay(1), zeropoint, New System.Drawing.Font("Arial",
10), Brushes.Black, Pbox)
1749         End If
1750         'draw x line
1751         If graphsetting.ShowTVLines = True Then
1752             DrawTimeVertLines(k, TimeMatrix, Index, graphsetting, graphsetting.
TimeIntervOfVertLine, Pbox) '10000
1753         ElseIf graphsetting.showVlines = True Then
1754             DrawVerticalLines(k, graphsetting.Verticallines, zeropoint,
graphsetting.PenofHVLines, graphsetting.TypeofVline, Pbox)
1755         End If
1756         'convert data matrix to point for graph
1757         Dim maxvalue As Integer
1758         Dim line1(,) As Array = {RpmMatrix, WantedRpm}, {PWMLoad, WantedPWMLoad}}
1759         Dim displayRpmLegend As Int16
1760         ReDim pointmat(graphsetting.NofPointsofLine - 1, 1)
1761         Dim witchcategorylines As Integer
1762         If showRpmgraph = True Then
1763             maxvalue = MaximumValues(0)
1764             witchcategorylines = 0
1765             displayRpmLegend = 0
1766         Else
1767             maxvalue = MaximumValues(1)
1768             witchcategorylines = 1
1769             displayRpmLegend = 1
1770         End If
1771         For i As Integer = 0 To 1
1772             drawgraphline(k, TimeMatrix, line1(witchcategorylines, i), maxvalue,
Index, pen(witchcategorylines, i), graphsetting, Pbox, zeropoint)
1773
1774     Next i
1775     Dim point As New System.Drawing.Point(0, Pbsize.Y - 20)
1776

```

```

1777         Dim size As Size
1778         size.Width = Pbox.Width
1779         size.Height = 20
1780         displayLineLegend(k, LegendsString(displayRpmLegend), Brushed
        (displayRpmLegend), showlegendofLines, graphsetting.ShowLegengofLines, New System.
        Drawing.Font("courier new", 10), point, size)
1781     End If
1782 End Sub
1783
1784 Private Sub PbGraph1Box1_Paint(ByVal sender As System.Object, ByVal e As System.
        Windows.Forms.PaintEventArgs) Handles PbGraph1Box1.Paint
1785     Try
1786         If Graphs = GraphicsDisplay.ShowGraphics Then
1787             If Not (-graph1settings.GraphLine1 - graph1settings.GraphLine2 -
        graph1settings.GraphLine3 - graph1settings.GraphLine4) = 2 Then
1788                 createGraphsfor2Box(e.Graphics, Tmatrix, Rmatrix, WRmatrix, gkazi,
        Wantedgkazi, MaxValueOfLinesData1box, displayLineValue1box, Unitsdisplayfor1box,
        Legendsof1box, graph1settings, pentolbox, Brushsestolbox, PbGraph1Box1,
        Index_to_Point_of_Graph1) 'ArraysMet - 1
1789             ElseIf (graph1settings.GraphLine1 = True And graph1settings.GraphLine2
        = True) Or (graph1settings.GraphLine3 = True And graph1settings.GraphLine4 =
        True) Then
1790                 If (graph1settings.GraphLine1 = True) Then
1791                     showRpm = True
1792                     showPwmLoad = False
1793                 Else
1794                     showRpm = False
1795                     showPwmLoad = True
1796                 End If
1797                 create2lines_graph(e.Graphics, Tmatrix, Rmatrix, WRmatrix, showRpm
        , gkazi, Wantedgkazi, graph1settings, showPwmLoad, PbGraph1Box1,
        Index_to_Point_of_Graph1) 'showRpm, showPwmLoad
1798             Else
1799                 createGraphsfor2Box(e.Graphics, Tmatrix, Rmatrix, WRmatrix, gkazi,
        Wantedgkazi, MaxValueOfLinesData1box, displayLineValue1box, Unitsdisplayfor1box,
        Legendsof1box, graph1settings, pentolbox, Brushsestolbox, PbGraph1Box1,
        Index_to_Point_of_Graph1) ' ArraysMet - 1
1800             End If
1801         End If
1802     Catch
1803     Stop
1804     End Try
1805 End Sub
1806
1807 Private Sub PbGraph2Box1_Paint(ByVal sender As System.Object, ByVal e As System.
        Windows.Forms.PaintEventArgs) Handles PbGraph2Box1.Paint
1808     Try
1809         If Graphs = GraphicsDisplay.ShowGraphics Then
1810             createGraphsfor2Box(e.Graphics, Tmatrix, gkazi, Loadcell, Voltage,
        Ampere, MaxValueOfLinesData2box, displayLineValue2box, Unitsdisplayfor2box,
        Legendsof2box, graph2settings, pento2box, Brushsesto2box, PbGraph2Box1,
        Index_to_Point_of_Graph2) ' ArraysMet - 1
1811         End If
1812     Catch
1813     Stop
1814     End Try
1815 End Sub
1816
1817 Private Sub PbGraph1Box1_MouseLeave(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles PbGraph1Box1.MouseLeave
1818     graph1settings.ShowLegengofLines = False
1819 End Sub
1820
1821 Private Sub PbGraph1Box1_MouseEnter(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles PbGraph1Box1.MouseEnter
1822     graph1settings.ShowLegengofLines = True
1823 End Sub
1824
1825 Private Sub PbGraph2Box1_MouseLeave(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles PbGraph2Box1.MouseLeave
1826     graph2settings.ShowLegengofLines = False
1827 End Sub
1828
1829 Private Sub PbGraph2Box1_MouseEnter(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles PbGraph2Box1.MouseEnter

```

```

1830     graph2settings.ShowLegengofLines = True
1831 End Sub
1832
1833 Private Sub loaddatatoworker(ByVal index As Int32, ByVal time() As UInt32, ByVal
1834 currentRpm() As Integer, ByVal wantedRpm() As Integer, ByVal Gkazi() As Integer
, ByVal wantedgkazi() As Integer, ByVal loadcell() As
Integer, ByVal voltage() As Integer, ByVal ampere() As Integer, ByVal
chart1setting As ChartSettings,
ByVal chart2setting As ChartSettings)
1835
1836
1837     ReDim matforworker(Math.Max(Me.Chart1Settings.NoPointstodisplay, Me.
Chart2Settings.NoPointstodisplay), 4)
1838
1839     Dim chart1show() As Boolean = {chart1setting.Graphrpm, chart1setting.
Graphgkazi, chart1setting.Graphloadcell,
chart1setting.Graphvoltage, chart1setting.Graphampere, chart1setting.Graphwrpm
, chart1setting.Graphwgkazi}
1840
1841
1842
1843     Dim chart2show() As Boolean = {chart2setting.Graphrpm, chart2setting.
Graphgkazi, chart2setting.Graphloadcell,
chart2setting.Graphvoltage, chart2setting.Graphampere, chart2setting.Graphwrpm
, chart2setting.Graphwgkazi}
1844
1845
1846     Dim linearray() As Array = {currentRpm, Gkazi, loadcell, voltage, ampere,
wantedRpm, wantedgkazi}
1847     Dim line1(), line2(), line3() As Int32
1848     Dim linefilled As Boolean = False
1849
1850     For i As Int16 = 0 To 6
1851         If chart1show(i) = True Then
1852             If linefilled = False Then
1853                 line1 = linearray(i)
1854                 matforworker(0, 1) = i
1855                 linefilled = True
1856             Else
1857                 line3 = linearray(i)
1858                 matforworker(0, 3) = i
1859             End If
1860
1861         End If
1862         If chart2show(i) = True Then
1863             line2 = linearray(i)
1864             matforworker(0, 2) = i
1865         End If
1866     Next i
1867     Dim maxpoints As Integer = Math.Max(Me.Chart1Settings.NoPointstodisplay, Me.
Chart2Settings.NoPointstodisplay)
1868
1869     For i As Int16 = 1 To maxpoints
1870         Try
1871             matforworker(i, 0) = time(index - maxpoints - 1 + i) 'arraymet -1=
index
1872             matforworker(i, 1) = line1(index - maxpoints - 1 + i) ' for chart1
1873             matforworker(i, 2) = line2(index - maxpoints - 1 + i) ' for chart2
1874             matforworker(i, 3) = line3(index - maxpoints - 1 + i) ' for chart1
1875         Catch e As Exception
1876         End Try
1877     Next i
1878
1879 End Sub
1880
1881 Private Sub MakeExcelChart(ByVal n(,) As Int64, ByVal worker As BackgroundWorker,
ByVal e As DoWorkEventArgs) ' As Image
1882     worker1completed = False
1883     CreateGraphs(n)
1884     If ChartCreationError = False Then
1885         DisplayChartatPbox()
1886     End If
1887 End Sub
1888
1889 Private Sub MyTimer_Tick(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MyTimer.Tick
1890     Try
1891         If MnDisconnect.Enabled And FlagStartSend And MySerialPort.IsOpen = True

```

```

1892     Then
1893         '----- timer at status bar -----
1894         -----
1895         'testtime()
1896         MyClassCode.Check_if_Timeout_Serialport(5000, WhichIsTheLang)
1897         TimeofConnection()
1898         If (Me.IARMotor <> Me.VBMotor) = True Or (VBMotor = MotorCondition.
MotorStopped And RPM >= 1) = True Then
1899
1900             Select Case Me.VBMotor
1901                 Case MotorCondition.MotorRunning
1902                     Me.InitiateMotorFromStopState()
1903                 Case MotorCondition.MotorStopped
1904                     Me.SendStopMotorSignal()
1905             End Select
1906
1907         End If
1908
1909         Select Case Me.IARSetPointRoot
1910             Case SetpointsRoot.File
1911                 If MySerialPort.BytesToWrite < 50 Then
1912                     MySerialPort.Write(MyClassCode.SecureTransmit(20001), 0,
6)
1913
1914                     End If
1915                 Case SetpointsRoot.FormControls
1916                     If Me.IARmode = VBstatus Then
1917                         MySerialPort.Write(MyClassCode.SecureTransmit(Val
6)
1918                         (TrBrOutStrofes.Value())), 0, 6)
1919
1920                     Else
1921                         MySerialPort.Write(MyClassCode.SecureTransmit(20001), 0,
6)
1922
1923                     End If
1924             End Select
1925         ElseIf MnDisconnect.Enabled And FlagStartSend = False Then
1926
1927             End If
1928             ServeClose_Conn_Form()
1929         Catch ex As Exception
1930             End Try
1931     End Sub
1932
1933
1934     Private Sub BackgroundWorker1_DoWork(ByVal sender As System.Object, ByVal e As
System.ComponentModel.DoWorkEventArgs) Handles BackgroundWorker1.DoWork
1935         If Not app Is Nothing Then
1936             Dim worker As BackgroundWorker = CType(sender, BackgroundWorker)
1937             MakeExcelChart(e.Argument, worker, e)
1938         End If
1939     End Sub
1940
1941     Private Sub BackgroundWorker1_ProgressChanged(ByVal sender As System.Object, ByVal
e As System.ComponentModel.ProgressChangedEventArgs) Handles BackgroundWorker1.
ProgressChanged
1942     End Sub
1943
1944     Private Sub BackgroundWorker1_RunWorkerCompleted(ByVal sender As System.Object,
ByVal e As System.ComponentModel.RunWorkerCompletedEventArgs) Handles
BackgroundWorker1.RunWorkerCompleted
1945         Try
1946             If (e.Error IsNot Nothing) Then
1947                 Stop
1948                 MessageBox.Show(e.Error.Message)
1949                 Stop
1950                 ChartCreationError = False 'reset flag ChartCreationError
1951             ElseIf e.Cancelled Then
1952                 ' Next, handle the case where the user canceled the
1953                 ' operation.
1954                 ' Note that due to a race condition in
1955                 ' the DoWork event handler, the Cancelled

```

```

1956         '' flag may not have been set, even though
1957         '' CancelAsync was called.
1958
1959         ChartCreationError = False 'reset flag ChartCreationError
1960     Else
1961         PBGraph1.Refresh()
1962         PBGraph2.Refresh()
1963
1964         ChartCreationError = False
1965     End If
1966
1967     worker1completed = True ' reset worker1completed for next run
1968     GraphRunFirstTime = False
1969 Catch
1970     worker1completed = True
1971 End Try
1972
1973 End Sub
1974
1975 Private Sub createbargrphs(ByVal PWM As Integer, ByVal loadcell As Integer, ByVal voltage As Integer, ByVal ampere As Integer, ByVal pbox As PictureBox, ByVal fonts As System.Drawing.Font, ByVal barsets As barsettings, ByVal e As Graphics)
1976     Dim size As Size
1977     Dim heightbars() As Integer = {PWM, loadcell, voltage, ampere}
1978     Dim legends() As String = {"PWM", " Cell", " Volt", " Amp"}
1979     size.Width = pbox.Width
1980     size.Height = pbox.Height - 20
1981     Dim distancebetweenbars As Integer = (size.Width - 4 * (3 * Font.SizeInPoints + barsets.width)) / 4
1982     Dim distancebetweenvals As Integer = size.Height / barsets.NofVal
1983     For i As Integer = 0 To 3
1984         Dim point As New System.Drawing.Point(i * (barsets.width + 3 * Font.SizeInPoints + distancebetweenbars), pbox.Height - ((heightbars(i) / MaxValues(i)) * size.Height))
1985         'draw vertival lines and values
1986         e.DrawString(legends(i), New System.Drawing.Font("Arial", 10), Brushes.Black, New System.Drawing.Point(point.X, 0))
1987         For k As Integer = 0 To 10
1988             Dim pointD As New System.Drawing.Point(point.X, pbox.Height - (k * size.Height / 10))
1989             Dim text As String = Format((k * MaxDispValue(i) / 10), "#0.#")
1990             Dim correcttextpos As Integer = 3 * Font.SizeInPoints - text.Length * Font.SizeInPoints
1991             Select Case k
1992                 Case 0
1993                     e.DrawString(Format((k * MaxDispValue(i) / 10), "#0.#"), Font, Brushes.Black, pointD.X + correcttextpos, pointD.Y - 2 * Font.SizeInPoints)
1994                     e.DrawLine(Pens.Black, pointD.X + 3 * Font.SizeInPoints, pointD.Y - 1, pointD.X + 4 * Font.SizeInPoints, pointD.Y - 1)
1995                 Case 5
1996                     e.DrawString(Format((k * MaxDispValue(i) / 10), "#0.#"), Font, Brushes.Black, pointD.X + correcttextpos, pointD.Y - 5)
1997                     e.DrawLine(Pens.Black, pointD.X + 3 * Font.SizeInPoints, pointD.Y, pointD.X + 4 * Font.SizeInPoints, pointD.Y)
1998                 Case 10
1999                     e.DrawString(Format((k * MaxDispValue(i) / 10), "#0.#"), Font, Brushes.Black, pointD.X + correcttextpos, pointD.Y - 5)
2000                     e.DrawLine(Pens.Black, pointD.X + 3 * Font.SizeInPoints, pointD.Y, pointD.X + 4 * Font.SizeInPoints, pointD.Y)
2001                 Case Else
2002                     e.DrawLine(Pens.Black, pointD.X + 3 * Font.SizeInPoints, pointD.Y, pointD.X + 4 * Font.SizeInPoints, pointD.Y)
2003             End Select
2004
2005         Next
2006
2007         Dim rect As New System.Drawing.Rectangle(4 * Font.SizeInPoints + point.X, point.Y, barsets.width, pbox.Height - point.Y)
2008         e.DrawRectangle(Pens.Blue, rect)
2009         e.FillRectangle(barsets.brushes(i), rect)
2010     Next
2011
2012 End Sub
2013
2014

```

```

2015 Private Sub PbBars_Paint(ByVal sender As System.Object, ByVal e As System.Windows.
Forms.PaintEventArgs) Handles PbBars.Paint
2016     Try
2017         createbargrphs((pwm - PWM_Min) * 1000 / PWM_Duty, cell, volt, amp, PbBars,
New System.Drawing.Font("Arial", 10), Barset, e.Graphics)
2018     Catch
2019     End Try
2020 End Sub
2021 Private Sub drawleds(ByVal e As System.Drawing.Graphics, ByVal pbbox As PictureBox
, ByVal ledsstatus() As Boolean, ByVal ledsetttting As LedSettings)
2022     Dim xcenter As Integer = pbbox.Width / 2
2023     Dim ycenter As Integer = pbbox.Height / 2
2024
2025     Dim ypoint As New System.Drawing.Point
2026     For i As Integer = 0 To 7
2027         Dim point As New System.Drawing.Point((i * ((pbbox.Width - ledsetttting.
diameter) / 7)), ycenter - (ledsetttting.diameter / 2))
2028         If i = 7 Then
2029             point.X = point.X
2030         End If
2031         ledstatus(e, ledsstatus(i), pbbox, ledsetttting, point)
2032     Next
2033 End Sub
2034 Private Sub ledstatus(ByVal e As System.Drawing.Graphics, ByVal onoff As Boolean,
ByVal Pbbox As PictureBox, ByVal ledsetttting As LedSettings, ByVal Point2Display
As System.Drawing.Point)
2035     e.DrawPie(ledsetttting.pen(0), Point2Display.X, Point2Display.Y, ledsetttting.
diameter, ledsetttting.diameter, 0, 360)
2036     Select Case onoff
2037     Case False
2038         e.FillPie(ledsetttting.Brush(0), Point2Display.X, Point2Display.Y,
ledsetttting.diameter, ledsetttting.diameter, 0, 360)
2039     Case True
2040         e.FillPie(ledsetttting.Brush(1), Point2Display.X, Point2Display.Y,
ledsetttting.diameter, ledsetttting.diameter, 0, 360)
2041     End Select
2042
2043 End Sub
2044
2045 Private Sub Pbleds_Paint(ByVal sender As System.Object, ByVal e As System.Windows.
Forms.PaintEventArgs) Handles Pbleds.Paint
2046     drawleds(e.Graphics, Pbleds, Errors, Ledset)
2047 End Sub
2048
2049 Private Sub Ledonoff(ByVal e As System.Drawing.Graphics, ByVal onoff As Boolean,
ByVal pbbox As PictureBox, ByVal ledsetttting As LedSettings)
2050     Dim Point2Display As New System.Drawing.Point(pbbox.Width / 2 - ledsetttting.
diameter / 2, pbbox.Height / 2 - ledsetttting.diameter / 2)
2051     e.DrawPie(ledsetttting.pen(0), Point2Display.X, Point2Display.Y, ledsetttting.
diameter, ledsetttting.diameter, 0, 360)
2052     e.FillPie(Brushes.Black, Point2Display.X - 2, Point2Display.Y - 2, ledsetttting
.diameter + 4, ledsetttting.diameter + 4, 0, 360)
2053     Select Case onoff
2054     Case False
2055         e.FillPie(ledsetttting.Brush(0), Point2Display.X, Point2Display.Y,
ledsetttting.diameter, ledsetttting.diameter, 0, 360)
2056     Case True
2057         e.FillPie(ledsetttting.Brush(1), Point2Display.X, Point2Display.Y,
ledsetttting.diameter, ledsetttting.diameter, 0, 360)
2058     End Select
2059 End Sub
2060
2061 Private Sub PbLed1_Paint(ByVal sender As System.Object, ByVal e As System.Windows.
Forms.PaintEventArgs) Handles PbLed1.Paint
2062     Ledonoff(e.Graphics, True, PbLed1, Ledset)
2063 End Sub
2064
2065 Private Sub TimeofConnection()
2066
2067     MyClassCode.CountTimeOfConnection(New System.DateTime(Now.Ticks))
2068     Me.StBTime.Text = MyClassCode.stTimeConnection
2069
2070
2071 End Sub
2072

```



```

2073 Private Sub MnShowGraphs_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MnShowGraphs.Click
2074     BtShowGraphs_Click(sender, e)
2075 End Sub
2076 '
2077 Private Sub MnHideGraphs_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MnHideGraphs.Click
2078     BtShowGraphs_Click(sender, e)
2079 End Sub
2080
2081 Private Sub DisplayGraphics()
2082     If (Graphs = GraphicsDisplay.showChartGraphics And NoofData > 150) And
PbShowAllGraphs Then
2083         NoofData = 0
2084
2085         If ArraysMet > Math.Max(Chart1Settings.NoPointstodisplay, Chart2Settings.
NoPointstodisplay) And Graphs = GraphicsDisplay.showChartGraphics Then
2086             If BackgroundWorker1.IsBusy = False And (worker1completed = True) Then
2087                 loadatatoworker(ArraysMet - 1, Tmatrix, Rmatrix, WRmatrix, gkazi,
Wantedgkazi, Loadcell, Voltage, Ampere, Chart1Settings, Chart2Settings)
2088                 BackgroundWorker1.RunWorkerAsync(matforworker) ' init
backgroundworker
2089             Else
2090
2091                 End If
2092             End If
2093         ElseIf Graphs = GraphicsDisplay.ShowGraphics Then
2094
2095             PbGraph1Box1.Refresh()
2096             PbGraph2Box1.Refresh()
2097         End If
2098     End Sub
2099
2100 Private Sub DisplayDefaultGraphics()
2101     PbBars.Refresh()
2102     PbRpmMeter1.Refresh()
2103     Pbleds.Refresh()
2104 End Sub
2105
2106 Private Sub MnShowNETGraphs_Click(ByVal sender As System.Object, ByVal e As System
.EventArgs) Handles MnShowNETGraphs.Click
2107     If MnShowNETGraphs.Checked = False Then
2108         MnShowNETGraphs.Checked = True
2109         MnShowExcelGraphs.Checked = False
2110         Graphs = IIf(MnShowNETGraphs.Checked, GraphicsDisplay.ShowGraphics,
GraphicsDisplay.showChartGraphics)
2111         MnNetParams.Enabled = True
2112         MnChartProperties.Enabled = False
2113         PbGraph1Box1.BringToFront()
2114         PbGraph2Box1.BringToFront()
2115
2116         If BackgroundWorker1.IsBusy Then
2117             BackgroundWorker1.CancelAsync()
2118         End If
2119
2120     If Me.Height - Me.PbGraph1Box1.Height - Me.PbGraph1Box1.Location.Y > 0
Then
2121         PbGraph1Box1.Visible = True
2122         PbGraph2Box1.Visible = True
2123     End If
2124
2125     PBGraph1.Visible = False
2126     PBGraph2.Visible = False
2127     PanelChartProperties.Visible = False
2128 End If
2129 End Sub
2130
2131 Private Sub MnShowExcelGraphs_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MnShowExcelGraphs.Click
2132     If MnShowExcelGraphs.Checked = False Then
2133         UncheckNETGrahSProps()
2134         MnShowNETGraphs.Checked = False
2135         MnShowExcelGraphs.Checked = True
2136         Graphs = IIf(MnShowNETGraphs.Checked, GraphicsDisplay.ShowGraphics,
GraphicsDisplay.showChartGraphics)

```

```

2137         MnNetParams.Enabled = False
2138         MnChartProperties.Enabled = True
2139
2140         If Me.Height - Me.PBGraph1.Height - Me.PBGraph1.Location.Y > 0 Then
2141             PBGraph1.Visible = True
2142             PBGraph2.Visible = True
2143         End If
2144
2145         PbGraph1Box1.Visible = False
2146         PbGraph2Box1.Visible = False
2147         PBGraph1.BringToFront()
2148         PBGraph2.BringToFront()
2149         Me.PanelPropertiesGraph1.Visible = False
2150         Me.PanelPropertiesGraphs2.Visible = False
2151     End If
2152 End Sub
2153
2154 Public Sub EnableMenu(ByVal IsEnabled As Boolean)
2155     MenuItem1.Enabled = IsEnabled
2156     MITefarmoges.Enabled = IsEnabled
2157     MenuItem6.Enabled = IsEnabled
2158 End Sub
2159
2160 Private Sub NUDNetC1NumOfPoints_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NUDNetC1NumOfPoints.ValueChanged
2161     graph1settings.NofPointsofLine = NUDNetC1NumOfPoints.Value
2162 End Sub
2163
2164 Private Sub PbGraph1Box1_MouseClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.MouseEventArgs) Handles PbGraph1Box1.MouseClick
2165 End Sub
2166
2167 Private Sub PanelPropertiesGraph1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PanelPropertiesGraph1.Click
2168     PanelPropertiesGraph1.Visible = False
2169     UncheckNETGrahSProps()
2170
2171 End Sub
2172
2173
2174 Private Sub NUDNetC1HorLines_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NUDNetC1HorLines.ValueChanged
2175     If NUDNetC1HorLines.Value = 0 Then
2176         graph1settings.ShowHLines = False
2177     Else
2178         graph1settings.ShowHLines = True
2179         graph1settings.horizontalLines = NUDNetC1HorLines.Value
2180     End If
2181 End Sub
2182
2183
2184 Private Sub NUDNetC1ConstVerLines_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NUDNetC1ConstVerLines.ValueChanged
2185     If NUDNetC1ConstVerLines.Value = 0 Then
2186         graph1settings.showVlines = False
2187     Else
2188         graph1settings.showVlines = True
2189         graph1settings.Verticallines = NUDNetC1ConstVerLines.Value
2190     End If
2191 End Sub
2192
2193 Private Sub ChkNetC1Rpm_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ChkNetC1Rpm.CheckedChanged
2194     graph1settings.GraphLine1 = ChkNetC1Rpm.Checked
2195 End Sub
2196
2197
2198 Private Sub ChkNetC1SetPointRPM_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ChkNetC1SetPointRPM.CheckedChanged
2199     graph1settings.GraphLine2 = ChkNetC1SetPointRPM.Checked
2200 End Sub
2201
2202
2203
2204

```

```

2205 Private Sub ChkNetC1PWMLoad_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ChkNetC1PWMLoad.CheckedChanged
2206
2207     graph1settings.GraphLine3 = ChkNetC1PWMLoad.Checked
2208
2209 End Sub
2210
2211 Private Sub ChkNetC1SetPointPwm_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ChkNetC1SetPointPwm.CheckedChanged
2212     graph1settings.GraphLine4 = ChkNetC1SetPointPwm.Checked
2213 End Sub
2214
2215 Private Sub VerticalLines1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles LblVerticalLines1.Click
2216
2217 End Sub
2218
2219 Private Sub NUD1NetC2NumOfPoints_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NUD1NetC2NumOfPoints.ValueChanged
2220     graph2settings.NofPointsofLine = NUD1NetC2NumOfPoints.Value
2221 End Sub
2222
2223 Private Sub NUDNetC2HorLines_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NUDNetC2HorLines.ValueChanged
2224     graph2settings.horizontalLines = NUDNetC2HorLines.Value
2225 End Sub
2226
2227 Private Sub NUDNetC2ConstVerLines_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NUDNetC2ConstVerLines.ValueChanged
2228     ' const verticals
2229     If NUDNetC2ConstVerLines.Value = 0 Then
2230         graph2settings.showVlines = False
2231     Else
2232         graph2settings.showVlines = True
2233         graph2settings.Verticallines = NUDNetC2ConstVerLines.Value
2234     End If
2235 End Sub
2236
2237 Private Sub ChkNetC2PWMLoad_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ChkNetC2PWMLoad.CheckedChanged
2238     graph2settings.GraphLine1 = ChkNetC2PWMLoad.Checked
2239 End Sub
2240
2241 Private Sub ChkNetC2LoadCell1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ChkNetC2LoadCell.CheckedChanged
2242     graph2settings.GraphLine2 = ChkNetC2LoadCell.Checked
2243 End Sub
2244
2245 Private Sub ChkNetC2Voltage_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ChkNetC2Voltage.CheckedChanged
2246     graph2settings.GraphLine3 = ChkNetC2Voltage.Checked
2247 End Sub
2248
2249 Private Sub ChkNetC2Ampere_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ChkNetC2Ampere.CheckedChanged
2250     graph2settings.GraphLine4 = ChkNetC2Ampere.Checked
2251 End Sub
2252
2253 Private Sub MnPropertiesGraph2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
2254     PanelPropertiesGraphs2.Visible = True
2255     PanelPropertiesGraph1.Visible = True
2256     PanelChartProperties.Visible = False
2257     PanelPropertiesGraphs2.BringToFront()
2258 End Sub
2259
2260 Private Sub MnGraph1Prop_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
2261     PanelPropertiesGraphs2.Visible = False
2262     PanelPropertiesGraph1.Visible = True
2263     PanelChartProperties.Visible = False
2264     PanelPropertiesGraph1.BringToFront()
2265 End Sub
2266
2267 Private Sub MnGraph2Prop_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

```

```

EventArgs)
2268     PanelPropertiesGraphs2.Visible = True
2269     PanelPropertiesGraph1.Visible = False
2270     PanelChartProperties.Visible = False
2271
2272 End Sub
2273
2274 Private Sub PanelPropertiesGraphs2_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles PanelPropertiesGraphs2.Click
2275     PanelPropertiesGraphs2.Visible = False
2276     UnCheckNETGrahSProps()
2277 End Sub
2278
2279 Private Sub NUD1NetC2Hor1Div_ValueChanged(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles NUD1NetC2Hor1Div.ValueChanged
2280     graph2settings.HLinesDivisions = NUD1NetC2Hor1Div.Value
2281 End Sub
2282
2283 Private Sub RbSolidLines2_CheckedChanged(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles RbSolidLines2.CheckedChanged
2284     If RbSolidLines2.Checked = True Then
2285         graph2settings.TypeofVline = typeofline.SolidLine
2286         graph2settings.TypeofHline = typeofline.SolidLine
2287     End If
2288
2289 End Sub
2290
2291 Private Sub RbDushedLines2_CheckedChanged(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles RbDushedLines2.CheckedChanged
2293     If RbDushedLines2.Checked = True Then
2294         graph2settings.TypeofVline = typeofline.DushedLine
2295         graph2settings.TypeofHline = typeofline.DushedLine
2296     End If
2297 End Sub
2298
2299 Private Sub PanelPropertiesGraphs2_Paint(ByVal sender As System.Object, ByVal e As
    System.Windows.Forms.PaintEventArgs) Handles PanelPropertiesGraphs2.Paint
2300
2301     e.Graphics.DrawString("2 Graph ", New System.Drawing.Font("Arial", 40),
    Brushes.Silver, New System.Drawing.Point(Me.PanelPropertiesGraphs2.Width - 280,
    (Me.PanelPropertiesGraphs2.Height - 60) / 2))
2302     e.Graphics.DrawRectangle(New System.Drawing.Pen(Color.Chocolate, 2), New
    System.Drawing.Rectangle(1, 1, PanelPropertiesGraphs2.Width - 2,
    PanelPropertiesGraphs2.Height - 2))
2303 End Sub
2304
2305 Private Sub DUDNetC1TypeLines_SelectedItemChanged(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles DUDNetC1TypeLines.SelectedItemChanged
2306
2307     If DUDNetC1TypeLines.SelectedIndex = 1 Then
2308         graph1settings.TypeofHline = typeofline.SolidLine
2309         graph1settings.TypeofVline = typeofline.SolidLine
2310         graph1settings.TypeofTVLines = typeofline.SolidLine
2311     ElseIf DUDNetC1TypeLines.SelectedIndex = 0 Then
2312         graph1settings.TypeofHline = typeofline.DushedLine
2313         graph1settings.TypeofVline = typeofline.DushedLine
2314         graph1settings.TypeofTVLines = typeofline.DushedLine
2315     End If
2316
2317 End Sub
2318
2319 Private Sub NUDNetC1Hor1Div_ValueChanged(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles NUDNetC1Hor1Div.ValueChanged
2321     graph1settings.HLinesDivisions = NUDNetC1Hor1Div.Value
2322 End Sub
2323
2324 Private Sub Set_Size_and_Location_Panel()
2325
2326 End Sub
2327
2328 Private Sub MnPropertiesGraph1_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs)
2329     PanelPropertiesGraphs2.Visible = False

```

```

2330     PanelPropertiesGraph1.Visible = True
2331     PanelChartProperties.Visible = False
2332 End Sub
2333
2334 Private Sub MnPropertiesGraph2_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs)
2335     PanelPropertiesGraphs2.Visible = True
2336     PanelPropertiesGraph1.Visible = False
2337     PanelChartProperties.Visible = False
2338 End Sub
2339
2340 Private Sub ContextMenuStrip1_Opening(ByVal sender As System.Object, ByVal e As System.ComponentModel.CancelEventArgs)
2341
2342 End Sub
2343
2344 Private Sub NUDExcelC1ChartPoints_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NUDExcelC1ChartPoints.ValueChanged
2345     Chart1Settings.NoPointstodisplay = NUDExcelC1ChartPoints.Value
2346 End Sub
2347
2348 Private Sub NUDExcelC2ChartPoints_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NUDExcelC2ChartPoints.ValueChanged
2349     Chart2Settings.NoPointstodisplay = NUDExcelC2ChartPoints.Value
2350 End Sub
2351
2352 Private Sub DUDExcelC1SelectLine_SelectedItemChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DUDExcelC1SelectLine.SelectedItemChanged
2353     Select Case DUDExcelC1SelectLine.SelectedItem
2354     Case "RPM"
2355         Chart1Settings.Graphrpm = True
2356         Chart1Settings.Graphgkazi = False
2357         Chart1Settings.Graphvoltage = False
2358         Chart1Settings.Graphloadcell = False
2359         Chart1Settings.Graphampere = False
2360         Chart1Settings.Graphwgkazi = False
2361         Chart1Settings.Graphwrpm = False
2362     Case "PWM Load"
2363         Chart1Settings.Graphrpm = False
2364         Chart1Settings.Graphgkazi = True
2365         Chart1Settings.Graphvoltage = False
2366         Chart1Settings.Graphloadcell = False
2367         Chart1Settings.Graphampere = False
2368         Chart1Settings.Graphwgkazi = False
2369         Chart1Settings.Graphwrpm = False
2370     Case "Load Cell"
2371         Chart1Settings.Graphrpm = False
2372         Chart1Settings.Graphgkazi = False
2373         Chart1Settings.Graphvoltage = False
2374         Chart1Settings.Graphloadcell = True
2375         Chart1Settings.Graphampere = False
2376         Chart1Settings.Graphwgkazi = False
2377         Chart1Settings.Graphwrpm = False
2378     Case "Amperes"
2379         Chart1Settings.Graphrpm = False
2380         Chart1Settings.Graphgkazi = False
2381         Chart1Settings.Graphvoltage = False
2382         Chart1Settings.Graphloadcell = False
2383         Chart1Settings.Graphampere = True
2384         Chart1Settings.Graphwgkazi = False
2385         Chart1Settings.Graphwrpm = False
2386     Case "Voltage"
2387         Chart1Settings.Graphrpm = False
2388         Chart1Settings.Graphgkazi = False
2389         Chart1Settings.Graphvoltage = True
2390         Chart1Settings.Graphloadcell = False
2391         Chart1Settings.Graphampere = False
2392         Chart1Settings.Graphwgkazi = False
2393         Chart1Settings.Graphwrpm = False
2394     Case "Setpoint RPM"
2395         Chart1Settings.Graphrpm = False
2396         Chart1Settings.Graphgkazi = False
2397         Chart1Settings.Graphvoltage = False
2398         Chart1Settings.Graphloadcell = False
2399         Chart1Settings.Graphampere = False

```

```

2400         Chart1Settings.Graphwgkazi = False
2401         Chart1Settings.Graphwrpm = True
2402     Case "Setpoint PWM"
2403         Chart1Settings.Graphrpm = False
2404         Chart1Settings.Graphgkazi = False
2405         Chart1Settings.Graphvoltage = False
2406         Chart1Settings.Graphloadcell = False
2407         Chart1Settings.Graphampere = False
2408         Chart1Settings.Graphwgkazi = True
2409         Chart1Settings.Graphwrpm = False
2410     Case Else
2411 End Select
2412
2413 End Sub
2414
2415 Private Sub DUDExcelC2SelectLine_SelectedItemChanged(ByVal sender As System.Object
, ByVal e As System.EventArgs) Handles DUDExcelC2SelectLine.SelectedItemChanged
2416     Select Case DUDExcelC2SelectLine.SelectedItem
2417     Case "RPM"
2418         Chart2Settings.Graphrpm = True
2419         Chart2Settings.Graphgkazi = False
2420         Chart2Settings.Graphvoltage = False
2421         Chart2Settings.Graphloadcell = False
2422         Chart2Settings.Graphampere = False
2423         Chart2Settings.Graphwgkazi = False
2424         Chart2Settings.Graphwrpm = False
2425     Case "PWM Load"
2426         Chart2Settings.Graphrpm = False
2427         Chart2Settings.Graphgkazi = True
2428         Chart2Settings.Graphvoltage = False
2429         Chart2Settings.Graphloadcell = False
2430         Chart2Settings.Graphampere = False
2431         Chart2Settings.Graphwgkazi = False
2432         Chart2Settings.Graphwrpm = False
2433     Case "Load Cell"
2434         Chart2Settings.Graphrpm = False
2435         Chart2Settings.Graphgkazi = False
2436         Chart2Settings.Graphvoltage = False
2437         Chart2Settings.Graphloadcell = True
2438         Chart2Settings.Graphampere = False
2439         Chart2Settings.Graphwgkazi = False
2440         Chart2Settings.Graphwrpm = False
2441     Case "Amperes"
2442         Chart2Settings.Graphrpm = False
2443         Chart2Settings.Graphgkazi = False
2444         Chart2Settings.Graphvoltage = False
2445         Chart2Settings.Graphloadcell = False
2446         Chart2Settings.Graphampere = True
2447         Chart2Settings.Graphwgkazi = False
2448         Chart2Settings.Graphwrpm = False
2449     Case "Voltage"
2450         Chart2Settings.Graphrpm = False
2451         Chart2Settings.Graphgkazi = False
2452         Chart2Settings.Graphvoltage = True
2453         Chart2Settings.Graphloadcell = False
2454         Chart2Settings.Graphampere = False
2455         Chart2Settings.Graphwgkazi = False
2456         Chart2Settings.Graphwrpm = False
2457     Case "Setpoint RPM"
2458         Chart2Settings.Graphrpm = False
2459         Chart2Settings.Graphgkazi = False
2460         Chart2Settings.Graphvoltage = False
2461         Chart2Settings.Graphloadcell = False
2462         Chart2Settings.Graphampere = False
2463         Chart2Settings.Graphwgkazi = False
2464         Chart2Settings.Graphwrpm = True
2465     Case "Setpoint PWM"
2466         Chart2Settings.Graphrpm = False
2467         Chart2Settings.Graphgkazi = False
2468         Chart2Settings.Graphvoltage = False
2469         Chart2Settings.Graphloadcell = False
2470         Chart2Settings.Graphampere = False
2471         Chart2Settings.Graphwgkazi = True
2472         Chart2Settings.Graphwrpm = False
2473     Case Else

```

```

2474         End Select
2475     End Sub
2476
2477     Private Sub GbChartProperties2_Enter(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles GbChartProperties2.Enter
2478
2479     End Sub
2480
2481
2482     Private Sub MnChartProperties_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
2483         PanelChartProperties.BringToFront()
2484         ShowApropiateControl(PanelChartProperties.Name.ToString)
2485     End Sub
2486
2487     Private Sub PanelChartProperties_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PanelChartProperties.Click
2488         PanelChartProperties.Visible = False
2489     End Sub
2490
2491     Private Sub DUDNetC2TypeLines_SelectedItemChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DUDNetC2TypeLines.SelectedItemChanged
2492         If DUDNetC2TypeLines.SelectedIndex = 1 Then
2493             graph2settings.TypeofHline = typeofline.SolidLine
2494             graph2settings.TypeofVline = typeofline.SolidLine
2495             graph2settings.TypeofTVLines = typeofline.SolidLine
2496
2497             ElseIf DUDNetC2TypeLines.SelectedIndex = 0 Then
2498                 graph2settings.TypeofHline = typeofline.DashedLine
2499                 graph2settings.TypeofVline = typeofline.DashedLine
2500                 graph2settings.TypeofTVLines = typeofline.DashedLine
2501
2502         End If
2503     End Sub
2504
2505     Private Sub CheckedListBox1_SelectedValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
2506
2507     End Sub
2508
2509     Private Sub StBStatus_LocationChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles StBStatus.LocationChanged
2510         Pbleds.Location = New System.Drawing.Point(StBStatus.Location.X + (StBStatus.Width - Pbleds.Width) / 2, StBStatus.Location.Y + 5)
2511     End Sub
2512
2513     Private Sub MnGraph1Prop_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MnGraph1Prop.Click
2514         If MnGraph1Prop.Checked = False Then
2515             MnGraph1Prop.Checked = True
2516             MnGraph2Prop.Checked = False
2517         End If
2518
2519         ShowApropiateControl(PanelPropertiesGraph1.Name.ToString)
2520         PanelPropertiesGraph1.BringToFront()
2521     End Sub
2522
2523     Private Sub MnGraph2Prop_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MnGraph2Prop.Click
2524         If MnGraph2Prop.Checked = False Then
2525             MnGraph2Prop.Checked = True
2526             MnGraph1Prop.Checked = False
2527         End If
2528
2529         ShowApropiateControl(PanelPropertiesGraphs2.Name.ToString)
2530         PanelPropertiesGraphs2.BringToFront()
2531     End Sub
2532
2533     Private Sub MnChartProperties_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MnChartProperties.Click
2534         PanelChartProperties.BringToFront()
2535         ShowApropiateControl(PanelChartProperties.Name.ToString)
2536     End Sub
2537
2538     Public Sub PopulateCombos()

```

```

2539         DUDNetC2TypeLines.Items.Clear()
2540         DUDNetC2TypeLines.Items.Add(MyClassCode.GetStr(WhichIsTheLang, 182))
2541         DUDNetC2TypeLines.Items.Add(MyClassCode.GetStr(WhichIsTheLang, 183))
2542     End Sub
2543
2544     Public Sub WhichLang(ByVal WhichLang As Integer)
2545         Me.Text = MyClassCode.GetStr(WhichLang, 149)
2546         MenuItem1.Text = MyClassCode.GetStr(WhichLang, 150)
2547         MnConnect.Text = MyClassCode.GetStr(WhichLang, 150)
2548         MnDisconnect.Text = MyClassCode.GetStr(WhichLang, 11)
2549         MnExit.Text = MyClassCode.GetStr(WhichLang, 151)
2550         MItEfarmoges.Text = MyClassCode.GetStr(WhichLang, 152)
2551         MnElegxosGkaziou.Text = MyClassCode.GetStr(WhichLang, 153)
2552         MnElegxosStrofwn.Text = MyClassCode.GetStr(WhichLang, 154)
2553         MenuItem6.Text = MyClassCode.GetStr(WhichLang, 155)
2554         MnShowGraphs.Text = MyClassCode.GetStr(WhichLang, 139)
2555         MnHideGraphs.Text = MyClassCode.GetStr(WhichLang, 142)
2556         GrBxDataSave.Text = MyClassCode.GetStr(WhichLang, 50)
2557         LblSaveData.Text = MyClassCode.GetStr(WhichLang, 54)
2558         GrBxIncoming.Text = MyClassCode.GetStr(WhichLang, 41)
2559         LblStrofes.Text = MyClassCode.GetStr(WhichLang, 158)
2560         LblGkazi.Text = MyClassCode.GetStr(WhichLang, 52)
2561         LblLoadCell.Text = MyClassCode.GetStr(WhichLang, 159)
2562         LblAmperes.Text = MyClassCode.GetStr(WhichLang, 160)
2563         LblTime.Text = MyClassCode.GetStr(WhichLang, 161)
2564         GrBxOutgoing.Text = MyClassCode.GetStr(WhichLang, 7)
2565         LblOutStrofes.Text = MyClassCode.GetStr(WhichLang, 168)
2566         BtEmergencyStop.Text = MyClassCode.GetStr(WhichLang, 162)
2567         BtClose.Text = MyClassCode.GetStr(WhichLang, 58)
2568         Me.StBPCConnStatusLbl.Text = MyClassCode.GetStr(WhichLang, 62)
2569         Me.StBTimeLbl.Text = MyClassCode.GetStr(WhichLang, 61)
2570         Me.LblGraph1.Text = MyClassCode.GetStr(WhichLang, 140)
2571         Me.LblGraph2.Text = MyClassCode.GetStr(WhichLang, 141)
2572         Me.BtShowGraphs.Text = MyClassCode.GetStr(WhichLang, 139)
2573         Me.BtStart.Text = MyClassCode.GetStr(WhichLang, 55)
2574         Me.BtStop.Text = MyClassCode.GetStr(WhichLang, 56)
2575         Me.LblVoltage.Text = MyClassCode.GetStr(WhichLang, 169)
2576         Me.MnShowNETGraphs.Text = MyClassCode.GetStr(WhichLang, 170)
2577         Me.MnShowExcelGraphs.Text = MyClassCode.GetStr(WhichLang, 171)
2578         Me.MnNetParams.Text = MyClassCode.GetStr(WhichLang, 156)
2579         Me.MnGraph1Prop.Text = MyClassCode.GetStr(WhichLang, 157)
2580         Me.MnGraph2Prop.Text = MyClassCode.GetStr(WhichLang, 172)
2581         Me.MnChartProperties.Text = MyClassCode.GetStr(WhichLang, 173)
2582         Me.LbNumOfPoints2.Text = MyClassCode.GetStr(WhichLang, 174)
2583         Me.LbHrLines2.Text = MyClassCode.GetStr(WhichLang, 175)
2584         Me.LbVtLines2.Text = MyClassCode.GetStr(WhichLang, 176)
2585         Me.LbHDivLines2.Text = MyClassCode.GetStr(WhichLang, 177)
2586         Me.ChkNetC2PWMLoad.Text = MyClassCode.GetStr(WhichLang, 178)
2587         Me.ChkNetC2LoadCell.Text = MyClassCode.GetStr(WhichLang, 179)
2588         Me.ChkNetC2Voltage.Text = MyClassCode.GetStr(WhichLang, 180)
2589         Me.ChkNetC2Ampere.Text = MyClassCode.GetStr(WhichLang, 181)
2590         Me.RbDushedLines2.Text = MyClassCode.GetStr(WhichLang, 182)
2591         Me.RbSolidLines2.Text = MyClassCode.GetStr(WhichLang, 183)
2592         Me.DUDNetC2TypeLines.Text = MyClassCode.GetStr(WhichLang, 184)
2593         Me.LblNumOfPoints.Text = MyClassCode.GetStr(WhichLang, 174)
2594         Me.LblNofHorLines.Text = MyClassCode.GetStr(WhichLang, 175)
2595         Me.LblVerticalLines1.Text = MyClassCode.GetStr(WhichLang, 176)
2596         Me.LbHorizontalDiv1.Text = MyClassCode.GetStr(WhichLang, 177)
2597
2598         Me.ChkNetC2Ampere.Text = MyClassCode.GetStr(WhichLang, 181)
2599
2600         Me.DUDNetC1TypeLines.Items.Clear()
2601         Me.DUDNetC1TypeLines.Items.Add(MyClassCode.GetStr(WhichLang, 182)) ' dushed
2602         Me.DUDNetC1TypeLines.Items.Add(MyClassCode.GetStr(WhichLang, 183)) ' solid
2603         Me.DUDNetC2TypeLines.Items.Clear()
2604         Me.DUDNetC2TypeLines.Items.Add(MyClassCode.GetStr(WhichLang, 182))
2605         Me.DUDNetC2TypeLines.Items.Add(MyClassCode.GetStr(WhichLang, 183))
2606         Me.DUDNetC1VerticalLine.Items.Clear()
2607         Me.DUDNetC1VerticalLine.Items.Add(MyClassCode.GetStr(WhichLang, 270))
2608         Me.DUDNetC1VerticalLine.Items.Add(MyClassCode.GetStr(WhichLang, 176))
2609         Me.DUDNetC2VerticalLine.Items.Clear()
2610         Me.DUDNetC2VerticalLine.Items.Add(MyClassCode.GetStr(WhichLang, 270))
2611         Me.DUDNetC2VerticalLine.Items.Add(MyClassCode.GetStr(WhichLang, 176))
2612
2613         Me.BtPastMoving1.Text = MyClassCode.GetStr(WhichLang, 185)

```



```

2614     Me.BtPastMoving2.Text = MyClassCode.GetStr(WhichLang, 185)
2615
2616     Me.ChkNetC1Rpm.Text = MyClassCode.GetStr(WhichLang, 187)
2617     Me.ChkNetC1SetPointRPM.Text = MyClassCode.GetStr(WhichLang, 188)
2618     Me.ChkNetC1PWMLoad.Text = MyClassCode.GetStr(WhichLang, 189)
2619     Me.ChkNetC1SetPointPwm.Text = MyClassCode.GetStr(WhichLang, 190)
2620     Me.ChbEnable1Graph.Text = MyClassCode.GetStr(WhichLang, 191)
2621     Me.ChbEnable2Graph.Text = MyClassCode.GetStr(WhichLang, 191)
2622
2623     '
2624     Me.GbChartPrpperties1.Text = MyClassCode.GetStr(WhichLang, 192)
2625     Me.GbChartProperties2.Text = MyClassCode.GetStr(WhichLang, 193)
2626     Me.LbchartLine1.Text = MyClassCode.GetStr(WhichLang, 194)
2627     Me.LbChartLine2.Text = MyClassCode.GetStr(WhichLang, 194)
2628     Me.LbNoChartPoints1.Text = MyClassCode.GetStr(WhichLang, 195)
2629     Me.LbNoChartPoints2.Text = MyClassCode.GetStr(WhichLang, 195)
2630
2631     Me.GBSentFromSpecialFile.Text = MyClassCode.GetStr(WhichLang, 298)
2632     Me.LbReadFromFile.Text = MyClassCode.GetStr(WhichLang, 232) & " : "
2633     Me.MnSendFromFile.Text = MyClassCode.GetStr(WhichLang, 300)
2634     Me.GrBRefreshRateofGraphics.Text = MyClassCode.GetStr(WhichLang, 301)
2635     Me.LbBasicGraphisRate.Text = MyClassCode.GetStr(WhichLang, 303)
2636     Me.LbGraphicsRate.Text = MyClassCode.GetStr(WhichLang, 304)
2637     Me.MnITimingGraphs.Text = MyClassCode.GetStr(WhichLang, 305)
2638
2639     WhichIsTheLang = WhichLang
2640 End Sub
2641
2642 Private Sub PbBars_Click(ByVal sender As System.Object, ByVal e As System.
2643     EventArgs) Handles PbBars.Click
2644 End Sub
2645
2646 Private Sub TrackBarControlTimePointDisplay2_ValueChanged(ByVal sender As System.
2647     Object, ByVal e As System.EventArgs) Handles TrackBarControlTimePointDisplay2.
2648     ValueChanged
2649     If TrackBarControlTimePointDisplay2.Value < Me.ArraysMet Then
2650         TxtTimePointofGraph2.Text = MyClassCode.ConvertMilstoHoursMintsSecs
2651         (Tmatrix(TrackBarControlTimePointDisplay2.Value))
2652     End If
2653 End Sub
2654
2655 Private Sub PanelControlTimePointDisplay2_VisibleChanged(ByVal sender As System.
2656     Object, ByVal e As System.EventArgs) Handles PanelControlTimePointDisplay2.
2657     VisibleChanged
2658     If PanelControlTimePointDisplay2.Visible = True Then
2659         Me.BtPastMoving2.Location = New System.Drawing.Point(Me.
2660         PanelPropertiesGraphs2.Width - Me.PanelControlTimePointDisplay2.Width - Me.
2661         BtPastMoving2.Width - 2, (Me.PanelPropertiesGraphs2.Height - Me.BtPastMoving2.
2662         Height) / 2)
2663         Me.BtPastMoving2.Text = Me.MyClassCode.GetStr(Me.WhichIsTheLang, 186)
2664         TrackBarControlTimePointDisplay2.Maximum = Me.ArraysMet - 1
2665         TrackBarControlTimePointDisplay2.Minimum = 0
2666         TrackBarControlTimePointDisplay2.Value = TrackBarControlTimePointDisplay2.
2667         Maximum
2668     Else
2669         Me.BtPastMoving2.Text = Me.MyClassCode.GetStr(Me.WhichIsTheLang, 185)
2670         Me.BtPastMoving2.Location = New System.Drawing.Point(Me.
2671         PanelPropertiesGraphs2.Width - Me.BtPastMoving2.Width - 2, (Me.
2672         PanelPropertiesGraphs2.Height - Me.BtPastMoving2.Height) / 2)
2673     End If
2674 End Sub
2675
2676 Private Sub BtPastMoving2_Click(ByVal sender As System.Object, ByVal e As System.
2677     EventArgs) Handles BtPastMoving2.Click
2678     If PanelControlTimePointDisplay2.Visible = True Then
2679         PanelControlTimePointDisplay2.Visible = False
2680     Else
2681         PanelControlTimePointDisplay2.Location = New System.Drawing.Point(Me.
2682         PanelPropertiesGraphs2.Width - Me.PanelControlTimePointDisplay2.Width - 1, (Me.
2683         PanelPropertiesGraphs2.Height - Me.PanelControlTimePointDisplay2.Height) / 2)
2684         PanelControlTimePointDisplay2.Visible = True
2685     End If
2686 End Sub
2687
2688
2689
2690
2691
2692
2693

```

```

2674 Private Sub PanelControlTimePointDisplay_Paint(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.PaintEventArgs)
2675
2676 End Sub
2677 Private Function Index_to_Point_of_Graph2() As Integer
2678
2679     If ((PanelControlTimePointDisplay2.Visible = True Or ChbEnable1Graph.Checked =
True) And (TrackBarControlTimePointDisplay2.Value < Me.ArraysMet)) Then
2680         If ChbEnable1Graph.Checked = True And TrackBarControlTimePointDisplay2.
Value < Me.ArraysMet And PanelControlTimePointDisplay1.Visible = True Then
2681             Return TrackBarControlTimePointDisplay1.Value
2682         End If
2683         Return IIf(TrackBarControlTimePointDisplay2.Value > ArraysMet - 1, Math.
Max(ArraysMet - 1 - 50000, 0), TrackBarControlTimePointDisplay2.Value)
2684     Else
2685         Return IIf(Me.ArraysMet - 1 > 0, Me.ArraysMet - 1, 0)
2686     End If
2687 End Function
2688 Private Function Index_to_Point_of_Graph1() As Integer
2689     If ((PanelControlTimePointDisplay1.Visible = True Or ChbEnable2Graph.Checked =
True) And (TrackBarControlTimePointDisplay1.Value < Me.ArraysMet)) Then
2690         If ChbEnable2Graph.Checked = True And TrackBarControlTimePointDisplay1.
Value < Me.ArraysMet And PanelControlTimePointDisplay2.Visible = True Then
2691             Return TrackBarControlTimePointDisplay2.Value
2692         End If
2693         Return IIf(TrackBarControlTimePointDisplay1.Value > ArraysMet - 1, Math.
Max(ArraysMet - 1 - 50000, 0), TrackBarControlTimePointDisplay1.Value)
2694     Else
2695         Return IIf(Me.ArraysMet - 1 > 0, Me.ArraysMet - 1, 0)
2696     End If
2697 End Function
2698
2699 Private Sub PbGraph1Box1_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles PbGraph1Box1.Click
2700
2701 End Sub
2702
2703 Private Sub TrackBarControlTimePointDisplay1_ValueChanged(ByVal sender As System.
Object, ByVal e As System.EventArgs) Handles TrackBarControlTimePointDisplay1.
ValueChanged
2704     If TrackBarControlTimePointDisplay1.Value < Me.ArraysMet Then
2705         TxtTimePointofGraph1.Text = MyClassCode.ConvertMilstoHoursMintsSecs
(Tmatrix(TrackBarControlTimePointDisplay1.Value))
2706     End If
2707 End Sub
2708
2709 Private Sub BtPastMoving1_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles BtPastMoving1.Click
2710     If PanelControlTimePointDisplay1.Visible = True Then
2711         PanelControlTimePointDisplay1.Visible = False
2712     Else
2713         PanelControlTimePointDisplay1.Location = New System.Drawing.Point
(PanelPropertiesGraph1.Width - PanelControlTimePointDisplay1.Width - 1, (Me.
PanelPropertiesGraph1.Height - Me.PanelControlTimePointDisplay1.Height) / 2)
2714         PanelControlTimePointDisplay1.Visible = True
2715     End If
2716 End Sub
2717
2718 Private Sub PanelPropertiesGraph1_Paint(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles PanelPropertiesGraph1.Paint
2719     Dim g As System.Drawing.Graphics = e.Graphics
2720     e.Graphics.DrawString("1 Graph ", New System.Drawing.Font("Arial", 40),
Brushes.Silver, New System.Drawing.Point(Me.PanelPropertiesGraph1.Width - 280, (Me.
PanelPropertiesGraph1.Height - 55) / 2))
2721     e.Graphics.DrawRectangle(New System.Drawing.Pen(Color.Chocolate, 2), New
System.Drawing.Rectangle(1, 1, PanelPropertiesGraph1.Width - 2,
PanelPropertiesGraph1.Height - 2))
2722 End Sub
2723
2724 Private Sub PanelControlTimePointDisplay1_VisibleChanged(ByVal sender As System.
Object, ByVal e As System.EventArgs) Handles PanelControlTimePointDisplay1.
VisibleChanged
2725     If PanelControlTimePointDisplay1.Visible = True Then
2726         Me.BtPastMoving1.Location = New System.Drawing.Point(Me.

```

```

PanelControlTimePointDisplay1.Location.X - Me.BtPastMoving1.Width - 2, (Me.
PanelPropertiesGraph1.Height - Me.BtPastMoving1.Height) / 2)
2728     Me.BtPastMoving1.Text = MyClassCode.GetStr(WhichIsTheLang, 186)
2729     TrackBarControlTimePointDisplay1.Maximum = Me.ArraysMet - 1
2730     TrackBarControlTimePointDisplay1.Minimum = 0
2731     TrackBarControlTimePointDisplay1.Value = TrackBarControlTimePointDisplay1.
Maximum
2732     Else
2733         Me.BtPastMoving1.Text = MyClassCode.GetStr(WhichIsTheLang, 185)
2734         Me.BtPastMoving1.Location = New System.Drawing.Point(Me.
PanelPropertiesGraph1.Width - Me.BtPastMoving1.Width - 5, (Me.
PanelPropertiesGraph1.Height - Me.BtPastMoving1.Height) / 2)
2735     End If
2736 End Sub
2737
2738 Private Sub UnCheckNETGrahsProps()
2739     MnGraph1Prop.Checked = False
2740     MnGraph2Prop.Checked = False
2741 End Sub
2742
2743 Private Sub PanelPropertiesGraph1_MouseHover(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles PanelPropertiesGraph1.MouseHover
2744     FixToolTips(True, 1)
2745 End Sub
2746
2747 Private Sub PanelPropertiesGraphs2_MouseHover(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles PanelPropertiesGraphs2.MouseHover
2748     FixToolTips(True, 1)
2749 End Sub
2750
2751 Private Sub PanelChartProperties_MouseHover(ByVal sender As Object, ByVal e As
System.EventArgs) Handles PanelChartProperties.MouseHover
2752     FixToolTips(True, 1)
2753 End Sub
2754
2755 Private Sub PanelChartProperties_Paint(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles PanelChartProperties.Paint
2756     e.Graphics.DrawRectangle(New System.Drawing.Pen(Color.Chocolate, 2), New
System.Drawing.Rectangle(1, 1, PanelChartProperties.Width - 2,
PanelChartProperties.Height - 2))
2757 End Sub
2758
2759 Private Sub testtime()
2760     Me.Testtimer(0) = Now()
2761     If (Me.Testtimer(0).Ticks - Me.Testtimer(1).Ticks) / (10 ^ 4) < 10 ^ 32 And Me.
Testtimer(1).Ticks > 0 Then
2762         MySerialPort.Write(MyClassCode.SecureTransmit((Me.Testtimer(0).Ticks - Me.
Testtimer(1).Ticks) / (10 ^ 4)), 0, 6)
2763         Me.Testtimer(1) = Me.Testtimer(0)
2764     Else
2765         Me.Testtimer(1) = Me.Testtimer(0)
2766     End If
2767 End Sub
2768
2769 Private Sub TimeTest_Tick(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles TimeTest.Tick
2770     If FlagStartSend = True Then
2771         testtime()
2772     End If
2773 End Sub
2774
2775 Private Sub OnTimedEvent(ByVal source As Object, ByVal e As ElapsedEventArgs)
' for system timer usage
2776     If FlagStartSend = True Then
2777         testtime()
2778     End If
2779 End Sub
2780
2781 Private Sub TimeDisplayGraphics(ByVal unused As Object)
2782
2783 End Sub
2784
2785 Public Sub ServeControlByte(ByVal Control() As Byte, ByRef vbmode As Integer,
ByRef IARMode As Byte, ByRef myserialport As System.IO.Ports.SerialPort)
2786     Dim bit_access As New BitArray(Control)

```

```

2787     If (MnElegxosGkaziou.Checked = True And VBstatus = Modeofapplication.
ControlPWMDuty) Or (MnElegxosGkaziou.Checked = False And VBstatus =
Modeofapplication.ControlledRPM) Then
2788     Else
2789         If MnElegxosGkaziou.Checked Then
2790             VBstatus = Modeofapplication.ControlPWMDuty
2791         Else
2792             VBstatus = Modeofapplication.ControlledRPM
2793         End If
2794     End If
2795
2796     If bit_access(0) <> Me.VBMotor Then
2797
2798     End If
2799
2800     If CByte(bit_access(1) And 1) <> Me.VBSetPointRoot And myserialport.IsOpen
Then
2801
2802         Me.IARSetPointRoot = bit_access(1) And 1
2803         Select Case VBSetPointRoot
2804             Case SetpointsRoot.File
2805                 myserialport.Write(MyClassCode.SecureTransmit(4294967283), 0, 6)
2806             Case SetpointsRoot.FormControls
2807                 myserialport.Write(MyClassCode.SecureTransmit(4294967284), 0, 6)
2808         End Select
2809     Else
2810         Me.IARSetPointRoot = bit_access(1) And 1
2811     End If
2812
2813     If Cint(bit_access(2)) <> Me.VBMotor Then
2814
2815     End If
2816
2817     If Cint(bit_access(3)) = CBool(DoSendFrame.DoNotSendFrame) Then ' request
current frame
2818     ElseIf bit_access(3) = DoSendFrame.SendFrame And Me.VBSetPointRoot =
SetpointsRoot.File Then
2819     End If
2820     If bit_access(4) = CBool(DoSendFrame.DoNotSendFrame) Then ' request new
frame
2821
2822     ElseIf bit_access(4) = CBool(DoSendFrame.SendFrame) And (Me.IARSetPointRoot =
Me.VBSetPointRoot) And Me.IARmode = Me.VBstatus Then
2823         If RequestDataServed = True Then
2824             RequestDataServed = False
2825             sentdatafromfile()
2826
2827         End If
2828
2829     End If
2830     Me.IARMotor = Math.Abs(CInt(bit_access(5)))
2831     If Math.Abs(CInt(bit_access(5))) <> Me.VBMotor Then
2832
2833         Select Case Me.VBMotor
2834             Case MotorCondition.MotorRunning
2835                 InitiateMotorFromStopState()
2836             Case MotorCondition.MotorStopped
2837                 SendStopMotorSignal()
2838         End Select
2839     End If
2840     IARMode = (CByte(bit_access(7)) And 2) Or (CByte(bit_access(6)) And 1)
2841     If myserialport.IsOpen Then
2842         If IARMode = 2 Then
2843             Me.BeginInvoke(New InvokeDelagateMode(AddressOf ElegxosStrofwn), True)
2844         ElseIf IARMode = 3 Then
2845             Me.BeginInvoke(New InvokeDelagateMode(AddressOf ElegxosStrofwn),
False)
2846         ElseIf IARMode = 1 And VBstatus <> 1 Then
2847             myserialport.Write(MyClassCode.SecureTransmit(4294967281), 0, 6)
2848         ElseIf IARMode = 0 And VBstatus <> 0 Then
2849             myserialport.Write(MyClassCode.SecureTransmit(4294967282), 0, 6)
2850         End If
2851     End If
2852
2853     IARMode = (CByte(bit_access(7)) And 2) Or (CByte(bit_access(6)) And 1)

```

```

2854     End Sub
2855
2856     Private Sub sentdatafromfile()
2857         If FilewithData.ReadInitStatus = True Then
2858             FilewithData.Nooflinesread = 6
2859             If GetValuesFromSelectedFile(FilewithData) = True Then
2860                 FilewithData.Nooflinesread = 10
2861                 FilewithData.ReadInitStatus = False
2862             End If
2863         ElseIf FilewithData.FileisReaded = False Then
2864             If GetValuesFromSelectedFile(FilewithData) = True Then
2865                 For i As Integer = 0 To FilewithData.values.Length - 1
2866                     MySerialPort.Write(MyClassCode.SecureTransmit(FilewithData.values
2867 (i)), 0, 6)
2867                 Next
2868             End If
2869         End If
2870         RequestDataServed = True
2871
2872     End Sub
2873
2874
2875     Private Sub MnSendFromFile_Click(ByVal sender As System.Object, ByVal e As System.
2876 EventArgs) Handles MnSendFromFile.Click
2877         Try
2878             If Me.VBSetPointRoot = SetpointsRoot.File Then
2879                 MnSendFromFile.Text = MyClassCode.GetStr(WhichIsTheLang, 300)
2880                 Me.VBSetPointRoot = SetpointsRoot.FormControls
2881                 Me.TrBrOutStrofes.Value = 0
2882                 Me.TrBrOutStrofes.Enabled = True
2883                 Me.MItEfarmoges.Enabled = True
2884                 ShowApropiateControl("no one")
2885                 sr.Close()
2886                 BeginInvoke(New InvokeDelegateWrite2File(AddressOf
2887 Write_at_Head_of_MeForm), "")
2888                 Me.BtOpenSpecialFile.Enabled = True
2889             Else
2890                 ShowApropiateControl(GBSentFromSpecialFile.Name.ToString)
2891                 MnSendFromFile.Text = MyClassCode.GetStr(WhichIsTheLang, 299)
2892                 If Not FilewithData.FileName Is Nothing Then
2893                     If MyClassCode.IsValidFile(Me.FilewithData.FileName) Then
2894                         Select Case MsgBox(MyClassCode.GetStr(WhichIsTheLang, 307),
2895 MsgBoxStyle.YesNoCancel, MyClassCode.GetStr(WhichIsTheLang, 109))
2896                             Case MsgBoxResult.Yes
2897                                 sr = New IO.StreamReader(FilewithData.FileName)
2898                                 InitParametersForFileTransmission(FilewithData)
2899                                 BeginInvoke(New InvokeDelegateWrite2File(AddressOf
2900 Write_at_Head_of_MeForm), " \ " & MyClassCode.GetStr(WhichIsTheLang, 324))
2901                                 Me.VBSetPointRoot = SetpointsRoot.File
2902                                 Me.TrBrOutStrofes.Enabled = False
2903                                 Me.MnSendFromFile.Text = MyClassCode.GetStr
2904 (WhichIsTheLang, 324)
2905                                 Me.MItEfarmoges.Enabled = False
2906                                 Me.BtOpenSpecialFile.Enabled = False
2907                             Case MsgBoxResult.No
2908                                 MnSendFromFile.Text = MyClassCode.GetStr
2909 (WhichIsTheLang, 300)
2910                                 ShowApropiateControl("no one")
2911                                 Me.BtOpenSpecialFile.Enabled = True
2912                             Case MsgBoxResult.Cancel
2913                                 Dim scrap As Int16 = 0
2914                                 Me.BtOpenSpecialFile.Enabled = True
2915                         End Select
2916                     End If
2917                 End If
2918             End If
2919         Catch exp As Exception
2920             Stop
2921         End Try
2922     End Sub
2923     Private Sub ChangeModefromSendFile_to_Send_controls()

```

```

2922     MnSendFromFile.Text = MyClassCode.GetStr(WhichIsTheLang, 300) "Send Controls"
mode
2923     Me.VBSetPointRoot = SetpointsRoot.FormControls
2924     Me.TrBrOutStrofes.Value = 0
2925     Me.TrBrOutStrofes.Enabled = True ' eneble trackbar
2926     Me.MITefarmoges.Enabled = True
2927     ShowApropiateControl("no one")
2928     If sr Is Nothing Then
2929     Else
2930         sr.Close()
2931     End If
2932     BeginInvoke(New InvokeDelegateWrite2File(AddressOf Write_at_Head_of_MeForm),
2933     ""))
2934     Me.BtOpenSpecialFile.Enabled = True
2935     ElegxosStrofwn(False)
2936 End Sub
2937 Private Sub DUDNetC1VerticalLine_SelectedItemChanged(ByVal sender As System.Object
, ByVal e As System.EventArgs) Handles DUDNetC1VerticalLine.SelectedItemChanged
2938     Select Case Me.DUDNetC1VerticalLine.SelectedIndex
2939     Case 0 'time vertical /ms
2940         Me.NUDNetC1TimeVertInterval.Visible = True
2941         Me.NUDNetC1ConstVerLines.Visible = False
2942         Me.graph1settings.ShowTVLines = True
2943         Me.graph1settings.showVlines = False
2944
2945     Case 1 ' const vertical
2946         Me.NUDNetC1TimeVertInterval.Visible = False
2947         Me.NUDNetC1ConstVerLines.Visible = True
2948         Me.NUDNetC1ConstVerLines.Visible = True
2949         Me.graph1settings.ShowTVLines = False
2950         Me.graph1settings.showVlines = True
2951
2952     Case Else
2953     End Select
2954 End Sub
2955 Private Sub DUDNetC2VerticalLine_SelectedItemChanged(ByVal sender As System.Object
, ByVal e As System.EventArgs) Handles DUDNetC2VerticalLine.SelectedItemChanged
2956     Select Case Me.DUDNetC2VerticalLine.SelectedIndex
2957     Case 0 'time vertical
2958         Me.NUDNetC2TimeVertInterval.Visible = True
2959         Me.NUDNetC2ConstVerLines.Visible = False
2960         Me.graph2settings.ShowTVLines = True
2961         Me.graph2settings.showVlines = False
2962
2963     Case 1 ' const vertical
2964         Me.NUDNetC2TimeVertInterval.Visible = False
2965         Me.NUDNetC2ConstVerLines.Visible = True
2966         Me.NUDNetC2ConstVerLines.Visible = True
2967         Me.graph2settings.ShowTVLines = False
2968         Me.graph2settings.showVlines = True
2969
2970     Case Else
2971     End Select
2972 End Sub
2973 Private Sub NUDNetC1TimeVertInterval_ValueChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles NUDNetC1TimeVertInterval.ValueChanged
2974     Me.graph1settings.TimeIntervOfVertLine = NUDNetC1TimeVertInterval.Value
2975 End Sub
2976 Private Sub NUDNetC2TimeVertInterval_ValueChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles NUDNetC2TimeVertInterval.ValueChanged
2977     Me.graph2settings.TimeIntervOfVertLine = NUDNetC2TimeVertInterval.Value
2978 End Sub
2979 Private Function GetValuesFromSelectedFile(ByRef FileRead As FileReading) As
Boolean
2980     Try
2981     ' Create an instance of StreamReader to read from a file.
2982     Dim line As String
2983     Dim StartLine As String
2984

```

```

2990     If FileRead.ReadInitStatus = True Then
2991         ReDim FileRead.values(FileRead.Nooflinesread - 1)
2992     ElseIf (FileRead.values.Length <> FileRead.Nooflinesread) Then
2993         ReDim FileRead.values(FileRead.Nooflinesread - 1)
2994     End If
2995     StartLine = FileRead.line + 1
2996     Do
2997         FileRead.line += 1
2998         line = sr.ReadLine()
2999
3000         If line = Nothing Then
3001             Exit Do
3002         End If
3003         Select Case FileRead.line ' met
3004             Case Is > 6
3005                 FileRead.values(FileRead.line - StartLine) = ValueValidated(Me
.VBstatus, IIf((Me.VBstatus), Val(line) * ControlActionDivider, Val(line))) ' true
= 1 = gkazi 'false =0 = control rpm
3006                 Case 3
3007                     If line.IndexOf("Control RPM") <> -1 Then
3008                         If Not Me.VBstatus = Modeofapplication.ControlledRPM Then
3009                             Me.BeginInvoke(New InvokeDelagateMode(AddressOf
ElegxosStrofwn), True)
3010                             End If
3011                         ElseIf line.IndexOf("PWM Cycle(%)") <> -1 Then
3012                             If Not Me.VBstatus = Modeofapplication.ControlPWMDuty Then
3013                                 Me.BeginInvoke(New InvokeDelagateMode(AddressOf
ElegxosStrofwn), False)
3014                             End If
3015                         Else
3016                             Me.BeginInvoke(New InvokeDelagateMode(AddressOf
ResetFileTransitSelection), True)
3017                             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 306),
MsgBoxStyle.Information, MyClassCode.GetStr(WhichIsTheLang, 109)) 'file is not
valid
3018                             End If
3019                         Case 4
3020                             If line.IndexOf("Manually") <> -1 Then
3021                                 Else
3022                                     If line.IndexOf("With Function") <> -1 Then
3023                                         Else
3024                                             End If
3025                                         End If
3026                                     Case 5
3027
3028                                     Case 6
3029                                         GoTo jump
3030                                 End Select
3031
3032                     Loop Until (line Is Nothing) Or ((FileRead.line - StartLine + 1) =
FileRead.Nooflinesread) '****
3033
3034 jump:     If FileRead.line =6 Then
3035             FileRead.Nooflinesread = 10
3036         End If
3037         If (FileRead.line - StartLine + 1) <> FileRead.Nooflinesread Then
3038             ReDim Preserve FileRead.values(FileRead.line - StartLine)
3039         End If
3040
3041         If line Is Nothing Then
3042             FileRead.FileisReaded = True
3043             BeginInvoke(New InvokeDelegateWrite2File(AddressOf
Write_at_Head_of_MeForm), " \ " & MyClassCode.GetStr(WhichIsTheLang, 297))
3044             sr.Close()
3045             Return True
3046         End If
3047         Return True
3048
3049         Catch E As Exception
3050             Me.BeginInvoke(New InvokeDelagateMode(AddressOf ResetFileTransitSelection)
, True)
3051             ' Let the user know what went wrong.
3052             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 296) & E.Message, MsgBoxStyle.
Information, MyClassCode.GetStr(WhichIsTheLang, 109))
3053         End Try

```

```

3054
3055 End Function
3056 Private Sub Write_at_Head_of_MeForm(ByVal text As String)
3057     Me.Text = MyClassCode.GetStr(WhichIsTheLang, 149) & " " & text
3058 End Sub
3059 Private Sub GBSentFromSpecialFile_Enter(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles GBSentFromSpecialFile.Enter
3060
3061 End Sub
3062
3063 Private Sub BtOpenSpecialFile_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtOpenSpecialFile.Click
3064     Try
3065         Dim myStream As IO.Stream
3066         Dim openFileDialog1 As New OpenFileDialog
3067
3068         openFileDialog1.InitialDirectory = "c:\Windows\Desktop"
3069         openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*"
3070         openFileDialog1.FilterIndex = 1
3071         openFileDialog1.RestoreDirectory = True
3072         openFileDialog1.Multiselect = False
3073
3074         If openFileDialog1.ShowDialog() = System.Windows.Forms.DialogResult.OK And openFileDialog1.FileName.EndsWith(".txt") Then
3075             myStream = openFileDialog1.OpenFile()
3076
3077             If Not (myStream Is Nothing) Then
3078
3079                 If MyClassCode.IsValidFile(openFileDialog1.FileName) Then
3080                     Me.FilewithData.FileName = openFileDialog1.FileName
3081                     Me.TBReadFromFile.Text = openFileDialog1.FileName
3082                     myStream.Close()
3083
3084                 End If
3085             End If
3086         Else
3087             MsgBox(MyClassCode.GetStr(WhichIsTheLang, 294), MsgBoxStyle.Information, MyClassCode.GetStr(WhichIsTheLang, 295))
3088
3089         End If
3090     Catch exp As Exception
3091         MsgBox(exp.Message, MsgBoxStyle.Information, MyClassCode.GetStr(WhichIsTheLang, 109))
3092     End Try
3093 End Sub
3094
3095 Private Sub TBReadFromFile_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TBReadFromFile.TextChanged
3096     Try
3097         If MyClassCode.IsValidFile(Me.FilewithData.FileName) And VBSetPointRoot = SetpointsRoot.File Then
3098             'free the sr and resources
3099             sr.Close()
3100             sr = New IO.StreamReader(FilewithData.FileName)
3101             InitParametersForFileTransmission(FilewithData)
3102             Me.VBSetPointRoot = SetpointsRoot.File
3103             Me.TrBrOutStrofes.Enabled = False
3104
3105             Me.MnSendFromFile.Text = MyClassCode.GetStr(WhichIsTheLang, 324)
3106             Me.MItEfarmoges.Enabled = False
3107         End If
3108
3109     Catch
3110     End Try
3111 End Sub
3112
3113 Private Sub Thrust_FormClosed(ByVal sender As System.Object, ByVal e As System.Windows.Forms.FormClosedEventArgs) Handles MyBase.FormClosed
3114     If e.CloseReason = CloseReason.UserClosing And CreatedExcelProcID <> 0 Then
3115         MyClassCode.KillExcelProc(ProcessAlreadyClosed, CreatedExcelProcID)
3116     End If
3117
3118 End Sub
3119 Private Sub BasicTimerGraphsEvent(ByVal source As Object, ByVal e As ElapsedEventArgs)

```



```

3120         GetMeanValues (MeanBasic)
3121         Me.BeginInvoke(New InvokeDelagateGraphs(AddressOf DisplayDefaultGraphics)) '
3122
3123     End Sub
3124
3125     Private Sub CmbBasicGraphisRate_SelectedIndexChanged(ByVal sender As System.Object
3126     , ByVal e As System.EventArgs) Handles CmbBasicGraphisRate.SelectedIndexChanged
3127         Select Case CmbBasicGraphisRate.SelectedIndex
3128             Case 0
3129                 Me.BasicGraphicTimer.Stop()
3130                 InitValofMeanBasic()
3131             Case Else
3132                 Me.BasicGraphicTimer.Interval = Val(CmbBasicGraphisRate.SelectedItem.
3133                 ToString.Replace("ms", ""))
3134                 If Me.BasicGraphicTimer.Enabled = False Then
3135                     Me.BasicGraphicTimer.Enabled = True
3136                 End If
3137             End Select
3138         End Sub
3139
3140     Private Sub Thrust_FormClosing(ByVal sender As System.Object, ByVal e As System.
3141     Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing
3142         Me.BasicGraphicTimer.Stop()
3143         Me.BasicGraphicTimer.Dispose()
3144         Me.TmrGraphsTimer.Stop()
3145         Me.TmrGraphsTimer.Dispose()
3146         Me.MyTimer.Stop()
3147         Me.MyTimer.Dispose()
3148         Me.MySerialPort.Dispose()
3149         If MySerialPort.IsOpen Then
3150             MySerialPort.DiscardOutBuffer()
3151             MySerialPort.DiscardInBuffer()
3152             MySerialPort.Close()
3153         End If
3154     End Sub
3155
3156     Private Sub CmbGraphicsRate_SelectedIndexChanged(ByVal sender As System.Object,
3157     ByVal e As System.EventArgs) Handles CmbGraphicsRate.SelectedIndexChanged
3158         Select Case CmbGraphicsRate.SelectedIndex
3159             Case Is <= 6
3160                 Me.TmrGraphsTimer.Interval = Int(Val(CmbGraphicsRate.SelectedItem.
3161                 ToString.Replace("ms", "")))
3162                 If TmrGraphsTimer.Enabled = False Then
3163                     Me.TmrGraphsTimer.Start()
3164                 End If
3165             Case Is > 6
3166                 Me.TmrGraphsTimer.Interval = Int(Val(CmbGraphicsRate.SelectedItem.
3167                 ToString.Replace("sec", "")) * 1000)
3168                 If Me.TmrGraphsTimer.Enabled = False Then
3169                     Me.TmrGraphsTimer.Start()
3170                 End If
3171             End Select
3172         End Sub
3173
3174     Private Sub MnITimingGraphs_Click(ByVal sender As System.Object, ByVal e As System
3175     .EventArgs) Handles MnITimingGraphs.Click
3176         If Me.GrBRefreshRateofGraphics.Visible = False Then
3177             ShowApropiateControl(GrBRefreshRateofGraphics.Name.ToString)
3178         Else
3179             GrBRefreshRateofGraphics.Visible = False
3180         End If
3181     End Sub
3182
3183     Private Sub GrBRefreshRateofGraphics_Click(ByVal sender As System.Object, ByVal e
3184     As System.EventArgs) Handles GrBRefreshRateofGraphics.Click
3185         GrBRefreshRateofGraphics.Visible = False
3186     End Sub
3187
3188     Private Sub ShowApropiateControl(ByRef Showitem As String)
3189         PanelPropertiesGraphs2.Visible = False
3190         PanelPropertiesGraph1.Visible = False
3191         PanelChartProperties.Visible = False
3192         GrBRefreshRateofGraphics.Visible = False
3193         GBSentFromSpecialFile.Visible = False
3194         Select Case Showitem
3195             Case PanelPropertiesGraphs2.Name.ToString

```

```

3187         PanelPropertiesGraphs2.Visible = True
3188     Case PanelPropertiesGraph1.Name.ToString
3189         PanelPropertiesGraph1.Visible = True
3190     Case PanelChartProperties.Name.ToString
3191         PanelChartProperties.Visible = True
3192     Case GrBRefreshRateofGraphics.Name.ToString
3193         GrBRefreshRateofGraphics.Visible = True
3194     Case GBSentFromSpecialFile.Name.ToString
3195         GBSentFromSpecialFile.Visible = True
3196     End Select
3197
3198 End Sub
3199 Private Sub SaveMeanvalue(ByVal RpmValue As Integer, ByVal pwm1 As Integer, ByVal load1 As Integer, ByVal volt1 As Integer, ByVal amp1 As Integer)
3200     Savecomplete = False
3201     If Me.BasicGraphicTimer.Enabled = True Then
3202         ReDim Preserve Me.MeanBasic.rpm(MeanBasic.rpm.Length)
3203         ReDim Preserve Me.MeanBasic.volt(MeanBasic.volt.Length)
3204         ReDim Preserve Me.MeanBasic.pwm(MeanBasic.pwm.Length)
3205         ReDim Preserve Me.MeanBasic.cell(MeanBasic.cell.Length)
3206         ReDim Preserve Me.MeanBasic.amp(MeanBasic.amp.Length)
3207         Me.MeanBasic.rpm(MeanBasic.rpm.Length - 1) = RpmValue
3208         Me.MeanBasic.volt(MeanBasic.rpm.Length - 1) = volt1
3209         Me.MeanBasic.pwm(MeanBasic.rpm.Length - 1) = pwm1
3210         Me.MeanBasic.amp(MeanBasic.amp.Length - 1) = amp1
3211         Me.MeanBasic.cell(MeanBasic.cell.Length - 1) = load1
3212     Else
3213     End If
3214     Savecomplete = True
3215 End Sub
3216 Public Sub GetMeanValues(ByRef MeansavedValues As meanvlaues)
3217     Try
3218         If Savecomplete = True Then
3219             If MeansavedValues.rpm.Length >= 2 Then
3220                 For i As Integer = 1 To MeansavedValues.rpm.Length - 1
3221                     MeansavedValues.rpm(0) += MeansavedValues.rpm(i)
3222                     MeansavedValues.amp(0) += MeansavedValues.amp(i)
3223                     MeansavedValues.cell(0) += MeansavedValues.cell(i)
3224                     MeansavedValues.volt(0) += MeansavedValues.volt(i)
3225                     MeansavedValues.pwm(0) += MeansavedValues.pwm(i)
3226                 Next
3227                 RPM = MeansavedValues.rpm(0) / MeansavedValues.rpm.Length
3228                 pwm = MeansavedValues.pwm(0) / MeansavedValues.pwm.Length
3229                 cell = MeansavedValues.cell(0) / MeansavedValues.cell.Length
3230                 volt = MeansavedValues.volt(0) / MeansavedValues.volt.Length
3231                 amp = MeansavedValues.amp(0) / MeansavedValues.amp.Length
3232             ElseIf MeansavedValues.rpm.Length = 1 Then
3233                 RPM = MeansavedValues.rpm(0)
3234                 pwm = MeansavedValues.pwm(0)
3235                 cell = MeansavedValues.cell(0)
3236                 volt = MeansavedValues.volt(0)
3237                 amp = MeansavedValues.amp(0)
3238             Else
3239                 Exit Sub
3240             End If
3241             ReDim MeansavedValues.rpm(-1)
3242             ReDim MeansavedValues.amp(-1)
3243             ReDim MeansavedValues.cell(-1)
3244             ReDim MeansavedValues.pwm(-1)
3245             ReDim MeansavedValues.volt(-1)
3246         End If
3247     Catch
3248     End Try
3249 End Sub
3250 Private Sub InitValofMeanBasic()
3251     ReDim Me.MeanBasic.rpm(-1)
3252     ReDim Me.MeanBasic.amp(-1)
3253     ReDim Me.MeanBasic.cell(-1)
3254     ReDim Me.MeanBasic.volt(-1)
3255     ReDim Me.MeanBasic.pwm(-1)
3256 End Sub
3257 Private Sub ResetFileTransitSelection(ByVal reset As Boolean)
3258     Select Case reset
3259     Case True
3260         RequestDataServed = True

```

```

3261         Me.TrBrOutStrofes.Value = 0 ' set control first to 0
3262         Me.BeginInvoke(New InvokeDelagateMode(AddressOf ElegxosStrofwn),
False) 'then change to control gkazi
3263         Me.VBSetPointRoot = SetpointsRoot.FormControls ' change to read
from controls
3264         Me.GrBRefreshRateofGraphics.Visible = False
3265         Me.TrBrOutStrofes.Enabled = True 'enable control strofvn
3266         Me.FilewithData.FileName = "" 'first so not when text change get name
of file and get valid and change to mode readfrom file again do not reverse the
order of lines here
3267         Me.TBReadFromFile.Text = "" 'do not use it her beacause it will get it
from TBReadFromFile.textchange event and will
3268         Me.MnSendFromFile.Text = Me.MyClassCode.GetStr(WhichIsTheLang, 300)
3269         BeginInvoke(New InvokeDelegateWrite2File(AddressOf
Write_at_Head_of_MeForm), "")
3270         Case False
3271         End Select
3272     End Sub
3273     Private Sub ResetBufferArrays()
3274         Array.Clear(Tmatrix, 0, Tmatrix.Length)
3275         Array.Clear(Rmatrix, 0, Rmatrix.Length)
3276         Array.Clear(WRmatrix, 0, WRmatrix.Length)
3277         Array.Clear(Loadcell, 0, Loadcell.Length)
3278         Array.Clear(Voltage, 0, Voltage.Length)
3279         Array.Clear(Ampere, 0, Ampere.Length)
3280         Array.Clear(gkazi, 0, gkazi.Length)
3281         Array.Clear(gkazi, 0, gkazi.Length)
3282         Array.Clear(Wantedgkazi, 0, Wantedgkazi.Length)
3283
3284         Me.ArraysMet = 0
3285     End Sub
3286
3287     Private Function ValueValitaded(ByVal ApplicationMode As Modeofapplication, ByVal
data As UInt32) As Integer
3288         Select Case ApplicationMode
3289             Case Modeofapplication.ControlledRPM
3290                 If data <= MaxStrofes And data >= MinStrofes Then
3291                     Return data
3292                 Else
3293                     Throw New Exception(MyClassCode.GetStr(WhichIsTheLang, 318))
3294                     Return CInt(False)
3295                 End If
3296             Case Modeofapplication.ControlPWMDuty
3297                 If data <= MaxControlAction And data >= MinControlAction Then
3298                     Return data
3299                 Else
3300                     Throw New Exception(MyClassCode.GetStr(WhichIsTheLang, 318))
3301                     Return CInt(False)
3302                 End If
3303         End Select
3304
3305     End Function
3306
3307     Private Sub InitFileBuffer()
3308         ReDim FileBufferValues(-1)
3309     End Sub
3310
3311     Private Sub InitParametersForFileTransmission(ByRef FileRead As FileReading)
3312         FileRead.ReadInitStatus = True
3313         FileRead.line = 0
3314         FileRead.FileisReaded = False
3315         InitFileBuffer()
3316         sendingindex = 0
3317         ReDim FilewithData.values(-1)
3318     End Sub
3319
3320     Private Sub Getinitialvalues()
3321         GraphsAndChartsDrawingGeneralProperties()
3322         InitDrawPropertiesofControls()
3323         Dim line As String, MyPosition As Integer, MyParam, MyValue As String
3324         Dim WasFound As Boolean = False
3325         If MyClassCode.IsValidFile(TheSelectedConfFile) Then
3326             Try
3327                 Dim sr As IO.StreamReader = New IO.StreamReader(TheSelectedConfFile)
3328

```

```

3329         line = sr.ReadLine()
3330         Do While (Not (line Is Nothing))
3331             If Trim(line) = "[Graphs Section]" Or WasFound = True Then
3332                 WasFound = True
3333                 If Trim(line) <> "[Graphs Section]" Then
3334                     MyPosition = line.IndexOf("=")
3335                     If MyPosition <> -1 Then
3336                         MyParam = Trim(line.Substring(vbTab.Length, MyPosition
- vbTab.Length))
3337                         MyValue = Trim(line.Substring(MyPosition + 1, Trim
(line).Length - MyPosition - 1))
3338                         Select Case MyParam
3339                             Case "NetGraphs"
3340
3341                                 If (MyValue = "0") Then : Me.
MnShowNETGraphs_Click("Getinitialvalues", New System.EventArgs())
3342                                 End If
3343                                 Case "ExcelGraphs"
3344
3345                                     If (MyValue = "0") And Me.ExcelIsInstalled =
True Then : Me.MnShowExcelGraphs_Click("Getinitialvalues", New System.EventArgs())
3346                                     ElseIf (Me.ExcelIsInstalled = False) Then : Me
.MnShowNETGraphs_Click("Getinitialvalues", New System.EventArgs())
3347                                     End If
3348                                     Case "NetC1NumOfPoints"
3349
3350                                         Me.graph1settings.NofPointsofLine = Val
(MyValue)
3351
3352                                         Case "NetC1HorLines"
3353                                             Me.graph1settings.horizontalLines = Val
(MyValue)
3354
3355                                         Case "NetC1HorlDiv"
3356                                             Me.graph1settings.HLinesDivisions = Val
(MyValue)
3357
3358                                         Case "NetC1VerticalLine"
3359                                             Me.graph1settings.ShowTVLines = IIf(MyValue =
"0", True, False)
3360                                             Me.graph1settings.showVlines = IIf(MyValue =
"1", True, False)
3361                                             Me.DUDNetC1VerticalLine.SelectedIndex = Val
(MyValue)
3362
3363                                         Case "NetC1TimeVertInterval"
3364                                             Me.graph1settings.TimeIntervOfVertLine = Val
(MyValue)
3365
3366                                         Case "NUDNetC1ConstVerLines"
3367                                             Me.graph1settings.Verticallines = IIf(Val
(MyValue) <= NUDNetC1ConstVerLines.Maximum, Val(MyValue), 0)
3368                                             Me.NUDNetC1ConstVerLines.Value = IIf(Val
(MyValue) <= NUDNetC1ConstVerLines.Maximum, Val(MyValue), 0)
3369
3370                                         Case "NetC1TypeLines"
3371                                             Me.graph1settings.TypeofHline = Val(MyValue)
3372                                             Me.graph1settings.TypeofVline = Val(MyValue)
3373                                             Me.graph1settings.TypeofTVLines = Val(MyValue)
3374                                             Me.DUDNetC1TypeLines.SelectedIndex = Val
(MyValue)
3375
3376                                         Case "NetC1Rpm"
3377                                             Me.graph1settings.GraphLine1 = IIf(MyValue =
"0", True, False)
3378
3379                                         Case "NetC1SetPointRPM"
3380                                             Me.graph1settings.GraphLine2 = IIf(MyValue =
"0", True, False)
3381
3382                                         Case "NetC1PWMLoad"
3383                                             Me.graph1settings.GraphLine3 = IIf(MyValue =
"0", True, False)
3384
3385                                         Case "NetC1SetPointPwm"

```

```

3386         Me.graph1settings.GraphLine4 = IIf(MyValue =
3387         "0", True, False)
3388         Case "NetC2NumOfPoints"
3389         Me.graph2settings.NofPointsofLine = Val
3390         (MyValue)
3391         Case "NetC2HorLines"
3392         Me.graph2settings.horizontalLines = Val
3393         (MyValue)
3394         Case "NetC2Hor1Div"
3395         Me.graph2settings.HLinesDivisions = Val
3396         (MyValue)
3397         Case "NetC2VerticalLine"
3398         Me.graph2settings.ShowTVLines = IIf(MyValue =
3399         "0", True, False)
3400         Me.graph2settings.showVlines = IIf(MyValue =
3401         "1", True, False)
3402         Me.DUDNetC2VerticalLine.SelectedIndex = Val
3403         (MyValue)
3404         Case "NetC2TimeVertInterval"
3405         Me.graph2settings.TimeIntervOfVertLine = Val
3406         (MyValue)
3407         Case "NUDNetC2ConstVerLines"
3408         Me.graph2settings.Verticallines = IIf(Val
3409         (MyValue) <= NUDNetC2ConstVerLines.Maximum, Val(MyValue), 0)
3410         Me.NUDNetC2ConstVerLines.Value = IIf(Val
3411         (MyValue) <= NUDNetC2ConstVerLines.Maximum, Val(MyValue), 0)
3412         Case "NetC2TypeLines"
3413         graph2settings.TypeofHline = Val(MyValue)
3414         graph2settings.TypeofVline = Val(MyValue)
3415         graph2settings.TypeofTVLines = Val(MyValue)
3416         Me.DUDNetC2TypeLines.SelectedIndex = Val
3417         (MyValue)
3418         Case "NetC2PWMLoad"
3419         Me.graph2settings.GraphLine1 = IIf(MyValue =
3420         "0", True, False)
3421         Case "NetC2LoadCell"
3422         Me.graph2settings.GraphLine2 = IIf(MyValue =
3423         "0", True, False)
3424         Case "NetC2Voltage"
3425         Me.graph2settings.GraphLine3 = IIf(MyValue =
3426         "0", True, False)
3427         Case "NetC2Ampere"
3428         Me.graph2settings.GraphLine4 = IIf(MyValue =
3429         "0", True, False)
3430         Case "ExcelC1SelectLine"
3431         Me.Chart1Settings.Graphrpm = IIf(MyValue = "0"
3432         , True, False)
3433         Me.Chart1Settings.Graphgkazi = IIf(MyValue =
3434         "1", True, False)
3435         Me.Chart1Settings.Graphloadcell = IIf(MyValue
3436         = "2", True, False)
3437         Me.Chart1Settings.Graphampere = IIf(MyValue =
3438         "3", True, False)
3439         Me.Chart1Settings.Graphvoltage = IIf(MyValue =
3440         "4", True, False)
3441         Me.Chart1Settings.Graphwrpm = IIf(MyValue = "5"
3442         , True, False)
3443         Me.Chart1Settings.Graphwgkazi = IIf(MyValue =
3444         "6", True, False)
3445         Case "ExcelC1ChartPoints"
3446         Me.Chart1Settings.NoPointstodisplay = Val

```

```

(MyValue)
3439         Case "ExcelC2SelectLine"
3440
3441             Me.Chart2Settings.Graphrpm = IIf(MyValue = "0"
, True, False)
3442             Me.Chart2Settings.Graphgkazi = IIf(MyValue =
"1", True, False)
3443             Me.Chart2Settings.Graphloadcell = IIf(MyValue
= "2", True, False)
3444             Me.Chart2Settings.Graphampere = IIf(MyValue =
"3", True, False)
3445             Me.Chart2Settings.Graphvoltage = IIf(MyValue =
"4", True, False)
3446             Me.Chart2Settings.Graphwrpm = IIf(MyValue = "5
", True, False)
3447             Me.Chart2Settings.Graphwgkazi = IIf(MyValue =
"6", True, False)
3448
3449         Case "ExcelC2ChartPoints"
3450             Me.Chart2Settings.NoPointstodisplay = Val
(MyValue)
3451         Case "BasicGraphisRate"
3452             Me.CmbBasicGraphisRate.SelectedIndex = IIf(Val
(MyValue) >= 0, Val(MyValue), CmbBasicGraphisRate.Items.Count - 1)
3453         Case "GraphicsRate"
3454             Me.CmbGraphicsRate.SelectedIndex = IIf(Val
(MyValue) >= 0, Val(MyValue), CmbGraphicsRate.Items.Count - 1)
3455         End Select
3456             MyValue.Remove(0, MyValue.Length)
3457         End If
3458     End If
3459 End If
3460     line = sr.ReadLine()
3461     Loop
3462     sr.Close()
3463
3464     Catch E As Exception
3465         MsgBox(MyClassCode.GetStr(WhichIsTheLang, 127), MsgBoxStyle.
Information, MyClassCode.GetStr(WhichIsTheLang, 295))
3466         DefaultSetsofGraphs() 'load default values to continue working
3467     End Try
3468 Else
3469     MsgBox(MyClassCode.GetStr(WhichIsTheLang, 127), MsgBoxStyle.Information,
MyClassCode.GetStr(WhichIsTheLang, 295))
3470     DefaultSetsofGraphs() 'load default values to continue working
3471 End If
3472
3473 EnumerateGroupBoxesandPanelsWithApropiateValuesSettings()
3474
3475 End Sub
3476 Private Sub GraphsAndChartsDrawingGeneralProperties()
3477     '----- Graphs Constants Desine Values -----
3478     graph2settings.ShowHLines = True
3479     graph2settings.showVlines = True
3480     graph2settings.PenofHVLines = Pens.Gray
3481     graph2settings.ShowLegengofLines = True
3482     graph2settings.Location = New System.Drawing.Point(518, 370)
3483     graph2settings.Size = New System.Drawing.Size(515, 350)
3484     PbGraph2Box1.Location = graph2settings.Location
3485     PbGraph2Box1.Size = graph2settings.Size
3486     PbGraph2Box1.BorderStyle = BorderStyle.Fixed3D
3487
3488     graph1settings.ShowHLines = True
3489     graph1settings.showVlines = True
3490     graph1settings.PenofHVLines = Pens.Gray
3491     graph1settings.ShowLegengofLines = True
3492     graph1settings.Location = New System.Drawing.Point(0, 370)
3493     graph1settings.Size = New System.Drawing.Size(515, 350)
3494
3495     PbGraph1Box1.Location = graph1settings.Location
3496     PbGraph1Box1.Size = graph1settings.Size
3497     PbGraph1Box1.BorderStyle = BorderStyle.Fixed3D
3498
3499     showRpm = True

```

```

3500     showPwmLoad = True
3501
3502     '----- CHARTS Constant Desine VALUES-----
-----
3503     PBGraph1.SizeMode = PictureBoxSizeMode.StretchImage
3504     Me.PBGraph1.BorderStyle = BorderStyle.Fixed3D
3505     Me.Chart1Settings.ChartLocation = graph1settings.Location
3506     Me.Chart1Settings.ChartSize = graph1settings.Size
3507     Me.PBGraph1.Location = Chart1Settings.ChartLocation
3508     Me.PBGraph1.Size = Me.Chart1Settings.ChartSize
3509
3510     PBGraph2.SizeMode = PictureBoxSizeMode.StretchImage
3511     Me.PBGraph2.BorderStyle = BorderStyle.Fixed3D
3512     Me.Chart2Settings.ChartLocation = graph2settings.Location
3513     Me.Chart2Settings.ChartSize = graph2settings.Size
3514     Me.PBGraph2.Location = Chart2Settings.ChartLocation
3515     Me.PBGraph2.Size = Me.Chart2Settings.ChartSize
3516
3517     '----- CONTROLS THE MODE OF DISPLAYED GRAPHS -----
-----
3518     Graphs = IIf(MnShowNETGraphs.Checked, GraphicsDisplay.ShowGraphics,
GraphicsDisplay.showChartGraphics)
3519     BackgroundWorker1.WorkerSupportsCancellation = True
3520     '----- CONTROLS THE MODE OF DISPLAYED GRAPHS -----
-----
3521     Graphs = IIf(MnShowNETGraphs.Checked, GraphicsDisplay.ShowGraphics,
GraphicsDisplay.showChartGraphics)
3522
3523     '----- -Controls the bar settings -----
-----
3524     Barset.width = 20
3525     Barset.NofVal = 3
3526     Barset.height = 30
3527     Dim barbrush() As System.Drawing.Brush = {Brushes.Aqua, Brushes.Aqua, Brushes.
Aqua, Brushes.Aqua}
3528     Barset.brushes = barbrush
3529
3530     '----- CONTROLS THE DISPLAY OF LEDS-----
-----
3531     Ledset.diameter = 10
3532     Dim brushleds() As System.Drawing.Brush = {Brushes.Green, Brushes.Red}
3533     Ledset.Brush = brushleds
3534     Dim penled() As System.Drawing.Pen = {Pens.Black}
3535     Ledset.pen = penled
3536
3537     '----- init time count -----
-----
3538     'Intialtime = True
3539     Me.PanelPropertiesGraph1.Size = New System.Drawing.Size(Me.Width - 15, 60)
3540     Me.PanelPropertiesGraph1.Location = New System.Drawing.Point(5, 0)
3541     Me.PanelPropertiesGraphs2.Size = New System.Drawing.Size(Me.Width - 15, 60)
3542     Me.PanelPropertiesGraphs2.Location = New System.Drawing.Point(5, 0)
3543     Me.PanelChartProperties.Size = New System.Drawing.Size(Me.Width - 15, 60)
3544     Me.PanelChartProperties.Location = New System.Drawing.Point(5, 0)
3545
3546     Me.NUDNetC1TimeVertInterval.Minimum = 20
3547     Me.NUDNetC1TimeVertInterval.Maximum = 100000
3548     Me.NUDNetC2TimeVertInterval.Minimum = 20
3549     Me.NUDNetC2TimeVertInterval.Maximum = 100000
3550
3551 End Sub
3552
3553 Private Sub EnumarateGroupBoxesandPanelsWithApropiateValuesSettings()
3554
3555     ' Enumarate Panel for Graph2
3556     DUDNetC2TypeLines.SelectedItem = Me.graph2settings.TypeofHLine
3557     Me.NUDNetC2TimeVertInterval.Value = Me.graph2settings.TimeIntervOfVertLine
3558     NUD1NetC2NumOfPoints.Value = graph2settings.NofPointsofLine
3559     NUDNetC2HorLines.Value = graph2settings.horizontalLines
3560     NUDNetC2ConstVerLines.Value = graph2settings.Verticallines
3561     NUDNetC2HorLines.Value = graph2settings.horizontalLines
3562     NUD1NetC2HorlDiv.Value = graph2settings.HLinesDivisions
3563     ChkNetC2PWMLoad.Checked = graph2settings.GraphLine1
3564     ChkNetC2LoadCell.Checked = graph2settings.GraphLine2
3565     ChkNetC2Voltage.Checked = graph2settings.GraphLine3

```

```

3566     ChkNetC2Ampere.Checked = graph2settings.GraphLine4
3567     ' Enumarate Panel for Graph1
3568     DUDNetC1TypeLines.SelectedItem = Me.graph1settings.TypeofHline
3569     Me.NUDNetC1TimeVertInterval.Value = Me.graph1settings.TimeIntervOfVertLine
3570     NUDNetC1NumOfPoints.Value = graph1settings.NofPointsofLine
3571     NUDNetC1HorLines.Value = graph1settings.horizontalLines
3572     NUDNetC1ConstVerLines.Value = graph1settings.Verticallines
3573     NUDNetC1HorlDiv.Value = graph1settings.HLinesDivisions
3574     ChkNetC1Rpm.Checked = graph1settings.GraphLine1
3575     ChkNetC1SetPointRPM.Checked = graph1settings.GraphLine2
3576     ChkNetC1PWMLoad.Checked = graph1settings.GraphLine3
3577     ChkNetC1SetPointPwm.Checked = graph1settings.GraphLine4
3578     DUDNetC1TypeLines.SelectedIndex = graph1settings.TypeofHline
3579     'Enumarate Panel for ChartsExcel
3580     NUDExcelC1ChartPoints.Value = Chart1Settings.NoPointstodisplay
3581     NUDExcelC2ChartPoints.Value = Chart2Settings.NoPointstodisplay
3582
3583     Me.DUDExcelC1SelectLine.SelectedIndex = Math.Abs(0 * Me.Chart1Settings.
Graphrpm + 1 * Me.Chart1Settings.Graphgkazi + 2 * Me.Chart1Settings.Graphloadcell
+ 3 * Me.Chart1Settings.Graphampere + 4 * Me.Chart1Settings.Graphvoltage + 5 *
Chart1Settings.Graphwrpm + 6 * Chart1Settings.Graphwgtkazi) ' set iniial values
of selected lines for chart
3584     Me.DUDExcelC2SelectLine.SelectedIndex = Math.Abs(0 * Me.Chart2Settings.
Graphrpm + 1 * Me.Chart2Settings.Graphgkazi + 2 * Me.Chart2Settings.Graphloadcell
+ 3 * Me.Chart2Settings.Graphampere + 4 * Me.Chart2Settings.Graphvoltage + 5 *
Chart2Settings.Graphwrpm + 6 * Chart2Settings.Graphwgtkazi) ' set iniial values
of selected lines for chart works only for 1 line display
3585
3586 End Sub
3587 Private Sub InitDrawPropertiesofControls()
3588     Me.BtPastMoving2.Location = New System.Drawing.Point(Me.PanelPropertiesGraphs2.
.Width - Me.BtPastMoving2.Width - 5, (Me.PanelPropertiesGraphs2.Height - Me.
BtPastMoving2.Height) / 2)
3589     Me.BtPastMoving1.Location = New System.Drawing.Point(Me.PanelPropertiesGraph1.
Width - Me.BtPastMoving1.Width - 5, (Me.PanelPropertiesGraph1.Height - Me.
BtPastMoving1.Height) / 2)
3590     ' menu send file
3591     If Me.VBSetPointRoot = SetpointsRoot.FormControls Then
3592         MnSendFromFile.Text = Me.MyClassCode.GetStr(WhichIsTheLang, 300)
3593     Else : MnSendFromFile.Text = Me.MyClassCode.GetStr(WhichIsTheLang, 299)
3594     End If
3595     ' group box send from file
3596     Me.GBSentFromSpecialFile.Location = Me.GrBxDataSave.Location
3597     Me.GBSentFromSpecialFile.Size = Me.GrBxDataSave.Size
3598     ' group box timer of graphs
3599     Me.GrBRefreshRateofGraphics.Location = Me.GrBxDataSave.Location
3600     Me.GrBRefreshRateofGraphics.Size = Me.GrBxDataSave.Size
3601     Me.GrBRefreshRateofGraphics.Visible = False
3602     Me.CmbBasicGraphisRate.Items(0) = MyClassCode.GetStr(WhichIsTheLang, 302)
3603
3604     Me.CmbBasicGraphisRate.Location = New System.Drawing.Point(Me.
LbBasicGraphisRate.Location.X + 0.75 * (Me.LbBasicGraphisRate.Text.Length * Me.
LbBasicGraphisRate.Font.Size), CmbBasicGraphisRate.Location.Y)
3605     Me.CmbGraphicsRate.Location = New System.Drawing.Point(Me.LbGraphicsRate.
Location.X + 0.75 * (Me.LbGraphicsRate.Text.Length * Me.LbGraphicsRate.Font.Size),
CmbGraphicsRate.Location.Y)
3606
3607 End Sub
3608
3609 Private Sub DefaultSetsofGraphs()
3610     '----- GRAPH Default INITIAL VALUES -----
3611     graph2settings.NofPointsofLine = 501
3612     graph2settings.TypeofHline = typeoffline.DushedLine
3613     graph2settings.TypeofVline = typeoffline.DushedLine
3614     graph2settings.Verticallines = 0
3615     graph2settings.horizontalLines = 5
3616     graph2settings.HLinesDivisions = 0
3617     graph2settings.GraphLine1 = True
3618     graph2settings.GraphLine2 = True
3619     graph2settings.GraphLine3 = False
3620     graph2settings.GraphLine4 = True
3621     graph2settings.ShowTVLines = True
3622     graph2settings.TimeIntervOfVertLine = 10000
3623

```



```

3624
3625     graph1settings.NofPointsofLine = 501
3626     graph1settings.TypeofHline = typeofline.DushedLine
3627     graph1settings.TypeofVline = typeofline.DushedLine
3628     graph1settings.Verticallines = 0
3629     graph1settings.horizontalLines = 0
3630     graph1settings.HLinesDivisions = 0
3631     graph1settings.GraphLine1 = True
3632     graph1settings.GraphLine2 = True
3633     graph1settings.GraphLine3 = False
3634     graph1settings.GraphLine4 = False
3635     graph1settings.ShowTVLines = True
3636     graph1settings.TimeIntervOfVertLine = 10000
3637     '----- CHARTS INITIAL VALUES-----
-----
3638     Chart1Settings.Graphrpm = True
3639     Chart1Settings.Graphvoltage = False
3640     Chart1Settings.Graphwvkazi = False
3641     Chart1Settings.Graphwrpm = False
3642     Chart1Settings.Graphgkazi = False
3643     Chart1Settings.Graphloadcell = False
3644     Chart1Settings.Graphampere = False
3645     Chart1Settings.NoPointstodisplay = 50
3646
3647     Chart2Settings.Graphrpm = False
3648     Chart2Settings.Graphvoltage = False
3649     Chart2Settings.Graphwvkazi = False
3650     Chart2Settings.Graphwrpm = False
3651     Chart2Settings.Graphgkazi = False
3652     Chart2Settings.Graphloadcell = True
3653     Chart2Settings.Graphampere = False
3654     Chart2Settings.NoPointstodisplay = 50
3655
3656     Me.DUDNetC1VerticalLine.SelectedIndex = 0
3657     Me.DUDNetC2VerticalLine.SelectedIndex = 0
3658
3659     Me.NUDNetC1TimeVertInterval.Value = 10000
3660     Me.NUDNetC2TimeVertInterval.Value = 10000
3661
3662     Me.CmbBasicGraphisRate.SelectedIndex = 0
3663     Me.CmbGraphicsRate.SelectedIndex = 1
3664
3665 End Sub
3666 Private Sub ServeClose_Conn_Form()
3667     Static DialogResult As MsgBoxResult
3668
3669     Select Case MyClassCode.VbClosingState
3670         Case MyPtyxClass.ThrustConnFormClose.Close_False
3671             If MyClassCode.IARCardIsClosedProperly.Closed = True Then
3672                 If DialogResult = Nothing And MySerialPort.IsOpen = True Then
3673                     DialogResult = MsgBoxResult.Ignore
3674                     If MsgBox(MyClassCode.GetStr(WhichIsTheLang, 320), MsgBoxStyle
3675 .Information, MyClassCode.GetStr(WhichIsTheLang, 109)) = MsgBoxResult.Ok Then
3676                         DialogResult = Nothing
3677                         Me.MnDisconnect_Click("", New System.EventArgs)
3678                     End If
3679                 End If
3680             Exit Select
3681         Case MyPtyxClass.ThrustConnFormClose.Connection_Close
3682             If MyClassCode.IARCardIsClosedProperly.Closed = True And MySerialPort.
3683 IsOpen = True Then
3684                 Me.MnDisconnect_Click("ServeClose_Conn_Form" & "(Conn Close) ",
3685 New System.EventArgs)
3686             ElseIf MySerialPort.IsOpen = True Then
3687                 MySerialPort.Write(MyClassCode.SecureTransmit(4294967293), 0, 6)
3688             End If
3689         Case MyPtyxClass.ThrustConnFormClose.Form_Close
3690             If MyClassCode.IARCardIsClosedProperly.Closed = True Then
3691                 MenuItem5_Click("ServeClose_Conn_Form" & "(Form Close) ", New
3692 System.EventArgs)
3693             ElseIf MySerialPort.IsOpen = True Then
3694                 MySerialPort.Write(MyClassCode.SecureTransmit(4294967293), 0, 6)
3695             End If
3696         End Select

```

```

3694     End Sub
3695
3696     Private Sub AnimateEmergencyStopMotorButton(ByVal source As Object, ByVal e As ElapsedEventArgs)
3697         If Me.VBMotor = MotorCondition.MotorStopped Then
3698             If Me.BtEmergencyStop.BackColor = Color.OrangeRed Then
3699                 Me.BtEmergencyStop.BackColor = Me.BackColor
3700             Else
3701                 Me.BtEmergencyStop.BackColor = Color.OrangeRed
3702             End If
3703         End If
3704         If MyClassCode.VbClosingState = (MyPtyxClass.ThrustConnFormClose.Connection_Close) Then
3705             Me.AnimateEmergencyStopButtonlighting.Stop()
3706             Me.BtEmergencyStop.BackColor = Color.OrangeRed
3707         End If
3708     End Sub
3709     Private Sub Reset_AnimColorOfEmergencyStopMotorButton()
3710         If Me.VBMotor = MotorCondition.MotorStopped Then
3711             If Me.AnimateEmergencyStopButtonlighting.Enabled = False Then
3712                 Me.AnimateEmergencyStopButtonlighting.Start()
3713             End If
3714         End If
3715     End Sub
3716
3717 End Class

```

```

1 Public Class Version
2     Inherits System.Windows.Forms.Form
3
4 #Region " Windows Form Designer generated code "
5
6     Public Sub New()
7         MyBase.New()
8
9         'This call is required by the Windows Form Designer.
10        InitializeComponent()
11
12        'Add any initialization after the InitializeComponent() call
13
14    End Sub
15
16    'Form overrides dispose to clean up the component list.
17    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
18        If disposing Then
19            If Not (components Is Nothing) Then
20                components.Dispose()
21            End If
22        End If
23        MyBase.Dispose(disposing)
24    End Sub
25
26    'Required by the Windows Form Designer
27    Private components As System.ComponentModel.IContainer
28
29    'NOTE: The following procedure is required by the Windows Form Designer
30    'It can be modified using the Windows Form Designer.
31    'Do not modify it using the code editor.
32    Friend WithEvents LblVersion As System.Windows.Forms.Label
33    Friend WithEvents LblLVersion As System.Windows.Forms.Label
34    Friend WithEvents LblLDate As System.Windows.Forms.Label
35    Friend WithEvents LblDate As System.Windows.Forms.Label
36    Friend WithEvents BtOk As System.Windows.Forms.Button
37    Friend WithEvents LblLCreator As System.Windows.Forms.Label
38    Friend WithEvents LblCreator As System.Windows.Forms.Label
39    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
40        Dim resources As System.ComponentModel.ComponentResourceManager = New System.
41        ComponentModel.ComponentResourceManager(GetType(Version))
42        Me.LblLVersion = New System.Windows.Forms.Label
43        Me.LblVersion = New System.Windows.Forms.Label
44        Me.LblLDate = New System.Windows.Forms.Label
45        Me.LblDate = New System.Windows.Forms.Label
46        Me.BtOk = New System.Windows.Forms.Button
47        Me.LblLCreator = New System.Windows.Forms.Label
48        Me.LblCreator = New System.Windows.Forms.Label
49        Me.SuspendLayout()
50        '
51        'LblLVersion
52        resources.ApplyResources(Me.LblLVersion, "LblLVersion")
53        Me.LblLVersion.Name = "LblLVersion"
54        '
55        'LblVersion
56        resources.ApplyResources(Me.LblVersion, "LblVersion")
57        Me.LblVersion.Name = "LblVersion"
58        '
59        'LblLDate
60        resources.ApplyResources(Me.LblLDate, "LblLDate")
61        Me.LblLDate.Name = "LblLDate"
62        '
63        'LblDate
64        resources.ApplyResources(Me.LblDate, "LblDate")
65        Me.LblDate.Name = "LblDate"
66        '
67        'BtOk
68        Me.BtOk.DialogResult = System.Windows.Forms.DialogResult.Cancel
69        resources.ApplyResources(Me.BtOk, "BtOk")
70        Me.BtOk.Name = "BtOk"
71
72
73
74

```

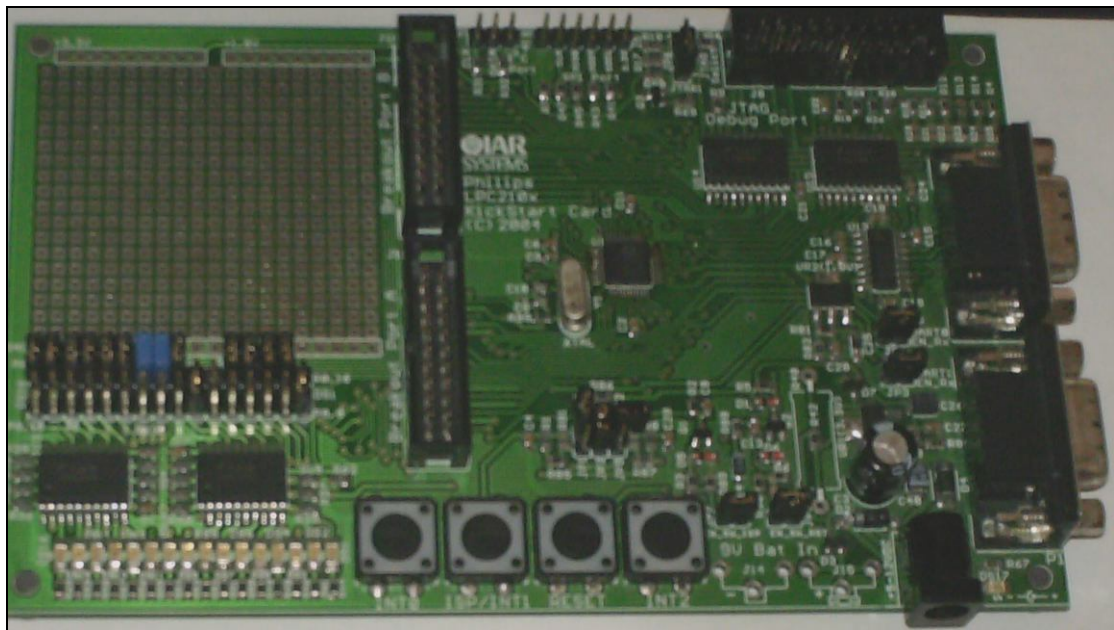
```

75         '
76         'LblLCreator
77         '
78         resources.ApplyResources(Me.LblLCreator, "LblLCreator")
79         Me.LblLCreator.Name = "LblLCreator"
80         '
81         'LblCreator
82         '
83         resources.ApplyResources(Me.LblCreator, "LblCreator")
84         Me.LblCreator.Name = "LblCreator"
85         '
86         'Version
87         '
88         resources.ApplyResources(Me, "$this")
89         Me.CancelButton = Me.BtOk
90         Me.Controls.Add(Me.LblCreator)
91         Me.Controls.Add(Me.LblLCreator)
92         Me.Controls.Add(Me.BtOk)
93         Me.Controls.Add(Me.LblDate)
94         Me.Controls.Add(Me.LblLDate)
95         Me.Controls.Add(Me.LblVersion)
96         Me.Controls.Add(Me.LblLVersion)
97         Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle
98         Me.MaximizeBox = False
99         Me.MinimizeBox = False
100        Me.Name = "Version"
101        Me.ResumeLayout(False)
102
103    End Sub
104
105 #End Region
106 Private MyClassCode As New MyPtyxClass
107
108 Private Sub BtOk_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
109     Handles BtOk.Click
110         Me.Close()
111 End Sub
112
113 Public Sub WhichLang(ByVal WhichLanguage As Integer)
114     'WhichLanguage = 0 for Greek
115     'WhichLanguage = 1 for English
116     LblDate.Text = MyClassCode.GetStr(WhichLanguage, 126)
117     LblLVersion.Text = MyClassCode.GetStr(WhichLanguage, 65)
118     LblVersion.Text = MyClassCode.GetStr(WhichLanguage, 66)
119     LblLDate.Text = MyClassCode.GetStr(WhichLanguage, 67)
120     LblLCreator.Text = MyClassCode.GetStr(WhichLanguage, 68)
121     LblCreator.Text = MyClassCode.GetStr(WhichLanguage, 69)
122     BtOk.Text = MyClassCode.GetStr(WhichLanguage, 70)
123     Me.Text = MyClassCode.GetStr(WhichLanguage, 71)
124 End Sub
125
126 Private Sub Version_Load(ByVal sender As System.Object, ByVal e As System.
127     EventArgs) Handles MyBase.Load
128 End Sub
129
130 Private Sub LblDate_Click(ByVal sender As System.Object, ByVal e As System.
131     EventArgs) Handles LblDate.Click
132 End Sub
133 End Class

```

ΠΑΡΑΡΤΗΜΑ Β

ΚΩΔΙΚΑΣ ΠΡΟΓΡΑΜΜΑΤΟΣ ΕΠΙ ΤΗΣ LPC2106



```

1 #include <stdbool.h>
2 #define MaxDataInBuff 100 // max 255
3 static unsigned long k14;
4 enum AppMode;
5 enum DataStatus;
6 struct MainBuffer;
7 struct BitAccess ;
8 union ControlByte ;
9 struct ErrorBitAccess ;
10 union ErrorByte ;
11 struct Long4Bytes ;
12 union ConvertLong2Bytes ;
13 enum DataStatus {Empty=0,Filled=1,Readed=-1} ;
14 static struct MainBuffer
15 {
16 enum DataStatus Flag ;
17 int Data ;
18 } MainBufferData[MaxDataInBuff];
19
20 void SecureTrasmitaData(bool flag ,unsigned long data ,unsigned char resettimes,unsigned
char status[] );
21 unsigned char IndextoWrite(unsigned char Windex ,struct MainBuffer Buffer[]);
22 unsigned char IndextoRead(unsigned char Rindex,struct MainBuffer Buffer[]);
23 void WritetoBuffer(unsigned char Windex,struct MainBuffer Buffer[],int Data);
24 unsigned char CalcFreeIndexestoWrite();
25 int ReadBuffer(unsigned char Rindex,struct MainBuffer Buffer[]);
26 bool LogicDataRequest(bool SaveMode);
27 void StoreData (unsigned long data,bool Savemode);
28 unsigned long GetData(bool Savemode);
29 void ResetBuffer();
30 unsigned char ConstractControlByte(unsigned char mode,bool Savemod ,bool MotorStopped,
bool RerquestData);
31 unsigned char ConstractErrorByte(bool bit7,bool bit6,bool bit5,bool bit4,bool bit3,bool
bit2,bool bit1,bool bit0);
32 unsigned char IntitialByteTransmission( unsigned long Gpio0Register);
33 void RefreshRecieveTime(unsigned long time);
34 void RecieveTimeoutService();
35 unsigned long FilterGetData(unsigned long Data,enum AppMode Mode );
36

```

```

1
2 #include <stdio.h>
3 #include <string.h>
4 #include <inarm.h>
5 #include <math.h>
6 #include "globalVariables.h"
7 #include <stdbool.h>
8 #include "UARTEcho.h"
9 #include "iolpc210x.h"
10 #define Timeout 921600 //ms 921.6
11 #define ResetTimeout 9216000 //ms 10 Sec
12 #define MaxPWMValue 10000
13 #define MinPWMValue 0
14 #define MaxRPMValue 20000
15 #define MinRPMValue 0
16
17 extern void Load(unsigned char mode, unsigned char MyChar);
18 extern void emergencystopmotor();
19 extern bool FlagGotoBeginningOfProg ;
20 extern void InitialResetFunctions();
21 ////////////////////////////////////////////////// externs for constructing controlByte
22 //////////////////////////////////////////////////
23 extern unsigned char mode;
24 extern enum AppMode IARAppMode;
25 extern bool IarMotorStopped ;
26
27 //////////////////////////////////////////////////
28 ////////////////////////////////////////////////// for buffer data functions //////////////////////////////////////////////////
29 //#####
30 unsigned char BuffReadIndex = 0,BuffWriteIndex = 1 ;
31 // for error check later will be removed #####
32 unsigned char CheckReadIndex = 0,CheckWriteIndex = 0 ;
33 //#####
34
35 extern bool PutSetPointsinBuffer ;
36 //#####
37 struct BitAccess
38 { unsigned char bit0 :1;
39   unsigned char bit1 :1;
40   unsigned char bit2 :1;
41   unsigned char bit3 :1;
42   unsigned char bit4 :1;
43   unsigned char bit5 :1;
44   unsigned char bit67 :2;
45 };
46 union ControlByte
47 { struct BitAccess ;
48   unsigned char Byte ;
49 };
50 union ControlByte ControllingByte;
51 struct ErrorBitAccess
52 {
53   unsigned char bit0 :1;
54   unsigned char bit1 :1;
55   unsigned char bit2 :1;
56   unsigned char bit3 :1;
57   unsigned char bit4 :1;
58   unsigned char bit5 :1;
59   unsigned char bit6 :1;
60   unsigned char bit7 :1;
61 };
62 union ErrorByte
63 {
64   struct ErrorBitAccess;
65   unsigned char ErrorChar ;
66 };
67 struct Long4Bytes
68 {
69   unsigned long Byte4 :8;
70   unsigned long Byte3 :8;

```

```

71  unsigned long Byte2 :8;
72  unsigned long Byte1 :8;
73  };
74  union ConvertLong2Bytes
75  {
76    struct Long4Bytes ;
77    unsigned long LongInt ;
78  };
79
80  unsigned char Xorbyte = 0;
81  union ConvertLong2Bytes ConvertedLong ;
82  void SecureTrasmitaData(bool flag ,unsigned long data ,unsigned char resettimes,
      unsigned char status[] )
83  {
84    ConvertedLong.LongInt = data ;
85
86    if ( resettimes > 254 )
87      { resettimes = 0 ;}
88
89    if( flag == true)
90    {
91      Xorbyte = resettimes^ConvertedLong.Byte2^ConvertedLong.Byte3^ConvertedLong.Byte4^
      ^status[0]^status[1]^Xorbyte ;
92      Load('w',(unsigned char) resettimes);
93      Load('w',(unsigned char) ConvertedLong.Byte2);
94      Load('w',(unsigned char) ConvertedLong.Byte3);
95      Load('w',(unsigned char) ConvertedLong.Byte4);
96      Load('w',(unsigned char) status[0]);
97      Load('w',(unsigned char) status[1]);
98      Load('w',Xorbyte);
99      Xorbyte = 0 ; //reset xorbe
100     Load('r', '0');
101
102   }
103   else
104   {
105     Xorbyte = ConvertedLong.Byte1^ConvertedLong.Byte2^ConvertedLong.Byte3^
      ConvertedLong.Byte4^Xorbyte ;
106     Load('w',(unsigned char) ConvertedLong.Byte1);
107     Load('w',(unsigned char) ConvertedLong.Byte2);
108     Load('w',(unsigned char) ConvertedLong.Byte3);
109     Load('w',(unsigned char) ConvertedLong.Byte4);
110   }
111 }
112
113 bool InitialIARstate = true;
114 unsigned char InitialdataArray[2] = {255,0};
115
116 void InitialDataRecieve(unsigned char byte)
117 {
118   if (byte == InitialdataArray[0])
119   { InitialIARstate = false ;}
120 }
121 }
122
123 void InitialTimer0interrupt()
124 {
125   Load('w',IntitialByteTransmiton(PINSEL0));
126   Load('r', '0');
127 }
128 }
129
130 unsigned char IndextoWrite(unsigned char Windex ,struct MainBuffer Buffer[])
131 { unsigned char ReturnIndex = 0 ;
132   if (Windex>=MaxDataInBuff-1)
133   {   if ((Buffer[0].Flag == Empty) || (Buffer[0].Flag == Readed))
134       { ReturnIndex = (0);}
135     else { ReturnIndex = (MaxDataInBuff-1);}}
136   else
137   {   if ((Buffer[Windex+1].Flag == Empty) || (Buffer[Windex+1].Flag == Readed))
138       { ReturnIndex = (Windex+1);}
139     else { ReturnIndex = (Windex);}}
140   return ReturnIndex ;
141 }
142 }

```



```

143
144 unsigned char IndextoRead(unsigned char Rindex,struct MainBuffer  Buffer[])
145 {  unsigned char ReturnIndex = 0 ;
146   if (Rindex>=MaxDataInBuff-1)
147   {if (Buffer[0].Flag == Filled )
148     { ReturnIndex = (0);}
149     else { ReturnIndex = (MaxDataInBuff-1);}
150   }
151   else {if (Buffer[Rindex+1].Flag ==  Filled)
152     { ReturnIndex = (Rindex+1);}
153     else { ReturnIndex = (Rindex);}
154   }
155   return ReturnIndex ;
156 }
157
158 void WritetoBuffer(unsigned char Windex,struct MainBuffer  Buffer[],int  Data)
159 {
160   Buffer[Windex].Data =  Data ;
161   Buffer[Windex].Flag = Filled ;
162   BuffWriteIndex =  Windex ;
163 }
164
165 unsigned char CalcFreeIndexestoWrite()
166 { unsigned char ReturnFreeCells=0;
167   for( int i =0; i< MaxDataInBuff;i++)
168   { if (MainBufferdData[i].Flag == Filled){ReturnFreeCells+=1;}    }
169   return (MaxDataInBuff-ReturnFreeCells) ;
170 }
171 }
172
173 int ReadBuffer(unsigned char Rindex,struct MainBuffer  Buffer[])
174 {
175   Buffer[Rindex].Flag = Readed;
176   BuffReadIndex = Rindex ;
177   return Buffer[Rindex].Data ;
178 }
179
180 bool LogicDataRequest(bool SaveMode)
181 { unsigned char Freeindexes  = CalcFreeIndexestoWrite();
182   bool ReturnBool;
183   switch(SaveMode)
184   {
185     case true:
186       if ((Freeindexes >=50) && (Freeindexes <= MaxDataInBuff))
187       {ReturnBool= true;}
188       else
189       {ReturnBool= false;}
190       break;
191     case false:
192       ReturnBool= false;
193       break;
194   }
195   return ReturnBool;
196 }
197
198 void StoreData (unsigned long data,bool Savemode)
199 {
200   switch(Savemode)
201   {
202     case true :
203       WritetoBuffer(IndextoWrite(BuffWriteIndex,MainBufferdData),MainBufferdData,data);
204     break;
205     case false :
206       WritetoBuffer(0,MainBufferdData,data);
207     break;
208     default :
209     break ;
210   }
211 }
212 unsigned long GetData(bool  Savemode)
213 {unsigned long Returndata;
214   switch(Savemode)
215   {
216     case true :
217       Returndata= FilterGetData(ReadBuffer(IndextoRead(BuffReadIndex,MainBufferdData),

```

```

        MainBufferdData), IARAppMode);
218     break;
219     case false :
220     Returndata= FilterGetData(ReadBuffer(0,MainBufferdData), IARAppMode);
221     break;
222 }
223 return Returndata;
224 }
225 void ResetBuffer()
226 {
227     BuffReadIndex = 0;
228     BuffWriteIndex = 0;
229     int i ;
230     for (i = 0 ; i <= (MaxDataInBuff-1) ; i++)
231     { MainBufferdData[i].Flag = Empty ;
232       MainBufferdData[i].Data = 0;
233     }
234 }
235 }
236
237 unsigned char  ConstractControlByte(unsigned char mode,bool Savemod ,bool MotorStopped
, bool RerquestData)
238 {
239     union ControlByte ConByte;
240     ConByte.bit0 = 0;
241     ConByte.bit1 = (unsigned char) Savemod ;
242     ConByte.bit4 = (unsigned char) RerquestData ;
243     ConByte.bit5 = (unsigned char) MotorStopped ;
244     ConByte.bit67 = (unsigned char) mode;
245     return  ConByte.Byte;
246 }
247
248 unsigned char ConstractErrorByte(bool bit7,bool bit6,bool bit5,bool bit4,bool bit3,bool
bit2,bool bit1,bool bit0)
249 {
250     union ErrorByte Error ;
251     Error.bit0 = bit0 ;
252     Error.bit1 = bit1 ;
253     Error.bit2 = bit2 ;
254     Error.bit3 = bit3 ;
255     Error.bit4 = bit4 ;
256     Error.bit5 = bit5 ;
257     Error.bit6 = bit6 ;
258     Error.bit7 = bit7 ;
259
260     return Error.ErrorChar ;
261 }
262 unsigned char IntitialByteTransmition( unsigned long Gpio0Register)
263 { static unsigned char ReturnChar;
264   switch (Gpio0Register)
265   {
266     case 0x20258005:
267       ReturnChar = 255;
268       break;
269     default :
270       ReturnChar = 254;
271       break;
272   }
273   return ReturnChar;
274 }
275 unsigned long RecTime = 0;
276 void RefreshRecieveTime(unsigned long time)
277 {
278     RecTime = time;
279 }
280 unsigned long  nowtime=0;
281 void RecieveTimeoutService()
282 { unsigned long Now = T1TC ;
283   if (Now > RecTime)
284   {unsigned long Dtime=Now-RecTime;
285     if ((Dtime>Timeout)&&(IarMotorStopped==false))
286     {emergencystopmotor();}
287     else if((Dtime>ResetTimeOut)&&(FlagGotoBeginingOfProg==false))
288     {FlagGotoBeginingOfProg = true;
289       InitialResetFunctions();}

```

```
290 }
291 else if(Now<RecTime)
292 { unsigned long Dtime=0xffffffff-RecTime+Now;
293   if ((Dtime>Timeout)&&(IarMotorStopped==false))
294     {emergencystopmotor();}
295   else if((Dtime>ResetTimeOut)&&(FlagGotoBeginingOfProg==false))
296     { FlagGotoBeginingOfProg = true;
297       InitialResetFunctions();}
298 }
299 else
300 {}
301
302 }
303
304 unsigned long FilterGetData(unsigned long Data,enum AppMode Mode )
305 { unsigned long ReturnData ;
306   switch ( Mode)
307   {
308     case ControlSpRPM:
309     ReturnData = Data <= MaxRPMValue? Data : 0 ;
310     break;
311     case ControlSpPWM:
312     ReturnData = Data <= MaxPWMValue ? Data : 0 ;
313     break;
314     default:
315     ReturnData = 0;
316     break;
317   }
318 }
319 return ReturnData ;
320 }
321
```

```

1
2 //XTAL frequency in Hz
3 #define XTALFREQ 14745600
4
5 //pclk must always be XTALFREQ/4 when using the ROM monitor.
6 #define PCLKFREQ (XTALFREQ/4)
7
8 void LPC210xSystemInit();
9
10 void LPC210xInitVIC();
11
12 void LPC210xInitTimerInterrupt(void(*timer_func)(),void(*timerMR1_func)());
13 void LPC210xInitTimerInterruptAtBegin(void(*timer_func)(),void(*timerMR1_func)());
14 void LPC210xInitTimer1Interrupt(void(*timer1_func)(),void (*timerlreset_func)());
15 void LPC210xInitExternalInterrupt0(void(*eint0_func)());
16 void LPC210xInitExternalInterrupt1(void(*eint1_func)(),void(*eint1soft_func)());
17 void LPC210xInitExternalInterrupt2(void(*eint2_func)());
18 void LPC210xInitUART1Interrupt(void(*uart1rx_func)(),
19                               void(*uart1tx_func)());
20 void LPC210xInitUART0Interrupt(void(*uart0rx_func)(unsigned char),void(*uart0tx_func)
21                               (unsigned char,unsigned char));
22 void LPC210xInitTimer();
23 void LPC210xStartTimer();
24 void LPC210xResetStopTimer();
25 void LPC210xInitTimer1();
26 void LPC210xStartTimer1();
27 void LPC210xResetStopTimer1();
28 void LPC210xInitPWM();
29 void LPC210xSTARTPWM();
30 void LPC210xSTOPPWM();
31 void LPC210xInitRPMCOUNT();
32 void LPC210xStartRPMCOUNT();
33 void LPC210xIniteEINT0();
34 void LPC210xIniteEINT1();
35 void LPC210xIniteEINT2();
36 unsigned char EINT0PinIsLow();
37 unsigned char EINT1PinIsLow();
38 unsigned char EINT2PinIsLow();
39
40 void LPC210xInitPIO();
41 void LPC210xLedSet(unsigned short setmask, unsigned short clearmask);
42 void LPC210xInitUART1();
43 void LPC210xStopResetUART1();
44 void LPC210xInitUART0();
45 void LPC210xStopResetUART();
46 //Transmits one byte via UART1-UART0
47 //Note: The UART1-UART0 THRE register must be empty before the call to this function.
48 void LPC210xUART1TxByte(unsigned char byte);
49 void LPC210xUART0TxByte(unsigned char byte);
50
51

```

```

1
2 /*
3  * $Revision: 1.8 $
4  */
5
6 #include <inarm.h>
7 #include <iolpc210x.h>
8 #include <stdio.h>
9 #include <string.h>
10 #include "lpc210x.h"
11
12
13
14 #define BAUDRATE          9600
15 #define BAUDRATEDIVISOR  (PCLKFREQ/(BAUDRATE*16))
16 #define BAUTERATE_UART0  57600//38400//115200
17 #define BAUDRATEDIVISOR_UART0 (PCLKFREQ/(BAUTERATE_UART0*16))
18 //Masks out the leds that must be driven from pins in the range p0.16-p0.31
19 //when ROM-monitor is used.
20 #define ROM_MONITOR_DEBUG_LEDMASK      0x8003
21
22 //Masks out the pins in the range p0.16-p0.31 that should be used to drive leds
23 //when ROM-monitor is used.
24 #define ROM_MONITOR_DEBUG_LEDPINMASK  0xc000000f
25
26 #define VIC_TIMER0_bit  (1 << VIC_TIMER0)
27 #define VIC_TIMER1_bit  (1 << VIC_TIMER1)
28 #define VIC_EINT0_bit   (1 << VIC_EINT0)
29 #define VIC_EINT1_bit   (1 << VIC_EINT1)
30 #define VIC_EINT2_bit   (1 << VIC_EINT2)
31 #define VIC_UART1_bit   (1 << VIC_UART1)
32 #define VIC_UART0_bit   (1 << VIC_UART0)
33 #define VIC_PWM0_bit    (1 << VIC_PWM0)
34
35 //Pointers to interrupt callback functions.
36 static void (*timer_function)();
37 static void (*timerMR1_function)();
38
39 static void (*timer1_function)();
40 static void (*timer1reset_function)();
41
42 static void (*eint0_function)();
43 static void (*eint1_function)();
44 static void (*eintlsoft_function)();
45 static void (*eint2_function)();
46 static void (*uartlrx_function)();
47 static void (*uartltx_function)();
48 static void (*uart0rx_function)(unsigned char);
49 static void (*uart0tx_function)(unsigned char,unsigned char);
50 static void (*PWM4_function)();
51 //
52 // Interrupt handlers.
53 //
54
55 //Dummy interrupt handler, called as default in irqHandler() if no other
56 //vectored interrupt is called.
57 static void DefDummyInterrupt()
58 {}
59
60 // Timer interrupt handler
61 static void TimerInterrupt()
62 {
63     if( T0IR_bit.MR0INT==1 )
64         {(*timer_function)();
65          T0IR_bit.MR0INT=1;
66         } // Call timer callback function.
67
68     if( T0IR_bit.MR1INT==1 )//T0IR_bit.CR2INT==1
69         {(*timerMR1_function)(); // Call timer callback capture 2 function.
70          T0IR_bit.MR1INT=1; // T0IR_bit.MR1INT=1
71         }
72
73 }
74
75

```

```

76
77 static void Timer1Interrupt()
78 {
79     if ( T1IR_bit.MR0INT == 1)
80     {(*timer1reset_function )(); //execute reset timer function
81         T1IR_bit.MR0INT = 1 ; // reset interrupt flag
82     }
83     if (T1IR_bit.CR0INT==1)
84     {
85         (*timer1_function)(); // Call timer callback function.
86         T1IR_bit.CR0INT=1; // reset interrupt flag
87     }
88
89 }
90
91 // The external interrupt handlers should call their callback functions
92 // only once for every button press. If the button is pressed the
93 // corresponding flag will not be reset when it is written to.
94 static void ExternalInterrupt0()
95 {
96
97     EXTINT_bit.EINT0 = 1 ; // Try to reset external interrupt flag.
98     if(!EXTINT_bit.EINT0) // Check if flag was reset (button not pressed).
99         (*eint0_function)(); // Call external interrupt callback function.
100
101 }
102
103 static void ExternalInterrupt1()
104 {
105     if ( VICSoftInt_bit.INT15 == 1)
106     {(*eint1soft_function)();
107         VICSoftIntClear_bit.INT15=1; // clears interrupt bit in VICSoftInt register
108     }
109     else
110     {
111         EXTINT_bit.EINT1 = 1; // Try to reset external interrupt flag.
112         if(!EXTINT_bit.EINT1) // Check if flag was reset (button not pressed).
113         {(*eint1_function)(); } // Call external interrupt callback function.*/
114     }
115     EXTINT_bit.EINT1 = 1;
116 }
117
118 }
119
120 static void ExternalInterrupt2()
121 {
122     EXTINT_bit.EINT2 = 1; // Try to reset external interrupt flag.
123     if(!EXTINT_bit.EINT2) // Check if flag was reset (button not pressed).
124         (*eint2_function)(); // Call external interrupt callback function.
125 }
126
127
128 //UART1 interrupt handler
129 static void UART1Interrupt()
130 {
131     switch(U1IIR_bit.IID)
132     {
133     case 0x1: //THRE interrupt
134         (*uart1tx_function)(); //Call tx buffer empty callback function
135         break;
136     case 0x2: //Receive data available
137         (*uart1rx_function)(); //Call received byte callback function
138         break;
139     case 0x0: //Modem interrupt
140     case 0x3: //Receive line status interrupt (RDA)
141     case 0x6: //Character time out indicator interrupt (CTI)
142         break;
143     default:
144         break;
145     }
146     U1SCR = U1IIR_bit.IID ;
147 }
148
149 //UART0 interrupt handler
150 static void UART0Interrupt()

```

```

151 {
152     switch(U0IIR_bit.IID)
153     {
154     case 0x1: //THRE interrupt
155         (*uart0tx_function)('r','*'); //Call tx buffer empty callback function
156         break;
157     case 0x2: //Receive data available
158         (*uart0rx_function)(U0RBR); //Call received byte callback function
159         break;
160     //case 0x0: //Modem interrupt
161     case 0x3: //Receive line status interrupt (RDA)
162     case 0x6: //Character time out indicator interrupt (CTI)
163     default:
164         break;
165     }
166 }
167 static void PWMInterrupt()
168 {
169
170     (*PWM4_function)(); // interrupt function call
171
172     PWMIR_bit.MR2INT=1; //clears interrupt flag for pwm2
173 }
174 }
175
176 // IRQ exception handler. Calls the interrupt handlers.
177 __irq __arm void irq_handler(void)
178 {
179     void (*interrupt_function)();
180     unsigned int vector;
181
182     vector = VICVectAddr; // Get interrupt vector.
183     interrupt_function = (void(*)())vector;
184     (*interrupt_function)(); // Call vectored interrupt function.
185
186     VICVectAddr = 0; // Clear interrupt in VIC.
187 }
188
189
190 //
191 // System initialization.
192 //
193 void LPC210xSystemInit()
194 {
195 #ifdef JTAG_DEBUG
196     // Map lowest 64 bytes of the address space to bottom of internal RAM.
197     // This will move the exception vectors in place.
198     MEMMAP = 2;
199 #endif
200 }
201
202
203 //
204 // Interrupt controller initialization.
205 //
206
207 void LPC210xInitVIC()
208 {
209     // Setup interrupt controller.
210     VICProtection = 0;
211     // Disable all interrupts
212     VICIntEnClear = 0xffffffff;
213     VICDefVectAddr = (unsigned int)&DefDummyInterrupt;
214 }
215
216 void LPC210xInitTimerInterrupt(void(*timer_func)(),void(*timerMR1_func)())
217 {
218     // Setup timer callback function.
219     timer_function = timer_func;
220     timerMR1_function = timerMR1_func;
221     VICIntSelect &= ~VIC_TIMER0_bit; // IRQ on timer 0 line.
222     VICVectAddr3 = (unsigned int)&TimerInterrupt;
223     VICVectCntl3 = 0x20 | VIC_TIMER0; // Enable vector interrupt for timer 0.
224     VICIntEnable = VIC_TIMER0_bit; // Enable timer 0 interrupt.
225 }

```

```

226 void LPC210xInitTimerInterruptAtBegin(void(*timer_func)(),void(*timerMR1_func)())
227 {
228     // Setup timer callback function.
229     timer_function = timer_func;
230     timerMR1_function = timerMR1_func;
231
232     VICIntSelect &= ~VIC_TIMER0_bit; // IRQ on timer 0 line.
233     VICVectAddr1 = (unsigned int)&TimerInterrupt;
234     VICVectCntl1 = 0x20 | VIC_TIMER0; // Enable vector interrupt for timer 0.
235     VICIntEnable = VIC_TIMER0_bit; // Enable timer 0 interrupt.
236 }
237 void LPC210xInitTimer1Interrupt(void(*timer1_func)(),void (*timer1reset_func)())
238 {
239     // Setup timer callback function.
240     timer1_function = timer1_func;
241     timer1reset_function = timer1reset_func ;
242
243     VICIntSelect &= ~VIC_TIMER1_bit; // IRQ on timer 1 line.
244     VICVectAddr2 = (unsigned int)&Timer1Interrupt;
245     VICVectCntl2 = 0x20 | VIC_TIMER1; // Enable vector interrupt for timer 1.
246     VICIntEnable = VIC_TIMER1_bit; // Enable timer 1 interrupt.
247 }
248
249 void LPC210xInitExternalInterrupt0(void(*eint0_func)())
250 {
251     // Setup eint0 callback function.
252     eint0_function = eint0_func;
253
254     VICIntSelect &= ~VIC_EINT0_bit; // IRQ on external int 0.
255     VICVectAddr5 = (unsigned int)&ExternalInterrupt0;
256     VICVectCntl5 = 0x20 | VIC_EINT0; // Enable vector interrupt for EINT 0.
257     VICIntEnable = VIC_EINT0_bit; // Enable EINT 0 interrupt.
258 }
259
260 void LPC210xInitExternalInterrupt1(void(*eint1_func)(),void(*eint1soft_func)())
261 {
262     // Setup eint1 callback function.
263     eint1_function = eint1_func;
264     eint1soft_function = eint1soft_func ;
265
266     VICIntSelect &= ~VIC_EINT1_bit; // IRQ on external int 1.
267     VICVectAddr6 = (unsigned int)&ExternalInterrupt1;
268     VICVectCntl6 = 0x20 | VIC_EINT1; // Enable vector interrupt for EINT1.
269     VICIntEnable = VIC_EINT1_bit; // Enable EINT 1 interrupt.
270 }
271
272 void LPC210xInitExternalInterrupt2(void(*eint2_func)())
273 {
274     // Setup eint2 callback function.
275     eint2_function = eint2_func;
276
277     VICIntSelect &= ~VIC_EINT2_bit; // IRQ on external int 2.
278     VICVectAddr7 = (unsigned int)&ExternalInterrupt2;
279     VICVectCntl7 = 0x20 | VIC_EINT2; // Enable vector interrupt for EINT2.
280     VICIntEnable = VIC_EINT2_bit; // Enable EINT 2 interrupt.
281 }
282
283 void LPC210xInitUART1Interrupt(void(*uart1rx_func)(),
284                                void(*uart1tx_func)())
285 {
286     // Setup uart1 callback functions.
287     uart1rx_function = uart1rx_func;
288     uart1tx_function = uart1tx_func;
289
290     VICIntSelect &= ~VIC_UART1_bit; // IRQ on UART1.
291     VICVectAddr1 = (unsigned int)&UART1Interrupt;
292     VICVectCntl1 = 0x20 | VIC_UART1; // Enable vector interrupt for UART1.
293     VICIntEnable = VIC_UART1_bit; // Enable UART 1 interrupt.
294 }
295 void LPC210xInitUART0Interrupt(void(*uart0rx_func)(unsigned char),void(*uart0tx_func)
296                                (unsigned char,unsigned char))
297 {
298     // Setup uart0 callback functions.
299     uart0rx_function = uart0rx_func;

```



```

300  uart0tx_function = uart0tx_func;
301
302  VICIntSelect &= ~VIC_UART0_bit; // IRQ on UART0.
303  VICVectAddr0 = (unsigned int)&UART0Interrupt;
304  VICVectCntl0 = 0x20 | VIC_UART0; // Enable vector interrupt for UART0.
305  VICIntEnable = VIC_UART0_bit; // Enable UART 0 interrupt.
306 }
307 void LPC210xinitPWMIInterrupt(void(*PWM4_func)())
308 {
309  //set PWM function callback functions
310  PWM4_function=PWM4_func ;
311
312  VICIntSelect &= ~VIC_PWM0_bit; // IRQ on UART0.
313  VICVectAddr4 = (unsigned int)&PWMIInterrupt;
314  VICVectCntl4 = 0x20 | VIC_PWM0; // Enable vector interrupt for UART0.
315  VICIntEnable = VIC_PWM0_bit; // Enable UART 0 interrupt.
316
317 }
318
319 //
320 /////////////////////////////////////////////////////////////////// Timer functions. ///////////////////////////////////////////////////////////////////
321 //
322
323 void LPC210xInitTimer()
324 {
325  T0TCR = 0; // Disable timer 0.
326  TOPR = 0; // Prescaler is set to no division.
327  TOMR0 = (PCLKFREQ*20)/1000; // Count up to this value. Generate 20 msecond interrupt.
328  TOMR1 = TOMR0/2;
329  TOMCR = 11;//3; // Reset and interrupt on MR0 (match register 0).
330  TOCCR = 0;// Capture is disabled.
331  TOEMR = 0; // No external match output.
332 }
333
334 void LPC210xStartTimer()
335 {
336  T0TCR = 1; // Enable timer 0.
337 }
338 void LPC210xResetStopTimer() // resets and stop timer//
339 { T0TCR =2;}
340
341 void LPC210xInitTimer1()
342 {
343  PINSEL0_bit.P0_10=0x2; //set pinsel0 to function capture 1.0 (Timer1)
344  T1TCR = 0; // Disable timer 0.
345  T1PR = 3 ;// PCLKFREQ/4 Prescaler is set to division 1/4 PCLKFREQ frequency
346  T1MR0 = 0xffffffff; // Count up to this value.
347  T1MCR = 3; // Reset and interrupt on MR0 (match register 0).
348  T1CCR_bit.CAP0RE=1; //capture is enabled at capture 1.0
349  T1CCR_bit.CAP0INT=1;//interrupt at capture 1.0
350  T1EMR = 0; // No external match output.
351 }
352
353
354 void LPC210xStartTimer1()
355 {
356  T1TCR = 1; // Enable timer 0.
357 }
358 void LPC210xResetStopTimer1() // resets and stop timer//
359 {
360  T1TCR =2;
361 }
362
363 //
364 // External interrupt functions
365 //
366
367 void LPC210xInitEINT0()
368 {
369  PINSEL1_bit.P0_16=0x1; // Set pin function to EINT0
370 }
371
372 void LPC210xInitEINT1()
373 {

```

```

374  PINSEL0_bit.P0_14=0x2; // Set pin function to EINT1
375  }
376
377  void LPC210xInitEINT2()
378  {
379    PINSEL0_bit.P0_15=0x2; // Set pin function to EINT2
380  }
381
382  unsigned char EINT0PinIsLow()
383  {
384    return !IOPIN_bit.P0_16;
385  }
386
387  unsigned char EINT1PinIsLow()
388  {
389    return !IOPIN_bit.P0_14;
390  }
391
392  unsigned char EINT2PinIsLow()
393  {
394    return !IOPIN_bit.P0_15;
395  }
396
397
398  //
399  // Parallel I/O functions.
400  //
401
402  void LPC210xInitPIO()
403  {
404    #ifdef ROM_MONITOR_DEBUG
405      unsigned int dir;
406      //Uses some of the GPIO pins P0.0-P0.15
407      PINSEL0 &= ROM_MONITOR_DEBUG_LEDPINMASK; // &= 1100 0000 0000 0000 0000 0000 0000 0000  ✎
408      //Uses some of the GPIO pins P0.16-P0.31
409      PINSEL1 &= ~ROM_MONITOR_DEBUG_LEDPINMASK; // &= 0011 1111 1111 1111 1111 1111 1111 1111  ✎
410      // Set LED pins on GPIO to output.
411      dir = (ROM_MONITOR_DEBUG_LEDMASK << 16)
412            | ~ROM_MONITOR_DEBUG_LEDMASK;
413      IODIR |= dir;
414    #else
415      //Uses GPIO pins P0.0-P0.15
416      PINSEL0 = 0x20258005 ; // function as gpio pin15 { 10 00 00 00 10 01 01 10 00 00  ✎
417                        // 00 00 00 01 01 } pin0 toy dinv apo thn arxh tis times oste na mhn exv problhmata me ✎
418                        // to dir ton pins kai kaoyn
419      IODIR |= 0x00000000; //direction input for better working
420      IOCLR |= 0x0000ffff;
421    #endif
422  }
423
424
425
426  //
427  // LED output drivers.
428  //
429
430  void LPC210xLedSet(unsigned short setmask, unsigned short clearmask)
431  {
432    // Set LED DS1 to DS16.
433    #ifdef ROM_MONITOR_DEBUG
434      // When the ROM-monitor is used some LED:s are driven by pins in the range
435      // P0.0-P0.15 and some by pins in the range P0.16-P0.31.
436      unsigned int clrhi = (ROM_MONITOR_DEBUG_LEDMASK & clearmask) << 16;
437      unsigned int clrlo = (~ROM_MONITOR_DEBUG_LEDMASK & clearmask);
438      unsigned int sethi = (ROM_MONITOR_DEBUG_LEDMASK & setmask) << 16;
439      unsigned int setlo = (~ROM_MONITOR_DEBUG_LEDMASK & setmask);
440      IOCLR = clrhi | clrlo;
441      IOSET = sethi | setlo;
442    #else
443      //When the program is run with J-Link or standalone, the LED:s are driven by
444      //pins P0.0-P0.15.

```

```

445  IOCLR = clearmask;
446  IOSET = setmask;
447 #endif
448 }
449
450 //
451 // UART functions
452 //
453
454 void LPC210xInitUART1()
455 {
456  //Set pins for use with UART
457  PINSEL0_bit.P0_8=0x1; // Set pin function to TxD (UART1)
458  PINSEL0_bit.P0_9=0x1; // Set pin function to RxD (UART1)
459
460  //Set the FIFO enable bit in the FCR register. This bit must be set for
461  //proper UART operation.
462
463  U1FCR_bit.FCRFE = 1;
464  U1FCR_bit.RTLS = 0 ; // 1 byte trigger level of receive data available interrupt of
    UART1
465
466  //Set baudrate
467  U1LCR_bit.DLAB = 1; //επιτρέπει την εισοδο στον ορισμο της ταχυτητας
468  U1DLL = BAUDRATEDIVISOR & 0x00ff;
469  U1DLM = (BAUDRATEDIVISOR >> 8) & 0x00ff;
470  U1LCR_bit.DLAB = 0; // απαγορευει την εισοδο στον ορισμο της ταχύτητας
471
472  //Set mode
473  U1LCR_bit.WLS = 0x3;    //8 bit word length
474  U1LCR_bit.SBS = 0x0;    //1 stop bit
475  U1LCR_bit.PE = 0x0;    //No parity
476
477  //Enable UART1 interrupts
478  U1IER_bit.RDAIE = 1; //Enable byte received interrupt
479
480 }
481 void LPC210xStopResetUART1()
482 {
483  U1FCR_bit.FCRFE = 0; //disables fifo and clears automatically clears FIFO control
484  U1IER_bit.RDAIE = 0; //Disable byte received interrupt
485  U1IER_bit.THREIE = 0; //Disable tx buf empty interrupt
486  U1IER_bit.RXLSIE = 0; //Disable the Rx line status interrupts
487 }
488 void LPC210xInitUART0()
489 {
490  //Set pins for use with UART
491  PINSEL0_bit.P0_0=0x1; // Set pin function to TxD (UART0)
492  PINSEL0_bit.P0_1=0x1; // Set pin function to RxD (UART0)
493
494  //Set the FIFO enable bit in the FCR register. This bit must be set for
495  //proper UART operation.
496  U0FCR_bit.FCRFE = 1;
497  U0FCR_bit.RFR = 1 ;//clears reciever buffer
498  U0FCR_bit.TFR = 1 ;//clears transmit buffer
499  U0FCR_bit.RTLS=0;
500  //Set baudrateTFR
501  U0LCR_bit.DLAB = 1;
502  U0DLL = BAUDRATEDIVISOR_UART0 & 0x00ff;
503  U0DLM = (BAUDRATEDIVISOR_UART0 >> 8) & 0x00ff;
504  U0LCR_bit.DLAB = 0;
505
506  //Set mode
507  U0LCR_bit.WLS = 0x3;    //8 bit word length
508  U0LCR_bit.SBS = 0x0;    //1 stop bit
509  U0LCR_bit.PE = 0x0;    //No parity
510
511  //Enable UART0 interrupts
512  U0IER_bit.RDAIE = 1; //Enable byte received interrupt
513  U0IER_bit.THREIE = 1; //Enable tx buf empty interrupt // Πρέπει να είναι
    ενεργοποιημένο
514 }
515 void LPC210xStopResetUART()
516 {
517  U0FCR_bit.FCRFE = 0; //disables fifo and clears automatically clears FIFO control

```

```

518  UOIER_bit.RDAIE = 0; //Disable byte received interrupt
519  UOIER_bit.THREIE = 0; //Disable tx buf empty interrupt
520  UOIER_bit.RXLSIE = 0; //Disable the Rx line status interrupts
521 }
522 void LPC210xInitPWM() // gia arxh tha energopoiiso
523 {
524 //Set pins for use with I2C
525 PINSEL0_bit.P0_7=0x2; // Set pin function to PWM ti pin name P0.7 ????? 10 pwm2
526 //(En)/Disable PWM Tmer Counter & Prescale Counter
527 //
528 PWMTCR_bit.CE=0; //PWM timer control register counter enable ( .CE ) when 1.when
    0 disable
529 // set PWM Timer Counter
530
531 PWMPR=0; //when 0 the TC is increment on every pclk
532
533 // Set PWM rate with PWM Match 0 and Match 5 (go low) will be save to shadow
    registers untill PWM enabled
534
535 //PWMMR0= 3697; //PCLKFREQ/1000; //(19.91*PCLKFREQ)/1000; // thelo h periodos tou
    pwm an einai 20msecond na do an xreizetai kana alos orismos
536 PWMMR0= PCLKFREQ / (2* 1000);
537 PWMMR2= 0; //PCLKFREQ*9/1000; //(den ksero an xreazetaoi edo)!!!!!!! (default case)
    prepei na oriso thn metblhth ti typos tha einai kai prepei na pernei times apo 0 ~
    2 milisecond se kykloys epi * pclk
538 PWMLER=0x5; //I give toy PWMLER the value 00000101
539 //set PWM Control Register enable and select the type of PWM channel. set ta channel
    5 single edge!!!!
540 // Set the the operation that will be performed when one of the PWM Match registers
541 // match the PWM timer counter
542 PWMMCR_bit.MR0RES=1; //when one, the PWMTC will be reset if PWMMR0 matches it.when
    zero this feature is disable.
543 PWMMCR_bit.MR2INT=0;
544 PWMPCR_bit.SEL2=0; //single edge ta PWM2
545 PWMPCR_bit.ENA2=1; // Enable OUT put on PWM2
546 //Enable PWM function and PWM Tmer Counter & Prescale Counter
547 PWMTCR_bit.PWMEN =1; // amatch event(PWM Match 0) will occur to cause shadow
    registers contents become effective
548 }
549
550 void LPC210xSTARTPWM()
551 { //
552 // (En)/Disable PWM Tmer Counter & Prescale Counter
553 //
554 PWMTCR_bit.CE=1; //PWM timer control register counter enable ( .CE ) when 1.when
    0 disable
555 }
556 void LPC210xSTOPPWM()
557 {
558 PWMTCR_bit.CE=0; //PWM timer control register counter enable ( .CE ) when 1.when
    0 disable
559 }
560
561 //Transmits one byte via UART1
562 //Note: The UART1 or UART0 THRE register must be empty before the call to this
563 // function. Otherwise data can be lost.
564 void LPC210xUART1TxByte(unsigned char byte)
565 {
566 U1THR = byte;
567 }
568 void LPC210xUART0TxByte(unsigned char byte)
569 {
570 U0THR = byte;
571 }
572
573
574

```

```
1 void Gkazi();  
2  
3
```

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <inarm.h>
4 #include "iolpc210x.h"
5 #include "PWMcosFunction.h"
6 #include "RpmcounterFunction.h"
7 #include "globalVariables.h"
8 #include "lpc210x.h"
9 #include "UARTEcho.h"
10
11 extern enum AppMode IARAppMode ;
12 extern unsigned long DutyTime ;
13 extern bool PutSetPointsinBuffer;
14 extern unsigned long GetData(bool Savemode);
15 extern float RPMmovingAverage(unsigned long time);
16 extern bool IarMotorStopped;
17 unsigned long SetpointPWM = 0;
18
19 void Gkazi()
20 {
21     switch(IarMotorStopped)
22     {
23         case false:
24             SetpointPWM = (unsigned long) GetData(PutSetPointsinBuffer) ;
25             DutyTime = (unsigned long) ( ( ( SetpointPWM*PWM_Duty )/ 10000 ) + PWM_Min) ;
26             PWMset(DutyTime);
27             break;
28
29         case true:
30             PWMset(PWM_Min);
31             break;
32         default:
33             break;
34     }
35 }
36
37
38
39
40
```

```

1 //////////////////////////////////////////////////////////////////RpmCounetrFunction.h////////////////////////////////
2 #include <stdbool.h>
3
4 void Load(unsigned char mode, unsigned char MyChar);
5 unsigned long Convert4Byteteto32bitInteger(unsigned long Byte1,unsigned long Byte2,
    unsigned long Byte3,unsigned long Byte4);
6 void callbackEint0Interrupt();
7 void ResetRpmCounter();
8 unsigned long ReturnRpmcount(void);
9 void SecureTrasmitaData(bool flag ,unsigned long data ,unsigned char resettimes,unsigned
    char status[] );
10 void callBackTimer0interrupt();
11 void CallBackTimer1interrupt();
12 void callBackTimer1ResetInterrupt();
13 void CallBackCaptureTimer0interrupt();
14 float RPMmovingAverage(unsigned long time);
15 void SenttoRpmFunction(unsigned long temp);
16 void loadcelldata2Rpmfunction(unsigned long temp);
17 static float tempRpm;
18 void tash ();
19 void GkaziOrTasi();
20
21

```

```

1
2 ////////////////////////////////////////////////// rpmcounter functions //////////////////////////////////////
3
4 #include <stdio.h>
5 #include <string.h>
6 #include <inarm.h>
7 #include <math.h>
8 #include <iolpc210x.h>
9 #include <stdbool.h>
10 #include "lpc210x.h"
11 #include "RpmCounterFunction.h"
12 #include "PWMcosFunction.h"
13 #include "UARTEcho.h"
14 #include "globalVariables.h"
15
16 #define N_calculationsofRpm 1
17 #define MovingAveragesamples 9 // from 2 to 10 values
18 #define Maxpointerofmatrix (MovingAveragesamples-1)
19 #define Dimension_of_mean_matrix (Maxpointerofmatrix+1)
20 #define second (PCLKFREQ/4)
21 #define ResetTimer1 T1MR0
22 #define milisecond 921.6
23 //***** for Tash function *****
24 #define Ts 1
25 #define Ki 100 //[0.000005] ->// last value before test 90 poly kalo symperifora me 45
26 #define Ki_div 10000//1000
27 #define Kp 2*45//487 //[0.005]-> // last value before test
28 #define Kp_div 1000//1000
29 #define Kd 1
30 #define Kd_div 10000//1000
31
32 long e[3],u=0;
33 unsigned long SetpointRPM = 0;
34 long RPM_Control =0;
35 unsigned int ReteTimer1 ;
36 static float tempRpm=0;
37 static unsigned long RpmCounter;
38 unsigned long RpmDataMatrix[Dimension_of_mean_matrix][2],PointertoMatrix,rpmcount;
39 static long long xronos; //Για την περίπτωση που έχουμε μεγάλες τιμές να μην
    υπερχειλήσει η μεταβλητή .
40 int N_calc=0;
41 float rpm;
42 unsigned long Loadcell;
43 bool secure = false ;
44 unsigned char status[2] ;
45 unsigned long temporary;
46 ////////////////////////////////////////////////// declare externs////////////////////////////////////
47 extern bool IarMotorStopped;
48 extern unsigned char errors ;
49 extern enum AppMode IARAppMode ;
50 extern bool PutSetPointsinBuffer ;
51 extern bool LogicDataRequest(bool SaveMode);
52 extern unsigned long SetpointPWM ;
53 extern unsigned long DutyTime ;
54 extern void SecureTrasmittaData(bool flag ,unsigned long data ,unsigned char resettimes,
    unsigned char status[] );
55 extern void RecieveTimeoutService(unsigned long Now) ;
56
57
58 unsigned long Convert4Byteto32bitInteger(unsigned long Byte1,unsigned long Byte2,
    unsigned long Byte3,unsigned long Byte4)
59 {
60 unsigned long Byte;
61
62 Byte = (Byte1<<24) | (Byte2<<16) | (Byte3<<8) | (Byte4);
63
64 return Byte;
65 }
66
67 void callbackEint0Interrupt()
68 {
69 ++RpmCounter;
70 }
71
72 ////////////////////////////////////////////////// Υπολογισμος του Rpm //////////////////////////////////////

```



```

    //////////////////////////////////////
73 //////////////////////////////////////////////////Υπολογισμος των Rpm με τη χρηση κινητου μεσου ορου //////////////////////////////////////
74
75 void callBackTimer0interrupt()
76 {
77
78     ++N_calc;
79
80     if (N_calc >= N_calculationsofRpm)
81         { //set values to status array
82
83             status[0] = ConstructControlByte(IARAppMode ,PutSetPointsinBuffer ,
            (IarMotorStopped==false)?1:0,LogicDataRequest(PutSetPointsinBuffer));
84
85             status[1] = errors ;
86             //end enumeration of status array
87             tempRpm = RPMmovingAverage( T1CR0 );
88             RPM_Control = tempRpm;
89
90             temporary = tempRpm;
91             temporary = ( (unsigned long) (temporary))|(SetpointRPM << 16);
92             // 1
93             SecureTrasmitaData(false ,temporary, ReteTimer1,status) ;
94             // 2
95             unsigned long PWM2Value = (PWMMR2-PWM_Min)*10000;
96             PWM2Value = PWM2Value / PWM_Duty ;
97             unsigned long PWM2SP = (SetpointPWM-PWM_Min)*10000;
98             PWM2SP = PWM2SP / PWM_Duty ;
99             SecureTrasmitaData(false , ( PWM2Value|(SetpointPWM<<16)),ReteTimer1,status) ; //
100         use it for normall control speed controler
101         // 3
102         SecureTrasmitaData(false ,0x03000000|Loadcell,ReteTimer1,status) ;
103         // 4
104         static unsigned long Vdata=(unsigned long) (0x04000000|0);
105         SecureTrasmitaData(false ,Vdata,ReteTimer1,status) ;
106         static unsigned long Adata = (unsigned long) (0x05000000|0) ;
107         SecureTrasmitaData(false ,Adata,ReteTimer1,status) ;
108         SecureTrasmitaData(true ,(T1TC/( milisecond)),(unsigned char) ReteTimer1,status)
109     ;
110     N_calc=0;
111 }
112
113 GkaziOrTasi();
114 }
115 //////////////////////////////////////////////////
116 //////////////////////////////////////////////////
117
118
119 void CallBackTimer1interrupt()
120 {
121     ++RpmCounter;
122     if (RpmCounter>=1000)
123     {RpmCounter = 999;}
124     RpmDataMatrix[PointertoMatrix][0]=RpmCounter;
125     RpmDataMatrix[PointertoMatrix][1]=T1CR0;
126 }
127 }
128
129 void callBackTimer1ResetInterrupt()
130 { ReteTimer1 +=1 ;
131     switch(ReteTimer1) //stop working because i can transmit only lbyte for ReteTimer1
132     {case 256:
133         InitialResetFunctions();
134         break;
135         default:
136         break;
137     }
138 }
139 }
140
141 //////////////////////////////////////////////////
    //////////////////////////////////////

```

```

142 float RpmmovingAverage(unsigned long time) // Σημείωση πρέπει το interrupt απο το
    οποίο θα υπολογιζετωαι οι στροφες να εχει χαμηλοτερη προταιρεοτητα απο του timer1
143 { switch (RpmCounter) // δεν εχει ενεργοποιηθει το Interrupt του capture μετρισεις τον
    στροφων
144 { case 0: // και περνωω δινω εγω τις τιμες στον πινακα
145 RpmDataMatrix[PointertoMatrix][0]=RpmCounter; // η πρωτη στηλη περιεχει τις
    μετρισεις των περιστροφων
146 RpmDataMatrix[PointertoMatrix][1]=time; // η 2 στηλη περιεχει την χρονικη
    στιγμη κατα την οποια συναιβει το τελαιυταιο interrupt του timer1 capture prin
    ολοκληρωθει ο υπολογισμος της ταχυτητας
147 break;
148 }
149 for (int i=0;i<MovingAveragesamples;i++)
150 {
151 rpmcount = rpmcount + RpmDataMatrix[i][0]; // Προσθετω ολες τις θεσεις του πινακα που
    περιεχουν τι μετρισεις του counter
152 }
153
154
155 switch (PointertoMatrix) // Υπολογιζω το χρονικο διαστημα μεταξυ της
    παλαιωτερης μετρισης και της τελαιυταιας .
156 {
157 case Maxpointerofmatrix : // στην περιπτωση που εχει συμπληρωθει η τελαιυταια θεση
    του πινακα και αλλαζει την θεση του δεικτη στην θεση 0 <1 θεση του πινακα>
158 rpmcount = rpmcount - RpmDataMatrix[0][0]; // και αφαιρω την παλαιωτερη μετριση ,
    με ενδιαφερον οι μετρισεις μεταξυ του τελαιυταιου capture και της παλαιωτερης
    μετρισης
159 xronos = RpmDataMatrix[Maxpointerofmatrix][1] - RpmDataMatrix[0][1];
160 PointertoMatrix=0;
161 break;
162 default : // στην περιπτωση που δεν εχει συμπληρωθει η τελαιυταια
    θεση του πινακα και αυξανεται η τιμη του δεικτη στην επομενη θεση του πινακα
163 rpmcount = rpmcount - RpmDataMatrix[PointertoMatrix+1][0]; // και αφαιρω την
    παλαιωτερη μετριση , με ενδιαφερον οι μετρισεις μεταξυ του τελαιυταιου capture και
    της παλαιωτερης μετρισης
164 xronos = RpmDataMatrix[PointertoMatrix][1] - RpmDataMatrix[PointertoMatrix+1][1]; //
    η τελαιυταια χρονικα μειων την παλαιωτερη.
165 ++PointertoMatrix;
166 break ;
167 }
168 if (xronos < 0) // case reset timer
169 { xronos = xronos + ResetTimer1 ;} // ελεγχο στην περιπτωση που ο timer1 κανει reset
    και προκυψουν αρνητικες τιμες χρονων λογο επανεκινησης του timer1
170
171
172 if ((rpmcount == 0) || (xronos <=1000)) // allxe apo ((rpmcount == 0) || (xronos ==
    0)) dioti 1000 ~ 921.6 = 1ms dhladh 60.000 rpm kai epomenos kati exei paei strava
    kai na mhn apeiristei o logos
173 { rpm = 0 ;
174 rpmcount=0;
175 RpmCounter=0;
176 return 0;
177
178 } // ελεγχοσ στην περιπτωση που εχω ταχυτητα μικροτερη απο 1 RPM .
179
180 rpm = (float) second/xronos;
181 rpm = rpmcount*rpm*60; // *100 για να εχω δυο δεκαδικα αναλιση στη μετριση στροφων
    ετσι χορισ ιδιαιτερο λογο .To επι 60 ειναι απαιαιτιτο !!
182 // rpm = rpmcount*rpm*6000 ; h synarthshs allxe apo 6000 se
    60 den moy xreiazetai o ;elegxos gia gia ta dyo dekadika
183 RpmCounter=0; //mhden: allakse oste na mhn xreiazetai //RESET μηδενισμος του
    αθροισματος
184 rpmcount=0; // μηδενισμος του δεικτη
185 return rpm ;
186
187 }
188
189 void loadcelldata2Rpmfunction(unsigned long temp)
190 {
191 Loadcell = temp ;
192 }
193
194
195 void tash ()
196 { // e[0]->e(k),e[1]->e(k-1),e[2]->e(k-2)

```

```

197  switch(IarMotorStopped)
198  {
199      case false:
200          SetpointRPM =  GetData(PutSetPointsinBuffer);
201          e[0]= SetpointRPM - RPM_Control ;
202          //u = u + ((Kp*(e[0] - e[1]))/Kp_div) + ((Ts*Ki*e[0])/Ki_div) + (((Kd*(e[0]-2*e[1]
+e[2]))/Ts)/Kd_div); //backward diference
203          u = u + ((Kp*(e[0] - e[1]))/Kp_div) + ((Ts*Ki*(e[0]+e[1]))/(2*Ki_div)) + (((Kd*(e
[0]-2*e[1]+e[2]))/Ts)/Kd_div); // trapezio
204          e[2]=e[1];
205          e[1]=e[0]; // moves e(k) to e(k-1)
206          if (u < 0 )
207          {
208              PWMset(PWM_Min);}
209          else if (u>(PWM_Duty+PWM_Min))
210          {
211              PWMset(PWM_Duty+PWM_Min);      }
212          else
213          {PWMset(u);}
214          if ( u > 100000){u=100000;}
215          else if(u < -100000) {u = -100000;}
216          break;
217
218      case true:
219          PWMset(PWM_Min);
220          break;
221
222      default :
223          break;
224  }
225 }
226 void GkaziOrTasi()
227 {
228     switch (IARAppMode)
229     {
230         case ControlSpRPM:
231             tash ();
232             break;
233
234         case ControlSpPWM:
235             Gkazi();
236             u=0;
237             break;
238
239         default:
240             break;
241     }
242
243
244 }
245
246

```

```

1
2 #define PWM_Min 0
3 #define PWM_Duty PWMMR0
4 static unsigned long kl;
5 void emergencystopmotor();
6 void InitMotorfromStopCommand();
7 enum AppMode {ControlSpRPM=0,ControlSpPWM=1,RequestControlSpRPM=2,RequestControlSpPWM=3}
;
8 void MainFunctions();
9 void InitialResetFunctions();
10 float ReturnRPM(enum AppMode Mode,long RPMCounted);
11 void PWMNullInterruptServiceRoutine();
12 void ReAllocateMyBuffer();
13 void PWMset(unsigned long dutycycle);
14 void sendbyte(unsigned char byte);
15 void byteReceived(unsigned char byte);
16 void Load(unsigned char mode, unsigned char MyChar);
17 void ReAllocateMyBuffer();
18 void Load(unsigned char mode, unsigned char MyChar);
19 void ReAllocateMyBuffer(int Offset);
20 void PWMset(unsigned long dutycycle);
21 void Modeselect();
22 void callbackEint1();
23 unsigned long ConvertCharMatrix2Int(unsigned char matrix[9]);
24 void UART1DataReceived();
25 void Uart1TransmitData();
26 void InitMotorfromStopCommand();
27 void ResetUart0Buffer();
28 void GotoStartToCloseortoStartAgain();
29 void InitialResetFunctions();
30 void PWMNullInterruptServiceRoutine();
31 void MainFunctions();
32

```

```

1
2 /*My ptyxiakh LPC2106 V1.1.32 */
3
4 #include <stdio.h>
5 #include <string.h>
6 #include <inarm.h>
7 #include <math.h>
8 #include "timer.h"
9 #include "lpc210x.h"
10 #include "iolpc210x.h"
11 #include "PWMCosFunction.h"
12 #include "UARTEcho.h"
13 #include "RpmCounterFunction.h"
14 #include "globalVariables.h"
15 #define platos 3686
16 #define FirstByte *
17 //##### define Control Incoming Integers #####
18 #define StopMotor 0xffffffff //4294967295 // #
19 #define InitMotorFromStop 0xFFFFF0FE //4294967294 // #
20 #define ChangetoSpRPMControl 0xffffffff1 // #
21 #define ChangetoSpPWMControl 0xffffffff2 // #
22 #define Change2FileSetpoints 0xffffffff3 // #
23 #define Change2ControlSetpoints 0xffffffff4 // #
24 #define CheckSerialCommConnection 0x00004E21 /// =20001 #
25 #define ResetCard 0xFFFFF0FC //4294967292 // #
26 #define StartTransmitCommPackages 0x52455345 //->"RESE" // #
27 #define CloseProgramm 0xFFFFF0FD // 4294967293 // #
28 //#####
29 #define MaxValueLoadCell 100000 //need for Loadcell recieve check
30 #define FillSizeTransfer 8 //need to UART0 Send Function
31 bool FlagGotoBeginingOfProg = false ;
32 int NoBytes2read = 0 ; //need for UART1 Recieve function
33 //int W; //W=Ω rad/s
34 int DEIKTHS = 0;
35 int BUFFER_SIZE=100;
36 unsigned char errors = 15;
37 //#####
38 //enum AppMode{ControlSpRPM=0,ControlSpPWM=1,RequestControlSpRPM=2,RequestControlSpPWM=
39 3}; //new #
40 enum AppMode IARAppMode = ControlSpRPM ;
41 //new #
42 //#####
43 bool IarMotorStopped = false ; //need for emergency stop motor
44
45 unsigned long DutyTime=1 ; //xreizetai edv oste na dhlvuei extern
46 int pointertoMRB; //need to uart1 recieve
47 unsigned char MyReceivedBuffer[6]; //need to uart0 recieve
48 unsigned char MyBuffer[100],tempu0iir; //need to uart0 transmit buffer
49 unsigned long loadcell=0 ; //Incoming loadcell data
50 //long k;
51 unsigned long DataRecieved;
52 unsigned char Uartldata[9]={'0','1','3','2','0','0','0','2','0'};
53 /////////////// declare extern ///////////////
54 extern bool InitialIARstate;
55 extern void InitialDataRecieve(unsigned char byte);
56 extern void InitialTimer0interrupt();
57 bool PutSetPointsinBuffer = false ; //need to store data to buffer
58 extern void RefreshRecieveTime(unsigned long time);
59 extern void ResetBuffer();
60 extern unsigned long SetpointPWM,SetpointRPM;
61 extern void Gkazi();
62 extern void tash ();
63
64 void Load(unsigned char mode, unsigned char MyChar);
65 void sendbyte(unsigned char byte)
66 {
67     switch (U0LSR_bit.THRE)
68     {
69         case 1:
70             U0THR=byte;
71             break;
72         case 0:
73             Load('w', byte);

```

```

71 //UOIER_bit.THREIE=1;
72 break;
73 }
74
75 }
76
77 //Byte received callback
78 void byteReceived(unsigned char byte)
79 {
80
81 switch (pointertoMRB)
82 {case 0:
83     MyReceivedBuffer[0]= byte;
84     ++pointertoMRB;
85     break;
86 case 1:
87     MyReceivedBuffer[1] = byte;
88     ++pointertoMRB;
89     break;
90 case 2:
91     MyReceivedBuffer[2] = byte;
92     ++pointertoMRB;
93     break;
94 case 3:
95     MyReceivedBuffer[3] = byte;
96     ++pointertoMRB;
97     break;
98 case 4:
99     MyReceivedBuffer[4] = byte;
100    ++pointertoMRB;
101    break;
102    case 5:
103        MyReceivedBuffer[5] = byte ;
104        if ( MyReceivedBuffer[0] == '*' && MyReceivedBuffer[5] == (MyReceivedBuffer[1]^
105        { DataRecieved = Convert4Byteto32bitInteger((unsigned long)MyReceivedBuffer[1],
106        (unsigned long) MyReceivedBuffer[2],(unsigned long) MyReceivedBuffer[3],(unsigned
107        long) MyReceivedBuffer[4]);
108        RefreshRecieveTime(T1TC);
109        switch (DataRecieved)
110        { case StopMotor : //0xffffffff:
111            emergencystopmotor(); //stamata ton PWM palmo ; einai synexeia low
112            break;
113            case InitMotorFromStop :
114            InitMotorfromStopCommand();
115            break;
116            case StartTransmitCommPackages :
117            InitialIARstate = false ;
118            MainFunctions();
119            break ;
120            case CloseProgramm: //send the programm at the start like softreset
121            InitialIARstate = true ;
122            FlagGotoBeginingOfProg = true;
123            InitialResetFunctions();
124            break;
125            case ChangetoSpRPMControl : //0xffffffff1: //elegxos strofon
126            IARAppMode = ControlSpRPM ; //sets mode to control rpm
127            // koympi mode gkazi
128            VICSoftInt_bit.INT15 = 1; //sets EINT1 software interrupt
129            break;
130            case ChangetoSpPWMControl : //0xffffffff2: //elegxos gkazioy
131            IARAppMode = ControlSpPWM ; //sets mode to control pwm (gkazi)
132            // koympi mode elegxos strofon
133            VICSoftInt_bit.INT15 = 1; //sets EINT1 software interrupt
134            break;
135            case Change2FileSetpoints :
136            PutSetPointsinBuffer = true ; // changes to read set points from buffer
137            ResetBuffer() ;
138            break;
139            case Change2ControlSetpoints :
140            PutSetPointsinBuffer = false ; // changes to read set points from buffer[0]
141            ;Running condition no fifo control on buffer
142            ResetBuffer() ;
143            break;
144            case CheckSerialCommConnection :

```

```

142         //check serial comm for timeout event ,Discard DataRecieved long
143         break;
144         default :
145         StoreData(DataRecieved,PutSetPointsinBuffer); //store data to buffer in fifo
mode or to [0] according to setpoint mode
146         break;
147     }
148
149     }
150     else
151     {
152         for (int i=0;i<5;++i)
153         { MyReceivedBuffer[i]=MyReceivedBuffer[i+1];
154           pointertoMRB=5;
155         }
156         // goto jump ;
157         break;
158     }
159
160     pointertoMRB=0;
161     break; //jump
162
163     default :
164     break;
165 }
166 tempu0iir = U0IIR_bit.IID;
167
168
169 }
170
171 void Load(unsigned char mode, unsigned char MyChar)
172 {
173     unsigned char ToBeReturned;
174     switch (mode)
175     { static int NoofByteSents;
176       case 'r':
177         NoofByteSents = DEIKTHS>FillSizeTransfer?FillSizeTransfer:DEIKTHS ;
178
179         if (DEIKTHS > 0)
180         { for (int i = 0; i<=NoofByteSents-1;i++){
181           ToBeReturned = MyBuffer[i];
182           U0THR = ToBeReturned;
183           }//end for
184         ReAllocateMyBuffer(NoofByteSents);
185         DEIKTHS=DEIKTHS - NoofByteSents ;
186         }//end if
187
188         if (DEIKTHS <= 0)
189         {
190           tempu0iir=U0IIR_bit.IID;// read U0iir
191         }
192         break;
193       case 'w':
194         MyBuffer[DEIKTHS] = MyChar;
195         if (DEIKTHS >= 99)
196         {DEIKTHS = 98;}
197         DEIKTHS++;
198         break;
199       default:
200         break;
201     }
202 }
203 }
204
205 void ReAllocateMyBuffer(int Offset)
206 {
207     int i;
208     for (i= Offset ; i <= DEIKTHS + Offset-1; i++)
209     { MyBuffer[i- Offset] = MyBuffer[i];}
210 }
211 }
212
213
214
215

```

```

216
217 void PWMset(unsigned long dutycycle)
218 {
219     if (dutycycle>=PWMMR0)
220     {dutycycle = PWMMR0;}
221
222     PWMMR2=dutycycle; // Set duty cycle
223     PWMLER_bit.EM2L=1; //Enables to set new value of dutycycle
224
225 }
226
227 void Modeselect()
228 {
229     ResetBuffer();
230     SetpointPWM = PWM_Min ;
231     SetpointRPM=0;
232     PWMset(PWM_Min);
233 }
234
235
236 // callback EINT1 pushbuton
237 void callbackEint1()
238 {
239     if (IARAppMode == ControlSpRPM)
240     { // throll control
241         IARAppMode = RequestControlSpPWM ; }
242     else if(IARAppMode == ControlSpPWM)
243     { //Rpm control default
244         IARAppMode = RequestControlSpRPM ; }
245 }
246 }
247
248
249 unsigned long ConvertCharMatrix2Int(unsigned char matrix[9])
250 { unsigned int convetionInt[9];
251   for (int i=0;i<7;i++)
252     { convetionInt[i]= matrix[i] - '0';}
253   loadcell=0;
254   loadcell += ((unsigned int) convetionInt[0])*100000 ;
255   loadcell += ((unsigned int) convetionInt[1])*10000 ;
256
257   loadcell += ((unsigned int) convetionInt[3])*1000 ;
258   loadcell += ((unsigned int) convetionInt[4])*100 ;
259   loadcell += ((unsigned int) convetionInt[5])*10 ;
260   loadcell += ((unsigned int) convetionInt[6])*1 ;
261
262   loadcell = ( loadcell <= MaxValueLoadCell) ? loadcell : 0 ;
263   return loadcell ;
264 }
265
266 void UART1DataReceived()
267 { unsigned long temp;
268
269   Uart1data[ NoBytes2read ]=U1RBR;
270   ++NoBytes2read ;
271   switch ( NoBytes2read )
272   {case 9:
273     if ( Uart1data[2]== '.' && Uart1data[7]== 10 && Uart1data[8]== 13 )
274     { temp = ConvertCharMatrix2Int(Uart1data) ;
275       loadcelldata2Rpmfunction(temp);
276       NoBytes2read = 0 ;
277     }
278     else
279     { for (int i=0;i<8;i++)
280       { Uart1data[i]=Uart1data[i+1]; }
281       NoBytes2read -= 1;
282     }
283     break;
284     default:
285     break;
286   }
287   temp = U1IIR_bit.IID;
288 }
289
290 void Uart1TransmitData ()

```



```

291 {}
292 void emergencystopmotor()
293 {
294
295     PWMset(PWM_Min);
296     IarMotorStopped = true ;
297
298 }
299 void InitMotorfromStopCommand()
300 {
301     PWMset(PWM_Min); //use it for motor control //
302     IarMotorStopped = false ;
303     LPC210xSTARTPWM() ; //starts pwm end execute and the pwm interrupts
304 }
305 void ResetUart0Buffer()
306 { DEIKTHS = 0;
307     for( int i=0;i==99;i++)
308         { MyBuffer[i]=0; }
309 }
310 void GotoStartToCloseortoStartAgain() //reset appropriate fields
311 {
312     LPC210xStopResetUART();
313     RefreshRecieveTime(0); //reest time receive
314     LPC210xResetStopTimer1();
315     LPC210xResetStopTimer();
316     LPC210xStopResetUART1();
317     LPC210xSTOPPWM();
318
319     InitialIARstate = true;
320 }
321
322
323 void InitialResetFunctions()
324 {
325     LPC210xInitPIO(); //sets pins 0,15 as gpio
326     GotoStartToCloseortoStartAgain();
327     ////////////////////////////////////////////////////////////////////----- initial state control-----
328     ////////////////////////////////////////////////////////////////////
329     LPC210xInitUART0();
330     LPC210xInitTimer();
331     LPC210xInitPWM();
332     __disable_interrupt();
333
334     LPC210xInitVIC();
335
336     LPC210xInitUART0Interrupt(byteReceived,Load);
337     LPC210xInitTimerInterruptAtBegin(InitialTimer0interrupt,RecieveTimeoutService);
338     ResetUart0Buffer();
339     RefreshRecieveTime(0); //reest time receive
340
341     __enable_interrupt();
342     LPC210xStartTimer();
343     LPC210xSTARTPWM();
344 }
345 void PWMNullInterruptServiceRutine()
346 {
347 }
348
349 void MainFunctions()
350 {
351     PutSetPointsinBuffer = false ; // changes to read set points from buffer[0] ;Running
352     condition no fifo control on buffer
353     ResetBuffer() ; //
354     ////////////////////////////////////////////////////////////////////
355     LPC210xResetStopTimer(); // will stay reset untill initialation of timer0
356     ////////////////////////////////////////////////////////////////////-----end initial state -----
357     ////////////////////////////////////////////////////////////////////
358     //Set up UART
359     LPC210xInitUART0();
360     LPC210xInitUART1();
361     LPC210xInitTimer();
362     LPC210xInitTimer1();

```

```

361
362     LPC210xInitPWM();
363     LPC210xInitEINT1();
364 // First disable interrupts.
365
366     __disable_interrupt();
367 // Setup interrupt controller.
368     LPC210xInitVIC();
369
370     LPC210xInitUART0Interrupt(byteReceived,Load);
371     LPC210xInitUART1Interrupt(UART1DataReceived,Uart1TransmitData);
372     LPC210xInitTimerInterrupt(callBackTimer0interrupt,RecieveTimeoutService);
373     LPC210xInitTimer1Interrupt(CallBackTimer1interrupt,callBackTimer1ResetInterrupt);
374     LPC210xinitPWMIInterrupt(PWMNullInterruptServiceRoutine);
375
376     IARAppMode = ControlSpPWM ;
377 // Setup external interrupt, used for button.
378     LPC210xInitExternalInterrupt1(callbackEint1,Modeselect);
379     RefreshRecieveTime(0); //reest time receive
380 // Enable interrupts.
381     __enable_interrupt();
382
383     Load('w','s');
384     Load('w','t');
385     Load('w','a');
386     Load('w','r');
387     Load('w','t');
388
389     LPC210xStartTimer();
390     RefreshRecieveTime(0); //reest time receive
391     LPC210xStartTimer1();
392     LPC210xSTARTPWM(); //start the PWM2
393     FlagGotoBeginingOfProg = false ;
394
395 }
396
397 //***** MAIN *****
398 void main(void)
399 {
400 // System initialization, this will map the exception vectors.
401     LPC210xSystemInit();
402     InitialResetFunctions();
403     for(;;)
404     {
405     }
406
407 }
408

```