



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΗΣ ΙΣΧΥΟΣ

**Έλεγχος και Λειτουργία Ημιαυτόματης Γεμιστικής
Μηχανής με Προγραμματιζόμενο Λογικό Ελεγκτή**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Τσοπανάκης Σπυρίδων-Πύρρος

Επιβλέπων : Γεώργιος Ν. Κορρές
Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2014



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΗΣ ΙΣΧΥΟΣ

Έλεγχος και Λειτουργία Ημιαυτόματης Γεμιστικής Μηχανής με Προγραμματιζόμενο Λογικό Ελεγκτή

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Τσοπανάκης Σπυρίδων-Πύρρος

Επιβλέπων : Γεώργιος Ν. Κορρές
Αν. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την :

.....
Γ. Κορρές
Αν. Καθηγητής

.....
Π. Γεωργιάκη
Λέκτορας

.....
Κ. Βουρνάς
Καθηγητής

Αθήνα, Μάρτιος 2014

.....

Τσοπανάκης Σπυρίδων-Πύρρος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Τσοπανάκης Σπυρίδων-Πύρρος

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας είναι η κατασκευή ενός πίνακα αυτοματισμού για τη σωστή λειτουργία και τον έλεγχο μιας ημιαυτόματης γεμιστικής μηχανής για υγρά.

Ο αυτοματισμός περιλαμβάνει τη χρήση προγραμματιζόμενου λογικού ελεγκτή (PLC). Όλος ο έλεγχος υλοποιείται μέσω αυτού. Παράλληλα περιλαμβάνεται οθόνη αφής HMI για την καλύτερη επικοινωνία ανθρώπου-μηχανής, δυναμοκυψέλη για την μέτρηση του βάρους και ενδείκτης βάρους για την αξιοποίηση του βάρους από την δυναμοκυψέλη. Εν συνεχεία με βάση το πρόγραμμα που είναι αποθηκευμένο στη μνήμη του PLC, ο χειριστής θα επιλέγει από την HMI την ποσότητα (με βάση το βάρος) που επιθυμεί να συσκευάσει ανά δοχείο, η μηχανή θα τροφοδοτεί το δοχείο με την κατάλληλη ποσότητα και μετά το τέλος των επιθυμητών πληρώσεων ο χειριστής θα έχει τη δυνατότητα εκτύπωσης κάποιων στατιστικών στοιχείων.

Εκτός του πίνακα αυτοματισμού, παρατίθενται κάποιες εισαγωγικές πληροφορίες για τη λειτουργία των προγραμματιζόμενων λογικών ελεγκτών καθώς επίσης και κάποιων βασικών πρωτοκόλλων και προτύπων επικοινωνίας, συγκεκριμένα για το RS-232, το RS-485 και το Profibus. Τα πρωτόκολλα επικοινωνίας χρησιμοποιήθηκαν και για την ανταλλαγή δεδομένων στον αυτοματισμό της γεμιστικής μηχανής.

Λέξεις Κλειδιά: Προγραμματιζόμενος Λογικός Ελεγκτής (PLC), Διεπαφή Ανθρώπου Μηχανής (HMI), Γεμιστική Μηχανή, Πρότυπα και Πρωτόκολλα Επικοινωνίας, Λογισμικό TIA Portal

Abstract

The scope of this thesis was to construct an automation panel for the use and control of a semi-automatic filling machine for liquids.

For the above automation a programming logic controller (PLC) is used. The whole control of the filling machine is realized through this PLC, a human machine interface (HMI) is concluded for better communication between operator and filling machine, a loadcell for the weighing and a weight indicator. Then, depending on the program that the programmable logic controller runs in its memory, the operator is going to choose from the HMI the weight which he wants to fill per bottle, the filling machine will fill the bottle with the appropriate weight and finally the operator can print some statistical data.

Apart from the automation panel, in this thesis is mentioned basic information about PLCs and about basic communication standards and protocols, specifically for RS-232, RS-485 and Profibus. These standards and protocols have used been for the exchange data on the above automation panel.

Keywords: Programming Logic Controller (PLC), Human Machine Interface (HMI), Filling Machine, Communication Standards, Communication Protocols, Software TIA Portal

Περιεχόμενα

Κεφάλαιο 1: Εισαγωγή στους προγραμματιζόμενους λογικούς ελεγκτές.....	9
1.1 Ιστορική αναδρομή στα PLC.....	9
1.2 Ορισμός του PLC.....	10
1.3 Η δομή ενός PLC.....	11
1.3.1 Μονάδα τροφοδοσίας.....	13
1.3.2 Κεντρική μονάδα επεξεργασίας CPU.....	13
1.3.2.1 Από τι αποτελείται η CPU.....	13
1.3.2.2 Τα είδη μνήμης της CPU.....	14
1.3.3 Μονάδες εισόδου/εξόδου.....	15
1.4 Τρόπος σύνδεσης των βασικών μερών του PLC.....	16
1.5 Πλαίσιο τοποθέτησης μονάδων.....	17
1.6 Συσκευές προγραμματισμού ενός PLC.....	18
1.6.1 Προγραμματιστής χειρός.....	18
1.6.2 PG.....	19
1.6.3 Προσωπικός Υπολογιστής (PC).....	19
1.6.4 Φωτεινή πένα.....	20
1.7 Κύριες λειτουργίες των PLC.....	20
1.8 Πλεονεκτήματα και Μειονεκτήματα χρήσης ενός PLC.....	20
Κεφάλαιο 2: Αρχή λειτουργίας και τρόπος προγραμματισμού ενός PLC.....	22
2.1 Κύκλος λειτουργίας του PLC.....	22
2.2 Τρόποι προγραμματισμού ενός PLC.....	23
2.2.1 Γραμμικός προγραμματισμός (linear program).....	23
2.2.2 Δομημένος Προγραμματισμός.....	24
2.3 Βασικές αρχές σχεδίασης προγράμματος.....	24
2.3.1 Λειτουργικό σύστημα.....	24
2.3.2 Πρόγραμμα εφαρμογής.....	25
2.3.3 Επικοινωνία λειτουργικού συστήματος και προγράμματος εφαρμογής.....	25
2.3.4 Είδη blocks.....	25
2.3.4.1 Organization blocks (OBs).....	26
2.3.4.2 Functions (FCs).....	28
2.3.4.3 Data Blocks (DBs).....	28

2.3.4.4	Function Blocks (FBs).....	29
2.4	Γλώσσα προγραμματισμού ενός PLC.....	29
2.4.1	STL (statement list).....	30
2.4.2	Ladder	30
2.4.3	FBD (function block diagram).....	30
2.5	Εισαγωγή στις HMI (Human Machine Interface).....	30
Κεφάλαιο 3: Πρότυπα και πρωτόκολλα επικοινωνίας.....		31
3.1	Εισαγωγή.....	31
3.2	Πρότυπο RS-232	31
3.3	Πρότυπο RS-485	33
3.3	Πρωτόκολλο Profibus	34
Κεφάλαιο 4: TIA PORTAL		41
4.1	Εισαγωγή.....	41
4.2	Δημιουργία προγράμματος & εισαγωγή PLC.....	41
4.4	Βασικές δυνατότητες του Project view	52
Κεφάλαιο 5: Ανάλυση προγράμματος γεμιστικής μηχανής		59
5.1	Τα μέρη από τα οποία αποτελείται η γεμιστική μηχανή.....	59
5.2	Δημιουργία προγράμματος με το TIA PORTAL	63
5.3	Δημιουργία των tag tables σε PLC και HMI.....	65
5.4	Τα Data Blocks του προγράμματος.....	66
5.5	Διαμόρφωση των Screens της HMI.....	69
5.6	Ρυθμίσεις για την σύνδεση μέσω Profibus	80
5.7	Ανάλυση προγράμματος στο OB1	83
Κεφάλαιο 6: Συμπεράσματα – Παρατηρήσεις.....		102
Βιβλιογραφία.....		103

ΕΙΣΑΓΩΓΗ

Αντικείμενο της παρούσας εργασίας αποτελεί μία εισαγωγή στα PLC καθώς και μία αναλυτική μελέτη για τον έλεγχο και τη λειτουργία μίας ημιαυτόματης γεμιστικής μηχανής υγρών υλικών με χρήση ενός PLC S7 1200 της εταιρείας Siemens.

Στο Κεφάλαιο 1 γίνεται ιστορική αναδρομή στον τρόπο με τον οποίο τα PLC εισχώρησαν στην αγορά εργασίας της βιομηχανίας. Γίνεται ανάλυση των υποσυστημάτων που απαρτίζουν ένα PLC καθώς και τι χρειάζεται για να λειτουργήσει. Τέλος αναφέρονται κάποια πλεονεκτήματα και μειονεκτήματα χρήσης των PLC.

Στο Κεφάλαιο 2 πραγματοποιείται αναλυτική επεξήγηση της αρχής λειτουργίας των PLC καθώς και κάποιοι τρόποι προγραμματισμού τους. Στο τέλος του κεφαλαίου γίνεται μια μικρή εισαγωγή για την επικοινωνία ανθρώπου-μηχανής (HMI).

Στο Κεφάλαιο 3 αναλύονται τα πρότυπα επικοινωνίας RS-232, RS-485 και το πρωτόκολλο επικοινωνίας Profibus.

Στο Κεφάλαιο 4 πραγματοποιείται μια εισαγωγή στο λογισμικό TIA Portal. Το λογισμικό αυτό είναι προϊόν της Siemens και χρησιμοποιείται κυρίως για τον προγραμματισμό λογικών ελεγκτών.

Στο κεφάλαιο 5 παρατίθεται το πρόγραμμα ελέγχου της γεμιστικής μηχανής, με χρήση του περιβάλλοντος TIA Portal.

Τέλος παρατίθενται ορισμένα συμπεράσματα που εξάγονται από την υλοποίηση του πίνακα ελέγχου της γεμιστικής μηχανής.

Κεφάλαιο 1: Εισαγωγή στους προγραμματιζόμενους λογικούς ελεγκτές



Εικόνα 1.1 : PLC , οθόνες αφής και PC

1.1 Ιστορική αναδρομή στα PLC

Οι πρώτοι αυτοματισμοί ήταν καθαρά μηχανικοί από γρανάζια και μοχλούς. Με τον ηλεκτρισμό βέβαια τα πράγματα άλλαξαν. Στους πρώτους ηλεκτρονικούς αυτοματισμούς το βασικό στοιχείο ήταν η λυχνία. Το 1945 κατασκευάζεται ο πρώτος ηλεκτρονικός υπολογιστής βασισμένος κυρίως σε λυχνίες.

Οι πρώτες σκέψεις για την αυτοματοποίηση της βιομηχανίας εμφανίστηκαν περίπου το 1960. Παρ' όλα αυτά η επανάσταση στην τεχνολογία ξεκινάει στις αρχές του 1980 με την κατασκευή του πρώτου μικροεπεξεργαστή. Με αυτό το τρόπο δημιουργείται η σκέψη εισαγωγής κάποιας μορφής υπολογιστικής μονάδας στη βιομηχανία για τον έλεγχο όλων των διαδικασιών της παραγωγής.

Στις αρχές της δεκαετίας του 1980 πρωτοεμφανίζεται μία μονάδα αυτοματισμού με την ονομασία PLC, με στόχο τον έλεγχο και την επιτήρηση των ανωτέρω. Η πλήρης ονομασία του PLC είναι Programmable Logic Controller που σημαίνει «Προγραμματιζόμενος Λογικός Ελεγκτής».

Αρχικά η βασική λειτουργία των PLC ήταν να αντικαταστήσουν τους πίνακες με ηλεκτρονόμους, κάτι που ήταν αρκετά δύσκολο για τις εταιρείες που τα κατασκεύαζαν, λόγω του ότι οι άνθρωποι της βιομηχανίας δεν ήταν συνηθισμένοι στη χρήση ενός «ηλεκτρονικού υπολογιστή». Για αυτό το λόγο προσπάθησαν να προσαρμόσουν τον τρόπο χρήσης των PLC σε ένα παρόμοιο τρόπο με τον οποίο δούλευε μέχρι τότε η βιομηχανία. Αυτός ήταν στο να σχεδιάζουν τα ηλεκτρικά κυκλώματά και να αντιγράφουν το ηλεκτρολογικό σχέδιο στο PLC. Με αυτό το τρόπο κατάφεραν οι εταιρείες να τα εισάγουν στην αγορά, και σήμερα να αποτελούν αναπόσπαστο κομμάτι της βιομηχανίας και άκρως απαραίτητο για τη λειτουργία της.

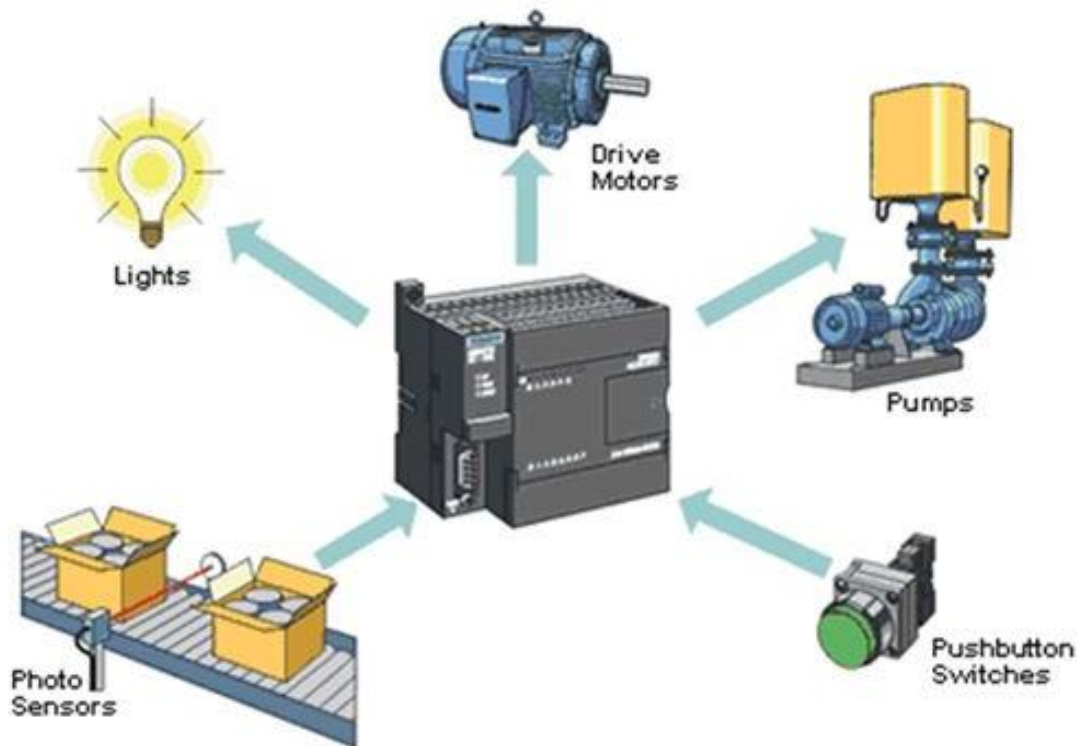
1.2 Ορισμός του PLC

Στις αρχές τα PLC αντικαθιστούσαν στον κλασικό αυτοματισμό τα βοηθητικά ρελέ, τα χρονικά, τους απαριθμητές, και όλες αυτές οι πολύπλοκες συνδέσεις πραγματοποιούνταν απλά «προγραμματίζοντας» σε ένα software.

Λόγω μεγάλου ανταγωνισμού δημιουργήθηκε η ανάγκη από μεριά των εταιρειών να αυτοματοποιήσουν εκτός από τις μηχανές και τους σταθμούς επεξεργασίας, ολόκληρη την παραγωγική τους διαδικασία ενσωματώνοντας το κάθε υποσύστημα-μηχανή σε ένα ολοκληρωμένο σύστημα. Έτσι σήμερα μέσω των PLC είναι δυνατή η διαμόρφωση και ο προγραμματισμός όλων των επιμέρους συστημάτων σαν ένα ενιαίο σύστημα, η κοινή διαχείριση δεδομένων καθώς και η επικοινωνία όλων των υποσυστημάτων που συμμετέχουν στον αυτοματισμό.

Ένας ορισμός για το PLC είναι ο εξής:

«Προγραμματιζόμενος λογικός ελεγκτής είναι ένα, ψηφιακό ηλεκτρονικό σύστημα, σχεδιασμένο για χρήση σε βιομηχανικό περιβάλλον, το οποίο χρησιμοποιεί μια προγραμματιζόμενη μνήμη για την αποθήκευση εντολών ώστε να εκτελούνται διάφορες λειτουργίες όπως λογικές, χρονικές, μετρητικές και αριθμητικές πράξεις και να ελέγχονται μέσω αναλογικών/ψηφιακών μονάδων διάφορες μηχανές ή διαδικασίες.»



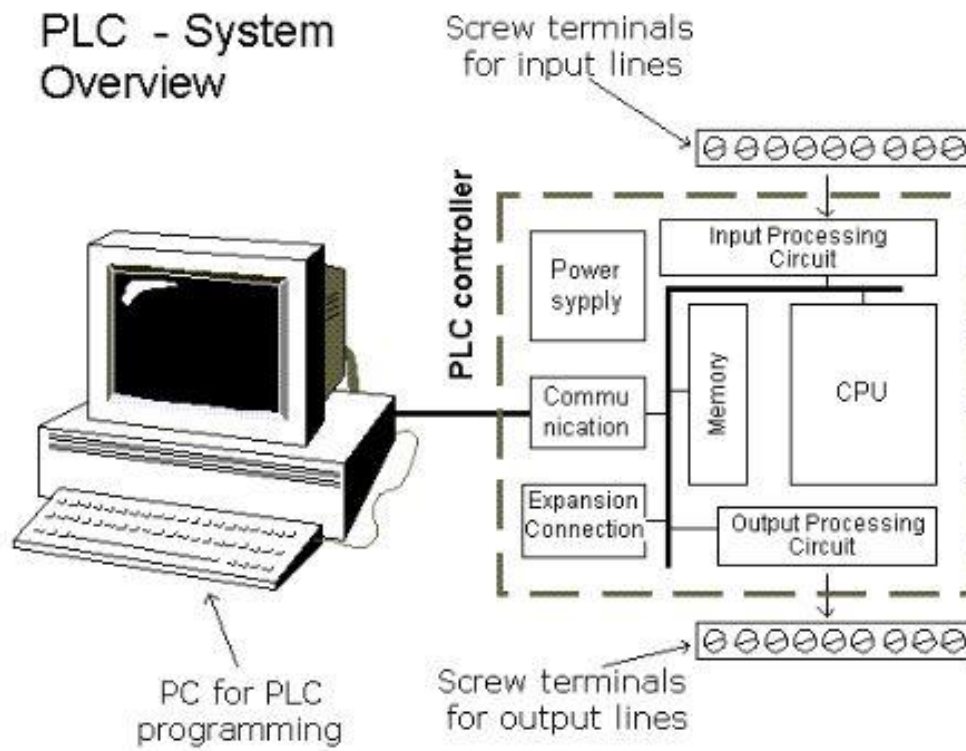
Εικόνα 1.2 : Πιθανοί είσοδοι και έξοδοι ενός PLC

1.3 Η δομή ενός PLC

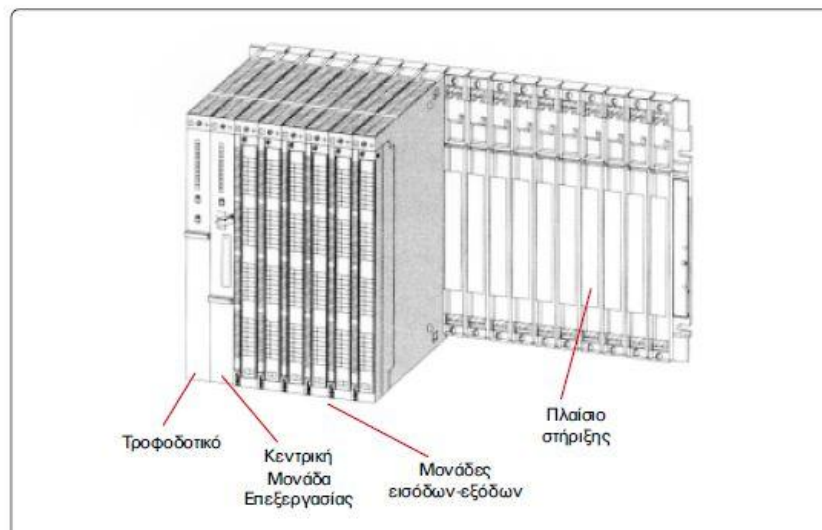
Τα βασικά μέρη ενός PLC είναι τα εξής:

- Η κεντρική μονάδα επεξεργασίας (CPU)
- Οι μονάδες εισόδων-εξόδων (I/O)
- Η μνήμη του επεξεργαστή
- Η μονάδα τροφοδοσίας

PLC - System Overview



Εικόνα 1.3: Τα βασικά μέρη που απαρτίζουν ένα PLC



Εικόνα 1.4 : PLC πάνω σε κατάλληλο πλαίσιο στήριξης

1.3.1 Μονάδα τροφοδοσίας

Η μονάδα τροφοδοσίας παρέχει στο PLC την κατάλληλη τάση λειτουργίας του, μετατρέποντας την τάση τροφοδοσίας του δικτύου στην επιθυμητή τιμή. Συνήθως αυτή η τάση είναι συνεχής στα 5V DC, 12V DC ή 24V DC λόγω του ότι σε τέτοιες τάσεις λειτουργούν τα ηλεκτρονικά στοιχεία ενός PLC.

Η μονάδα τροφοδοσίας επιλέγεται έτσι ώστε το ονομαστικό της ρεύμα να είναι μεγαλύτερο από το ρεύμα που καταναλώνουν αθροιστικά όλες οι υπόλοιπες μονάδες του PLC.

Σχετικά με την τροφοδοσία του PLC θα πρέπει να τηρούνται τα εξής:

- Το κάλυμμα του τροφοδοτικού να είναι κατάλληλο ώστε να απάγεται η θερμοκρασία που αναπτύσσεται και να αποφεύγεται η υπερθέρμανση.
- Η τροφοδοσία να πληρεί πρότυπα σύμφωνα με τα οποία επέρχεται απομόνωση από τον ανεπιθύμητο θόρυβο.

1.3.2 Κεντρική μονάδα επεξεργασίας CPU

Η κεντρική μονάδα επεξεργασίας ελέγχει και εκτελεί όλες τις λειτουργίες του PLC. Είναι στην ουσία ένας μικρουπολογιστής και διαθέτει μικροεπεξεργαστή και μνήμη.

Η σχεδίαση της CPU καθορίζει το αν μπορεί το PLC να επεκταθεί ή να τροποποιηθεί σε μελλοντική αναβάθμιση, καθώς και την ταχύτητά του.

Η ταχύτητα εκφράζεται από τον ρυθμό σάρωσης κάποιας δεδομένης περιοχής μνήμης. Όσο μεγαλύτερη η ταχύτητα τόσο ανεβαίνει και το κόστος αγοράς του PLC.

1.3.2.1 Από τι αποτελείται η CPU

Η CPU αποτελείται από τα εξής τμήματα:

- Αριθμητική μονάδα (ALU-Arithmetic Logical Unit): Αποτελείται από δύο ή τέσσερις κεντρικούς καταχωρητές στους οποίους αποθηκεύονται οι πληροφορίες από τις εισόδους και τις εξόδους. Επίσης σ' αυτή εκτελούνται οι λογικές και οι αριθμητικές πράξεις.
- Επεξεργαστής(Processor): Καλεί στη μνήμη και εκτελεί τις εντολές του προγράμματος με τη σειρά που είναι δομημένες. Αυτό πραγματοποιείται συνέχεια και με κυκλική διαδικασία, την οποία και θα μελετήσουμε αργότερα. Παράλληλα υπολογίζει τα χρονικά, τους απαριθμητές και τις βοηθητικές μονάδες από τις μνήμες, ενώ επεξεργάζεται και τα σήματα από τις εισόδους
- Λειτουργικό σύστημα (ROM): Περιέχει τα system programs (προγράμματα του συστήματος), τα οποία δε μπορούν να επηρεαστούν

και να αλλαχτούν από το χρήστη. Καθορίζουν τον τρόπο εκτέλεσης του προγράμματος ελέγχου, το πώς κατανέμεται η μνήμη, το πώς διακινούνται τα δεδομένα και πώς αντιμετωπίζονται τα σήματα εισόδων και εξόδων

- Χρονικά, απαριθμητές, βοηθητικά: Περιοχή της μνήμης RAM όπου περιέχονται τα χρονικά, οι απαριθμητές και τα βοηθητικά
- Μνήμη απεικόνισης εισόδων: Περιοχή της μνήμης RAM όπου αποθηκεύονται οι εισοδοί του PLC
- Μνήμη απεικόνισης εξόδων: Περιοχή της μνήμης RAM όπου αποθηκεύονται οι έξοδοι του PLC όπως διαμορφώθηκαν από την αλληλουχία του προγράμματος

1.3.2.2 Τα είδη μνήμης της CPU

Η μνήμη της κεντρικής μονάδας διακρίνεται σε μνήμη RAM, ROM και EEPROM.

Μνήμη RAM: Στη μνήμη RAM (Random Access Memory) αποθηκεύεται το πρόγραμμα ελέγχου. Λόγω του ότι στο πρόγραμμα ελέγχου θέλουμε οποιαδήποτε στιγμή να κάνουμε εύκολα αλλαγές, η μνήμη RAM μας δίνει τη δυνατότητα να γράφουμε και να σβήνουμε. Βέβαια είναι πτητική, δηλαδή σβήνεται το περιεχόμενό της όταν σταματήσει η τροφοδοσία και γι' αυτό το λόγο παραμένει πάντα σε τροφοδοσία μέσω μιας μπαταρίας.

Χωρίζεται σε 3 κομμάτια:

- Μνήμη φόρτωσης (load memory): Περιλαμβάνει αντικείμενα που δημιουργήθηκαν στη συσκευή προγραμματισμού όπως μπλοκ λογικής, μπλοκ δεδομένων και επιπρόσθετες πληροφορίες.
- Μνήμη εργασίας (Work memory): Περιλαμβάνει δεδομένα που είναι απαραίτητα για την εκτέλεση του προγράμματος
- Μνήμη συστήματος: Περιλαμβάνει περιοχές για
 - Απεικόνιση εισόδων και εξόδων (PII, PIQ)
 - Βοηθητικά
 - Χρονικά
 - Απαριθμητές
 - L στοίβα

Μνήμη EEPROM (electrically programmable read only memory): Είναι μνήμη μόνο ανάγνωσης και είναι μη πτητική και για αυτό το λόγο χρησιμοποιείται ως μόνιμο μέσο αποθήκευσης του προγράμματος. Είναι ασφαλέστερο μέσο αποθήκευσης για βιομηχανικές συνθήκες. Βέβαια ο αριθμός των τροποποιήσεων των αποθηκευμένων πληροφοριών είναι περιορισμένος σε αντίθεση με τη μνήμη RAM που είναι απεριόριστος.

Μνήμη ROM: Σε αυτή τη μνήμη ο κατασκευαστής του PLC αποθηκεύει το λειτουργικό σύστημα, που περιλαμβάνει τις οδηγίες για όλες τις βασικές λειτουργίες που απαιτούνται για να λειτουργήσει το PLC.

1.3.3 Μονάδες εισόδου/εξόδου

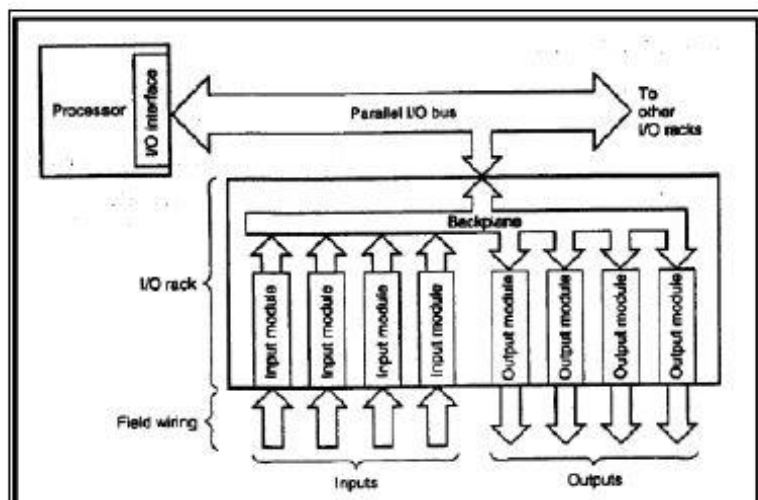
Οι μονάδες εισόδου/εξόδου αποτελούν τις μονάδες επικοινωνίας της κεντρικής μονάδας με τον εξοπλισμό που δίνει πληροφορίες στο PLC όπως αισθητήρες, φωτοκύτταρα, επαγωγικά αλλά και με τον εξοπλισμό στον οποίο δίνει εντολές το PLC όπως κινητήρες και έμβολα.

Η κεντρική μονάδα δέχεται ψηφιακά σήματα εισόδου και εξόδου χαμηλής τάσης και πολύ μικρού ρεύματος. Για αυτό το λόγο οι μονάδες εισόδων και εξόδων αναλαμβάνουν να προσαρμόσουν αυτά τα σήματα σε τιμές που να μπορεί να τις δεχτεί η κεντρική μονάδα, τόσο από άποψη τάσεως όσο και από άποψη ρεύματος χρησιμοποιώντας τρανζίστορ ισχύος, θυρίστορ, triac αλλά και κατάλληλους μικροηλεκτρονόμεους.

Τα συστήματα εισόδων και εξόδων διακρίνονται στις εξής κατηγορίες:

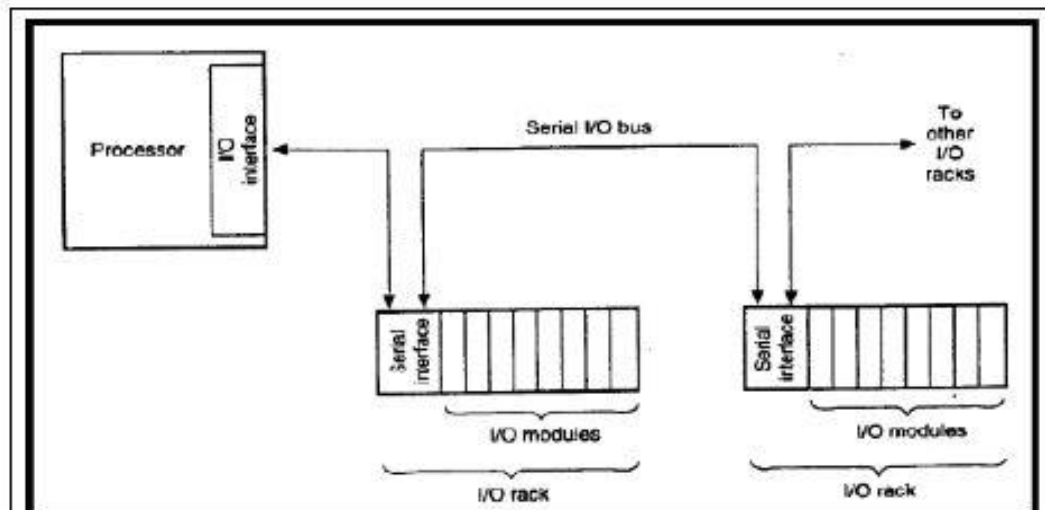
- Άμεσο σύστημα I/O: Διατίθεται ένα σήμα εισόδου και ένα σήμα εξόδου, τα οποία αντιστοιχούν στο σύνολο εισόδων και εξόδων που υποστηρίζει ο επεξεργαστής. Είναι οικονομικότερο από τα άλλα συστήματα αλλά χρησιμοποιείται μόνο για μικρά PLC.

- Παράλληλο σύστημα: Υπάρχει ένας δίαυλος που ξεκινάει από το τμήμα εισόδων-εξόδων του επεξεργαστή, ενώ ανεξάρτητες μονάδες εισόδου-εξόδου είναι τοποθετημένες στο δίαυλο αυτό. Οι μονάδες αυτές περιλαμβάνουν κατάλληλα κυκλώματα για την αποκωδικοποίηση των σημάτων του διαύλου και την μετατροπή τους σε επίπεδα τάσης τα οποία μπορούν να οδηγήσουν τα φορτία.



Εικόνα 1.5: Παράλληλο σύστημα

- Σειριακό σύστημα: Έχει το πλεονέκτημα ότι μπορεί να μεταδώσει δεδομένα εισόδου-εξόδου σε μεγάλες αποστάσεις κάτι το οποίο δεν υπάρχει στο παράλληλο σύστημα αφού έχει περιορισμό επέκτασης του διαύλου εισόδου-εξόδου. Βέβαια ένα μειονέκτημα του σειριακού συστήματος είναι ότι πολλά από αυτά μπορούν να αποσυγχρονιστούν από τη λογική σάρωση εισόδων-εξόδων και ότι σε κρίσιμες εφαρμογές χρόνου είναι πιο αργό από το παράλληλο.



Εικόνα 1.6: Σειριακό σύστημα

1.4 Τρόπος σύνδεσης των βασικών μερών του PLC

Υπάρχουν δύο τρόποι συνδέσεις των βασικών μερών που απαρτίζουν ένα PLC.

- Ο πρώτος τρόπος σύνδεσης των PLC είναι όταν οι τρεις παραπάνω μονάδες είναι ενσωματωμένες σε μία και ονομάζονται compact PLC. Συνήθως compact είναι τα μικρά PLC, κατάλληλα για μικρές εφαρμογές, εύκολα στην αντικατάσταση αλλά συνήθως μη επισκευάσιμα. Τις περισσότερες φορές οι εισοδοι και οι εξοδοι είναι το πολύ μέχρι 20 σε πλήθος και δεν υπάρχει δυνατότητα αλλαγής του τύπου των εισόδων ή των εξόδων. Μερικά σου δίνουν και τη δυνατότητα επέκτασης μερικών εισόδων ή εξόδων επιπλέον.



Εικόνα 1.7: Compact PLC

- Ο δεύτερος τρόπος σύνδεσης είναι όταν κάθε μονάδα είναι ξεχωριστό κομμάτι και συνδέεται εν συνεχεία με τις υπόλοιπες και ονομάζονται modular PLC. Η δυνατότητα σύνδεσης και προσαρμογής των μονάδων με βάση τις ανάγκες του προγραμματιστή τα κάνει πολύ ευέλικτα και κατάλληλα για μεγάλο πλήθος επιλογών. Είναι ιδανικά για σύνθετες εφαρμογές και παρέχουν τη δυνατότητα σύνδεσης και επικοινωνίας πολλών PLC μαζί. Υπάρχει ποικιλία στο είδος των εξόδων όπως τύπου ρελέ, αναλογικές ή τρανζιστορ. Τέλος σε περίπτωση βλάβης, αντικαθίσταται μόνο το module που έχει τη βλάβη, ενώ όλο το υπόλοιπο σύστημα λειτουργεί κανονικά.



Εικόνα 1.8: Modular PLC

1.5 Πλαίσιο τοποθέτησης μονάδων

Απαραίτητο για τη λειτουργία ενός PLC είναι το πλαίσιο στήριξης αυτού και των επεκτάσεων του (Rack) καθώς και η συσκευή προγραμματισμού του PLC.

Πλαίσιο τοποθέτησης μονάδων



Εικόνα 1.9: Κεντρικό πλαίσιο στήριξης

Οι μονάδες του PLC τοποθετούνται σε ένα κεντρικό πλαίσιο στήριξης. Σε εφαρμογές απαιτητικές που χρησιμοποιούνται μεγαλύτερα PLC, το πλαίσιο στήριξης περιέχει ενσωματωμένο σύστημα αγωγών μέσω του οποίου επικοινωνούν οι διάφορες μονάδες με την CPU. Σε περίπτωση που οι θέσεις του κεντρικού πλαισίου δεν επαρκούν για την τοποθέτηση των υπολοίπων modules, τότε χρησιμοποιείται πλαίσιο επέκτασης. Αυτό το πλαίσιο δέχεται μόνο μονάδες εισόδων/εξόδων και συνδέεται με το κεντρικό πλαίσιο μέσω ειδικής μονάδας διασύνδεσης καλωδίου. Τα πλαίσια επέκτασης μπορεί να είναι παραπάνω από ένα σε πλήθος, αναλόγως τις απαιτήσεις της εφαρμογής και μπορεί να βρίσκονται σε απόσταση από το κεντρικό μέχρι και 600 μέτρα.

1.6 Συσσκευές προγραμματισμού ενός PLC

Για τη λειτουργία ενός PLC δεν αρκεί μόνο το hardware. Απαραίτητο είναι και το software μέσω του οποίου δημιουργείται το πρόγραμμα που εκτελεί τον αυτοματισμό. Για τη δημιουργία του προγράμματος χρησιμοποιείται κάποια συσκευή προγραμματισμού. Η συσκευή προγραμματισμού είναι τελείως ξεχωριστή συσκευή από τον λογικό ελεγκτή. Χρησιμοποιείται για την εισαγωγή του προγράμματος στο PLC και για την παρακολούθηση του αυτοματισμού μέσα από την οθόνη που διαθέτει. Με μία μόνο συσκευή προγραμματισμού μπορούμε να προγραμματίσουμε όλα τα PLC της ίδιας εταιρείας.

Υπάρχουν διάφορες μορφές συσκευών προγραμματισμού.

1.6.1 Προγραμματιστής χειρός

Οι mini προγραμματιστές ή αλλιώς προγραμματιστές χειρός, περιλαμβάνουν αριθμητικά πλήκτρα, κάποια πλήκτρα ειδικών λειτουργιών, πλήκτρα εντολών και κάποια οθόνη συνήθως LED ή LCD. Είναι μικροί στο μέγεθος, φορητοί, οικονομικοί και βολικοί. Βέβαια χρησιμοποιούνται για τον προγραμματισμό μικρών PLC και για εφαρμογές με όχι πάρα πολλές απαιτήσεις.



Εικόνα 1.10 : Προγραμματιστής χειρός

Ο προγραμματιστής χειρός συνδέεται με το PLC μέσω ειδικής θύρας, μέσω της οποίας κατεβαίνει (downloading) το πρόγραμμα στο PLC. Οι προγραμματιστές χειρός των διαφόρων εταιρειών δεν μοιάζουν στον τρόπο προγραμματισμού τους, κάτι που δυσκολεύει την χρήση τους και την εκμάθησή τους στους χρήστες. Παρ' όλα αυτά οι προγραμματιστές χειρός τελευταία έχουν εξελιχθεί αρκετά και δίνουν δυνατότητες όπως σύνδεση με εκτυπωτή ή PC, ανίχνευση βλαβών και έλεγχο προγράμματος.

1.6.2 PG (Programiergeräte – Συσκευή Προγραμματισμού)

Τα PG είναι συσκευές προγραμματισμού κατάλληλες για χρήση σε βιομηχανικό περιβάλλον. Είναι ισχυρές και εύκολες στη χρήση, εξειδικευμένες για το προγραμματισμό ενός PLC, αλλά χρησιμοποιούνται κυρίως για τη συντήρηση και το service. Συνήθως είναι φορητές συσκευές, σε διαστάσεις notebook, με πολύ ισχυρό hardware, δυνατότητες επέκτασης και εξοπλισμένες με όλες τις απαραίτητες θύρες προγραμματισμού.



Εικόνα 1.11: PG

1.6.3 Προσωπικός Υπολογιστής (PC)

Άλλος τρόπος προγραμματισμού ενός PLC είναι συνδέοντας το με κάποιο προσωπικό υπολογιστή (PC), στο οποίο έχει εγκατασταθεί το κατάλληλο software της αντίστοιχης εταιρείας. Η εγκατάσταση του γίνεται παρόμοια με ένα άλλο απλό πρόγραμμα. Σε αυτό προγραμματίζεται ο κώδικας που θα γίνει download στο PLC για την υλοποίηση του αυτοματισμού.

Αυτός ο τρόπος είναι και ο επικρατέστερος, αφού οι περισσότεροι χρήστες είναι εξοικειωμένοι με τη χρήση ενός PC, καθώς παρέχονται και όλα τα πλεονεκτήματα αυτού, όπως αρχειοθετημένα προγράμματα ή εκτυπώσεις.

1.6.4 Φωτεινή πένα

Η φωτεινή πένα είναι συσκευή προγραμματισμού βασισμένη στις γραφικές γλώσσες. Περιλαμβάνει μία οθόνη, πάνω στην οποία σχεδιάζεται το πρόγραμμα με την πένα.

1.7 Κύριες λειτουργίες των PLC

Κάποιες από τις πιο βασικές λειτουργίες ενός PLC είναι οι εξής:

- Δυνατότητα πραγματικού ρολογιού. Αυτό δίνει τη δυνατότητα προγραμματισμού με βάση τις πραγματικές ανάγκες και την εξαγωγή αποτελεσμάτων σε πραγματικό χρόνο, ώρα και ημερομηνία
- Επεξεργασία αναλογικών εισόδων-εξόδων: εκτός από ψηφιακά σήματα, τα PLC έχουν δυνατότητα επεξεργασίας και αναλογικών σημάτων, όπως πίεση, θερμοκρασία, στροφές κινητήρων. Αυτή η δυνατότητα μας επιτρέπει να καλύπτουμε όλες τις ανάγκες αυτομάτου ελέγχου μέσω ενός PLC. Βέβαια το PLC μετατρέπει τα αναλογικά σήματα σε ψηφιακά και μετά πραγματοποιεί την επεξεργασία τους.
- Δικτύωση PLC: σήμερα υπάρχει η δυνατότητα διασύνδεσης πολλών PLC μαζί, με αποτέλεσμα να ανταλλάσσουν δεδομένα μεταξύ τους και να αντιμετωπίζεται όλη η παραγωγική διαδικασία σαν ένα ενιαίο σύστημα.
- Σύνδεση με οθόνες αφής, με αποτέλεσμα όλο το σύστημα να γίνεται πιο φιλικό προς το χρήστη. Ο χρήστης εισάγει δεδομένα στο PLC μέσω της touch screen αλλά και το PLC εξάγει αποτελέσματα σε αυτόν.

Η επιλογή ενός PLC γίνεται βάση ποιών από τις παραπάνω λειτουργίες επιθυμούμε να υλοποιεί. Πιο συγκεκριμένα ένα PLC επιλέγεται με βάση τις ανάγκες στα σήματα εισόδων-εξόδων, με βάση τη μνήμη εργασίας για πρόγραμμα και δεδομένα, με βάση την ταχύτητα της CPU(χρόνο εκτέλεσης για δυαδικές εντολές), με το αν θέλουμε επεκτάσεις ή θύρες επικοινωνίας σε αυτό και με διάφορες άλλες πιο εξειδικευμένες απαιτήσεις.

1.8 Πλεονεκτήματα και Μειονεκτήματα χρήσης ενός PLC

Γενικά η χρήση των PLC στο χώρο της βιομηχανίας και των αυτοματισμών προσφέρει πολλά πλεονεκτήματα. Μερικά από αυτά είναι:

- Το κόστος κατασκευής ενός PLC είναι μικρότερο σε σχέση με το κόστος κατασκευής μεγάλου αριθμού ηλεκτρονόμων, χρονικών και απαριθμητών.
- Ευκολία χρήσης, μεγάλη αξιοπιστία και ταχύτητα.
- Εξοικονόμηση χώρου σε σχέση με ένα κλασικό πίνακα αυτοματισμού.
- Πολύ μικρότερος χρόνος προγραμματισμού σε σχέση με κατασκευή και καλωδίωση ενός κλασικού πίνακα αυτοματισμού.

- Δυνατότητα επέκτασης ή αναβάθμισης.
- Πολύ μικρότερο κόστος συντήρησης σε σχέση με ένα πίνακα αυτοματισμού. Αυτό έχει να κάνει και ως προς τη συχνότητα των βλαβών αλλά και ως προς την ευκολία εντοπισμού της βλάβης.
- Πολύ μεγαλύτερη ευελιξία σε αλλαγές και σε επέκταση του αυτοματισμού, αφού το μόνο που χρειάζεται να συμβεί είναι αλλαγές στο κώδικα του προγράμματος, κάτι που είναι πολύ δύσκολο στον κλασικό αυτοματισμό.
- Δυνατότητα χειρισμού και παρακολούθησης μέσω ηλεκτρονικού υπολογιστή και ανταλλαγή δεδομένων με άλλα προγράμματα όπως ERP.

Παρά όλα αυτά δεν συνίσταται πάντα η χρήση ενός PLC. Μερικά μειονεκτήματα των PLC είναι ότι:

- Δεν συμφέρει η χρήση τους σε πολύ απλούς αυτοματισμούς λόγω κόστους καθώς και σε εφαρμογές με πολλούς ηλεκτρονόμους ισχύος.
- Φαίνονται πολύ περίπλοκα, κυρίως στην Ελλάδα, λόγω έλλειψης γνώσεων και ενημέρωσης για τη χρήση και λειτουργία τους.

Κεφάλαιο 2: Αρχή λειτουργίας και τρόπος προγραμματισμού ενός PLC

Μόλις ο χρήστης δημιουργήσει το πρόγραμμα σε μία από τις συσκευές προγραμματισμού που αναφέρθηκαν στο Κ 1.6, το κατεβάζει στο PLC και όταν αυτό βρεθεί σε κατάσταση λειτουργίας run, πραγματοποιείται η εκτέλεση του προγράμματος.

2.1 Κύκλος λειτουργίας του PLC

Αρχικά το PLC διαβάζει τις εισόδους και τις αποθηκεύει σε μια ειδική περιοχή μνήμης που ονομάζεται εικόνα εισόδων. Στη συνέχεια το πρόγραμμα, συνδιάζει τις εντολές του προγράμματος και τις τιμές των εισόδων και παράγει τις τιμές των εξόδων, οι οποίες αποθηκεύονται σε μια ειδική περιοχή της μνήμης που ονομάζεται αντίστοιχα εικόνα εξόδων. Τέλος το PLC αποδίδει τα αποτελέσματα στις εξόδους.

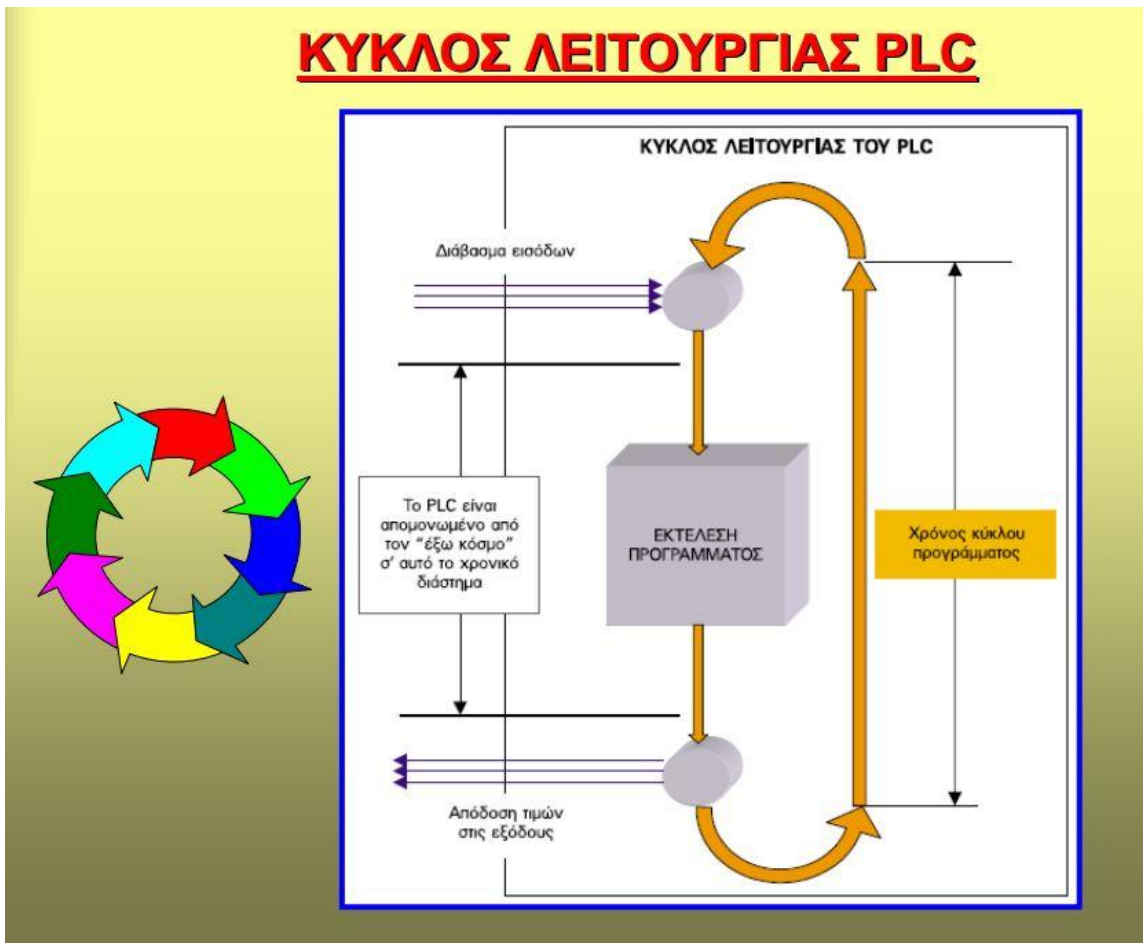
Βέβαια καθώς το PLC «τρέχει» το πρόγραμμα με βάση τις εισόδους που έχει διαβάσει, εκείνο το χρονικό διάστημα δεν έχει καμία επαφή με τις αλλαγές που γίνονται στις εισόδους ή στις εξόδους, δηλαδή είναι απομονωμένο από τον έξω κόσμο. Απλά ο χρόνος εκτέλεσης του προγράμματος είναι τόσο μικρός, που στις περισσότερες εφαρμογές δεν προλαβαίνουν να αλλάξουν οι είσοδοι και οι έξοδοι ή να επηρεαστεί η λειτουργία του προγράμματος από αυτές τις αλλαγές.

Με αυτό το τρόπο ολοκληρώνεται ένας κύκλος λειτουργίας και η διαδικασία επαναλαμβάνεται. Ο χρόνος που απαιτείται για να εκτελεστεί ένας πλήρης κύκλος λειτουργίας του PLC λέγεται χρόνος κύκλου. Αυτός ο χρόνος εξαρτάται από δύο πράγματα:

- πόσο γρήγορη είναι η CPU
- από το μέγεθος του προγράμματος.

Βέβαια και στο πιο αργό PLC ο χρόνος κύκλου του δεν ξεπερνά τα μερικά εκατοντάδες milliseconds (ένας σχετικά αργός χρόνος κύκλου είναι 300ms).

ΚΥΚΛΟΣ ΛΕΙΤΟΥΡΓΙΑΣ PLC



Εικόνα 2.1 : Κύκλος λειτουργίας

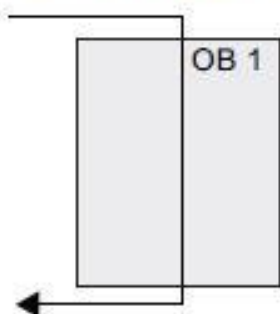
2.2 Τρόποι προγραμματισμού ενός PLC

Υπάρχουν γενικά δύο τρόποι προγραμματισμού ενός PLC.

2.2.1 Γραμμικός προγραμματισμός (linear program)

Στο γραμμικό προγραμματισμό, όλο το πρόγραμμα βρίσκεται σε ένα block (συγκεκριμένα το OB1, Operation Block 1) και το PLC τρέχει αυτό το block συνέχεια κυκλικά. Αυτό σημαίνει ότι μόλις διαβάσει και την τελευταία εντολή του προγράμματος ξαναπηγαίνει στην πρώτη. Ο τρόπος αυτός είναι απλός και πολύ γρήγορος, αλλά προτείνεται για μικρά και απλά προγράμματα.

Linear structure:

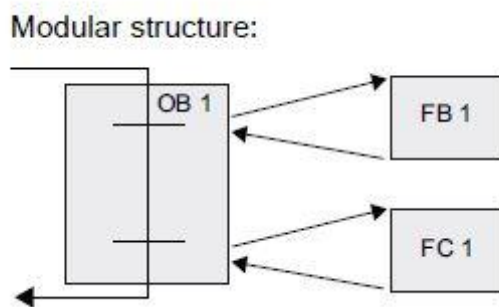


Εικόνα 2.2: Γραμμικός προγραμματισμός

2.2.2 Δομημένος Προγραμματισμός

Στο δομημένο προγραμματισμό, ο χρήστης “σπάει” το πρόγραμμα σε πολλά υποπρογράμματα, χωρίζοντας το έτσι σε πολλά μικρότερα κομμάτια (blocks). Στο OB1 προγραμματίζει πότε και κάτω υπό ποιές συνθήκες θα γίνει η κλήση και η εκτέλεση του κάθε υποπρογράμματος. Το πρόγραμμα απλοποιείται πολύ σε σχέση με το γραμμικό προγραμματισμό, που είναι ολόκληρο σε ένα block. Δηλαδή ο δομημένος προγραμματισμός είναι κατάλληλος για πιο δύσκολες και περίπλοκες εφαρμογές, αφού είναι πιο εύκολες οι αλλαγές σε κάποιο οργανωμένο κομμάτι κώδικα και πιο εύκολη η εύρεση σφαλμάτων σε μικρότερα κομμάτια κώδικα παρά σε ένα ενιαίο και πολύ μεγάλο.

Επίπλέον επειδή το κάθε κομμάτι-block είναι αυτόνομο, μπορεί να χρησιμοποιηθεί σε πολλές διαφορετικές εφαρμογές.



Εικόνα 2.3: Δομημένος προγραμματισμός

2.3 Βασικές αρχές σχεδίασης προγράμματος

Στη CPU εκτελούνται συνεχώς δύο διαφορετικά προγράμματα:

- Το λειτουργικό σύστημα (operating system)
- Το πρόγραμμα εφαρμογής (application program)

2.3.1 Λειτουργικό σύστημα

Το λειτουργικό σύστημα οργανώνει όλες τις λειτουργίες και τις ακολουθίες στη CPU, οι οποίες δεν συσχετίζονται με μία συγκεκριμένη διαδικασία ελέγχου. Εργασίες του λειτουργικού συστήματος περιλαμβάνουν:

- Κλήση του προγράμματος εφαρμογής
- Ενημέρωση του πίνακα εισόδων και εξόδων
- Ανίχνευση και διαχείριση σφαλμάτων
- Διαχείριση της περιοχής μνήμης
- Επικοινωνία με συσκευές προγραμματισμού
- Ανίχνευση διακοπών και κλήση των αντίστοιχων OBs
- Θερμή επανεκκίνηση

2.3.2 Πρόγραμμα εφαρμογής

Το πρόγραμμα εφαρμογής κατασκευάζεται από το χρήστη και αποθηκεύεται στη CPU.



Εικόνα 2.4 : Πρόγραμμα εφαρμογής

Οι εργασίες του προγράμματος εφαρμογής περιλαμβάνουν:

- Επεξεργασία δεδομένων εισόδου και εκχώρηση κατάλληλων τιμών στις εξόδους
- Τρόπο αντίδρασης σε διακοπές ή διαταραχές
- Προσδιορισμό συνθηκών θερμής επανεκκίνησης

2.3.3 Επικοινωνία λειτουργικού συστήματος και προγράμματος εφαρμογής

Το PLC, μέσω των οργανωτικών τμημάτων (Organization Blocks ή OBs) φέρνει σε επικοινωνία το λειτουργικό σύστημα με το πρόγραμμα εφαρμογής. Τα οργανωτικά τμήματα καλούνται από το λειτουργικό σύστημα και ελέγχουν την εκκίνηση του PLC αλλά και την κυκλική εκτέλεση του προγράμματος. Έτσι προγραμματίζοντας ο χρήστης τα οργανωτικά τμήματα προσδιορίζει τη συμπεριφορά της CPU.

2.3.4 Είδη blocks

Υπάρχουν διάφορα blocks, όπως τα OBs, τα FBs, τα SFBs, τα FCs και τα SFCs. Το κάθε ένα χρησιμοποιείται για την κατάλληλη λειτουργία του συνολικού προγράμματος.

2.3.4.1 Organization blocks (OBs)

Τα Organization blocks (OBs) είναι η διασύνδεση του λειτουργικού συστήματος της CPU και του προγράμματος του χρήστη. Τα OBs καθορίζουν με ποιά σειρά θα εκτελεστούν τα διαφορετικά κομμάτια του προγράμματος και δεν μπορούν να κληθούν από άλλα blocks. Καλούνται από το λειτουργικό σύστημα αποκρινόμενα σε ορισμένα γεγονότα, όπως:

- Εκκίνηση της CPU
- Συγκεκριμένη ώρα της ημέρας
- Σε σταθερά χρονικά διαστήματα
- Με την παρέλευση συγκεκριμένου χρόνου
- Με την εμφάνιση σφαλμάτων
- Με την εμφάνιση προβλημάτων στο hardware

Τα OBs καλούνται με βάση την σειρά προτεραιότητάς τους, δηλαδή κάθε OB με υψηλότερη προτεραιότητα μπορεί να διακόψει την εκτέλεση οποιουδήποτε OB με χαμηλότερη προτεραιότητα. Σε περίπτωση που έχουν ίδια προτεραιότητα, τότε δεν διακόπτει το ένα το άλλο, αλλά εκτελούνται με την σειρά που διαγνωστήκανε.

Το OB1 έχει την μικρότερη προτεραιότητα και είναι αυτό που περιέχει το πρόγραμμα του χρήστη καθώς και αυτό που εκτελεί κυκλικά το εκτελέσιμο πρόγραμμα ή το οργανώνει καλώντας άλλα blocks. Επομένως όταν το λειτουργικό σύστημα καλέσει κάποιο άλλο OB, η εκτέλεσή του OB1 διακόπτεται με σκοπό την εκτέλεση του άλλου OB. Ύστερα συνεχίζεται η εκτέλεση του OB1 από το σημείο που έγινε η διακοπή. Παρακάτω φαίνονται η προτεραιότητες των OBs.

OB	Start Event	Default Priority Class	Explanation	
OB1	End of startup or end of OB1	1	Free cycle	
OB10	Time-of-day interrupt 0	2	No default time specified	
OB11	Time-of-day interrupt 1	2		
OB12	Time-of-day interrupt 2	2		
OB13	Time-of-day interrupt 3	2		
OB14	Time-of-day interrupt 4	2		
OB15	Time-of-day interrupt 5	2		
OB16	Time-of-day interrupt 6	2		
OB17	Time-of-day interrupt 7	2		
OB20	Time-delay interrupt 0	3	No default time specified	
OB21	Time-delay interrupt 1	4		
OB22	Time-delay interrupt 2	5		
OB23	Time-delay interrupt 3	6		
OB30	Cyclic interrupt 0 (default interval: 5 s)	7	Cyclic interrupts	
OB31	Cyclic interrupt 1 (default interval: 2 s)	8		
OB32	Cyclic interrupt 2 (default interval: 1 s)	9		
OB33	Cyclic interrupt 3 (default interval: 500 ms)	10		
OB34	Cyclic interrupt 4 (default interval: 200 ms)	11		
OB35	Cyclic interrupt 5 (default interval: 100 ms)	12		
OB36	Cyclic interrupt 6 (default interval: 50 ms)	13		
OB37	Cyclic interrupt 7 (default interval: 20 ms)	14		
OB38	Cyclic interrupt 8 (default interval: 10 ms)	15		
OB40	Hardware interrupt 0	16	Hardware interrupts	
OB41	Hardware interrupt 1	17		
OB42	Hardware interrupt 2	18		
OB43	Hardware interrupt 3	19		
OB44	Hardware interrupt 4	20		
OB45	Hardware interrupt 5	21		
OB46	Hardware interrupt 6	22		
OB47	Hardware interrupt 7	23		
OB55	Status interrupt	2	DPV1 interrupts	
OB56	Update interrupt	2		
OB57	Manufacturer specific interrupt	2		
OB60	SFC35 "MP_ALM" call	25	Multicomputing interrupt Synchronous Cycle Interrupt	
OB 61	Synchronous Cycle Interrupt 1	25		
OB 62	Synchronous Cycle Interrupt 2	25		
OB 63	Synchronous Cycle Interrupt 3	25		
OB 64	Synchronous Cycle Interrupt 4	25		
OB 65	Technology synchronization interrupt	25	Technology synchronization interrupt	
OB70	I/O redundancy error (only in H CPUs)	25	Redundancy error interrupts	
OB72	CPU redundancy error (only in H CPUs)	28		
OB 73	Communication redundancy error OB (only in H CPUs)	25		
OB80	Time error	26, 28 ¹⁾	Asynchronous error interrupts	
OB81	Power supply fault	26, 28 ¹⁾ with S7-300, 25, 28 ¹⁾ with S7-400 and CPU 318		
OB82	Diagnostic interrupt	26, 28 ¹⁾ with S7-300, 25, 28 ¹⁾ with S7-400 and CPU 318		
OB83	Insert/remove module interrupt	26, 28 ¹⁾ with S7-300, 25, 28 ¹⁾ with S7-400 and CPU 318		
OB84	CPU hardware fault	26, 28 ¹⁾ with S7-300, 25, 28 ¹⁾ with S7-400 and CPU 318		
OB85	Program error	26, 28 ¹⁾ with S7-300, 25, 28 ¹⁾ with S7-400 and CPU 318		
OB86	Failure of an expansion rack, DP master system or station for distributed I/Os	26, 28 ¹⁾ with S7-300, 25, 28 ¹⁾ with S7-400 and CPU 318		
OB87	Communication error	26, 28 ¹⁾ with S7-300, 25, 28 ¹⁾ with S7-400 and CPU 318		
OB 88	Processing interrupt	28		
OB90	Warm or cold restart or delete a block being executed in OB90 or load an OB90 on the CPU or terminate OB90	29 ²⁾		Background cycle

Εικόνα 2.5: Προτεραιότητες των OBs

Όπως αναφέρθηκε και προηγουμένως δε συνίσταται σε περίπλοκα προγράμματα ο χρήστης να δημιουργεί όλο τον κώδικα μέσα στο OB1 λόγω απλότητας. Για αυτό το λόγο χρησιμοποιεί υποπρογράμματα τα οποία τα καλεί μέσα από τα FCs και τα FBs.

2.3.4.2 Functions (FCs)

Όταν υπάρχουν συχνά επαναλαμβανόμενες διαδικασίες σε ένα πρόγραμμα, αντί να γράψουμε πολλές φορές τον ίδιο κώδικα, συνίσταται η δημιουργία ενός block που θα καλείται από το OB1 όσες φορές επιθυμείται η εκτέλεση της συγκεκριμένης διαδικασίας. Με αυτό τον τρόπο το block αποθηκεύεται στη μνήμη μία φορά και απλά καλείται όποτε απαιτείται. Στα FCs ορίζονται κάποιες παράμετροι, στις οποίες αποδίδονται πραγματικές τιμές μόνο όταν καλούνται από το πρόγραμμα. Επομένως αποθηκεύονται προσωρινά πληροφορίες καθ' όλη τη διάρκεια εκτέλεσης του block. Μόλις ολοκληρωθεί η εκτέλεση του συγκεκριμένου FB τα δεδομένα αυτά χάνονται. Οι παράμετροι που ορίζουμε μπορεί να είναι :

- είσοδοι που χρησιμοποιούνται μόνο για ανάγνωση από το block που καλέστηκε
- έξοδοι που χρησιμοποιούνται μόνο για ανάθεση αποτελέσματος και επιστρέφονται στο block από το οποίο καλέστηκε
- είσοδοι-έξοδοι που χρησιμοποιούνται και για ανάγνωση και για γράψιμο αποτελέσματος

2.3.4.3 Data Blocks (DBs)

Κατά την εκτέλεση του προγράμματος, εκτός από την λογική του, ο χρήστης επιθυμεί να έχει πληροφορίες και για τις τιμές των σημάτων που επεξεργάζεται. Αυτές μπορεί να τις αποθηκεύσει σε απλές μεταβλητές. Σε περίπτωση όμως που επιθυμεί την αποθήκευση των τιμών αυτών, χρησιμοποιεί τα data blocks(blocks δεδομένων). Όταν τελειώσει η κλήση ενός block δεδομένων, οι τιμές του δεν χάνονται αλλά διατηρούνται, σε αντίθεση με τα προσωρινά δεδομένα. Επομένως τα data blocks χρησιμοποιούνται για την αποθήκευση των δεδομένων του χρήστη.

Τα data blocks καταλαμβάνουν χώρο στην ωφέλιμη μνήμη του προγράμματος και περιλαμβάνουν και αυτά δεδομένα σε μορφή bit, byte, word και double word.

Υπάρχουν δύο κατηγορίες block δεδομένων:

- τα global data blocks των οποίων οι πληροφορίες είναι προσπελάσιμες από όλα τα block λογικής του προγράμματος

- τα instance data blocks τα οποία αναφέρονται σε κάποιο συγκεκριμένο FB(θα αναλυθούν παρακάτω) και μπορούν να χρησιμοποιηθούν μόνο από αυτό

Η CPU διαθέτει δύο καταχωρητές για block δεδομένων, τους DB και DI. Συνεπώς μπορούν να χρησιμοποιηθούν μέχρι δύο block δεδομένων ταυτόχρονα.

2.3.4.4 Function Blocks (FBs)

Τα FBs σε αντίθεση με τα FCs έχουν μνήμη, δηλαδή σε κάθε FB αποδίδεται και ένα block δεδομένων, που ονομάζεται instance data block. Το instance data block χρησιμοποιείται για την αποθήκευση των τοπικών μεταβλητών του συγκεκριμένου FB. Με αποτέλεσμα μόλις τελειώσει η εκτέλεση του συγκεκριμένου FB, οι τιμές των μεταβλητών του να διατηρούνται.

Κάθε instance data block μπορεί να αναφέρεται σε ένα FB. Όμως ένα FB μπορεί να έχει περισσότερα από ένα instance data block.

Εκτός από τα παραπάνω blocks που ορίζονται από το χρήστη, υπάρχουν και τα μπλοκ του συστήματος, που είναι προκαθορισμένες λειτουργίες που προϋπάρχουν στο λειτουργικό σύστημα της CPU, τα οποία όμως, καλούνται από το πρόγραμμα του χρήστη. Αυτά έχουν τον ίδιο κωδικό, το ίδιο interface και τον ίδιο αριθμό σε όλο το σύστημα και για αυτό το λόγο το πρόγραμμα του χρήστη μπορεί να γίνει download σε διάφορα PLC χωρίς αλλαγές.

2.4 Γλώσσα προγραμματισμού ενός PLC

Όπως προαναφέρθηκε εκτός από το hardware, χρειάζεται και το software, για τη δημιουργία του προγράμματος που θα εκτελέσει τον επιθυμητό αυτοματισμό. Το πρόγραμμα γράφεται σε γλώσσες προγραμματισμού (συγκεκριμένα συμβολικές γλώσσες ή διαγράμματα), οι οποίες έχουν καθορισθεί και τυποποιηθεί από το πρότυπο IEC1131-3. Εδώ υπάρχει το μειονέκτημα ότι δεν υπάρχει μόνο μία γλώσσα προγραμματισμού για PLC. Υπάρχουν διαφορές από εταιρεία σε εταιρεία, αλλά ακόμη και στο ίδιο το PLC, αφού στο ίδιο PLC έχεις τη δυνατότητα να προγραμματίσεις με παραπάνω από μία γλώσσα.

Η βάση όλων των γλωσσών είναι η λίστα εντολών (STL ή Statement List). Βέβαια όλες οι γλώσσες βασίζονται στην ίδια λογική και για αυτό το λόγο ο χρήστης μπορεί να χρησιμοποιήσει όποια γλώσσα επιθυμεί, ανάλογα τις προτιμήσεις του. Άλλες γλώσσες είναι οι Ladder, FBD, SCL και Graph. Στην αρχή υπήρχαν κομμάτια κώδικα που μπορούσαν να γραφτούν με STL αλλά όχι με όλες τις υπόλοιπες γλώσσες. Σήμερα έχουν εξελιχθεί αρκετά και οι υπόλοιπες γλώσσες και παρέχουν τη δυνατότητα δημιουργίας πλήρους προγράμματος.

Οι 3 πιο σημαντικές και πιο συχνά χρησιμοποιούμενες γλώσσες είναι οι εξής:

2.4.1 STL (statement list)

Η γλώσσα αυτή αποτελείται από εντολές και μοιάζει με γλώσσα μηχανής Assembly. Για αυτό το λόγο προτιμάται από χρήστες που είναι συνηθισμένοι σε άλλες γλώσσες προγραμματισμού. Ήταν από τις πρώτες γλώσσες που χρησιμοποιήθηκαν στα PLC, αλλά στην αρχή δεν πολύ χρησιμοποιούταν λόγω του ότι οι χρήστες δεν ήταν εξοικωμένοι με τις γλώσσες προγραμματισμού. Έχει εξελιχθεί πάρα πολύ σε δυνατότητες προγραμματισμού και ο προγραμματισμός μπορεί να γίνει τόσο ελεύθερα με την STL που μερικές φορές είναι δύσκολο για κάποιον να παρακολουθήσει τη ροή του προγράμματος.

2.4.2 Ladder

Η Ladder ή γλώσσα ηλεκτρολογικών γραφικών, μοιάζει με το διάγραμμα ενός κυκλώματος. Ο χρήστης χρησιμοποιεί σύμβολα όπως επαφές και πηνία και ήταν η γλώσσα που προωθήθηκε στην αρχή των PLC διότι ήταν πιο φιλική στους μηχανικούς που είχαν συνηθίσει να δουλεύουν πάνω σε ένα ηλεκτρολογικό σχέδιο. Η γλώσσα Ladder δεν χρησιμοποιεί την ευρωπαϊκή προτυποποίηση στο σχεδιασμό των ηλεκτρικών επαφών, αλλά την αμερικάνικη.

2.4.3 FBD (function block diagram)

Η γλώσσα FBD ή γλώσσα λογικών γράφων, είναι γραφική σαν τη Ladder, αλλά χρησιμοποιεί «κουτιά», όπου κάθε κουτί αντιπροσωπεύει κάποια επιμέρους λειτουργία και τελικά το αντίστοιχο λογικό κύκλωμα.

2.5 Εισαγωγή στις HMI (Human Machine Interface – διεπαφή ανθρώπου μηχανής)

HMI ή αναλυτικά Human Machine Interface. Είναι μία συσκευή η οποία επιτρέπει την επικοινωνία του χρήστη-χειριστή με την μηχανή ή το αυτοματοποιημένο σύστημα, στη συγκεκριμένη περίπτωση το PLC. Η HMI είναι συνήθως touch screen οθόνη η οποία περιλαμβάνει πληκτρολόγιο με αριθμητικά ψηφία και LCD οθόνη, όπου υπάρχει δυνατότητα απεικόνισης δεδομένων. Το πληκτρολόγιο χρησιμοποιείται για την εισαγωγή δεδομένων στην εφαρμογή, κυρίως αριθμητικές τιμές, όπως χρόνος κλπ. Η χρήση μίας HMI και ο προγραμματισμός της έτσι ώστε να είναι φιλική προς τον χειριστή, μπορεί να είναι η διαφορά ενός αναποτελεσματικού συστήματος από ένα σχετικά οικονομικό και πολύ αποτελεσματικό. Στην LCD οθόνη μπορούν να απεικονιστούν λειτουργικά μηνύματα, πληροφορίες για διάφορες μεταβλητές του προγράμματος, καθώς και πληροφορίες του συστήματος. Στην συγκεκριμένη περίπτωση ο προγραμματισμός μίας HMI οθόνης γίνεται από το ίδιο software το TIA PORTAL της Siemens, κάτι το οποίο θα αναλυθεί με λεπτομέρεια στη ανάλυση του προγράμματος.

Κεφάλαιο 3: Πρότυπα και πρωτόκολλα επικοινωνίας

3.1 Εισαγωγή

Όπως αναφέρθηκε και προηγουμένως τα PLC παρέχουν τη δυνατότητα επικοινωνίας με άλλες συσκευές. Παρακάτω αναλύονται κάποιοι τρόποι επικοινωνίας. Αρχικά δίνεται η διαφορά μεταξύ σειριακής και παράλληλης επικοινωνίας.

Παράλληλη Επικοινωνία: Στέλνει και λαμβάνει οχτώ bits κάθε στιγμή, χρησιμοποιώντας οχτώ ανεξάρτητα καλώδια. Αυτό του παρέχει τη δυνατότητα επικοινωνίας σε πολύ υψηλές ταχύτητες, αλλά έχει το μειονέκτημα ότι είναι ογκώδη τα καλώδια λόγω των πολλών κλώνων που περιέχει.

Σειριακή Επικοινωνία: Στέλνει ένα bit δεδομένων κάθε φορά πάνω σε μία και μοναδική γραμμή μεταφοράς. Είναι οχτώ φορές πιο αργή από την παράλληλη, αλλά είναι πιο απλή και έχει και δυνατότητα αμφίδρομης επικοινωνίας με τρεις μόνο κλώνους, ένα για αποστολή δεδομένων, ένα για λήψη δεδομένων και ένα για γείωση.

Για την επικοινωνία εκτός από τα καλώδια χρειάζεται και το λογισμικό που καθορίζει τους κανόνες αποστολής δεδομένων. Για να επικοινωνήσουν δύο συσκευές πρέπει να χρησιμοποιούν την ίδια σύμβαση. Συνήθως υπάρχει ένα bit εκκίνησης, που αποστέλλει ο πομπός στο δέκτη. Στη συνέχεια στέλνει το πραγματικό πακέτο δεδομένων, που αποτελείται συνήθως από 5, 6, 7 ή 8 bits. Πιο σύνηθες είναι τα 7 ή τα 8 bits. Στη συνέχεια αποστέλλεται ένα bit τερματισμού για να καταλάβει ο δέκτης πότε τελειώνει η αποστολή δεδομένων. Το bit τερματισμού είναι πάντα 1 και διαφέρει από το 1 που μπορεί να περιέχεται σε κάποιο πακέτο δεδομένων π.χ. μπορεί να διαρκεί περισσότερο η περίοδος του. Επίσης μπορεί να χρησιμοποιείται και κάποιο bit ισοτιμίας για τον έλεγχο του αν το πακέτο δεδομένων εστάλη σωστά. Δύο τρόποι είναι οι πιο συνηθισμένοι, η περιπτή ή η άρτια ισοτιμία. Στην άρτια μετράμε τα bit 0(ή 1) και αν αυτά είναι άρτια τότε το bit ισοτιμίας είναι 0, αλλιώς 1. Το αντίθετο γίνεται με την περιπτή.

3.2 Πρότυπο RS-232

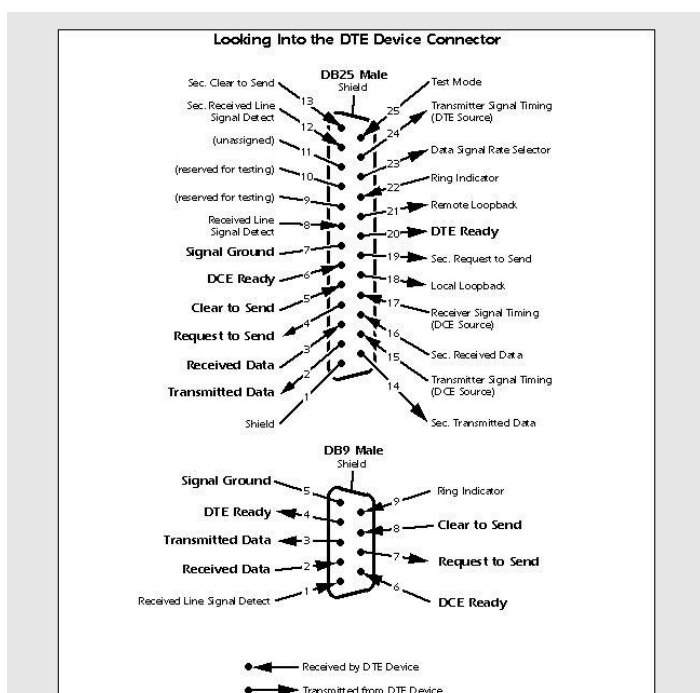
Στις αρχές του 1960 μια ένωση προτύπων, γνωστή ως Electronic Industries Association, ανέπτυξε ένα κοινό πρότυπο για τον εξοπλισμό που σχετιζόταν με την επικοινωνία δεδομένων. Τότε επικοινωνία δεδομένων εννοείτο η ψηφιακή ανταλλαγή δεδομένων μεταξύ ενός κεντρικού υπολογιστή και ενός άλλου απομακρυσμένου ή πιθανώς ανάμεσα σε δύο συσκευές χωρίς να περιλαμβάνεται κάποιος υπολογιστής. Αυτές οι συσκευές συνδέονταν με τηλεφωνική γραμμή και για αυτό το λόγο υπήρχε η ανάγκη ενός modem στη κάθε μία για τη μετάφραση του σήματος. Όμως η μεταβίβαση των δεδομένων μέσω αναλογικών καναλιών είχε σαν αποτέλεσμα μεγάλες απώλειες. Έτσι δημιουργήθηκε η ανάγκη για τη

δημιουργία ενός προτύπου με σκοπό την αξιόπιστη επικοινωνία διαφόρων συσκευών, αλλά και τη δυνατότητα επικοινωνίας συσκευών από διαφορετικές κατασκευαστικές εταιρείες. Το αποτέλεσμα ήταν το πρότυπο RS232 ή Recommended Standard Number 232 (Προτεινόμενο Πρότυπο αριθμός 232) ή αλλιώς EIA232.

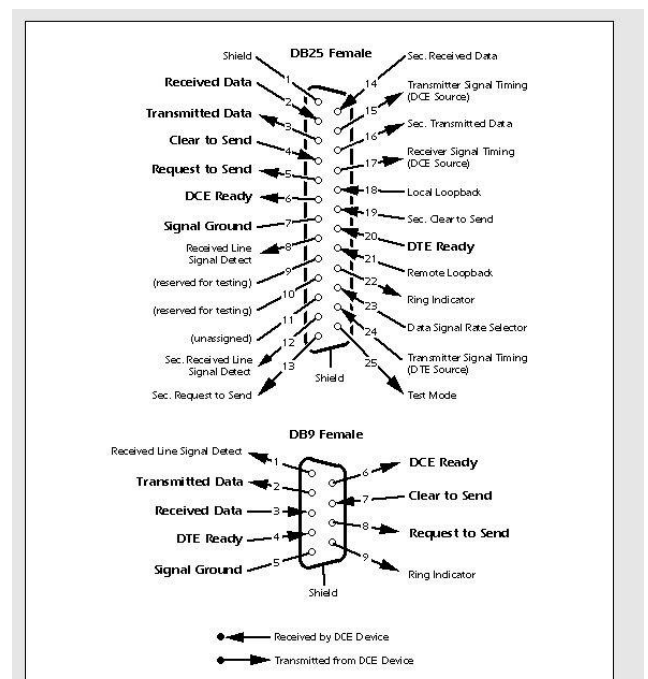
Το RS-232 είναι ένα πρότυπο για σειριακή μετάδοση δυαδικών σημάτων δεδομένων μεταξύ δύο συσκευών. Το πρότυπο αυτό προσδιορίζει τάση του σήματος, χρόνο του σήματος, λειτουργία του σήματος και μηχανικούς αντάπτορες για την σύνδεση του εξοπλισμού.

Σε όλα αυτά τα χρόνια από τότε που αναπτύχθηκε αυτό το πρότυπο, η Electronic Industries Association δημοσίευσε τρεις εκδόσεις του προτύπου με πιο πρόσφατη αυτή του EIA232F το 1997. Με την εξέλιξη της τεχνολογίας στην ηλεκτρονική, οι κατασκευαστές ενέκριναν την απλοποίηση του προτύπου λόγω εφαρμογών που ήταν δύσκολο να οραματιστούν την περίοδο που πρωτοδημιουργήθηκε το πρότυπο. Σήμερα οι περισσότερες σειριακές συνδέσεις είναι EIA232, όσον αφορά τη τάση σήματος, τους αντάπτορες και τα πρωτόκολλα, είτε περιλαμβάνεται modem είτε όχι.

Στο πλήρες πρότυπο EIA232 η συσκευή που αποστέλλει τα δεδομένα ονομάζεται DTE (data terminal equipment) και φέρει 25πino αρσενικό βύσμα, από τα οποία χρησιμοποιεί τα 22pins από τα 25 για σήματα ή γείωση. Η συσκευή που λαμβάνει τα δεδομένα ονομάζεται DCE (data circuit-terminating equipment) και φέρει 25πino θηλυκό βύσμα όπου χρησιμοποιεί επίσης τα 22 από τα 25 pins για σήματα ή γείωση. Βέβαια και τα 22 pins χρησιμοποιούνται στη περίπτωση που στη μεριά του DCE η συσκευή είναι modem. Σε περίπτωση που δεν είναι, χρησιμοποιούνται πολύ λιγότερα pins και για αυτό το λόγο χρησιμοποιείται βύσμα 9πino. Στην εικόνα 3.1 και 3.2 φαίνονται τα pins όταν χρησιμοποιούνται όλα ή όταν χρησιμοποιούνται τα πιο συνηθισμένα.



Εικόνα 3.1: RS-232 για DTE συσκευή



Εικόνα 3.2: RS-232 για DCE συσκευή

Παρ' όλα αυτά έχει επικρατήσει μία άτυπη σύμβαση, να φέρει αρσενικό βύσμα 9πινο η συσκευή που αποστέλλει τα δεδομένα και θηλυκό 25πινο η συσκευή που λαμβάνει τα δεδομένα.

Οι λειτουργίες του προτύπου EIA232 μπορούν να χωριστούν σε 6 κατηγορίες:

- Σήμα γείωσης.
- Βασικό κανάλι επικοινωνίας, που χρησιμοποιείται για ανταλλαγή δεδομένων και περιλαμβάνει και σήματα ελέγχου ροής.
- Δευτερεύον κανάλι επικοινωνίας, που χρησιμοποιείται μόνο όταν υπάρχει modem για τον έλεγχο αυτού, και απαιτείται για να ξαναστείλει τα δεδομένα όταν συμβεί κάποιο πρόβλημα.
- Κατάσταση του modem και έλεγχο σημάτων. Αυτά τα σήματα δείχνουν την κατάσταση του modem και παρέχουν ενδιάμεσα σήματα ελέγχου όταν το τηλεφωνικό κανάλι είναι εγκατεστημένο.
- Χρονικά σήματα πομπού και δέκτη. Εάν χρησιμοποιείται κάποιο σύγχρονο πρωτόκολλο, αυτά τα σήματα παρέχουν χρονική πληροφορία για τον πομπό και το δέκτη έτσι ώστε να μπορούν να λειτουργούν σε διαφορετικά baud rate.
- Σήματα για έλεγχο καναλιού. Πριν την ανταλλαγή δεδομένων, το κανάλι μπορεί να ελεγχθεί για την ακεραιότητά του, και το baud rate αυτόματα προσαρμόζεται στο μέγιστο rate που μπορεί να υποστηρίξει το κανάλι.

Το πρότυπο EIA232 χρησιμοποιεί δυαδική λογική στη οποία αρνητική τάση αντιπροσωπεύει το λογικό «1» και θετική τάση το λογικό «0». Βέβαια αυτό συνέβαινε με το αρχικό πρότυπο. Η αντίθετη λογική εκχωρήθηκε στη συνέχεια και η αντιστροφή αυτού βρίσκεται στα περισσότερα μοντέρνα ηλεκτρικά κυκλώματα και συσκευές.

Στο πρότυπο RS232 ως μέγιστη απόσταση μετάδοσης έχει καθοριστεί τα 17 μέτρα, αλλά παρόλα αυτά αν το καλώδιο έχει καλή μόνωση και ταχύτητα μετάδοσης 19,2 kbaud, τότε μπορεί να αποστείλει δεδομένα μέχρι και 3000 μέτρα.

3.3 Πρότυπο RS-485

Το πρότυπο 485, όπως είναι σήμερα γνωστό, είναι ένα σύστημα μετάδοσης δεδομένων το οποίο προσφέρει λύσεις για μεταφορά δεδομένων σε βιομηχανικούς χώρους με θορύβους και σε μακρινές αποστάσεις. Δεν προσδιορίζει ένα συγκεκριμένο λογικό πρωτόκολλο και χρησιμοποιείται ως φυσικό στρώμα από πολλά πρωτόκολλα όπως το Modbus, το Profibus και άλλα.

Το TIA/EIA-485 πρότυπο, γνωστό και ως RS-485 περιγράφει ένα interface επικοινωνίας το οποίο χρησιμοποιεί μεταφορά δεδομένων μέσω ενός ή δύο ζευγαριών καλωδίων για να δημιουργήσει επικοινωνία ανάμεσα σε 32 μονάδες

φορτίου. Συνήθως κάθε συσκευή (πομπός και δέκτης) αντιστοιχεί σε μία μονάδα φορτίου. Βέβαια νέες συσκευές μπορούν να έχουν κλασματική μονάδα φορτίου(έως και 1/8), με αποτέλεσμα ο αριθμός των συσκευών να είναι πολύ μεγαλύτερος των 32 και να φτάνει μέχρι και τις 256. Δίκτυα RS-485 συνήθως επικοινωνούν με ένα στρεπτό ζεύγος καλωδίων όπου τα δεδομένα μεταφέρονται και προς τις δύο κατευθύνσεις. Κάθε συσκευή είναι ενεργοποιημένη μόνο όταν μεταδίδει δεδομένα και το υπόλοιπο χρονικό διάστημα επιτρέπει στις άλλες να εκπέμπουν δεδομένα. Μόνο μία συσκευή μπορεί να εκπέμπει κάθε στιγμή. Αυτό το δίκτυο ονομάζεται half-duplex. Βέβαια ένα RS-485 δίκτυο μπορεί να λειτουργήσει και με δύο στρεπτά ζεύγη καλωδίων και να λειτουργεί ως full duplex.

Το πρότυπο RS-485 χρησιμοποιεί διαφορετικές γραμμές για επικοινωνία. Ως δεδομένα πληροφορίας χρησιμοποιείται η διαφορά δυναμικού ανάμεσα στις δύο γραμμές του στρεπτού ζεύγους. Δυαδικά δεδομένα αναγνωρίζονται από την πολικότητα της τάσης, αφού τα δεδομένα αντιστοιχούν σε λογικό 1 όταν η πολικότητα είναι θετική και σε λογικό 0 όταν η πολικότητα είναι αρνητική. Η ισορροπημένη μεταφορά δεδομένων ακυρώνει τον θόρυβο σε πολύ μεγάλο βαθμό, αφού ο ίδιος ο θόρυβος εμπεριέχεται και στους δύο αγωγούς του ζευγαριού, διαφυλάσσοντας έτσι την διαφορά δυναμικού που μεταφέρει την πληροφορία. Βέβαια οι συσκευές πρέπει να μοιράζονται κοινή τάση αναφοράς και για αυτό είναι λάθος να λέμε ότι το RS-485 χρησιμοποιεί μόνο δύο καλώδια. Συγκεκριμένα για τη σχεδίαση ενός ευσταθούς δικτύου, ειδικά όταν μιλάμε για μεγάλες αποστάσεις παίζει τεράστιο ρόλο η γείωση.

Το πρότυπο 485 σε συνέχεια του στρεπτού ζεύγους, επισήμως συστήνει το καλώδιο του δικτύου να έχει ηλεκτρομαγνητική θωράκιση με την οποία διαφυλάσσεται η ασφάλεια του δικτύου από κάθε φύσης ηλεκτρικό θόρυβο.

Τέλος όταν υπάρχει ανάγκη ενός μεγάλου δικτύου ή η ανάγκη μεγάλων ταχυτήτων τότε, εμφανίζονται φαινόμενα γραμμών μεταφοράς όπως αντανάκλασεις. Σε αυτή τη περίπτωση παίζει σημαντικό ρόλο και η τοπολογία του δικτύου. Γενικά υπάρχουν αρκετοί τρόποι για τη δομή της τοπολογίας του δικτύου όπως σπονδυλική στήλη με κλαδιά, σπονδυλική στήλη με δέσμες, δακτύλιος και άλλοι. Αυτή που συστήνεται κυρίως για τέτοιες περιπτώσεις είναι αυτή της αλυσωτής σύνδεσης μαργαρίτας ή αλλιώς Daisy chain.

3.3 Πρωτόκολλο Profibus

Το πρωτόκολλο Profibus χρησιμοποιείται στο επίπεδο πεδίου (field level) μιας βιομηχανικής εγκατάστασης, δηλαδή εκεί όπου βρίσκονται καταναμημένα περιφερειακά, όπως μονάδες εισόδου-εξόδου, μορφοτροπείς, βαλβίδες, κινητήρες, ελεγκτές, τερματικά και συστήματα επικοινωνιών και ελέγχου, τα οποία πρέπει να επικοινωνούν με ένα αξιόπιστο και αποτελεσματικό τρόπο σε πραγματικό χρόνο. Το Profibus αναλυτικά προέρχεται από το Process-Field-Bus και επικράτησε σε τέτοιες χρήσεις γιατί έχει κάποια πλεονεκτήματα έναντι του

Ethernet που χρησιμοποιείται σε άλλα πεδία της βιομηχανίας και όχι μόνο. Αρχικά δίνει τη δυνατότητα σε όλους τους κόμβους ενός δικτύου να “ακούγονται” σε τακτά χρονικά διαστήματα καθώς επίσης ορίζει και βαθμούς προτεραιότητας στην μετάδοση των σημάτων. Αυτό σημαίνει ότι η μετάδοση των δεδομένων είναι κυκλική, δίνοντας όμως τη δυνατότητα σε δεδομένα υψηλής προτεραιότητας να κινηθούν μη κυκλικά. Σε ένα δίκτυο Profibus η ταχύτητα κυμαίνεται από τα 9,6kbps μέχρι 12.000kbps. Βέβαια αυτό εξαρτάται και από το μέσο ή την τεχνολογία που χρησιμοποιείται. Παρ’ όλα αυτά παρέχει πολύ καλή ηλεκτρομαγνητική θωράκιση λόγω της χρήσης σε βιομηχανικούς χώρους. Ενδιαφέρον χαρακτηριστικό του Profibus είναι ότι το πρωτόκολλο είναι το ίδιο ανεξαρτήτως του τι μέσο μεταφοράς ή τεχνολογίας χρησιμοποιείται.

Υπάρχουν τρία είδη του Profibus:

- Profibus DP (Decentralized Peripherals): Είναι κατάλληλο για επικοινωνία μεταξύ συστημάτων αυτοματισμού. Χρησιμοποιείται για τη μεταφορά της πληροφορίας όχι ως αναλογικό σήμα αλλά ως δεδομένα και χρησιμοποιεί RS485 ως φυσικό επίπεδο μετάδοσης. Παρέχει υψηλές ταχύτητες με σχετικά μικρό κόστος. Το baud rate μετάδοσης δεν είναι σταθερό. Τέλος όπως θα δούμε και παρακάτω υπάρχουν τρεις εκδόσεις του, η DP-V0 (1993), η DP-V1 (1997) και η DP-V2(2002).
- Profibus PA (Process Automation): Είναι κατάλληλο για χρήσεις στην καρδιά της αυτοματοποιημένης διαδικασίας, δηλαδή στο επίπεδο πεδίου. Έχει ιδιαίτερα χαρακτηριστικά που καθορίζονται από ειδικά πρότυπα (IEC 1158-2), δεν έχει κύριους σταθμούς master και είναι υποχρεωτικά κλώνος ενός μεγαλύτερου δικτύου Profibus DP. Χρησιμοποιείται σε πετροχημικές και χημικές βιομηχανίες, σε φινιριστήρια και σταθμούς ηλεκτρικής ενέργειας. Αναπτύχθηκε για να αντικαταστήσει βιομηχανικές διαδικασίες με μεταφορά 4-20mA. Χρησιμοποιεί για τη μεταφορά MBP μετατροπείς DPJRA για μετατροπή από RS485 σε MBP και το baud rate του είναι σταθερό στα 31,25 kbits/sec.
- Profibus FMS (Field Message Specification): Χρησιμοποιείται για απαιτητικούς επικοινωνιακούς σκοπούς και προσφέρει πολλές προηγμένες δυνατότητες. Όμως τείνει να εξαλειφθεί λόγω της ανάπτυξης του DP.

Στο πρότυπο Profibus χρησιμοποιούνται τέσσερις τρόποι μετάδοσης της πληροφορίας σε φυσικό επίπεδο :

- Δίκτυα RS485
- Δίκτυα με οπτικές ίνες
- Δίκτυα με υπέρυθρη ακτινοβολία
- Πρότυπο IEC 1158-2 δηλαδή MBP

Βέβαια, είτε το Profibus DP που χρησιμοποιεί καλώδιο RS-485 ή οπτικές ίνες, είτε το Profibus PA που χρησιμοποιεί MBP-IS, χρησιμοποιούν το ίδιο

πρωτόκολλο και οι συσκευές μπορούν να ανταλλάξουν κυκλικά δεδομένα εισόδου–εξόδου με έναν DP master.

Στη συνέχεια αναλύεται το Profibus DP. Γενικά ως πρωτόκολλο δεν είναι στατικό. Με την πάροδο του χρόνου έχει επεκταθεί σε διάφορες εταιρείες με διάφορες εφαρμογές. Γενικά υπάρχουν 3 βασικές εκδόσεις. Η πρώτη DPV0 είναι η βάση του Profibus και είναι το το αυθεντικό πρωτόκολλο του Profibus για την υποστήριξη γρήγορης ανταλλαγής δεδομένων εισόδου–εξόδου. Η επόμενη έκδοση, η PROFIBUS DPV1, πρόσθεσε επεκτάσεις που επέτρεπαν διάβασμα/γράψιμο κάποιων επιπλέον παραμέτρων για πιο ευφυείς συσκευές όπως έξυπνους κινητήρες και PROFIBUS PA για όργανα που συμμετέχουν στο επίπεδο πεδίου, όπως βαλβίδες θέσης, μεταδότες πίεσης κλπ. Τέλος η DPV2 προσέθεσε επεκτάσεις ώστε εφαρμογές ελέγχου κίνησης να μπορούν να επιτευχθούν απευθείας με χρήση Profibus και όχι χρησιμοποιώντας επιπλέον ένα δευτερεύοντα δίαυλο επικοινωνίας για αυτό το λόγο.

Το Profibus DP αποτελείται από δύο είδη συσκευών συνδεδεμένες πάνω στον ίδιο δίαυλο επικοινωνίας, τις master και τις slave. Είναι αμφίδρομο δίκτυο, δηλαδή μια συσκευή, ο master, στέλνει ένα αίτημα σε μία slave, και ο slave ανταποκρίνεται σε αυτό το αίτημα. Έτσι η συνθήκη του διαύλου επικοινωνίας δεν είναι πρόβλημα, αφού μόνο ένας master μπορεί να ελέγξει το bus κάθε φορά και η συσκευή slave πρέπει να ανταποκριθεί αμέσως στο αίτημα του master. Επειδή ένα αίτημα του master φαίνεται σε όλες τις slave συσκευές που είναι συνδεδεμένες πάνω στο δίαυλο, πρέπει να υπάρχει ένας μηχανισμός που θα αναγνωρίζει κάθε slave ότι το μήνυμα ήταν για αυτόν. Για αυτό το λόγο κάθε συσκευή σε ένα δίκτυο Profibus πρέπει να έχει μια συγκεκριμένη διεύθυνση, address.

Το πρωτόκολλο Profibus υποστηρίζει διευθύνσεις από το 0 έως το 127. Όμως επειδή οι διευθύνσεις 0, 126, 127 είναι για ειδικές περιπτώσεις, οι διευθύνσεις που πρέπει να χρησιμοποιούνται στη πράξη για λειτουργικές συσκευές όπως PLC, drivers, encoders κλπ είναι από 1 έως 125. Ο χρήστης συγκεκριμενοποιεί ποιους slaves πρέπει να βρει ο master πάνω στο bus καθώς και τι πληροφορίες θα μεταφερθούν από το master σε κάθε slave ξεχωριστά.

Όλες οι πληροφορίες που πρέπει να γνωρίζει ο master για να ξεκινήσει η λειτουργία πάνω στο bus προέρχονται από ένα αρχείο βάσης δεδομένων που παράγεται από ένα εργαλείο (tool) configuration του Profibus.

Κάθε κατασκευαστής master συσκευών σε δίκτυα Profibus παρέχει ένα εργαλείο για τη δημιουργία του αρχείου της βάσης δεδομένων. Γενικά αν μάθει κάποιος χρήστης να χρησιμοποιεί κάποιο εργαλείο ενός συγκεκριμένου κατασκευαστή, τότε εύκολα μπορεί να χρησιμοποιήσει και οποιοδήποτε άλλο, αφού όλα τα εργαλεία για configuration Profibus μοιράζονται παρόμοιες λειτουργίες.

Ένα τέτοιο εργαλείο για κυκλική λειτουργία I/O δεδομένων πρέπει να έχει τη δυνατότητα να πραγματοποιεί τα ακόλουθα :

- Να δέχεται GSD αρχεία (δηλαδή αρχεία που να περιγράφουν την αντίστοιχη λειτουργία) και να διατηρεί ένα κατάλογο με τις συσκευές που βρίσκονται πάνω στον αντίστοιχο δίαυλο επικοινωνίας.
- Να επιτρέπει τον προσδιορισμό των διευθύνσεων των συσκευών που συμμετέχουν στο Profibus δίκτυο.
- Να επιτρέπουν την συγκεκριμενοποίηση των δεδομένων εισόδου και εξόδου ανάμεσα σε master και slaves συσκευές.
- Να επιτρέπουν την επιλογή του Baud rate του συστήματος.
- Να παράγουν το αρχείο βάσης δεδομένων ώστε να μπορεί να χρησιμοποιηθεί από το master.
- Να επιτρέπουν την επιλογή κάποιων βασικών παραμέτρων ώστε να ενεργοποιούν βασικά λειτουργικά χαρακτηριστικά της κάθε συσκευής.

Παράλληλα ένας κατασκευαστής μίας συσκευής slave αναπτύσσει ένα αρχείο GSD που περιγράφει κατά κάποιο τρόπο τη συσκευή. Αυτό το αρχείο περιγράφει πλήρως τις λειτουργίες της συσκευής σε ότι αφορά το Profibus π.χ. τι baud rate υποστηρίζει , πιθανά δεδομένα εισόδου / εξόδου ή κάποιες αρχικές παραμέτρους επιλογής.

Αυτά τα αρχεία μπορεί να τα κατεβάσει κάποιος μέσω Internet από το site www.profibus.com ή από το site του ίδιου του προμηθευτή που παρέχει τη συγκεκριμένη slave συσκευή.

Μόλις γίνει install το GSD αρχείο στο εργαλείο του Profibus , αυτή η συσκευή θα εμφανίζεται στο κατάλογο του συγκεκριμένου εργαλείου και θα μπορεί να χρησιμοποιηθεί σε οποιαδήποτε λειτουργία απαιτηθεί. Η διαδικασία με την οποία γίνονται install αυτά τα αρχεία διαφέρει από εργαλείο σε εργαλείο διαφορετικών προμηθευτών, παρ'όλα αυτά σε κάθε περίπτωση είναι πολύ απλή.

Όταν όλα τα επιθυμητά GSD αρχεία έχουν γίνει install, τότε μπορεί να προσδιοριστεί ένα πλήρες σύστημα διαύλου επικοινωνίας. Αρχικά επιλέγεται η κατάλληλη master συσκευή και της ανατίθεται μία Profibus διεύθυνση. Είναι λογικό στον κατάλογο επιλογής master συσκευών να υπάρχουν μόνο συσκευές του συγκεκριμένου κατασκευαστή. Στην συνέχεια επιλέγεται μία slave συσκευή, στην οποία ανατίθεται επίσης μία άλλη διεύθυνση. Προσδιορίζονται τα δεδομένα εισόδου-εξόδου (I/O) που θα ανταλλάσσονται και επιλέγονται οι κατάλληλες παράμετροι για την σωστή λειτουργία της συσκευής. Στη συνέχεια σώζουμε αυτό το bus configuration και παράγεται η αντίστοιχη βάση δεδομένων. Μετά είμαστε σε θέση να φορτώσουμε αυτή τη βάση δεδομένων στη master συσκευή. Ο μηχανισμός φόρτωσης του αρχείου της βάσης δεδομένων μπορεί να ποικίλει από κατασκευαστή σε κατασκευαστή. Το πιο σύνηθες είναι να γίνεται μέσω κάποιας

σειριακής θύρας. Μερικές φορές φορτώνεται το αρχείο σε flash memory και μέσω αυτής εισέρχεται στη master συσκευή.

Στη συνέχεια η master συσκευή πρέπει να προσδιορίσει άμα οι συσκευές στις οποίες έχει ανατεθεί κάποια διεύθυνση εμπεριέχονται πραγματικά στο δίκτυο και τους αρχικοποιεί για λειτουργική χρήση ή για ανταλλαγή δεδομένων. Για να αναθέσει η master συσκευή σε τι κατάσταση βρίσκονται οι slave συσκευές, το κάνει μέσω κάποιας καλά προσδιορισμένης αλληλουχίας αλληλεπιδράσεων με κάθε συσκευή που βρίσκεται πάνω στο δίαυλο.

Σε περίπτωση που ο master καταλάβει κάποια έλλειψη ενέργειας όταν κάνει back up, χρησιμοποιεί τη βάση δεδομένων που βρίσκεται στη μνήμη ώστε μέσω τις διαδικασίες αρχικοποίησης να τις επαναφέρει σε κατάσταση λειτουργίας. Άμα κάποια συσκευή δεν μπορεί να επανέλθει τότε πρέπει να αντικατασταθεί. Μόλις αντικατασταθεί, ο master καταλαβαίνει ότι κάποια άλλη μπήκε στη θέση της. Βέβαια πρέπει να είναι ίδιου τύπου και με ίδια διεύθυνση. Όταν συμβεί το καταλαβαίνει και τη φέρνει σε θέση λειτουργίας.

Μόλις το δίκτυο λειτουργήσει, αυτό που συμβαίνει είναι η ανταλλαγή I/O δεδομένων μεταξύ του master και των slaves. Ο master, π.χ. ένα PLC με Profibus interface, στέλνει δεδομένα εξόδου σε μία συσκευή slave. Ο slave με την κατάλληλη διεύθυνση αμέσως ανταποκρίνεται με τα δεδομένα εισόδου του. Εδώ φαίνεται ότι δεδομένα εξόδου είναι αυτά που στέλνονται από τον master στους slaves, ενώ εισόδου αυτά που φεύγουν από τους slaves και εισέρχονται στο master. Αυτή η κυκλική ανταλλαγή δεδομένων συμβαίνει ασύγχρονα και επαναλαμβάνεται όσο πιο γρήγορα γίνεται. Το πιο συνηθισμένο baud rate είναι 1500 kbits/sec και ο κύκλος ανταλλαγής δεδομένων συνήθως επαναλαμβάνεται πολλές φορές κατά τη διάρκεια ενός κύκλου λογικού ελέγχου.

Το 85% των δικτύων Profibus υλοποιούνται με ένα master. Ωστόσο συστήματα και με περισσότερους master λειτουργούν εξίσου καλά. Σε τέτοιες περιπτώσεις κάθε συσκευή master ελέγχει το δίαυλο επικοινωνίας για κάποιο μικρό χρονικό διάστημα και κατά τη διάρκεια αυτού του χρόνου ανταλλάσει δεδομένα με κάθε συσκευή slave που του έχει ανατεθεί. Στην συνέχεια δίνει τον έλεγχο στην επόμενη master συσκευή, μέσω ενός μηνύματος, που αποκαλείται σκυτάλη και τότε αυτός ο master ανταλλάσει I/O δεδομένα με τους δικούς του slaves. Μόνο ο master που έχει τη σκυτάλη επιτρέπεται να ξεκινήσει επικοινωνία με τους slaves του. Όταν και ο τελευταίος master τελειώσει με την ανταλλαγή δεδομένων, τότε δίνει τον έλεγχο και πάλι στον πρώτο και η διαδικασία ξεκινάει από την αρχή.

Το πρωτόκολλο Profibus παρέχει αρκετά ακριβές διαγνωστικές δυνατότητες που μπορούν να τις σχεδιάσουν οι κατασκευαστές στα προϊόντα τους. Μέσω του Profibus προσφέρεται η δυνατότητα διάγνωσης ενός λειτουργικού προβλήματος, π.χ. μιας υπέρτασης σε μία είσοδο ή ένα σπασμένο καλώδιο σε μία έξοδο.

Κατά τη διάρκεια της ανταλλαγής δεδομένων, μία συσκευή slave μπορεί να δείξει στη master συσκευή ότι έχει ανιχνευθεί μία διαγνωστική κατάσταση. Στον επόμενο κύκλο ανταλλαγής δεδομένων ο master παραλαμβάνει την πληροφορία διάγνωσης από το slave. Μία συσκευή μπορεί να εκδώσει διαγνωστικές πληροφορίες σε 4 διαφορετικούς τύπους διαγνωστικών:

- Standard διαγνωστικά.
- Διαγνωστικά που σχετίζονται με τη συσκευή.
- Διαγνωστικά που σχετίζονται με modules.
- Διαγνωστικά που σχετίζονται με το κανάλι επικοινωνίας.

Κάθε master πρέπει να σώζει οτιδήποτε διαγνωστικά δεδομένα από ένα slave έτσι ώστε το πρόγραμμα του χρήστη να έχει πρόσβαση σε αυτό. Κάθε master το κάνει με λίγο διαφορετικό τρόπο και για αυτό το λόγο πρέπει ο χρήστης να εξοικειωθεί με τον συγκεκριμένο master.

Τα standard διαγνωστικά μεγέθους 6 bytes από κάθε slave συσκευή, απαιτούνται για πληροφορίες που κυρίως έχουν να κάνουν με προβλήματα εκκίνησης. Για παράδειγμα αν τα I/O που έχουν δηλωθεί στο εργαλείο, δεν ταιριάζουν με αυτά που περιμένει ο slave θα δείξει configuration fault. Ενώ αν μια συσκευή slave υπάρχει στο configuration file αλλά ο slave βρεθεί με διαφορετική διεύθυνση, τότε εμφανίζεται parameterization fault. Γενικά τα 6 standard diagnostic bytes χρησιμοποιούνται για να εκδώσουν σφάλματα που είναι κοινά σε όλες τις slave συσκευές.

Ο κατασκευαστής μπορεί να χρησιμοποιήσει τα διαγνωστικά που σχετίζονται με τη συσκευή ώστε να εκδώσει πληροφορίες που ίσως είναι ειδικές σε σχέση με τη συγκεκριμένη συσκευή και δεν μπορούν να εκδοθούν χρησιμοποιώντας τα standard διαγνωστικά ή αυτά που σχετίζονται με το κανάλι επικοινωνίας. Αυτού του είδους τα διαγνωστικά προσδιορίζονται από τον κατασκευαστή. Η δομή τους δεν προσδιορίζεται στο πρότυπο του Profibus. Για αυτό το λόγο πρέπει να ελέγχεται το documentation της συσκευής για το προσδιορισμό της ακριβής μορφής τους.

Τα modules διαγνωστικά χρησιμοποιούνται για τα διαγνωστικά ενός modular slave, δηλαδή ενός slave που αποτελείται από ένα κεντρικό module και από άλλα δευτερεύοντα. Αυτού του είδους τα διαγνωστικά δίνουν την δυνατότητα στο κεντρικό module να κάνει αναφορά ότι ένα συγκεκριμένο δευτερεύον module έχει κάποιο διαγνωστικό. Βέβαια δεν αναφέρεται ακριβώς τι πρόβλημα υπάρχει, απλά ότι το συγκεκριμένο module έχει πρόβλημα. Αυτό σημαίνει ότι κάποιο λογικό block είναι δομημένο ώστε να αποκωδικοποιεί αυτού του τύπου διαγνωστικές πληροφορίες. Αυτή η λογική αποκωδικοποίησης δουλεύει για κάθε συσκευή, οποιουδήποτε κατασκευαστή, που παρέχει διαγνωστικά σχετιζόμενα με modules.

Ο τελευταίος τύπος διαγνωστικών είναι αυτός που δίνει διαγνωστικά για το κανάλι. Μια συσκευή μπορεί να χρησιμοποιήσει αυτού του είδους τα διαγνωστικά

για να αναφέρει ότι ένα μεμονωμένο κανάλι ενός ειδικού module έχει πρόβλημα βραχυκυκλώματος, υπέρτασης ή κάποιο κομμένο καλώδιο. Αυτό είναι πολύ χρήσιμο για να διαγνωστούν προβλήματα στο επίπεδο καλωδίωσης. Τέτοιου είδους προβλήματα προσδιορίζονται και στα standard διαγνωστικά.

Ο κατασκευαστής έχει την επιλογή να συμπεριλάβει και μερικές πρόσθετες λειτουργίες στις συσκευές. Τέτοιες λειτουργίες είναι π.χ. ότι επιτρέπεται να τοποθετηθεί συγκεκριμένη διεύθυνση πάνω στο δίαυλο επικοινωνίας, να υποστηρίζει στις εξόδους το χαρακτηριστικό SYNC και στις εισόδους το χαρακτηριστικό FREEZE. Σε περίπτωση που υποστηρίζονται, ο κατασκευαστής τοποθετεί λέξεις κλειδιά που εμφανίζονται στο GSD αρχείο.

Υπάρχει μια πολύ στοιχειώδης δυνατότητα ελέγχου της κίνησης ώστε να συγχρονίζει εξόδους. Αυτή η λειτουργία SYNC επιτρέπει για παράδειγμα σε ένα group κινητήρων πάνω σε μία μεταφορική ταινία να συγχρονίσουν την ταχύτητα τους. Μια SYNC λειτουργία μπορεί να αρχικοποιηθεί μέσω μιας εντολής στο πρόγραμμα ελέγχου, με την προϋπόθεση ότι το υποστηρίζει ο master. Αυτό εξαναγκάζει όλες τις άμεσα επηρεαζόμενες συσκευές σε άμεση διαδικασία μέσω των τελευταίων δεδομένων εξόδου από τον master, π.χ. μια εντολή για ταχύτητα και κατεύθυνση. Ο master στέλνει αυτή την εντολή ανάμεσα σε κύκλους I/O. Οι συσκευές αγνοούν τα δεδομένα μέχρι το πρόγραμμα ελέγχου να προκαλέσει μια άλλη εντολή SYNC. Όταν η διαδικασία έντασης ολοκληρωθεί, το πρόγραμμα ελέγχου μπορεί να προκαλέσει μία εντολή UNSYNC και να επιστρέψουν οι συσκευές πίσω σε κανονική λειτουργία λήψης δεδομένων εξόδου.

Υπάρχει επίσης δυνατότητα, το πρόγραμμα ελέγχου να προκαλέσει σε ένα σύνολο συσκευών πάγωμα των εικόνων τους που στέλνουν στη master συσκευή. Αυτό δίνει τη δυνατότητα να πάρθει ένα στιγμιότυπο σε μια συγκεκριμένη χρονική στιγμή διαφορετικών αναλογικών εισόδων από διαφορετικές συσκευές. Πιθανό να χρειάζεται για κάποιους υπολογισμούς στο πρόγραμμα να περιλαμβάνονται αυτές οι αναλογικές τιμές και να πρέπει όλες οι τιμές να παγώσουν την ίδια χρονική στιγμή. Αυτή η λειτουργία παγώματος λειτουργεί όταν μια εντολή στο πρόγραμμα την αρχικοποιεί. Οι συσκευές συνεχίζουν να διαβάζουν τα πεδία εισόδου αλλά δεν αλλάζουν την εικόνα εξόδου τους προς το master. Όταν δε χρειάζεται άλλο να είναι παγωμένες και πρέπει να επιστρέψουν σε κανονική λειτουργία, ανανεώνουν την εικόνα που στέλνουν στο master εκτελώντας το πρόγραμμα μία εντολή UNFREEZE.

Κεφάλαιο 4: TIA PORTAL

4.1 Εισαγωγή

Η Siemens για την διαχείριση των λογικών ελεγκτών της χρησιμοποιούσε ως βασικό software το Simatic Manager. Τα τελευταία χρόνια όμως εισήγαγε στην αγορά ένα καινούργιο engineering software για την διαχείρισή τους, το TIA portal. Τα αρχικά του TIA portal προέρχονται από το «totally integrated automation portal», το οποίο συνεπάγεται ένα βιομηχανικό λογισμικό κατάλληλο για όλα τα έργα αυτοματισμού όπως λογικούς ελεγκτές, HMI panels και drivers. Σε ένα λογισμικό ενσωματώνεται ο έλεγχος όλων των συσκευών που απαιτούνται για τις ανάγκες του αυτοματισμού. Πιο συγκεκριμένα είναι software για προγραμματισμό, ρύθμιση και έλεγχο όλων των controllers της οικογενείας Simatic της Siemens καθώς και τον έλεγχο άλλων βιομηχανικών στοιχείων όπως σερβοκινητήρες και stepper motors.

Η ενσωμάτωση διαχείρισης όλων των συσκευών αυτοματισμού από ένα κοινό software, έχει ως αποτέλεσμα μείωση κόστους, μείωση χρόνου αλλά και πραγματοποίηση ενός πλήρους έργου αυτοματισμού με πολύ πιο απλό και αποδοτικό τρόπο.

Στο TIA portal πραγματοποιείται χρήση πολλών editors μαζί, όπως του program editor (δυνατότητα πολλαπλής επιλογής όπως ladder, STL, FBD, SCL, Graph), του device editor, του watch table editor, του WinCC editor (για διαχείριση των HMI). Αποτέλεσμα αυτού όλα τα δεδομένα να βρίσκονται σε μία σταθερή βάση και σε ένα κοινό περιβάλλον, κάτι που παρέχει πολύ μεγάλη διευκόλυνση στο χρήστη.

Ο προγραμματισμός είναι προσαρμοσμένος στις τεχνολογίες των Windows, μιας και οι περισσότεροι χρήστες είναι εξοικειωμένοι στην χρήση αυτών. Παραδείγματα αποτελούν το drag & drop, με το οποίο μεταφέρεις δεδομένα από τον ένα editor του προγράμματος στον άλλο, καθώς και το ότι όταν υπάρχουν πολλά παράθυρα από διαφορετικούς editors ανοιχτά, αυτά εμφανίζονται σε μία λίστα στο κάτω μέρος του παραθύρου του TIA portal για εύκολη πρόσβαση από το ένα στο άλλο. Επιπλέον παρέχεται η δυνατότητα επιλογής της γλώσσας του interface του TIA portal. Οι γλώσσες επιλογής είναι Αγγλικά, Γερμανικά, Γαλλικά, Ισπανικά, Ιταλικά και Κινέζικα.

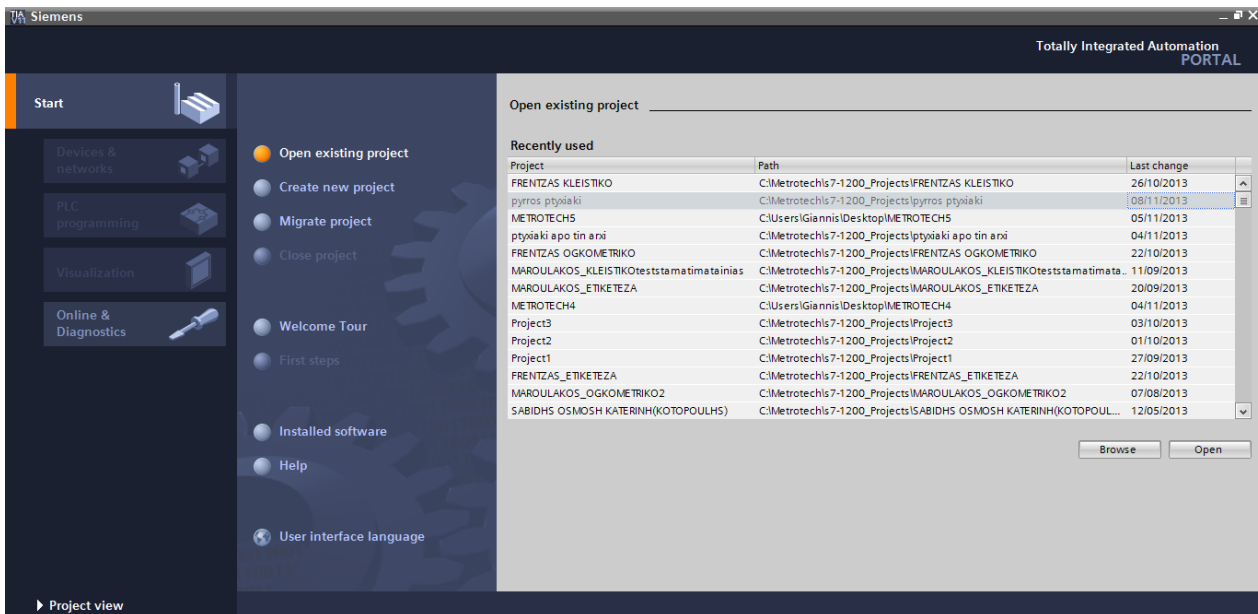
4.2 Δημιουργία προγράμματος & εισαγωγή PLC

Ο χρήστης έχει τη δυνατότητα προβολής του προγράμματος σε δύο όψεις:

- Την όψη του Portal view
- Την όψη του Project view

Η επιλογή γίνεται με βάση τις προτιμήσεις του χρήστη και τίποτα άλλο. Οι διαφορές θα φανούν παρακάτω. Στην προκειμένη περίπτωση θα χρησιμοποιηθεί η Portal view για την αρχική εγκατάσταση των συσκευών και η Project view για όλον τον υπόλοιπο προγραμματισμό.

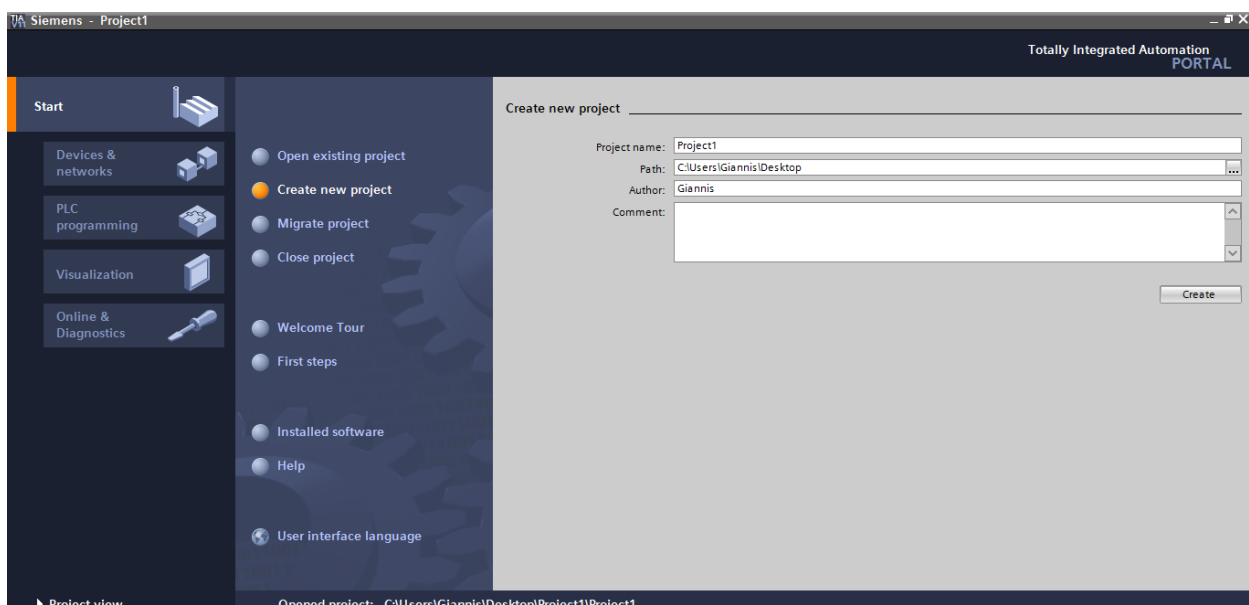
Κάνοντας διπλό αριστερό κλικ στο εικονίδιο συντόμευσης του TIA portal στην επιφάνεια εργασίας ανοίγει το πρόγραμμα και εμφανίζεται η παρακάτω οθόνη (εικόνα 4.1).



Εικόνα 4.1: Εισαγωγική οθόνη του TIA portal

Για τη δημιουργία νέου project πιέζουμε «Create new project». Στη συνέχεια εμφανίζεται μία οθόνη (εικόνα 4.2) στην οποία εισάγεται το όνομα του project, το μέρος που θα αποθηκευτεί και κάποιο προσωπικό σχόλιο που επιθυμούμε να εγχωρίσουμε σε αυτό. Μόλις εισάγουμε τα παραπάνω στοιχεία, πιέζουμε «create».

Η όψη αυτή του προγράμματος είναι η «Portal view», η οποία μας δίνει την δυνατότητα να περιηγηθούμε στο project με τον τρόπο που θα περιγραφεί.

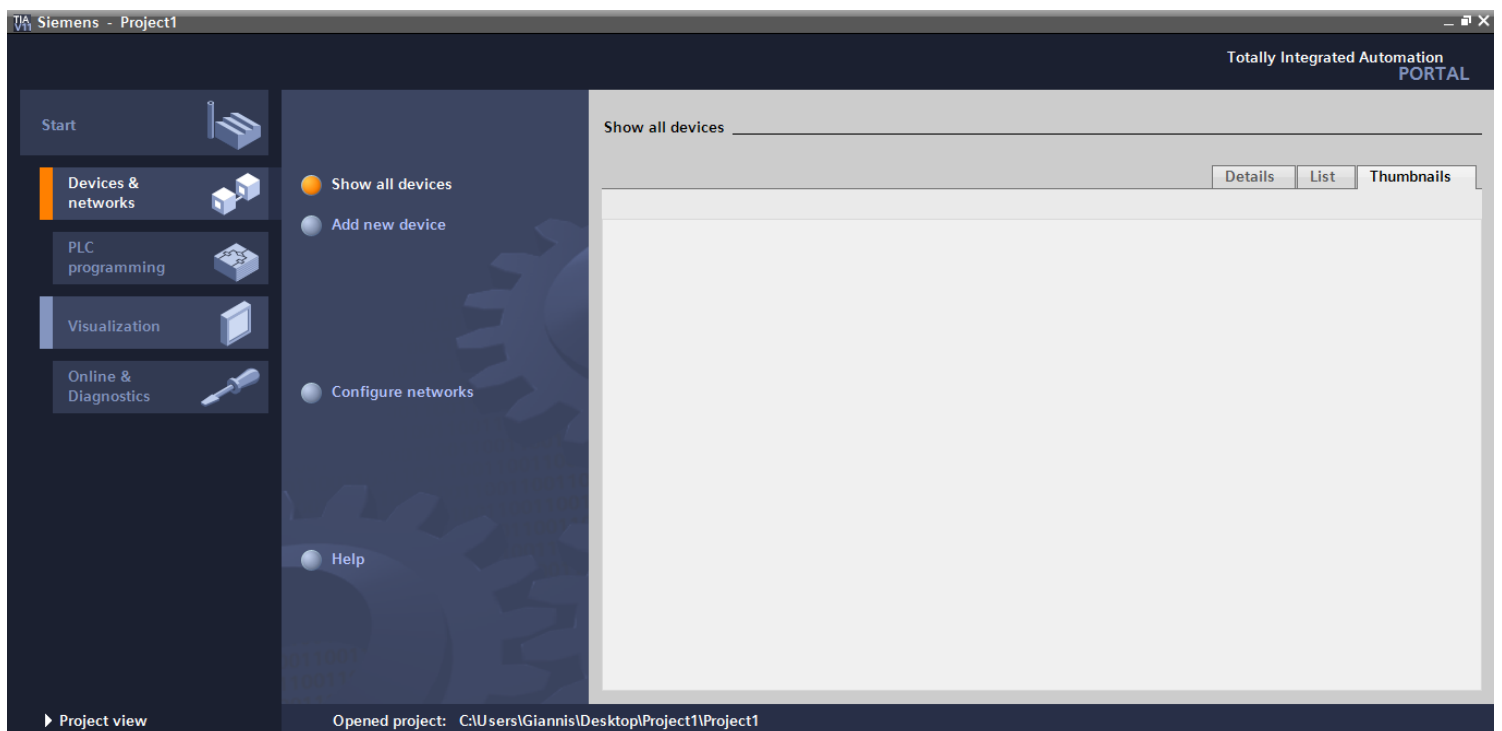


Εικόνα 4.2: Οθόνη δημιουργίας νέου project



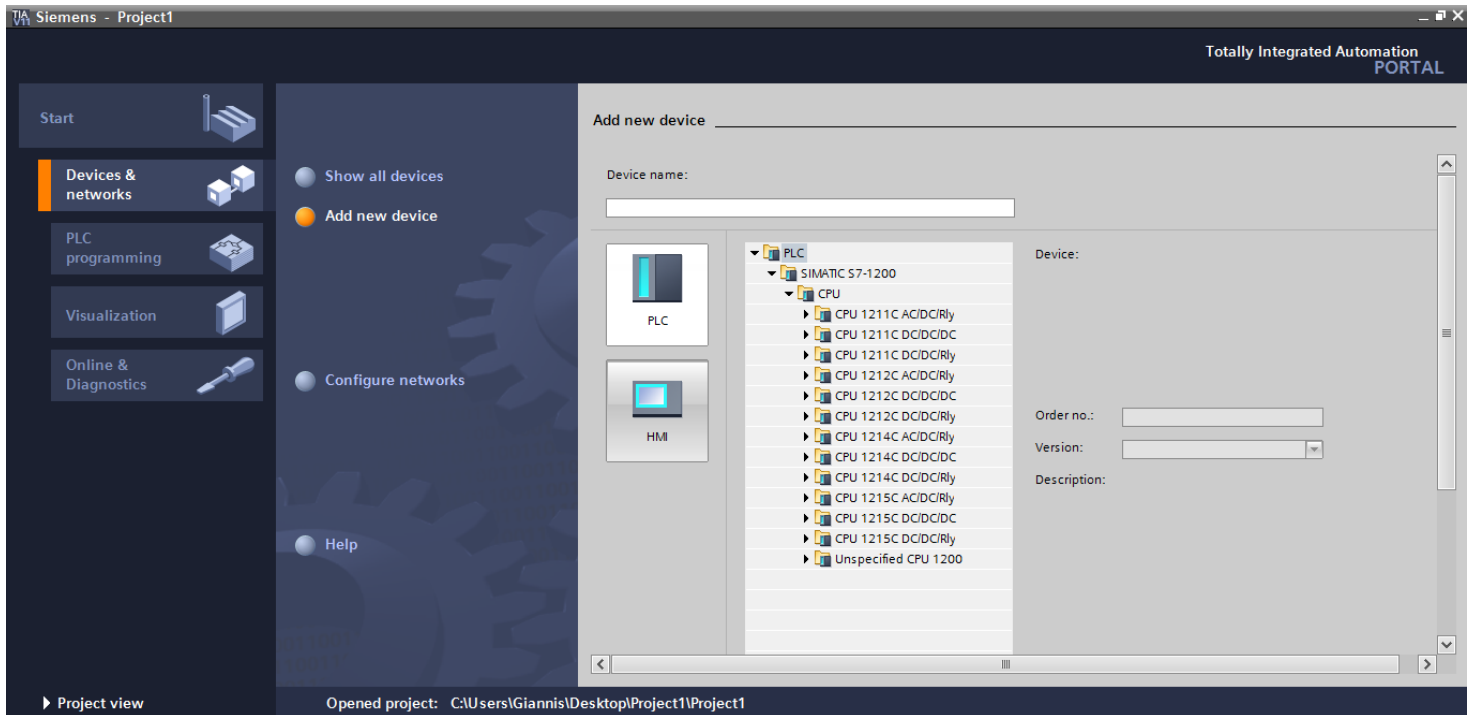
Εικόνα 4.3: Όψη Portal view για συγκεκριμένο πρόγραμμα

Πιέζουμε «Devices and networks (configure a device)» στα δεξιά της οθόνης (εικόνα 4.3) για να δούμε τις συσκευές που συμμετέχουν στο project ή να εισάγουμε κάποια καινούργια. Η οθόνη που εμφανίζεται φαίνεται παρακάτω (εικόνα 4.4).



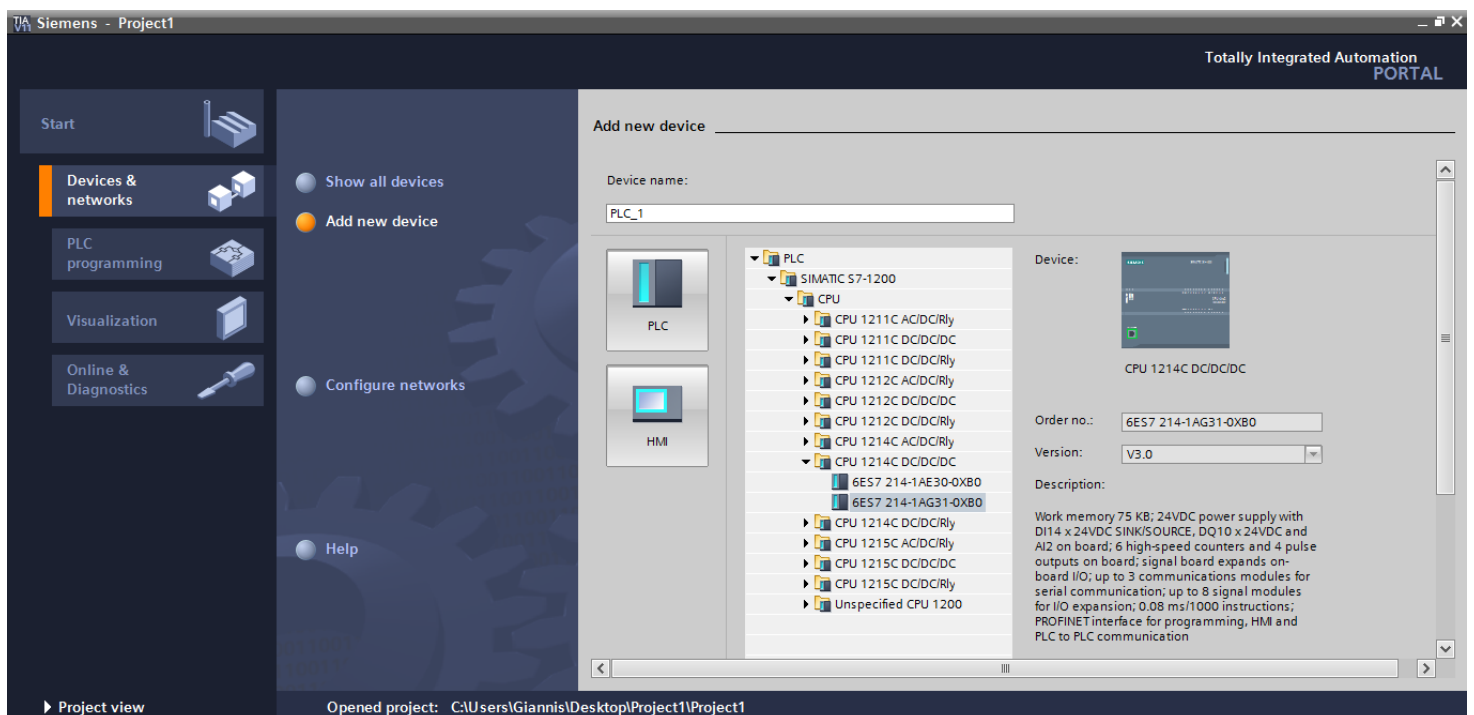
Εικόνα 4.4: Οθόνη για εισαγωγή νέας συσκευής και εμφάνιση των ήδη υπαρχόντων στο project

Επιλέγουμε «Add new device» και εμφανίζεται η οθόνη (εικόνα 4.5) όπου επιλέγουμε αν θέλουμε να εισάγουμε κάποιο PLC ή κάποια HMI.



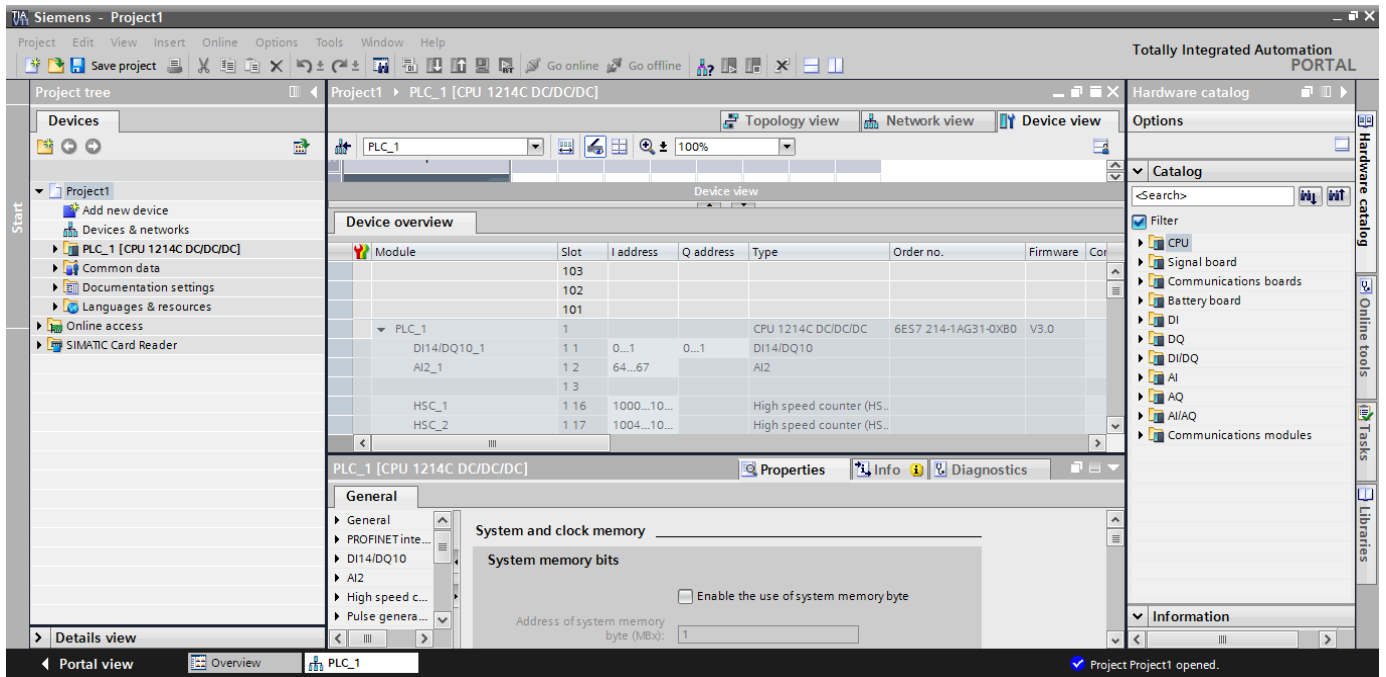
Εικόνα 4.5: Επιλογή εισαγωγής PLC ή HMI

Αρχικά επιλέγουμε το PLC που θα χρησιμοποιήσουμε στην υλοποίηση του συγκεκριμένου αυτοματισμού (εικόνα 4.6).



Εικόνα 4.6: Επιλογή συγκεκριμένου PLC στο πρόγραμμα

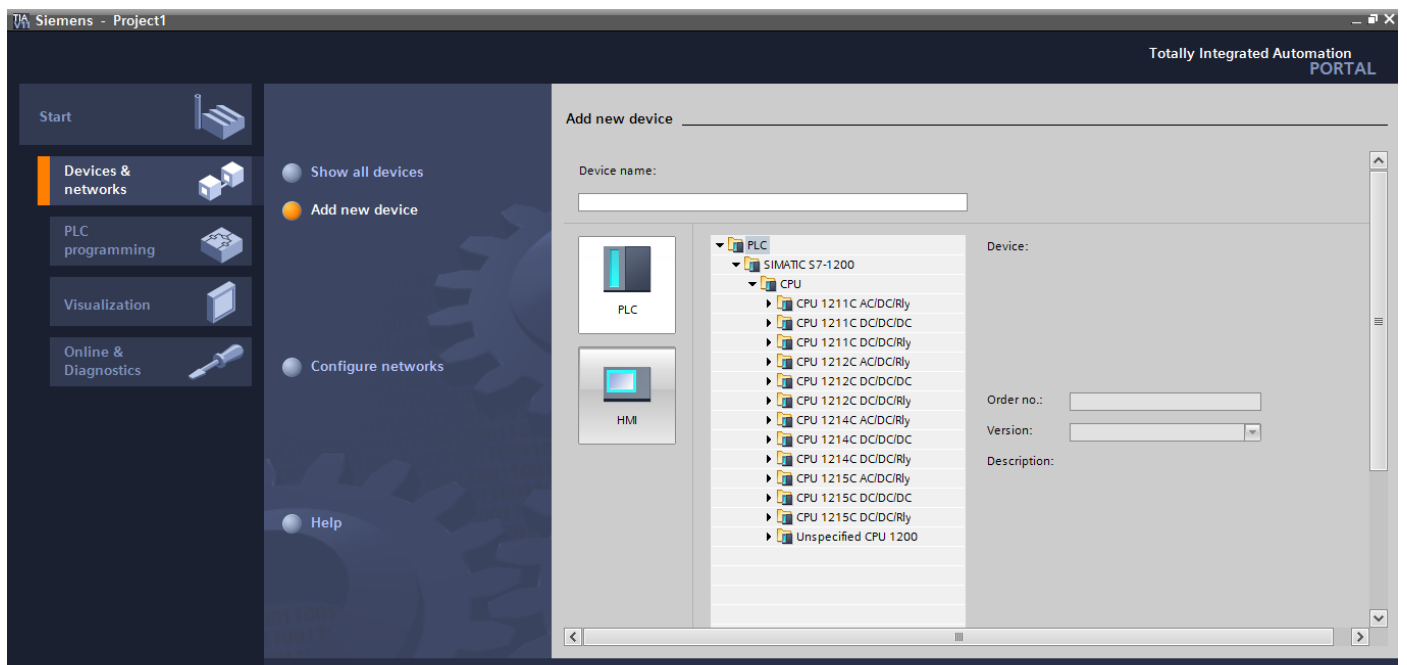
Δίνονται τα βασικά στοιχεία του συγκεκριμένου PLC (εικόνα 4.6), όπως ο κωδικός παραγγελίας του, η έκδοσή του και τα τεχνικά χαρακτηριστικά του. Η επιλογή γίνεται καθαρά βάση των αναγκών και των ιδιοτεροτήτων του αυτοματισμού που πρόκειται να υλοποιηθεί. Πιέζοντας «add» ή κάνοντας διπλό κλικ πάνω στον κωδικό του PLC (εικόνα 4.6) πραγματοποιείται η εισαγωγή του. Στη συνέχεια εμφανίζεται η όψη Project view του TIA portal (εικόνα 4.7).



Εικόνα 4.7: Εισαγωγική οθόνη του project view

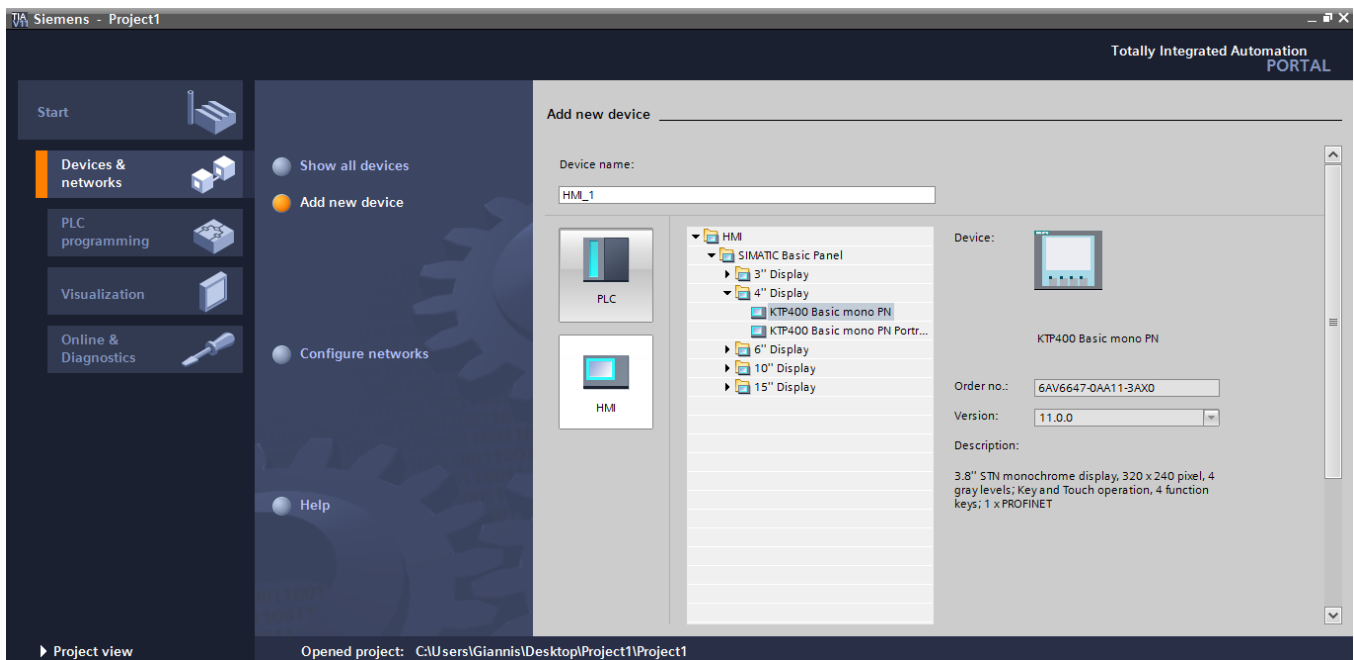
4.3 Εισαγωγή HMI & χρήση του HMI Device Wizard

Στη προηγούμενη οθόνη (εικόνα 4.7) πραγματοποιείται όλος ο βασικός προγραμματισμός. Σε περίπτωση εισαγωγής κάποιου επιπλέον PLC ή κάποιας HMI, πιέζοντας «Portal view» στο κάτω αριστερά μέρος της οθόνης μεταφερόμαστε στην οθόνη επιλογής συσκευής, δηλαδή ξανά στο Portal view.



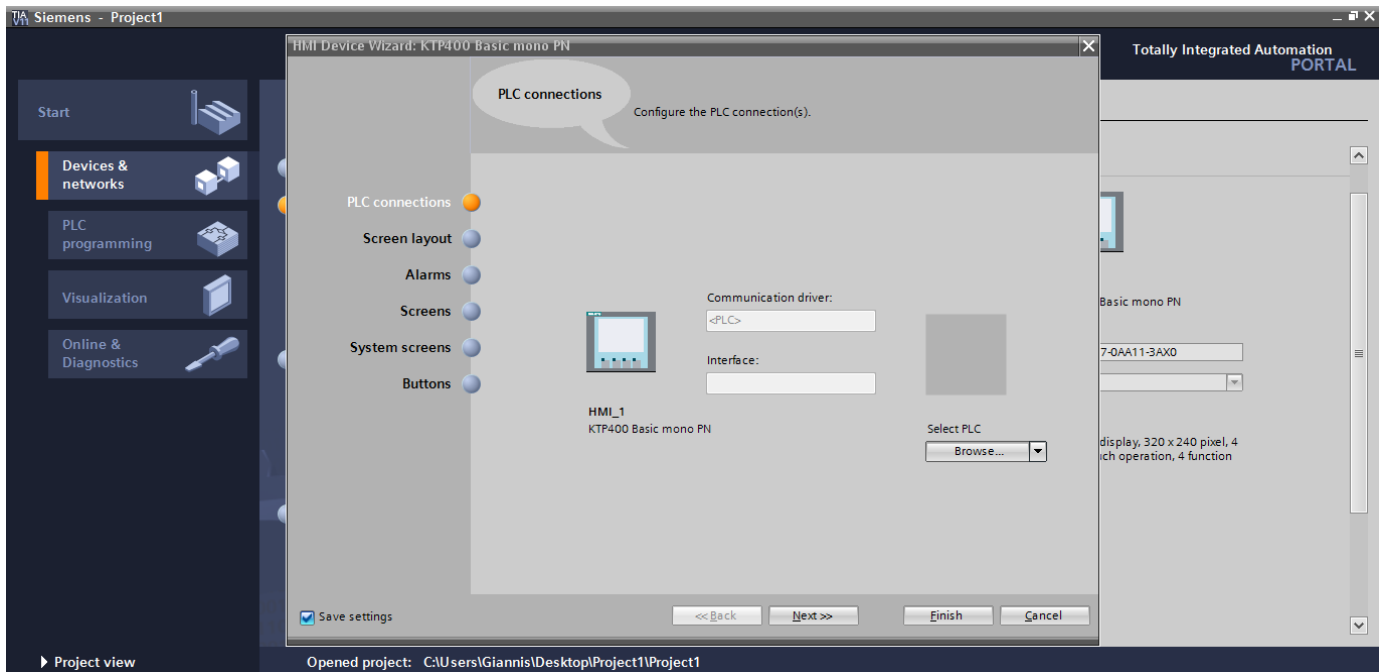
Εικόνα 4.8: Επιλογή συγκεκριμένου PLC στο πρόγραμμα

Για την εισαγωγή μίας HMI, επιλέγουμε «HMI» (εικόνα 4.8) και διαλέγουμε το συγκεκριμένο μοντέλο που θα χρησιμοποιηθεί στο project (εικόνα 4.9).



Εικόνα 4.9: Επιλογή συγκεκριμένης HMI στο πρόγραμμα

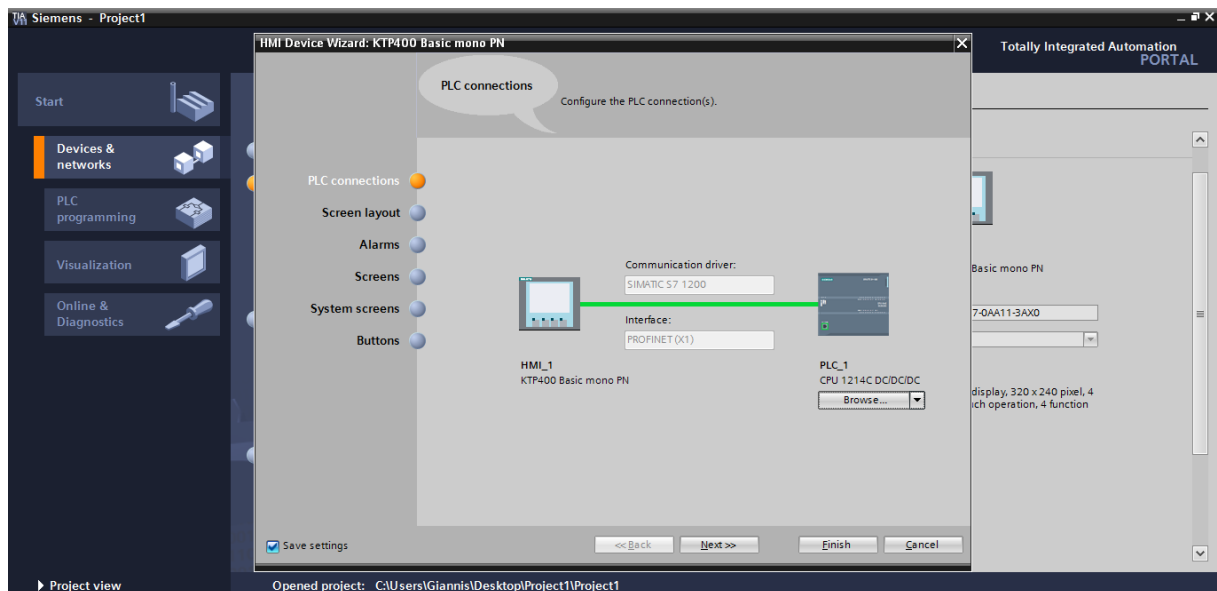
Παρόμοια με τα PLC, δίνεται ο αριθμός παραγγελίας της HMI, η έκδοση της, καθώς και κάποια βασικά χαρακτηριστικά και δυνατότητες αυτής. Με διπλό κλικ ή «add» εμφανίζεται το «HMI Device Wizard» (εικόνα 4.10), το οποίο παρέχει διάφορα default βήματα για τις βασικές λειτουργίες και τη δομή της συγκεκριμένης HMI στο συγκεκριμένο project.



Εικόνα 4.10: Βήμα 1^ο του HMI Device Wizard

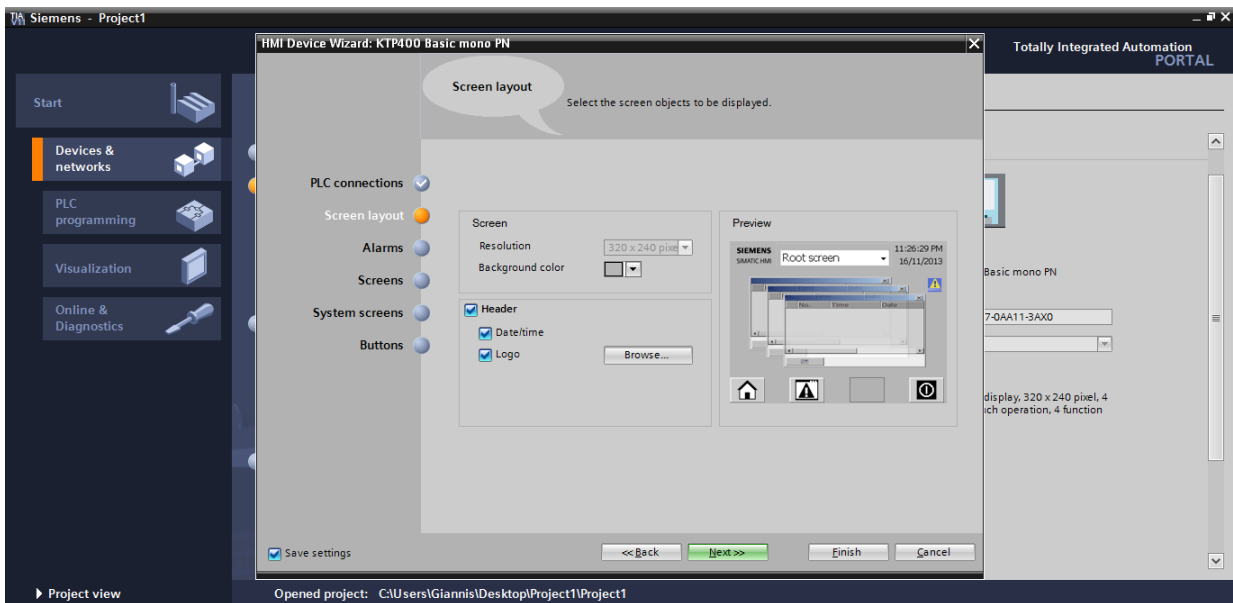
Στο πρώτο βήμα, στο «PLC connections» (εικόνα 4.10), επιλέγεται με ποιο PLC του project θα συνδεθεί η συγκεκριμένη HMI. Η επιλογή γίνεται πιέζοντας «browse» .

Μόλις γίνει η επιλογή, εμφανίζεται η παρακάτω οθόνη (εικόνα 4.11) που απεικνίζει τη σύνδεση του PLC με την συγκεκριμένη HMI.



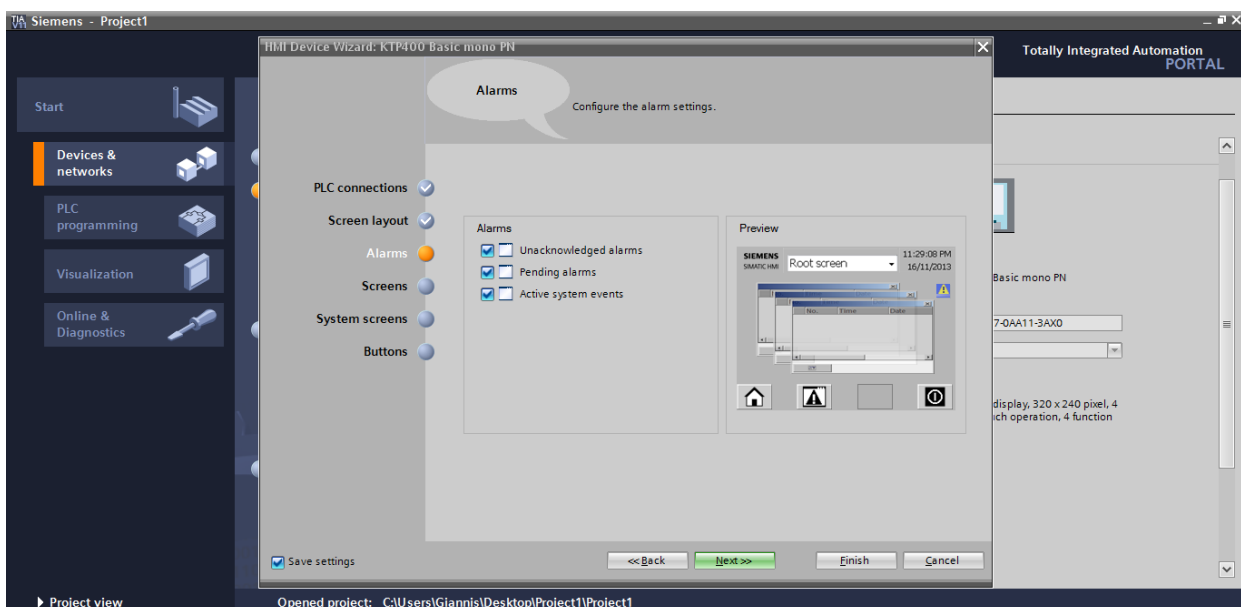
Εικόνα 4.11 : Οθόνη διασύνδεσης HMI με PLC

Πιέζοντας «next» (εικόνα 4.11) μεταφερόμαστε στο επόμενο βήμα, που είναι το «Screen layout» (εικόνα 4.12). Σε αυτό το βήμα επιλέγεται το χρώμα που θα έχουν οι οθόνες στο background τους, η εμφάνιση ή όχι της ημερομηνίας, καθώς και η δυνατότητα εμφάνισης κάποιου λογότυπου επιλογής μας στις οθόνες της HMI.



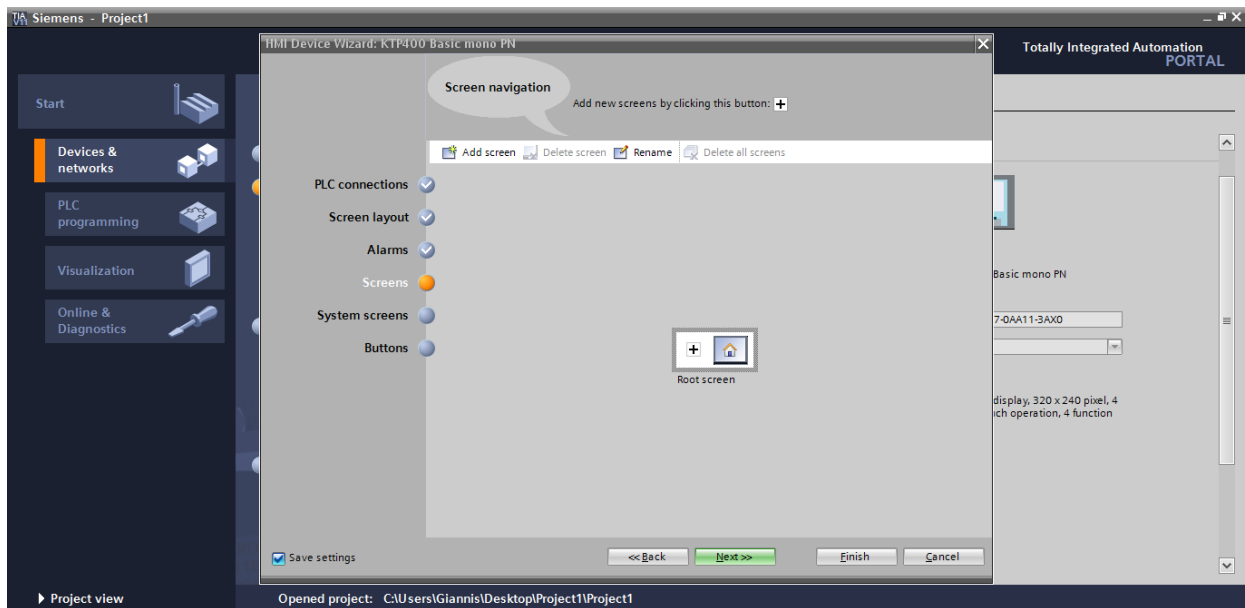
Εικόνα 4.12: Βήμα 2^ο του HMI Device Wizard

Μόλις πραγματοποιηθούν οι παραπάνω επιλογές, πιέζουμε «next» και μεταφερόμαστε στο επόμενο βήμα, τα «Alarms» (εικόνα 4.13). Σε αυτό το βήμα επιλέγεται αν θα εμφανίζονται διάφορα alarms κατά τη λειτουργία της HMI, όπως το ότι έχει ενεργοποιηθεί η σύνδεση με το PLC.



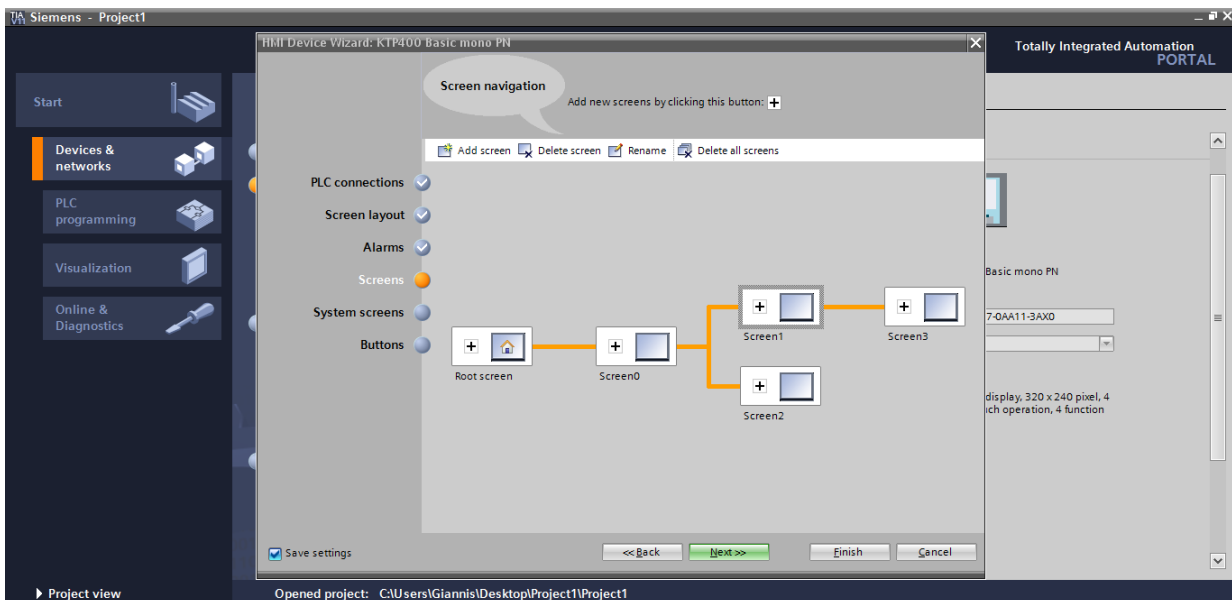
Εικόνα 4.13: Βήμα 3^ο του HMI Device Wizard

Στη συνέχεια πιέζουμε «next». Το επόμενο βήμα με ονομασία «Screens» (εικόνα 4.14) παρέχει τη δυνατότητα δημιουργίας της δομής των οθονών στο project, δηλαδή ποια οθόνη οδηγεί σε ποια.



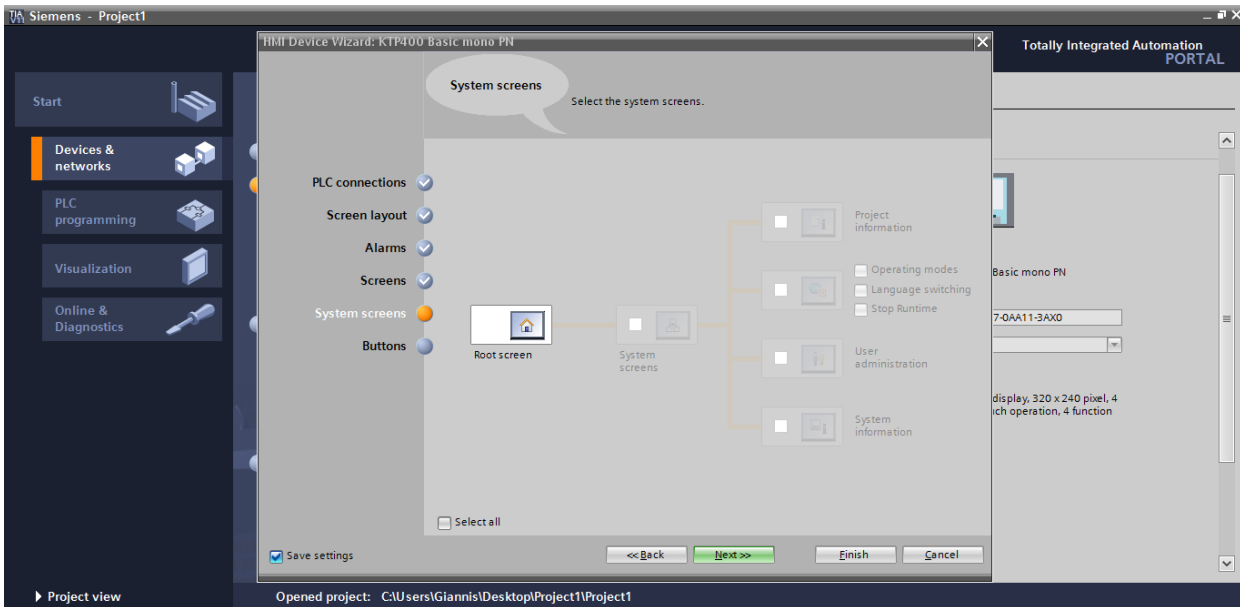
Εικόνα 4.14: Βήμα 4^ο του HMI Device Wizard

Έστω ότι δημιουργούμε τη δομή οθονών που επιθυμούμε, που θα έχει μια μορφή παρόμοια με την παρακάτω (εικόνα 4.15).



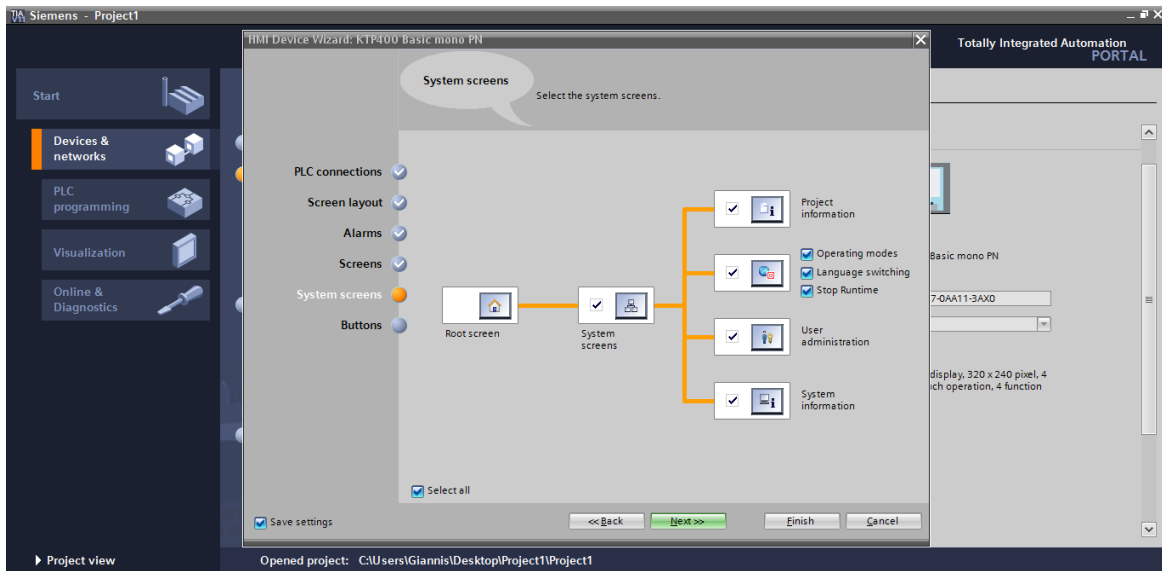
Εικόνα 4.15: Τυπική δομή οθονών σε μια HMI

Μόλις δημιουργήσουμε την επιθυμητή δομή πιέζουμε «next» και μεταφερόμαστε στο επόμενο βήμα, το «System screens» (εικόνα 4.16).



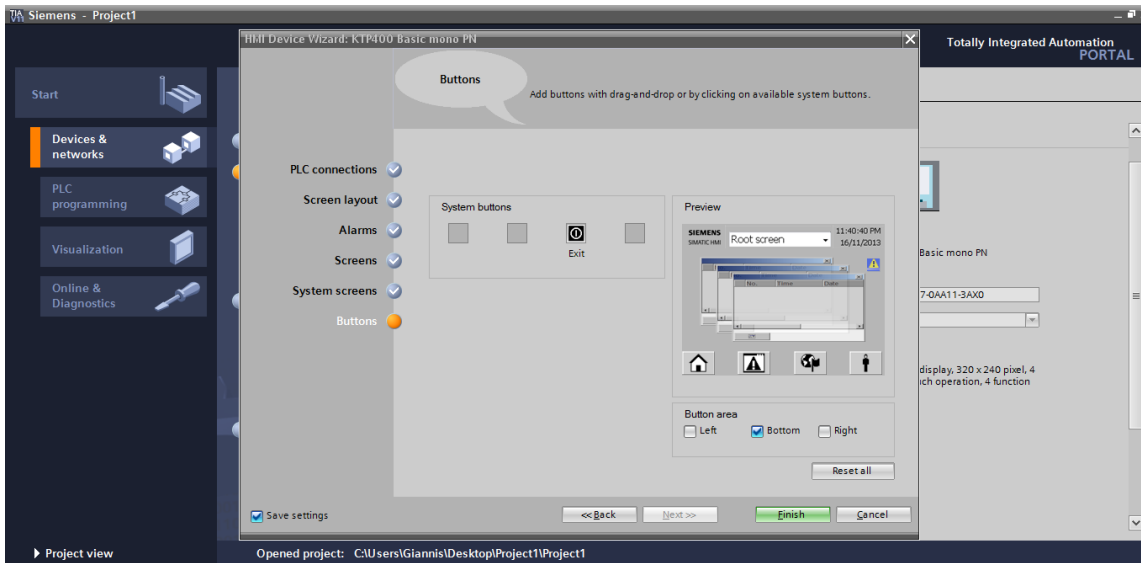
Εικόνα 4.16: Βήμα 5^ο του HMI Device Wizard

Σε αυτό το βήμα δίνεται η δυνατότητα να συμπεριλάβουμε στην εφαρμογή μας μια ποικιλία από οθόνες του συστήματος (εικόνα 4.17).



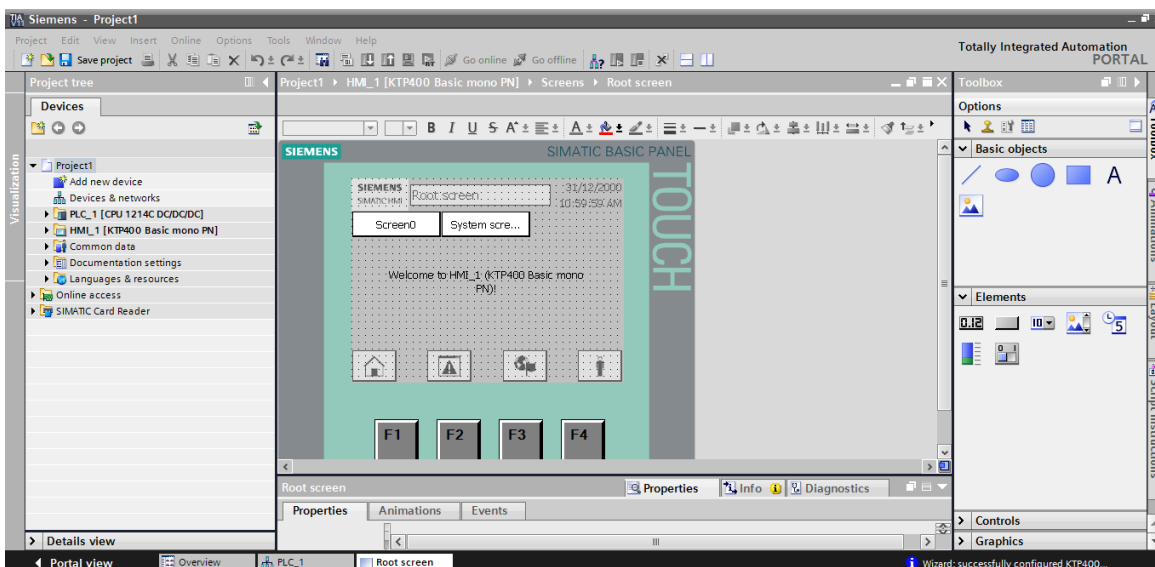
Εικόνα 4.17: Δυνατές οθόνες του συστήματος που μπορεί να συμπεριλάβει ο χρήστης στο πρόγραμμά του

Μόλις επιλέξουμε αυτές που επιθυμούμε, πιέζουμε «next» και μεταφερόμαστε στο τελευταίο βήμα του «HMI Device Wizard», που είναι το «Buttons» (εικόνα 4.18).



Εικόνα 4.18: Βήμα 6^ο του HMI Device Wizard

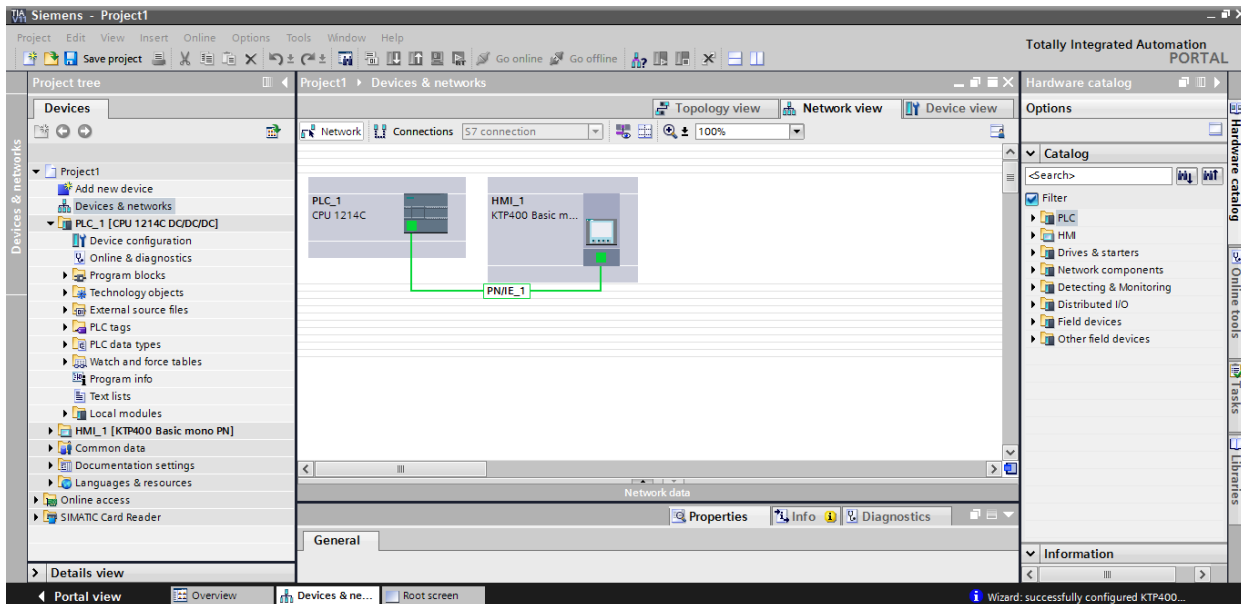
Σε αυτό το βήμα επιλέγουμε κάποια buttons με συγκεκριμένες δυνατότητες, όπως start screen, log on, language και exit. Επιπλέον δίνεται η δυνατότητα επιλογής σε ποια θέση της οθόνης να εμφανίζονται αυτά τα buttons (δεξιά, αριστερά ή κάτω). Μόλις τελειώσουμε με τις επιλογές μας, πιέζουμε «finish» και μεταφερόμαστε πίσω στο Project view έχοντας εισάγει και την HMI (εικόνα 4.19).



Εικόνα 4.19: Οθόνη του project view μετά την εισαγωγή κάποιας HMI

4.4 Βασικές δυνατότητες του Project view

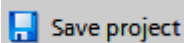
Σε αυτό το κεφάλαιο θα παρουσιαστούν οι βασικές δυνατότητες προγραμματισμού που παρέχει το TIA portal και ο τρόπος με τον οποίο υλοποιούνται. Παρουσιάζεται (εικόνα 4.20) η κεντρική οθόνη του Project view, όπου πραγματοποιείται ο προγραμματισμός του PLC.

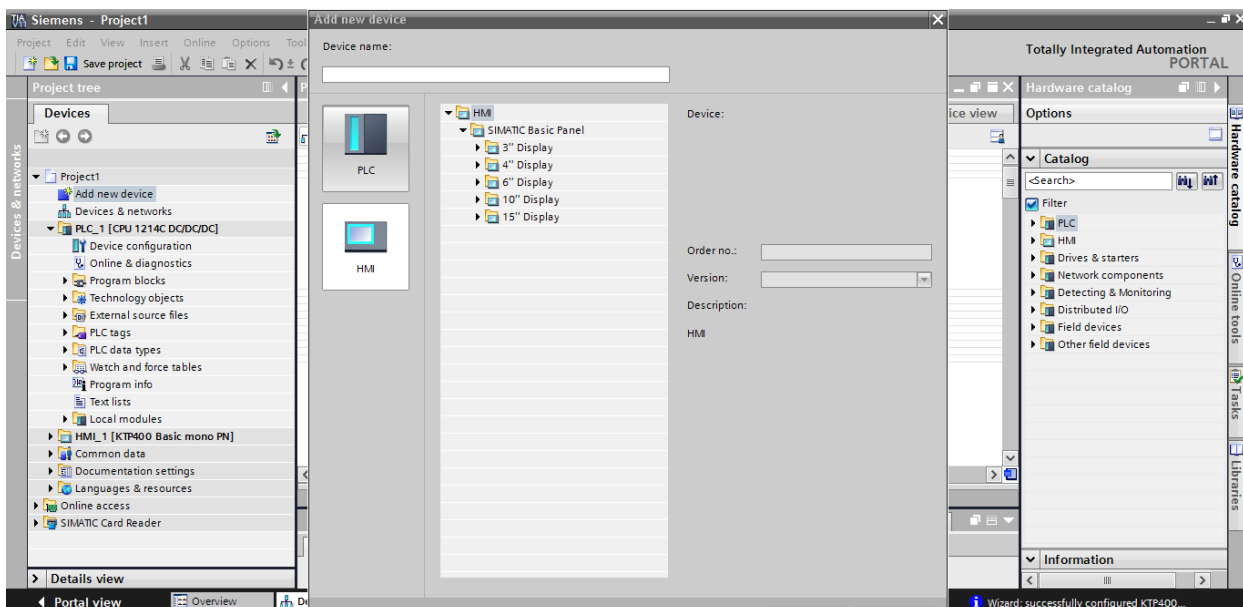


Εικόνα 4.20: Οθόνη του network view στο project view

Στην αριστερή πλευρά εμφανίζεται το Project tree (εικόνα 4.20). Περιέχει όλη τη δομή του project και οργανωμένα όλα τα αντικείμενα. Οι περισσότερες δυνατότητες επιλέγονται από το Project tree.

Οι πιο βασικές ενέργειες αναλύονται παρακάτω:

- Πιέζοντας το  εικονίδιο ή από το μενού «Project», σώζουμε το project στην κατάσταση που είναι εκείνη τη στιγμή.
- Πιέζοντας «add new device» δίνεται η δυνατότητα εισαγωγής κάποιου PLC ή κάποιας HMI μέσω του Project view (εικόνα 4.21), με παρόμοιο τρόπο με



Εικόνα 4.21: Εισαγωγή PLC ή HMI μέσω του project view

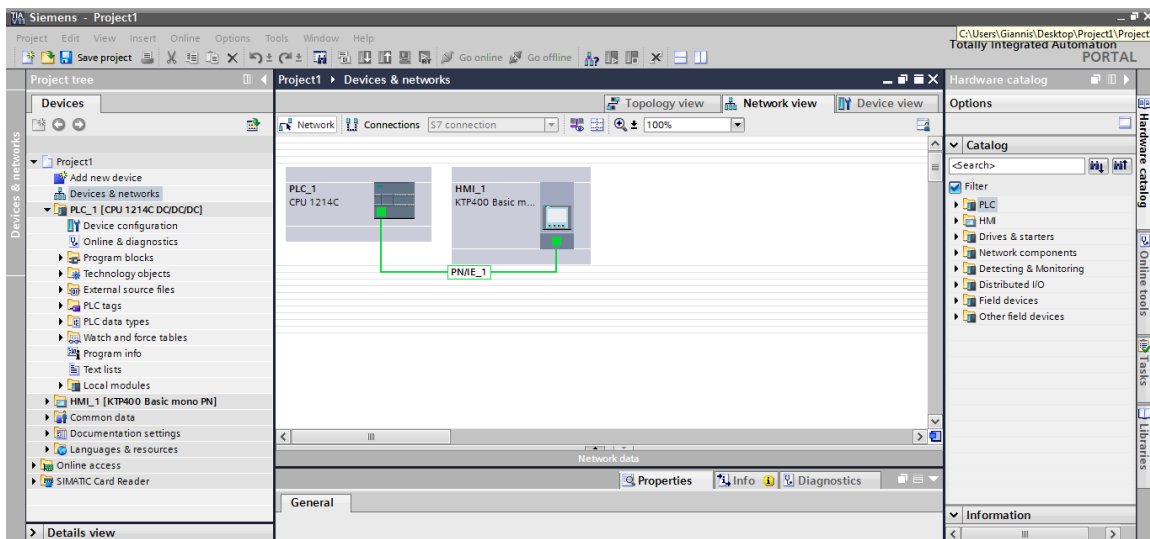
αυτών που περιγράφηκε στο Portal view.

- Πιέζοντας «Devices & networks» μεταφερόμαστε στο Network & device configuration, από όπου γίνεται εύκολα η εγκατάσταση και η παραμετροποίηση των συσκευών. Παράλληλα φαίνεται γραφικά το υπάρχον network.

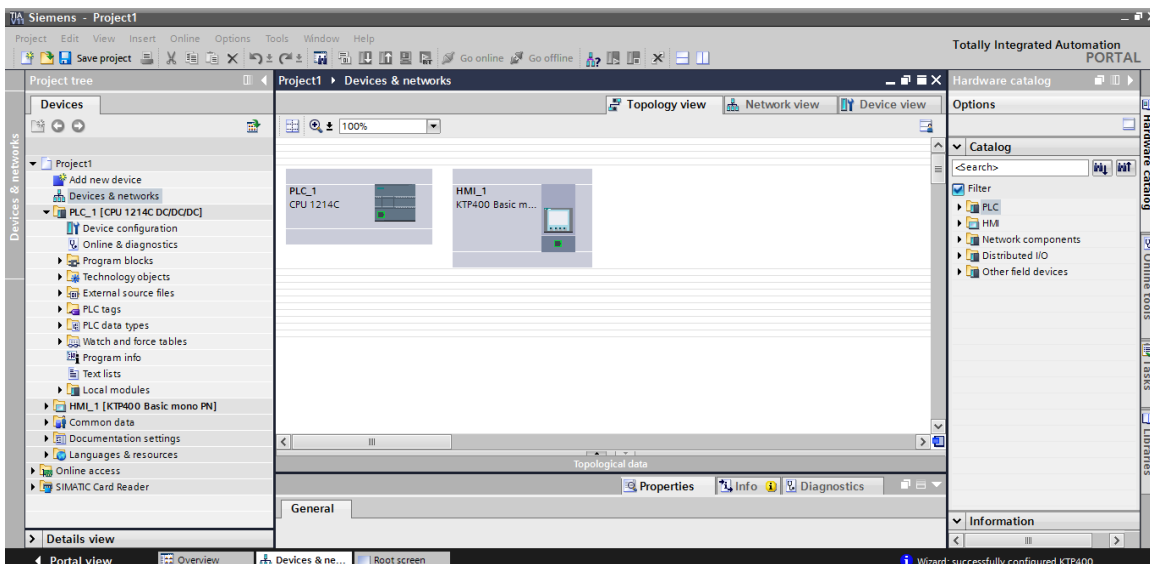
Υπάρχουν τρεις διαφορετικές όψεις:

- **Network view:** Δείχνει τη γραφική απεικόνιση όλου του hardware configuration που βασίζεται σε σύνδεση Profinet, Profibus ή AS-i. Δεξιά εμφανίζεται το αντίστοιχο task για την εισαγωγή οποιασδήποτε συσκευής συμμετέχει στο project όπως PLC, HMI, κινητήρες, ενδείκτες κλπ.
- **Topology view:** Φαίνεται μόνο η διασύνδεση των Profinet devices.
- **Device view:** Απεικονίζει μόνο την συγκεκριμένη συσκευή που επιλέχθηκε και δίνεται η δυνατότητα παραμετροποίησης αυτής.

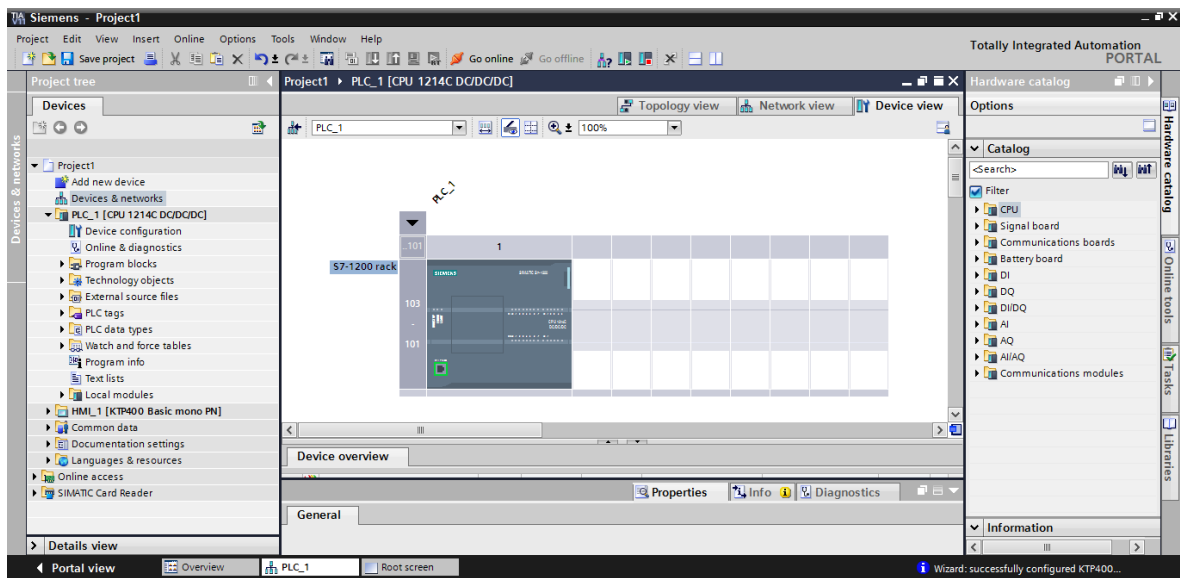
Όταν προσθέσουμε κάποιο PLC από το project view, για να το συνδέσουμε με κάποια HMI, στο «network view» σέρνουμε μια γραμμή από το πράσινο κουτάκι του PLC στο πράσινο της HMI. Έτσι δημιουργείται μία network connection μεταξύ των δύο συσκευών και δίνεται η δυνατότητα να μοιράζονται τα αντίστοιχα tags.



Εικόνα 4.22: Network view

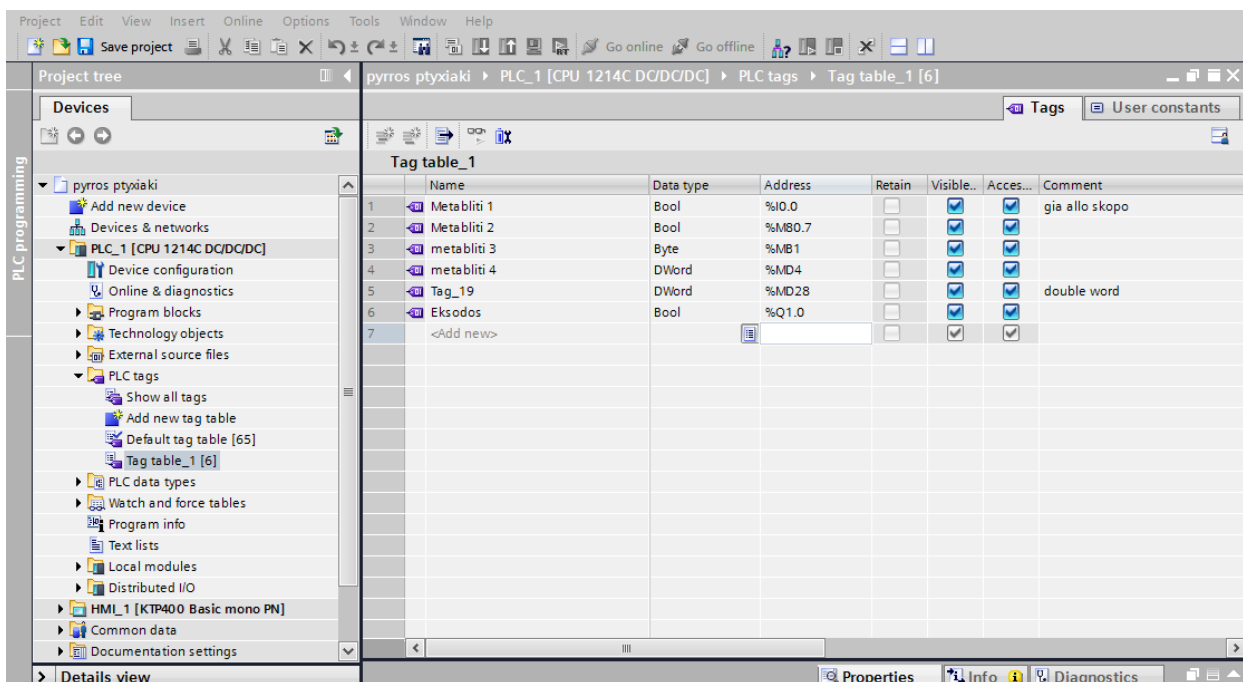


Εικόνα 4.23: Topology view



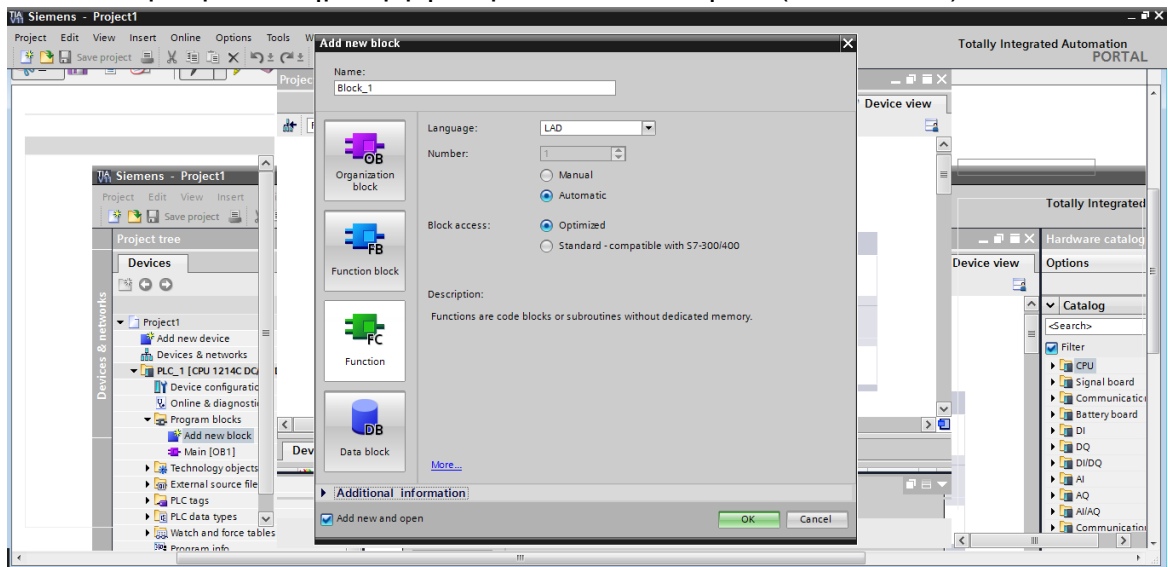
Εικόνα 4.24: Device view

- Πιέζοντας «PLC tags» εμφανίζεται το tag table (εικόνα 4.25), όπου λειτουργεί ως ένας κοινός editor, στον οποίο γίνεται η καταχώρηση όλων των μεταβλητών και σταθερών που χρησιμοποιούνται στο πρόγραμμα. Με αυτό το τρόπο μόλις αποθηκευτεί μια PLC tag σε κάποιο tag table, όλοι οι editors του project έχουν πρόσβαση σε αυτή τη μεταβλητή, λόγω του ότι μοιράζονται το tag table (μοιράζονται κοινή βάση δεδομένων). Με αυτό το τρόπο ενημερώνονται αυτόματα οι μεταβλητές και στο πρόγραμμα αλλά και στο hardware configuration και έτσι αποφεύγονται τυχόν εύκολα λάθη από απροσεξία. Επίσης παρέχεται εξοικονόμηση χρόνου. Παράλληλα δίνεται η δυνατότητα προσδιορισμού και εμφάνισης του τύπου (data type) κάθε μεταβλητής καθώς και η χρήση συμβολικών ονομάτων σε αυτές για πιο εύκολη κατανόηση του προγράμματος από τον χρήστη.



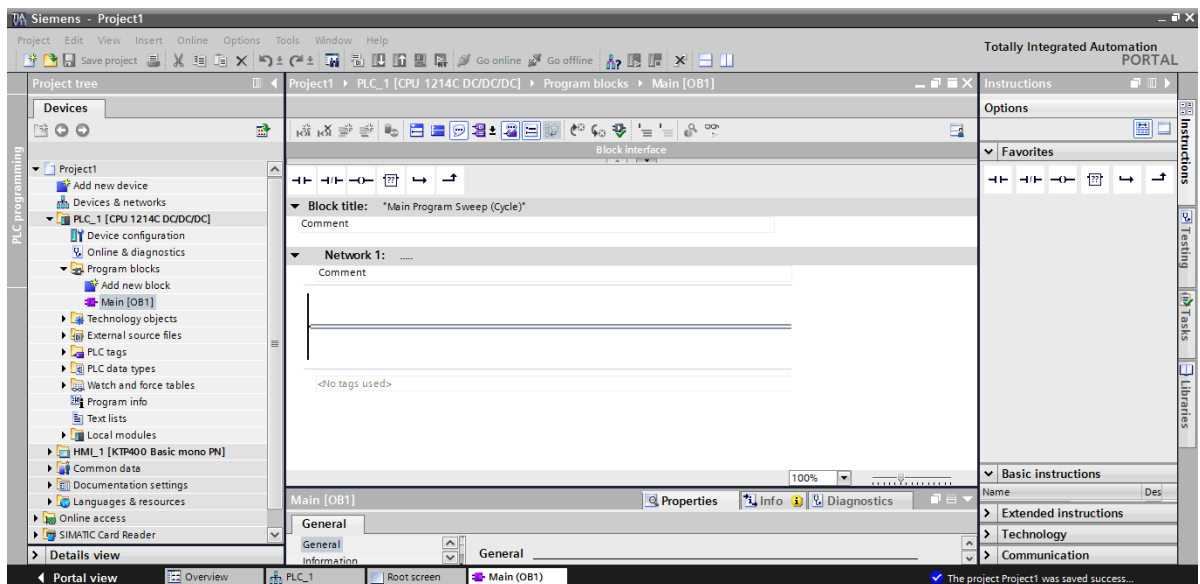
Εικόνα 4.25: Tag table του προγράμματος

- Κάνοντας διπλό κλικ στο Program blocks εμφανίζεται όλη η δομή του προγράμματος (OB1,DB1, κλπ.)
Κάνοντας διπλό κλικ στο «Add new block» επιλέγουμε το είδος του block που επιθυμούμε να δημιουργήσουμε, OB,FC,FB ή DB (εικόνα 4.26).



Εικόνα 4.26: Επιλογή δημιουργίας κάποιου program block

Κάνοντας διπλό κλικ στο OB1, εμφανίζεται ο program editor, δηλαδή εκεί που θα δημιουργηθεί και θα επεξεργαστεί ο κώδικας του προγράμματος.

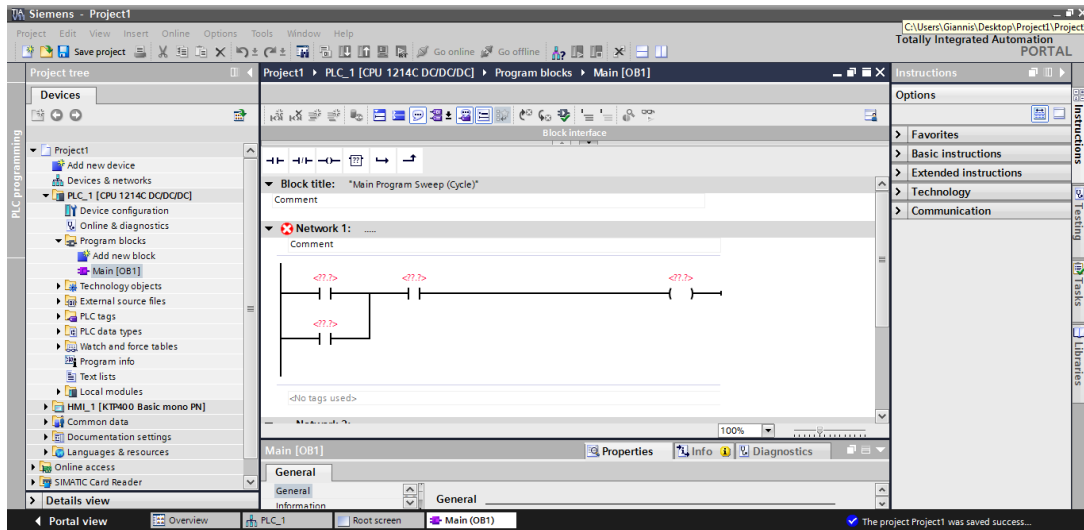


Εικόνα 4.27: Εμφάνιση του OB1, στον οποίο γίνεται και η σύνταξη του κώδικα

- Ο project editor είναι προσαρμοσμένος σε γλώσσα Ladder (κάτι το οποίο αλλάζει). Εδώ γίνεται η δημιουργία του προγράμματος, που δεν είναι κάτι άλλο από μια αλληλουχία από networks. Παρέχεται στα δεξιά task, που

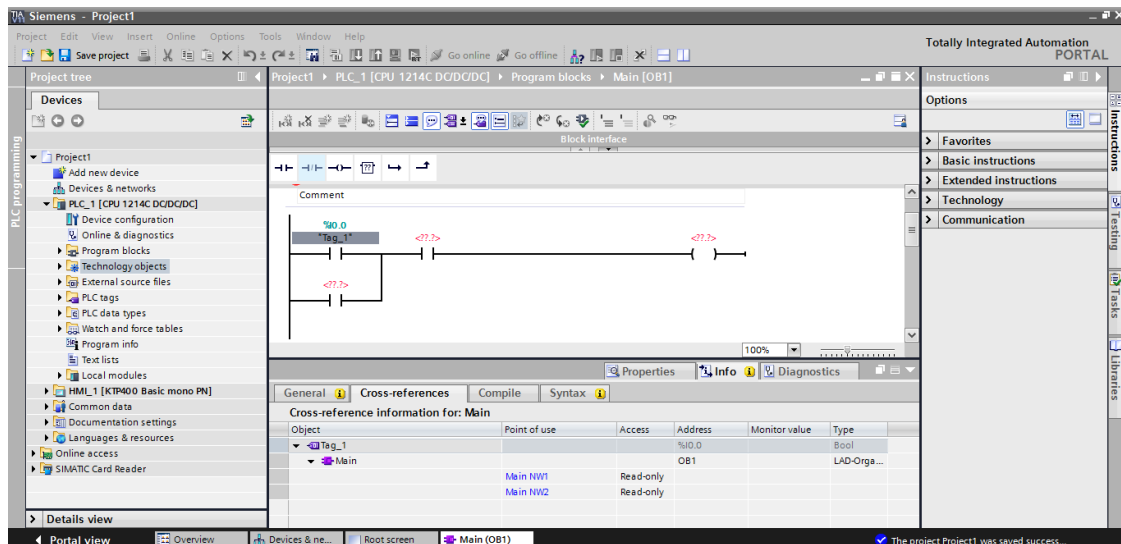
περιλαμβάνει τις εντολές για τη δημιουργία του προγράμματος. Αυτές είναι χωρισμένες σύμφωνα με τις λειτουργίες που εκτελούν. Για τη δημιουργία του προγράμματος ισχύει το drag & drop σέρνοντας την εντολή από το task στο επιθυμητό network. Επίσης περιλαμβάνει τα «favourites», στα οποία τοποθετούμε τις εντολές που είναι πιο συχνά χρησιμοποιήσιμες για εξοικονόμηση χρόνου.

Σε κάθε επαφή ή πηνίο (και γενικά εντολή) αντιστοιχούμε τις PLC tags ορίζουν για το χρήστη τη σωστή λειτουργία του προγράμματος.

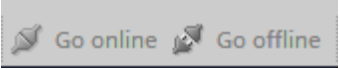


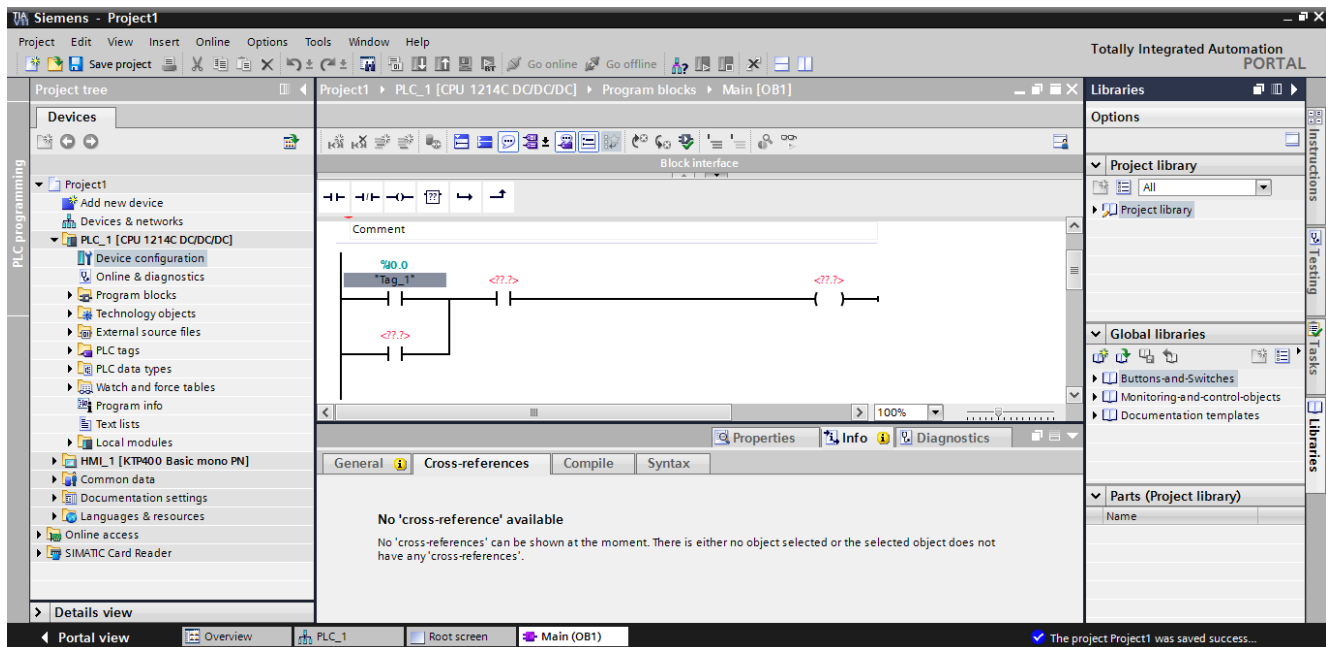
Εικόνα 4.28: Παράδειγμα κώδικα στο OB1

- Μια άλλη δυνατότητα του συγκεκριμένου software είναι το «cross conference», το οποίο μας δείχνει σε ποιο άλλο σημείο του προγράμματος χρησιμοποιείται η συγκεκριμένη μεταβλητή.



Εικόνα 4.29: Μέσω του cross conference βλέπουμε που αλλού χρησιμοποιείται κάποια μεταβλητή

- Στο project tree χαμηλά υπάρχει η δυνατότητα επιλογής του detail window, πιέζοντας «details view». Μέσω αυτού μας παρέχονται όλες οι λεπτομέρειες για την επιλεγμένη συσκευή από το project tree.
- Υπάρχουν οι Online functions μέσω του button  με το οποίο παρέχεται η δυνατότητα να δούμε το status λειτουργίας του προγράμματος σε πραγματικό χρόνο και ταυτόχρονα να έχουμε ένα overview για διαγνώσεις αλλά και τη δυνατότητα γρήγορης σύγκρισης του κάθε block από κατάσταση online σε offline και ανάποδα.
- Μία πολύ σημαντική ενσωματωμένη δυνατότητα στο TIA portal είναι αυτή του «Simulation», με την οποία ο χρήστης βλέπει online την κατάσταση και τις τιμές των μεταβλητών. Με αυτό το τρόπο είναι πολύ πιο εύκολος ο έλεγχος του προγράμματος σε ζωντανές καταστάσεις και φυσικά η διόρθωσή του αν χρειάζεται.
- Άλλο κομμάτι του TIA portal είναι το diagnostic systems, το οποίο δεν απαιτεί κάποια επιπλέον άδεια και είναι ενσωματωμένο με το υπόλοιπο software. Μας παρέχει όλες τις σχετικές πληροφορίες που αφορούν λάθη που παρουσιάζονται στο σύστημα. Αυτές οι πληροφορίες εμπεριέχονται σε πακέτα που περιλαμβάνουν λεπτομέρειες όπως:
 - Το συγκεκριμένο κομμάτι προγραμματισμού (module)
 - Κάποιο κείμενο μηνύματος
 - Κάποιο status μηνύματος
- Το TIA portal παρέχει επίσης βιβλιοθήκες για να σώσουμε ολόκληρα project ή και κομμάτια ενός project που χρησιμοποιούνται συχνά. Υπάρχουν δύο είδη βιβλιοθηκών, οι local libraries και οι global libraries. Με τις global τα αποθηκευμένα στοιχεία μπορούν να ανταλλαχθούν ανάμεσα σε διαφορετικά projects, ενώ οι Local σώζονται στο ίδιο το project και ο χρήστης σώζει πράγματα που θέλει να ξαναχρησιμοποιήσει σε αυτό το project.



Εικόνα 4.30: Στα δεξιά εμφανίζεται το task των libraries (βιβλιοθήκες)

- Επιπλέον δυνατότητα του TIA portal είναι η δυνατότητα εισαγωγής κωδικού για την είσοδο στις λειτουργίες της CPU για την ασφάλεια των εφαρμογών.
- Όπως είδαμε παραπάνω στο ίδιο λογισμικό ενσωματώνεται ο προγραμματισμός των PLC και των HMI. Με αυτή τη δυνατότητα προγραμματίζονται γρήγορα και έξυπνα τα Basic HMI που χρησιμοποιούνται στο project και με πολύ εύκολο τρόπο πραγματοποιείται η σύνδεση των tags της HMI με τα αντίστοιχα tags του PLC. Η Step 7 σου παρέχει βασικές βιβλιοθήκες για την εισαγωγή βασικών σχημάτων, κάποιων standard γραφικών καθώς και διάφορα διαδραστικά στοιχεία σε μια HMI. Με drag & drop βάζουμε το επιλεγμένο στοιχείο πάνω στην επιθυμητή screen και από τα properties ρυθμίζουμε τις λειτουργίες που θα πραγματοποιούνται με αυτό το στοιχείο καθώς και τα tags που επηρεάζονται από αυτό.

Κεφάλαιο 5: Ανάλυση προγράμματος γεμιστικής μηχανής

5.1 Τα μέρη που συντελούν τη γεμιστική μηχανή

Στο κεφάλαιο αυτό αναλύεται η λειτουργία μίας ημιαυτόματης γεμιστικής μηχανής με χρήση ενός PLC S7-1200. Ο βασικός τρόπος χειρισμού της είναι ότι ο χειριστής προεπιλέγει τη ποσότητα που θα συσκευάσει ανά δοχείο με βάση το βάρος, τοποθετεί ένα δοχείο πάνω στη ζυγισζόμενη επιφάνεια, πιάζει το button εκκένωσης και η γεμιστική μηχανή εκκενώνει στο δοχείο την κατάλληλη ποσότητα υγρού. Σταματάει αυτόματα όταν πιάσει το σωστό βάρος-στόχο.

Για τη λειτουργία της γεμιστικής μηχανής χρησιμοποιείται ο παρακάτω εξοπλισμός:

- Μία δυναμοκυψέλη που αντιλαμβάνεται το βάρος του δοχείου κάθε στιγμή.
- Ένας ηλεκτρονικός ενδείκτης βάρους DGTQPB, της εταιρείας DINI ARGEO, ο οποίος είναι υπεύθυνος για την απεικόνιση του βάρους στην οθόνη του. Συγκεκριμένα παραλαμβάνει ηλεκτρικό σήμα από τη δυναμοκυψέλη, το επεξεργάζεται με χρήση μικροεπεξεργαστών και το εμφανίζει στην οθόνη του. Επιπλέον παρέχει την δυνατότητα επικοινωνίας μέσω του πρωτοκόλλου Profibus. Μέσω του Profibus γίνεται η αποστολή του βάρους από τον ενδείκτη στο PLC.
- Ένα PLC S7-1200 της Siemens, μέσω του οποίου υλοποιείται όλος ο έλεγχος και η λειτουργία της γεμιστικής μηχανής.
- Μία HMI KTP 400 της Siemens, η οποία συνδέεται με το PLC και χρησιμοποιείται για καλύτερη επικοινωνία του χρήστη με τη μηχανή καθώς και για την εισαγωγή και εξαγωγή κάποιων στοιχείων από το χρήστη στη μηχανή και ανάποδα.
- Έναν εκτυπωτή EPSON LX-300 για την εκτύπωση κάποιων στατιστικών στοιχείων.
- Μία αντλία μέσω της οποίας γίνεται η μεταφορά του υγρού από τον αποθηκευτικό χώρο στο τοποθετημένο δοχείο πάνω στη πλατφόρμα ζύγισης.
- Έναν ρυθμιστή στροφών (inverter) SINUS, για την ρύθμιση των στροφών της αντλίας με σκοπό να επιτευχθεί η επιθυμητή ταχύτητα πλήρωσης αλλά και ακρίβεια στη ζύγιση.
- Ένα Module CM 1241 RS232, για επικοινωνία του PLC μέσω RS-232 με τον εκτυπωτή LX-300.
- Ένα Module CM 1243-5 PROFIBUS DP MASTER, για επικοινωνία του PLC μέσω Profibus με τον ενδείκτη DGTQPB.

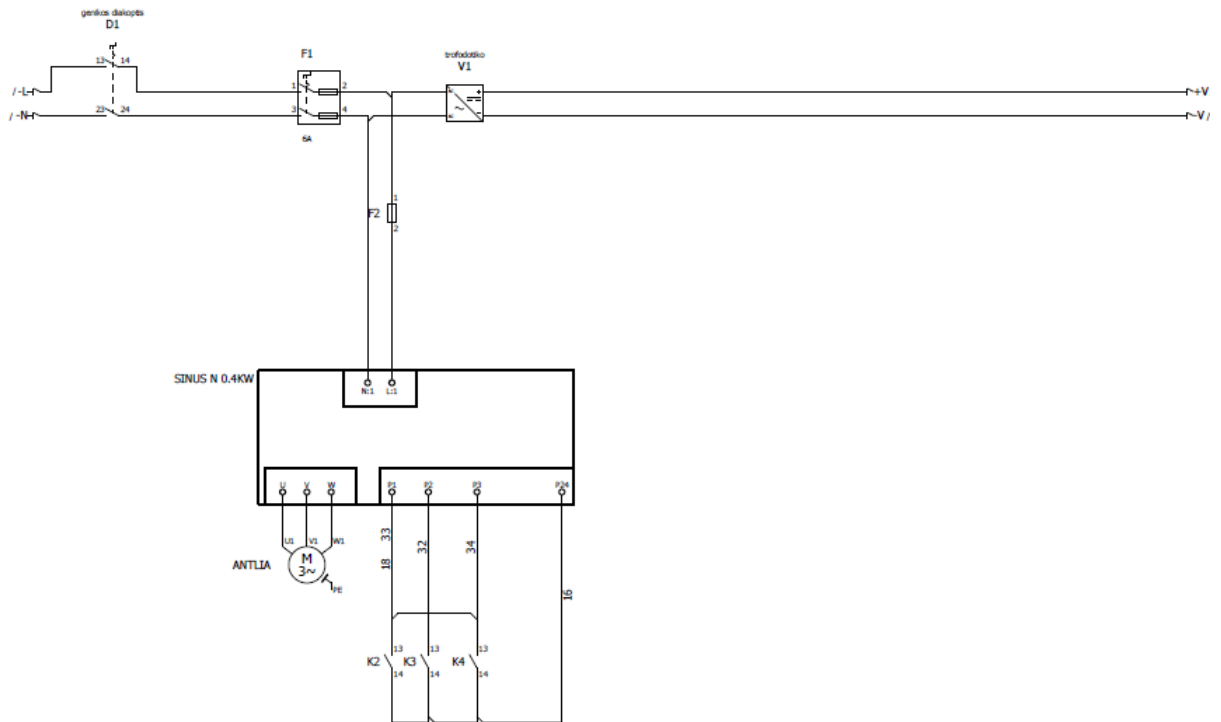
Όλος ο παραπάνω εξοπλισμός βρίσκεται στο εσωτερικό του πίνακα αυτοματισμού της μηχανής. Εξαιρείται ο εκτυπωτής που είναι επιτραπέζιος, η δυναμοκυψέλη που βρίσκεται κάτω από την επιφάνεια ζύγισης και η αντλία.



Εικόνα 5.1 : Το εξωτερικό και το εσωτερικό του πίνακα αυτοματισμού της γεμιστικής μηχανής

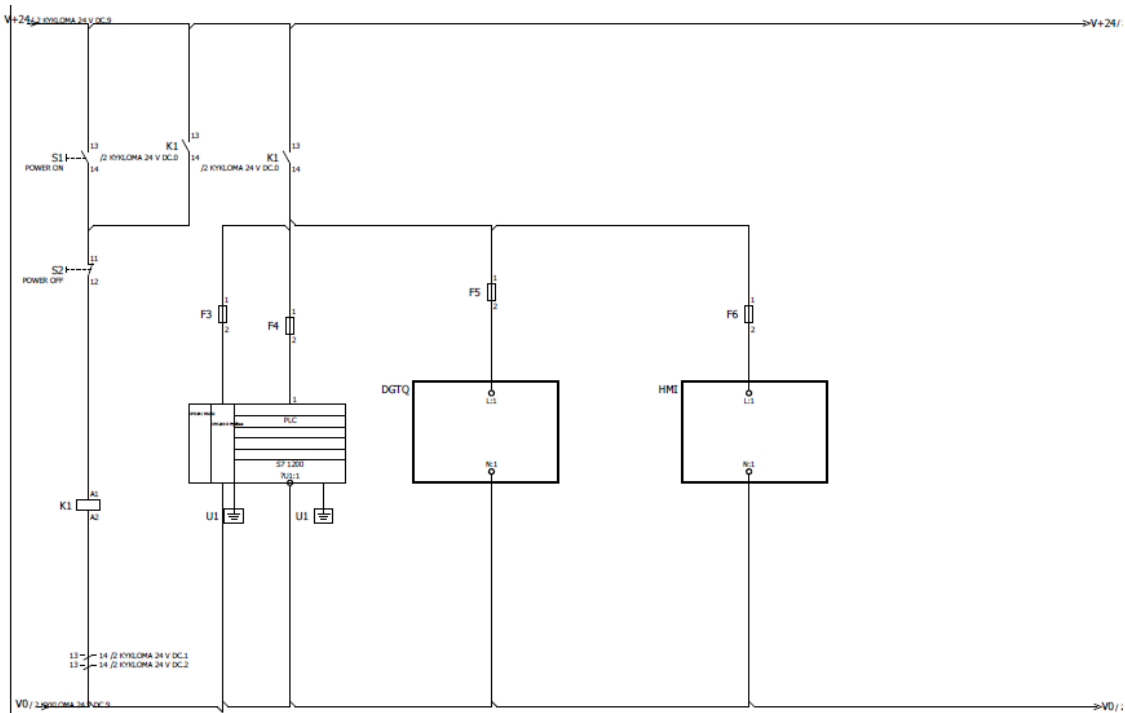
Στη συνέχεια παρουσιάζεται το ηλεκτρολογικό σχέδιο του κυκλώματος που υλοποιήθηκε με χρήση του λογισμικού E-Plan.

Στην εικόνα 5.2 φαίνεται το σχέδιο όσον αφορά την εναλλασσόμενη τάση του κυκλώματος και το τροφοδοτικό που την μετατρέπει σε συνεχή.



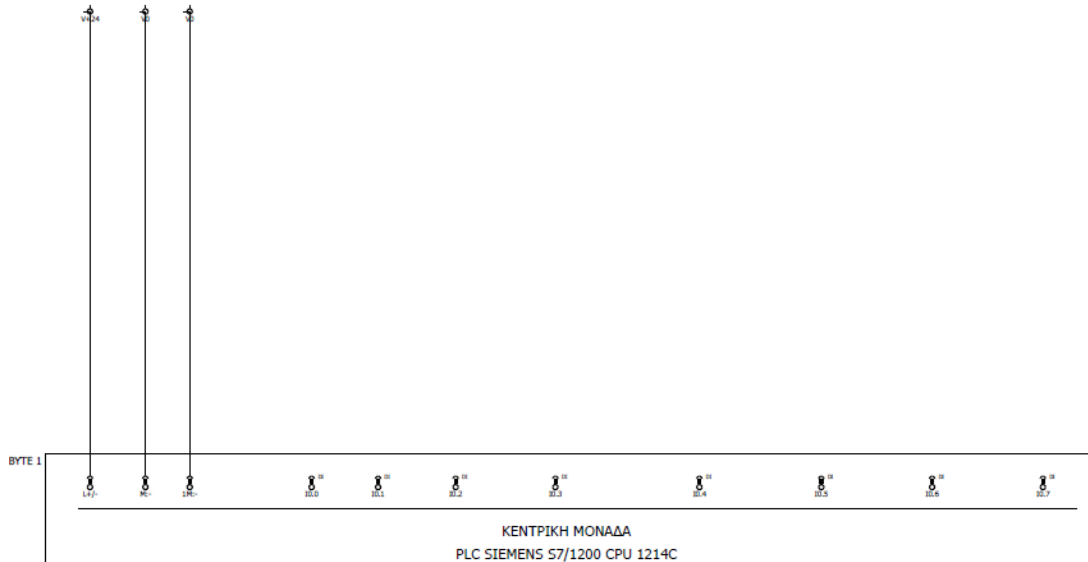
Εικόνα 5.2 : Κύκλωμα εναλλασσόμενης τάσης

Στην εικόνα 5.3 φαίνεται το σχέδιο της συνεχούς τάσης μετά το τροφοδοτικό. Φαίνεται η αυτοσυγκράτηση που πραγματοποιείται με τα Buttons S1 & S2 και οι τροφοδοσίες των PLC, DGTQPB και HMI.



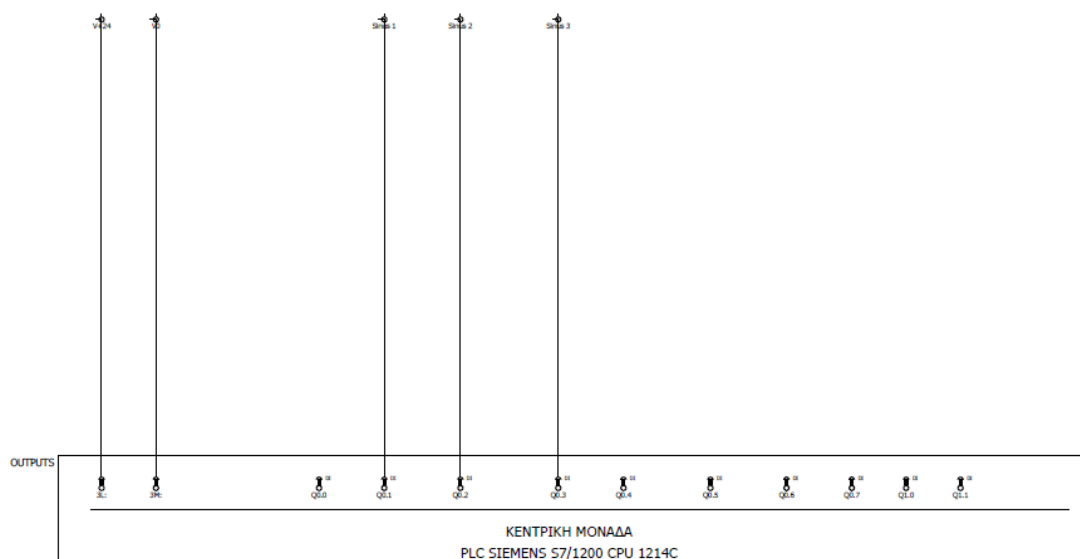
Εικόνα 5.3: Κύκλωμα συνεχούς τάσης

Στην εικόνα 5.4 εμφανίζονται οι εισόδους του PLC. Βέβαια το μόνο που φαίνεται είναι η τροφοδοσία των εισόδων αφού στο πρόγραμμα δεν χρησιμοποιούμε καμία είσοδο του PLC, λόγω του ότι όλες οι επιλογές του χρήστη γίνονται από την HMI.



Εικόνα 5.4: Είσοδοι του PLC

Στην εικόνα 5.5 φαίνονται οι εξόδοι του PLC καθώς και η τροφοδοσία αυτών. Τα σήματα των εξόδων Q0.1, Q0.2 και Q0.3 στέλνονται προς το inverter για την κατάλληλη ρύθμιση των στροφών της αντλίας.



Εικόνα 5.5: Εξοδοι του PLC

Οι είσοδοι και οι έξοδοι απεικονίστηκαν σε διαφορετικά σχήματα απλά για λόγους καλύτερης ευκρίνειας. Θα μπορούσαν να απεικονίζονται όλα μαζί στην εικόνα 5.3 του σχήματος συνεχούς τάσης πάνω στο PLC.

Στη συνέχεια η συνδεσμολογία είναι η εξής. Η δυναμοκυψέλη συνδέεται με τον ενδείκτη βάρους και του αποστέλλει το βάρος. Ο ενδείκτης συνδέεται με το PLC, στο οποίο στέλνει επίσης το βάρος μέσω Profibus. Το PLC συνδέεται με την HMI, από όπου γίνεται αμφίδρομη ανταλλαγή δεδομένων, με τον εκτυπωτή στον οποίο στέλνει τι να εκτυπώσει και με το inverter στο οποίο στέλνει με ποια συχνότητα να λειτουργεί κάθε φορά, ώστε το inverter να κινήσει την αντλία με τις κατάλληλες στροφές.

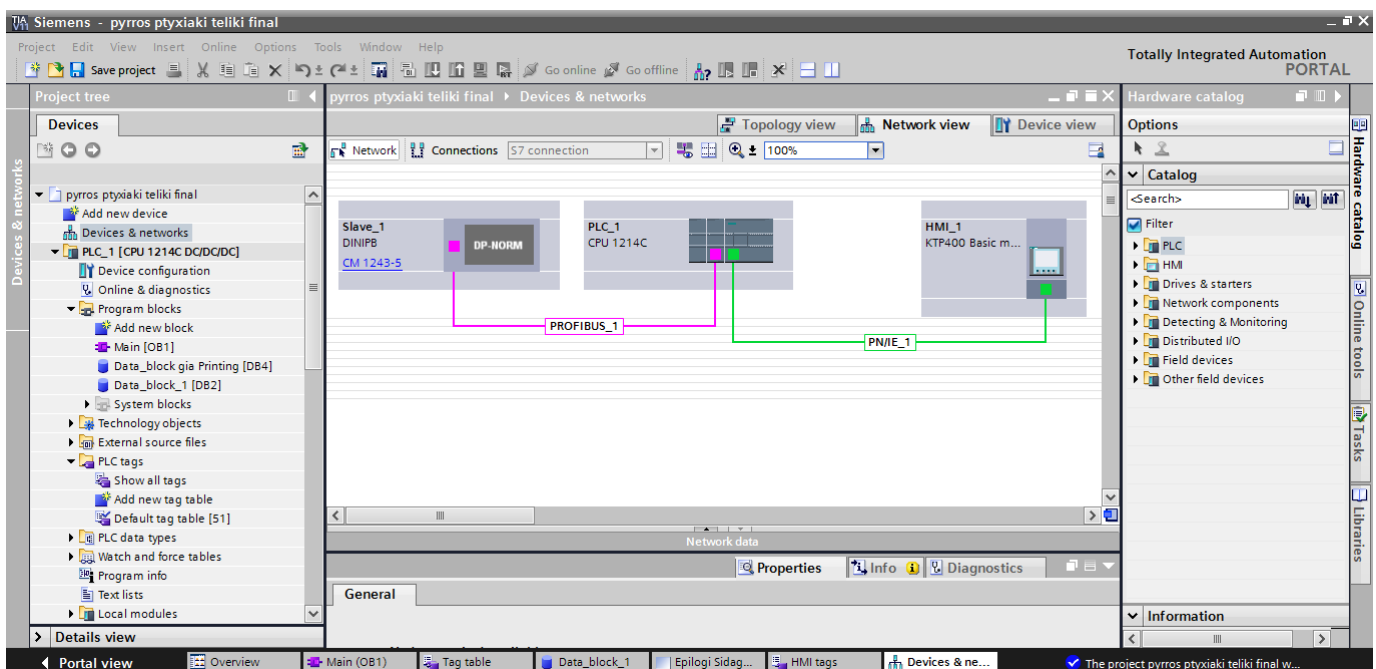
5.2 Δημιουργία προγράμματος με το TIA PORTAL

Το πρόγραμμα που υλοποιεί τον έλεγχο της μηχανής, δημιουργήθηκε με το λογισμικό TIA Portal. Η ανάλυση του προγράμματος περιγράφεται παρακάτω.

Αρχικά γίνεται η επιλογή του hardware configuration που θα χρησιμοποιηθεί για τη λειτουργία της μηχανής, στο επίπεδο του TIA portal. Το hardware configuration αποτελείται από τα εξής modules:

- PLC S7-1200, με κωδικό CPU 1214C.
- HMI KTP 400 basic mono PN.
- Module επέκτασης για την επικοινωνία με RS232 με τον εκτυπωτή με κωδικό CM1241 RS232.
- Module επέκτασης για την επικοινωνία του PLC με τον ενδείκτη μέσω Profibus με κωδικό CM 1243-5 PROFIBUS DP MASTER.
- Τον ενδείκτη DGTQPB, του οποίου έχει γίνει install το κατάλληλο gsd file για να εμφανίζεται στο κατάλογο του hardware configuration του TIA portal.

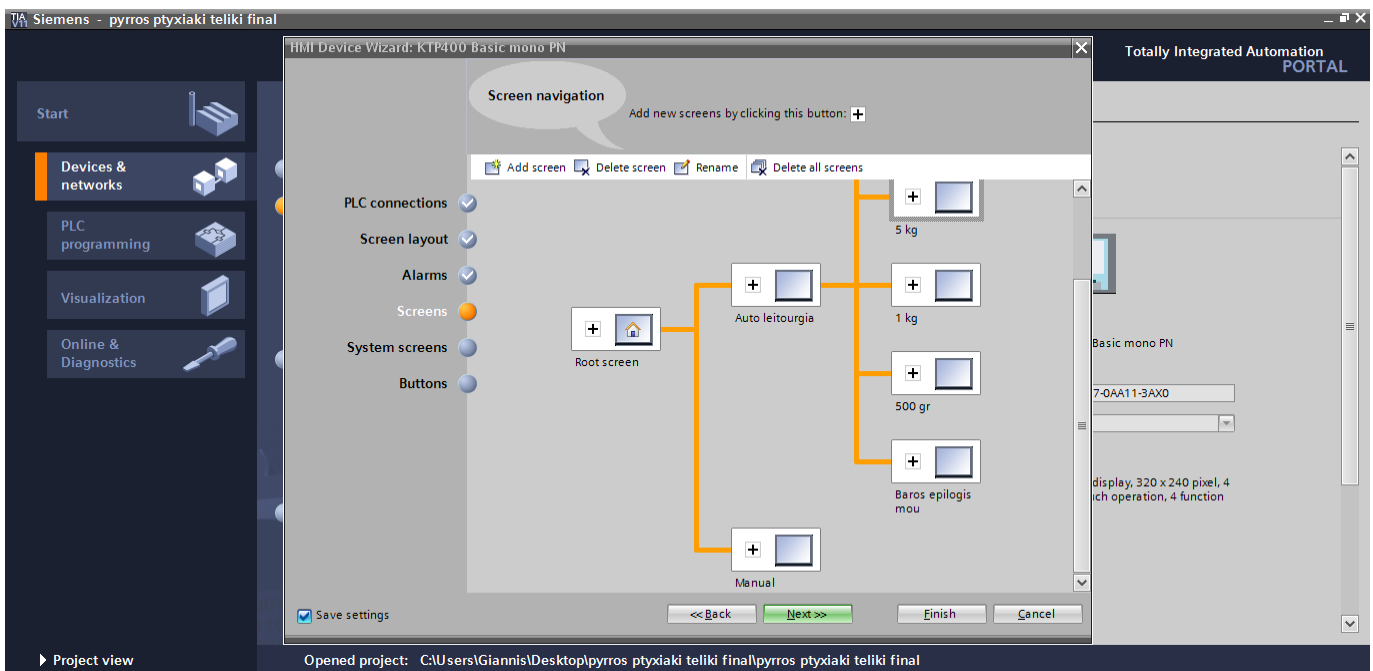
Όλα τα παραπάνω φαίνονται στο «network view».



Εικόνα 5.6: Network view του προγράμματος ελέγχου της γεμιστικής μηχανής

Στο δεξί μέρος του Network view (εικόνα 5.6) φαίνεται η HMI, στο κέντρο φαίνεται το PLC S7-1200 μαζί με τα δύο modules επικοινωνίας και αριστερά ο ενδείκτης DGTQPB με το όνομα Slave_1. Επίσης με πράσινη γραμμή απεικονίζεται η Profinet σύνδεση μεταξύ PLC και HMI και με ροζ γραμμή φαίνεται η Profibus σύνδεση μεταξύ PLC (συγκεκριμένα module CM 1243-5) και ενδείκτη DGTQPB. Ο εκτυπωτής δεν εμφανίζεται στο hardware configuration του TIA portal.

Η εισαγωγή της οθόνης KTP 400 basic έγινε μέσω του «Portal view» ακολουθώντας όλη την διαδικασία του «HMI Device Wizard» που περιγράφηκε στο κεφ 4.3. Η δομή των οθονών (screens) φαίνεται στην εικόνα 5.7.



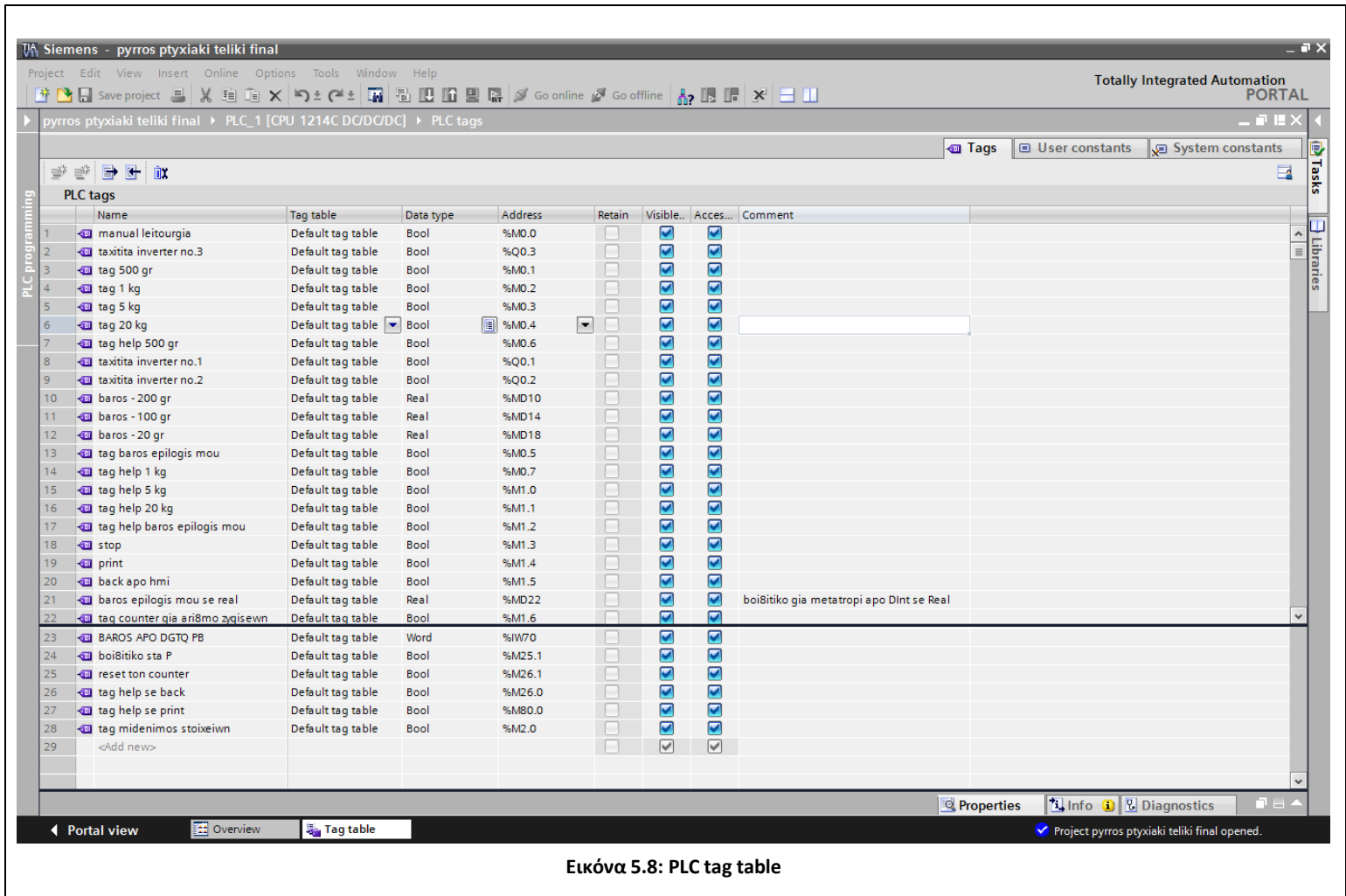
Εικόνα 5.7: Δομή της HMI

Φαίνεται ότι από την αρχική οθόνη, δηλαδή την «Root screen» που εμφανίζεται μόλις ανοίξει η HMI, ο χρήστης έχει το δικαίωμα μετάβασης σε άλλες δύο οθόνες, τις Manual και Auto leitourgia. Στη συνέχεια από την Auto leitourgia έχει το δικαίωμα μετάβασης σε άλλες 5 οθόνες (φαίνονται οι 4 στην εικόνα 5.7).

Στην αρχική οθόνη, ο χρήστης επιλέγει αν θέλει να γεμίσει manual (χειροκίνητα) εισάγοντας σε ένα δοχείο όση ποσότητα επιθυμεί ή να επιλέξει κάποια συγκεκριμένη ποσότητα γέμισης. Αν επιλέξει «manual» μεταφέρεται στην επόμενη οθόνη όπου του δίνεται η δυνατότητα να βάλει όσο επιθυμεί πατώντας συνεχόμενα ένα button. Αν επιλέξει «auto leitourgia», δηλαδή μία συγκεκριμένη ποσότητα, μεταφέρεται σε μία οθόνη στην οποία επιλέγει την ακριβή ποσότητα γέμισης ανά δοχείο. Έχει να επιλέξει ανάμεσα σε 4 τυποποιημένες συνταγές ή τη δυνατότητα εισαγωγής ως βάρος-στόχο κάποιον διαφορετικό. Μόλις επιλέξει συνταγή, μεταφέρεται στην αντίστοιχη οθόνη που δίνεται η εντολή γέμισης των δοχείων.

5.3 Δημιουργία των tag tables σε PLC και HMI

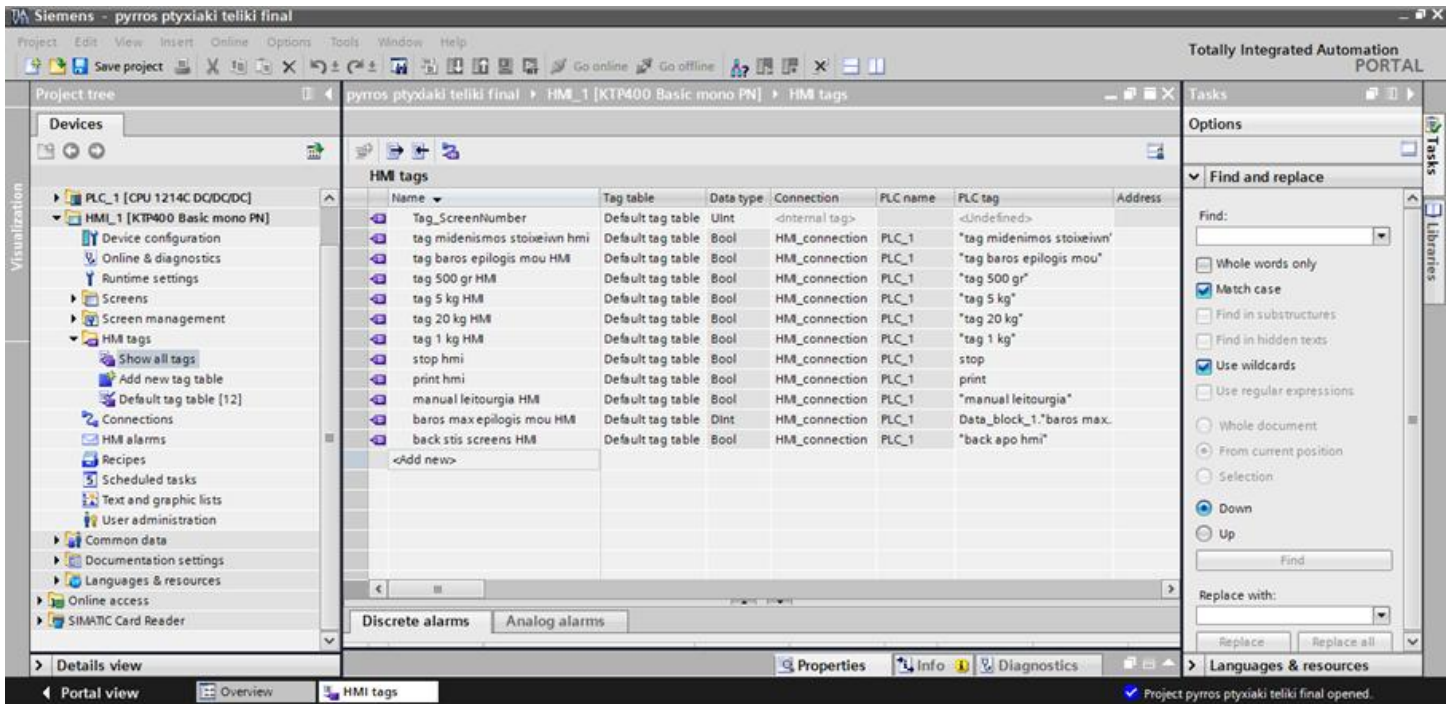
Μετά την επιλογή του hardware configuration, δημιουργούμε το PLC tag table με όλες τις μεταβλητές που θα χρησιμοποιήσουμε στο πρόγραμμα (εικόνα 5.8). Γίνεται και η ανάθεση ονομάτων για καλύτερη και ευκολότερη χρήση αυτών.



Εικόνα 5.8: PLC tag table

Στο PLC tag table υπάρχουν όλες οι εισοδοι, οι έξοδοι και οι μνήμες που χρησιμοποιούνται. Παράλληλα υπάρχουν και κάποιες βοηθητικές μεταβλητές τύπου word ή double word.

Στη συνέχεια υλοποιείται η δημιουργία του HMI tag table, με τις μεταβλητές που χρησιμοποιούνται για τη λειτουργία της HMI καθώς και η δήλωση με ποιες μεταβλητές του PLC tag table συνδέονται (εικόνα 5.9).



Εικόνα 5.9: HMI tag table

Στο δεξί μέρος της εικόνας 5.9 υπάρχουν τρεις στήλες που έχουν σημασία. Η μία δείχνει το «data type» της κάθε μεταβλητής. Στη συγκεκριμένη περίπτωση όλες οι μεταβλητές μας είναι Bool. Η άλλη δείχνει αν υπάρχει σύνδεση της συγκεκριμένης μεταβλητής με κάποια του PLC. Το PLC αναγνωρίζεται ως PLC_1 στο συγκεκριμένο δίκτυο. Η τρίτη δείχνει με ποια μεταβλητή συγκεκριμένα υπάρχει η σύνδεση. Αυτό σημαίνει ότι μόλις μια εντολή ενεργοποιήσει ή απενεργοποιήσει κάποια συγκεκριμένη μεταβλητή της HMI, τότε ακριβώς το ίδιο αποτέλεσμα συμβαίνει και στην αντίστοιχη μεταβλητή με την οποία είναι συνδεδεμένη από το tag table του PLC. Με αυτή την σύνδεση των μεταβλητών γίνεται και η επικοινωνία του PLC με την HMI. Συνδέονται μεταξύ τους με καλώδιο Ethernet και μεταξύ τους ανταλλάσσουν τιμές για τις μεταβλητές που συνδέονται.

5.4 Τα Data Blocks του προγράμματος

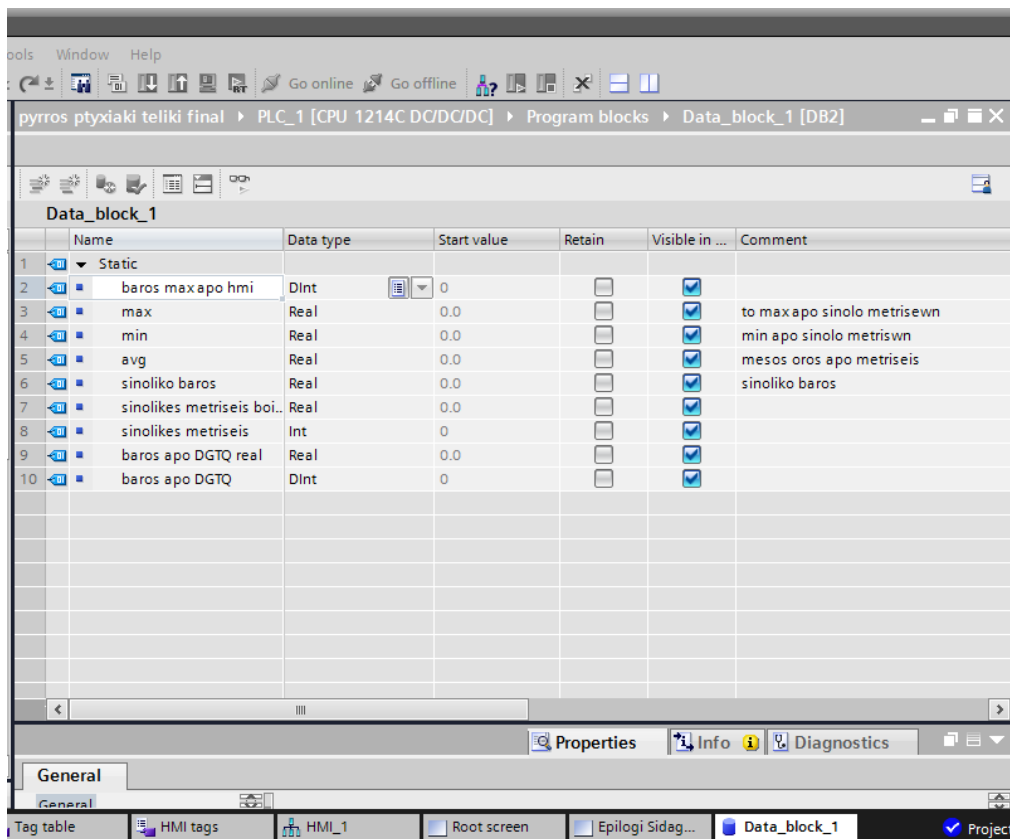
Αφού γίνει η δήλωση και η ανάθεση των μεταβλητών του PLC tag table και του HMI tag table, δημιουργούμε δύο data blocks για την αποθήκευση κάποιων δεδομένων που είναι πολύ χρήσιμα για το πρόγραμμα.

Στο πρώτο data-block, με όνομα «Data_block_1», γίνεται η αποθήκευση όλων των δεδομένων που χρησιμοποιούνται για τον υπολογισμό κάποιων στατιστικών στοιχείων. Αυτά είναι τα εξής:

- Το βάρος που στέλνει μέσω Profibus ο ενδείκτης DGTQPB.
- Το βάρος σε μορφή «real» (επειδή ο ενδείκτης το στέλνει σε μορφή «Dint», ενώ για τον υπολογισμό αριθμητικών πράξεων το χρειαζόμαστε σε real)
- Τη μέγιστη ποσότητα γέμισης.
- Τη μικρότερη ποσότητα γέμισης.
- Το μέσο όρο όλων των γεμίσεων.

- Τη συνολική ποσότητα που συσκευάστηκε.
- Τον αριθμό των γεμίσεων που πραγματοποιήθηκαν.
- Το σύνολο των γεμίσεων, αποθηκευμένο σε τύπο δεδομένων real, κατάλληλος για την πράξη της διαίρεσης που υπολογίζει το μέσο όρο των γεμίσεων.
- Το βάρος-στόχο που έχει επιλέξει ο χρήστης να συσκευάσει στην περίπτωση που δεν είναι κάποιο από τα 4 τυποποιημένα.

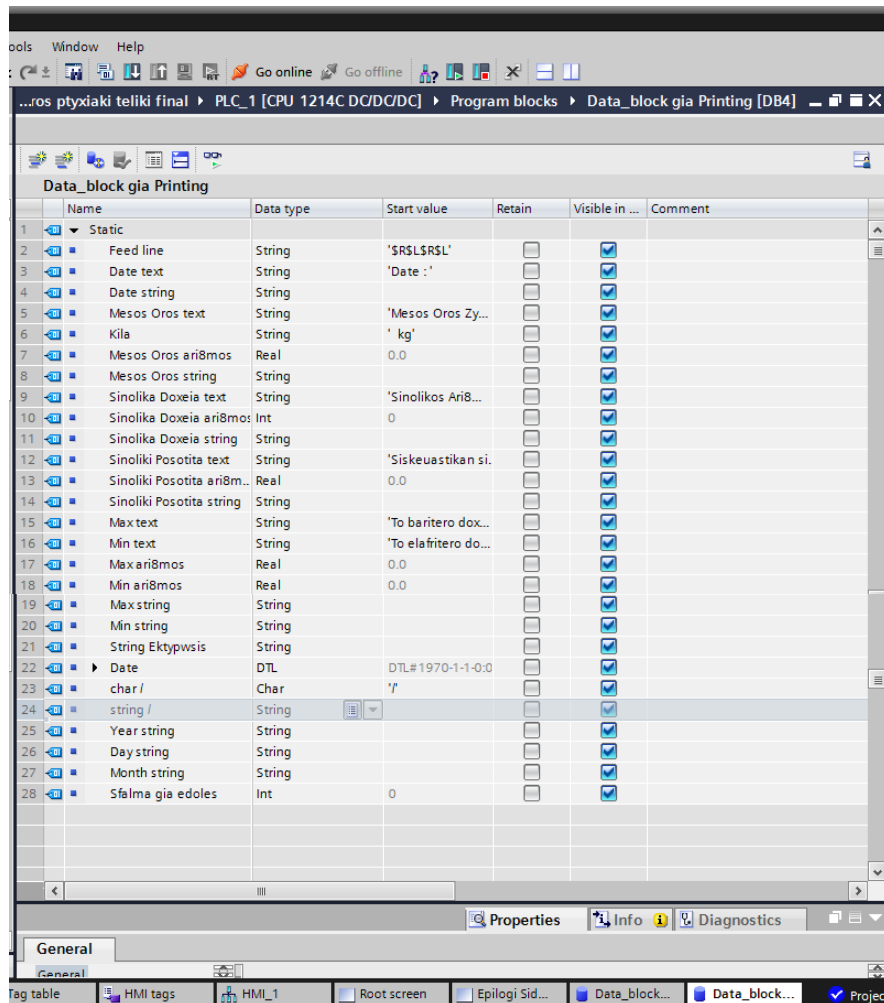
Οι μεταβλητές του Datablock_1 φαίνονται και στην εικόνα 5.10.



Εικόνα 5.10: Data_block_1

Στο δεύτερο data-block, το οποίο έχει όνομα «Data_block gia printing», αποθηκεύονται όλα τα δεδομένα που χρησιμοποιούνται στην εκτύπωση των στατιστικών στοιχείων. Αποθηκεύονται τα strings που παραμένουν σταθερά σε όλες τις εκτυπώσεις, αλλά και τα στατιστικά δεδομένα που κάθε φορά αλλάζουν, ανάλογα με τις γεμίσεις που έχει πραγματοποιήσει ο χρήστης.

Στην εικόνα 5.11 φαίνονται όλες οι μεταβλητές που χρησιμοποιούνται για την δημιουργία του τελικού string εκτύπωσης. Θα φανεί στη συνέχεια της ανάλυσης του προγράμματος πως από όλες αυτές τις μεταβλητές, αυτή που τελικά θα εκτυπωθεί είναι το «String Ektyprwsis» και σχεδόν όλες οι άλλες χρησιμοποιούνται βοηθητικά για τη δημιουργία αυτού.



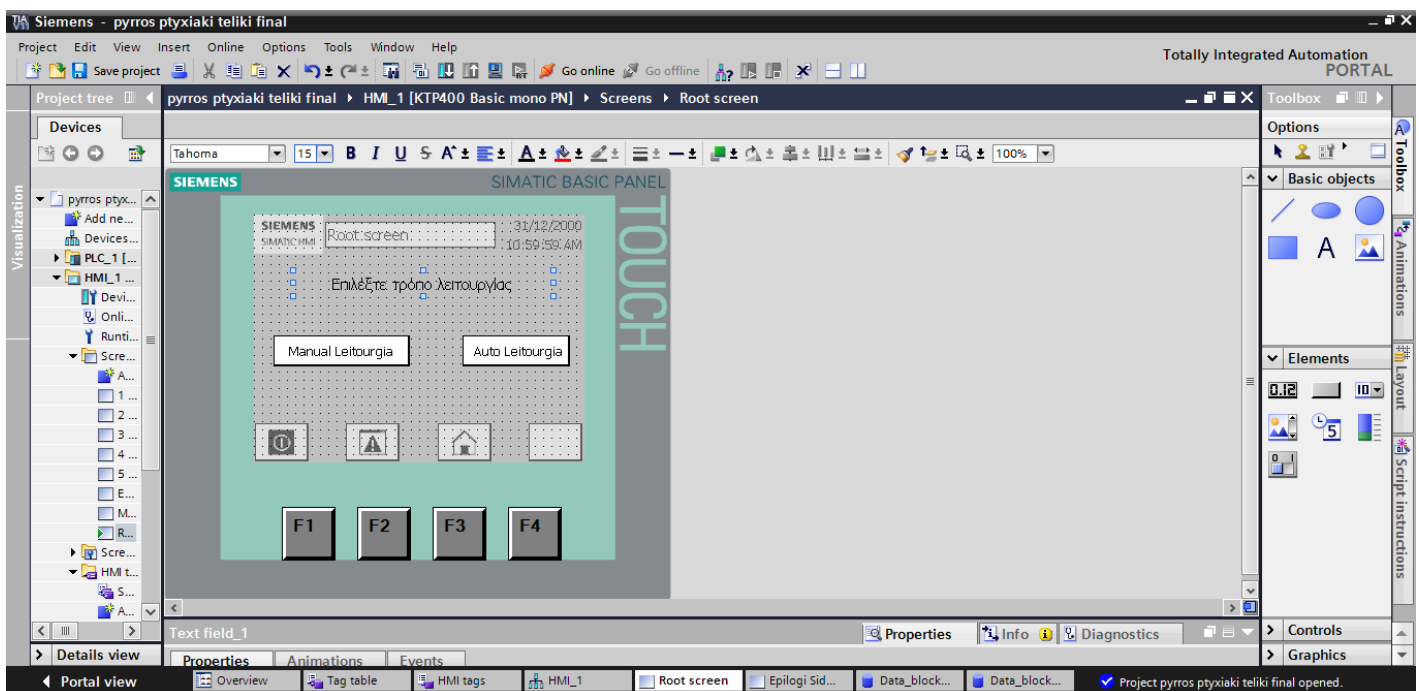
Εικόνα 5.11: Data_block gia printing

5.5 Διαμόρφωση των Screens της HMI

Στη συνέχεια γίνεται η διαμόρφωση και η παραμετροποίηση της κάθε screen (οθόνης) ξεχωριστά, που εμφανίζεται στο χρήστη μέσω της HMI.

- 1^η οθόνη «Root screen»

Σε αυτή την οθόνη (εικόνα 5.12) ο χρήστης επιλέγει αν θέλει να γεμίσει δοχεία με αυτόματο ή manual τρόπο.



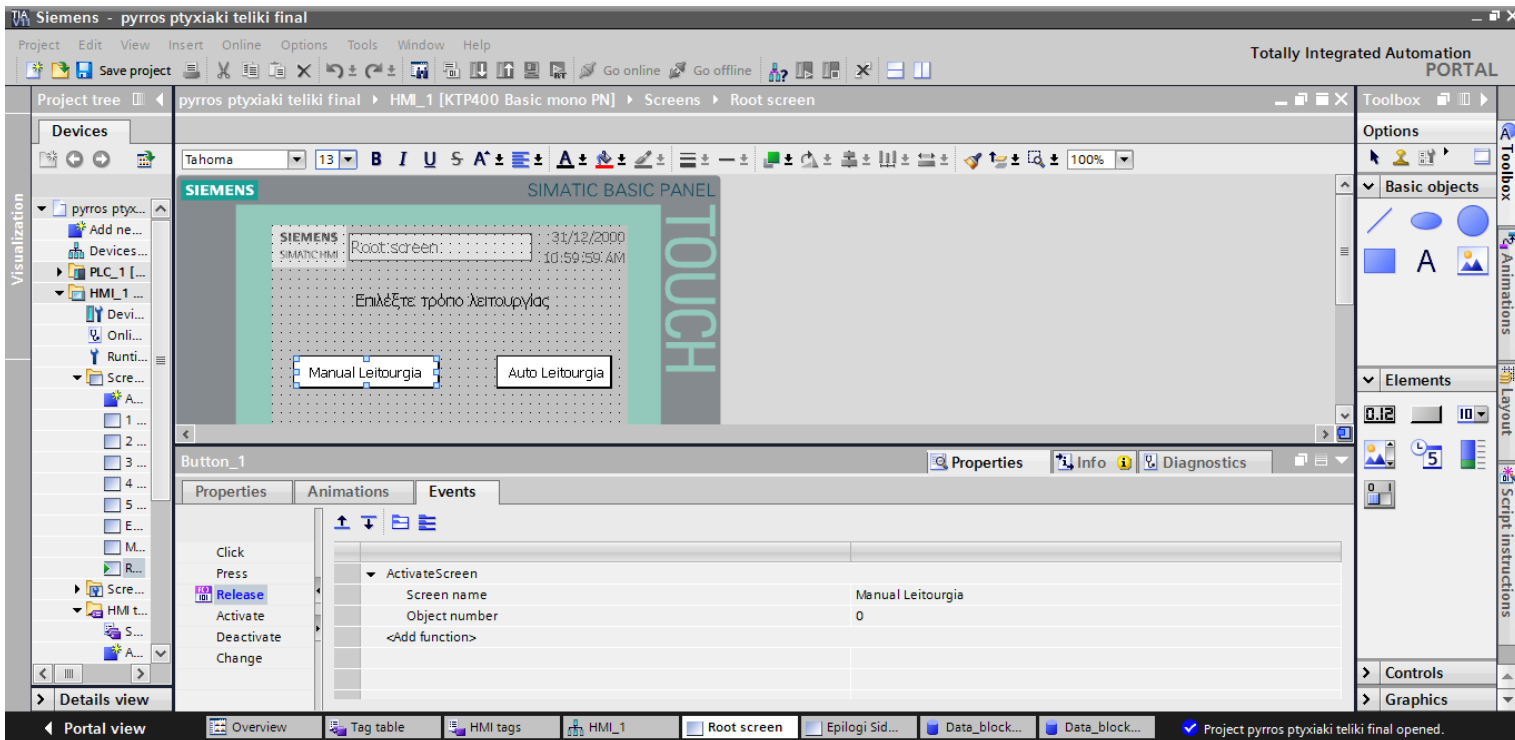
Εικόνα 5.12: Screen 1

Στην οθόνη υπάρχουν τρία πράγματα, ένα text field και δύο buttons.

- Το text field γράφει «Επιλέξτε τρόπο λειτουργίας». Χρησιμοποιείται απλά για να απεικονίσει κάτι στατικό στο χρήστη, έχοντας στόχο την καλύτερη κατανόηση της λειτουργίας αυτής της οθόνης. Εισάγεται από το task των «basic objects» που βρίσκεται στα δεξιά της οθόνης του TIA portal. Με drag & drop τοποθετείται στην επιθυμητή θέση και απλά στο κείμενο πληκτρολογείται το επιθυμητό κείμενο.
- Τα 2 buttons μόλις τα πιάσει ο χρήστης τον μεταβιβάζουν στην αντίστοιχη οθόνη. Αυτά υπάρχουν δύο τρόποι για να δημιουργηθούν. Ο ένας είναι να τοποθετηθεί ένα button από το task των «elements» που βρίσκεται επίσης στο δεξί μέρος του TIA portal. Τοποθετείται στην επιθυμητή θέση με drag & drop και πληκτρολογείται πάνω του, ότι επιθυμείται να εμφανίζεται (στη προκειμένη περίπτωση «Manual Leitourgia» και «Auto leitourgia»). Στη

συνέχεια επιλέγοντας το κατάλληλο button με αριστερό κλικ, επιλέγουμε το μενού properties→events. Εκεί επιλέγεται σαν εντολή «activate screen» και στο screen name μπαίνει το όνομα της οθόνης που επιθυμούμε να γίνεται η μεταβίβαση.

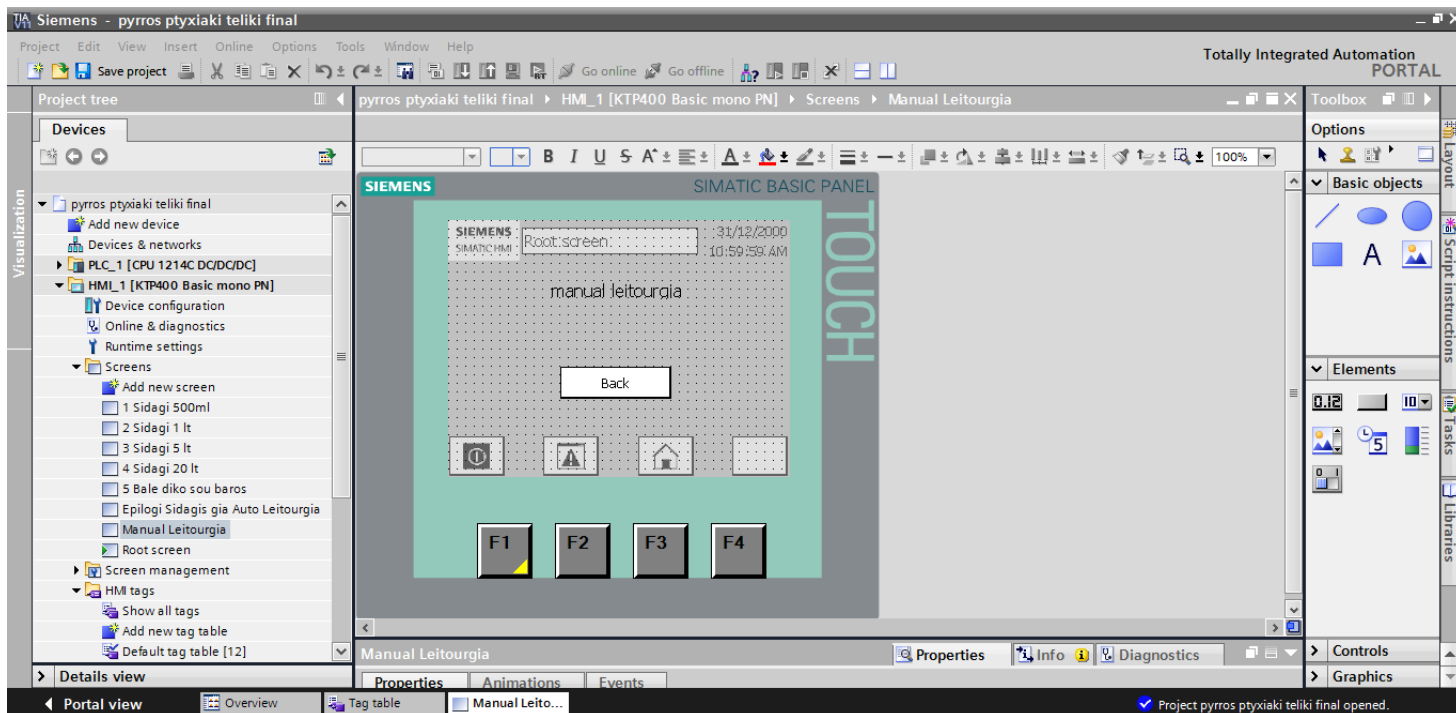
Ο δεύτερος τρόπος είναι να δημιουργηθούν από μόνα τους από το σύστημα. Αυτό βέβαια συμβαίνει μόνο αν η εισαγωγή της HMI έχει γίνει μέσω του «HMI Device Wizard». Επομένως, όταν στο 4^ο βήμα γίνει η δομή των screens, δημιουργούνται αυτόματα buttons που σαν εντολές εκτελούν την μεταφορά από τη μία οθόνη στην άλλη. Το μόνο που πρέπει να αλλαχτεί είναι η θέση του button μέσα στην οθόνη και το τι θα λέει το button ώστε να γίνεται κατανοητό από το χρήστη. Με αυτό τον τρόπο έχουν δημιουργηθεί τα δύο buttons «Manual leitourgia» και «Auto leitourgia» στη συγκεκριμένη περίπτωση, που μόλις πατηθούν πραγματοποιείται η μεταβίβαση στη αντίστοιχη οθόνη. Στην εικόνα 5.13 φαίνονται οι αντίστοιχες επιλογές για το button «Manual leitourgia».



Εικόνα 5.13: Ρυθμίσεις λειτουργίας του button «Manual leitourgia»

2^η οθόνη «Manual Leitourgia»

Σε αυτή την οθόνη ο χρήστης έχει την δυνατότητα να ρίξει σε κάποιο δοχείο όση ποσότητα υγρού επιθυμεί ανεξαρτήτως βάρους. Απλά όσο πιέζει το button, τόση ώρα ρίχνει η γεμιστική μηχανή υγρό στο δοχείο. Αυτό μπορεί να συμβεί είτε άμα θέλει να συμπληρώσει λίγη ποσότητα σε κάποιο δοχείο, είτε σε περίπτωση που θέλει να γεμίσει πολύ μικρό αριθμό δοχείων χωρίς να ενδιαφέρεται για την ακρίβεια του βάρους.



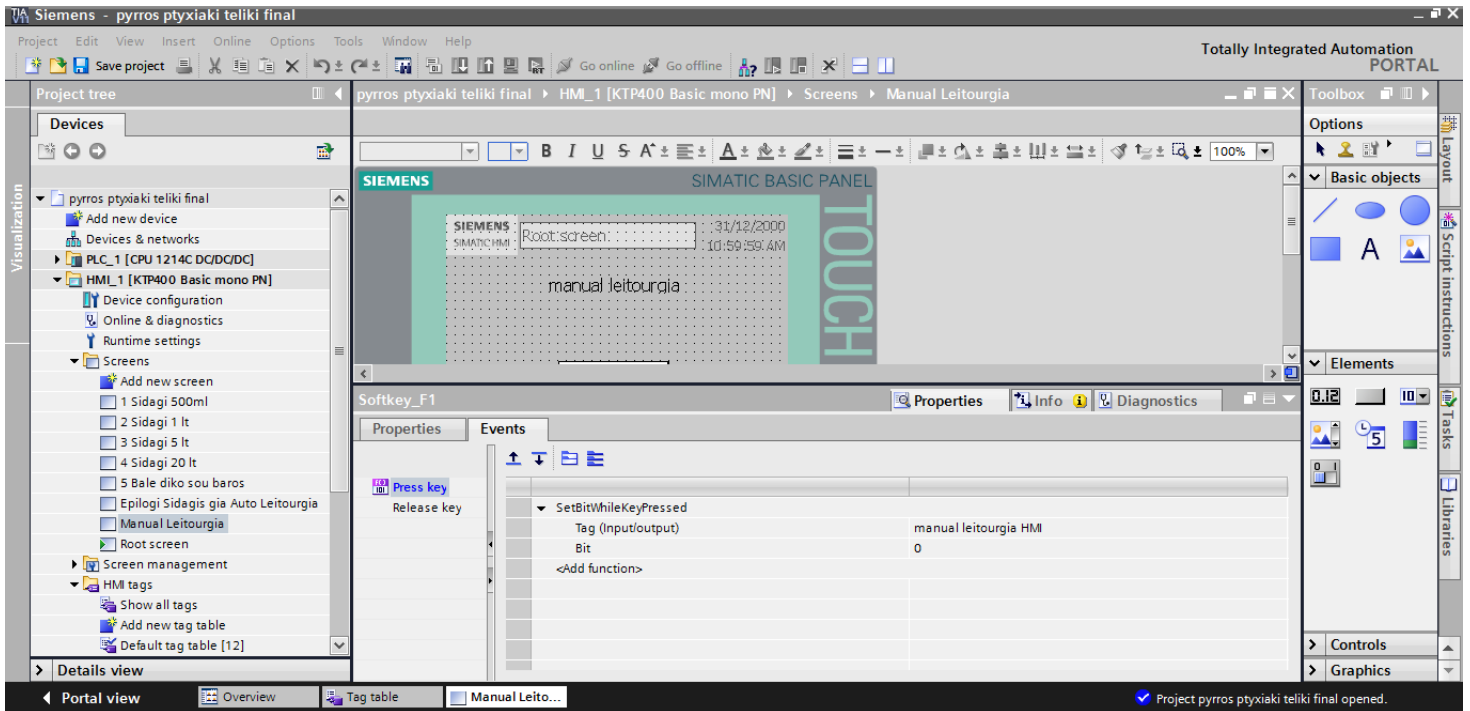
Εικόνα 5.14: Screen 2

Σε αυτή την οθόνη διακρίνουμε επίσης ένα text field που γράφει «manual leitourgia» (εικόνα 5.14) και χρησιμοποιείται απλά για να ενημερώνει το χρήστη σε ποια οθόνη βρίσκεται. Η τοποθέτησή του γίνεται με τον ίδιο ακριβώς τρόπο που περιγράφηκε προηγουμένως.

Υπάρχει ένα button «Back» το οποίο μόλις πατηθεί, πραγματοποιείται η μεταβίβαση στη προηγούμενη οθόνη, δηλαδή την root screen. Αυτό το button επίσης, είτε έχει δημιουργηθεί από μόνο του αν η δομή των οθονών έχει γίνει από το «HMI Device Wizard», είτε μπορούμε να το δημιουργήσουμε με την διαδικασία που περιγράφηκε. Η διαφορά με προηγουμένως είναι ότι στο activate screen σαν επιλογή οθόνης έχει την root screen και σαν text επιλέχθηκε να γράφει «Back».

Τέλος για την πλήρωση των δοχείων με τρόπο manual δε δημιουργήθηκε button από το task των «elements», αλλά χρησιμοποιήθηκε κάποιο από τα κουμπιά που έχει η HMI ενσωματωμένα πάνω της. Αυτή η επιλογή έγινε για μεγαλύτερη διάρκεια ζωής της HMI. Επειδή στη manual λειτουργία ο χρήστης κρατάει πατημένο το κουμπί όση ώρα επιθυμεί να αδειάζει η μηχανή υγρό, αν δημιουργούσαμε κάποιο button πάνω στην touch screen, θα χαλούσε πολύ πιο γρήγορα λόγω συνεχόμενης πίεσης σε αυτή. Έτσι επιλέχθηκε το button F1 για αυτή τη λειτουργία (φαίνεται στην εικόνα 5.14 που είναι κιτρινωμένο στο κάτω δεξί του μέρος). Στη συνέχεια στα events του F1 επιλέξαμε σαν εντολή «SetBitWhileKeyPressed» και την συνδέσαμε με την tag «manual leitourgia HMI» (εικόνα 5.15). Με αυτό το τρόπο όσο πιέζεται το συγκεκριμένο button κάνει λογικό 1 την παραπάνω tag, η οποία συνδέεται από το HMI tag table με την tag «manual leitourgia» του PLC tag table. Μόλις ο χειριστής αφήσει το button F1, δηλαδή release, τότε γίνεται reset η ίδια μεταβλητή «manual leitourgia HMI», και αντίστοιχα η μεταβλητή του PLC «manual leitourgia». Θα φανεί στη συνέχεια που θα γίνει ανάλυση του προγράμματος, τι συμβαίνει όταν στέλνει η οθόνη

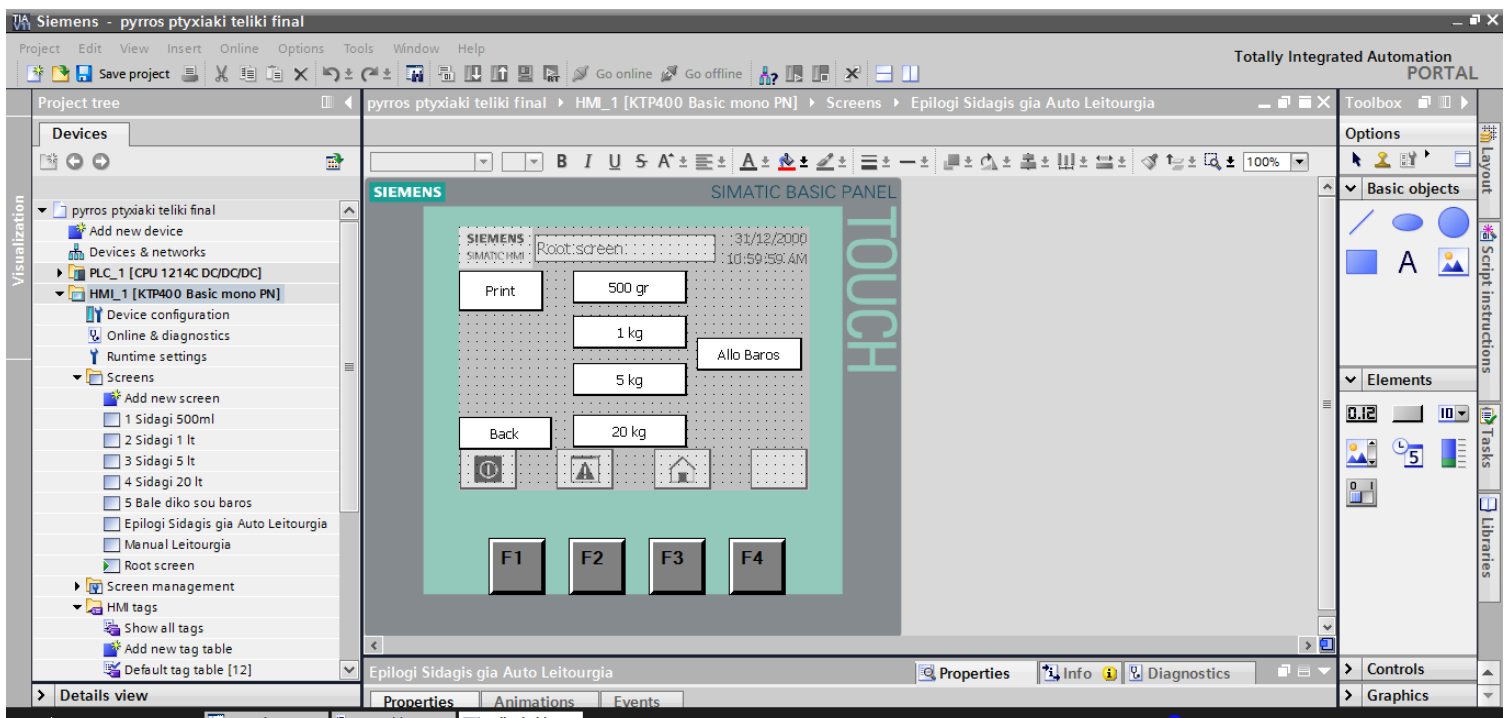
συνεχόμενα λογικό 1 όσο πιέζεται αυτό το button και αντίστοιχα 0 όταν δεν είναι πατημένο.



Εικόνα 5.15: Ρυθμίσεις του button F1

- 3^η οθόνη «Επιλογή συνταγής για αυτόματη λειτουργία»

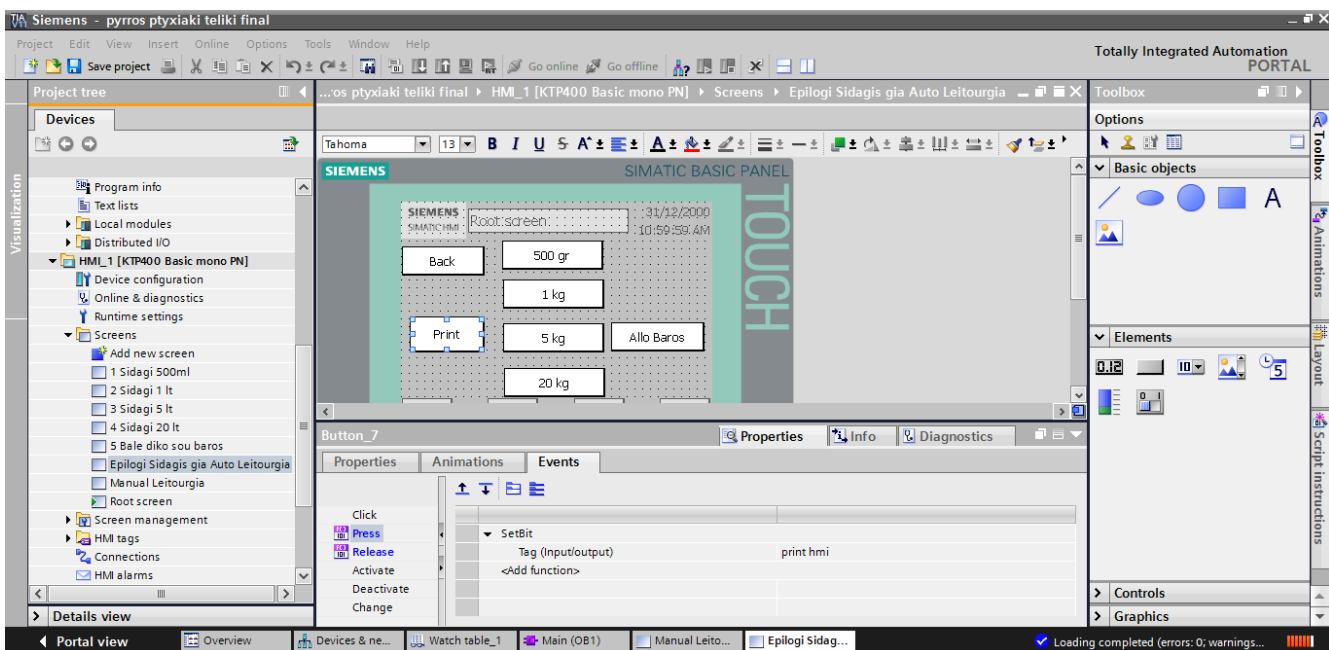
Σε αυτή την οθόνη ο χρήστης επιλέγει την ποσότητα υγρού που θέλει να συσκευάσει. Έχει την δυνατότητα επιλογής κάποιας τυποποιημένης ποσότητας ή να διαλέξει μόνος του όποια ποσότητα επιθυμεί. Η επιλογή των τυποποιημένων ποσοτήτων έχει γίνει με βάση το ότι είναι οι πιο συνηθισμένες ποσότητες της αγοράς. Επίσης από αυτή την οθόνη μπορεί ο χρήστης να εκτυπώσει κάποια στατιστικά στοιχεία από τις γεμίσεις που έχει πραγματοποιήσει.



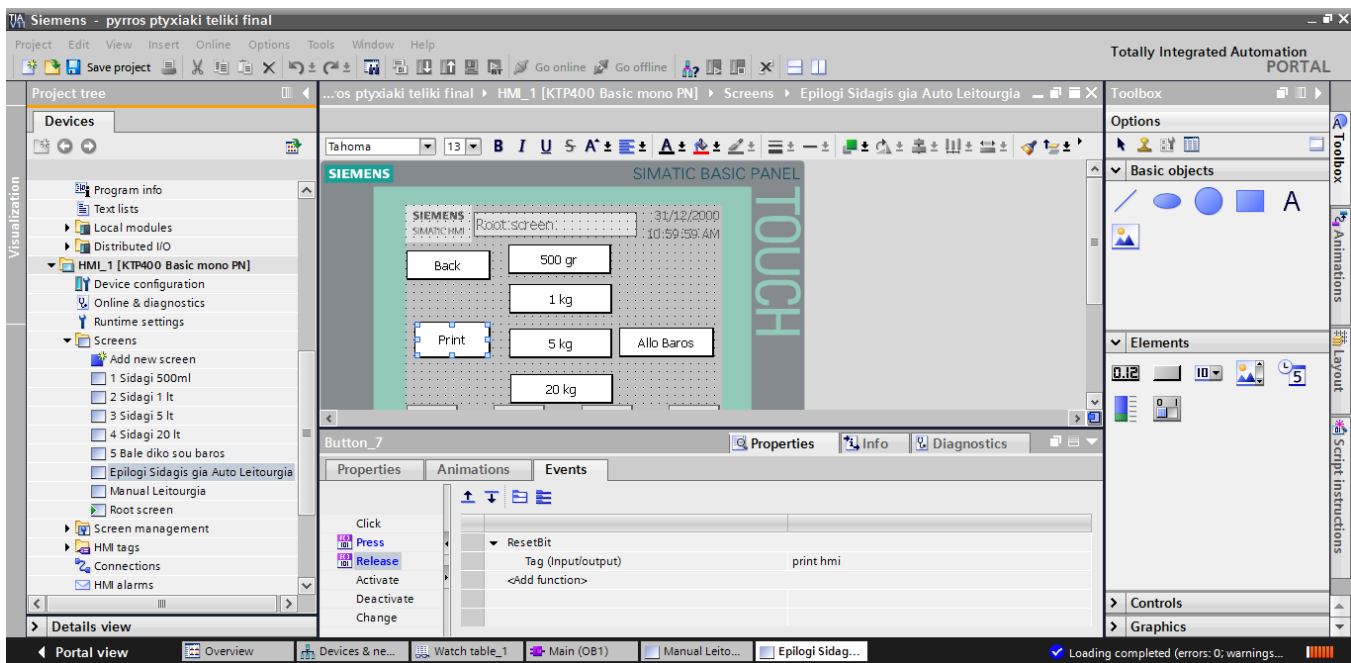
Εικόνα 5.16: Screen 3

Σε αυτή την οθόνη (εικόνα 5.16) υπάρχουν 6 buttons για μεταβίβαση σε διαφορετική οθόνη. Αυτά δημιουργήθηκαν αυτόματα μέσω του «HMI Device Wizard». Το button «500 gr» σε μεταφέρει στην οθόνη όπου γίνεται η γέμιση των δοχείων χωρητικότητας 500 gr υλικού. Το button «1 kg» σε μεταφέρει αντίστοιχα στην οθόνη όπου γίνεται η γέμιση των δοχείων χωρητικότητας 1kg υλικού. Το button «5 kg» σε αυτή όπου γίνεται η γέμιση των δοχείων χωρητικότητας 5kg υλικού και το button «20 kg» σε αυτή όπου γίνεται η γέμιση των δοχείων χωρητικότητας 20kg υλικού. Το button «Allo baros» πραγματοποιεί τη μεταβίβαση στην οθόνη όπου ο χρήστης επιλέγει από μόνος του τι ποσότητα υλικού θέλει να συσκευάσει ανά δοχείο. Επίσης κάθε button μόλις πατηθεί, press, κάνει setbit την tag «tag midenismos stoiceiwn HMI» και όταν αφεθεί, release, κάνει resetbit την tag «tag midenismos stoiceiwn HMI» για λόγο που θα φανεί στην ανάλυση του προγράμματος. Υπάρχει επίσης το button «back» που πραγματοποιεί τη μεταφορά στην προηγούμενη οθόνη, δηλαδή την «root screen».

Τέλος υπάρχει το button «print» που μόλις πατηθεί, press (εικόνα 5.17), κάνει setbit την tag «print hmi» και όταν αφεθεί, release (εικόνα 5.18), κάνει resetbit την tag «print hmi». Αυτό συμβαίνει για να σταλεί στο πρόγραμμα λογικό 1 και να δοθεί εντολή για εκτύπωση, κάτι που θα περιγραφεί στην ανάλυση του προγράμματος. Γενικά όπως θα φανεί στην ανάλυση και των υπολοίπων οθονών σε πολλά buttons πραγματοποιείται «setbit» κάποιας μεταβλητής πατώντας το συγκεκριμένο button, δηλαδή press, και «resetbit» της ίδιας μεταβλητής αφήνοντάς το, δηλαδή release. Αυτό γίνεται διότι στο πρόγραμμα οι συγκεκριμένες μεταβλητές θέλουμε να λειτουργούν ως button και όχι ως διακόπτες. Σε περίπτωση που δεν γινόταν resetbit των μεταβλητών θα ίσχυε το setbit και θα λειτουργούσαν συνέχεια σαν λογικό 1.



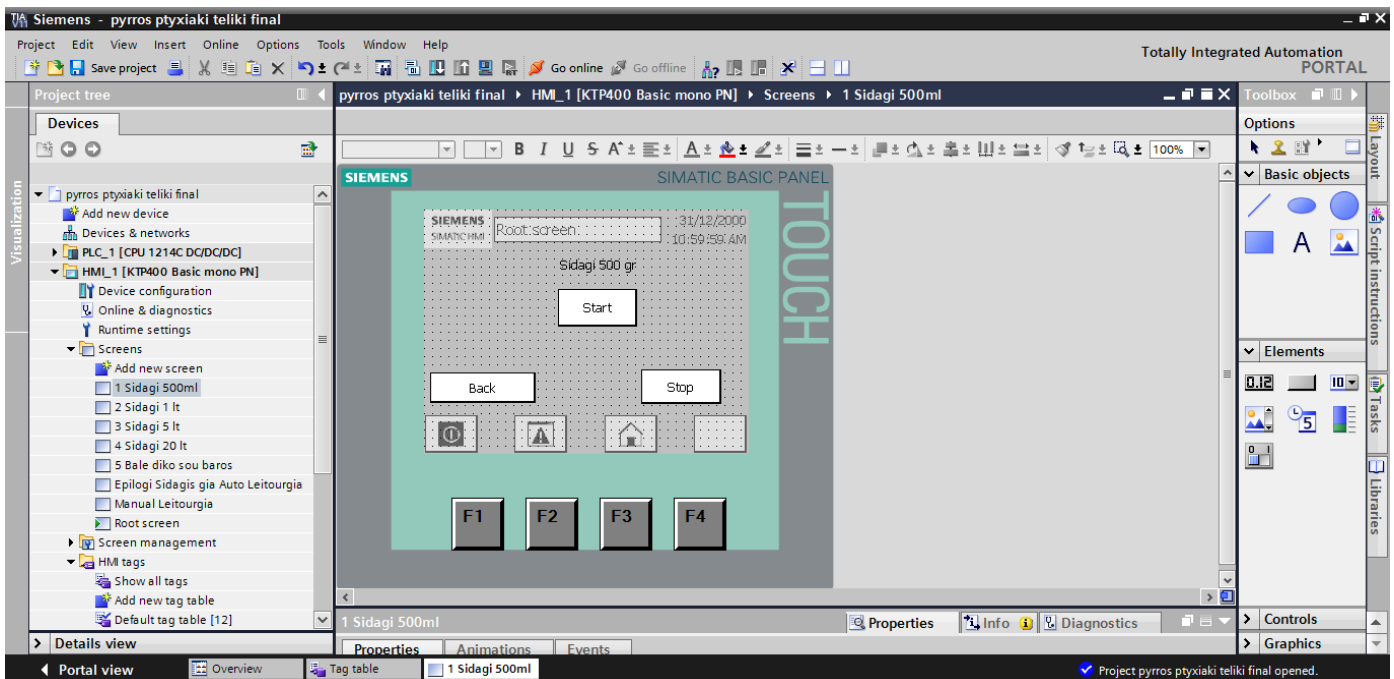
Εικόνα 5.16 Εικόνα 5.17: Ρυθμίσεις λειτουργίας του button «Print» στο menu Press



Εικόνα 5.18: Ρυθμίσεις λειτουργίας του button «Print» στο menu Release

4^η οθόνη «Sidagi 500 gr»

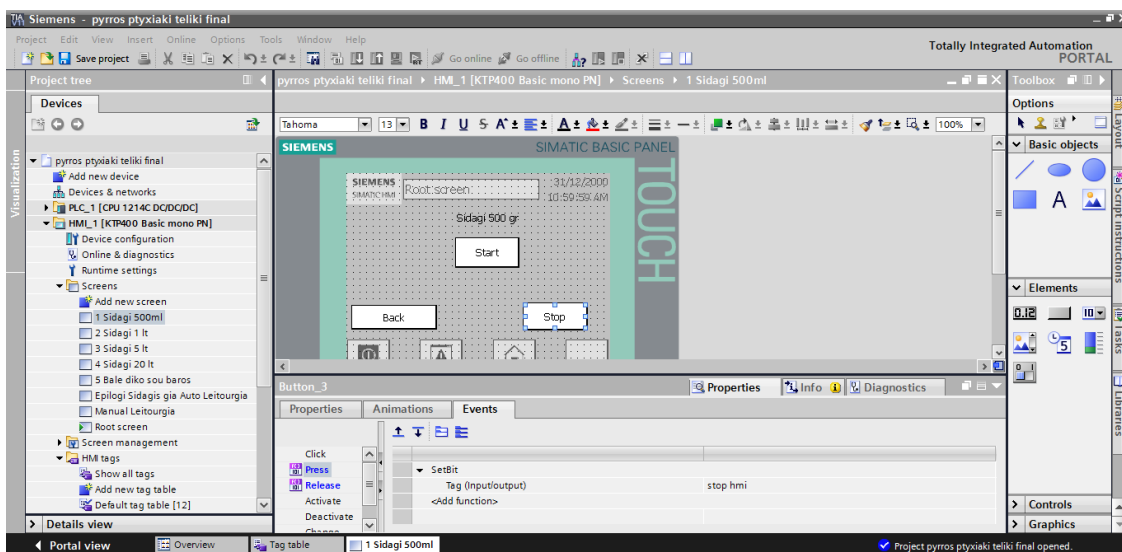
Σε αυτή την οθόνη ο χρήστης πιέζοντας το «Start» (εικόνα 5.19), η μηχανή ρίχνει υγρό και σταματάει αυτόματα μόλις πιάσει το στόχο-βάρους των 500gr.



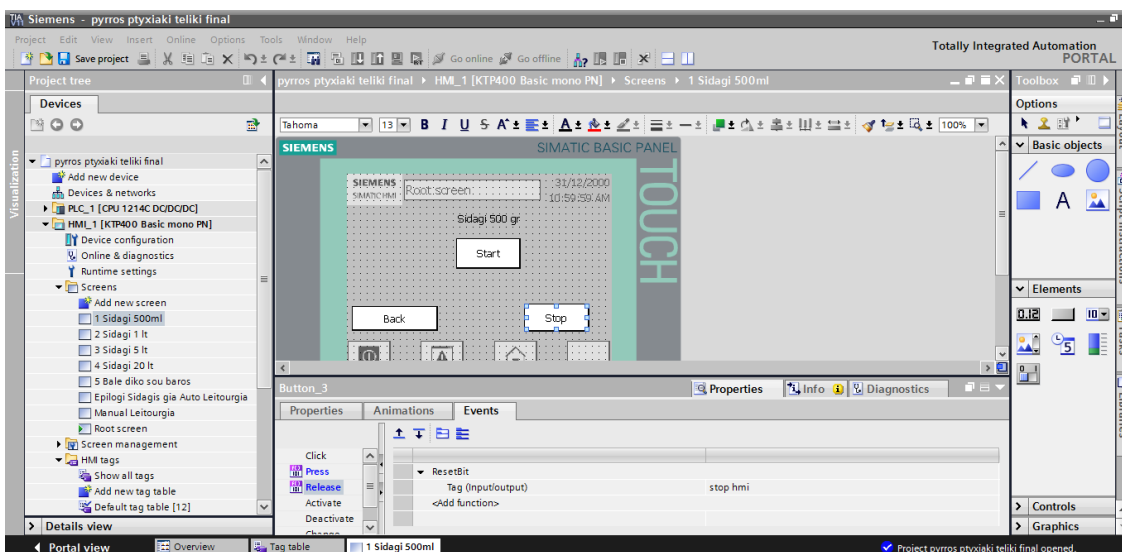
Εικόνα 5.19: Screen 4

Διακρίνεται επίσης ένα text field που γράφει «Sidagi 500 gr» και χρησιμοποιείται απλά για να ενημερώνει το χρήστη σε ποια οθόνη βρίσκεται. Η τοποθέτησή του γίνεται με τον ίδιο ακριβώς τρόπο που περιγράφηκε παραπάνω.

Επιπλέον υπάρχουν τρία buttons, ένα εκ των οποίων είναι το «Back» για την επιστροφή στη προηγούμενη οθόνη, όπου γινόταν η επιλογή της συνταγής. Το δεύτερο είναι το button «Stop», που μόλις πατηθεί σταματάει η αντλία να λειτουργεί μέσω του inverter. Το πως θα φανεί στην ανάλυση του προγράμματος. Αυτό το button υπάρχει για την περίπτωση που ο χειριστής θέλει να σταματήσει τη μηχανή να ρίχνει υγρό για κάποιον λόγο π.χ. γιατί πατήθηκε το «Start» και δεν υπάρχει δοχείο στην πλατφόρμα ζύγισης και χύνεται το υγρό έξω. Στα events του έχει σαν εντολές μόλις πατηθεί, «press» (εικόνα 5.20), να κάνει «SetBit», δηλαδή λογικό 1 την tag «stop hmi» από το HMI tag table και μόλις αφεθεί να κάνει «ResetBit» την ίδια μεταβλητή (εικόνα 5.21). Το Resetbit γίνεται για τον λόγο που περιγράφηκε και προηγουμένως, δηλαδή να λειτουργεί ως button και όχι ως διακόπτης.



Εικόνα 5.20: Ρυθμίσεις λειτουργίας του button «Stop» στο menu Press

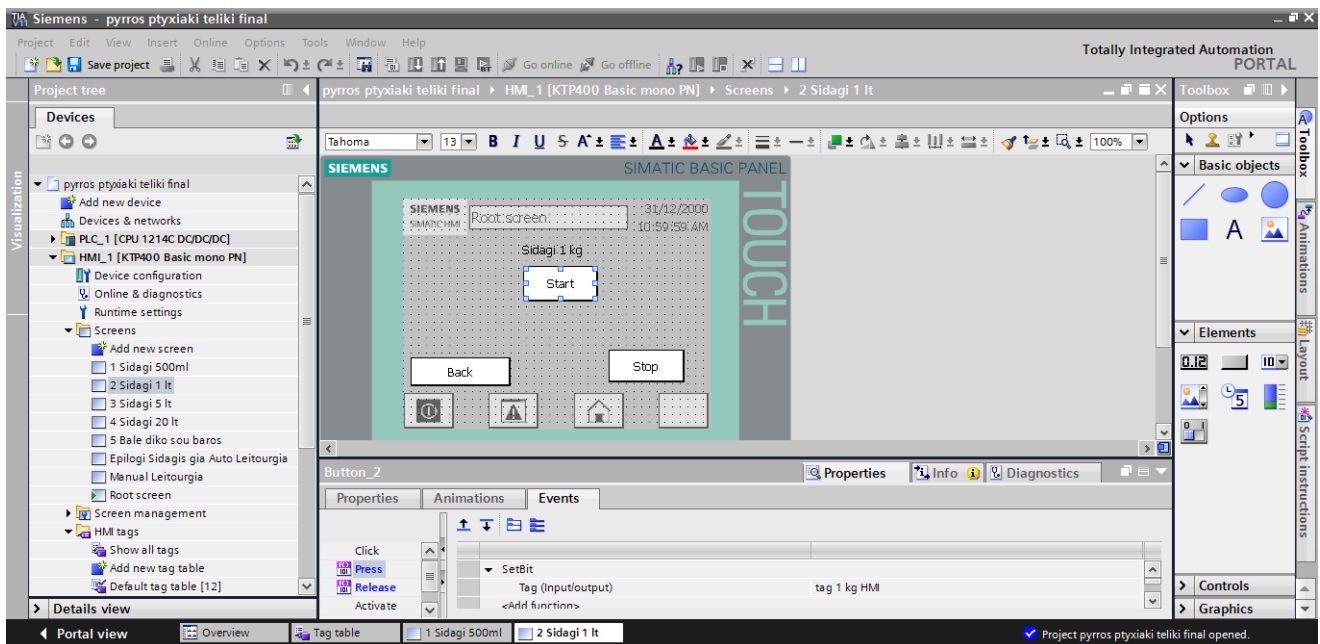


Εικόνα 5.21: Ρυθμίσεις λειτουργίας του button «Stop» στο menu Release

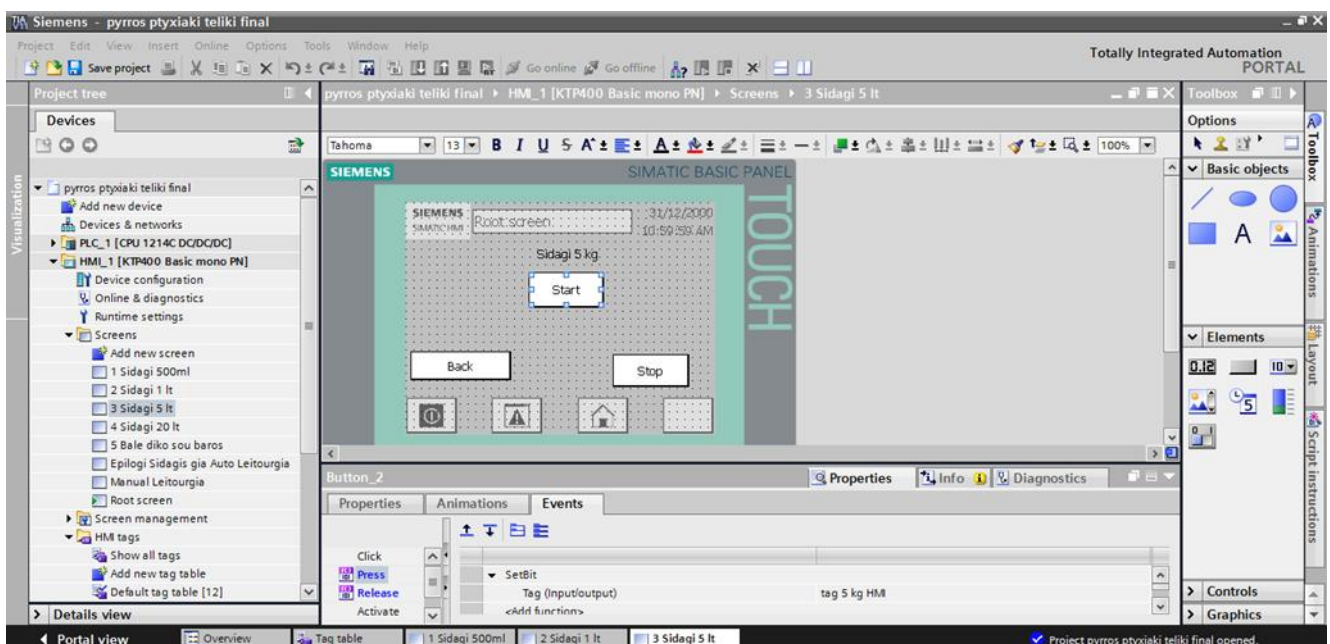
Το button «Start» μόλις πατηθεί, με παρόμοιο τρόπο με το «Stop», κάνει setbit την tag «tag 500 gr HMI» και όταν αφηθεί, κάνει resetbit την ίδια μεταβλητή. Με αυτό το τρόπο στέλνει η HMI σήμα στο PLC να ξεκινήσει η γέμιση των 500gr.

- 5^η, 6^η και 7^η οθόνη «Sidagi 1 kg», «Sidagi 5 kg», «Sidagi 20 kg»

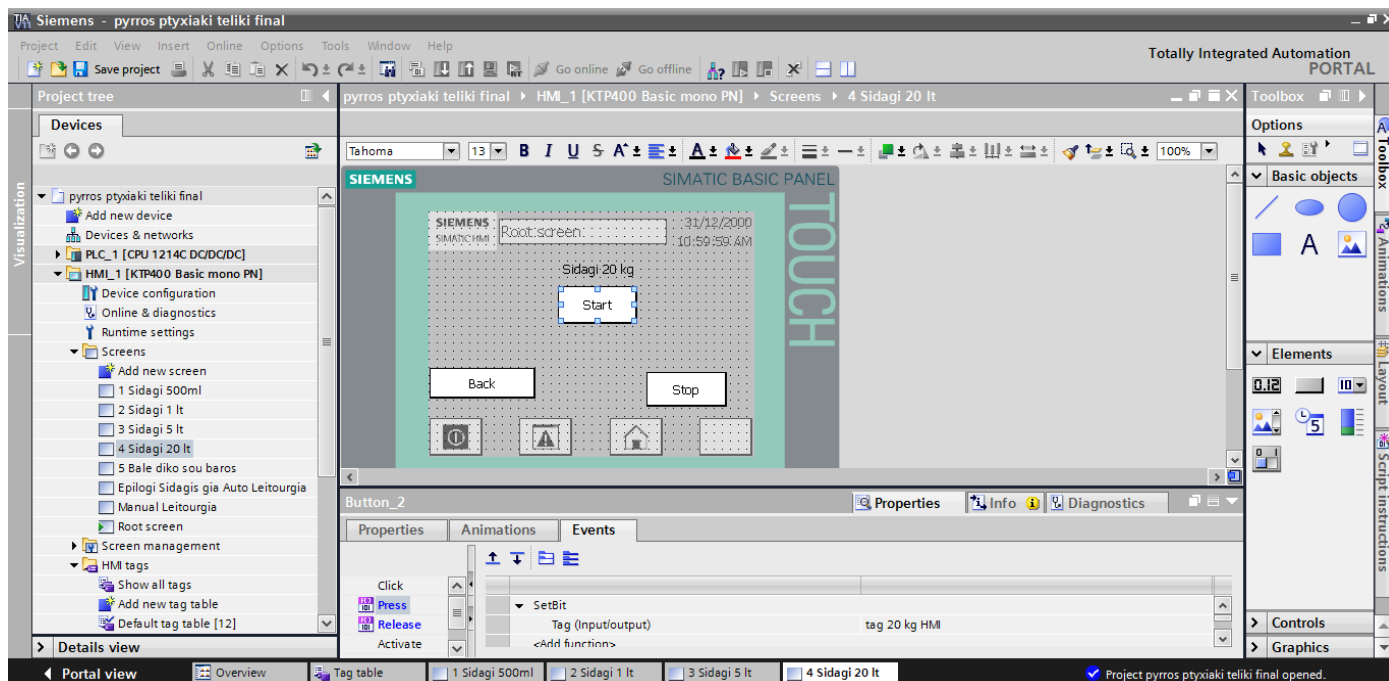
Και οι τρεις αυτές οθόνες έχουν πανομοιότυπη λειτουργία με αυτή της οθόνης «Sidagi 500 gr». Η διαφορά τους έγκειται στο ότι πιέζοντας το αντίστοιχο «Start» κάθε φορά, εισάγεται διαφορετική ποσότητα στα δοχεία.



Εικόνα 5.22: Screen 5



Εικόνα 5.23: Screen 6



Εικόνα 5.24: Screen 7

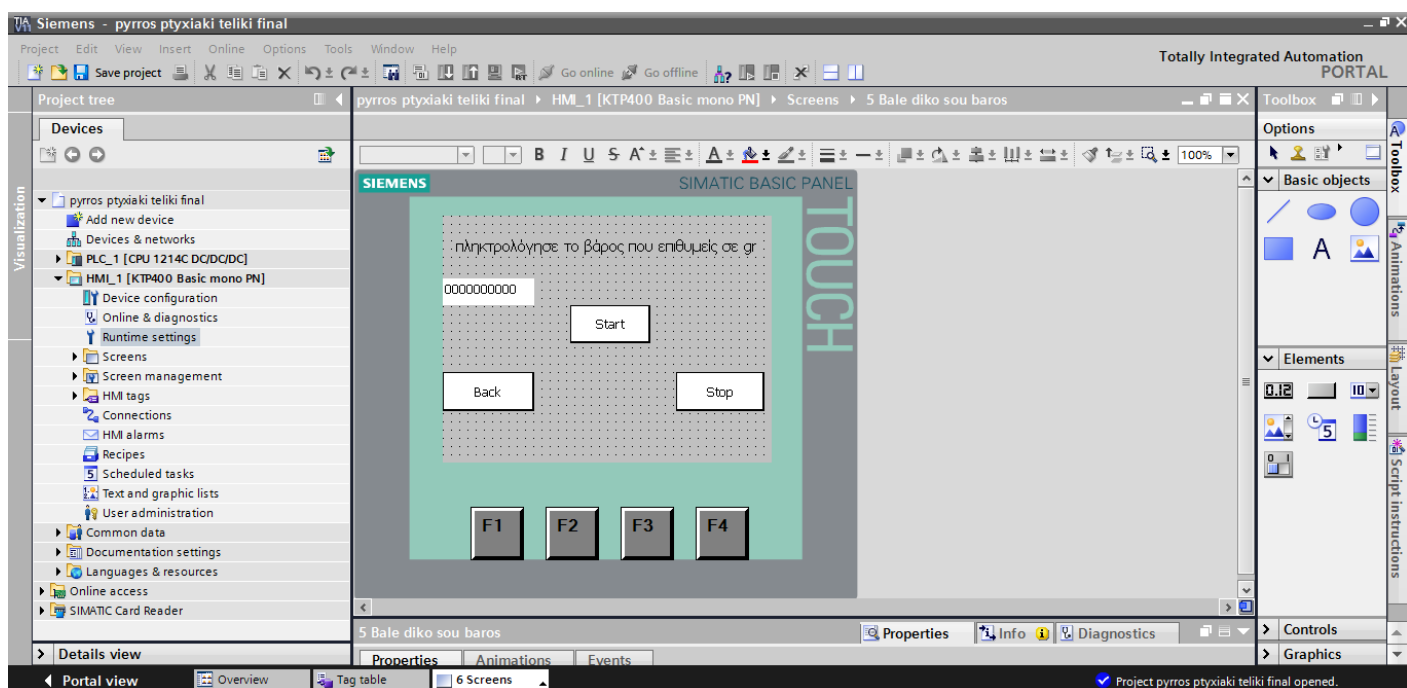
Σε αυτές τις οθόνες εμφανίζεται το αντίστοιχο text field που ενημερώνει σε ποια οθόνη είμαστε.

Και στις 3 οθόνες μόλις πατηθεί το «Back» γίνεται μεταβίβαση στην προηγούμενη οθόνη της επιλογής συνταγής.

Και στις 3 οθόνες το «Stop» έχει ίδια συμπεριφορά με αυτό της οθόνης των 500gr, δηλαδή δίνει λογικό 1 για να σταματήσει η λειτουργία της αντλίας σε περίπτωση λάθους ή ανάγκης.

Το μόνο που αλλάζει είναι ότι το κάθε «Start» σε κάθε οθόνη, κάνει setbit και στη συνέχεια resetbit την αντίστοιχη tag που ενεργοποιεί την κατάλληλη ποσότητα γέμισης. Δηλαδή το Start στην οθόνη «Sidagi 1 kg» κάνει setbit και resetbit την tag «tag 1 kg HMI», το Start στην οθόνη «Sidagi 5 kg» κάνει setbit και resetbit την tag «tag 5 kg HMI» και το Start στην οθόνη «Sidagi 20 kg» κάνει setbit και resetbit την tag «tag 20 kg HMI».

- 8^η οθόνη «Bale diko sou baros»



Εικόνα 5.25: Screen 8

Σε αυτή την οθόνη ο χειριστής έχει την δυνατότητα να επιλέξει οποιοδήποτε βάρος επιθυμεί ο ίδιος να συσκευάσει.

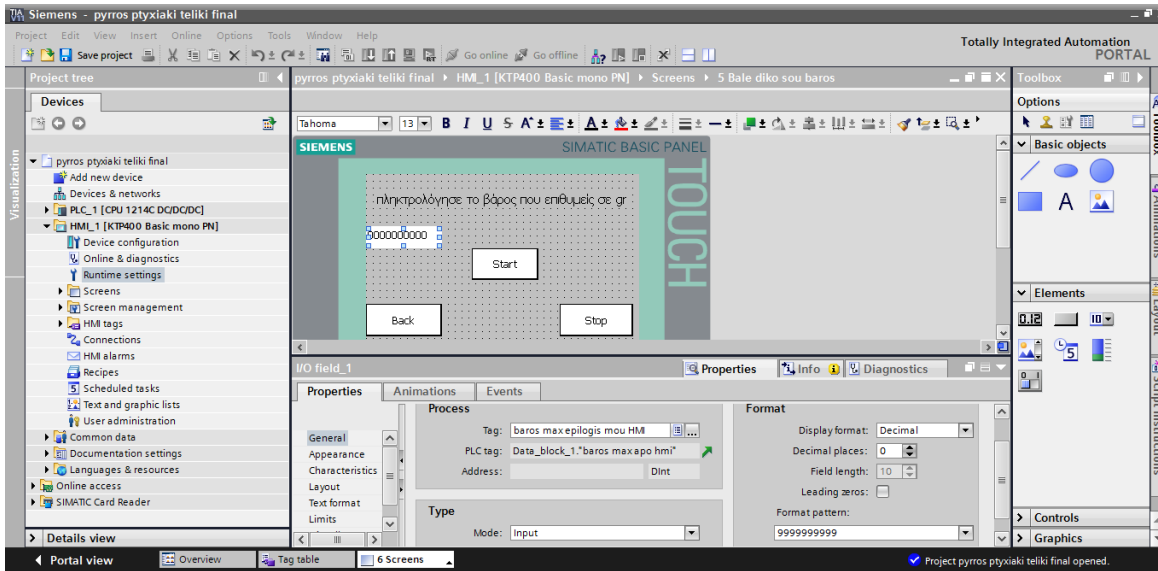
Υπάρχει ένα text field που γράφει «πληκτρολόγησε το βάρος που επιθυμείς σε gr», το οποίο και παραπέμπει το χρήστη στο ότι πρέπει να πληκτρολογήσει το βάρος που επιθυμεί αλλά και σε ποια screen βρίσκεται.

Υπάρχει ένα button «Back», το οποίο μόλις πιεστεί πραγματοποιεί τη μεταβίβαση στην προηγούμενη οθόνη, όπου γινόταν η επιλογή της συνταγής.

Υπάρχει επίσης ένα button «Stop», το οποίο μόλις πατηθεί κάνει setbit και resetbit μια tag, μέσω της οποίας γίνεται η απενεργοποίηση της λειτουργίας της αντλίας και ένα button «Start» το οποίο μόλις πατηθεί δίνει εντολή για να ξεκινήσει η γέμιση των δοχείων με την προεπιλεγμένη ποσότητα, κάνοντας setbit και resetbit την αντίστοιχη μεταβλητή.

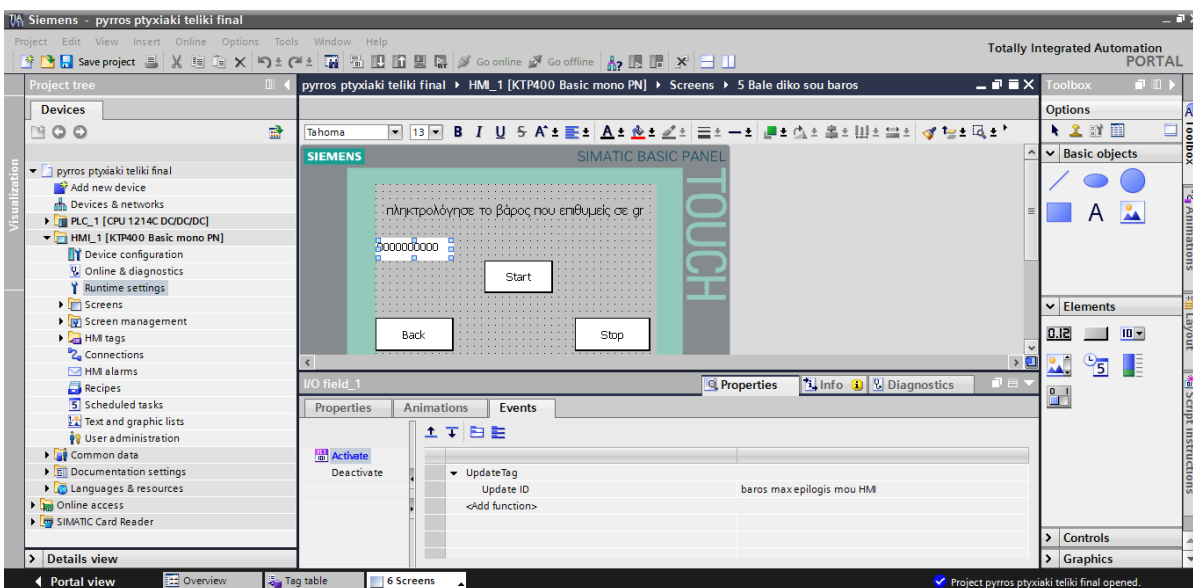
Τέλος αυτό που διαφέρει από τις προηγούμενες οθόνες είναι ότι υπάρχει ένα «I/O field». Μόλις ο χρήστης πιάσει πάνω του, ανοίγει ένα παράθυρο, στο οποίο πληκτρολογεί το επιθυμητό βάρος και πιάζει OK. Στα properties του συγκεκριμένου I/O field έχουν γίνει οι εξής ρυθμίσεις (εικόνα 5.26). Αρχικά στο process (στο menu των properties) επιλέγεται σε ποια μεταβλητή να εκχωρείται η τιμή που έδωσε ο χρήστης. Η μεταβλητή εκχώρησης είναι το «baros max epilogis mou HMI», η οποία συνδέεται με την μεταβλητή «Data_block_1.baros max apo hmi». Με αυτό τον τρόπο η «baros max apo hmi» που βρίσκεται στο Data_block_1 παίρνει την τιμή που επέλεξε ο χρήστης. Στη συνέχεια στο type επιλέγεται «input», δηλαδή ότι αυτή η τιμή είναι είσοδος για το σύστημα και

δηλώνεται από το χρήστη. Τέλος στο format δηλώνεται ο αριθμητικός τύπος. Η επιλογή είναι «decimal», δηλαδή σε δεκαδική μορφή και ο αριθμός των δεκαδικών ψηφίων είναι μηδέν, αφού η εκχώρηση θεωρείται ότι είναι σε γραμμάριο και δε χρειάζεται δεκαδικά.



Εικόνα 5.26: Ρυθμίσεις του I/O field όσον αφορά τα properties

Στη συνέχεια στα Events επιλέγεται στο activate ως εντολή η «updateTag» και στο update ID (ποια τιμή να γίνεται update) δηλώθηκε ξανά η μεταβλητή «baros max epilogis mou HMI» (εικόνα 5.27). Σε αυτό το σημείο δηλώνεται η ενέργεια που συμβαίνει με το πάτημα του συγκεκριμένου I/O field, και επιλέχθηκε η ανανέωση της παραπάνω μεταβλητής με την τιμή που επέλεξε ο χρήστης.



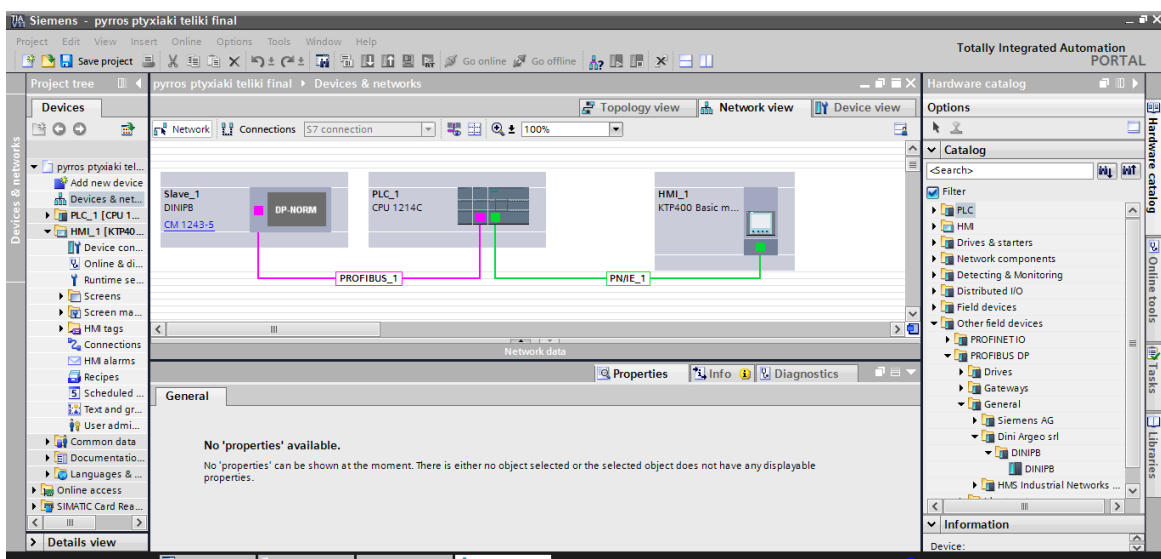
Εικόνα 5.27: Ρυθμίσεις του I/O field όσον αφορά τα events

5.6 Ρυθμίσεις για την σύνδεση μέσω Profibus

Εν συνεχεία πριν υλοποιηθεί η δημιουργία του προγράμματος θα πραγματοποιηθούν οι ρυθμίσεις που χρειάζονται για την λειτουργία και την επικοινωνία του PLC με τον ενδείκτη DGTQPB που πραγματοποιείται μέσω Profibus.

Στο «Devices and networks» από το Project tree, φαίνεται το hardware configuration που έχει δημιουργηθεί μετά την παραπάνω διαδικασία που περιγράφηκε. Η εισαγωγή του ενδείκτη έχει γίνει απ' το hardware catalog στο δεξί μέρος της οθόνης (εικόνα 5.28), ο οποίος κατόπιν αναζήτησης έχει βρεθεί και με διπλό κλικ έχει εισαχθεί στο network. Μετά από το ροζ κουτάκι του PLC, που βρίσκεται στο module «CM1243-5 PROFIBUS DP-MASTER» έχουμε σύρει με το ποντίκι μία γραμμή μέχρι το ροζ κουτάκι που έχει η συσκευή, που αναπαριστά τον ενδείκτη. Αυτόματως αυτά τα δύο έχουν ενωθεί με μία συνεχόμενη ροζ γραμμή που αντικατοπτρίζει την μεταξύ τους σύνδεση.

Βέβαια για να πραγματοποιηθεί και στη πράξη η σύνδεση τους πρέπει να γίνουν κάποιες ρυθμίσεις στον ενδείκτη, στο module αλλά και στη μεταξύ τους σύνδεση.

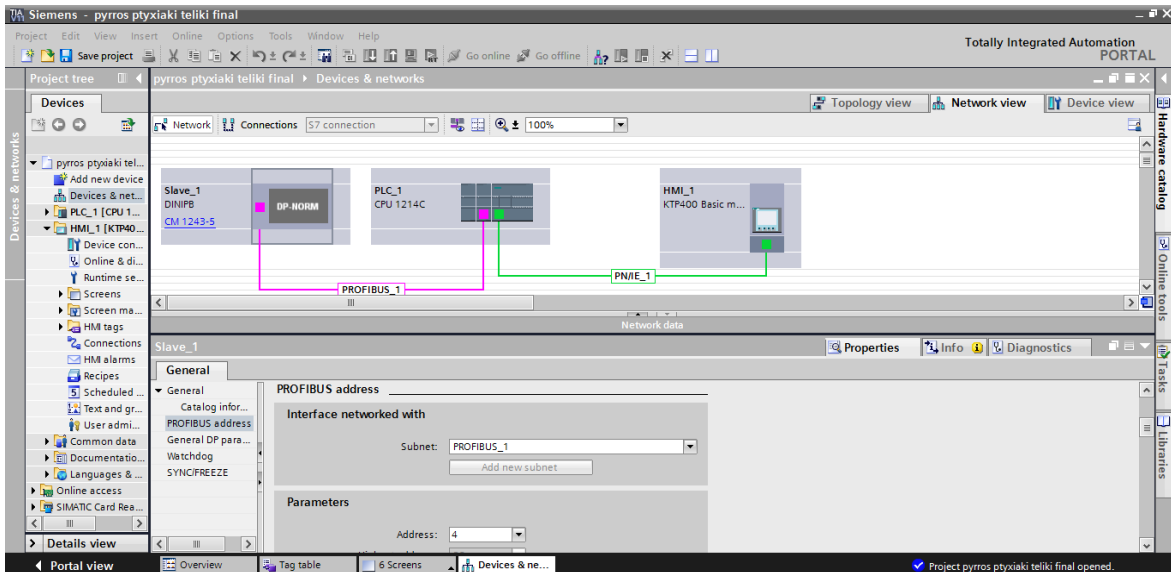


Εικόνα 5.28: Network view και η λίστα του hardware catalog

- Ρυθμίσεις στον ενδείκτη DGTQPB (Slave 1)

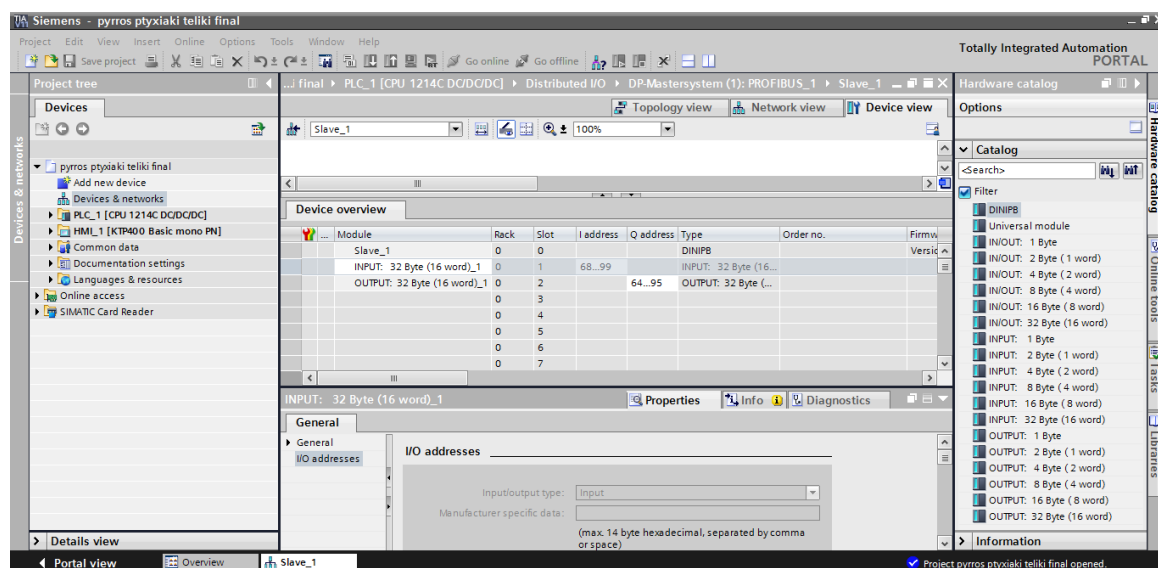
Πιέζοντας πάνω στον ενδείκτη, εμφανίζονται τα properties του Slave_1, δηλαδή του ενδείκτη (εικόνα 5.29). Οι ρυθμίσεις που απαιτούνται βρίσκονται στη κατηγορία PROFIBUS address. Εκεί δόθηκε μία διεύθυνση στον ενδείκτη, η οποία θα είναι αυτή που τον αντικατοπτρίζει στο συγκεκριμένο Profibus δίκτυο. Επιλέχθηκε η διεύθυνση 4 (αυτή η ρύθμιση έχει γίνει και στις ρυθμίσεις του menu του ίδιου του ενδείκτη). Επίσης επιλέχθηκε σε ποιο δίκτυο Profibus θα συμμετέχει (στη συγκεκριμένη περίπτωση υπάρχει μόνο ένα), το PROFIBUS_1. Τέλος πάνω στο σχήμα του ενδείκτη με δεξί κλικ του ανατέθηκε ποιος είναι ο

master του συγκεκριμένου δικτύου. Στη συγκεκριμένη περίπτωση ο CM1243-5, όπου είναι το module επικοινωνίας του PLC για σύνδεση μέσω Profibus .



Εικόνα 5.29: Ρυθμίσεις του ενδείκτη DGTQPB για επικοινωνία μέσω Profibus

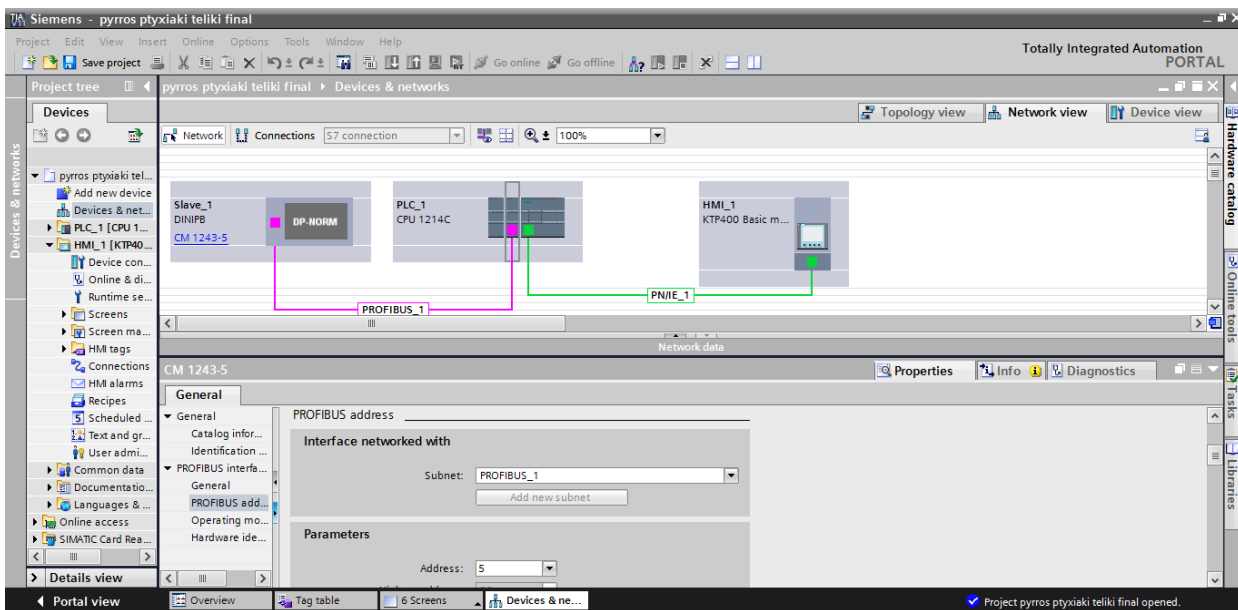
Επιπλέον στο device overview του ενδείκτη επιλέγεται ποιος καταχωρητής του DGTQPB θα διαβάζεται από το PLC. Ο καταχωρητής που χρειάζεται είναι αυτός που περιέχει το βάρος. Από το manual του ενδείκτη το, βάρος βρίσκεται στον καταχωρητή INPUT. Αυτός έχει μέγεθος 32bytes, το οποίο χωρίζεται σε 16words όπου κάθε word απεικονίζει κάποια ξεχωριστή πληροφορία. Αυτά τα 32 bytes τα αποθηκεύει το PLC από το 168 έως το 199. Βέβαια εμάς δεν μας ενδιαφέρουν στη συγκεκριμένη περίπτωση όλα τα δεδομένα όλων των 16 words, αλλά μόνο αυτό που έχει το βάρος και το οποίο βρίσκεται στο 170.



Εικόνα 5.30: Επιλογή του καταχωρητή Input που περιέχει το βάρος

- Ρυθμίσεις στο module CM1243-5 PROFIBUS DP

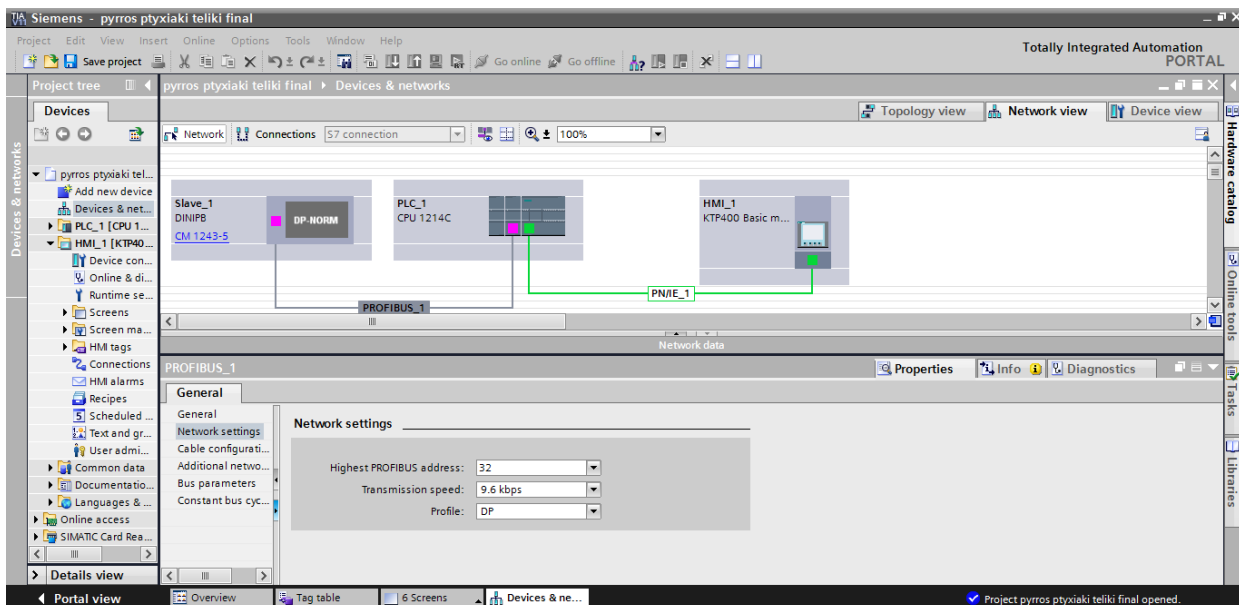
Πιέζοντας πάνω στο module CM1243-5 στην οθόνη του network & devices, επιλέγεται το menu general->PROFIBUS interface->PROFIBUS address (εικόνα 5.31). Εδώ ορίζεται σε ποιο δίκτυο Profibus συμμετέχει και ποια θα είναι η address του συγκεκριμένου module στο συγκεκριμένο δίκτυο. Επιλέγεται στο μενου subnet το «PROFIBUS_1» και ως address το 5. Η address είναι τυχαία με τον μόνο περιορισμό να είναι μοναδική στο συγκεκριμένο δίκτυο.



Εικόνα 5.31: Ρυθμίσεις για το CM1243-5 PROFIBUS DP σε σύνδεση δικτύου Profibus

- Γενικές ρυθμίσεις του συγκεκριμένου δικτύου Profibus

Πάνω στη ροζ γραμμή που ενώνει τις δύο συσκευές επιλέγονται κάποιες γενικές ρυθμίσεις. Στα properties, στα network settings επιλέγεται σαν «highest profibus address» το 32 (μία τιμή επιλογής μας). Σαν «transmission speed» επιλέγεται 9,6kbps γιατί τόση δίνει το εγχειρίδιο χρήσης του ενδείκτη DGTQPB και σαν «profile» το DP γιατί αυτό το είδος Profibus χρησιμοποιείται.



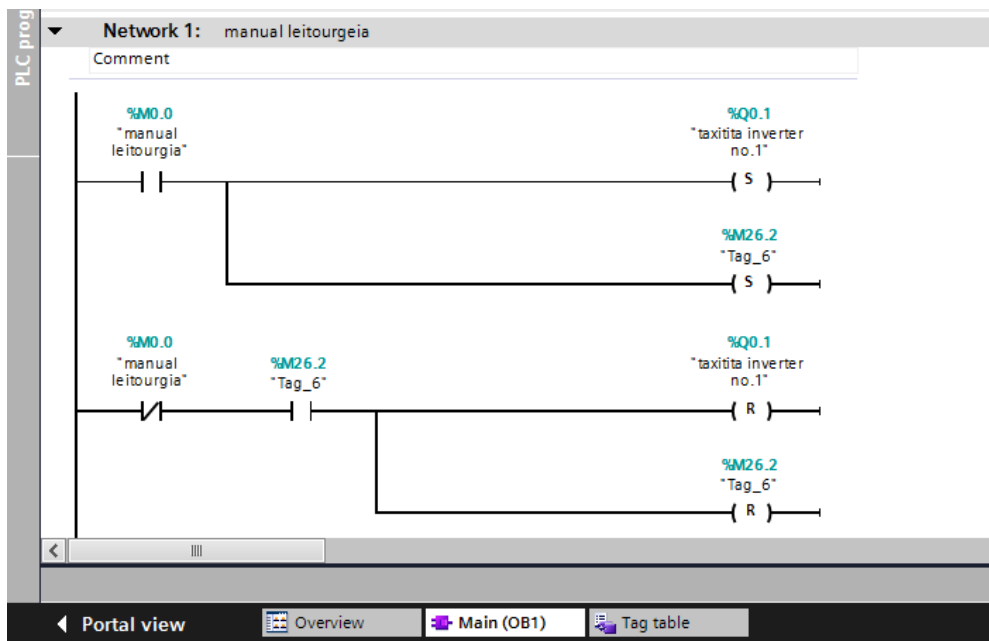
Εικόνα 5.32: Γενικές ρυθμίσεις στην σύνδεση Profibus

5.7 Ανάλυση προγράμματος στο OB1

Στο κεφάλαιο αυτό αναλύεται το πρόγραμμα που υλοποιεί τη λειτουργία και τον έλεγχο της γεμιστικής μηχανής. Το OB1 χωρίζεται σε 15 διαφορετικά networks. Το κάθε network περιέχει ένα ξεχωριστό κομμάτι κώδικα, που πραγματοποιεί κάποια συγκεκριμένη διαδικασία του αυτοματισμού. Το πρόγραμμα έχει δημιουργηθεί με βάση το γραμμικό προγραμματισμό. Στη συνέχεια αναλύονται οι διαδικασίες του κάθε network.

Network 1: Manual Leitourgia

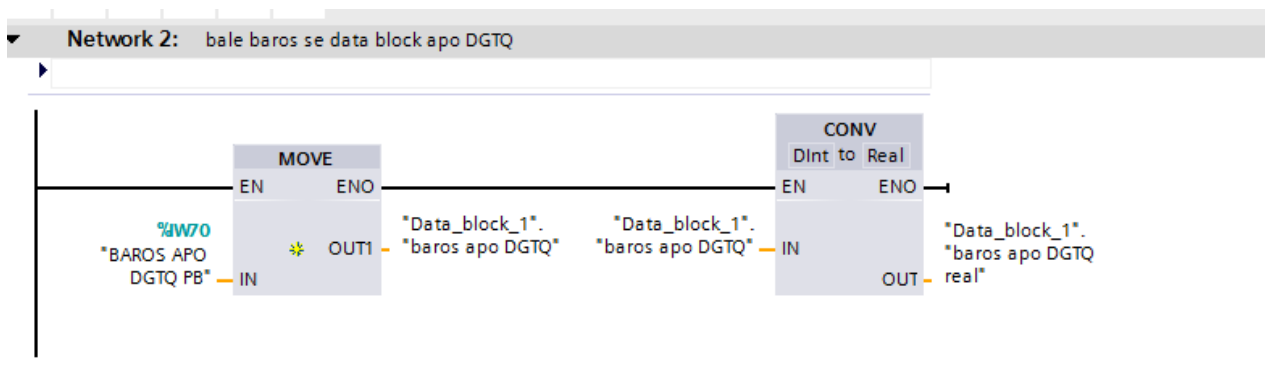
Στο Network 1 πραγματοποιείται η manual λειτουργία. Όπως φαίνεται στην εικόνα 5.33 μόλις γίνει λογικό 1 η μεταβλητή «manual leitourgia», τότε γίνεται 1 (set) και η έξοδος του PLC Q0.1. Για να γίνει 1 η μεταβλητή «manual leitourgia» πρέπει να πατηθεί το button F1 από την 2^η οθόνη (εικόνα 5.14). Η έξοδος Q0.1 του PLC συνδέεται σε μία είσοδο του Inverter, η οποία μόλις γίνει 1, ενεργοποιεί το inverter σε συχνότητα τέτοια ώστε η αντλία να λειτουργεί με την γρήγορη ταχύτητα. Αυτό συμβαίνει όση ώρα είναι 1 η μεταβλητή «manual leitourgia», διότι μαζί με τη Q0.1 γίνεται set και η μνήμη M26.2 την οποία κάνουμε reset μόλις αφήσουμε το κουμπί από την 2^η οθόνη. Παράλληλα γίνεται reset και η Q0.1. Με αυτό το τρόπο επιτυγχάνεται η Q0.1 να είναι λογικό 1 μόνο όσο πιέζεται το button και μόλις αφηθεί να σταματάει να δίνει στροφές το inverter στην αντλία. Με την παραπάνω διαδικασία ο χρήστης έχει τη δυνατότητα να ρίχνει όσο υγρό επιθυμεί πιέζοντας το button F1 από την 2^η οθόνη και μόλις το αφήνει να διακόπτεται και η τροφοδοσία του υγρού.



Εικόνα 5.33: Network 1

Network 2: Τοποθέτηση βάρους από το DGTQPB στο data_block_1

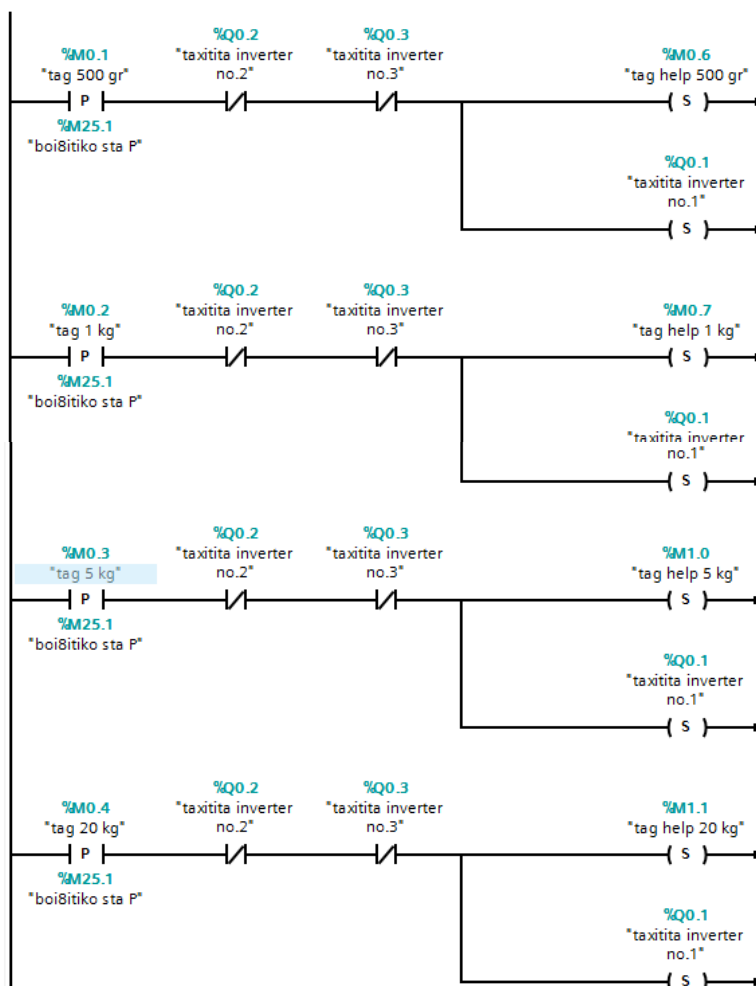
Με την εντολή move μεταφέρεται το περιεχόμενο της μεταβλητής που βρίσκεται στην αριστερή θέση, δηλαδή στο IN, στο περιεχόμενο αυτής που βρίσκεται δεξιά, δηλαδή στο OUT1. Επομένως ότι έχει η μεταβλητή IW70 μεταφέρεται στο «baros apo DGTQ» που ανήκει στο data_block_1. Θυμόμαστε ότι στο IW70 αποθηκεύει το PLC το βάρος από τον ενδείκτη. Μετά τη μεταφορά, το «baros apo DGTQ» έχει data type «Double integer». Επειδή όμως το βάρος θα χρειαστεί σε διάφορες αριθμητικές πράξεις, στη συνέχεια μετατρέπεται σε data type «real» μέσω της εντολής CONV.



Εικόνα 5.34: Network 2

Network 3 :Στην αυτόματη λειτουργία τυποποιημένων συνταγών ξεκινάει η γέμιση με την πρώτη ταχύτητα, τη γρήγορη.

Στην εικόνα 5.35 φαίνεται ότι μόλις γίνει 1 το «tag 500 gr» πρέπει να ξεκινήσει η αντλία μέσω της Q0.1. Αν δούμε τη 4^η οθόνη (εικόνα 5.19), μόλις πατηθεί το button Start γίνεται 1 η μεταβλητή «M0.1» και παραμένει 1 για όση ώρα είναι πατημένο, ενώ μόλις αφεθεί γίνεται 0. Στη συγκεκριμένη περίπτωση δεν ενδιαφέρει η διάρκεια αλλά μόνο η στιγμή που θα πατηθεί, δηλαδή από 0 θα γίνει 1. Για αυτό το λόγο και χρησιμοποιείται η εντολή «/P/» που αντιλαμβάνεται την αλλαγή από 0 σε 1. Μόλις γίνει 1 η «M0.1», κάνει set την έξοδο Q0.1 που αντιστοιχεί στην πρώτη ταχύτητα γέμισης, την γρήγορη. Επίσης κάνει set μία βοηθητική μεταβλητή την «tag help 500 gr», η οποία χρησιμοποιείται για την συνέχεια του προγράμματος και θα δηλώνει ότι βρισκόμαστε σε κατάσταση γέμισης. Εκτός από την M0.1 υπάρχουν σε σειρά και άλλες δύο μεταβλητές, οι Q0.2 και Q0.3, οι οποίες αντιστοιχούν στη 2^η και 3^η ταχύτητα της αντλίας. Αυτό σημαίνει ότι η πρώτη ταχύτητα, δηλαδή η Q0.1 και η βοηθητική μεταβλητή «tag help 500 gr», θα γίνουν 1 μόνο σε περίπτωση που αυτές οι δύο μεταβλητές είναι λογικό 0, δηλαδή δεν δουλεύει καμία εκ των δύο ταχυτήτων. Αυτό επιτυγχάνεται με το οι Q0.2 και Q0.3 να είναι σε κλειστή επαφή. Αυτόν τον περιορισμό τον παίρνουμε σε περίπτωση που το γεμιστικό γεμίζει ήδη με μία εκ των δύο ταχυτήτων και ο χρήστης πιέσει εκείνη την στιγμή καταλάθος start.



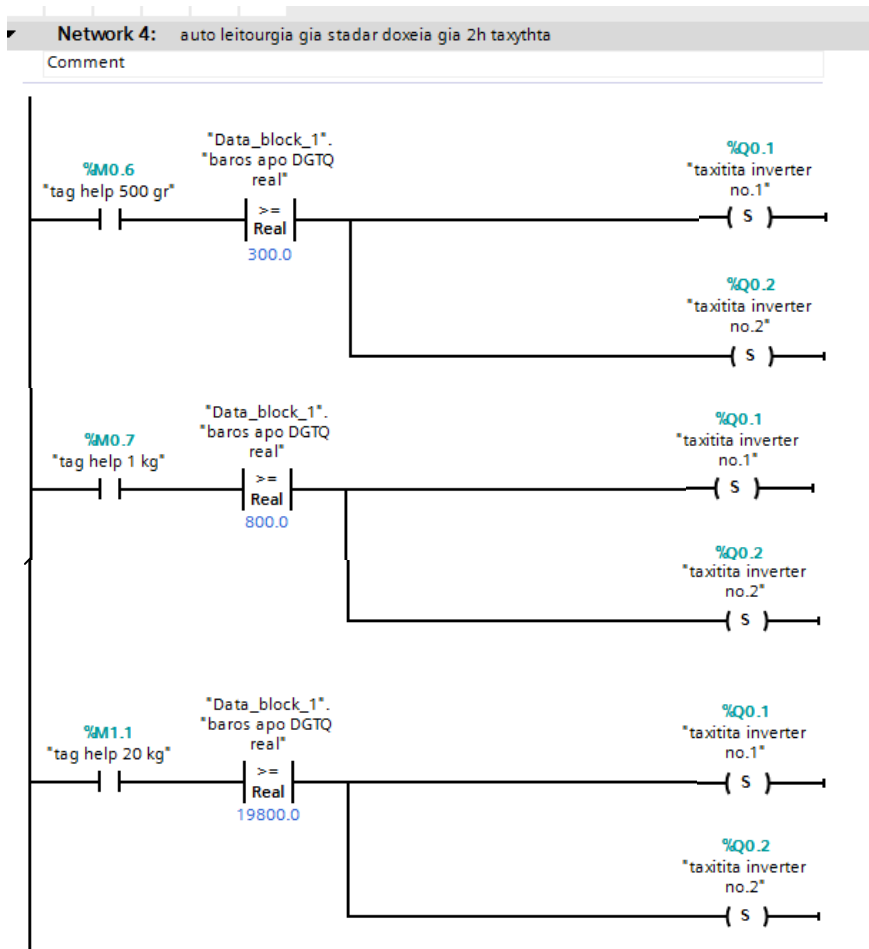
Εικόνα 5.35: Network 3

Η ακριβώς πανομοιότυπη λογική ισχύει και με τις υπόλοιπες μεταβλητές που αντιπροσωπεύουν τις υπόλοιπες τυποποιημένες ποσότητες (εικόνα 5.35). Όλες ενεργοποιούν την πρώτη ταχύτητα της αντλίας, με την προϋπόθεση ότι δεν δουλεύει ήδη η αντλία, δηλαδή κάποια από τις ταχύτητες No2 ή No3. Βέβαια σε κάθε περίπτωση κάνουν set διαφορετική βοηθητική μεταβλητή. Η «tag 1 kg» κάνει λιγικό 1 την «tag help 1 kg», η «tag 5 kg» κάνει 1 την «tag help 5 kg» και η «tag 20 kg» κάνει 1 την «tag help 20 kg», οι οποίες θα δείχνουν ότι είμαστε σε κατάσταση γέμισης και σε ποια τυποποιημένη συνταγή συγκεκριμένα.

Network 4 : Auto λειτουργία για τυποποιημένα δοχεία με τη 2^η ταχύτητα

Μόλις πατηθεί από την αντίστοιχη screen της κάθε συνταγής στην HMI το button Start, φάνηκε στο προηγούμενο network ότι ξεκινάει να ρίχνει η αντλία με την γρήγορη ταχύτητα. Παράλληλα ενεργοποιήθηκε και η αντίστοιχη βοηθητική μεταβλητή. Άρα για να ενεργοποιηθεί εδώ η δεύτερη ταχύτητα πρέπει απλά το βάρος να γίνει μεγαλύτερο ή ίσο της αντίστοιχης τιμής κάθε φορά.

Το αποτέλεσμα αυτού του network είναι να γίνονται λογικό 1 οι έξοδοι του PLC Q0.1 και Q0.2 (εικόνα 5.36). Και οι δύο έξοδοι δίνουν σήμα στο inverter, το οποίο όταν πάρει ταυτόχρονα λογικό 1 από αυτές τις εξόδους, τότε λειτουργεί με κάποια άλλη συχνότητα, η οποία αντιστοιχεί στη δεύτερη ταχύτητα πλήρωσης (πιο αργή από την πρώτη). Αυτό για να συμβεί, πρέπει να είναι 1 η μεταβλητή «tag help 500 gr», η οποία έχει γίνει από το network 3, αλλά και το βάρος να είναι πάνω από 300 gr. Ο δεύτερος περιορισμός δεν είναι απόλυτος, αλλά οι δοκιμές έχουν δείξει ότι είναι ένας πολύ καλός στόχος για αλλαγή ταχύτητας στα 500 gr. Η αλλαγή μείωσης της ταχύτητας συμβαίνει για να υπάρξει ακρίβεια στη γέμιση, καθώς επιθυμούμε όσο φτάνουμε στο στόχο-βάρος να ρέει το υγρό με πιο αργή ταχύτητα για πιο εύκολη επίτευξη ακρίβειας.



Εικόνα 5.36: Network 4

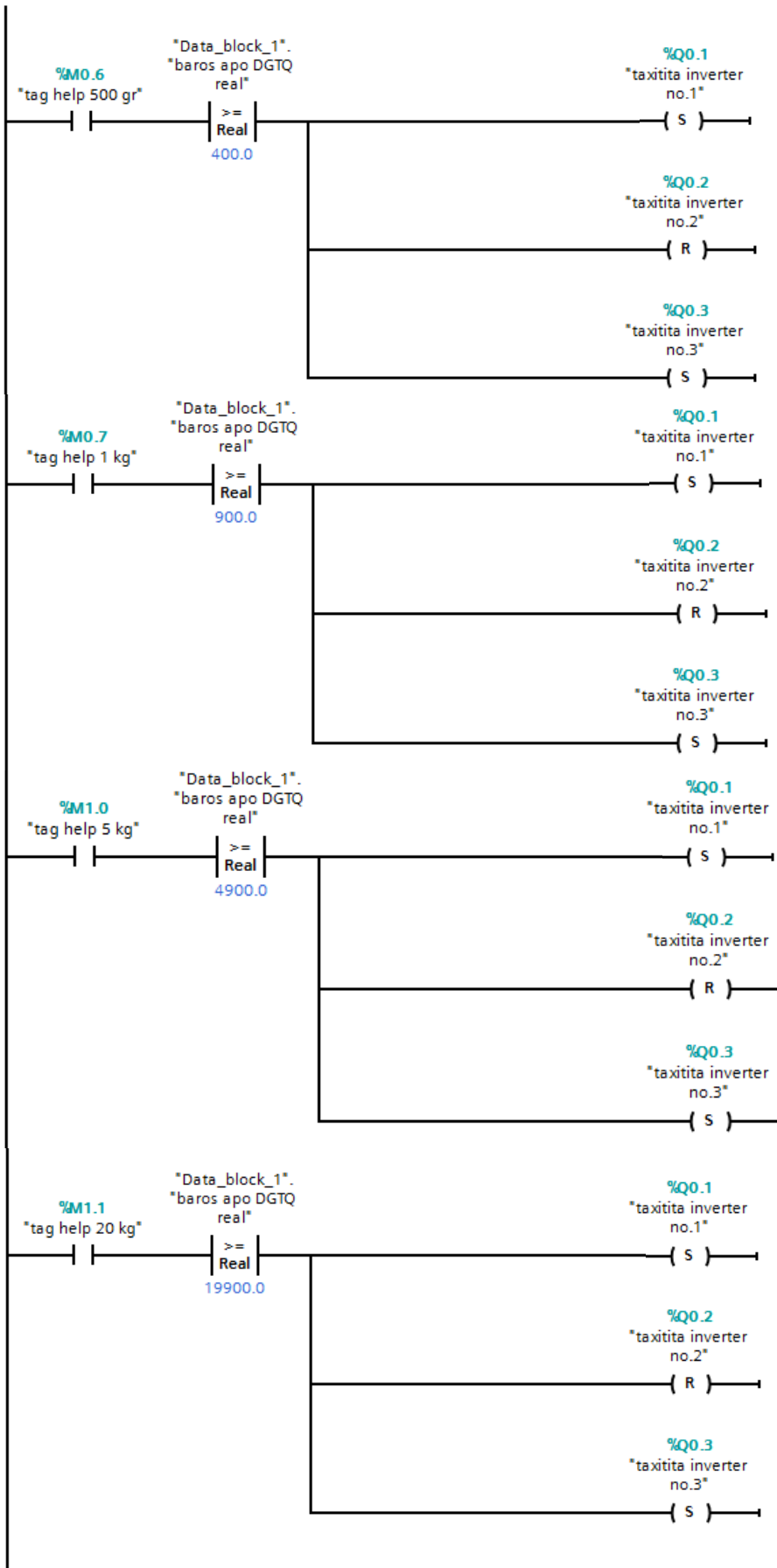
Ακριβώς τα ίδια συμβαίνουν και με τις υπόλοιπες συσκευασίες. Η διαφορά είναι ότι για να ενεργοποιηθεί η αντλία στη 2^η ταχύτητα πρέπει να είναι λογικό 1 η βοηθητική μεταβλητή της συγκεκριμένης συνταγής. Αυτό που διαφέρει επίσης είναι ότι η ενεργοποίηση γίνεται σε διαφορετικά βάρη. Φυσικό είναι όσο μεγαλώνει το δοχείο να μεγαλώνει και ο στόχος αλλαγής. Αυτό συμβαίνει γιατί σε ένα μεγάλο δοχείο όπως των 20kg δεν υπάρχει λόγος να πέσει η ταχύτητα νωρίς γιατί θα αργήσει να τελειώσει η γέμιση. Οι στόχοι στους οποίους ενεργοποιείται η δεύτερη ταχύτητα είναι 800gr για το 1kg, 4800gr για τα 5kg και 19800gr για τα 20kg.

Network 5: Auto λειτουργία για τυποποιημένα δοχεία με τη 3^η ταχύτητα

Η λειτουργία αυτού του network (εικόνα 5.37) είναι η ενεργοποίηση της τρίτης ταχύτητας της αντλίας που είναι και η πιο αργή. Η λογική είναι παρόμοια με το προηγούμενο network. Για να λειτουργήσει το inverter με αυτή τη συχνότητα πρέπει να λάβει λογικό 1 ταυτόχρονα από τις εξόδους Q0.1 και Q0.3. Για αυτό το λόγο το αποτέλεσμα του network είναι να γίνονται set οι Q0.1 και Q0.3 και reset η Q0.2. Η αλλαγή αυτή συμβαίνει σε μεγαλύτερο βάρος-στόχο για κάθε συνταγή ξεχωριστά. Η μείωση της ταχύτητας πλήρωσης συντελεί στην καλύτερη ακρίβεια γέμισης. Οι αλλαγές για κάθε δοχείο συμβαίνουν 100gr από το τελικό στόχο, δηλαδή στα 400gr για τα 500gr, στα 900gr για το 1kg, στα 4900gr για τα 5kg, και στα 19900gr για τα 20 kg.

Network 5: auto leitourgia gia stadar doxeia gia 3h taxitita

Comment



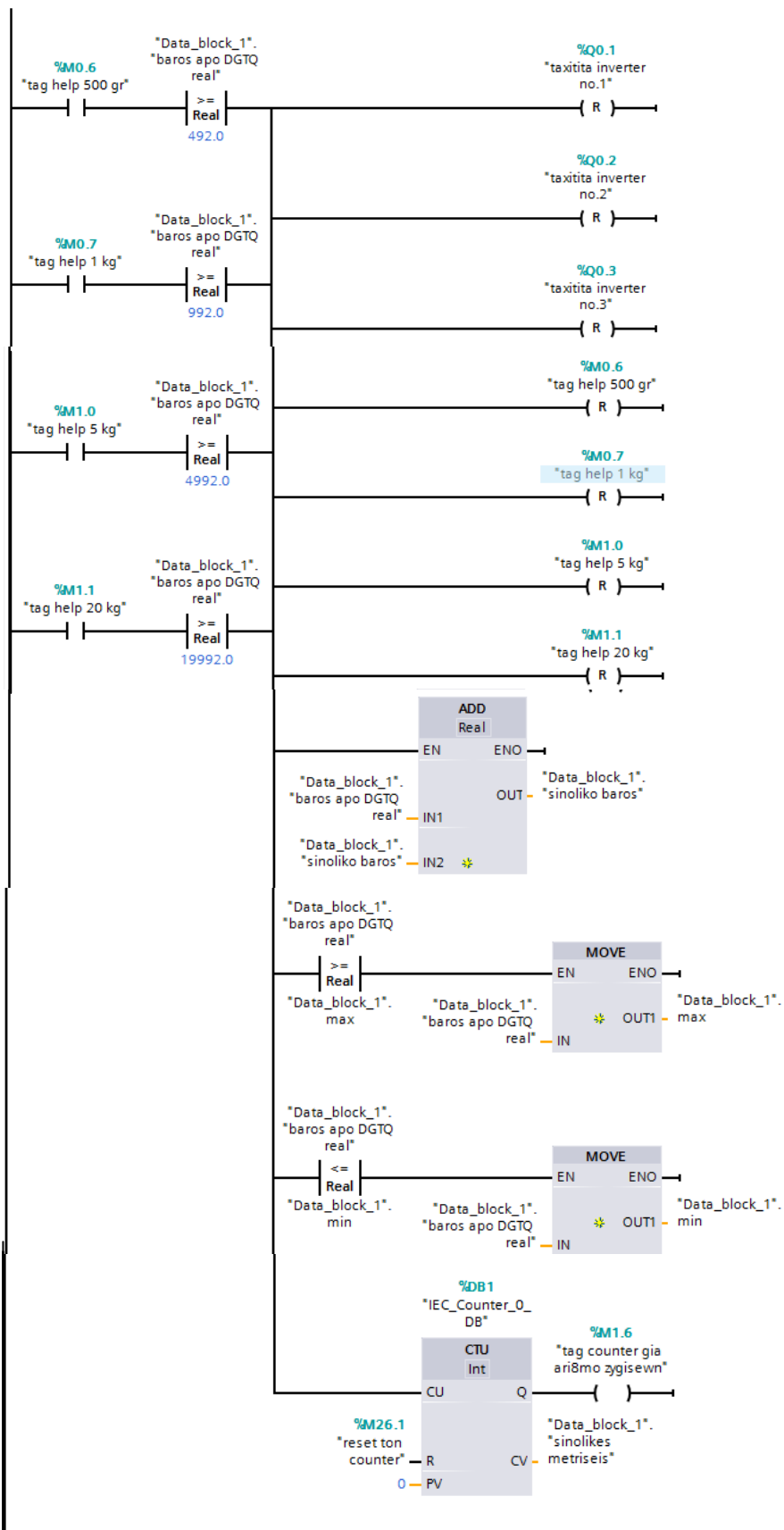
Εικόνα 5.37: Network 5

Network 6: Διακοπή λειτουργίας της αντλίας και υπολογισμός στατιστικών στοιχείων

Σε αυτό το network μόλις η μηχανή είναι στο σωστό στόχο-βάρος κάνει reset όλες τις εξόδους προς το inverter, με αποτέλεσμα η συχνότητα του inverter να πέφτει στα 0 Hz και να σταματάει η λειτουργία της αντλίας. Σε αυτό το σημείο ολοκληρώνεται η πλήρωση ενός δοχείου. Φαίνεται ότι η διακοπή της αντλίας δε γίνεται ακριβώς στο στόχο αλλά 8 gr λιγότερο λόγω του ότι υπάρχουν κάποιες σταγόνες στον αέρα που δεν ζυγίζονται εκείνη τη στιγμή. Στη συνέχεια με το τέλος της γέμισης του δοχείου γίνεται reset η αντίστοιχη βοηθητική μεταβλητή η οποία είχε γίνει set από το 3^ο network και έδειχνε ότι είμαστε σε διαδικασία πλήρωσης. Μόλις σταματήσει το inverter, δηλαδή γεμίσει το δοχείο έχουμε και το τελικό βάρος του, που είναι πολύ χρήσιμο για τις εκτυπώσεις. Στη συνέχεια το βάρος του δοχείου προστίθεται στη μεταβλητή «sinoliko baros», ώστε στο τέλος να υπολογιστεί πόσα κιλά συσκευάστηκαν συνολικά. Επιπλέον συγκρίνεται με το max και αν είναι μεγαλύτερο από το max εκχωρείται στο max, δηλαδή μέχρι εκείνη τη στιγμή ποιο δοχείο ήταν με τη περισσότερη ποσότητα υγρού. Το ίδιο συμβαίνει και με το min. Συγκρίνεται το τελικό βάρος κάθε φορά με τη τιμή που έχει το min και αν είναι μικρότερη από αυτή αποθηκεύεται σε αυτή, δηλαδή μέχρι εκείνη τη στιγμή είναι το ελαφρύτερο δοχείο. Τέλος αυξάνεται ένα counter up μετά από το τέλος κάθε γέμισης κατά ένα, ώστε να υπολογιστεί στο τέλος πόσες γεμίσεις πραγματοποιήθηκαν συνολικά.

Σε περίπτωση που τοποθετήσει ο χρήστης κάποιο άλλο δοχείο πάνω στη πλατφόρμα ζύγισης και πιάσει Start, θα επαναληφθεί η διαδικασία των networks 3,4,5 και 6 και θα ολοκληρωθεί η πλήρωση του επόμενου δοχείου.

Comment

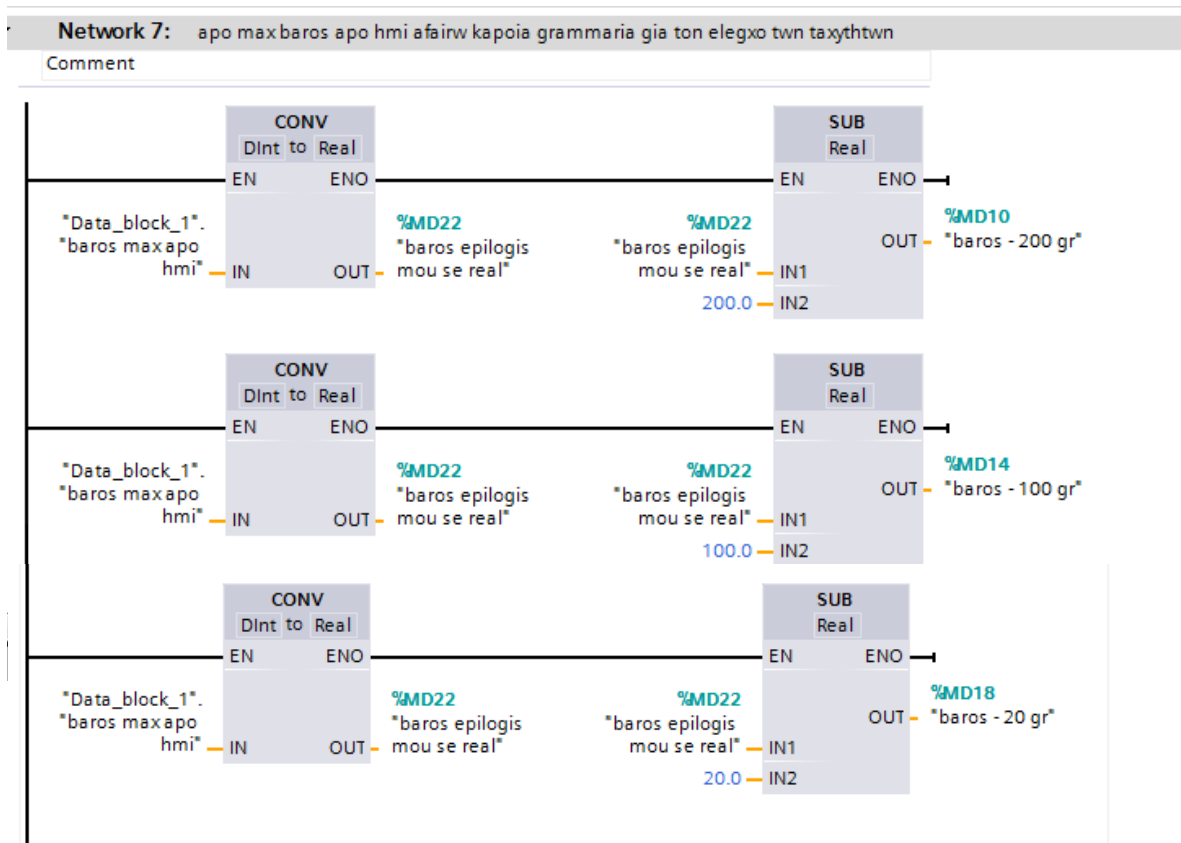


Εικόνα 5.38: Network 6

Network 7 : Δημιουργία στόχων σε βάρος επιλογής μου

Τα networks 7,8,9 και 10 σχετίζονται με τη γέμιση δοχείων στην περίπτωση που ο χρήστης έχει επιλέξει κάποιο δικό του βάρος εκτός των τυποποιημένων.

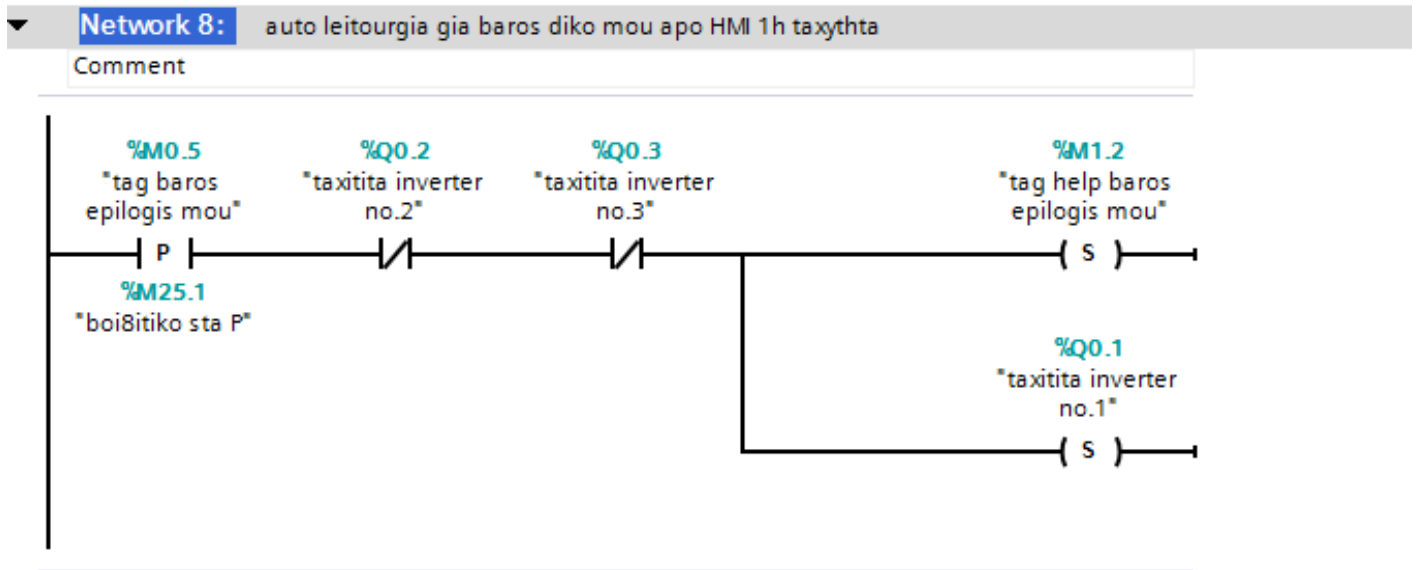
Στο network 7 από την μεταβλητή «baros max apo hmi», που είναι αποθηκευμένη στο data block_1 και περιέχει την τιμή που έδωσε ο χρήστης στην οθόνη με το I/O field, αφαιρούνται κάποια γραμμάκια ώστε να δημιουργηθούν οι στόχοι στους οποίους θα αλλάζει συχνότητα το inverter και επομένως η αντλία στροφές. Παρατηρείται ότι για να γίνει η αριθμητική πράξη της αφαίρεσης απαιτείται πρώτα από όλα να μετατραπεί η μεταβλητή σε real. Στη συνέχεια φαίνεται ότι από την πρώτη ταχύτητα αλλάζει στη δεύτερη πάντα 200gr κάτω από τον στόχο, από την δεύτερη στη τρίτη πάντα 100gr κάτω από το στόχο και από την τρίτη σταματάμε την αντλία πάντα 20 gr κάτω από το στόχο.



Εικόνα 5.39: Network 7

Network 8 : Σε βάρος επιλογής μου ξεκινάει η γεμιση με τη πρώτη ταχύτητα

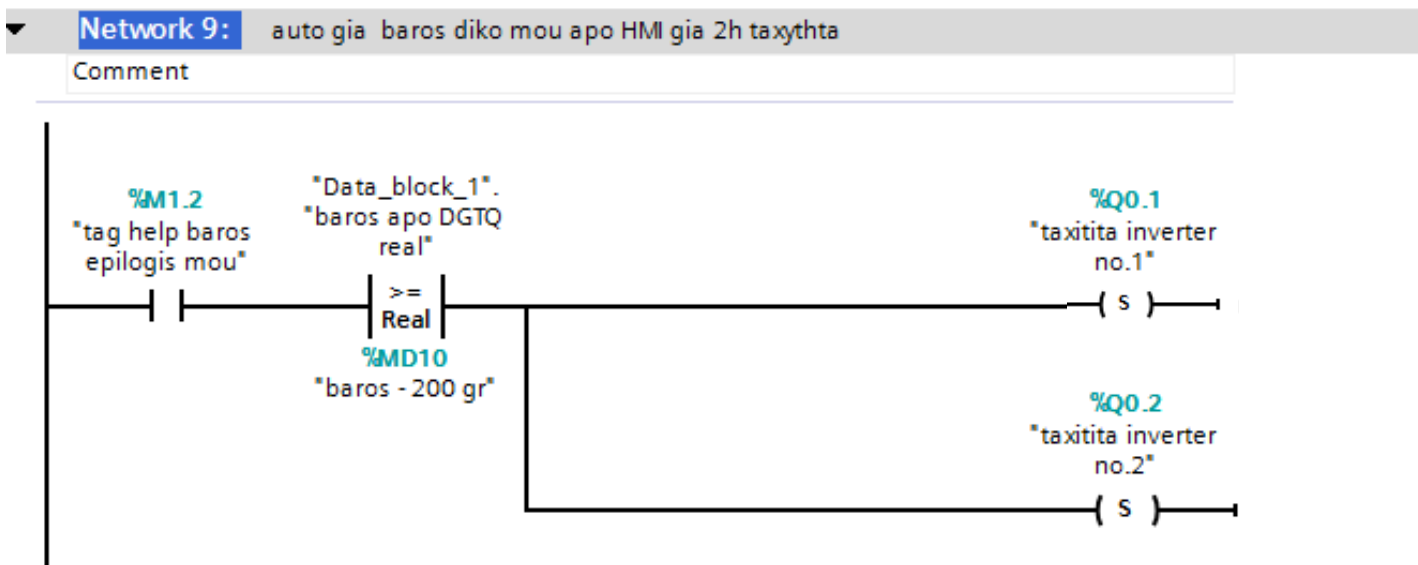
Σε αυτό το network η διαδικασία είναι ίδια με αυτή των τυποποιημένων δοχείων. Αυτό σημαίνει ότι η μηχανή ξεκινάει να ρίχνει υγρό μόλις πατηθεί το start από την 8^η οθόνη (εικόνα 5.25), με την προϋπόθεση ότι πριν το πάτημα του button η αντλία δεν δουλεύει, δηλαδή δεν βρίσκεται ήδη σε διαδικασία γέμισης. Επίσης κάνει set την βοηθητική μνήμη «tag help baros epilogis mou» με την οποία θα φαίνεται ότι βρίσκεται σε διαδικασία γέμισης με βάση το βάρος επιλογής μου.



Εικόνα 5.40: Network 8

Network 9 : Σε βάρος επιλογής μου ενεργοποίηση 2^{ης} ταχύτητας πλήρωσης

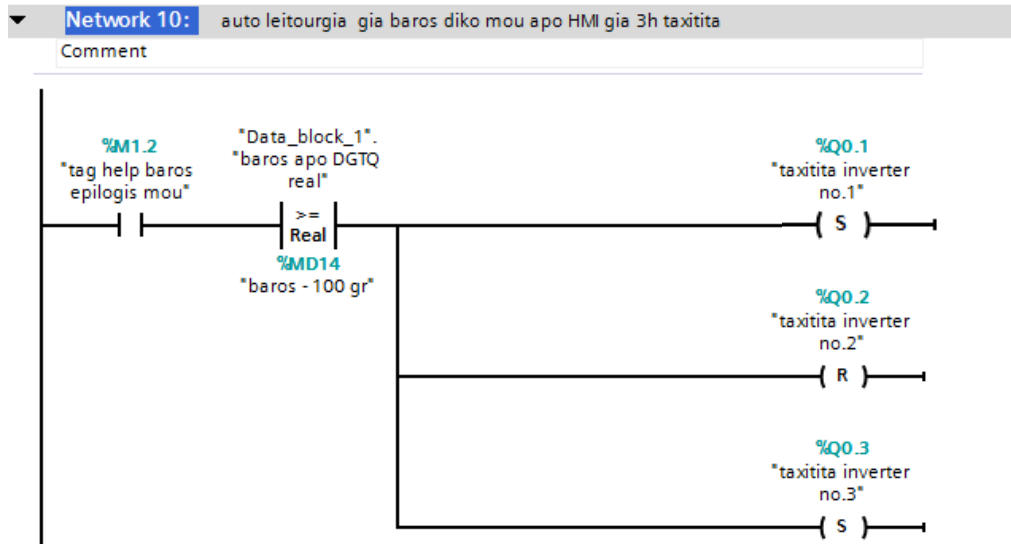
Στο 9^ο network ενεργοποιείται η 2^η ταχύτητα της αντλίας μόλις το βάρος φθάσει 200gr πριν το βάρος-στόχο. Αυτό γίνεται κάνοντας λογικό 1 την Q0.2 και την Q0.1 κάνοντας το inverter να λειτουργήσει με την 2^η συχνότητα.



Εικόνα 5.41: Network 9

Network 10: Σε βάρος επιλογής μου ενεργοποίηση 3^{ης} ταχύτητας πλήρωσης

Ακριβώς η ίδια λειτουργία συμβαίνει και σε αυτό το network. Η μόνη αλλαγή είναι ότι σε μεγαλύτερο βάρος-στόχο ενεργοποιείται η τρίτη ταχύτητα που είναι και η πιο αργή. Η αλλαγή από την δεύτερη στη τρίτη ταχύτητα για κάθε δοχείο συμβαίνει 100gr από το τελικό στόχο κάνοντας λογικό 1 μέσω της εντολής set τις Q01 και Q0.3 και reset τη Q0.2.



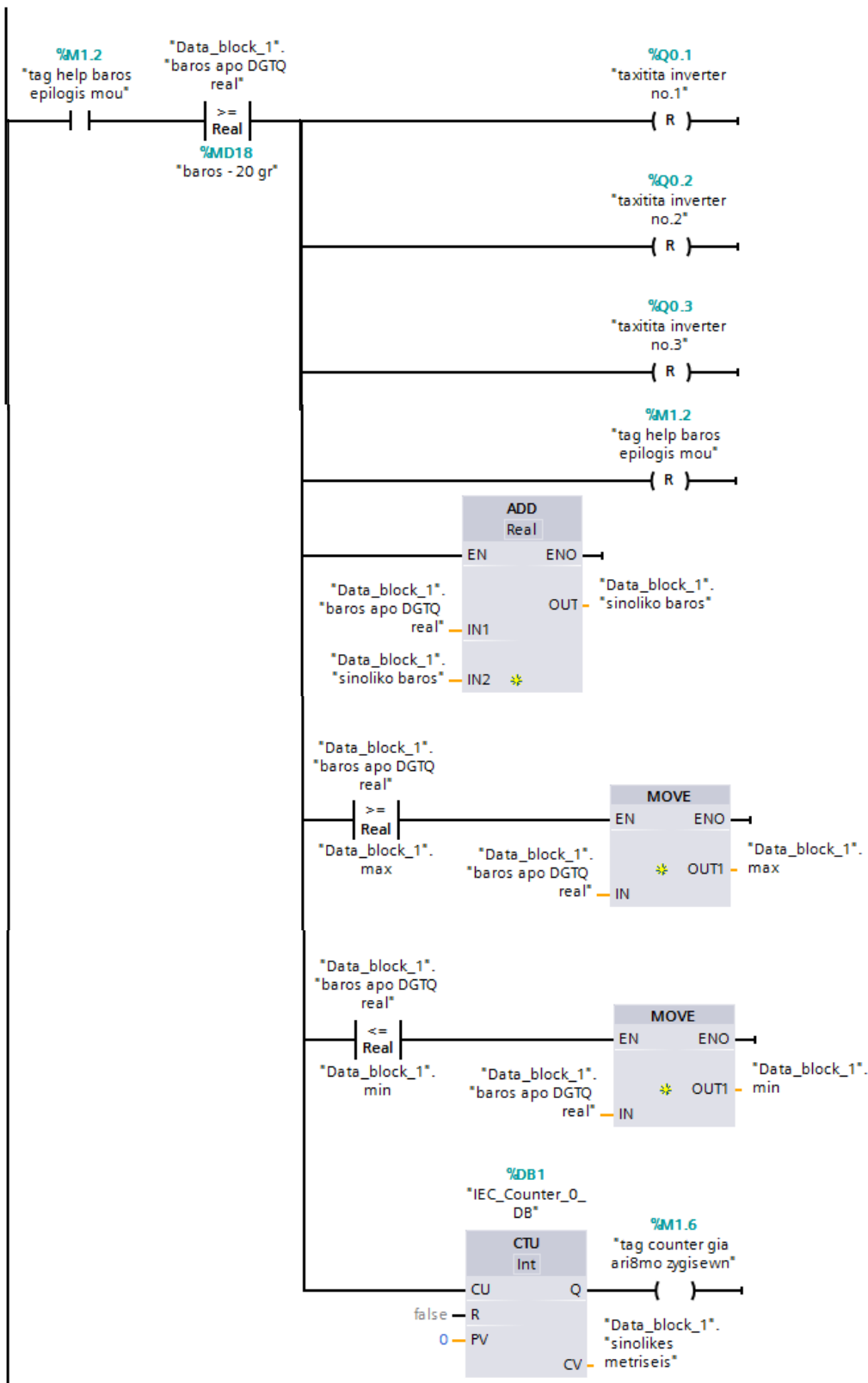
Εικόνα 5.42: Network 10

Network 11: Κλείσιμο της αντλίας και υπολογισμός κάποιων στατιστικών στοιχείων σε βάρος επιλογής μου

Σε αυτό το network μόλις η μηχανή είναι στο σωστό βάρος-στοχο κλείνει την αντλία για να μην φέρει άλλο υγρό (εικόνα 5.43). Με αυτό τον τρόπο πραγματοποιείται η επίτευξη του στόχου. Αυτό συμβαίνει κάνοντας reset όλες τις εξόδους προς το inverter, με αποτέλεσμα να μην δουλεύει καθόλου η αντλία. Επίσης με το τέλος της γέμισης του δοχείου γίνεται reset η βοηθητική μεταβλητή M1.2, η οποία είχε γίνει set από το network 8 και έδειχνε ότι είμαστε σε διαδικασία πλήρωσης. Παράλληλα μόλις σταματήσει το inverter, δηλαδή γεμίσει το δοχείο έχουμε το τελικό βάρος του, που είναι πολύ χρήσιμο για τις εκτυπώσεις. Το τελικό βάρος του δοχείου προστίθεται στη μεταβλητή «sinoliko baros», ώστε στο τέλος να προκύψει πόσα κιλά συσκευάστηκαν συνολικά. Στη συνέχεια συγκρίνεται με το max και αν είναι μεγαλύτερο από το max τότε εκχωρείται σε αυτό. Το αντίστοιχο συμβαίνει και με το min. Συγκρίνεται το βάρος κάθε φορά με την τιμή που έχει το min και αν είναι μικρότερη από αυτή αποθηκεύεται σε αυτή. Τέλος αυξάνεται ένα counter up μετά από το τέλος κάθε γέμισης κατά ένα, ώστε να φανεί στο τέλος πόσες γεμίσεις πραγματοποιήθηκαν.

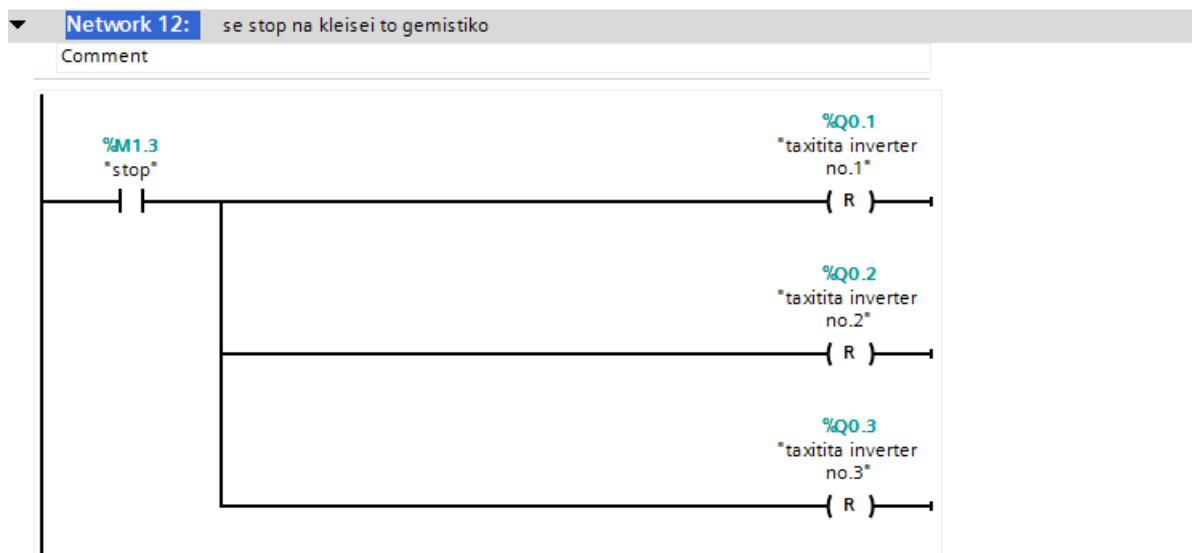
Network 11: auto leitourgia gia baros diko mou apo HMI kleiseimo moter

Comment



Network 12: Stop

Σε κάθε οθόνη πλήρωσης της HMI υπήρχε και ένα κουμπί Stop. Αυτό όποτε πατηθεί απενεργοποιεί τις εξόδους του PLC προς τον inverter, άρα και την αντλία και αποφεύγεται οποιαδήποτε ανεπιθύμητη γέμιση.



Εικόνα 5.44: Network 12

Network 13: Επιστροφή από κάποια συνταγή στην οθόνη 3 της HMI και υπολογισμός των στατιστικών στοιχείων

Σε κάθε οθόνη γέμισης υπήρχε ένα button «Back». Μόλις πατηθεί οδηγεί το χρήστη στην οθόνη επιλογής συνταγής. Ταυτόχρονα μόλις πατηθεί το back, το πρόγραμμα υπολογίζει όλα τα στατιστικά στοιχεία που επρόκειτο να σταλούν προς εκτύπωση (εικόνα 5.45). Ο εκτυπωτής για να εκτυπώσει πρέπει να του σταλεί ένα string. Για αυτό το λόγο σε αυτό το network αρχικά υπολογίζεται κάθε στοιχείο με την αριθμητική του τιμή και στη συνέχεια μετατρέπεται σε string με χρήση κατάλληλων εντολών.

Αρχικά υπολογίζεται ο μέσος όρος. Για να γίνει αυτός ο υπολογισμός πρέπει να γίνει η διαίρεση του συνολικού βάρους που συσκευάστηκε με τον αριθμό των γεμίσεων. Για να συμβεί αυτό πρέπει να έχουν και οι δύο αριθμοί τον ίδιο data type, για αυτό αρχικά μετατρέπονται οι «sinolikes metriseis» από integer σε real και στη συνέχεια πραγματοποιείται διαίρεση DIV του «sinoliko baros» με το «sinolikes metriseis boiðitiko real». Το αποτέλεσμα αυτού αποθηκεύεται στο «avg». Αυτό τώρα είναι ένας αριθμός τύπου real. Για να σταλεί προς εκτύπωση πρέπει να είναι string και για αυτό το λόγο χρησιμοποιείται εντολή μετατροπής

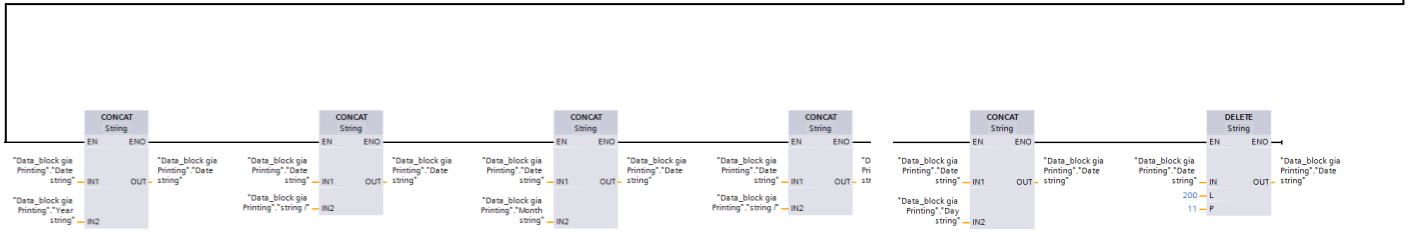
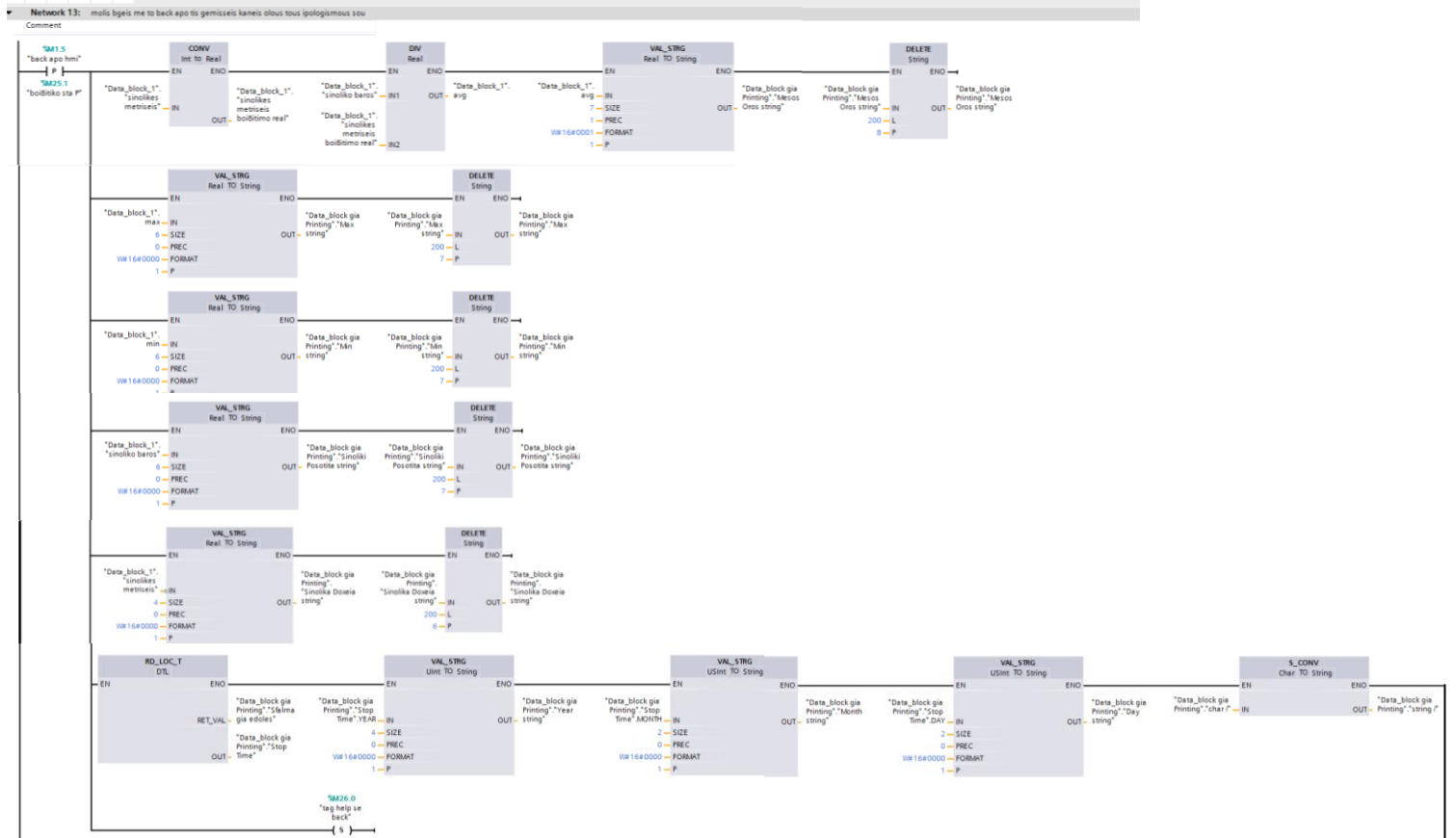
από val σε strg και αποθηκεύεται στο «mesos oros string». Αυτό σαν αριθμός θα έχει περίπου 7 ψηφία αλλά τώρα που έχει μετατραπεί σε string έχει μέγεθος 254 χαρακτήρων (μέγεθος ενός string). Οι περισσότεροι χαρακτήρες είναι κενοί αφού ο αριθμός έχει μόνο 7 ψηφία. Για αυτό το λόγο με την εντολή delete διατηρούνται μόνο αυτοί που ενδιαφέρουν και διαγράφονται όλοι οι υπόλοιποι.

Το ίδιο ακριβώς γίνεται και με όλες τις άλλες μεταβλητές, δηλαδή τις max, min, sinoliko baros και sinolikes metriseis. Αρχικά μετατρέπονται από το data type, που είναι δηλωμένες στο πρόγραμμα, σε string μέσω της εντολής «VAL_STRG» και στη συνέχεια με την εντολή «DELETE» διαγράφονται οι κενοί χαρακτήρες που δεν χρειάζονται.

Στο τέλος του network με την εντολή «RD_LOC_T» αποθηκεύεται στην μεταβλητή Stop Time σε μορφή DTL η ημερομηνία και η ώρα. Βέβαια θα γίνει χρήση μόνο της ημερομηνίας. Για να γίνει εκτύπωση της ημερομηνίας πρέπει να μετατραπεί σε μορφή string το έτος, ο μήνας και η ημέρα, χρησιμοποιώντας 3 διαφορετικές εντολές «VAL_STRG». Στη συνέχεια με χρήση των αντίστοιχων εντολών «DELETE» θα διατηρηθούν οι χαρακτήρες που ενδιαφέρουν και μόνο.

Μόλις μετατραπεί σε string η κάθε πληροφορία, τότε βήμα βήμα γίνεται η συνένωση των strings σε ένα ενιαίο, που εν τέλη περιέχει όλη την ημερομηνία. Αυτό γίνεται με την εντολή «CONCAT» η οποία παίρνει τα strings που περιέχουν τα IN1 και IN2 και τα βάζει και τα δύο σε ένα ενιαίο string, με πρώτο το IN1 και στη συνέχεια το IN2. Το αποτέλεσμα εκχωρείται στο OUT. Έτσι σε ένα string που λέγεται «Date string» γίνεται concat 5 διαφορετικών strings που όλα μαζί σε σειρά απεικονίζουν την ημερομηνία. Αυτά τα strings είναι τα «year», «month», «day» και 2 φορές τα διαχωριστικά ανάμεσά τους «/» , δηλαδή της μορφής YY/MM/DD.

Τέλος γίνεται set η μεταβλητή «tag help se back», η χρήση της οποίας θα φανεί σε παρακάτω network.



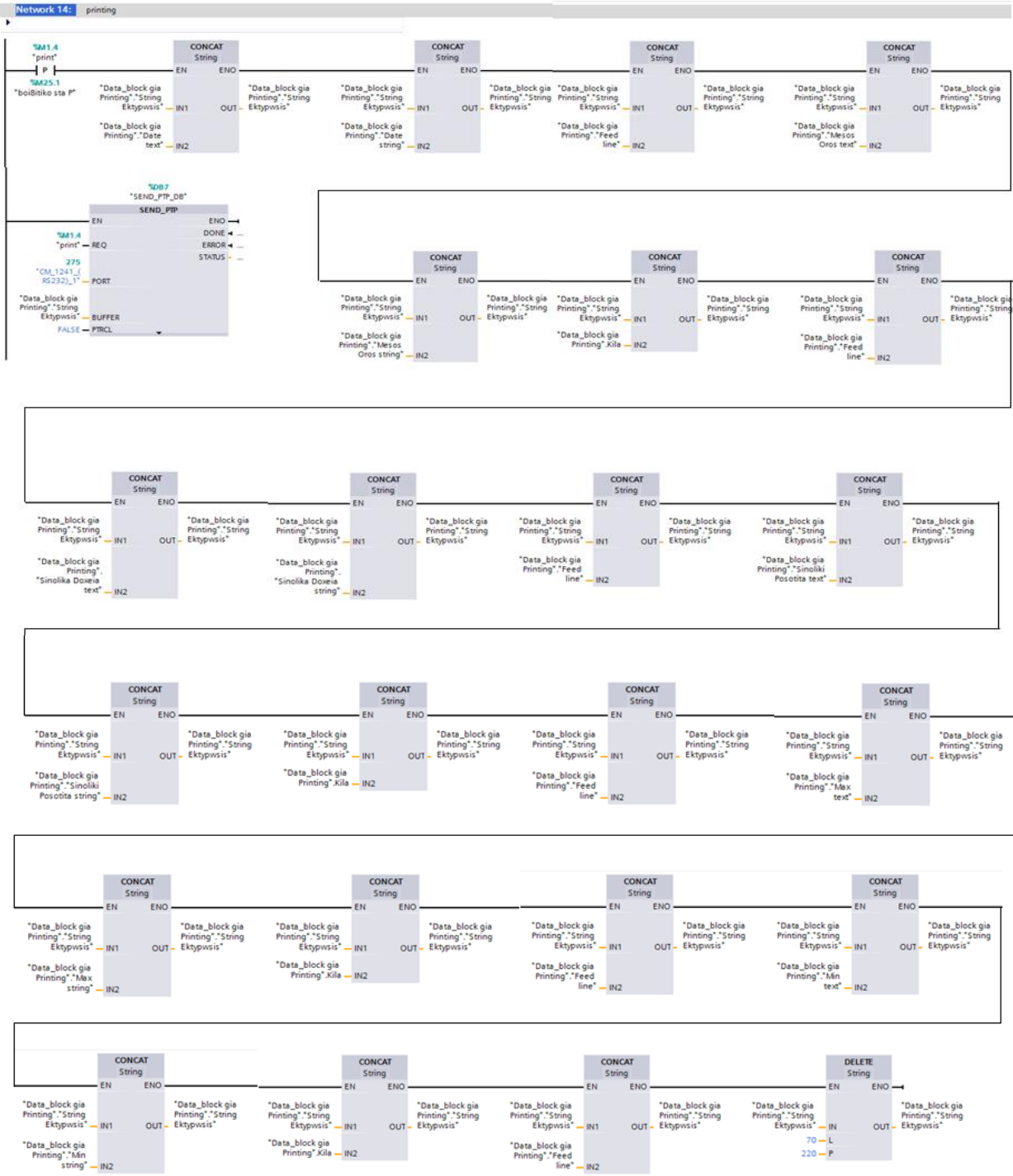
Εικόνα 5.45: Network 13

Network 14: Printing

Σε αυτό το network συμβαίνουν δύο πράγματα. Αρχικά υπολογίζεται το string που θα σταλεί στον εκτυπωτή για εκτύπωση και στη συνέχεια πραγματοποιείται η αποστολή του.

Ο υπολογισμός του ενός string είναι παρόμοιος με τον υπολογισμό της ημερομηνίας. Δηλαδή με την εντολή Concat λαμβάνονται κομμάτια εκτύπωσης ξεχωριστά και γίνονται ένα string μέχρι να συμπληρωθεί αυτό που επιθυμείται. Πιο αναλυτικά λαμβάνεται το «Date text» και εκχωρείται δίπλα στο αρχικά άδειο «String Ektyrwsis» και το αποτέλεσμα μπαίνει στο «String Ektyrwsis». Στη συνέχεια λαμβάνεται το «String Ektyrwsis» και τοποθετείται δίπλα του το «Date String» και όλο μαζί εκχωρείται στο «String Ektyrwsis». Μετά παίρνεται ξανά το «String Ektyrwsis» και εκχωρείται δίπλα του το «Feed line» με το οποίο πηγαίνει ο κέρσοντας στην από κάτω γραμμή και το αποτέλεσμα τοποθετείται ξανά στο «String Ektyrwsis». Με αυτή τη διαδικασία να επαναλαμβάνεται δημιουργείται το επιθυμητό string εκτύπωσης. Στην ονομασία των παραπάνω strings παρατηρείται ότι τα μισά περίπου περιέχουν τη λέξη «text» και τα άλλα μισά τη λέξη «string». Αυτός ο διαχωρισμός είναι προσωπικός και δηλώνει ότι αυτά που είναι text δεν αλλάζουν καθόλου από εκτύπωση σε εκτύπωση, ενώ αυτά που είναι string είναι αυτά κάθε φορά υπολογίζει το πρόγραμμα μετατρέποντάς τα από αριθμούς σε strings.

Τέλος με την εντολή «SEND_PTP_DB» στέλνεται στον εκτυπωτή τι θα εκτυπωθεί. Πιο αναλυτικά στο REQ δηλώνεται πότε θα ενεργοποιηθεί αυτή η εντολή. Στη συγκεκριμένη περίπτωση μόλις γίνει λογικό 1 το «print», δηλαδή μόλις πατηθεί το «Print» στην HMI. Στη συνέχεια δηλώνεται από ποια port θα σταλεί το σήμα εκτύπωσης. Στη συγκεκριμένη περίπτωση δηλώνεται το module του RS232, από το οποίο φεύγει καλώδιο και καταλήγει στον εκτυπωτή. Τέλος δηλώνεται τι θα σταλεί για εκτύπωση, δηλαδή το «String Ektyrwsis» στο οποίο προηγουμένως είχε αποθηκευτεί όλη η χρήσιμη πληροφορία με τον τρόπο που περιγράφηκε .

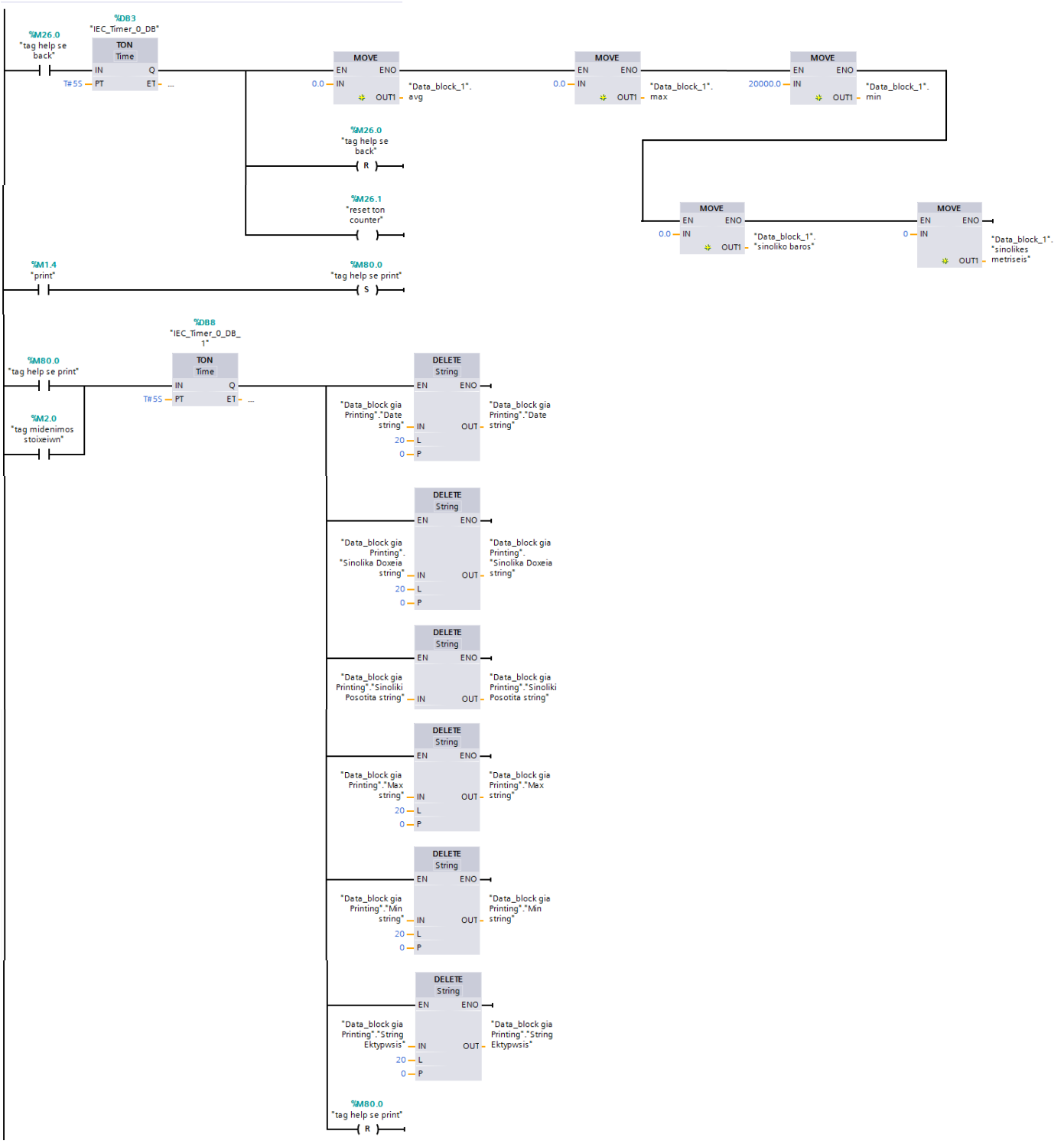


Εικόνα 5.46: Network 14

Network 15: Αρχικοποίηση των μεταβλητών και μηδενισμός των strings εκτύπωσης

Σε αυτό το network συμβαίνουν τρία πράγματα. Αρχικά μηδενίζονται οι τιμές των `avg`, `max`, `sinoliko baros`, `sinolikes metriseis`, του counter που υπολογίζει τις μετρήσεις και τοποθετείται στο `min` μια μεγάλη τιμή για να γίνει η 1^η σύγκριση χωρίς πρόβλημα. Αυτό συμβαίνει ώστε όλες οι μεταβλητές να είναι έτοιμες προς υπολογισμό μόλις ο χρήστης επιλέξει κάποια άλλη συνταγή. Στη συγκεκριμένη περίπτωση υπάρχει ένα χρονικό που προσδίδει μία καθυστέρηση 5 δευτερολέπτων από τη στιγμή που θα πατήσει `back` ο χρήστης. Το `back` πραγματοποιείται μέσω της «`tag help se back`».

Στη συνέχεια μόλις πατηθεί το `print`, γίνεται `set` μια άλλη βοηθητική μεταβλητή, η «`tag help se print`». Αυτή χρησιμοποιείται με παρόμοιο τρόπο με την «`tag help se back`», με τη διαφορά ότι τώρα μηδενίζει τα `strings` ώστε να είναι άδεια για την επόμενη εκτύπωση. Αυτό που φαίνεται εδώ είναι ότι το άδειασμα των `strings` δεν συμβαίνει μόνο όταν πατηθεί το `print` αλλά και όταν γίνει λογικό 1 η «`tag midenismos stoxeiwn`». Αυτή η μεταβλητή συνδέεται με την μεταβλητή «`tag midenismos stoxeiwn HMI`» από το `tag table` της HMI. Η «`tag midenismos stoxeiwn HMI`» ενεργοποιείται όποτε πατηθεί η επιλογή κάποιας συνταγής. Αυτό συμβαίνει για την περίπτωση που ο χρήστης έκανε κάποιες γεμίσεις με κάποια συνταγή, δεν επιθυμούσε να κάνει εκτύπωση των στοιχείων και στη συνέχεια επέλεξε να συνεχίσει με κάποια άλλη συνταγή. Σε αυτή την περίπτωση μηδενίζονται τα στοιχεία της προηγούμενης συνταγής και ετοιμάζεται το πρόγραμμα να δεχτεί τα καινούργια.



Εικόνα 5.47: Network 15

Κεφάλαιο 6: Συμπεράσματα – Παρατηρήσεις

Στο προηγούμενο κεφάλαιο έγινε αναλυτική περιγραφή του τρόπου λειτουργίας της συγκεκριμένης γεμιστικής μηχανής. Πράγματα που θα μπορούσαν να είναι διαφορετικά στο πρόγραμμα, είναι τα σημεία του βάρους στα οποία πραγματοποιήθηκαν οι αλλαγές των συχνοτήτων του inverter. Αυτό εξαρτάται από την ακρίβεια ή την ταχύτητα που θέλει να επιτευχθεί από τον πελάτη. Στη συγκεκριμένη περίπτωση οι αλλαγές πραγματοποιήθηκαν περίπου 20gr από τους αντίστοιχους στόχους. Βέβαια στόχος της συγκεκριμένης διπλωματικής εργασίας ήταν η υλοποίηση του αυτοματισμού και όχι η ακριβέστερη και ταχύτερη γέμιση.

Το συγκεκριμένο γεμιστικό δεν μπορεί εύκολα να προωθηθεί στην ελληνική αγορά λόγω του ότι για τον αυτοματισμό του έγινε χρήση του PLC, κάτι που σημαίνει αρκετά υψηλότερο κοστολόγιο και εν συνεχεία εκτός ανταγωνισμού. Η λειτουργία της γεμιστικής μηχανής θα μπορούσε να πραγματοποιηθεί και με χρήση ρελέ (χωρίς τις εκτυπώσεις και τη χρήση HMI). Παρόλα αυτά η συγκεκριμένη γεμιστική μηχανή περιλαμβάνει αρκετές « ανέσεις » για το χρήστη, όπως για παράδειγμα HMI, με την οποία γίνεται πιο αναλυτική, άμεση και εύκολη η επαφή του χρήστη με αυτή. Σαφώς για πλήρεις αυτόματες γραμμές συσκευασίας χρησιμοποιούνται και PLC και HMI, με πολύ πιο πολύπλοκη λογική και κώδικα.

Ένα θέμα που θα μπορούσε να μελετηθεί εν συνεχεία με βάση τη παρούσα διπλωματική είναι το remote control του PLC. Αυτό σημαίνει ότι παρέχεται η δυνατότητα ελέγχου του PLC από απομακρυσμένο χώρο. Κάτι τέτοιο παρέχει πολλά πλεονεκτήματα όπως μειωμένο κόστος, διότι δεν χρειάζεται να πραγματοποιηθεί επίσκεψη στο χώρο που βρίσκεται η γεμιστική μηχανή. Από το χώρο σου μπορείς να δεις κάποια πιθανή αιτία βλάβης και αν δεν είναι κάτι πολύ σημαντικό να λυθεί χωρίς επίσκεψη.

Κλείνοντας αυτή την διπλωματική εργασία θα ήθελα να τονίσω ότι μπόρεσα και αποκόμισα κάποιες βασικές γνώσεις γύρω από αρκετά θέματα που αφορούν τους βιομηχανικούς αυτοματισμούς, όπως προγραμματισμός ενός PLC, προγραμματισμός μιας HMI, ανταλλαγή δεδομένων μέσω Profibus καθώς και εκτύπωση από PLC μέσω RS232. Αυτό το οφείλω στον κύριο Κορρέ που μου έδωσε την ευκαιρία μέσα από αυτή την διπλωματική εργασία να ασχοληθώ με τα παραπάνω θέματα αλλά και στον πατέρα μου και το θείο μου που με βοήθησαν σε γνωστικό αλλά και σε οικονομικό επίπεδο.

Βιβλιογραφία

- 1] Ζούλης Νικόλαος, Καφφετζάκης Παναγιώτης, Σούλης Γεώργιος. 2000. Συστήματα Αυτοματισμών. Β΄ Τόμος
- 2] Κορρές Γεώργιος. 2008. ΕΠΟΠΤΕΙΑ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΕΝΕΡΓΕΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
- 3] Μαραντίδης Νικόλαος. Αυτοματισμοί με Simatic S7. Siemens A.E. 2000
- 4] Πανταζής Α. Νικόλαος. Αυτοματισμοί με PLC
- 5] Ρουμπή Σταύρου. Αυτοματισμός με προγραμματιζόμενους λογικούς ελεγκτές. Siemens A.E.
- 6] Siemens. AUTOMATION TRAINING. Εκπαίδευση Βιομηχανικών Αυτοματισμών
- 7] Siemens. Simatic S7-1200 Easy Book Manual. 04/2012. A5E02486774-05
- 8] Siemens. SIMATIC STEP 7 in the TIA Portal
- 9] Christopher E. Strangio. The RS232 STANDARD
(http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html#anchor367782)
- 10] International Society of Automation. BASICS OF PROFIBUS OPERATION
(http://www.isa.org/Template.cfm?Section=Books1&template=Ecommerce/FileDisplay.cfm&ProductID=6959&file=Chapter1_Profibus.pdf)
- 11] Unitronics. What is an HMI.
(http://www.unitronics.com/KnowledgeBase/U90Ladder/HMI/Display/What_is_an_HMI_.htm)
- 12] ΣΑΕ, PLC και Ηλεκτρονόμοι. Αθήνα. 15 Απριλίου 2005
(http://users.ntua.gr/el09791/plc_hlektronomoi.pdf)