

Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός συστήματος μέτρησης περιβαλλοντικών συνθηκών με χρήση της μικροελεγκτικής πλακέτας Arduino και κατάλληλων αισθητήρων. Πιο συγκεκριμένα, με τη χρησιμοποίηση αισθητήρων πίεσης, θερμοκρασίας και υγρασίας, υπάρχει η δυνατότητα παρακολούθησης των τιμών αυτών καθώς και αποθήκευσής τους για περαιτέρω ανάλυση. Η όλη διαδικασία γίνεται με το Arduino να διαβάζει τις τιμές των περιβαλλοντικών συνθηκών από τους αισθητήρες και να στέλνει τα δεδομένα στο WinCC_OA μέσω ενός OPC Server ώστε να γίνει η επεξεργασία τους. Το project αυτό ενσωματώθηκε στο ήδη υπάρχον SLOW Control System (SLOCSY) του RD51 Collaboration του CERN. Η δυνατότητα παρακολούθησης των περιβαλλοντικών παραμέτρων κατά την διάρκεια των test beams αποτελεί σημαντική προσθήκη στις ήδη υπάρχουσες λειτουργίες, όπως την παρακολούθηση των τάσεων των ανιχνευτών.

Abstract

The goal of this thesis is the development of a system for the measurement of the environmental conditions. This system is based on the use of the appropriate sensors and the Arduino microcontroller. More precisely, the Arduino is used to read the environmental conditions via the sensors and send the data to a WinCC_OA project responsible for monitoring via an OPC Server for further elaboration. This WinCC_OA project was then integrated in the RD51 Slow Control System. The development of this system as well as the WinCC_OA project integration, is very important because it gives the user of the Slow Control System the ability of monitoring the environmental conditions at which the experiment is being carried out. Given that environmental conditions may be critical to detector response, such a provision can be considered as a substantial contribution to data collected during test beams.

Περιεχόμενα

Περίληψη	2
Abstract	4
Κατάλογος Σχημάτων	7
Κατάλογος Πινάκων	9
1 Μικροϋπολογιστική πλακέτα Arduino, Αισθητήρες, OPC Server	11
1.1 Μικροϋπολογιστική πλακέτα Arduino	11
1.1.1 Εισαγωγή	11
1.1.2 Arduino Mega 2560	12
1.1.3 Arduino Ethernet Shield	17
1.1.4 Αλληλεπίδραση με το Arduino	18
1.2 Αισθητήρες	21
1.2.1 Αισθητήρας MPX5700AP	21
1.2.2 Αισθητήρας BMP085	28
1.2.3 Αισθητήρας MPL3115A2	38
1.2.4 Αισθητήρας DHT22	43
1.2.5 Κοινή σύνδεση όλων των αισθητήρων	48
1.2.6 Σύγκριση των αισθητήρων με το HMP233 της Vaisala	50
1.2.7 Τελικές μετρήσεις	53
1.3 OPC Server	55
1.3.1 Εισαγωγή	55
1.3.2 Arduino OPC Server	56
1.3.3 Εγκατάσταση και χρήση του Arduino OPC Server	57
2 Ανάπτυξη Συστήματος με WinCC_OA	61
2.1 WinCC_OA	61
2.1.1 Εισαγωγή	61
2.1.2 Βασικά Χαρακτηριστικά του WinCC_OA	62
2.1.3 JCOP Framework	63
2.2 Slow Control System	64
2.3 Οδηγίες Σύνδεσης του Arduino με το WinCC_OA	70
2.3.1 Εισαγωγή (import) project στο WinCC_OA	70
2.3.2 Δημιουργία καινούργιου project στο WinCC_OA	71
2.3.3 Εγκατάσταση JCOP Framework Components	75

2.3.4	Σύνδεση του project με το Arduino	78
3	Συμπεράσματα	83
	Βιβλιογραφία	86

Κατάλογος Σχημάτων

1.1	Μικροϋπολογιστική πλακέτα Arduino	11
1.2	Arduino Mega 2560	12
1.3	ATMega2560 Pin Mapping	16
1.4	Arduino Ethernet Shield	17
1.5	Πλακέτα Arduino και Ethernet Shield	18
1.6	Arduino IDE	19
1.7	Αισθητήρας MPX5700AP	21
1.8	Εσωτερικό του αισθητήρα MPX5700AP	22
1.9	Σχηματικό του αισθητήρα MPX5700AP	23
1.10	Προτεινόμενη από τον κατασκευαστή συνδεσμολογία για τον MPX5700AP	23
1.11	Συνδεσμολογία του αισθητήρα MPX5700AP με το Arduino	24
1.12	Διάγραμμα της πίεσης συναρτήσει της τάσης για το αισθητήρα MPX5700AP	24
1.13	Διάγραμμα πίεσης συναρτήσει του χρόνου για τον αισθητήρα MPX5700 με μετρήσεις που πάρθηκαν με το Arduino σε διάστημα 2 ωρών με βήμα μέτρησης (sampling) 2 δευτερόλεπτα	25
1.14	Μετρήσεις πίεσης συναρτήσει του χρόνου για τον αισθητήρα MPX5700 με μετρήσεις που πάρθηκαν με το Arduino σε διάστημα 2 ωρών με βήμα μέτρησης 2 δευτερόλεπτα και 5 δείγματα ανά μέτρηση	26
1.15	Μετρήσεις πίεσης συναρτήσει του χρόνου για τον αισθητήρα MPX5700 με μετρήσεις που πάρθηκαν με το Arduino σε διάστημα 2 ωρών με βήμα μέτρησης 2 δευτερόλεπτα και 10 δείγματα ανά μέτρηση	26
1.16	Μετρήσεις πίεσης συναρτήσει του χρόνου για τον αισθητήρα MPX5700 με μετρήσεις που πάρθηκαν με το Arduino σε διάστημα 2 ωρών με βήμα μέτρησης 2 δευτερόλεπτα και 100 δείγματα ανά μέτρηση	27
1.17	Διάγραμμα πίεσης συναρτήσει του χρόνου και για τους δύο αισθητήρες MPX5700AP	28
1.18	Ψηφιακός αισθητήρας πίεσης και θερμοκρασίας BMP085	28
1.19	Atmel AVR4201 με τον αισθητήρα BMP085	30
1.20	Συνδεσμολογία διάταξης που χρησιμοποιεί το πρωτόκολλο επικοινωνίας I ² C	31

1.21	Απεικόνιση των γραμμών SDA, SCL κατά τη διάρκεια επικοινωνίας με χρήση του πρωτοκόλλου I ² C	31
1.22	Απεικόνιση των γραμμών SDA, SCL κατά τη διάρκεια επικοινωνίας με χρήση του πρωτοκόλλου I ² C κατά την οποία πραγματοποιείται repeated start	32
1.23	Σκαρίφημα διαδικασίας λήψης δεδομένων από τον αισθητήρα BMP085	33
1.24	Αναλυτική αναπαράσταση της διαδικασίας λήψης δεδομένων από τον αισθητήρα BMP085	34
1.25	Adafruit bidirectional level shifter	35
1.26	Σχηματικό κυκλώματος αισθητήρα BMP085 πάνω στην πλακέτα AVR4201	35
1.27	Διαγράμμα θερμοκρασίας συναρτήσει του χρόνου για τον αισθητήρα BMP085 σε διάστημα 4 ωρών με μία μέτρηση ανά δύο δευτερόλεπτα	36
1.28	Διαγράμμα πίεσης συναρτήσει του χρόνου για τον αισθητήρα BMP085 σε διάστημα 4 ωρών με μία μέτρηση ανά δύο δευτερόλεπτα	37
1.29	Ψηφιακός αισθητήρας πίεσης και θερμοκρασίας MPL3115A2	38
1.30	Xtrinsic sensor board της Element14 με τους αισθητήρες MAG3110, MMA8491, MPL3115A2	38
1.31	Μετρήσεις πίεσης συναρτήσει του χρόνου για τον αισθητήρα MPL3115A2 για διάστημα 4 ωρών με μία μέτρηση ανά 2 δευτερόλεπτα	42
1.32	Μετρήσεις θερμοκρασίας συναρτήσει του χρόνου για τον αισθητήρα MPL3115A2 για διάστημα 4 ωρών με μία μέτρηση ανά 2 δευτερόλεπτα	42
1.33	Ψηφιακός αισθητήρας θερμοκρασίας και υγρασίας DHT22	43
1.34	Σχηματικό του αισθητήρα θερμοκρασίας και υγρασίας DHT22	43
1.35	Συνδεσμολογία που χρησιμοποιείται για τη σύνδεση του αισθητήρα DHT22 με το Arduino	45
1.36	Διαγράμματα θερμοκρασίας συναρτήσει του χρόνου από τη διπλωματική εργασία “Ανάπτυξη Συστήματος μέτρησης Θερμοκρασίας και Υγρασίας βασισμένο σε Arduino και WinCC_OA”	46
1.37	Διαγράμματα υγρασίας συναρτήσει του χρόνου από τη διπλωματική εργασία “Ανάπτυξη Συστήματος μέτρησης Θερμοκρασίας και Υγρασίας βασισμένο σε Arduino και WinCC_OA”	46
1.38	Διάγραμματα θερμοκρασίας συναρτήσει του χρόνου για τον αισθητήρα DHT22 κατά την προσπάθεια αναπαραγωγής του προβλήματος που επισημάνθηκε στην εργασία “Ανάπτυξη Συστήματος μέτρησης Θερμοκρασίας και Υγρασίας βασισμένο σε Arduino και WinCC_OA”	47

1.39	Διάγραμμα υγρασίας συναρτήσει του χρόνου για τον αισθητήρα DHT22 κατά την προσπάθεια αναπαραγωγής του προβλήματος που επισημάνθηκε στην εργασία “Ανάπτυξη Συστήματος μέτρησης Θερμοκρασίας και Υγρασίας βασισμένο σε Arduino και WinCC_OA”	47
1.40	Διάγραμμα θερμοκρασίας συναρτήσει του χρόνου για τους αισθητήρες BMP085, MPL3115A2, DHT22	48
1.41	Διάγραμμα υγρασίας συναρτήσει του χρόνου για τους αισθητήρες DHT22	49
1.42	Διάγραμμα πίεσης συναρτήσει του χρόνου για τους αισθητήρες BMP085, MPL3115A2,MPX5700	49
1.43	Vaisala HMP233	50
1.44	Διάγραμμα θερμοκρασίας συναρτήσει της τάσης για το HMP233 της Vaisala και η συνάρτηση μεταφοράς του . .	51
1.45	Διάγραμμα υγρασίας συναρτήσει της τάσης για το HMP233 της Vaisala και η συνάρτηση μεταφοράς του	52
1.46	Διάγραμμα υγρασίας συναρτήσει του χρόνου για όλους τους αισθητήρες μαζί με το HMP233 της Vaisala	52
1.47	Διάγραμμα θερμοκρασίας συναρτήσει του χρόνου για όλους τους αισθητήρες μαζί με το HMP233 της Vaisala . .	53
1.48	Διάγραμμα θερμοκρασίας συναρτήσει του χρόνου για όλους τους αισθητήρες για διάστημα δύο εβδομάδων	54
1.49	Διάγραμμα υγρασίας συναρτήσει του χρόνου για όλους τους αισθητήρες για διάστημα δύο εβδομάδων	54
1.50	Διάγραμμα πίεσης συναρτήσει του χρόνου για όλους τους αισθητήρες για διάστημα δύο εβδομάδων	55
1.51	Διάγραμμα επικοινωνίας με χρήση του πρωτοκόλλου Open Platform Communications (OPC)	55
1.52	Arduino OPC Server	57
1.53	Εκκίνηση Arduino OPC Server	58
1.54	Κύριο παράθυρο Arduino OPC Server	59
1.55	Arduino OPC Server Configuration	59
2.1	WinCC_OA Data Structure	63
2.2	Κύριο παράθυρο Slow Control Sysyem	65
2.3	Κύριο παράθυρο project μέτρησης περιβαλλοντικών συνθηκών	65
2.4	Παράθυρο επιλογής διαγραμμάτων απεικόνισης περιβαλλοντικών συνθηκών	66
2.5	Σύνολο επιλογών για τα διαγράμματα απεικόνισης περιβαλλοντικών συνθηκών	66
2.6	Διαγράμματα περιβαλλοντικών συνθηκών μετά την επιλογή “All Plots” στο παράθυρο επιλογής διαγραμμάτων	67
2.7	Παράθυρο επιλογής και διαμόρφωσης για το Archiving των τιμών των περιβαλλοντικών συνθηκών	67
2.8	Παράθυρο εξαγωγής δεδομένων περιβαλλοντικών συνθηκών	68

2.9	Κύριο παράθυρο Slow Control System με την προσθήκη του κουμπιού Environmental Parameters	69
2.10	Άνοιγμα του WinCC_OA project administrator	70
2.11	Επιλογή Register New Project στον project administrator του WinCC_OA	71
2.12	Εισαγωγή της περιοχής του υπολογιστή στην οποία είναι αποθηκευμένο το project που θα εισαχθεί στο WinCC_OA	71
2.13	Επιλογή Distributed Project κατά τη διαδικασία δημιουργίας καινούργιου project στο WinCC_OA	72
2.14	Εισαγωγή ονόματος καθώς και System Name κατά τη δημιουργία καινούργιου project στο WinCC_OA	72
2.15	Αλλαγές στο config file του καινούργιου project με σκοπό τη σύνδεσή του project με τον OPC Server	73
2.16	Δημιουργία καινούργιου manager μέσω της επιλογής "append new manager" για τον Arduino OPC Server	74
2.17	Διαδικασία δημιουργίας του DataPoint "_DriverCommon" για τη σύνδεση του Arduino OPC Server με το WinCC_OA	74
2.18	Δήλωση του ArduinoOPCServer στο project στο WinCC_OA	75
2.19	Δημιουργία του Arduino Group για τις τιμές μέτρησης των περιβαλλοντικών συνθηκών	75
2.20	Άνοιγμα fwInstallation Panel με την επιλογή Module → Vision	76
2.21	Επιλογή του φακέλου fwInstallation κατά τη διαδικασία ανοίγματος του fwInstallation Panel	76
2.22	Επιλογή και άνοιγμα του fwInstallation Panel	77
2.23	Παράθυρο fwInstallation για την εγκατάσταση των components του WinCC_OA. Στο δεξί μέρος του panel φαίνονται τα components τα οποία είναι εγκατεστημένα στο project αυτό, τα components αυτά είναι τα fwAccessControl, fwAtlas, fwCaen, fwCore, fwFsmAtlas, fwTrending, fwXML που είναι απαραίτητα για τη λειτουργία του project μέτρησης περιβαλλοντικών συνθηκών	78
2.24	Εισαγωγή του πλήθους των Arduino που θέλουμε να συνδέσουμε στο σύστημα μας	79
2.25	Επιλογή ποιού από τα Arduino που είναι συνδεδεμένα στο σύστημα μας θέλουμε να ρυθμίσουμε	79
2.26	Δήλωση είδους και ποσότητας αισθητήρων που είναι συνδεδεμένοι στο κάθε Arduino	80
2.27	Παράθυρο σύνδεσης με τα Items του ArduinoOPCServer	81
2.28	Κύριο παράθυρο περιβαλλοντικών συνθηκών στο οποίο εμφανίζονται οι τιμές για τους αισθητήρες που δηλώθηκαν με τη διαδικασία που περιγράφεται στην ενότητα αυτή	82
3.1	Επιλογή "All Plots" στο παράθυρο επιλογής διαγραμμάτων του project μέτρησης περιβαλλοντικών συνθηκών κατά τη διάρκεια του Test Beam στις εγκαταστάσεις του RD51 collaboration στο CERN	84

3.2	Διάγραμμα πίεσης συναρτήσει του χρόνου για τον αισθητήρα BMP085 με δεδομένα που λήφθηκαν κατά τη διάρκεια του Test Beam στις εγκαταστάσεις του RD51 collaboration στο CERN	84
3.3	Διάγραμμα νερμοκρασίας συναρτήσει του χρόνου για τους αισθητήρες BMP085, DHT22 με δεδομένα που λήφθηκαν κατά τη διάρκεια του Test Beam στις εγκαταστάσεις του RD51 collaboration στο CERN	85
3.4	Διάγραμμα υγρασίας συναρτήσει του χρόνου για τον αισθητήρα DHT22 με δεδομένα που λήφθηκαν κατά τη διάρκεια του Test Beam στις εγκαταστάσεις του RD51 collaboration στο CERN	85

Κατάλογος Πινάκων

1.1	Χαρακτηριστικά του αισθητήρα MPX5700AP	22
1.2	Χαρακτηριστικά του αισθητήρα BMP085	29
1.3	Αναλυτικός πίνακας των registers του αισθητήρα BMP085 καθώς και ποιιά σταθερά υπάρξει αποθηκευμένη σε ποιό register	33
1.4	Αντιστοιχία των pins του AVR4201 με αυτά του Arduino	36
1.5	Χαρακτηριστικά αισθητήρα θερμοκρασίας MPL3115A2 . .	39
1.6	Χαρακτηριστικά αισθητήρα πίεσης MPL3115A2	39
1.7	Ηλεκτρικά χαρακτηριστικά του αισθητήρα πίεσης και θερ- μοκρασίας MPL3115A2	40
1.8	Xtrinsic Sensor Board CN1 Pins	41
1.9	Xtrinsic Sensor Board CN2 Pins	41
1.10	Αντιστοίχιση των pins Xtrinsic sensor board με αυτά του Arduino	41
1.11	Πίνακας με τα pins του αισθητήρα DHT22	44
1.12	Τεχνικά χαρακτηριστικά του αισθητήρα θερμοκρασίας και υγρασίας DHT22	44
2.1	Αντιστοίχιση των Data Point Elements του WinCC_OA με τα items του server	81

Κεφάλαιο 1

Μικροϋπολογιστική πλακέτα Arduino, Αισθητήρες, OPC Server

Η ενότητα αυτή περιλαμβάνει την γενική παρουσίαση της μικροϋπολογιστικής πλακέτας Arduino [4] , την περιγραφή της πλακέτας Arduino Mega 2560 [2] που χρησιμοποιήθηκε στην παρούσα εργασία, καθώς και βασικές οδηγίες για τη χρήση της πλακέτας Arduino.

1.1 Μικροϋπολογιστική πλακέτα Arduino

1.1.1 Εισαγωγή



Σχήμα 1.1: Μικροϋπολογιστική πλακέτα Arduino

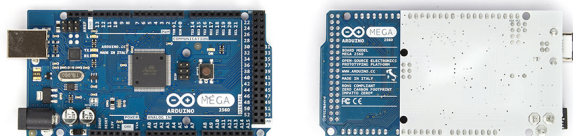
Το Arduino είναι μια μικροϋπολογιστική πλατφόρμα, δηλαδή μία πλακέτα βασισμένη σε έναν μικροελεγκτή, η οποία μπορεί να χρησιμοποιηθεί για την κατασκευή ηλεκτρονικών συστημάτων τα οποία αλληλεπιδρούν με τον πραγματικό κόσμο. Αυτό γίνεται είτε με τη χρήση αισθητήρων, έτσι ώστε το Arduino μπορεί να μετρήσει παραμέτρους α-

πό τον έξω κόσμο, είτε με τη χρήση κινητήρων μέσω των οποίων μπορεί να αλληλεπιδράσει με τον έξω κόσμο.

Η πλακέτα, αποτελείται από έναν μικροελεγκτή και μια πληθώρα επαφών εισόδου-εξόδου (I/O pins). Το είδος του μικροελεγκτή, ο αριθμός, καθώς και η λειτουργία των pins έχουν σαν αποτέλεσμα την ύπαρξη μίας ολόκληρης οικογένειας πλακετών Arduino. Το Arduino μπορεί να αγοραστεί είτε προ-συναρμολογημένο είτε σε κομμάτια και να συναρμολογηθεί από το χρήστη.

Τα pins του Arduino χωρίζονται σε δύο κατηγορίες, αναλογικά και ψηφιακά, και είναι δυνατόν να συνδεθούν σε αυτά πλακέτες επέκτασης ("shields"). Οι πλακέτες αυτές προσδίδουν περαιτέρω δυνατότητες στο Arduino όπως σύνδεση σε κάποιο δίκτυο μέσω Ethernet ή wifi κ.α.

1.1.2 Arduino Mega 2560



(α) Front View

(β) Back View

Σχήμα 1.2: Arduino Mega 2560

Γενικά στοιχεία

Στη συγκεκριμένη εργασία χρησιμοποιήθηκε η πλακέτα Arduino Mega 2560. Η βάση της εν λόγω πλακέτας είναι ο μικροελεγκτής ATmega2560. Το συγκεκριμένο Arduino, διαθέτει

- 54 ψηφιακά pins τα οποία μπορούν να χρησιμοποιηθούν είτε σαν εισοδοι είτε σαν εξοδοι. Τα 15 απο αυτά μπορούν επίσης να χρησιμοποιηθούν σαν εξοδοι διαμόρφωσης πλάτους παλμού (Pulse Width Modulation(PWM)).
- 16 αναλογικά pins
- 4 ασύγχρονες σειριακές θύρες (Universal Asynchronous Receiver/Transmitter, UARTs)
- ένα κρυσταλλικό ταλαντωτή (clock) 16MHz
- μια θύρα USB,
- μία υποδοχή ρεύματος,
- μια επαφή ICSP (In-Circuit Serial Programming)

- ένα κουμπί επαναφοράς (Reset Button)

Παροχή τάσης

Το Arduino μπορεί να τροφοδοτηθεί είτε μέσω θύρας USB είτε μέσω εξωτερικής πηγής (π.χ. μπαταρία ή μετασχηματιστής AC to DC). Για εξωτερική παροχή ενέργειας, το Arduino μπορεί να τροφοδοτηθεί από 6 έως 20 volts. Ωστόσο, το συνιστώμενο από τον κατασκευαστή εύρος είναι 7 έως 12 volts. Το Arduino, διαθέτει pins, τα οποία μπορούν να χρησιμοποιηθούν για την παροχή ρεύματος/τάσης σε άλλα εξωτερικά κυκλώματα.

Πιο συγκεκριμένα, τα pins αυτά είναι :

- **Vin:** Παρέχει τάση ίση με την τάση εισόδου, όταν το Arduino τροφοδοτείται από εξωτερική πηγή τάσης.
- **5V:** Παρέχει τάση ίση με 5V.
- **3V3:** Παρέχει 3.3V και 50mA (ενώ τα υπόλοιπα παρέχουν 40mA)
- **GND:** Γειωμένα pins.
- **IOREF:** Παρέχει την τάση αναφοράς, με την οποία λειτουργεί η πλακέτα.

Μνήμη

Το Arduino διαθέτει συνολικά 256 KB μνήμης, εκ των οποίων τα 8 KB είναι SRAM (Static Random Access Memory) και τα 4 KB EEPROM (Electrically Erasable Programmable Read Only Memory).

Pin mapping

Καθένα από τα ψηφιακά pins μπορούν να χρησιμοποιηθούν είτε σαν είσοδοι είτε σαν έξοδοι ψηφιακών σημάτων, δηλαδή για να διαβάσουν ή να στείλουν ψηφιακά σήματα. Η πλειοψηφία των pins του Arduino, εκτός από την λειτουργία τους σαν είσοδοι/έξοδοι του μικροελεγκτή, έχουν και κάποιες επιπρόσθετες λειτουργίες. Πιο συγκεκριμένα, οι λειτουργίες αυτές είναι:

- **Serial:** Τα pins που υποστηρίζουν αυτή τη λειτουργία είναι Serial 0(RX¹) 1(TX²), Serial1 19(RX) 18(TX), Serial2 17(RX) 16(TX), Serial3 15(RX) 14(TX). Τα pins αυτά λειτουργούν ανά ζεύγη και μπορούν να δέχονται (RX) και να μεταφέρουν (TX) TTL (transistor-transistor logic) σειριακά δεδομένα.

¹Transmitter

²Receiver

- **External Interrupts:** Η λειτουργία αυτή υποστηρίζεται από τα pins 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), και 21 (interrupt 2). Τα pins αυτά μπορούν να ρυθμιστούν, ώστε να διακόπτεται η λειτουργία του μικροελεγκτή και να ακολουθείται από κάποια άλλη προγραμματισμένη ενέργεια, ανάλογα με τη συμπεριφορά της τιμής κάποιου από αυτά τα pins.
- **PWM:** Τα pins 2, 13, 44, 46 προσφέρουν 8-bit διαμόρφωση πλάτους παλμού (Pulse Width Modulation PWM).
- **SPI:** Το Arduino Mega 2560, μέσω των pins 50 (MISO³), 51 (MOSI⁴), 52 (SCK), 53 (SS) δίνει τη δυνατότητα επικοινωνίας μέσω του πρωτοκόλλου SPI (Serial Peripheral Interface).
- **LED:** Στο pin 13, υπάρχει συνδεδεμένο ένα λαμπάκι LED, το οποίο ο χρήστης μπορεί να χειριστεί αλλάζοντας την έξοδο του pin 13.
- **TWI (I2C):** Υποστήριξη του πρωτοκόλλου επικοινωνίας TWI[9] (Two Wire Interface ή I²C, Inter-Integrated Circuit) μέσω των pins 20 (SDA) 21 (SCL).
- **AREF:** Είναι το pin, το οποίο λειτουργεί ως σημείο αναφοράς για το εύρος μέτρησης των αναλογικών εισόδων του Arduino.
- **Reset:** Επιτρέπει την επανεκκίνηση του μικροελεγκτή. Η ουσιαστική του χρήση είναι ότι επιτρέπει στις πλακέτες επέκτασης να έχουν ένα κουμπί reset, καθώς αυτές, όταν προστεθούν, μπλοκάρουν το ήδη υπάρχον κουμπί στην πλακέτα του Arduino.

Τέλος, το Arduino διαθέτει και μια σειρά από αναλογικά pins. Τα pins αυτά, μπορούν να λειτουργήσουν σαν εισοδοί τόσο για ψηφιακά όσο και για αναλογικά σήματα. Το Arduino Mega 2560 έχει 16 αναλογικές εισόδους, κάθε μία από τις οποίες έχει 10-bit ανάλυση, έχει τη δυνατότητα δηλαδή μέτρησης 1024 διαφορετικών τιμών. Τα pins αυτά είναι ρυθμισμένα να μετρούν αυτές τις 1024 τιμές σε ένα εύρος 0 έως 5 V. Μέσω του pin AREF, υπάρχει η δυνατότητα αλλαγής του εύρους αυτού. Εφαρμόζοντας στο συγκεκριμένο pin μία τάση μικρότερη των 5 V αλλάζει το εύρος των τιμών των αναλογικών pins του Arduino από 0 έως την τιμή της τάσης που εφαρμόστηκε στο AREF pin. Με το τρόπο αυτό επιτυγχάνεται μεγαλύτερη ακρίβεια, καθώς τώρα το σύνολο των 1024 τιμών που μετρούν τα pins κατανέμεται σε ένα μικρότερο εύρος τιμών.

Λίστα χαρακτηριστικών στοιχείων του Arduino Mega 2560

Μικροελεγκτής: ATmega2560

³Master Input Slave Output

⁴Master Output Slave Input

Τάση λειτουργίας: 5 V

Τάση εισόδου(προτεινόμενη): 7-12 V

Τάση εισόδου(όρια): 6-20 V

Ψηφιακά I/O Pins: 54 (15 από τα οποία παρέχουν PWM έξοδο)

Αναλογικά pins εισόδου: 16

DC Ρεύμα ανά pin: 40 μ A

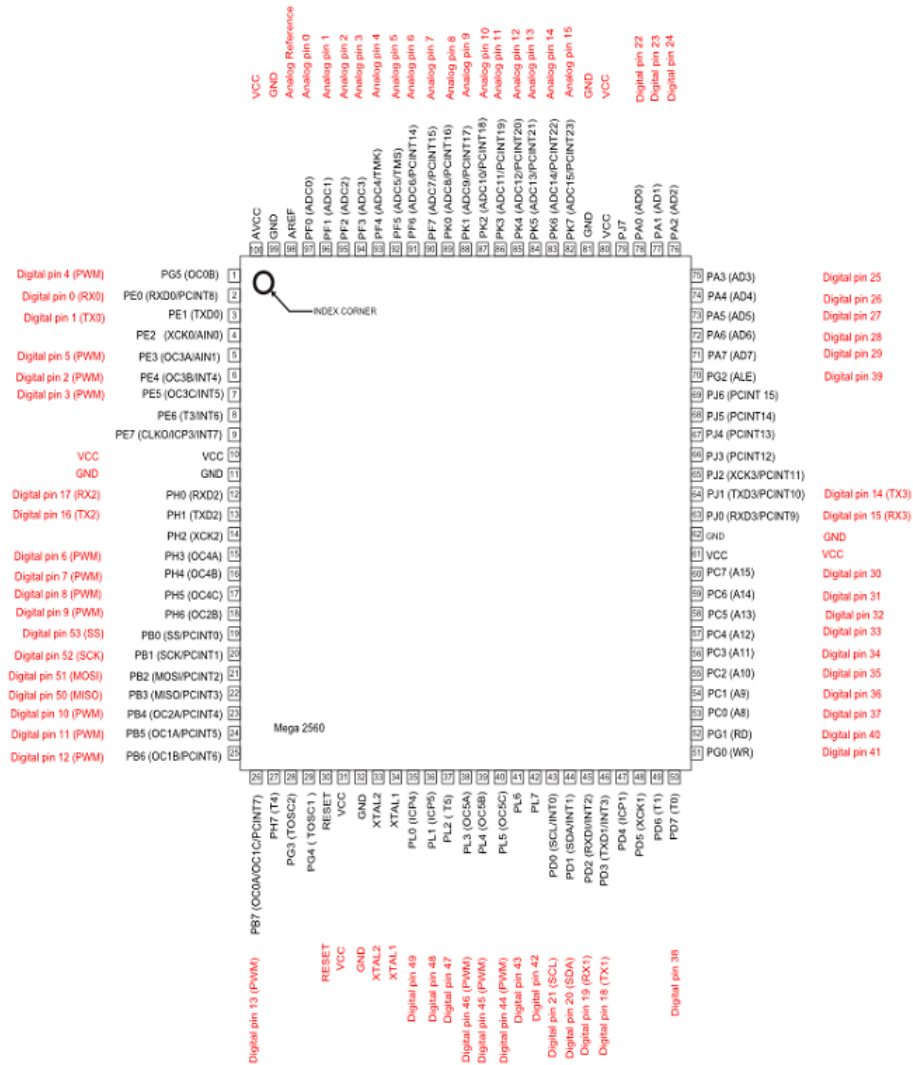
DC Ρεύμα για το 3.3V pin: 50 μ A

Μνήμη: 256 KB από τα οποία 8 KB χρησιμοποιούνται από τον boot-loader

SRAM: 8 KB

EEPROM: 4 KB

Συχνότητα ταλαντωτή: 16 MHz



Σχήμα 1.3: ATMega2560 Pin Mapping

1.1.3 Arduino Ethernet Shield



Σχήμα 1.4: Arduino Ethernet Shield

Στην εργασία αυτή, χρησιμοποιήθηκε επιπλέον μία από τις πλακέτες επέκτασης για Arduino, η πλακέτα Arduino Ethernet Shield [1.4]. Το Shield αυτό, δίνει τη δυνατότητα σύνδεσης του Arduino με κάποιο δίκτυο. Για να πραγματοποιηθεί η σύνδεση, θα πρέπει το Shield να συνδεθεί με το Arduino καθώς και μέσω ενός καλωδίου RJ45 με ένα router. Απαραίτητη προϋπόθεση είναι να προγραμματίσει κανείς το Arduino, όπως θα δούμε παρακάτω.

Το Arduino Ethernet Shield είναι βασισμένο στο Wiznet W5100 Ethernet Chip [31]. Το chip αυτό υποστηρίζει τα πρωτόκολλα TCP (Transmission Control Protocol) και UDP (User Datagram Protocol). Ο τρόπος, με τον οποίο συνδέεται το Ethernet Shield μέσω μεγάλων συρμάτων επαφών, δίνει τη δυνατότητα προσθήκης και άλλου Ethernet Shield.

Το shield διαθέτει επίσης μια υποδοχή για microSD Card, μέσω της οποίας δίνεται η δυνατότητα αποθήκευσης αρχείων από το Arduino. Αυτή η δυνατότητα είναι πολύ σημαντική, καθώς το Arduino δε διαθέτει τη δυνατότητα αποθήκευσης κάποιου αρχείου στον υπολογιστή.

Το Arduino επικοινωνεί τόσο με το W5100 chip όσο και με την SD κάρτα, μέσω του πρωτοκόλλου επικοινωνίας SPI. Αυτή επιτυγχάνεται μέσω των pins 50, 51, 52, ενώ σαν SS pin χρησιμοποιείται το pin 10 για το W5100 chip και το pin 4 για την SD κάρτα. Το pin 53 δεν χρησιμοποιείται σαν SS pin και πρέπει να δηλωθεί σαν output. Εδώ αξίζει να σημειωθεί ότι δεν μπορούν να χρησιμοποιούνται ταυτόχρονα και το W5100 Ethernet Chip και η SD κάρτα, καθώς και τα 2 χρησιμοποιούν το SPI πρωτόκολλο. Θα πρέπει δηλαδή, όταν το ένα χρησιμοποιείται, το άλλο να απενεργοποιείται, θέτοντας το SS pin που αντιστοιχεί σε αυτό στην κατάσταση low.

Επιπλέον, το Ethernet Shield διαθέτει μία υποδοχή RJ45 για ethernet καλώδιο, καθώς κι ένα κουμπί reset, το οποίο λειτουργεί τόσο στο Shield όσο στο Arduino. Τέλος, το Shield διαθέτει μια πληθώρα από ενυμερωτικά LEDs. Αναλυτικά :

- **PWR:** Όταν αυτό το LED είναι αναμμένο, καταδεικνύει ότι το Shield και το Arduino τροφοδοτούνται με ρεύμα.

- **LINK:** Δείχνει ότι υπάρχει σύνδεση δικτύου και αναβοσβήνει όταν το Shield λαμβάνει ή στέλνει δεδομένα.
- **FULLD:** Δείχνει ότι η σύνδεση είναι πλήρως αμφίδρομη.
- **100M:** Δείχνει ότι υπάρχει σύνδεση δικτύου με ταχύτητα 100 Mb/s.
- **RX:** Αναβοσβήνει όταν το Shield λαμβάνει δεδομένα.
- **TX:** Αναβοσβήνει όταν το Shield μεταδίδει δεδομένα.
- **COLL:** Αναβοσβήνει όταν ανιχνευεται κάποια "σύγκρουση" (conflict) στο δίκτυο.



Σχήμα 1.5: Πλακέτα Arduino και Ethernet Shield

1.1.4 Αλληλεπίδραση με το Arduino

Το πρώτο και βασικότερο που πρέπει να κάνει κάποιος για να χρησιμοποιήσει το Arduino, είναι να το προγραμματίσει. Αυτό γίνεται με το περιβάλλον εργασίας Arduino ή αλλιώς Arduino IDE [3] (Integrated Development Environment). Το IDE διατίθεται δωρεάν στο διαδίκτυο και χρησιμοποιείται τόσο για το γράψιμο του κώδικα, όσο και για να "φορτωθεί" (upload) ο κώδικας στο Arduino.

Το IDE αποτελείται από έναν επεξεργαστή κειμένου (text editor), μία περιοχή μνημάτων, μία κονσόλα κειμένου, μία γραμμή εργαλείων με κουμπιά διαφόρων λειτουργιών και μια σειρά απο μενού, όπως φαίνεται στην παρακάτω εικόνα.[1.6]



Σχήμα 1.6: Arduino IDE

Δημιουργία "σκίτσων"

Ο κώδικας που γράφεται στο Arduino IDE, ονομάζεται **σκίτσο (sketch)**. Αυτά γράφονται με τον επεξεργαστή κειμένου του IDE και σώζονται σαν αρχεία με την κατάληξη ".ino". Τα sketches, σώζονται σε ένα φάκελο με όνομα Sketchbook, ο οποίος δημιουργείται αυτόματα την πρώτη φορά που εκτελείται το IDE.

Στην περιοχή μηνυμάτων, εμφανίζονται πληροφορίες σχετικά με τις ενέργειες του χρήστη, παραδείγματος χάρι όταν σώζει ένα αρχείο ή όταν το κάνει upload στην πλακέτα. Στην κονσόλα, εμφανίζονται πληροφορίες σχετικές με τυχόν λάθη στον κώδικα και πληροφορίες για τον κώδικα, όπως το πόση μνήμη καταναλώνει κλπ.

Το IDE διαθέτει μια πληθώρα κουμπιών και μενού. Πιο αναλυτικά, τα κουμπιά είναι :

- **Verify:** Έλεγχος του κώδικα για τυχόν συντακτικά λάθη, το κουμπί αυτό δεν "φορτώνει" τον κώδικα στο Arduino.
- **Upload:** Ελέγχει τον κώδικα για λάθη και στη συνέχεια κάνει "upload" τον κώδικα στο Arduino.
- **New:** Δημιουργία νέου κενού Sketch
- **Open:** Άνοιγμα λίστας με όλα τα αποθηκευμένα Sketch απο την οποία μπορεί να γίνει επιλογή κάποιου Sketch ώστε αυτό να ανοιχθεί στον επεξεργαστή κειμένου.

- **Save:** Αποθήκευση παρόντος Sketch.
- **Serial Monitor:** Ανοίγει τη σειριακή οθόνη (Serial Monitor), στην οποία το Arduino μπορεί να μεταδίδει δεδομένα. Η οθόνη αυτή μπορεί επίσης να χρησιμοποιηθεί για αποστολή δεδομένων από τον χρήστη στον Arduino.

Ακόμα υπάρχουν τα menus "File", "Edit", "Sketch", "Tools", "Help", τα οποία προσφέρουν περαιτέρω δυνατότητες στο χρήστη.

Εδώ να σημειωθεί, ότι ένας κώδικας που "φορτώνεται" στο Arduino μπορεί να αποτελείται από περισσότερα από ένα Sketch ή αρχεία, γραμμένα είτε σε γλώσσα Arduino είτε σε C ή C++.

"Φόρτωση" κώδικα (Uploading) στο Arduino

Πριν γίνει η φόρτωση του κώδικα στο Arduino, πρέπει να επιλεγεί η θύρα στην οποία είναι συνδεδεμένο το Arduino καθώς και το είδος του Arduino στο οποίο θα φορτωθεί ο κώδικας (π.χ. Arduino Mega). Όταν πατηθεί το κουμπί Upload, αρχίζουν να αναβοσβίνουν τα TX, RX LED στην πλακέτα, που υποδηλώνουν ότι η πλακέτα στέλνει και λαμβάνει δεδομένα. Τέλος, στην κονσόλα εμφανίζονται μηνύματα σχετικά με τη διαδικασία, το κατά πόσο αυτή ολοκληρώθηκε με επιτυχία ή όχι και αν όχι, για ποιό λόγο.

Βιβλιοθήκες

Οι **βιβλιοθήκες** είναι κώδικες ξεχωριστοί από τα σκίτσα, οι οποίοι χρησιμοποιούνται για να δώσουν περαιτέρω δυνατότητες σε αυτά, για παράδειγμα επικοινωνία με κάποια εξωτερική συσκευή επεξεργασία δεδομένων κ.λ.π. Μια βιβλιοθήκη, για παράδειγμα, μπορεί να είναι η διαδικασία που πρέπει να ακολουθηθεί από το Arduino για να επικοινωνήσει με έναν αισθητήρα. Η βιβλιοθήκη θα καθορίζει πώς το Arduino λαμβάνει και στέλνει σήματα. Οι βιβλιοθήκες δηλώνονται στην αρχή κάθε σκίτσου που τις χρησιμοποιεί. Για να μπορέσουν να διαβαστούν από το Arduino οι επιθυμητές βιβλιοθήκες, θα πρέπει να είναι αποθηκευμένες σε ένα φάκελο με όνομα "libraries", ο οποίος είναι μέσα στο Sketchbook. Ο φάκελος αυτός δημιουργείται αυτόματα μαζί με το φάκελο Sketchbook από το IDE. Πολλές βιβλιοθήκες βρίσκονται διαθέσιμες στο διαδίκτυο, άλλες υπάρχουν ήδη στο IDE, ενώ ο χρήστης μπορεί πάντα να φτιάξει τις δικές του.

Serial Monitor

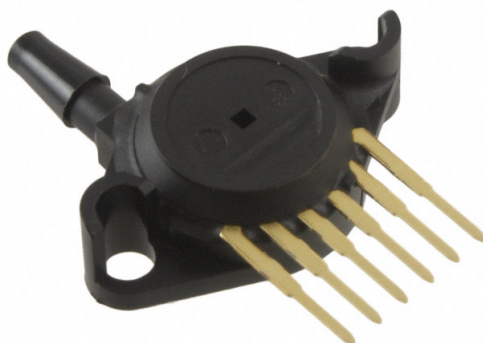
Πατώντας το κουμπί Serial Monitor, ανοίγει η σειριακή οθόνη. Σε αυτή, τυπώνονται δεδομένα από το Arduino εφόσον αυτό έχει επιλεγεί στον κώδικα, επίσης ο χρήστης μπορεί να στείλει δεδομένα στο Arduino μέσω του Serial Monitor, και πάλι έχοντας "φορτώσει" τις κατάλληλες γραμμές στον κώδικα στην πλακέτα.

1.2 Αισθητήρες

Στην ενότητα αυτή παρουσιάζονται οι αισθητήρες που χρησιμοποιήθηκαν στην παρούσα εργασία. Πιο συγκεκριμένα οι αισθητήρες αυτοί είναι οι MPX5700AP [24], BMP085 [29], MPL3115A2 [25], DHT22 [20]. Η ενότητα αυτή περιλαμβάνει την περιγραφή κάθε αισθητήρα και μετρήσεις που έγιναν για τον καθένα από αυτούς μεμονομένα, καθώς και συνδιαστικά σε κοινή συνδεσμολογία. Επιπλέον παρουσιάζονται συγκριτικές μετρήσεις της κοινής συνδεσμολογίας των αισθητήρων με αυτές του αισθητήρα HMP233 της Vaisala [30], καθώς και παρακολούθηση της λειτουργικής συμπεριφοράς συστήματος με Arduino και τους αισθητήρες που περιγράφονται στην παρούσα ενότητα.

1.2.1 Αισθητήρας MPX5700AP

A. Περιγραφή



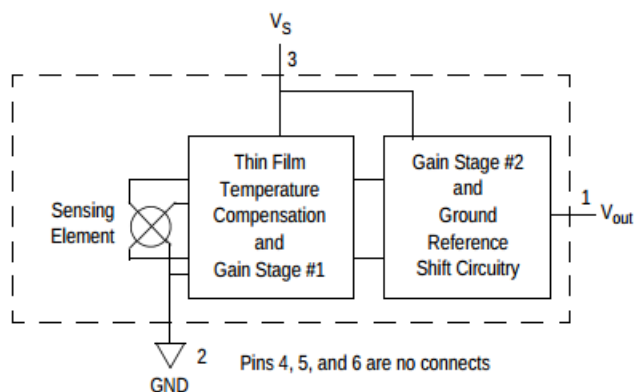
Σχήμα 1.7: Αισθητήρας MPX5700AP

Ο MPX5700AP, εικόνα [1.7], είναι ένας αισθητήρας απόλυτης πίεσης, μετράει, δηλαδή, την πίεση με σημείο αναφοράς το απόλυτο κενό. Είναι ένας αισθητήρας αναλογικού σήματος, δηλαδή η έξοδος του έχει μια διαφορά δυναμικού σε σχέση με την γειώσή του και από τη διαφορά τους αυτή, μπορεί κανείς να υπολογίζει την πίεση. Η τάση εξόδου του συνδέεται με την πίεση μέσω ενός πολυωνύμου πρώτου βαθμού το οποίο δίνεται από τον κατασκευαστή κάνοντας έτσι την μετατροπή της μετρούμενης τάσης από το Arduino σε πίεση δυνατή. Στο πίνακα που ακολουθεί [1.1] παραθέτονται τα χαρακτηριστικά του αισθητήρα.

Characteristic	Min	Typical	Max	Unit
Pressure Range	15	-	700	kPa
Supply Voltage	4.75	6.0	5.25	Vdc
Supply Current	-	7.0	10	mAdc
Zero Pressure Offset (0 to 85 °C)	0.184	-	0.409	Vdc
Full Scale Output (0 to 85 °C)	4.587	4.7	4.813	Vdc
Full Scale Span (0 to 85 °C)	-	4.5	-	Vdc
Accuracy	-	-	±2.5	% V _{FSS}
Sensitivity	-	6.4	-	mv/kPa
Response Time	-	1.0	-	ms
Output Source Current at Full Scale Output	-	0.1	-	mAdc
Warm-Up Time	-	20	-	ms

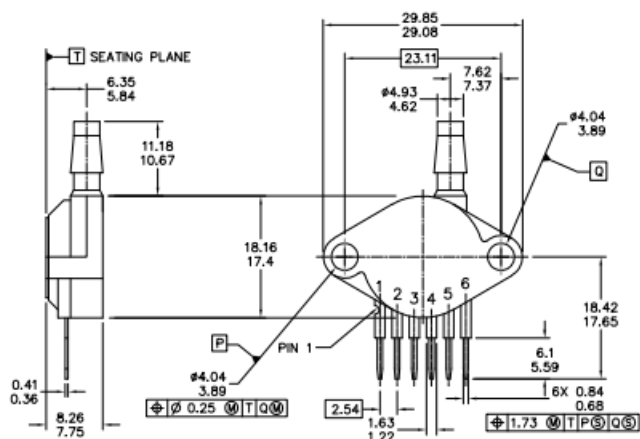
Πίνακας 1.1: Χαρακτηριστικά του αισθητήρα MPX5700AP

Στις τέσσερις εικόνες που ακολουθούν [1.8], [1.9], [1.10], [1.11], παρουσιάζονται τα χαρακτηριστικά σύνδεσης του αισθητήρα αυτού με το Arduino. Πιο συγκεκριμένα, στην εικόνα [1.8] φαίνεται το εσωτερικό του αισθητήρα καθώς και ποιο pin του αισθητήρα πρέπει να συνδεθεί με ποιο pin του Arduino. Στην εικόνα [1.9], φαίνεται το "σηματικό" του αισθητήρα. Στην εικόνα [1.10], φαίνεται η προτεινόμενη συνδεσμολογία που πρέπει να ακολουθηθεί κατά τη σύνδεση του αισθητήρα ώστε να αποκοπούν από το σήμα ανεπιθύμητες συνεισφορές, όπως για παράδειγμα ο θόρυβος. Τέλος στην εικόνα [1.11], φαίνεται πώς πραγματοποιείται η συνδεσμολογία του αισθητήρα με το Arduino κατά τις μετρήσεις που πάρθηκαν στην παρούσα εργασία.



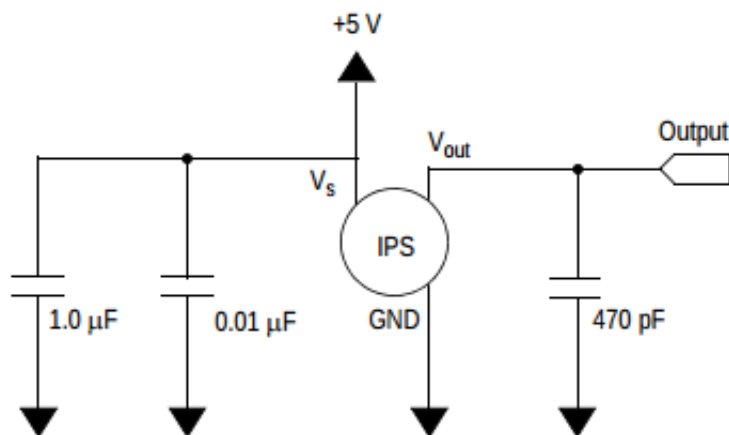
Σχήμα 1.8: Εσωτερικό του αισθητήρα MPX5700AP

PACKAGE DIMENSIONS

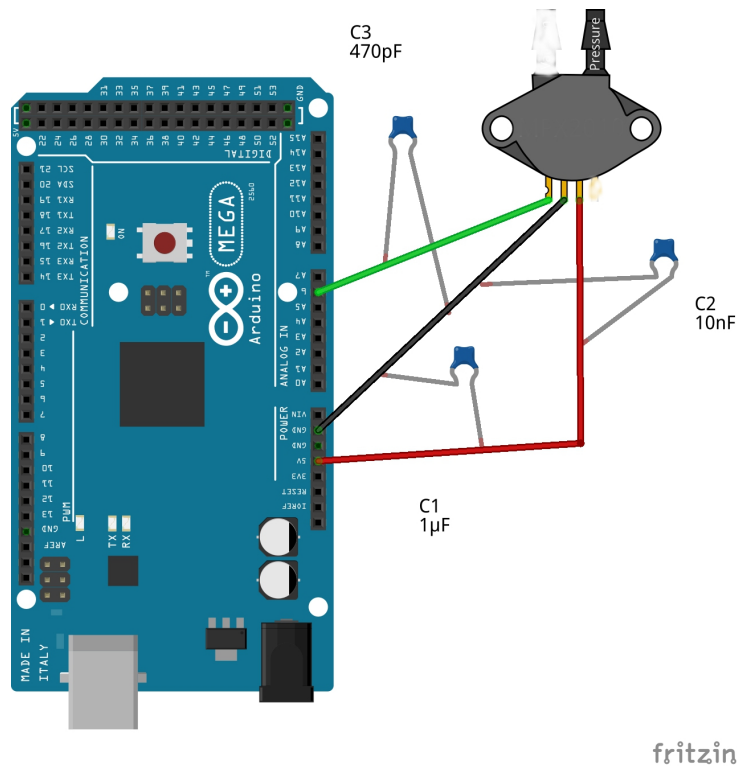


© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: SENSOR, 6 LEAD UNIBODY CELL, AP & GP 01ASB09087B	DOCUMENT NO: 98ASB42796B CASE NUMBER: B67B-04 STANDARD: NON-JEDEC	REV: G 28 JUL 2005

Σχήμα 1.9: Σχηματικό του αισθητήρα MPX5700AP

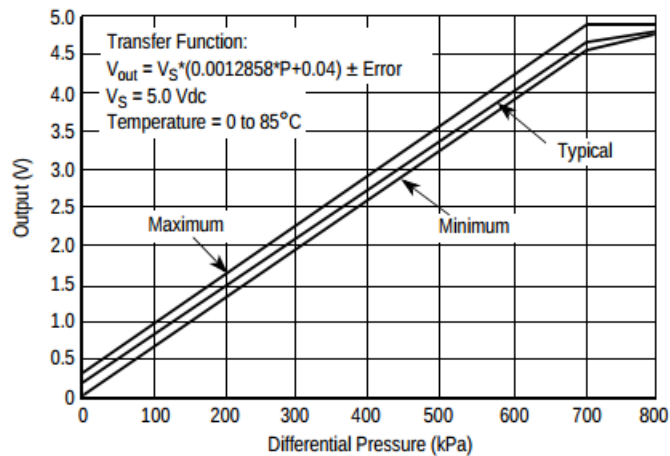


Σχήμα 1.10: Προτεινόμενη από τον κατασκευαστή συνδεσμολογία για τον MPX5700AP



Σχήμα 1.11: Συνδεσμολογία του αισθητήρα MPX5700AP με το Arduino

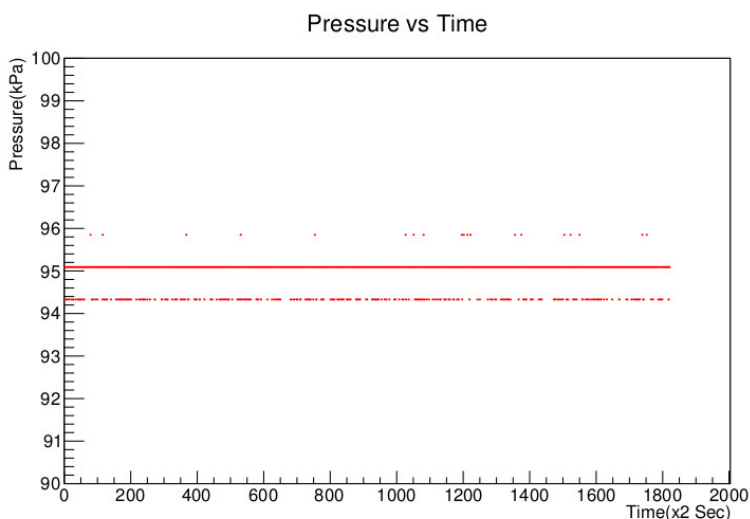
Τέλος, ο κατασκευαστής παρέχει ένα διάγραμμα, εικόνα [1.12], το οποίο δείχνει την εξάρτηση της τάσης από την πίεση και επιτρέπει έτσι τη μετατροπή της μετρούμενης από το Arduino τάσης σε πίεση.



Σχήμα 1.12: Διάγραμμα της πίεσης συναρτήσει της τάσης για το αισθητήρα MPX5700AP

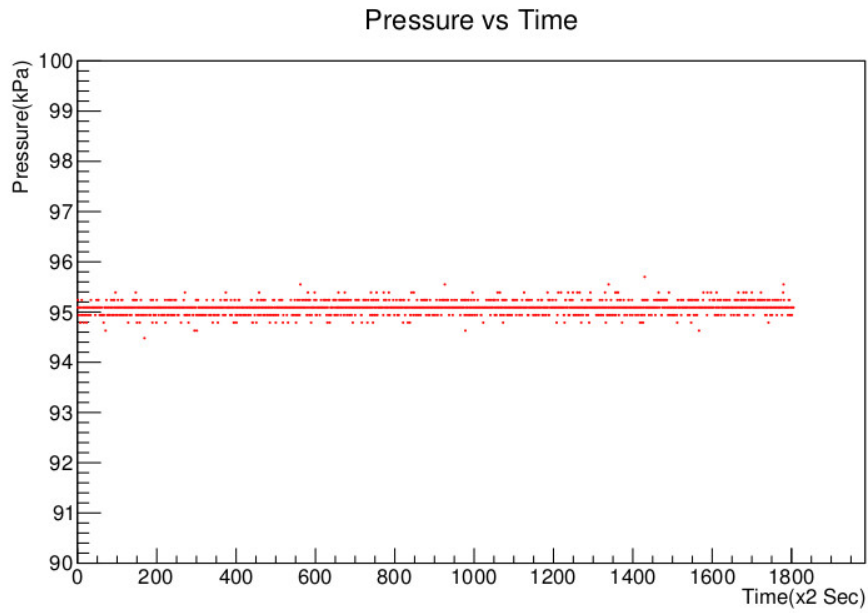
B. Μετρήσεις

Για να δοκιμαστεί η λειτουργία του αισθητήρα, ο τελευταίος συνδέθηκε στο Arduino με τη συνδεσμολογία που φαίνεται στην εικόνα [1.11]. Πάρθηκαν μετρήσεις σε ένα διάστημα 2 ωρών με βήμα μέτρησης (sampling) 2 δευτερόλεπτα. Οι μετρήσεις αποθηκεύθηκαν στην SD κάρτα καθώς, όπως αναφέρθηκε και πιο πάνω, το Arduino δεν έχει δυνατότητα αποθήκευσης στον υπολογιστή. Στο παρακάτω σχήμα φαίνεται η πίεση στο διάστημα αυτό, συναρτήσει του χρόνου [1.13].

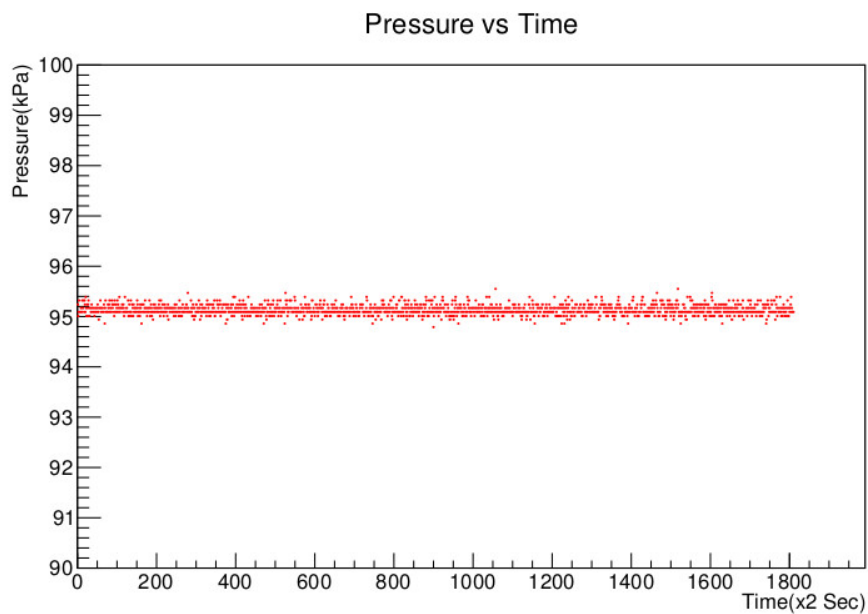


Σχήμα 1.13: Διάγραμμα πίεσης συναρτήσει του χρόνου για τον αισθητήρα MPX5700 με μετρήσεις που πάρθηκαν με το Arduino σε διάστημα 2 ωρών με βήμα μέτρησης (sampling) 2 δευτερόλεπτα

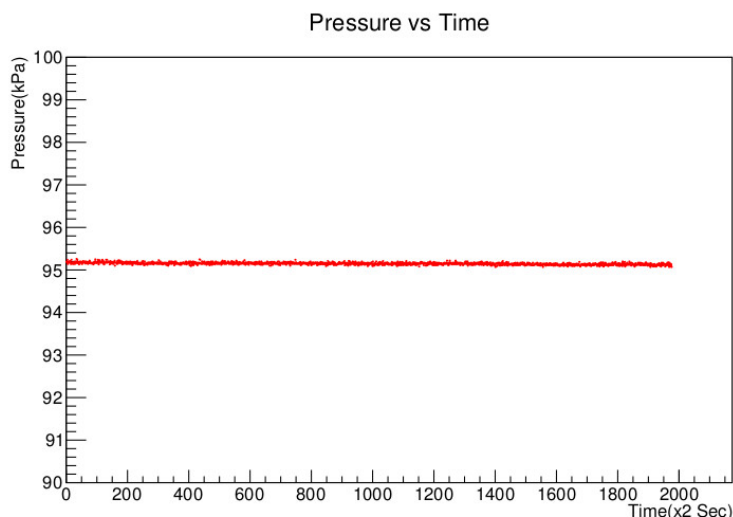
Είναι φανερό ότι οι μετρήσεις δεν είναι καθόλου σταθερές. Ένας από τους λόγους μπορεί να είναι ότι το Arduino μετράει σε τυχαία χρονικά διαστήματα και ο παλμός που παράγει ο αισθητήρας δεν έχει προλάβει να φτάσει στη μέγιστη τιμή του, με αποτέλεσμα να μη μετράει πάντα την ίδια τάση. Για το λόγο αυτό, αποφασίστηκε αντί για μία μέτρηση κάθε 2 δευτερόλεπτα, να λαμβάνεται ένα σύνολο δειγμάτων (samples) για το διάστημα αυτό και η τελική μέτρηση να προκύπτει από το μέσο όρο των samples. Στις εικόνες [1.14], [1.15] και [1.16] εμφανίζονται διαγράμματα τα οποία προέκυψαν από μετρήσεις στις οποίες για το διάστημα αυτό των 2 δευτερολέπτων πάρθηκαν 5, 10 και 100 δείγματα αντίστοιχα.



Σχήμα 1.14: Μετρήσεις πίεσης συναρτήσει του χρόνου για τον αισθητήρα MPX5700 με μετρήσεις που πάρθηκαν με το Arduino σε διάστημα 2 ωρών με βήμα μέτρησης 2 δευτερόλεπτα και 5 δείγματα ανά μέτρηση



Σχήμα 1.15: Μετρήσεις πίεσης συναρτήσει του χρόνου για τον αισθητήρα MPX5700 με μετρήσεις που πάρθηκαν με το Arduino σε διάστημα 2 ωρών με βήμα μέτρησης 2 δευτερόλεπτα και 10 δείγματα ανά μέτρηση

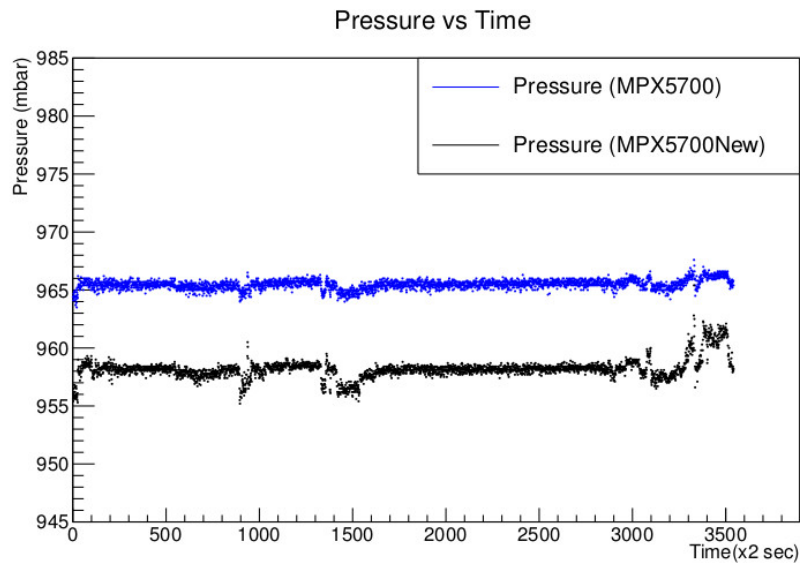


Σχήμα 1.16: Μετρήσεις πίεσης συναρτήσει του χρόνου για τον αισθητήρα MPX5700 με μετρήσεις που πάρθηκαν με το Arduino σε διάστημα 2 ωρών με βήμα μέτρησης 2 δευτερόλεπτα και 100 δείγματα ανά μέτρηση

Παρατηρείται ότι όσο αυξάνεται ο αριθμός των δειγμάτων, τόσο σταθεροποιείται η τιμή της μέτρησής. Για τα 100 δείγματα, η σταθερότητα θεωρήθηκε ικανοποιητική και αποφασίστηκε να μην αυξηθεί άλλο ο αριθμός τους. Από εδώ και πέρα, όλες οι μετρήσεις με αυτόν τον αισθητήρα θα αναφέρονται στην περίπτωση των 100 samples στον χρόνο στον οποίο πάρθηκε η κάθε μέτρηση (δηλαδή όταν θα αναφέρεται ότι πάρθηκε 1 μέτρηση σε διάστημα 4 δευτερολέπτων θα εννοείται ότι λήφθηκαν 100 samples στο διάστημα των 4 δευτερολέπτων).

Συνολικά στο εργαστήριο υπήρχαν δύο αισθητήρες MPX5700. Στην παρακάτω εικόνα [1.17] φαίνεται ένα διάγραμμα με μετρήσεις και από τους δύο αισθητήρες (100 samples) σε ένα διάστημα 4 ωρών με μετρήσεις ανά 2 δευτερόλεπτα. Στο διάγραμμα αυτό μπορεί κανείς να δει ότι υπάρχει αρκετά μεγάλη διαφορά μεταξύ των μετρήσεων των δύο αισθητήρων. Ωστόσο, αν παρατηρήσει κανείς τον πίνακα με τα χαρακτηριστικά του αισθητήρα [1.1], η ακρίβεια του αισθητήρα (accuracy) είναι $\pm 2,5\% V_{FSS}^5$, οπότε βρισκόμαστε μέσα στα όρια του σφάλματος.

⁵Full Scale Span Voltage ορίζεται ως η αλγεβρική διαφορά μεταξύ της τάσης για τη μεγαλύτερη τιμή της πίεσης που μπορεί να καταγράψει ο αισθητήρας μείον τη μικρότερη τιμή



Σχήμα 1.17: Διάγραμμα πίεσης συναρτήσει του χρόνου και για τους δύο αισθητήρες MPX5700AP

1.2.2 Αισθητήρας BMP085

A. Περιγραφή



Σχήμα 1.18: Ψηφιακός αισθητήρας πίεσης και θερμοκρασίας BMP085

Ο BMP085 [29], εικόνα [1.18], είναι ένας ψηφιακός αισθητήρας μέτρησης πίεσης και θερμοκρασίας. Η εξαιρετικά υψηλή του ακρίβεια, η χαμηλή κατανάλωση του σε ισχύ, το μικρό του μέγεθος καθώς και το ότι χρησιμοποιεί το πρωτόκολλο επικοινωνίας I²C τον κάνουν ιδανικό για χρήση σε εφαρμογές με μικροεπεξεργαστές, μικροελεγχτές όπως το Arduino. Ο BMP085 προσφέρεται από πολλές εταιρίες, στην περίπτωση μας, χρησιμοποιήθηκε το sensor board AVR4201 της ATMEL [7] το οποίο φαίνεται στην εικόνα [1.19] και έχει ενσωματωμένο τον BMP085. Στον παρακάτω πίνακα [1.2] βρίσκονται κάποια από τα χαρακτηριστικά του αισθητήρα αυτού.

Characteristic	Condition	Min	Typical	Max	Unit
Operating Temperature	operational	-40		+85	°C
	full accuracy	0		+65	°C
Supply Voltage	ripple max. 50mV _{pp}	1.8	2.5	3.3	V
		1.62	2.5	3.6	V
Supply Current	ultra low power mode		3		μA
	standard mode		5		μA
	high resolution mode		7		μA
	ultra high res, mode		12		μA
Peak Current	during conversion		650	1000	μA
Standby Current	at 25 °C		0.1		μA
Serial Data Clock				3.4	MHz
Conversion Time Temperature	standard mode		3	4.5	ms
Conversion Time Pressure	ultra low power mode		3	4.5	ms
	standard mode		5	7.5	ms
	high resolution mode		9	13.5	ms
	ultra high res, mode		17	25.5	ms
Absolute accuracy pressure(3.3V)*	700..1100hPa 0..65 °C	-2.5	±1.0	+2.5	hPa
	300..700hPa 0..65 °C	-3.0	±1.0	+3.0	hPa
	300..1100hPa -20..0 °C	-4.0	±1.5	+4.0	hPa
Resolution of output data	pressure		0.01		hpa
	temperature		0.1		°C
Relative accuracy pressure(3.3V)	700..1100hPa 25 °C		±0.2		hPa
	0..65 °C p const		±0.5		hPa
Absolute accuracy temperature(3.3V)	@ 25 °C	-1.57	±0.5	+1.5	°C
	0..+65 °C	-2.0	±1.0	+2.0	°C
Long Term Stability	12 months		±1.0		hpa

Πίνακας 1.2: Χαρακτηριστικά του αισθητήρα BMP085



Σχήμα 1.19: Atmel AVR4201 με τον αισθητήρα BMP085

Στο σημείο αυτό, θα γίνει αναφορά στον τρόπο λειτουργίας του πρωτοκόλλου I²C. Το πρωτόκολλο επικοινωνίας αυτό (Inter-Integrated Circuit) δημιουργήθηκε από την εταιρία **Philips Semiconductor** [26] (σήμερα γνωστή ως NXP Semiconductors) τη δεκαετία του '80. Χρησιμοποιείται συνήθως για την σύνδεση "αργών" περιφερειακών συσκευών με επεξεργαστές ή μικροελεγχτές. Ο μικροελεγχτής ονομάζεται **master** και όλες οι περιφερειακές συσκευές **slaves**. Η συνδεσμολογία αποτελείται από 2 μόνο γραμμές την SDA (Serial Data Line) και την SCL (Serial Clock Line), δηλαδή ανεξάρτητα από τον αριθμό των slaves που είναι συνδεδεμένοι στο κύκλωμα, απαιτούνται μόνο δύο καλώδια για να επιτευχθεί επικοινωνία με όλους αυτούς. Εννοείται βέβαια, πως κάθε μία από τις συσκευές που υπάρχουν στο κύκλωμα, θα πρέπει να είναι συνδεδεμένη με μία τάση και μία γείωση. Επίσης οι γραμμές επικοινωνίας (SDA, SCL) πρέπει να συνδέονται μέσω μίας pull up αντίστατης με την τάση λειτουργίας του συστήματος. Η γραμμή SCL ταλαντώνεται μεταξύ της κατάστασης 0 και 1 με μία συχνότητα, η οποία ποικίλλει από 10 kbit/s έως 3.4 Mbit/s. Η συνηθισμένη συχνότητα, ωστόσο, είναι 100 kbit/s. Κάθε φορά που η SCL βρίσκεται στην κατάσταση 1, η SDA αντιστοιχεί στο bit που διαβάζει ή στέλνει ο master. Στην εικόνα [1.20], φαίνεται η συνδεσμολογία που ακολουθείται από μία διάταξη με πρωτόκολλο I²C.

Κάθε ένας από τους slaves έχει μία διεύθυνση (Address) η οποία μπορεί να είναι είτε 7-bit είτε 10-bit και εξαρτάται από τη συσκευή. Για να εξηγηθεί καλύτερα πώς ακριβώς γίνεται η επικοινωνία, παρατίθεται μία εικόνα [1.21] και με βάση αυτή θα αναλυθούν τα βήματα που γίνονται κατά τη διάρκεια επικοινωνίας με χρήση του συγκεκριμένου πρωτοκόλλου. Η εικόνα αυτή έχει ληφθεί με τη βοήθεια logic Analyzer.

Στην αρχή και οι δύο γραμμές είναι στο υψηλό τους σημείο. Η επικοινωνία ξεκινάει με το λεγόμενο **start condition** το οποίο, όπως φαίνεται στο σχήμα, είναι όταν το SDA γίνεται 0 ενώ το SCL παραμένει

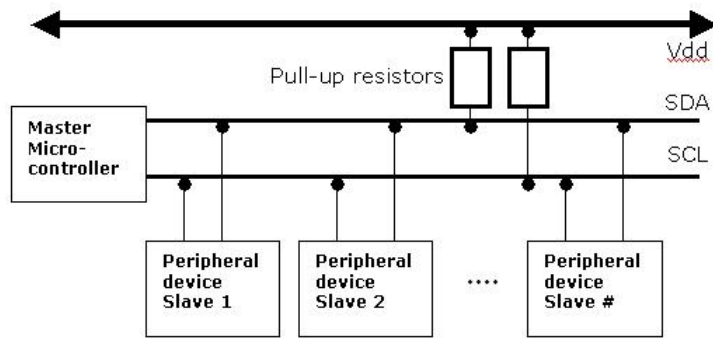
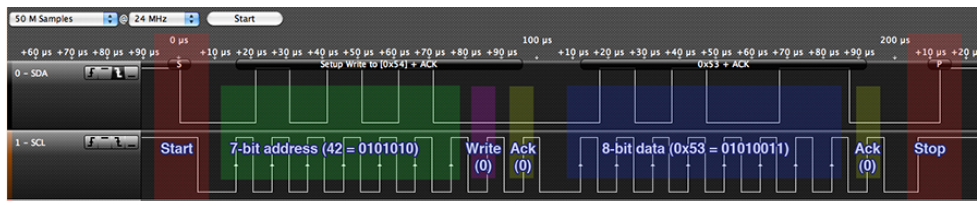


Fig- 1 (Basic I²C blocks)

Σχήμα 1.20: Συνδεσμολογία διάταξης που χρησιμοποιεί το πρωτόκολλο επικοινωνίας I²C



Σχήμα 1.21: Απεικόνιση των γραμμών SDA, SCL κατά τη διάρκεια επικοινωνίας με χρήση του πρωτοκόλλου I²C

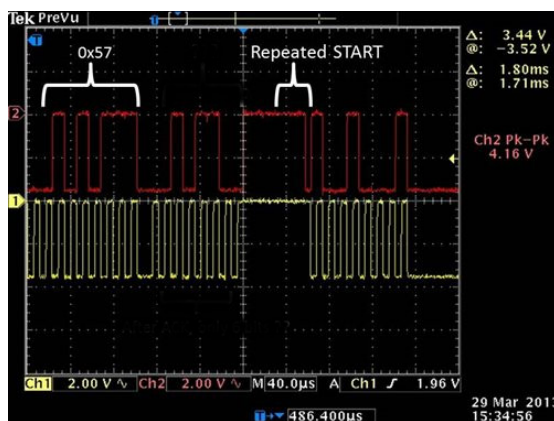
υψηλό. Η ενέργεια αυτή ενημερώνει όλους τους slaves στο σύστημα ότι η επικοινωνία ξεκινάει.

Στη συνέχεια, ο master στέλνει τη διεύθυνση του slave με τον οποίο θέλει να επικοινωνήσει. Στην περίπτωση του σχήματός μας, η διεύθυνση αυτή είναι η 42. Στην συνέχεια, ακολουθεί ένα bit, το οποίο υποδηλώνει τη λειτουργία της γραμμής. Αν το bit αυτό είναι 1, τότε ο master διαβάζει από τον slave, ενώ αν είναι 0, τότε ο master γράφει στο slave. Στην κατάσταση που βρίσκεται το σύστημα το οποίο μελετάται ο master γράφει στον slave. Στη συνέχεια, ακολουθεί ένα bit, με το οποίο ο κάθε slave δηλώνει αν έχει όντως αυτή τη διεύθυνση ή όχι, με ένα "Ack" bit (Acknowledge bit). Αυτό επιτυγχάνεται κατεβάζοντας τη γραμμή SDA χαμηλά ή διατηρώντας τη γραμμή σταθερή. Πιο αναλυτικά, εάν η γραμμή κρατηθεί στην κατάσταση 1 από τον slave, στέλνεται από αυτόν ένα **negative acknowledge** (NAK) bit, ενώ αν η γραμμή αλλάξει στην κατάσταση 0 τότε στέλνεται ένα **acknowledge** (ACK) bit. Σε κάθε γραμμή υπάρχει μόνο ένας slave με τη συγκεκριμένη διεύθυνση, επομένως ο master επικοινωνεί με έναν slave κάθε φορά.

Έπειτα, ο master ή ο slave, ανάλογα με το σύστημα βρίσκεται σε κατάσταση read ή write, στέλνει ή διαβάζει τα δεδομένα. Στην περίπτωση μας, ο master στέλνει το byte 0x53 που αντιστοιχεί στο χαρακτήρα S. Μετά από κάθε byte δεδομένων, ο slave στέλνει πάλι ένα Ack/Nak bit

για το αν η διαδικασία ολοκληρώθηκε με επιτυχία. Η διαδικασία ολοκληρώνεται με ένα **stop condition**. Πιο συγκεκριμένα, η γραμμή SDA αφήνεται να επιστρέψει στην κατάσταση **high**, ενώ η SCL παραμένει στην ίδια κατάσταση.

Το stop condition μπορεί να παραλειφθεί και μετά το δεύτερο Ack bit, να σταλθεί πάλι ένα start condition (**repeated start**), εικόνα [1.22], έτσι ώστε να μπορούν να σταλούν περισσότερα του ενός byte κάθε φορά.



Σχήμα 1.22: Απεικόνιση των γραμμών SDA, SCL κατά τη διάρκεια επικοινωνίας με χρήση του πρωτοκόλλου I²C κατά την οποία πραγματοποιείται repeated start

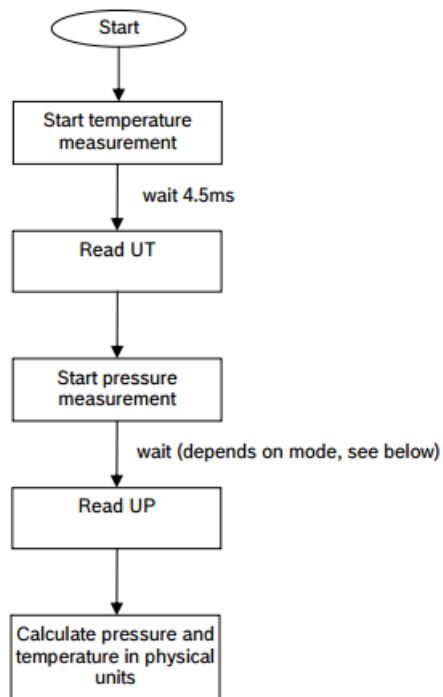
Ένα σκαρίφημα της διαδικασίας, με την οποία διαβάζουμε τα δεδομένα από τον αισθητήρα BMP085, φαίνεται στην εικόνα [1.23]. Ενώ στην εικόνα [1.24] φαίνονται αναλυτικά όλα τα βήματα της διαδικασίας αυτής.

Στην αρχή, στέλνεται το start condition και τη διεύθυνση όπως, αναλύθηκε στην παραπάνω παράγραφο για το πρωτόκολλο επικοινωνίας I²C. Στη συνέχεια, διαβάζεται η πίεση και η θερμοκρασία από τον αισθητήρα, αλλά τα δεδομένα μας δεν είναι βαθμονομημένα. Ο αισθητήρας έχει μνήμη τύπου EEPROM (Registers), στην οποία είναι αποθηκευμένες κάποιες σταθερές, με τις οποίες βαθμονομούνται τα δεδομένα που μετράει ο αισθητήρας. Στον πίνακα 1.3 παρατίθενται αναλυτικά όλα τα registers καθώς και ποιά σταθερά παίρνουμε από κάθε ένα από αυτά.

Τέλος, λόγω του ότι το πρωτόκολλο I²C του Arduino λειτουργεί στα 5V ενώ του αισθητήρα στα 3.3V, χρησιμοποιήθηκε ένας **bidirectional level shifter** [28], [27], [23], ο οποίος φαίνεται στην εικόνα [1.25]. Ο bidirectional level shifter μετατρέπει τα 5V που στέλνει το Arduino σε 3.3V ώστε ο αισθητήρας να λειτουργεί χωρίς να υπάρχει κίνδυνος καθώς η τάση 5V είναι μεγαλύτερη από αυτή που προτείνει ο κατασκευαστής. Από την άλλη τα 3.3V που στέλνει ο αισθητήρας μετατρέπονται σε 5V ώστε το Arduino να τα διαβάσει χωρίς πρόβλημα.

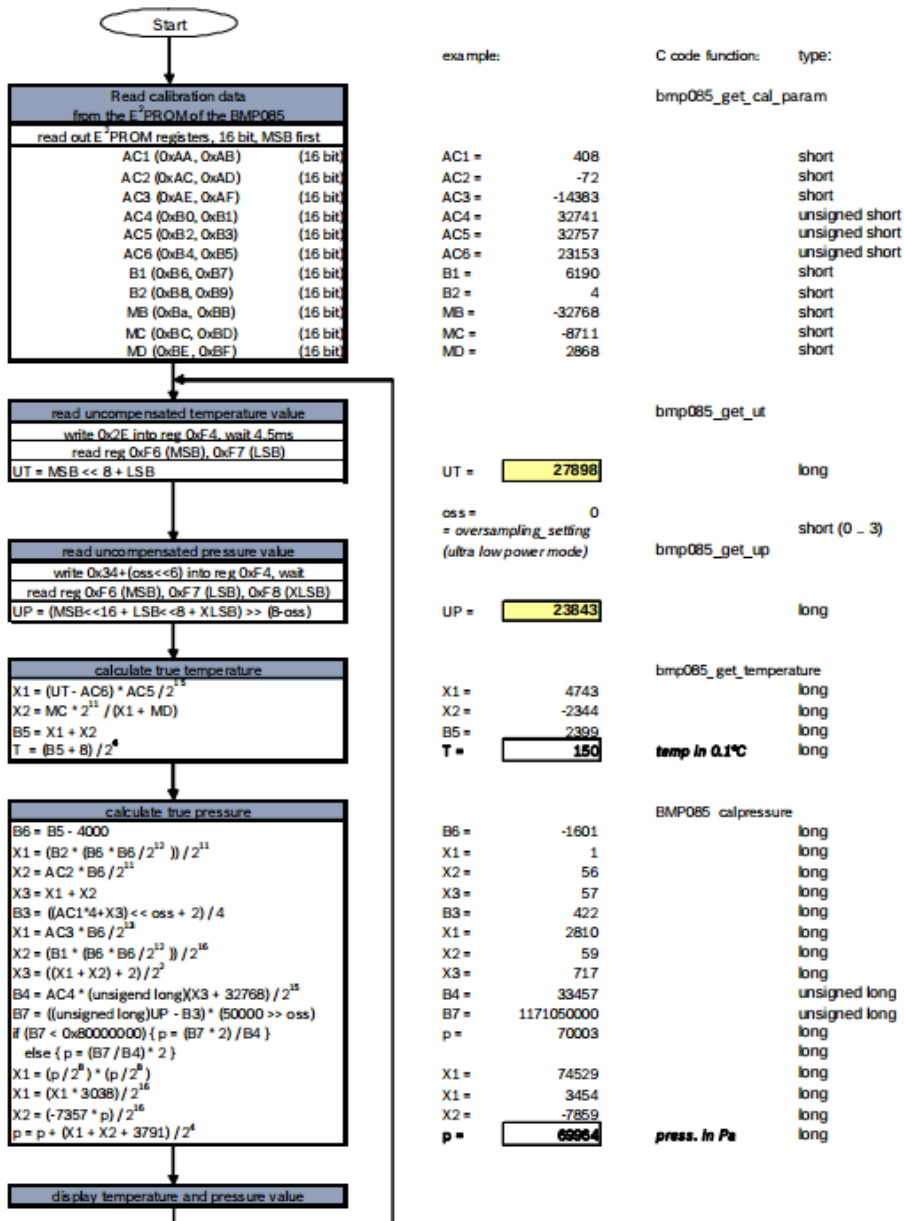
Parameter	BMP085 reg adr	
	MSB	LSB
AC1	0xAA	0xAB
AC2	0xAC	0xAD
AC3	0xAE	0xAF
AC4	0xB0	0xB1
AC5	0xB2	0xB3
AC6	0xB4	0xB5
B1	0xB6	0xB7
B2	0xB8	0xB9
MB	0xBA	0xBB
MC	0xBC	0xBD
MD	0xBE	0xBF

Πίνακας 1.3: Αναλυτικός πίνακας των registers του αισθητήρα BMP085 καθώς και ποιά σταθερά υπάρχει αποθηκευμένη σε ποιο register



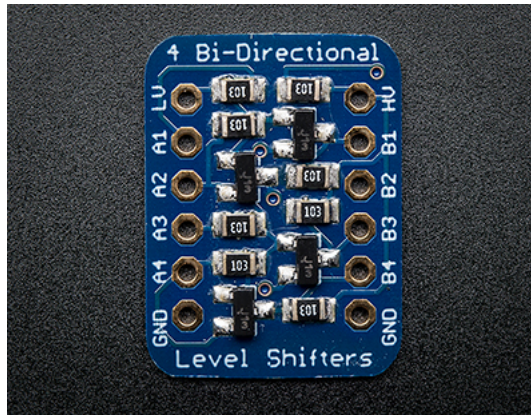
Σχήμα 1.23: Σκαρίφημα διαδικασίας λήψης δεδομένων από τον αισθητήρα BMP085

Calculation of pressure and temperature for BMP085

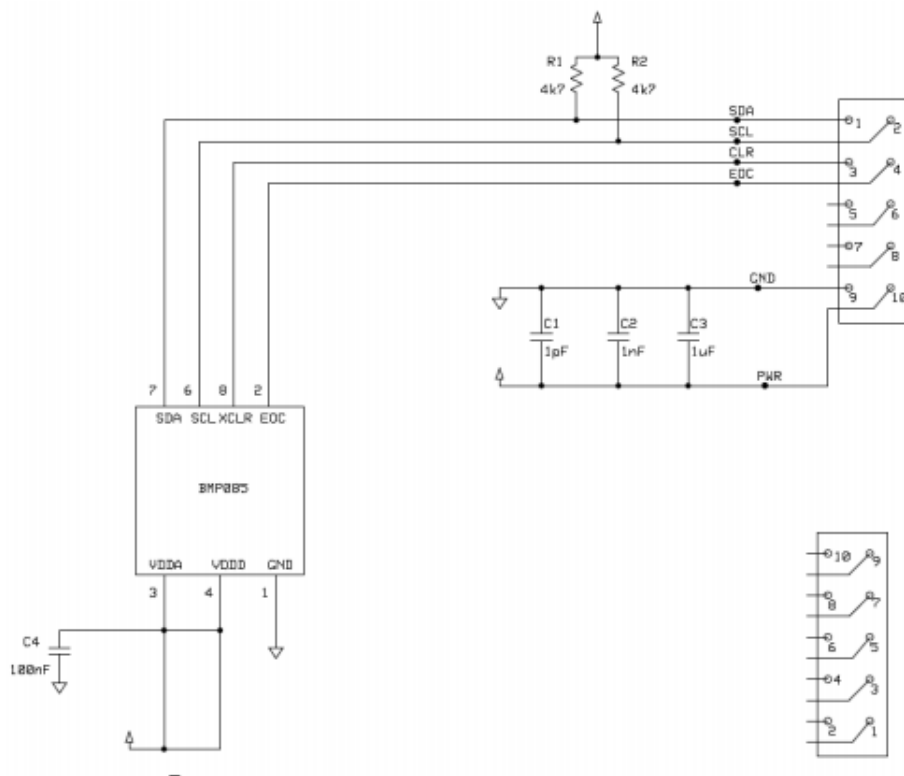


Σχήμα 1.24: Αναλυτική αναπαράσταση της διαδικασίας λήψης δεδομένων από τον αισθητήρα BMP085

Στην συνέχεια, παρατίθεται μία εικόνα [1.26], στην οποία φαίνεται το κύκλωμα πάνω στην πλακέτα AVR4201, τα pins της πλακέτας, καθώς και ένας πίνακας 1.4 με την αντιστοιχία των pins της πλακέτας και του Arduino.



Σχήμα 1.25: Adafruit bidirectional level shifter



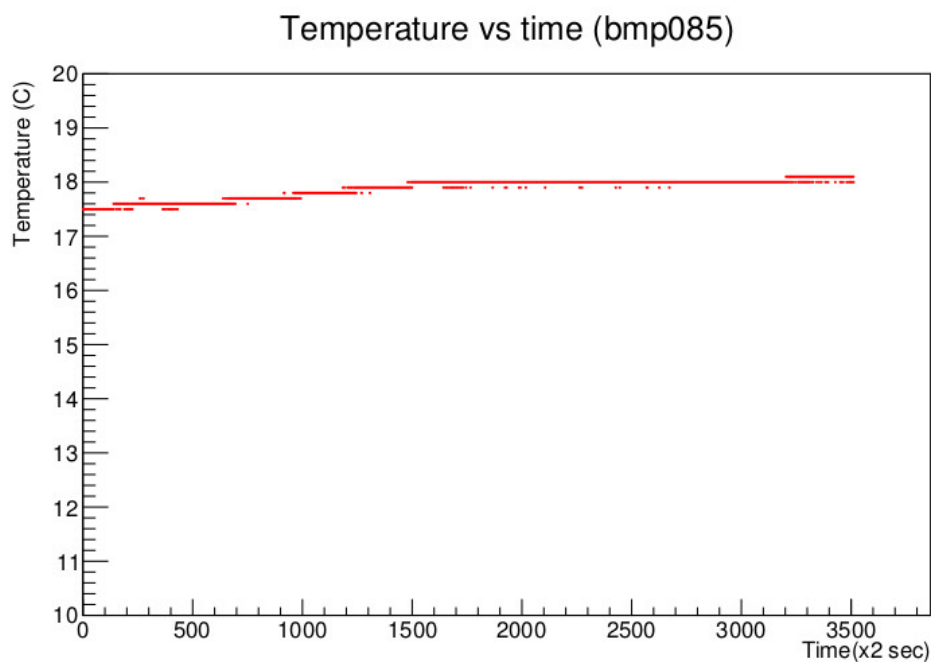
Σχήμα 1.26: Σχηματικό κυκλώματος αισθητήρα BMP085 πάνω στην πλακέτα AVR4201

AVR4201 pin	Arduino pin
1	20(SDA)
2	21(SCL)
9	GND
10	3V3

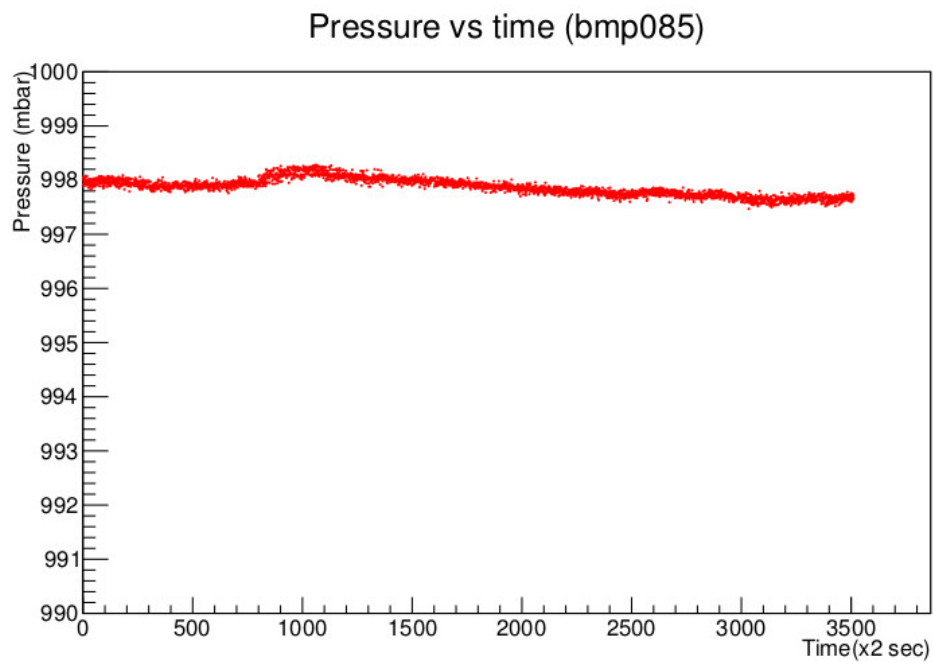
Πίνακας 1.4: Αντιστοιχία των pins του AVR4201 με αυτά του Arduino

B. Μετρήσεις

Στις εικόνες [1.27], [1.28] περιέχονται διαγράμματα θερμοκρασίας και πίεσης αντίστοιχα συναρτήσει του χρόνου, από μετρήσεις που πάρθηκαν στο εργαστήριο με τον αισθητήρα BMP085 σε ένα διάστημα 4 ωρών με μια μέτρηση ανά 2 δευτερόλεπτα. Οι μετρήσεις διακρίνονται από σταθερότητα και ο αισθητήρας κρίνεται κατάλληλος για χρήση στο σύστημα μας.



Σχήμα 1.27: Διαγράμμα θερμοκρασίας συναρτήσει του χρόνου για τον αισθητήρα BMP085 σε διάστημα 4 ωρών με μία μέτρηση ανά δύο δευτερόλεπτα



Σχήμα 1.28: Διαγράμμα πίεσης συναρτήσει του χρόνου για τον αισθητήρα BMP085 σε διάστημα 4 ωρών με μία μέτρηση ανά δύο δευτερόλεπτα

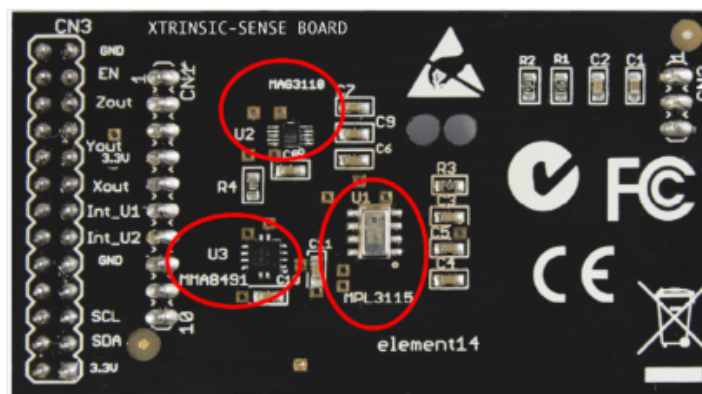
1.2.3 Αισθητήρας MPL3115A2

A. Περιγραφή



Σχήμα 1.29: Ψηφιακός αισθητήρας πίεσης και θερμοκρασίας MPL3115A2

Ο MPL3115A2 [25], εικόνα [1.29], είναι ένας ψηφιακός αισθητήρας πίεσης και θερμοκρασίας, ο οποίος χαρακτηρίζεται από την υψηλή του ακρίβεια και την χαμηλή του κατανάλωση. Όπως και ο BMP085 χρησιμοποιεί το πρωτόκολλο επικοινωνίας I²C, κάτι που τον κάνει ιδανικό για χρήση με μικροεπεξεργαστές/μικροελεγκτές. Στην περίπτωση μας χρησιμοποιήθηκε το **xtrinsic sensor board**[8] της **Element14**. Το συγκεκριμένο board έχει τρεις αισθητήρες επάνω του. Ωστόσο, στην εργασία αυτή έγινε χρήση μόνο του **MPL3115A2**. Οι άλλοι δύο αισθητήρες είναι ένα μαγνητόμετρο καθώς και ένα επιταχυνσιόμετρο. Στην εικόνα [1.30], φαίνεται το board αυτό ενώ οι τρεις αισθητήρες MPL3115A2, MAG3110 (μαγνητόμετρο) MMA8491 (επιταχυνσιόμετρο) είναι κυκλωμένοι για επισήμανση. Επάνω βρίσκεται ο MAG3110, κάτω δεξιά ο MPL3115A2 και κάτω αριστερά ο MMA8491. Ορισμένα από τα τεχνικά χαρακτηριστικά του αισθητήρα φαίνονται στους παρακάτω πίνακες [1.5], [1.6], [1.7].



Σχήμα 1.30: Xtrinsic sensor board της Element14 με τους αισθητήρες MAG3110, MMA8491, MPL3115A2

Parameter	Test Conditions	Min	Typ	Max	Unit
Measurement Range		-40		+85	°C
Temperature Accuracy	25 °C		±1		°C
	Over Temperature Range		±3		°C
Operating Temperature Range	-40		+85		°C

Πίνακας 1.5: Χαρακτηριστικά αισθητήρα θερμοκρασίας MPL3115A2

Parameter	Test Conditions	Min	Typ	Max	Unit
Measurement Range	Calibrated Range	50		110	kPa
	Operational Range	20		110	kPa
Pressure Reading Noise	1x Oversampel		19		Pa RMS
	128x Oversample		1.5		Pa RMS
Pressure Absolute Accuracy	50 to 110 kPa over 0 to 50 °C	-0.4		0.4	kPa
	50 to 110 kPa over -10 to 70 °C		±0.4		kPa
Pressure Relative Accuracy	Accuracy pressure change between constant 70 to 110 kPa at any constant temperature between -10 to 50 °C		±0.05		kPa
	Accuracy tempeprature change between -10 to 50 °C at any constant pressure between 50 to 100 kPa		±0.1		kPa
Pressure/Altitude Resolution	Barometer Mode	0.25	1.5		Pa
	Altimeter Mode	0.0625	0.3		m

Πίνακας 1.6: Χαρακτηριστικά αισθητήρα πίεσης MPL3115A2

Parameter	Test Conditions	Min	Typ	Max	Unit
I/O Supply Voltage		1.62	1.8	3.6	V
Operating Supply Voltage		1.95	2.5	3.6	V
Integrated Current 1 update per second	Highest Speed Mode Oversample=1		8.5		μA
	Standard Mode Oversample=16		40		
	High Resolution Mode Oversample = 128		265		
Max Current during Acquisition and Conversion	During Acquisition		2		μA
Supply Current Draing in STANDBY Mode	STANDBY Mode selected SBYB = 0		2		
Digital High Level Input Voltage SCL,SDA		0.75			V_{DDIO}
Digital Low Level Input Voltage				0.3	V_{DDIO}
High Level Output Voltage INT1, INT2	$I_0 = 500\mu\text{A}$	0.9			V_{DDIO}
Low Level Output Voltage INT1, INT2	$I_0 = 500\mu\text{A}$			0.1	V_{DDIO}
Low Level Output Voltage SDA	$I_0 = 500$			0.1	V_{DDIO}
Turn-on time	Highest Speed Mode			60	ms
	Highest Resolution Mode			1000	ms
Operating Temperature Range		-40	25	85	$^{\circ}\text{C}$

Πίνακας 1.7: Ηλεκτρικά χαρακτηριστικά του αισθητήρα πίεσης και θερμοκρασίας MPL3115A2

Στους δύο επόμενους πίνακες [1.8], [1.9], περιέχονται τα pins του board. Οι πίνακες αυτοί ακολουθούνται από έναν άλλο πίνακα [1.10] με την αντιστοιχία των pins της πλακέτας και του Arduino. Επειδή και αυτός ο αισθητήρας λειτουργεί με 3.3V, χρησιμοποιήθηκε bidirectional level shifter όπως και με τον αισθητήρα BMP085.

Pin No.	Pin Name	Description
1	EN	MMA8491 Enable Pin
2	ZOUT	MMA8491 Push-Pull Z-Axis Tilt Detection Output
3	YOUT	MMA8491 Push-Pull Y-Axis Tilt Detection Output
4	XOUT	MMA8491 Push-Pull X-Axis Tilt Detection Output
5	INT_MPL3115	MPL3115 Interrupt
6	INT_MAG3110	MAG3110 Interrupt
7	GND	Ground
8	NC	No Connection
9	SDA_SENSOR	I2C Slave Data Line
10	SCL_SENSOR	I2C Slave Clock Line

Πίνακας 1.8: Xtrinsic Sensor Board CN1 Pins

Pin No.	Pin Name	Description
1	VDD	+3.3V DC Power Supply
2	NC	No Connection
3	GND	Ground

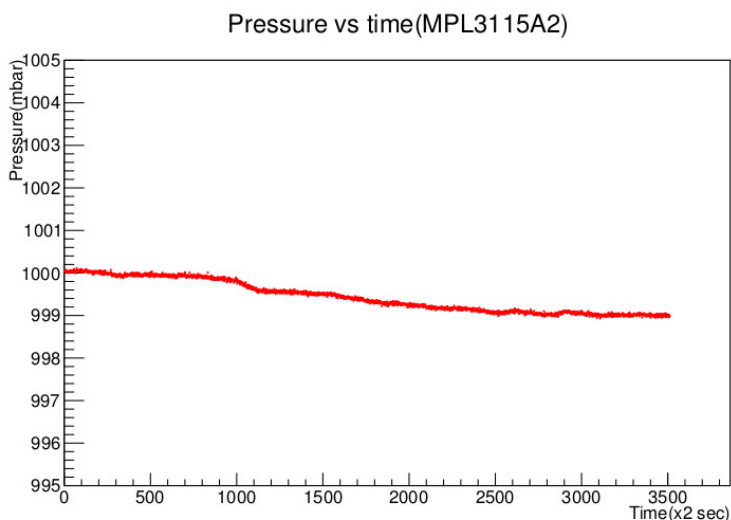
Πίνακας 1.9: Xtrinsic Sensor Board CN2 Pins

Xtrinsic Sensor Board Pin	Arduino Pin
CN1 Pin 7	GND
CN1 Pin 9	20(SDA)
CN1 Pin 10	21(SCL)
CN2 Pin 1	3V3
CN2 Pin 3	GND

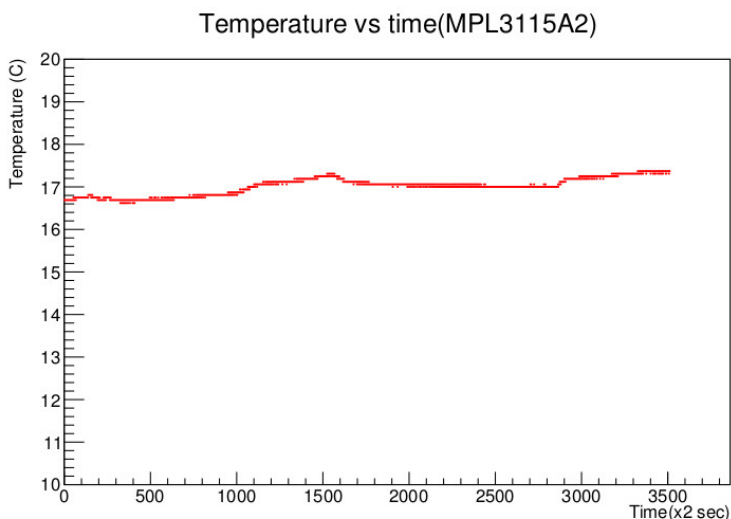
Πίνακας 1.10: Αντιστοίχιση των pins Xtrinsic sensor board με αυτά του Arduino

B. Μετρήσεις

Ακολουθούν δύο εικόνες [1.32], [1.31] με μετρήσεις πίεσης και θερμοκρασίας για τον αισθητήρα MPL3115A2. Οι μετρήσεις αυτές είναι για ένα διάστημα 4 ωρών με μία μέτρηση ανά 2 δευτερόλεπτα. Όπως και ο BMP085 έτσι αυτός χαρακτηρίζεται από μεγάλη σταθερότητα και κρίνεται ικανός για να χρησιμοποιηθεί στο σύστημα μας.



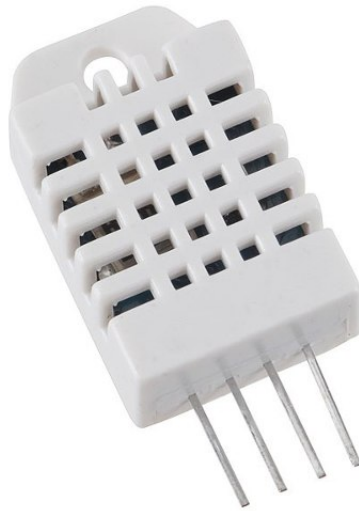
Σχήμα 1.31: Μετρήσεις πίεσης συναρτήσει του χρόνου για τον αισθητήρα MPL3115A2 για διάστημα 4 ωρών με μία μέτρηση ανά 2 δευτερόλεπτα



Σχήμα 1.32: Μετρήσεις θερμοκρασίας συναρτήσει του χρόνου για τον αισθητήρα MPL3115A2 για διάστημα 4 ωρών με μία μέτρηση ανά 2 δευτερόλεπτα

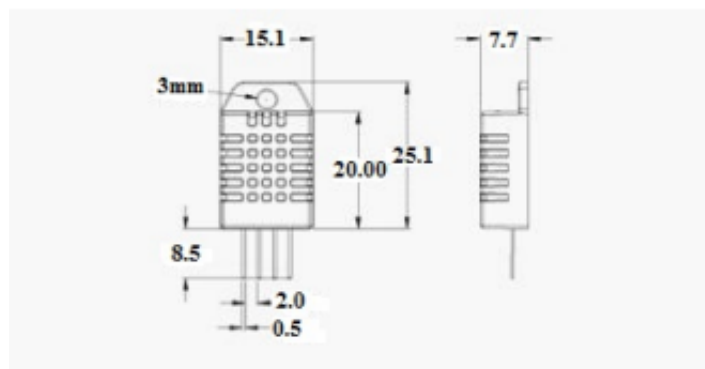
1.2.4 Αισθητήρας DHT22

A. Περιγραφή



Σχήμα 1.33: Ψηφιακός αισθητήρας θερμοκρασίας και υγρασίας DHT22

Ο αισθητήρας **DHT22** ή αλλιώς **AM2302** [20], εικόνα [1.33], είναι ένας ψηφιακός αισθητήρας μέτρησης υγρασίας και θερμοκρασίας. Η δυνατότητα μεταφοράς του σήματος σε μεγάλες αποστάσεις, καθώς και η χαμηλή κατανάλωση καθιστούν τον αισθητήρα αυτό ιδανικό για χρήση με Arduino. Στην παρακάτω εικόνα [1.34] φαίνονται ένα σχηματικό του. Η εικόνα ακολουθείται από έναν πίνακα [1.11] με τα pins του αισθητήρα αυτού .



Σχήμα 1.34: Σχηματικό του αισθητήρα θερμοκρασίας και υγρασίας DHT22

Pin	Function
1	VDD – power supply
2	DATA – signal
3	NULL
4	GND

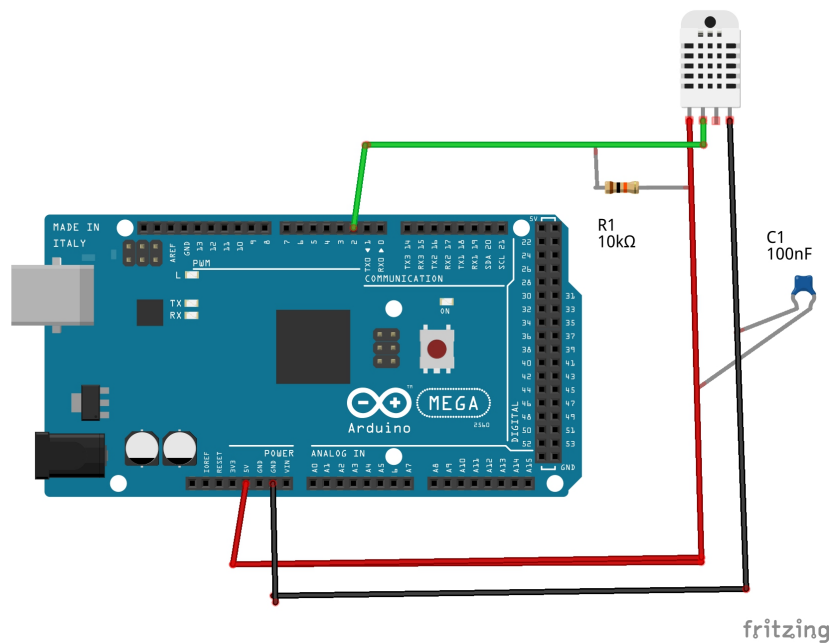
Πίνακας 1.11: Πίνακας με τα pins του αισθητήρα DHT22

Στον πίνακα που ακολουθεί [1.12] περιέχονται τα τεχνικά χαρακτηριστικά του αισθητήρα.

Parameter	Value
Power supply	3.3-6V DC
Output signal	Digital signal via single bus
Sensing element	polymer capacitor
Operating range	humidity 0 = 100%RH
	temperature -40 80 C
Accuracy	humidity $\pm 2\%RH$ (Max $\pm 5\%RH$)
	temperature $< \pm 0.5C$
Sensitivity	humidity 0.1%RH
	temperature 0.1C
Repeatability	humidity $\pm 1\%RH$
	temperature $\pm 0.2C$
Humidity hysteresis	$\pm 0.3\%RH$
Long-term Stability	$\pm 0.5\%RH/year$
Sensing period	Average: 2 sec
Interchangeability	fully interchangeable
Dimensions	small size 14*18*5.5 mm
	big size 22*28*5mm

Πίνακας 1.12: Τεχνικά χαρακτηριστικά του αισθητήρα θερμοκρασίας και υγρασίας DHT22

Για τη σύνδεση του αισθητήρα με το Arduino χρησιμοποιείται μία pull up αντίσταση 10kΩ, καθώς και ένας πυκνωτής 100 nF για φιλτράρισμα της τάσης εισόδου, όπως φαίνεται στην παρακάτω εικόνα [1.35].

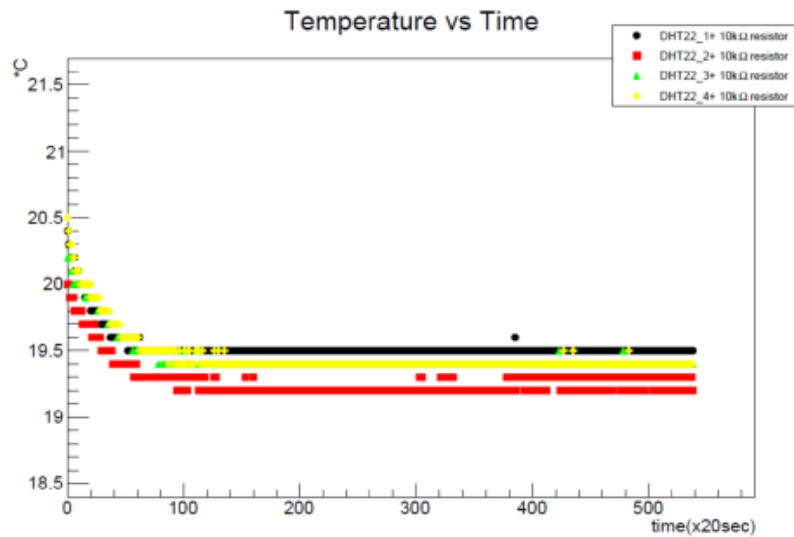


Σχήμα 1.35: Συνδεσμολογία που χρησιμοποιείται για τη σύνδεση του αισθητήρα DHT22 με το Arduino

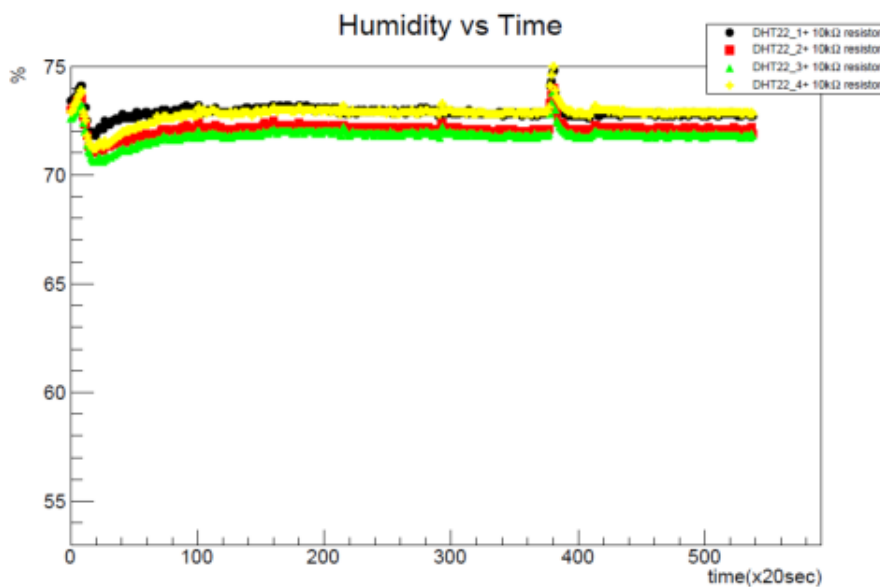
B. Μετρήσεις

Οι αισθητήρες DHT22 είχαν χρησιμοποιηθεί στη διπλωματική εργασία "Ανάπτυξη Συστήματος Μέτρησης Θερμοκρασίας και Υγρασίας βασισμένο σε Arduino και WinCC_OA" [32] όπου και προτείνεται η χρήση της συγκεκριμένης συνδεσμολογίας, εικόνα [1.35]. Αρχικά πραγματοποιήθηκαν μετρήσεις της υγρασίας και της θερμοκρασίας, και προσπάθειες αναπαραγωγής του προβλήματος που καταγράφηκε στην εν λόγω εργασία. Πιο συγκεκριμένα στις μετρήσεις της εργασίας αυτής, παρατηρήθηκε κατά την πρώτη περίπου ώρα, μία σταθερή πτώση της θερμοκρασίας και μια σταθερή άνοδος της υγρασίας. Στη συνέχεια, οι τιμές που προέκυπταν από τον αισθητήρα, σταθεροποιούνταν. Η συμπεριφορά αυτή φαίνεται στα διαγράμματα [1.36], [1.37].

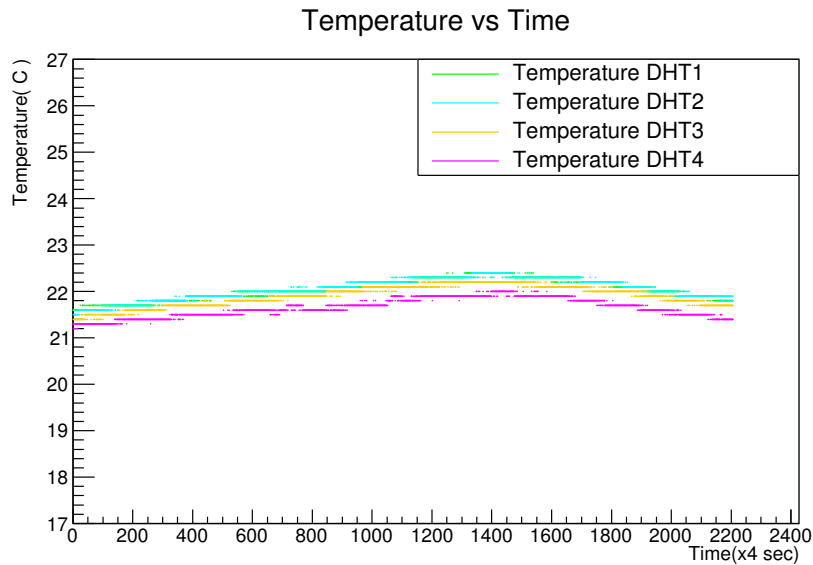
Στη συνέχεια ακολουθούν διαγράμματα από μετρήσεις που ελήφθησαν κατά την προσπάθεια αναπαραγωγής του φαινομένου αυτού, για διάστημα 1 ώρας και 30 λεπτών με μία μέτρηση ανά 4 δευτερόλεπτα. Τα αποτελέσματα των μετρήσεων αυτών φαίνονται στις εικόνες [1.38], [1.39].



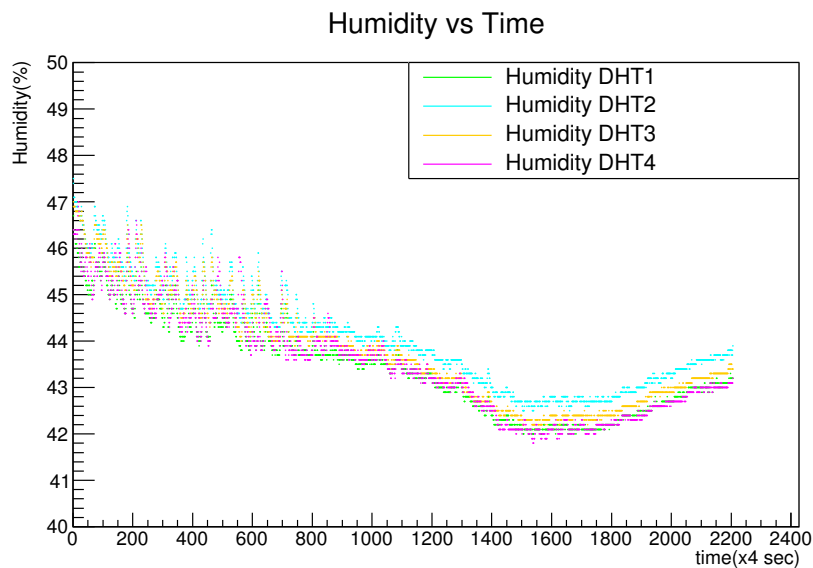
Σχήμα 1.36: Διαγράμματα θερμοκρασίας συναρτήσει του χρόνου από τη διπλωματική εργασία “Ανάπτυξη Συστήματος μέτρησης Θερμοκρασίας και Υγρασίας βασισμένο σε Arduino και WinCC_OA”



Σχήμα 1.37: Διαγράμματα υγρασίας συναρτήσει του χρόνου από τη διπλωματική εργασία “Ανάπτυξη Συστήματος μέτρησης Θερμοκρασίας και Υγρασίας βασισμένο σε Arduino και WinCC_OA”



Σχήμα 1.38: Διάγράμματα θερμοκρασίας συναρτήσει του χρόνου για τον αισθητήρα DHT22 κατά την προσπάθεια αναπαραγωγής του προβλήματος που επισημάνθηκε στην εργασία “Ανάπτυξη Συστήματος μέτρησης Θερμοκρασίας και Υγρασίας βασισμένο σε Arduino και WinCC_OA”



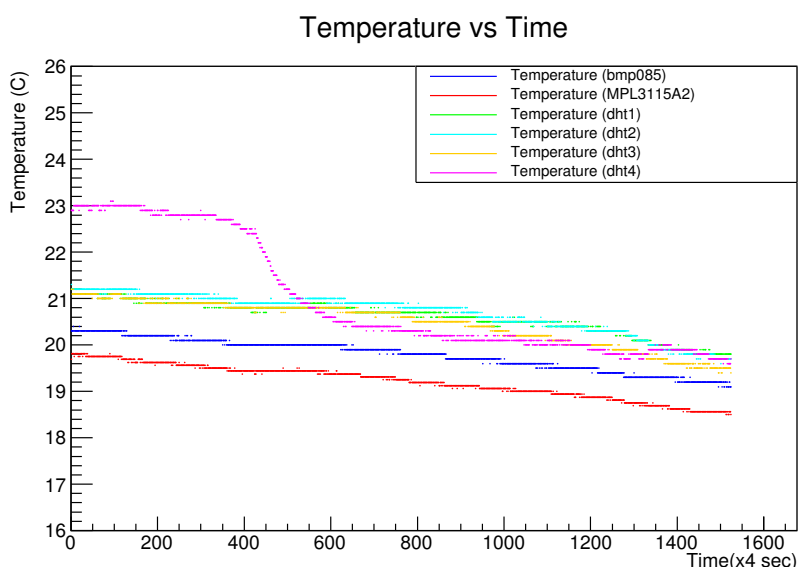
Σχήμα 1.39: Διάγράμματα υγρασίας συναρτήσει του χρόνου για τον αισθητήρα DHT22 κατά την προσπάθεια αναπαραγωγής του προβλήματος που επισημάνθηκε στην εργασία “Ανάπτυξη Συστήματος μέτρησης Θερμοκρασίας και Υγρασίας βασισμένο σε Arduino και WinCC_OA”

Παρατηρούνται διακυμάνσεις στις μετρήσεις, κυρίως σε αυτές της υγρασίας, ωστόσο η συμπεριφορά που είχε επισημανθεί στην εργασία “Ανάπτυξη Συστήματος Μέτρησης Θερμοκρασίας και Υγρασίας

βασισμένο σε Arduino και WinCC OA” δεν εμφανίζεται. Επομένως συμπεραίνεται ότι δεν υπήρχε κάποιο πρόβλημα με τους αισθητήρες. Πιθανώς το πρόβλημα να εντοπίζεται στη συνδεσμολογία, για παράδειγμα κάποιο προβληματικό καλώδιο.

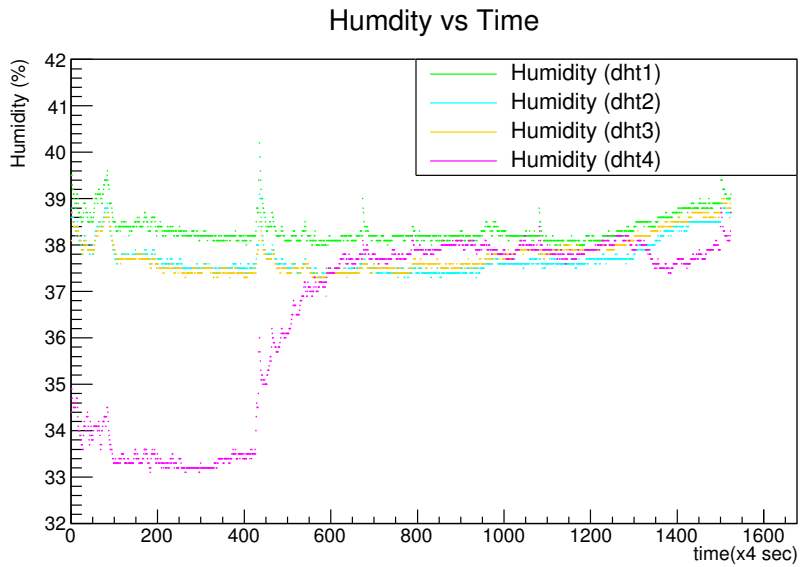
1.2.5 Κοινή σύνδεση όλων των αισθητήρων

Εφόσον πραγματοποιήθηκε έλεγχος της λειτουργίας όλων των αισθητήρων και επιβεβαιώθηκε η ορθή λειτουργία τους, το επόμενο βήμα ήταν η σύνδεση όλων των αισθητήρων σε μια κοινή συνδεσμολογία. Στις εικόνες που ακολουθούν [1.40], [1.41], [1.42], παρουσιάζονται διαγράμματα πίεσης υγρασίας και θερμοκρασίας από όλους τους αισθητήρες ταυτόχρονα.

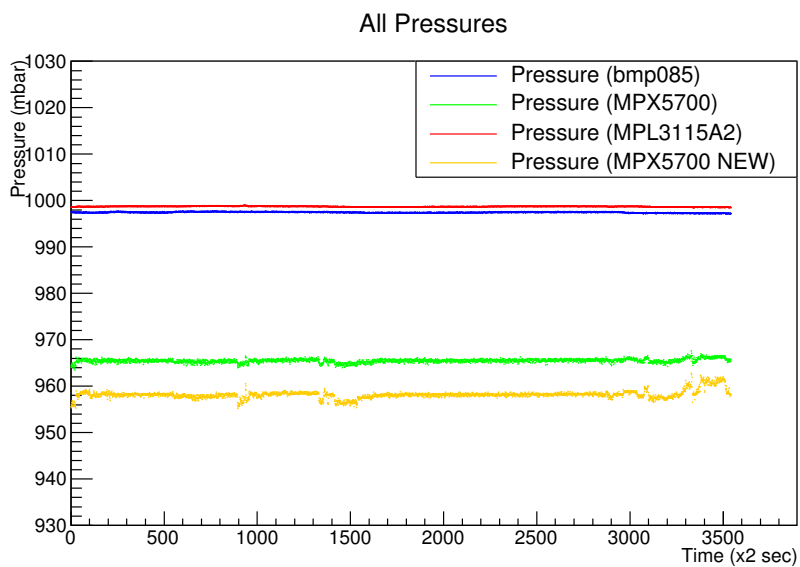


Σχήμα 1.40: Διάγραμμα θερμοκρασίας συναρτήσει του χρόνου για τους αισθητήρες BMP085, MPL3115A2, DHT22

Στα διαγράμματα θερμοκρασίας και υγρασίας, έχουμε απόκλιση του ενός αισθητήρα. Αυτό συνέβη, διότι ο αισθητήρας αυτός είχε αφηθεί πάνω στο πληκτρολόγιο του laptop κατά τη διάρκεια των μετρήσεων. Όταν αυτό έγινε αντιληπτό (στα 1600 secs ή ένδειξη 400 στο διάγραμμα) μετακινήθηκε και παρατηρήθηκε επαναφορά των τιμών των μετρήσεών του σε επίπεδα παρόμοια με αυτά των υπόλοιπων αισθητήρων, όπως φαίνεται στην ένδειξη 600 για τον χρόνο στο διάγραμμα. Στα δύο διαγράμματα αυτά, παρατηρείται ίδια συμπεριφορά από όλους τους αισθητήρες, δηλαδή οι καμπύλες τους εμφανίζουν τις ίδιες κλίσεις. Οι τιμές που καταγράφουν όλοι αισθητήρες βρίσκονται κοντά μεταξύ τους, με τη μεγαλύτερη διαφορά να μην ξεπερνάει την μία μονάδα όσον αφορά τη θερμοκρασία αλλά και την υγρασία, κάτι που θεωρείται ικανοποιητικό για το σύστημά μας.



Σχήμα 1.41: Διάγραμμα υγρασίας συναρτήσει του χρόνου για τους αισθητήρες DHT22



Σχήμα 1.42: Διάγραμμα πίεσης συναρτήσει του χρόνου για τους αισθητήρες BMP085, MPL3115A2,MPX5700

Παρόλα αυτά, μεγάλη διαφορά εμφανίζεται στο διάγραμμα πίεσης. Πιο συγκεκριμένα οι αναλογικοί αισθητήρες MPX5700 παρουσιάζουν μεγάλη απόκλιση σε σχέση με τους δύο ψηφιακούς αισθητήρες πίεσης. Οι αισθητήρες MPX5700 δεν χαρακτηρίζονται από μεγάλη ακρίβεια, αν κοιτάξει κανείς τον πίνακα με τα χαρακτηριστικά τους, πίνακας 1.1 θα δει ότι η τιμή της είναι $2.5\% V_{FSS}$. Ωστόσο το συγκεκριμένο σύστημα έχει σα σκοπό τη μέτρηση της συμπεριφοράς κάποιας ποσότητας (π.χ. πίεσης) και όχι τόσο την τιμή της, δηλαδή τα σχετικά σφάλματα παίζουν μεγαλύτερο ρόλο από τα απόλυτα. Για το λόγο αυτό, η ευαισθησία του αισθητήρα αυτού παίζει μεγαλύτερο ρόλο η οποία όπως φαίνεται στον ίδιο πίνακα, είναι $6.4 \mu V/kPa$ η οποία είναι κοντά στη διακριτική ικανότητα του Arduino άρα η συμπεριφορά του κρίνεται ικανοποιητική.

1.2.6 Σύγκριση των αισθητήρων με το HMP233 της Vaisala

A. Περιγραφή Vaisala HMP233



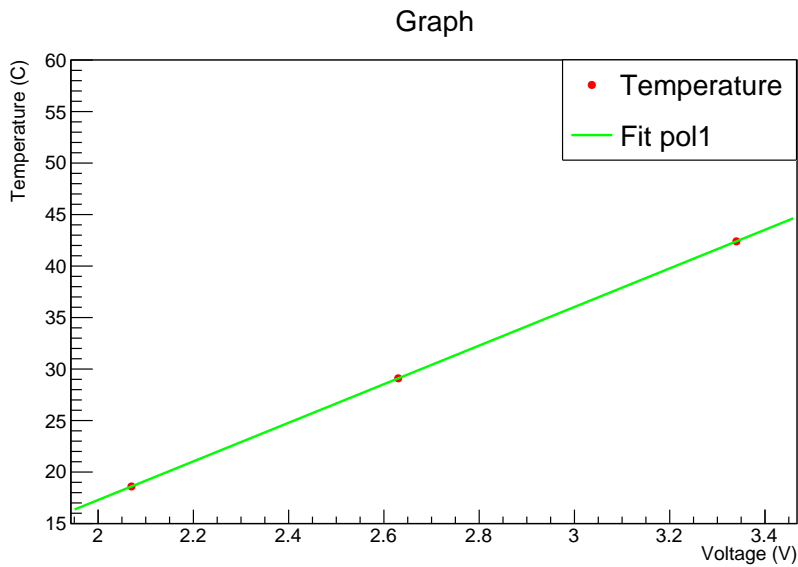
Σχήμα 1.43: Vaisala HMP233

Το **HMP233** της **Vaisala** [30], όπως φαίνεται στην εικόνα [1.43], είναι ένα εργαλείο μέτρησης θερμοκρασίας και απόλυτης υγρασίας το οποίο είναι βασισμένο σε ένα μικροεπεξεργαστή. Η μέτρηση της θερμοκρασίας γίνεται με έναν αισθητήρα **PT100**. Το HMP233 διαθέτει μία οθόνη, στην οποία προβάλλει τις μετρήσεις. Ακόμη, διαθέτει δυο αναλογικές εξόδους τάσης, μία για τη θερμοκρασία και μία για την υγρασία. Η εξαιρετικά υψηλή του ακρίβεια τον καθιστά ιδανικό για σύγκριση των μετρήσεων του με τις μετρήσεις του συστήματος που αναπτύσσεται από την εργασία αυτή. Ωστόσο, ο κατασκευαστής δε δίνει κάποια συνάρτηση μεταφοράς για την μετατροπή του αναλογικού σήματος εξόδου σε θερμοκρασία ή υγρασία. Αυτό συμβαίνει διότι η τιμή της εξόδου είναι πλήρως προσαρμόσιμη. Μπορεί, για παράδειγμα, ο χρήστης να ορίσει το διάστημα $0-60 \text{ }^\circ\text{C}$ να αντιστοιχεί σε μια έξοδο $0-10 \text{ V}$ κλπ. Έτσι, λοιπόν, έπρεπε αρχικά να υπολογιστεί η συνάρτηση μεταφοράς. Επειδή το HMP233 διαθέτει δυο εξόδους, απαιτείται ο υπολογισμός δυο

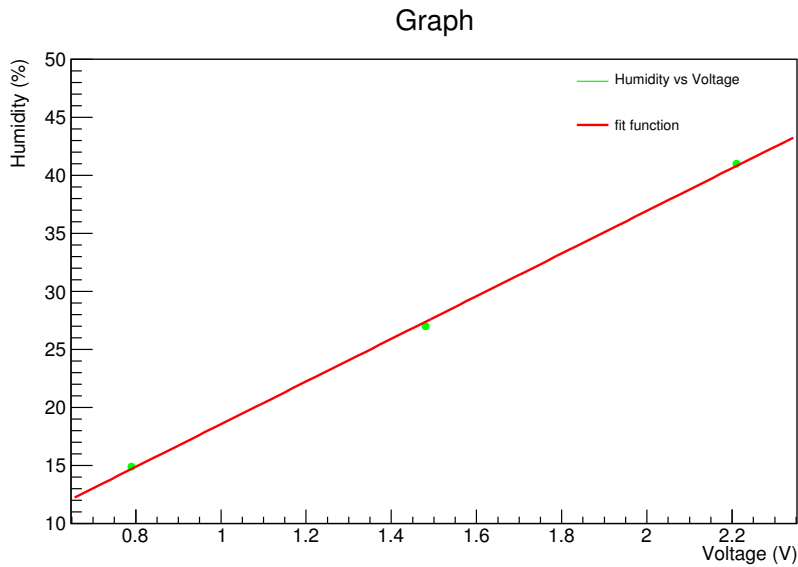
συναρτήσεως μεταφοράς, μία για τη θερμοκρασία και μία για την υγρασία. Για το σκοπό αυτό, για τρεις τιμές της θερμοκρασίας και υγρασίας, μετρήθηκε η τάση στην αντίστοιχη αναλογική έξοδο. Στη συνέχεια, έγιναν 2 διαγράμματα [1.44], [1.45] με τα σημεία αυτά και τραβήχθηκε μια ευθεία στο κάθε διάγραμμα. Οι συναρτήσεις που προκύπτουν είναι:

$$\text{Humidity} = -28,024 * \text{Voltage} + 100,744$$

$$\text{Temperature} = 18,7398 * \text{Voltage} - 20,1894$$

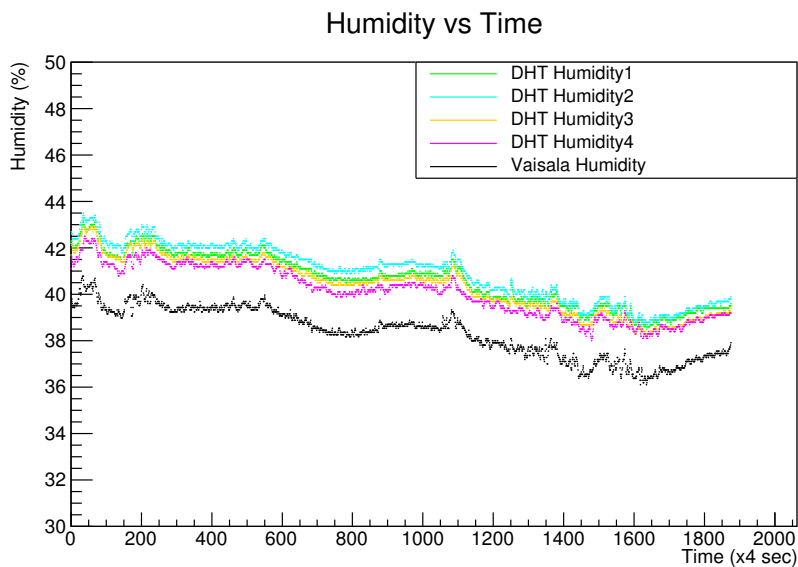


Σχήμα 1.44: Διάγραμμα θερμοκρασίας συναρτήσει της τάσης για το HMP233 της Vaisala και η συνάρτηση μεταφοράς του

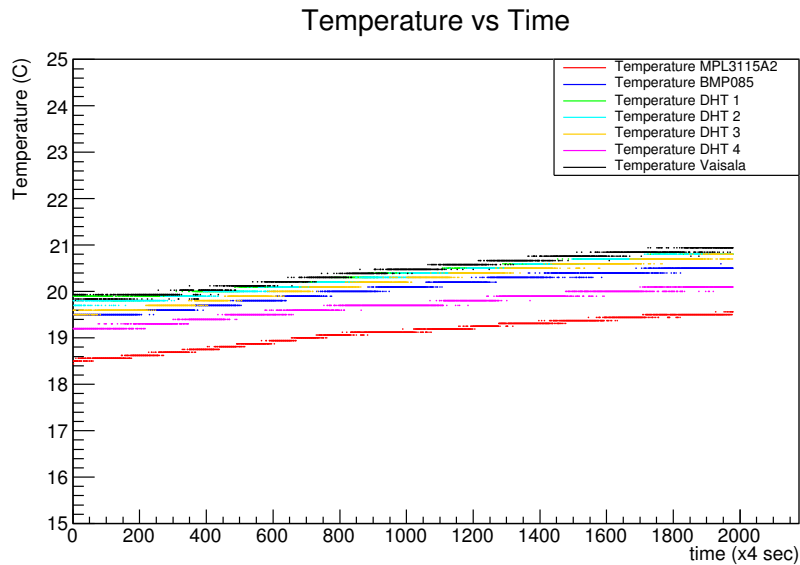


Σχήμα 1.45: Διάγραμμα υγρασίας συναρτήσει της τάσης για το HMP233 της Vaisala και η συνάρτηση μεταφοράς του

Στις εικόνες [1.46], [1.47] φαίνονται οι μετρήσεις για υγρασία και θερμοκρασία αντίστοιχα που πάρθηκαν με τους αισθητήρες του συστήματος και αυτές με το Vaisala HMP233.



Σχήμα 1.46: Διάγραμμα υγρασίας συναρτήσει του χρόνου για όλους τους αισθητήρες μαζί με το HMP233 της Vaisala

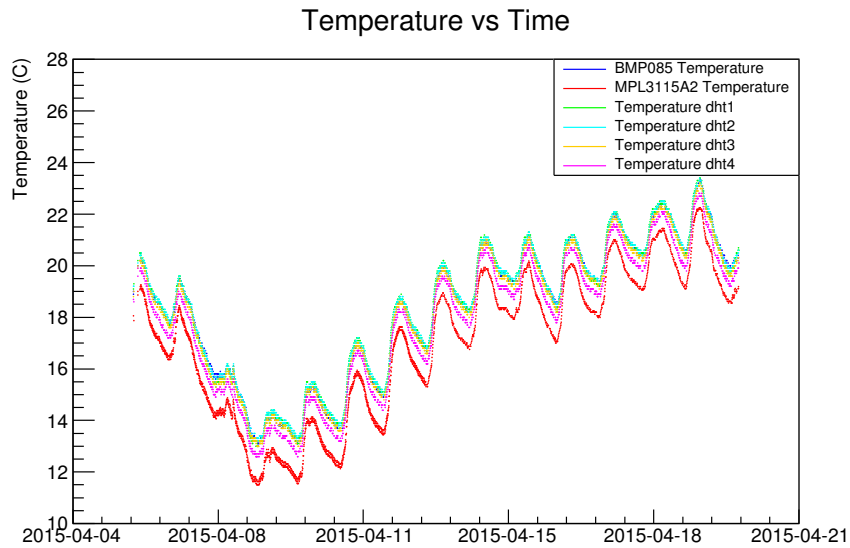


Σχήμα 1.47: Διάγραμμα θερμοκρασίας συναρτήσει του χρόνου για όλους τους αισθητήρες μαζί με το HMP233 της Vaisala

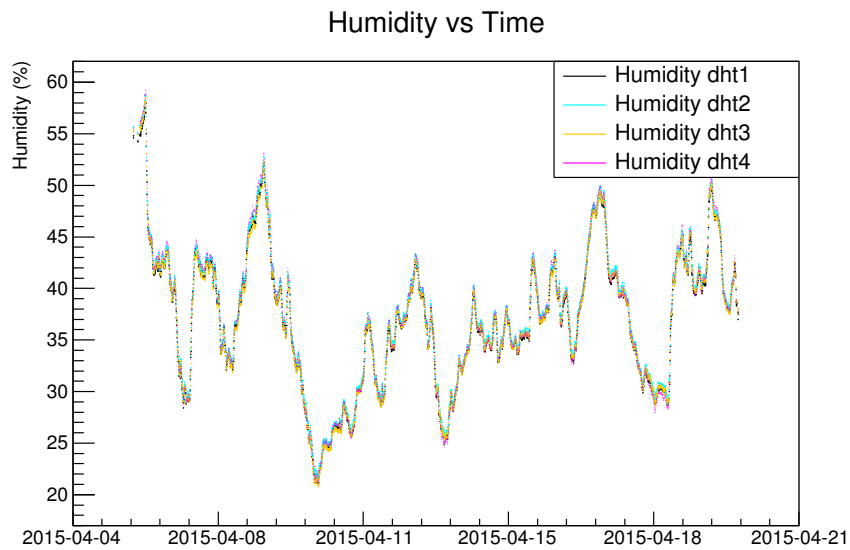
Θα πρέπει να σημειωθεί εδώ ότι για τη θερμοκρασία, η απόκλιση των μετρήσεων με τη συνάρτηση μεταφοράς από αυτές με το Vaisala HMP233 ήταν +0,5, ενώ η αντίστοιχη απόκλιση για την υγρασία ήταν -0,7. Από τα διαγράμματα προκύπτει ότι οι μετρήσεις του συστήματος μας είναι αρκετά ικανοποιητικές και κοντά στις πραγματικές τιμές.

1.2.7 Τελικές μετρήσεις

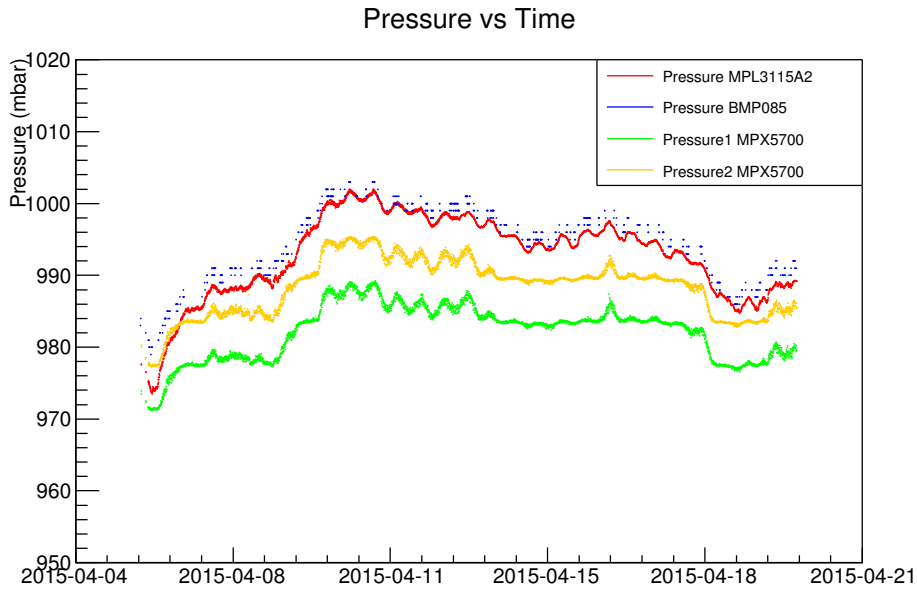
Τέλος, το σύστημά μας, αποτελούμενο από το Arduino και συνολικά 8 αισθητήρες (4 DHT22, 2 MPX5700, 1 BMP085, 1 MPL3115A2) που διαθέταμε, έπρεπε να εξεταστεί ως προς τη μακροπρόθεσμη σταθερότητα. Έτσι οι τελικές μετρήσεις πάρθηκαν αφήνοντας το σύστημά σε λειτουργία για ένα διάστημα 2 εβδομάδων. Οι μετρήσεις φαίνονται στις παρακάτω εικόνες όπου μπορεί να δει κανείς ότι το σύστημα είναι σταθερό σε βάθος χρόνου [1.48], [1.49], [1.50].



Σχήμα 1.48: Διάγραμμα θερμοκρασίας συναρτήσει του χρόνου για όλους τους αισθητήρες για διάστημα δύο εβδομάδων



Σχήμα 1.49: Διάγραμμα υγρασίας συναρτήσει του χρόνου για όλους τους αισθητήρες για διάστημα δύο εβδομάδων

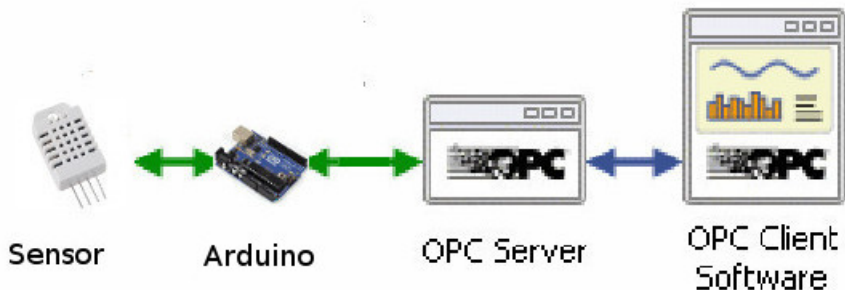


Σχήμα 1.50: Διάγραμμα πίεσης συναρτήσει του χρόνου για όλους τους αισθητήρες για διάστημα δύο εβδομάδων

1.3 OPC Server

Στην ενότητα αυτή, παρουσιάζεται ο Arduino OPC (Open Platform Communications) Server. Συγκεκριμένα, μετά τη γενική περιγραφή ενός συστήματος που χρησιμοποιεί το OPC πρωτόκολλο για επικοινωνία, περιγράφεται αναλυτικά ο OPC Server για Arduino καθώς και η μεθοδολογία εγκατάστασης και χρήσης του.

1.3.1 Εισαγωγή



Σχήμα 1.51: Διάγραμμα επικοινωνίας με χρήση του πρωτοκόλλου Open Platform Communications (OPC)

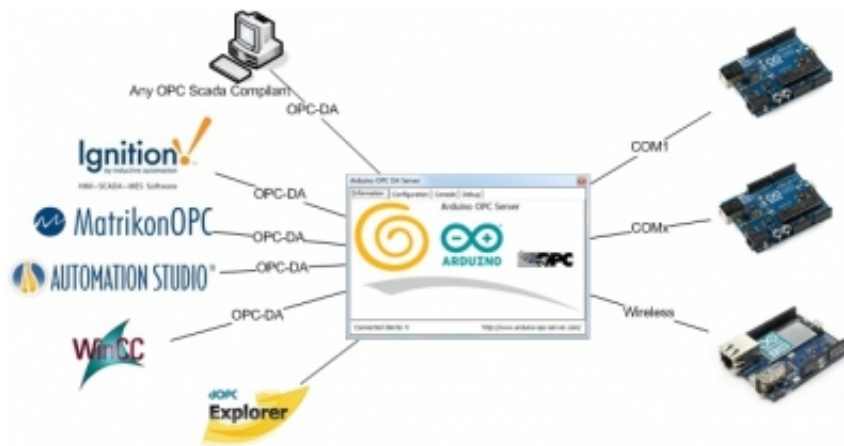
Το **OPC** (Open Platform Communications) πρωτόκολλο [11] είναι μία σειρά από προδιαγραφές που διέπουν την επικοινωνία μεταξύ λογισμικού (κυρίως προγραμμάτων σε περιβάλλον Windows) και εξωτερικών συσκευών. Η επικοινωνία πραγματοποιείται σε ζεύγη **διακομιστή (Server)** και **πελάτη (Client)**. Οι προδιαγραφές αυτές πρωτοδημιουργήθηκαν το 1996 με το όνομα **"OLE"** (Object Linking and Embedding), με σκοπό την αλληλεπίδραση λογισμικού **SCADA** (Supervisory Control And Data Acquisition) με ένα **PLC** (Programmable Logic Controller, ενός μικροϋπολογιστή δηλαδή που ελέγχει μία συσκευή). Ωστόσο, με την πάροδο του χρόνου, το πρωτόκολλο επικοινωνίας εξελίχθηκε και πλέον χρησιμοποιείται από χιλιάδες συσκευές στον χώρο του αυτομάτου ελέγχου. Το OLE είναι ένα πρωτόκολλο που δημιουργήθηκε από τη Microsoft για την σύνδεση και ενσωμάτωση αρχείων και αντικειμένων, γι αυτό το OPC χρησιμοποιείται κυρίως από προγράμματα Windows. Ωστόσο, τα τελευταία χρόνια, έχει αρχίσει μια προσπάθεια για την χρήση του OPC και από άλλα λειτουργικά συστήματα.

Ένας πελάτης OPC μπορεί να είναι ένα οποιοδήποτε πρόγραμμα, το οποίο θέλει να επικοινωνήσει με μία συσκευή. Ο ρόλος του διακομιστή είναι να μετατρέπει το αίτημα αυτό του πελάτη σε μια μορφή την οποία η συσκευή μπορεί να καταλάβει. Με τον τρόπο αυτό, ο πελάτης μπορεί να επικοινωνήσει με τη συσκευή στέλνοντάς της κάποια εντολή μέσω του διακομιστή ή διαβάζοντας δεδομένα από αυτή.

Ένα από τα μεγάλα πλεονεκτήματα του OPC είναι το ότι οι προδιαγραφές αυτές είναι προσβάσιμες σε όλους έτσι ο καθένας μπορεί να φτιάξει έναν πελάτη ή έναν διακομιστή καθώς έχει στη διάθεση του τα στοιχεία τα οποία πρέπει να έχει καθένας από αυτούς ώστε να επικοινωνεί με χρήση του πρωτοκόλλου αυτού. Επιπλέον, οι προδιαγραφές είναι κοινές για όλους τους διακομιστές και τους πελάτες, επιτρέποντας σε οποιοδήποτε πελάτη να συνδεθεί με οποιονδήποτε διακομιστή. Με τον τρόπο αυτό το κόστος για τους κατασκευαστές συσκευών διατηρείται χαμηλό, καθώς μόνο ένας διακομιστής είναι απαραίτητος και έτσι η συσκευή μπορεί να επικοινωνήσει με οποιοδήποτε λογισμικό που παρέχει σύνδεση με OPC. Οι προμηθευτές λογισμικού, από την άλλη, συμπεριλαμβάνοντας ένα πελάτη OPC στο πρόγραμμά τους, του δίνουν τη δυνατότητα να συνδεθεί με οποιαδήποτε συσκευή που διαθέτει OPC διακομιστή. Τέλος, οι χρήστες που θέλουν να συνδεθούν με κάποια συσκευή, έχουν μια πληθώρα προγραμμάτων μέσα από την οποία μπορούν να επιλέξουν καθώς είναι σίγουρο ότι, εφόσον το πρόγραμμα αυτό υποστηρίζει OPC επικοινωνία, θα συνδεθεί με τη συσκευή που επιθυμούν.

1.3.2 Arduino OPC Server

Για το Arduino, υπάρχει ένας διακομιστής, ο οποίος δημιουργήθηκε από τον **Ildefonso Martínez Marchena** και διατίθεται δωρεάν στο διαδίκτυο μέσω της ιστοσελίδας <http://www.st4makers.com/> [17]. Ο διακομιστής αυτός είναι πλήρως συγχρονισμένος με τις προδιαγραφές του OPC



Σχήμα 1.52: Arduino OPC Server

και μπορεί να συνδεθεί με οποιοδήποτε πρόγραμμα SCADA, δίνοντας έτσι τη δυνατότητα στο χρήστη να χρησιμοποιήσει το Arduino με μια πληθώρα προγραμμάτων αυτομάτου ελέγχου. Κάποια από τα χαρακτηριστικά του είναι:

- Δυνατότητα ταυτόχρονης σύνδεσης με πολλές πλακέτες Arduino οποιουδήποτε είδους.
- Ταυτόχρονη χρήση σειριακής, Ethernet αλλά και wifi επικοινωνίας.
- Μεγάλη ταχύτητα και ισχύ.
- Ο διακομιστής είναι συμβατός με JSON (Javascript Object Notation) όταν γίνεται χρήση του Arduino YUN. Μπορεί να χρησιμοποιηθεί οποιοδήποτε πρόγραμμα για λήψη δεδομένων και περιήγησης στο διαδίκτυο.

Για να είναι δυνατή η χρήση του διακομιστή με το Arduino, πρέπει να συμπεριλαμβάνεται η βιβλιοθήκη OPC στα σκίτσα που θέλουν να χρησιμοποιήσουν τον διακομιστή. Τόσο η βιβλιοθήκη όσο και ο διακομιστής διατίθενται δωρεάν απο την ίδια ιστοσελίδα.

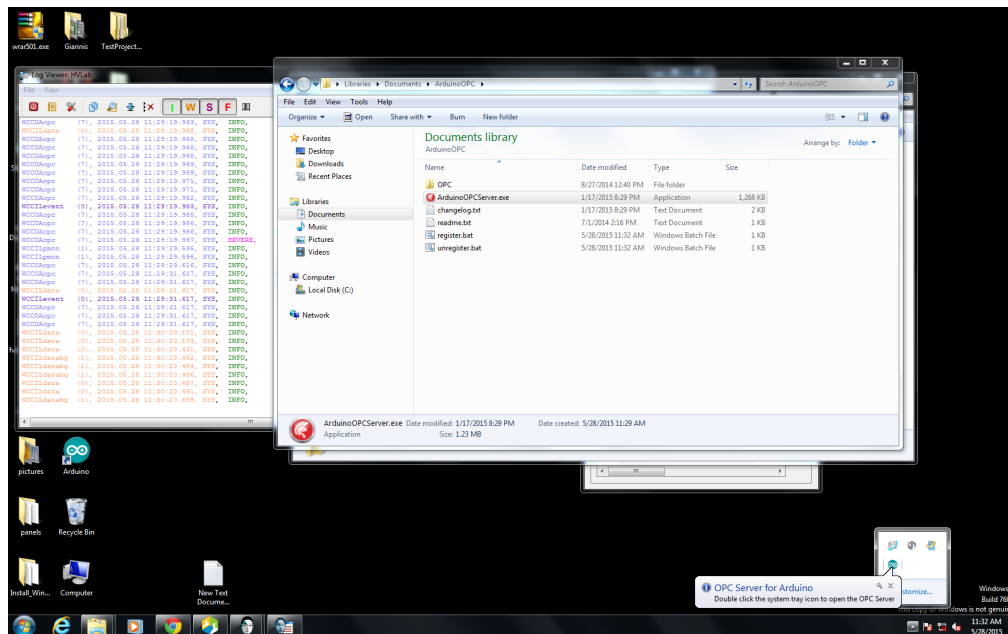
1.3.3 Εγκατάσταση και χρήση του Arduino OPC Server

Για να γίνει δυνατή η αλληλεπίδραση με το Arduino μέσω ενός προγράμματος SCADA, όπως είναι το WinCC OA που θα χρησιμοποιηθεί στην παρούσα εργασία, πρέπει χρησιμοποιηθεί ο Arduino OPC Server. Αρχικά πρέπει να γίνει λήψη του Server από την ιστοσελίδα που αναφέρθηκε στην προηγούμενη ενότητα, ο οποίος θα αποθηκευτεί στον υπολογιστή μας ως αρχείο "zip". Είναι σημαντικό να γίνει λήψη της πιο πρόσφατης έκδοσης του server, η οποία τη δεδομένη στιγμή είναι η έκδοση 1.6. Στη συνέχεια πρέπει να γίνει εξάγωγή των περιεχομένων του

αρχείου αυτού σε ένα φάκελο ώστε να είναι όλα μαζί. Στο φάκελο αυτό, θα περιέχεται ένας φάκελος με όνομα OPC, ο οποίος περιέχει τη βιβλιοθήκη με την οποία μπορεί το Arduino να επικοινωνήσει με το Server. Ο φάκελος αυτός θα πρέπει να μεταφερθεί στο φάκελο libraries όπως αναφέρθηκε στην ενότητα 1.1.4. Στον φάκελο θα πρέπει να περιέχονται τρία ακόμα αρχεία:

- **ArduinoOPCServer**
- **chanelog**
- **readme**

Κάνοντας δεξί κλικ στο αρχείο ArduinoOPCServer και επιλέγοντας "Run as Administrator" δημιουργούνται δύο καινούργια αρχεία register.bat και unregister.bat. Ο server ανοίγει με τον τρόπο που φαίνεται στην εικόνα [1.53].



Σχήμα 1.53: Εκκίνηση Arduino OPC Server

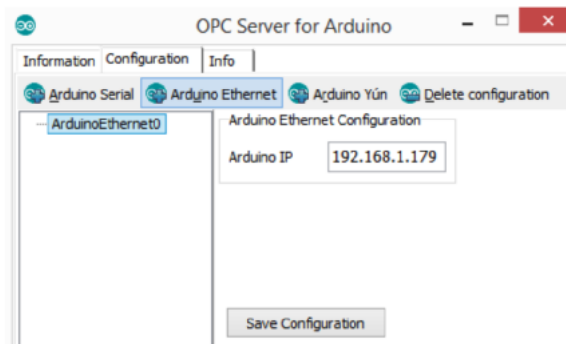
Στην οθόνη εμφανίζεται το κύριο παράθυρο του Arduino OPC Server εικόνα [1.54].

Στο Configuration tab, εικόνα [1.55] γίνεται επιλογή Arduino Ethernet. Δημιουργείται το ArduinoEthernet0 και στη συνέχεια εισάγεται η IP Address του Arduino. Η IP Address αυτή θα πρέπει να είναι ίδια με αυτή που έχει δηλωθεί στο Sketch το οποίο έχει "φορτωθεί" στο Arduino. Για παράδειγμα στην εικόνα [1.55] εισάγεται η IP 192.168.1.179

Στη συνέχεια πατάμε το κουμπί "Save Configuration". Έπειτα στο φάκελο στον οποίο είχε γίνει εξαγωγή όλων των αρχείων, στο



Σχήμα 1.54: Κύριο παράθυρο Arduino OPC Server



Σχήμα 1.55: Arduino OPC Server Configuration

αρχείο register.bat επιλέγεται "Run as administrator". Ο Server έχει εγκατασταθεί.

Για να μπορεί το Arduino να συνδεθεί με το Server θα πρέπει να βρίσκονται στο ίδιο subnet. Αν και τα δυο είναι συνδεδεμένα σε κάποιο router τότε απλά πρέπει η IP που θα δοθεί στο Arduino να είναι ίδια με αυτές που δίνει το router. Για παράδειγμα αν το router δίνει διευθύνσεις 192.168.1.X θα πρέπει στο sketch να ορισθεί μία IP για το Arduino η οποία να είναι και αυτή της μορφής 192.168.1.X. Αν το Arduino συνδέεται κατευθείαν στον υπολογιστή ή συνδέεται μέσω κάποιου switch τότε θα πρέπει να ορισθεί στον υπολογιστή μία IP. Έστω ότι όπως στην προηγούμενη εικόνα έχει δοθεί στο Arduino την IP 192.168.1.179. Τότε επιλέγεται στον πίνακα ελέγχου→Network and Internet→View network status and tasks. Στη συνέχεια local area connection→Properties→Internet Protocol Version 4 και επιλέγεται "Use the following IP Address". Γίνεται εισαγωγή της επιθυμητής IP, θα πρέπει αυτή να είναι μία IP στο ίδιο subnet με την IP του Arduino π.χ. 192.168.1.1. Στην συνέχεια η γραμμή, Subnet Mask γίνεται αυτόματα 255.255.255.0, αφήνεται όπως είναι και σε όλα τα παράθυρα πατιέται το κουμπί OK.

Τέλος πρέπει να γίνει εγκατάσταση της βιβλιοθήκης OPC για το Arduino. Αυτό μπορεί να γίνει είτε όπως αναφέρθηκε πιο πάνω

μεταφέροντας τη βιβλιοθήκη στο φάκελο libraries είτε μέσω του Arduino IDE. Αφού ανοιχθεί το Arduino IDE στα menu που βρίσκονται στην κορυφή του παραθύρου του IDE γίνεται επιλογή Sketch→Import Library→ Add Library. Στο παράθυρο που ανοίγει γίνεται επιλογή του φακέλου στον οποίο έγινε εξαγωγή το zip αρχείο. Το Arduino μας τώρα μπορεί να συνδεθεί με το Server.

Κεφάλαιο 2

Ανάπτυξη Συστήματος με WinCC_OA

Στο κεφάλαιο αυτό, παρουσιάζεται το WinCC_OA, ένα πρόγραμμα με χρήση στην κατασκευή συστημάτων αυτομάτου ελέγχου. Γίνεται αναφορά στο SLOW Control SYstem (SLOCSY) [19] και στις νέες λειτουργίες που προστέθηκαν κατά τη διάρκεια της εργασίας αυτής. Τέλος περιγράφεται αναλυτικά η διαδικασία σύνδεσης του Arduino με το WinCC_OA και το slow control system.

2.1 WinCC_OA

2.1.1 Εισαγωγή

Το WinCC_OA ανήκει στην κατηγορία προγραμμάτων SCADA (Supervisory Control and Data Acquisition) τα οποία είναι συστήματα βιομηχανικού αυτομάτου ελέγχου. Το κυριότερο χαρακτηριστικό τους, είναι ότι αποτελούνται από ένα κεντρικό σταθμό εργασίας (Master Station) και πολλούς επιμέρους τοπικούς σταθμούς οι οποίοι ελέγχουν επιμέρους στοιχεία και μονάδες μίας εγκατάστασης. Το Master Station επικοινωνεί και ελέγχει τις επιμέρους αυτές μονάδες και συλλέγει δεδομένα. Τα δεδομένα αυτά μπορεί στη συνέχεια να σταλούν σε άλλους σταθμούς εργασίας μέσα σε ένα τοπικό δίκτυο (LAN (Local Area Network)) ή σε σημεία εκτός του δικτύου αυτού μέσω κάποιου συστήματος τηλεπικοινωνίας.

Στον τομέα της Φυσικής Υψηλών Ενεργειών η ανάγκη για έλεγχο των ανιχνευτών εξ αποστάσεως είναι επιτακτική, γεγονός που καθιστά τη χρήση συστημάτων τύπου SCADA αναγκαία. Από την πληθώρα συστημάτων SCADA έχει επικρατήσει η χρήση του WinCC_OA το οποίο είναι ένα πρόγραμμα που χρησιμοποιείται για την κατασκευή συστημάτων αυτομάτου ελέγχου. Το φιλικό του για το χρήστη περιβάλλον, η εύκολη παραμετροποίησή του καθώς και η δυνατότητά του για εύκολη σύνδεση εξωτερικών συσκευών με το λογισμικό (OPC) για συλλογή δεδομένων και περαιτέρω επεξεργασία αποτέλεσαν βασικά κριτήρια για την επιλογή του έναντι των υπολοίπων συστημάτων.

2.1.2 Βασικά Χαρακτηριστικά του WinCC_OA

Στη συνέχεια παρουσιάζονται ορισμένα από τα χαρακτηριστικά του WinCC_OA

- **Γραφικό Περιβάλλον (Graphics EDItor(GEDI))**

Στο γραφικό περιβάλλον (GEDI) ο χρήστης έχει τη δυνατότητα να δημιουργήσει μια σειρά απο παράθυρα (panels) τα οποία μπορεί να εμπλουτίσει με μία πληθώρα γραφικών στοιχείων όπως κουμπιά, πλαίσια, πίνακες κλπ. Στη συνέχεια ο χρήστης μπορεί γράφοντας κώδικα να καθορίσει πλήρως την λειτουργία των γραφικών στοιχείων αυτών και κατά συνέπεια των παραθύρων. Η μεγάλη ποικιλία σε γραφικά που παρέχει το WinCC_OA κάνει την επικοινωνία και χρήση των εξωτερικών συσκευών εύκολη.

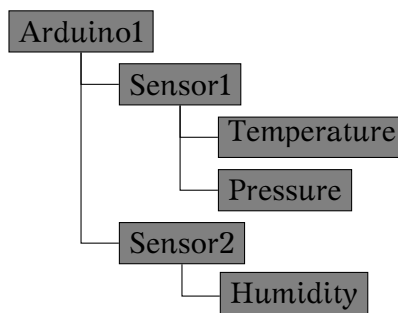
- **Κείμενα Ελέγχου (Control Scripts)**

Το WinCC_OA παρέχει τη δυνατότητα στο χρήστη να φτιάξει δικά του Control Scripts τα οποία εκτελούνται είτε από κάποιο panel είτε σαν ανεξάρτητη διαδικασία. Τα scripts γράφονται σε μία γλώσσα η οποία μοιάζει σε δομή και λογική με την C ωστόσο παρέχει πολύ περισσότερες λειτουργίες και συναρτήσεις σε σχέση με αυτή. Με τον τρόπο αυτό ο χρήστης κάνει το WinCC_OA να συμπεριφέρεται με συγκεκριμένο τρόπο ανάλογα με τις συνθήκες και μπορεί έτσι να προσθέσει στο πρόγραμμα περαιτέρω λειτουργίες τις οποίες δεν είχε όταν πρωτοεγκαταστάθηκε στον υπολογιστή.

- **Βάση Δεδομένων (Database)**

Το WinCC_OA διαθέτει μια εσωτερική βάση δεδομένων όπου μπορεί να αποθηκεύει δεδομένα τα οποία διαβάζει από τις εξωτερικές συσκευές με τις οποίες συνδέεται. Τα δεδομένα ομαδοποιούνται σε Data Points (DP) ενός προκαθορισμένου είδους Data Point Type (DPT). Το είδος των δεδομένων DPT περιγράφει τη δομή των δεδομένων ενός συγκεκριμένου είδους συσκευής. Τα DP περιέχουν τα δεδομένα και τις πληροφορίες για μία συγκεκριμένη συσκευή. Κάθε DP περιέχει μία σειρά από Data Point Elements (DPE) κάθε ένα από τα οποία αντιπροσωπεύει μία συγκεκριμένη τιμή ή πληροφορία για μία συγκεκριμένη συσκευή. Για παράδειγμα ένα Arduino μπορεί να έχει έναν αισθητήρα πίεσης-θερμοκρασίας και έναν υγρασίας. Το Arduino αυτό μπορεί να αποτελεί ένα DataPoint Type. Έστω ότι έχει δημιουργηθεί ένα Data Point με όνομα Arduino1, είδους (DPT) Arduino. Το Data Point αυτό, έχει τη δομή του DPT Arduino. Περιέχει δηλαδή πληροφορίες και τιμές για τους δύο αυτούς αισθητήρες. Αν συνδεθεί κάποιο άλλο Arduino το οποίο είχε συνδεδεμένους άλλους αισθητήρες θα πρέπει να δημιουργηθεί ένα άλλο DP διαφορετικού DPT. Τέλος, κάθε αισθητήρας θα έχει κάποια DataPoint Elements π.χ. ο πρώτος θα έχει 2 (Temperature, Pressure) και ο δεύτερος θα έχει 1 (Humidity). Όλη αυτή η αναπαράσταση είναι παράδειγμα για να φανεί η λογική με την οποία είναι δομημένο το

WinCC_OA. Ο κάθε χρήστης μπορεί να ομαδοποιήσει τα δεδομένα του όπως τον βολεύει. Το παράδειγμα αυτό μπορεί να φανεί και στο παρακάτω σχήμα [2.1]. Έτσι, για κάθε μεταβλητή για την οποία θέλουμε να πάρουμε την τιμή από μια εξωτερική συσκευή ή για κάθε μεταβλητή την οποία θέλουμε να θέσουμε ίση με μια τιμή την συνδέουμε με ένα DataPoint Element.



Σχήμα 2.1: WinCC_OA Data Structure

- **Σύνδεση με εξωτερικές συσκευές**

Η σημαντικότερη ίσως λειτουργία του WinCC_OA είναι η δυνατότητά του να συνδέεται με εξωτερικές συσκευές. Έτσι παρέχεται η δυνατότητα στο χρήστη να ζητήσει κάποια πληροφορία από τις εξωτερικές συσκευές, για παράδειγμα στην περίπτωση του συστήματός μας να ζητήσει τις τιμές των περιβαλλοντικών συνθηκών που μετρούν οι αισθητήρες. Στη συνέχεια, ο χρήστης επιλέγει μέσω του WinCC_OA πώς θα χειριστεί τα δεδομένα αυτά, έχοντας τη δυνατότητα να τα αποθηκεύσει στη βάση δεδομένων η οποία αναφέρθηκε πιο πάνω. Επίσης η δυνατότητα επικοινωνίας με τις συσκευές επιτρέπει στον χρήστη να τις χειριστεί εφόσον αυτές υποστηρίζουν τέτοια λειτουργία. Μπορεί να ανοίξει ή να κλείσει κάποια συσκευή κλπ. Ο τρόπος επικοινωνίας ποικίλει ανάλογα με το ποιο πρωτόκολλο υποστηρίζει η κάθε συσκευή. Για παράδειγμα, στη δική μας περίπτωση, το Arduino θα συνδεθεί με το WinCC_OA μέσω σύνδεσης OPC όπως αναφέρθηκε πιο πάνω. Το λογισμικό μας δηλαδή διαθέτει έναν OPC Client ο οποίος θα συνδεθεί με τον OPC Server και έτσι το WinCC_OA θα έχει τη δυνατότητα αλληλεπίδρασης με το Arduino. Το WinCC_OA προσφέρει τη δυνατότητα σύνδεσης μέσω των περισσότερων πρωτοκόλλων επικοινωνίας συσκευής-λογισμικού όπως TLS, MODBUS κ.α.

2.1.3 JCOP Framework

Στα περισσότερα project, στη φυσική υψηλών ενεργειών που χρησιμοποιούν το WinCC_OA γίνεται χρήση του JCOP Framework (Joint COntrols Project Framework) [10]. Το JCOP είναι μία σειρά από panels, control scripts και γενικότερα εργαλείων για τον WinCC_OA τα οποία δημιουργήθηκαν από κοινού από τις ομάδες των 4 πειραμάτων του LHC

(Large Hadron Colider) καθώς και με τη συνεισφορά και άλλων φυσικών και μηχανικών στο CERN. Σκοπός της δημιουργίας του είναι η παροχή κατευθυντήριων γραμμών για να δημιουργούνται ανεξάρτητα προγράμματα τα οποία να μπορούν ωστόσο όταν χρειαστεί να χρησιμοποιηθούν μαζί.

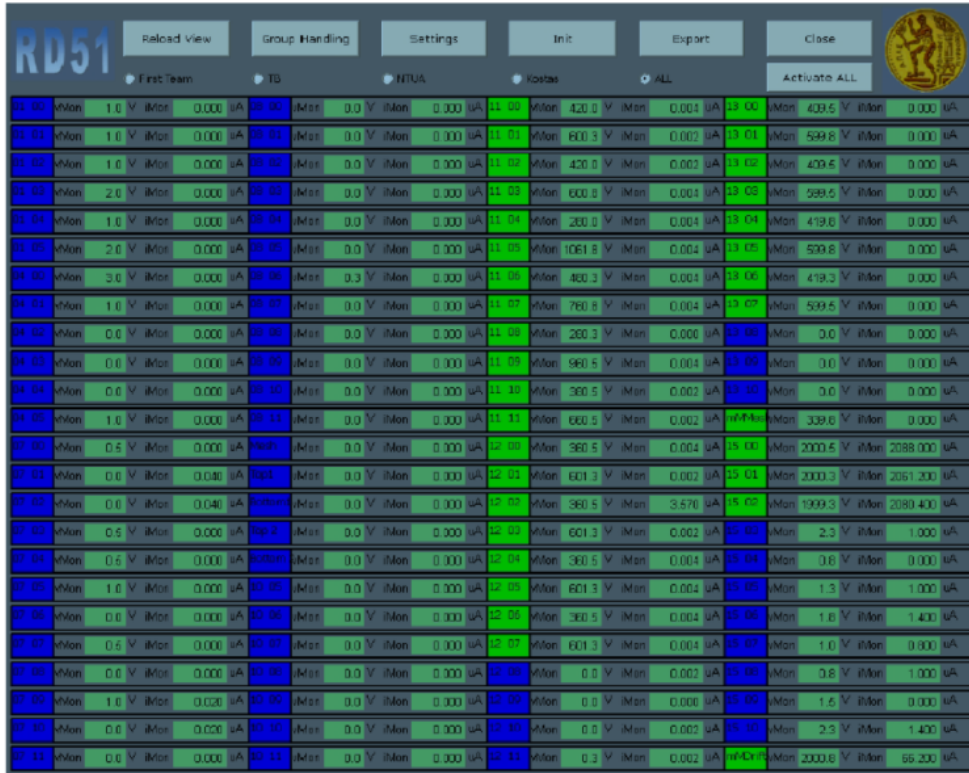
Όπως αναφέρθηκε και πιο πάνω το JCOP είναι ουσιαστικά μία σειρά από εργαλεία τα οποία έχει στη διάθεσή του ο χρήστης και κάνουν τη σύνδεση των διαφόρων συσκευών με το WinCC_OA εφικτή και σίγουρη. Τα εργαλεία που παρέχονται από το JCOP χρησιμοποιούνται σε δομικός λίθος στη δημιουργία των διαφόρων projects. Παρέχουν επίσης δυνατότητα για εύκολη ένταξη νέων συσκευών σε ήδη υπάρχοντα projects με την άμεση δημιουργία νέων datapoints και όπως είπώθηκε πιο πάνω τη δυνατότητα ενσωμάτωσης ενός project σε ένα άλλο, καθώς τα βασικά τους στοιχεία είναι κοινά.

2.2 Slow Control System

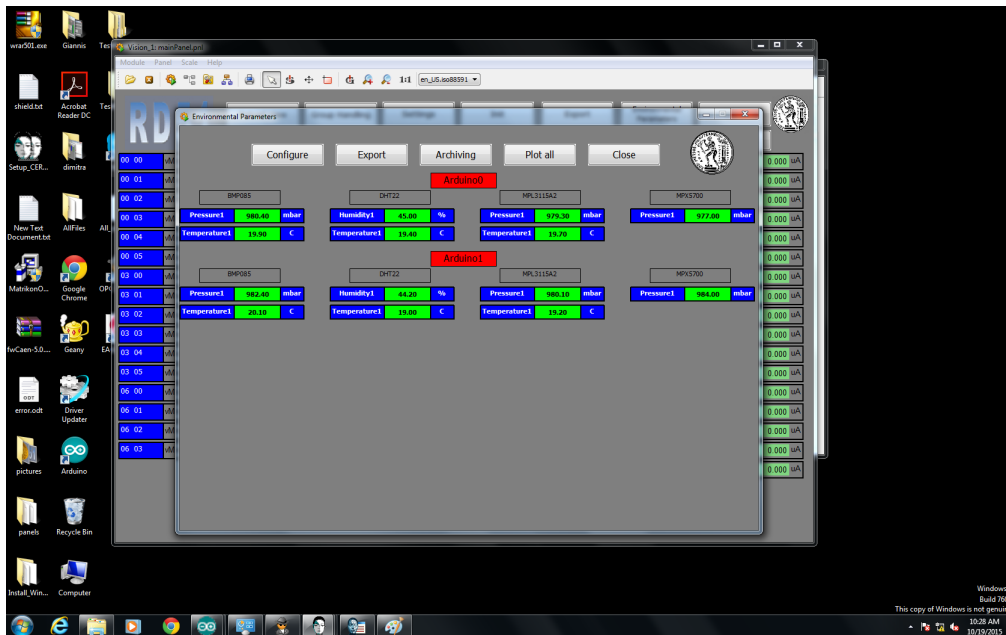
Στο εργαστήριο, υπήρχε ένα project στο PVSS, το οποίο είναι μια προηγούμενη έκδοση του WinCC_OA, με σκοπό την παρακολούθηση και συλλογή δεδομένων από τους ανιχνευτές. Το project αυτό είχε αναπτυχθεί στα πλαίσια της μεταπτυχιακής διπλωματικής εργασίας "Micro-megas Telescope for the RD51 Test Beams" [33]. Το project αυτό μεταφέρθηκε από το PVSS στο WinCC_OA στα πλαίσια της διπλωματικής εργασίας "Ανάπτυξη συστήματος μέτρησης θερμοκρασίας και υγρασίας βασισμένο σε Arduino και WinCC_OA" [32]. Το κύριο παράθυρο του συστήματος αυτού φαίνεται στην παρακάτω εικόνα [2.2]. Περισσότερες πληροφορίες για τις λειτουργίες του project αυτού βρίσκονται στην [33] και στο σχετικό note [19].

Σε αυτό το κομμάτι της παρούσας εργασίας σκοπός ήταν η κατασκευή ενός project στο WinCC_OA το οποίο να συνδέεται με το Arduino και να παίρνει δεδομένα από αυτό. Στη συνέχεια το project αυτό έπρεπε να ενσωματωθεί στο ήδη υπάρχον Slow Control System. Όπως προαναφέρθηκε, πολύ σημαντικό ρόλο σε όλη αυτή τη διαδικασία παίζει ο OPC Server ο οποίος λειτουργεί σαν συνδετικός κρίκος μεταξύ Arduino και WinCC_OA. Το κύριο παράθυρο του project φαίνεται στην εικόνα [2.3].

Στο παράθυρο αυτό, φαίνονται οι μετρήσεις όλων των αισθητήρων. Κάνοντας δεξί κλικ σε οποιοδήποτε από τα πεδία στα οποία αναγράφονται οι μετρήσεις ενός αισθητήρα, ανοίγει ένα παράθυρο με ένα διάγραμμα με την τιμή της μέτρησης αυτής συναρτήσει του χρόνου. Ακόμα, στο παράθυρο υπάρχει μία σειρά από κουμπιά, τα οποία προσδίδουν περαιτέρω δυνατότητες στο project μας. Υπάρχει το κουμπί "Plot All", το οποίο δίνει τη δυνατότητα στο χρήστη να ανοίξει ένα παράθυρο με ένα διάγραμμα με περισσότερες από μία καμπύλες. Πιο συγκεκριμένα, πατώντας το κουμπί, ανοίγει ένα άλλο μικρό παράθυρο, στο οποίο ο χρήστης μπορεί να επιλέξει το είδος των διαγραμμάτων που

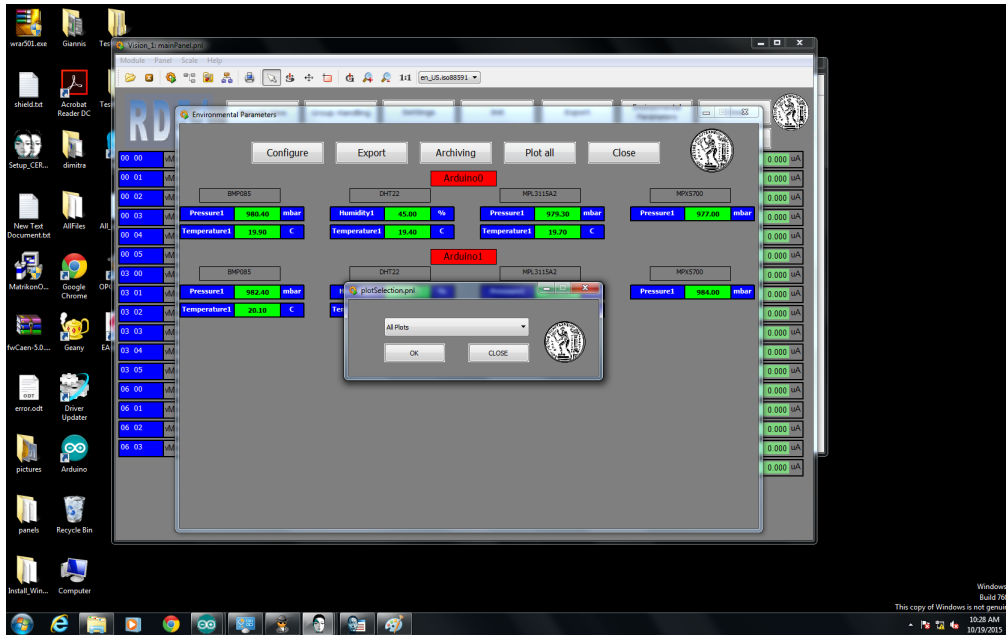


Σχήμα 2.2: Κύριο παράθυρο Slow Control System

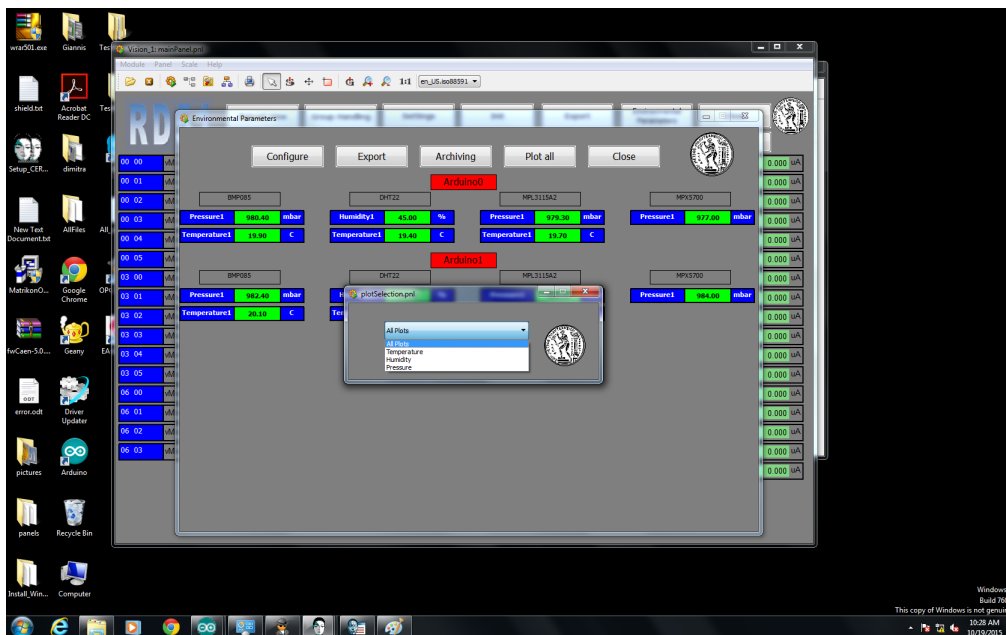


Σχήμα 2.3: Κύριο παράθυρο project μέτρησης περιβαλλοντικών συνθηκών

επιθυμεί, εικόνες [2.4], [2.5].

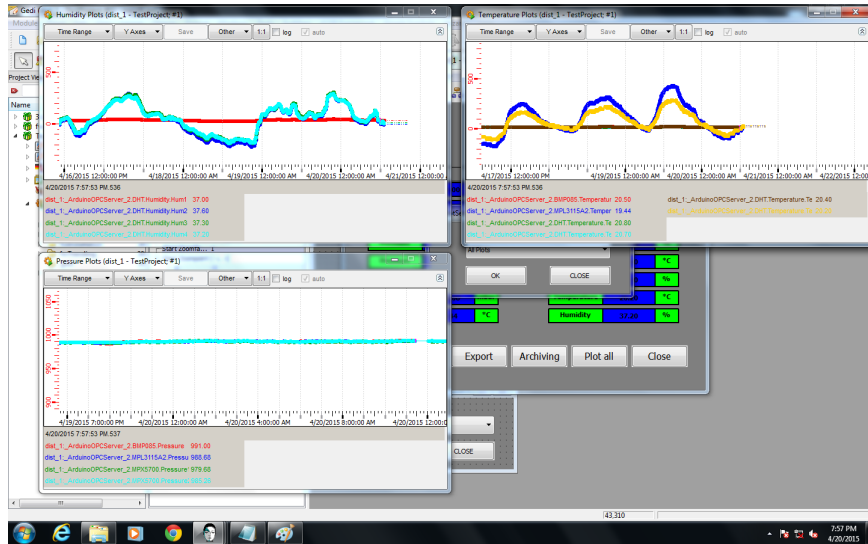


Σχήμα 2.4: Παράθυρο επιλογής διαγραμμάτων απεικόνισης περιβαλλοντικών συνθηκών



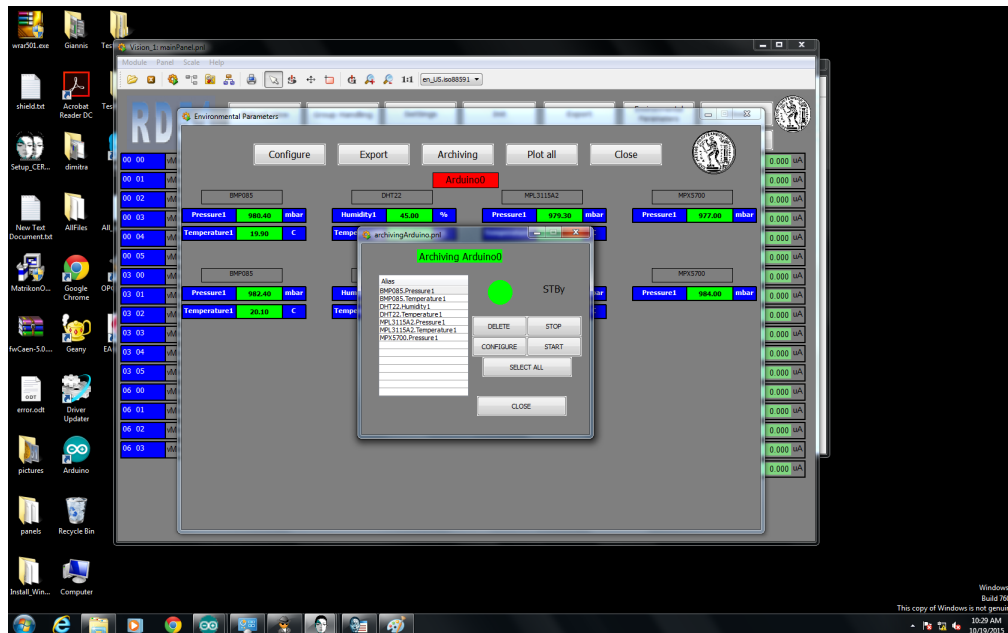
Σχήμα 2.5: Σύνολο επιλογών για τα διαγράμματα απεικόνισης περιβαλλοντικών συνθηκών

Αφού ο χρήστης επιλέξει τι θέλει και πατήσει "OK", ανοίγουν παράθυρα με διαγράμματα ανάλογα με την επιλογή του. Στην εικόνα [2.6], για παράδειγμα, έχει γίνει η επιλογή "All Plots".



Σχήμα 2.6: Διαγράμματα περιβαλλοντικών συνθηκών μετά την επιλογή "All Plots" στο παράθυρο επιλογής διαγραμμάτων

Το επόμενο είναι το κουμπί "Archiving". Το κουμπί αυτό δίνει στο χρήστη τη δυνατότητα αποθήκευσης των μετρούμενων τιμών από τους αισθητήρες οι οποίες στέλνονται στο WinCC_OA από το Arduino. Πατώντας το, ο χρήστης οδηγείται στο παράθυρο που φαίνεται στην εικόνα [2.7].

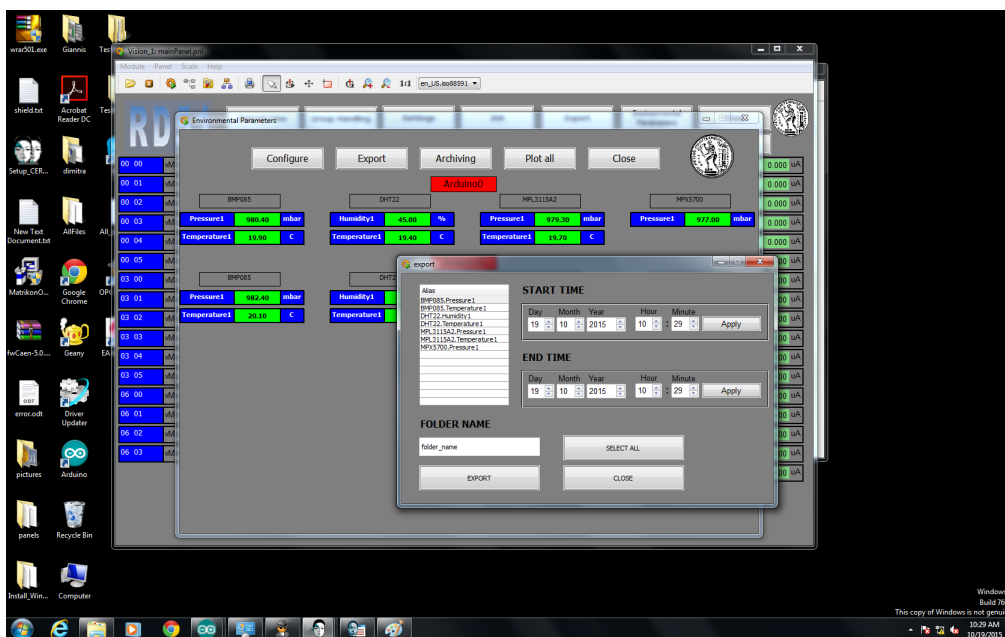


Σχήμα 2.7: Παράθυρο επιλογής και διαμόρφωσης για το Archiving των τιμών των περιβαλλοντικών συνθηκών

Στο παράθυρο αυτό, ο χρήστης μπορεί να επιλέξει μία ή περισσότερες τιμές (με το κουμπί select all να τις επιλέξει όλες) και στη

συνέχεια να δώσει εντολή στο WinCC_OA να αρχίσει να τις αποθηκεύει. Για να συμβεί αυτό, θα πρέπει να πατήσει το κουμπί "CONFIGURE" με το οποίο ουσιαστικά "ηνμερώνει" το WinCC_OA ότι θέλει να αποθηκεύει αυτές τις τιμές και στη συνέχεια το "START" για να αρχίσει το WinCC_OA να τις αποθηκεύει. Μπορεί, επίσης, για τη διακοπή της διαδικασίας να πατήσει το "STOP" για να τη σταματήσει και το "DELETE" για να αναιρέσει τη διαδικασία που έγινε με το "CONFIGURE".

Με το κουμπί "EXPORT" μπορεί να γίνει εξαγωγή των δεδομένων σε ένα φάκελο της επιλογής του χρήστη. Πατώντας το, ανοίγει το παράθυρο που φαίνεται στην εικόνα [2.8].



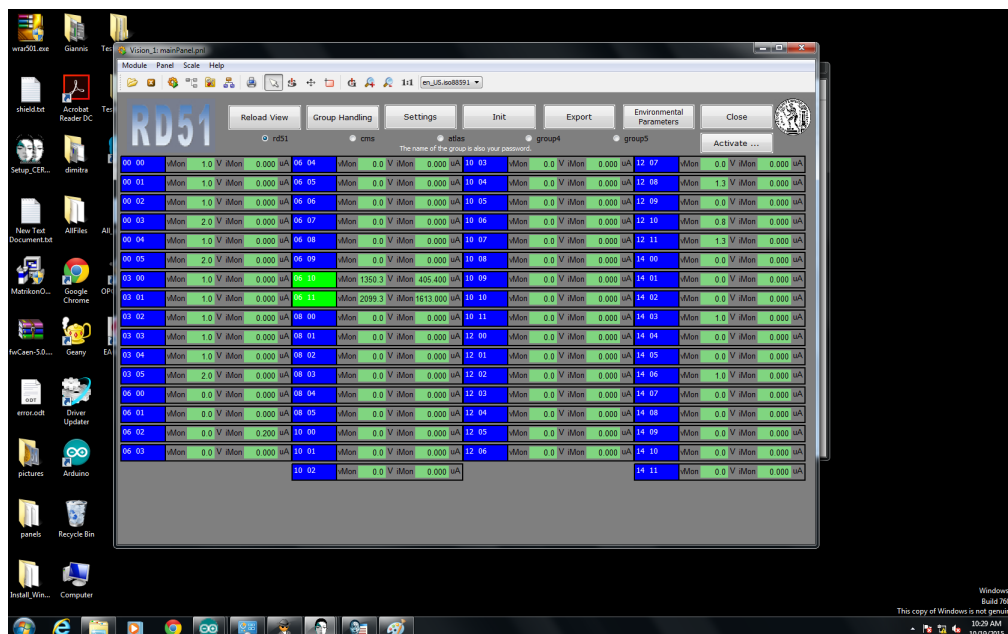
Σχήμα 2.8: Παράθυρο εξαγωγής δεδομένων περιβαλλοντικών συνθηκών

Υπάρχει και πάλι η επιλογή για το ποιές τιμές θα εξαχθούν ή με το "SELECT ALL" μπορούν να επιλέχθούν όλες. Κατόπιν, επιλέγονται οι ημερομηνίες μεταξύ των οποίων θα γίνει εξαγωγή των δεδομένων. Εδώ να σημειωθεί ότι γίνεται εξαγωγή των δεδομένων τα οποία είναι αποθηκευμένα στην database του WinCC_OA, δηλαδή, πρέπει πρώτα να έχει επιλεγθεί ποιές τιμές θα αρχειοθετηθούν (Archive) και στη συνέχεια, μέσω της επιλογής export, θα γίνει εξαγωγή των αρχειοθετημένων τιμών. Μετά την επιλογή των ημερομηνιών, στο πεδίο "FOLDER NAME" διαλέγεται το όνομα του αρχείου μέσα στο οποίο επιθυμεί ο χρήστης να αποθηκευτούν τα αρχεία με τα δεδομένα (ο φάκελος δεν πρέπει να υπάρχει ήδη). Ένας φάκελος με το όνομα FOLDER NAME θα δημιουργηθεί. Τέλος, μέσω του κουμπιού "EXPORT" τα δεδομένα εξαγονται. Τα δεδομένα αποθηκεύονται στο directory C:/FOLDER NAME και δημιουργείται ένα αρχείο για κάθε μία από τις τιμές που επιλέξαμε με το όνομα της τιμής αυτής και την ημερομηνία που έγινε η εξαγωγή. Αν έχει επιλεγθεί, για παράδειγμα, η τιμή BMP085 Pressure1 με

την επιλογή "EXPORT" στις 25/7 θα δημιουργηθεί ένα αρχείο με όνομα BMP085.Pressure1_25_7 στο φάκελο "folder name".

Σε όλα τα προηγούμενα παράθυρα, υπάρχει και ένα κουμπί "CLOSE" μέσω του οποίου υπάρχει η δυνατότητα ανά πάσα στιγμή να κλείσει το παράθυρό το οποίο είναι ανοιχτό. Η επιλογή αυτή υπάρχει και στο κύριο παράθυρο του προγράμματός μέτρησης περιβαλλοντικών συνθηκών. Τέλος, υπάρχει το κουμπί "Configure" με το οποίο ουσιαστικά διαμορφώνεται το παράθυρο και ορίζονται οι αισθητήρες οι οποίοι θα περιέχονται στο σύστημά. Το κουμπί αυτό χρησιμοποιείται μία φορά, κατά την πρώτη χρήση του συστήματός μας και κάθε φορά που αλλάζει η διάταξη που είναι συνδεδεμένη στο Arduino, με προσθήκη ή αφαίρεση κάποιου αισθητήρα κλπ. Το κουμπί αυτό και οι λειτουργίες του αναλύονται εκτενέστερα στην ενότητα 2.3, στην οποία υπάρχουν οδηγίες για το πώς μπορεί ο χρήστης να συνδέσει το Arduino με το WinCC_OA και το project μας.

Όπως προαναφέρθηκε, το project ενσωματώθηκε στο ήδη υπάρχον Slow Control System. Για το σκοπό αυτό, στο κύριο παράθυρο προστέθηκε ένα κουμπί με όνομα "Environmental Parameters", με το οποίο ο χρήστης οδηγείται στο κύριο παράθυρο του project μέτρησης περιβαλλοντικών συνθηκών και κατ'επέκταση σε όσα αναφέρθηκαν παραπάνω [2.9].



Σχήμα 2.9: Κύριο παράθυρο Slow Control System με την προσθήκη του κουμπιού Environmental Parameters

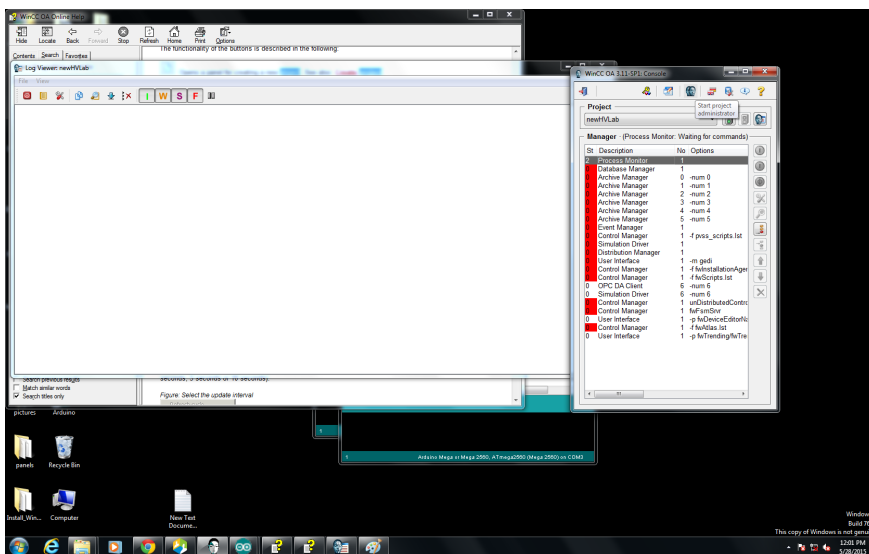
2.3 Οδηγίες Σύνδεσης του Arduino με το WinCC_OA

Η πρώτη ενέργεια για να συνδεθεί το Arduino με το WinCC_OA είναι να "φορτώσει" ο χρήστης (upload) τον κώδικα στο Arduino, στον οποίο θα δηλώνει ένα στοιχείο (item) για κάθε μεταβλητή στην οποία θέλει να έχει πρόσβαση το WinCC_OA. Ακολούθως, πρέπει να γίνει εγκατάσταση του server, όπως περιγράφεται στην παράγραφο 1.3.2.

Το επόμενο βήμα είναι να δημιουργηθεί το project στο WinCC_OA. Στο σημείο αυτό, υπάρχουν 2 επιλογές. Η πρώτη είναι να χρησιμοποιηθεί το project που δημιουργήθηκε στην εργασία αυτή, ενώ η δεύτερη είναι να δημιουργηθεί ένα καινούργιο project και απλά να χρησιμοποιηθούν τα panels από το project αυτό.

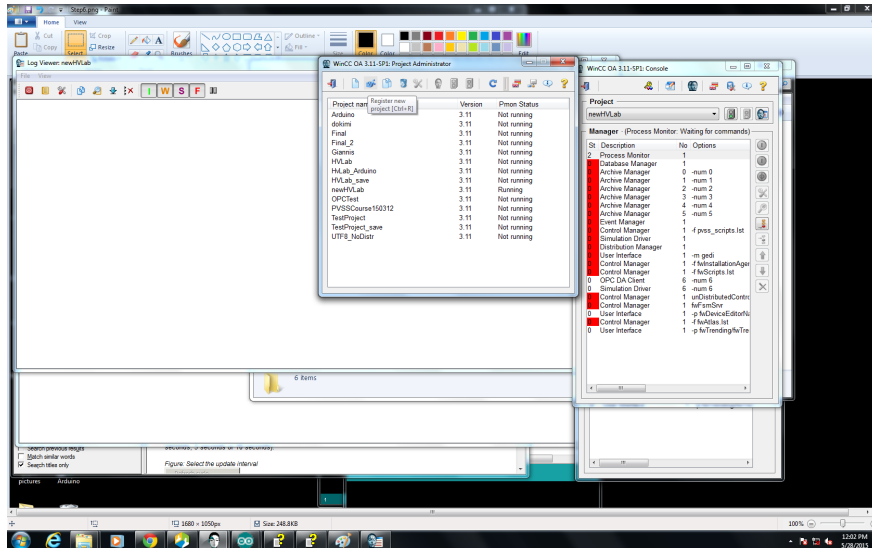
2.3.1 Εισαγωγή (import) project στο WinCC_OA

Στην ενότητα αυτή θα εξεταστεί η πρώτη περίπτωση, η εισαγωγή δηλαδή του project που δημιουργήθηκε στην παρούσα εργασία. Γίνεται εκκίνηση του WinCC_OA και στη συνέχεια του project administrator, εικόνα [2.10].

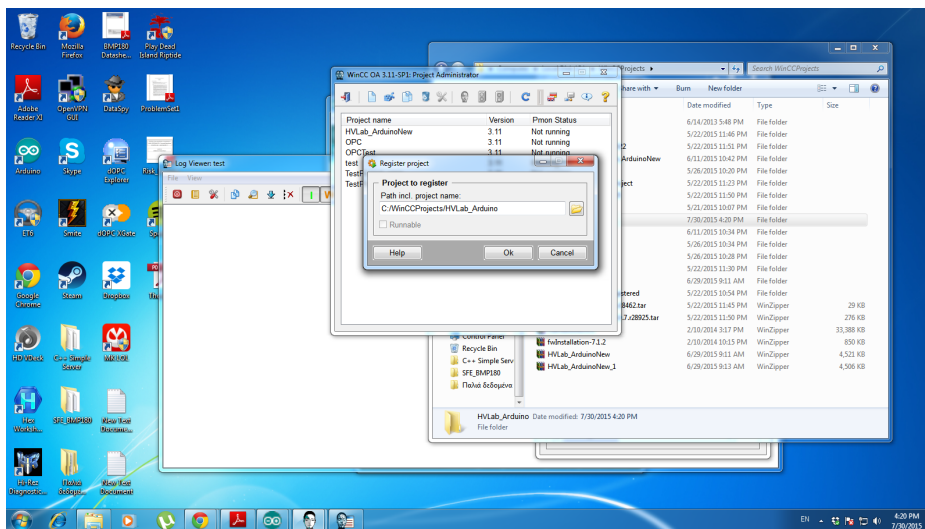


Σχήμα 2.10: Άνοιγμα του WinCC_OA project administrator

Επιλέγεται "Register New Project" και εισάγεται το μέρος του υπολογιστή στο οποίο βρίσκεται το επιθυμητό project, στην περίπτωση μας HVLab_Arduino όπως φαίνεται στις εικόνες [2.11], [2.12]. Το project έχει εισαχθεί με επιτυχία στο WinCC_OA.



Σχήμα 2.11: Επιλογή Register New Project στον project administrator του WinCC_OA

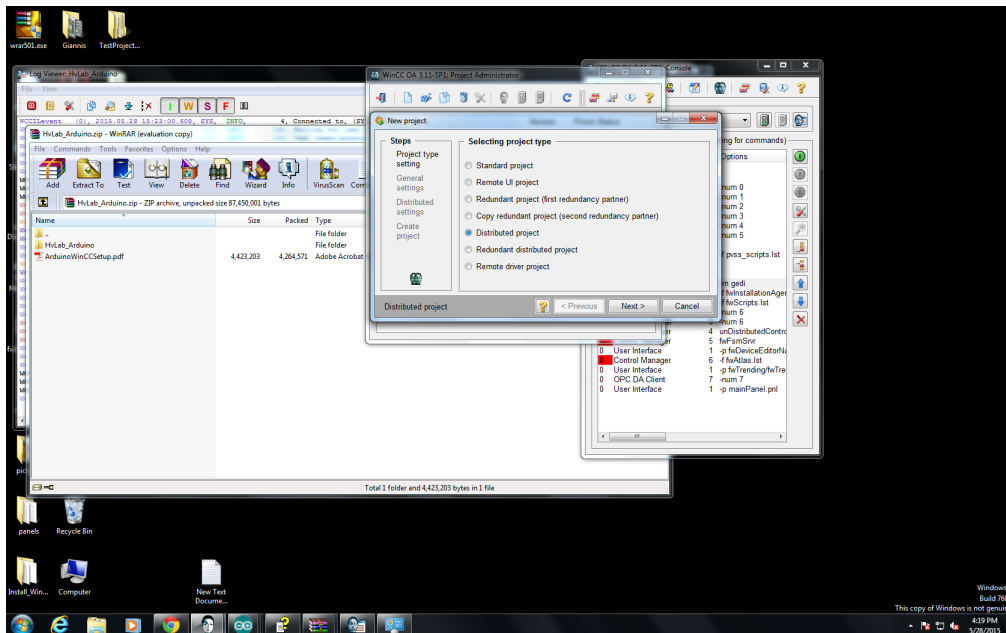


Σχήμα 2.12: Εισαγωγή της περιοχής του υπολογιστή στην οποία είναι αποθηκευμένο το project που θα εισαχθεί στο WinCC_OA

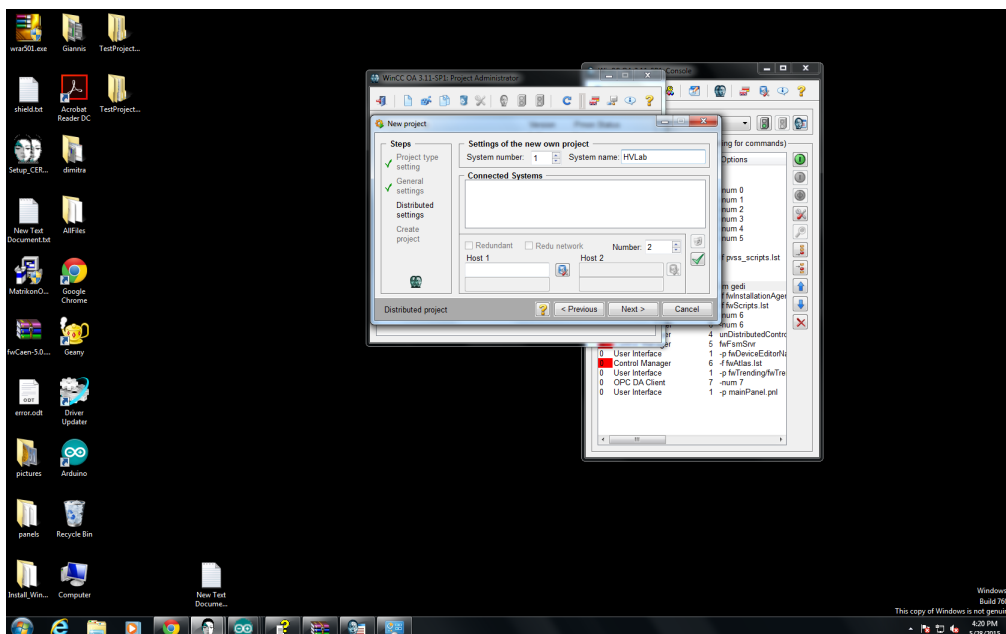
2.3.2 Δημιουργία καινούργιου project στο WinCC_OA

Στην ενότητα αυτή θα περιγραφεί η διαδικασία δημιουργίας καινούργιου project για την σύνδεση του Arduino με το WinCC_OA . Γίνεται εκκίνηση του WinCC_OA καθώς και του project administrator αλλά αυτή τη φορά επιλέγεται "New Project" και στη συνέχεια "distributed project", όπως φαίνεται στην εικόνα [2.13]. Στη συνέχεια εισάγεται το επιθυμητό όνομα για το project, το μέρος στον υπολογιστή όπου θα αποθηκευτεί

και τελικά γίνεται εισαγωγή ενός System Name για το project, όπως για παράδειγμα HVLab, εικόνα [2.14].



Σχήμα 2.13: Επιλογή Distributed Project κατά τη διαδικασία δημιουργίας καινούργιου project στο WinCC_OA



Σχήμα 2.14: Εισαγωγή ονόματος καθώς και System Name κατά τη δημιουργία καινούργιου project στο WinCC_OA

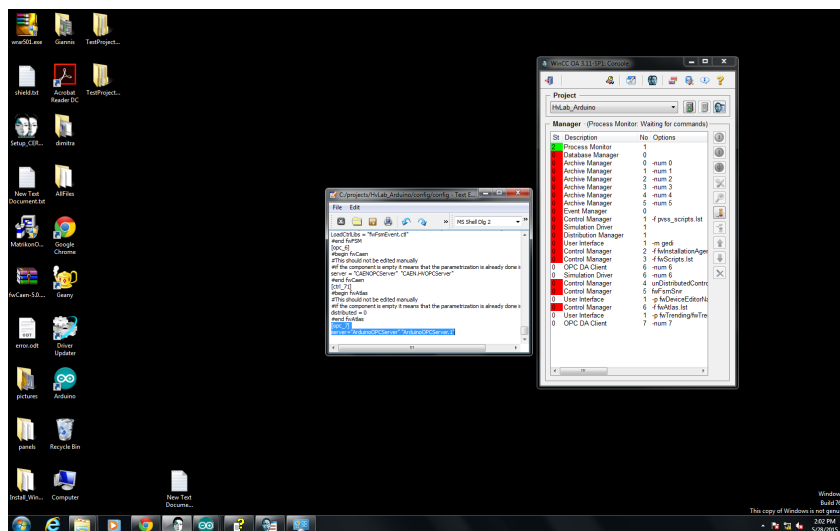
Το project έχει δημιουργηθεί και τώρα πρέπει να συνδεθεί με τον OPC Server. Επιλέγεται το project που μόλις δημιουργήθηκε και πατάμε

”edit config file”. Αφού διαλεχθεί το ”config file”, ανοίγει με διπλό κλικ και στο τέλος προστίθενται:

[opc_7]

server = ”ArduinoOPCServer””ArduinoOPCServer.1”

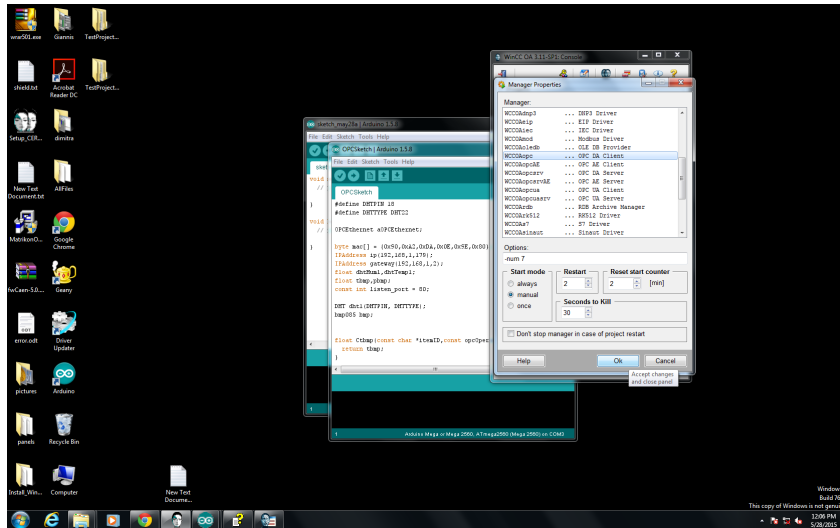
στη συνέχεια πατάμε ”save, exit”, όπως φαίνεται στην εικόνα [2.15].



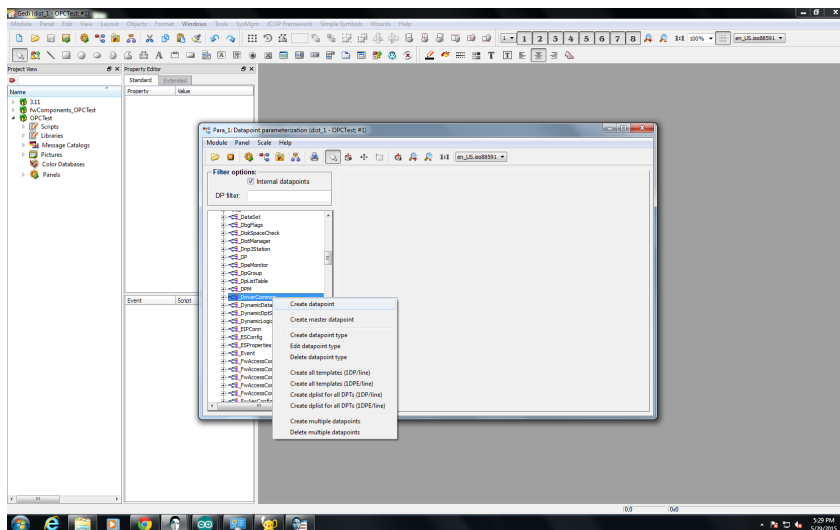
Σχήμα 2.15: Αλλαγές στο config file του καινούργιου project με σκοπό τη σύνδεσή του project με τον OPC Server

Τέλος, μέσω του κουμπιού ”append new manager” ανοίγει ένα παράθυρο στο οποίο επιλέγεται ”OPC DA Client” εικόνα [2.16]. Στο επόμενο παράθυρο στο πεδίο Options εισάγεται -num 7 και στο Start mode επιλέγεται manual και ”OK”.

Γίνεται εκκίνηση του project και του manager που μόλις δημιουργήθηκε. Ανοίγεται ο GEDI και ο ”para”. Στη συνέχεια επιλέγεται ”Internal datapoints” και στο DPT ”_DriverCommon ” ο χρήστης κάνει δεξί κλικ και επιλέγει ”Create datapoint” [2.17]. Στο παράθυρο που μόλις άνοιξε εισάγεται το όνομα ”_Driver7 ” και επιλέγεται ”OK”. Η ίδια διαδικασία επαναλαμβάνεται και για το DPT ”_Statistics_Driver_Configs”, όπου αυτή τη φορά εισάγεται το όνομα ”_Stat_Configs_driver_7 ”.



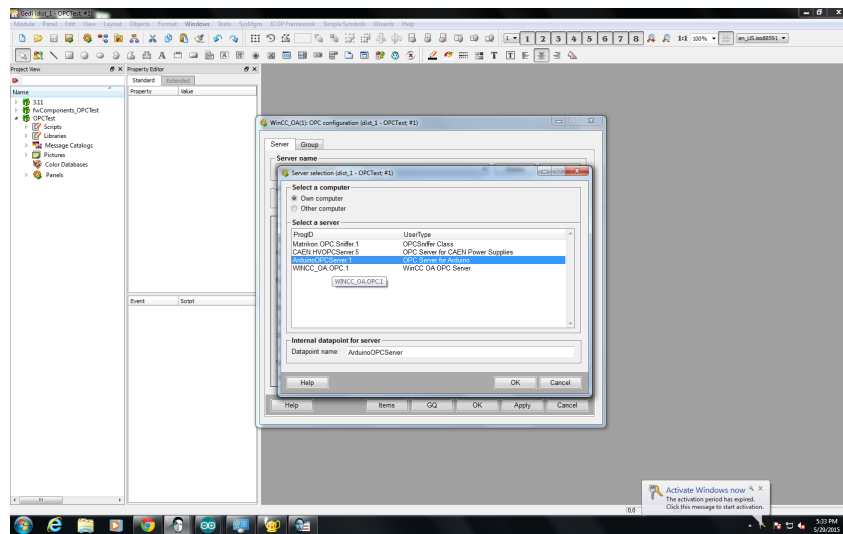
Σχήμα 2.16: Δημιουργία καινούργιου manager μέσω της επιλογής "append new manager" για τον Arduino OPC Server



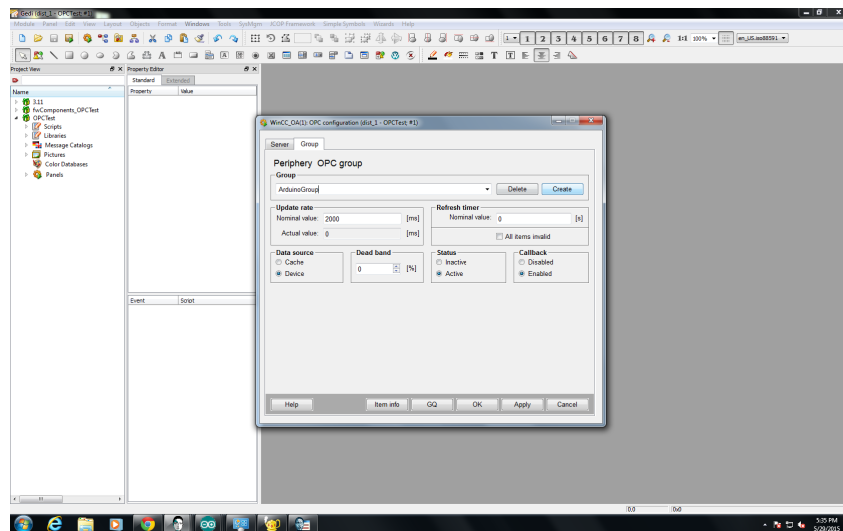
Σχήμα 2.17: Διαδικασία δημιουργίας του DataPoint "_DriverCommon" για τη σύνδεση του Arduino OPC Server με το WinCC_OA

Στο επάνω μέρος του GEDI ο επιλέγεται "SysMgm" και στη συνέχεια OPC Driver → OPC Driver. Ανοίγει ένα παράθυρο με 2 καρτέλες. Στην καρτέλα Server, πατάμε το κουμπί "create" και στο παράθυρο που ανοίγει επιλέγεται ο server ArduinoOPCServer.1, σαν όνομα αφήνεται το ArduinoOPCServer και πατάμε "APPLY" "OK" και όπως φαίνεται στην εικόνα [2.18]. Στη συνέχεια, στην καρτέλα GROUP εισάγεται το επιθυμητό όνομα για το group, για παράδειγμα ArduinoGroup και, στα 3 πεδία, επιλέγουμε Data Source → Device, Status → Active, Callback → Enabled και πατάμε "APPLY" και "OK", εικόνα [2.19]. Τέλος, πρέπει να μεταφερθεί ο φάκελος με τα panels από το έτοιμο project στο καινούργιο

που δημιουργήθηκε.



Σχήμα 2.18: Δήλωση του ArduinoOPCServer στο project στο WinCC_OA

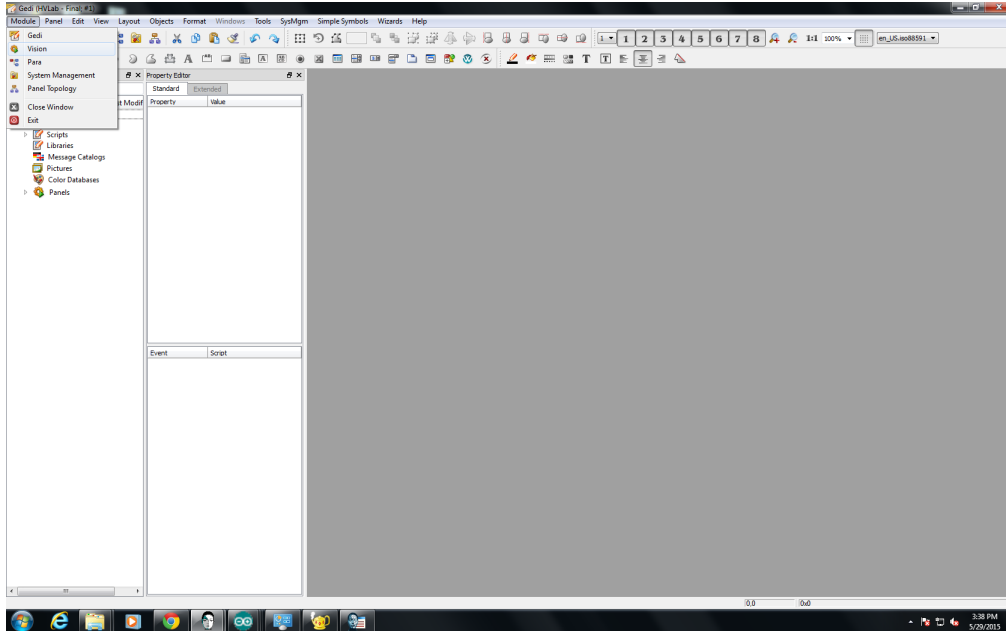


Σχήμα 2.19: Δημιουργία του Arduino Group για τις τιμές μέτρησης των περιβαλλοντικών συνθηκών

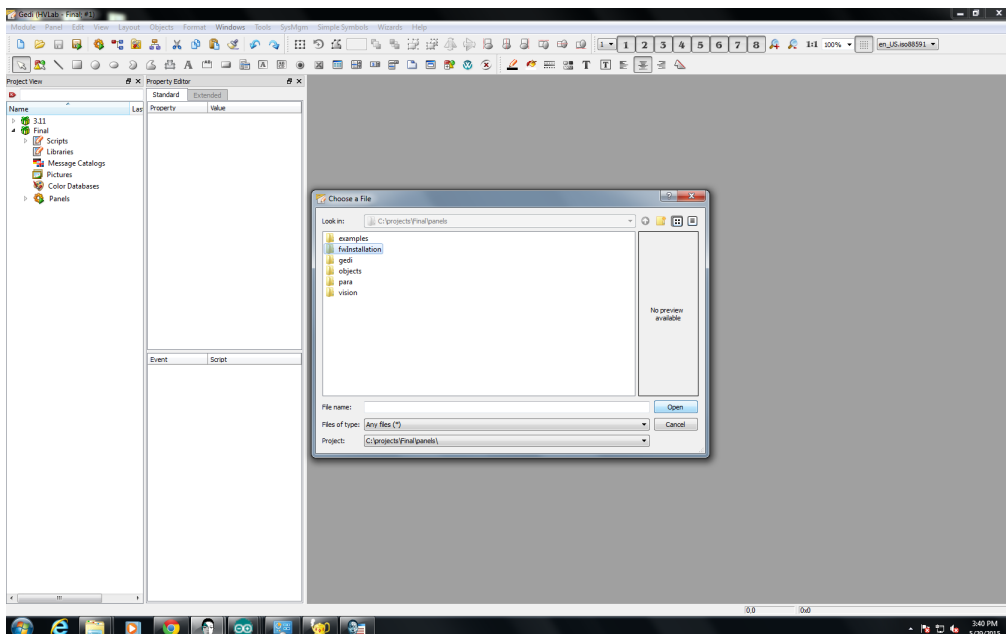
2.3.3 Εγκατάσταση JCOP Framework Components

Ο χρήστης επιλέγει το αρχείο **fwInstaller.zip** και το εξάγει (extract) στο φάκελο του project που δημιουργήθηκε στην προηγούμενη ενότητα. Στη συνέχεια, πρέπει να κάνει εξαγωγή το αρχείο **framework.zip** σε οποιοδήποτε μέρος του υπολογιστή, χωρίς να είναι υποχρεωτικό να είναι στον ίδιο φάκελο με το project. Κατόπιν, πρέπει να γίνει εκκίνηση του project στο WinCC_OA και να ανοιχθεί το **fwInstallation** panel, όπως

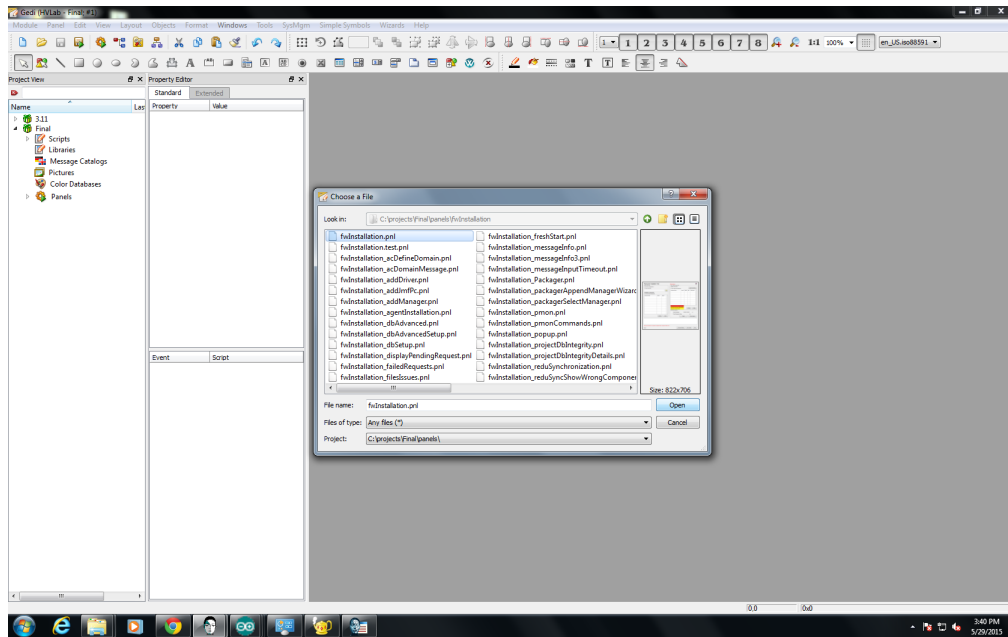
δείχνουν οι παρακάτω εικόνες [2.20], [2.21], [2.22]. Για να γίνει αυτό, στο πάνω αριστερά μέρος του GEDI ο χρήστης πρέπει να πατήσει Module → Vision. Στη συνέχεια στο παράθυρο που ανοίγει επιλέγεται ο φάκελος fwInstallation και ανοίγεται το fwInstallation.pnl.



Σχήμα 2.20: Άνοιγμα fwInstallation Panel με την επιλογή Module → Vision

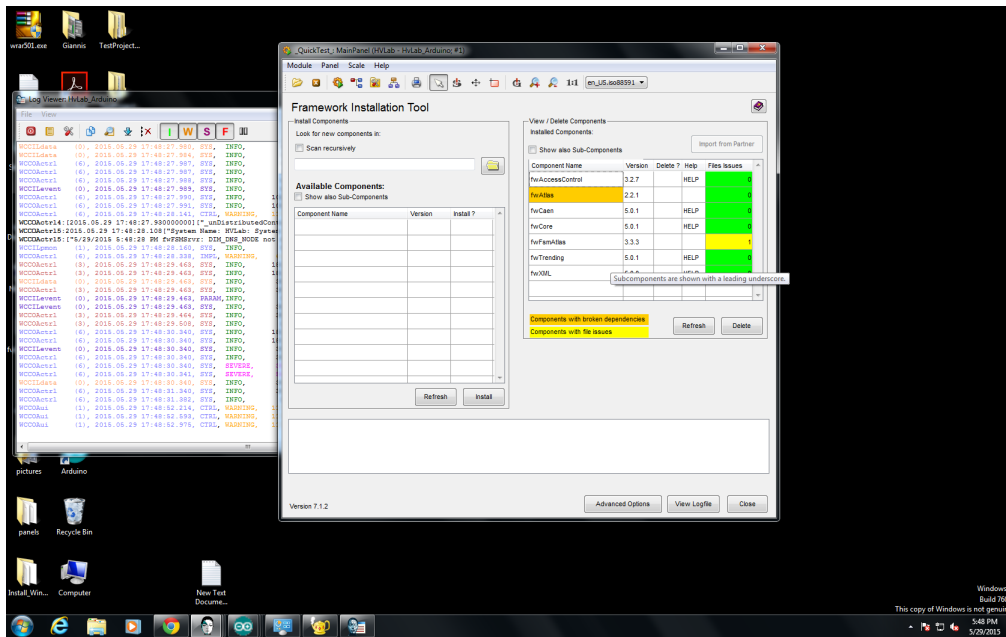


Σχήμα 2.21: Επιλογή του φακέλου fwInstallation κατά τη διαδικασία ανοίγματος του fwInstallation Panel



Σχίμα 2.22: Επιλογή και άνοιγμα του fwInstallation Panel

Όταν ανοίξει, το **fwInstallation panel** ζητάει από το χρήστη να επιλέξει ένα φάκελο, όπου θα αποθηκευτούν τα components. Αφού επιλέχθει (εάν αυτός δεν υπάρχει θα δημιουργηθεί) στη συνέχεια πρέπει να δηλωθεί ο φάκελος στον οποίο είχαν εξαχθεί τα περιεχόμενα του **framework.zip** αρχείου. Ο πίνακας Available Components γεμίζει με όλα τα components που μπορούν να εγκατασταθούν. Γίνεται επιλογή των επιθυμητών Components κάνοντας αριστερό κλικ στη στήλη Install δίπλα από το όνομα του component που θα εγκατασταθεί. Για το συγκεκριμένο project, θα χρειαστούν τα **fwAccessControl**, **fwCaen**, **fwCore**, **fwTrending**, **fwXML**. Αφού επιλεγθούν όλα, πατάμε το κουμπί "Install" και "OK". Στα 2 επόμενα παράθυρα, επιλέγεται "Activate Driver" και "Restart Project" αντίστοιχα. Για το συγκεκριμένο project, θα χρειαστούν επίσης και τα components **fwAtlas**, **fwFsmAtlas**. Τα components αυτά εγκαθιστώνται μέσω του **fwInstallation panel**, αλλάζοντας απλά το φάκελο όπου έχει εξαχθεί το αρχείο **framework.zip** με αυτό στον οποίο έχουμε εξαγάγει τα **fwAtlas.zip** και **fwFsmAtlas.zip** αντίστοιχα. Θα πρέπει, δηλαδή, συνολικά να είναι εγκατεστημένα τα components: **fwAccessControl**, **fwAtlas**, **fwCaen**, **fwCore**, **fwFsmAtlas**, **fwTrending**, **fwXML** εικόνα [2.23].



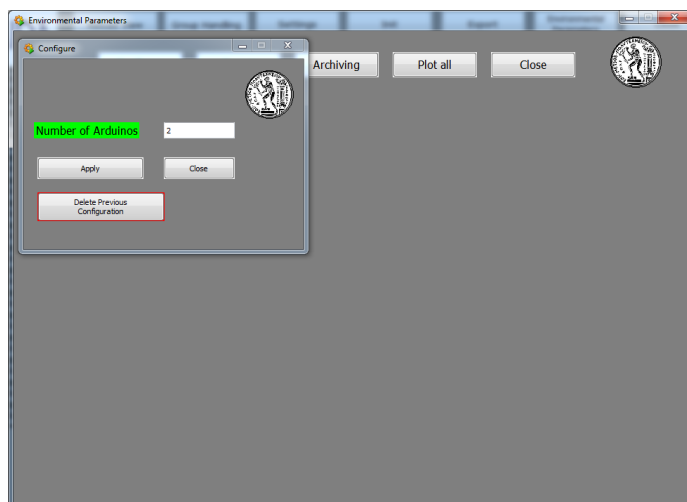
Σχήμα 2.23: Παράθυρο fwInstallation για την εγκατάσταση των components του WinCC_OA. Στο δεξί μέρος του panel φαίνονται τα components τα οποία είναι εγκατεστημένα στο project αυτό, τα components αυτά είναι τα **fwAccessControl**, **fwAtlas**, **fwCaen**, **fwCore**, **fwFsmAtlas**, **fwTrending**, **fwXML** που είναι απαραίτητα για τη λειτουργία του project μέτρησης περιβαλλοντικών συνθηκών

Τέλος, μέσω του κουμπιού "append new manager", θα δημιουργηθεί ένας manager για να ανοίγει το main panel του project ώστε να μη χρειάζεται αυτό να γίνεται κάθε φορά μέσω του gedi. Αφού πατηθεί το κουμπί, επιλέγεται "User Interface" και στο πεδίο Options εισάγεται -p mainPanel.rpl και στο start mode → once. Έχοντας ολοκληρώσει αυτές τις επιλογές το project μας είναι έτοιμο. Το μόνο που απομένει τώρα είναι να ορισθούν οι αισθητήρες που θα χρησιμοποιηθούν και να αρχίσει η διαδικασία των μετρήσεων.

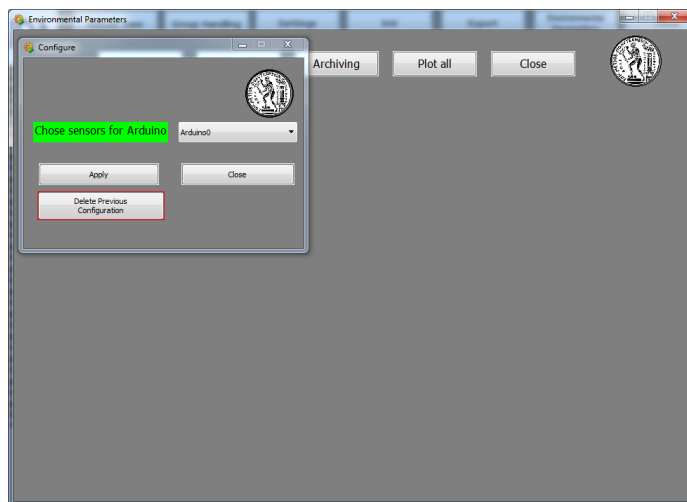
2.3.4 Σύνδεση του project με το Arduino

Βασική προϋπόθεση για τη σύνδεση του project με το Arduino, αποτελεί το να "ανέβει" στην πλακέτα ο κώδικας και να δημιουργηθεί η σωστή συνδεσμολογία μεταξύ του Arduino και των αισθητήρων. Έπειτα γίνεται εκκίνηση του WinCC_OA, του project, και τέλος του OPC Client. Μόλις το project ξεκινήσει, θα ανοίξει το κύριο παράθυρο του, το οποίο περιέχει δεδομένα σχετικά με τις τάσεις που μετρούν οι ανιχνευτές. Στη συνέχεια, μέσω του κουμπιού "Environmental Parameters" ανοίγει το παράθυρο μέτρησης περιβαλλοντικών συνθηκών, το οποίο είναι άδειο καθώς δεν έχει ακόμα ορισθεί κάποιος αισθητήρας από τον οποίο να δέχεται δεδομένα το WinCC_OA. Για να γίνει αυτό, πρέπει να πατηθεί το κουμπί "Configure" και στο παράθυρο που ανοίγει να εισα-

χθεί ο αριθμός των Arduino από τα οποία επιθυμεί ο χρήστης να λάβει δεδομένα. Εισάγεται ο επιθυμητός αριθμός, για παράδειγμα 2 όπως φαίνεται στην εικόνα [2.24]. Με την εισαγωγή του αριθμού των Arduino ανοίγει το επόμενο παράθυρο, στο οποίο πρέπει να επιλέχθει ποιά από τα συνδεδεμένα πλέον στο σύστημά Arduino επιθυμεί ο χρήστης να ρυθμίσει, εικόνα [2.25]. Το πρώτο Arduino είναι το Arduino0, ενώ το δεύτερο είναι το Arduino1 κ.λ.π. ανάλογα με το πλήθος των Arduino που ορίστηκε στο πρώτο παράθυρο.



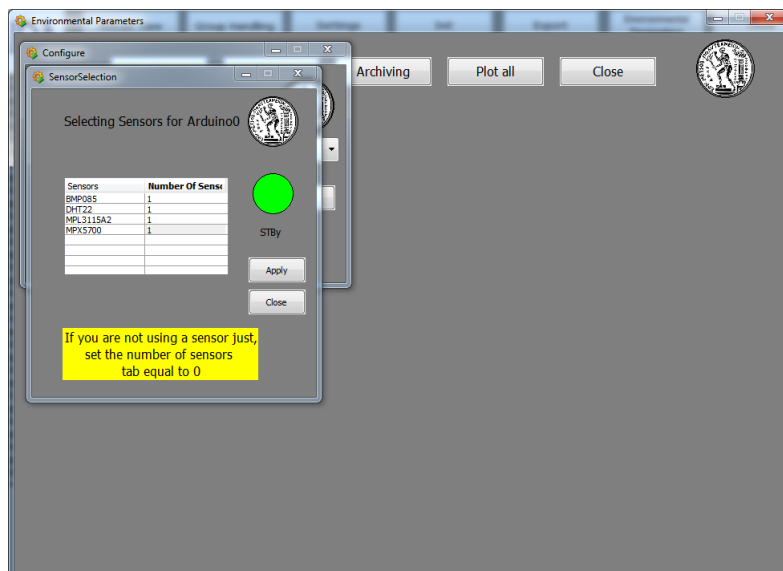
Σχήμα 2.24: Εισαγωγή του πλήθους των Arduino που θέλουμε να συνδέσουμε στο σύστημα μας



Σχήμα 2.25: Επιλογή ποιού από τα Arduino που είναι συνδεδεμένα στο σύστημα μας θέλουμε να ρυθμίσουμε

Αφού πατηθεί "Apply" ανοίγει το επόμενο παράθυρο, στο οποίο πρέπει να δηλωθεί το είδος και την ποσότητα των αισθητήρων που είναι συνδεδεμένοι στο Arduino. Η επιλογή γίνεται μεταξύ των αισθητήρων

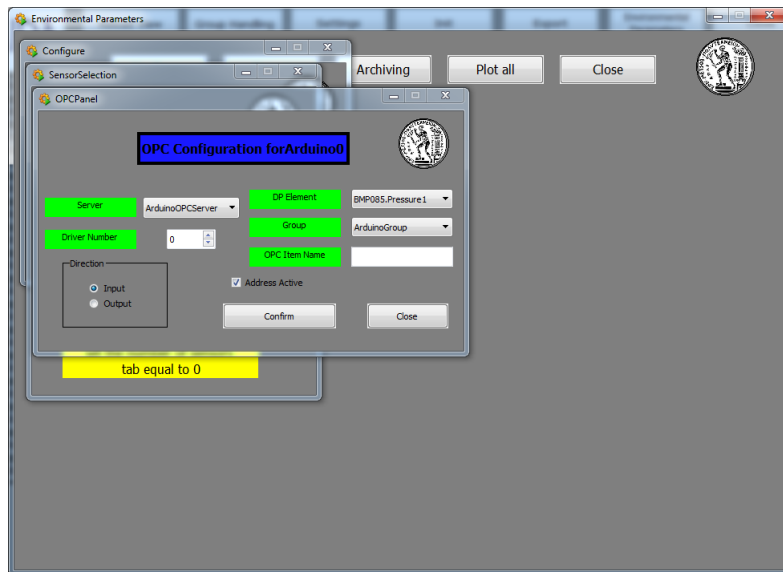
BMP085, DHT22, MPL3115A2, MPX5700, οι οποίοι αναλύθηκαν στο προηγούμενο κεφάλαιο. Εδώ να σημειωθεί ότι, αν δεν χρησιμοποιείται κάποιος από αυτούς, πρέπει να οριστεί η ποσότητά του ίση με 0, εικόνα [2.26]. Μόλις συμπληρωθούν τα επιθυμητά πεδία επιλέγεται "OK" και στο παράθυρο το οποίο είναι ανοιχτό αλλά και στο μικρό παράθυρο που θα ανοίξει.



Σχήμα 2.26: Δήλωση είδους και ποσότητας αισθητήρων που είναι συνδεδεμένοι στο κάθε Arduino

Παρατηρούμε την εμφάνιση μίας νέας επιλογής στο παράθυρο, της "Configure OPC", μέσω της οποίας θα πραγματοποιηθεί η σύνδεση με το Server. Με την επιλογή αυτή καταλήγουμε στο τελευταίο παράθυρο, εικόνα [2.27].

Στο παράθυρο αυτό, θα γίνει σύνδεση κάθε **DataPoint Element** του **WinCC_OA** το οποίο αντιστοιχεί σε μια τιμή (πίεση, θερμοκρασία κλπ) την οποία επιθυμεί ο χρήστης να μετρήσει με ένα item του OPC server. Στα πεδία που υπάρχουν στο παράθυρο γίνεται εισαγωγή των παρακάτω στοιχείων: **Server**, → "ArduinoOPCServer" **DriverNumber** → "7", **Direction** → "Input". Στο πεδίο **group** επιλέγεται είτε "ArduinoGroup" (αν ακολουθήθηκε η διαδικασία της ενότητας 1.3.1) είτε αυτό που δημιουργήθηκε για τις μεταβλητές των περιβαλλοντικών συνθηκών με βάση τις οδηγίες της ενότητας 1.3.2. Τέλος, στο πεδίο **DPElement** επιλέγεται το επιθυμητό DPE και στο πεδίο **OPCItemName** εισάγεται το όνομα του αντίστοιχου "item". Το Arduino είναι προγραμματισμένο να δημιουργεί items με τα παρακάτω ονόματα πίνακας 2.1. Εάν στο ίδιο Arduino, είναι συνδεδεμένοι περισσότεροι από ένας αισθητήρες για κάποιο είδος, το item το οποίο αντιστοιχεί στο δεύτερο αισθητήρα θα είναι ίδιο με το πρώτο αλλά ο αριθμός στο τέλος του θα είναι 2 (αντί για 1), 3 για τον τρίτο αισθητήρα, κ.λ.π. Για παράδειγμα αν είχαμε 2 αισθητήρες MPX5700 στον πρώτο θα αντιστοιχούσε το όνομα pmpx1 και στο δε-



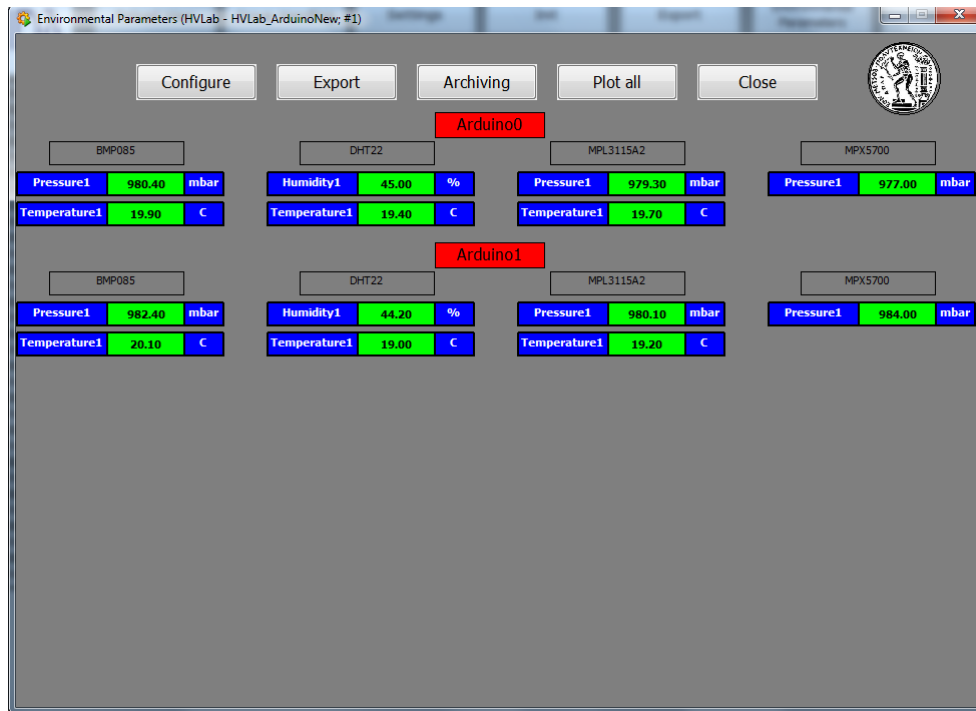
Σχήμα 2.27: Παράθυρο σύνδεσης με τα Items του ArduinoOPCServer

DPElements	Item Names
BMP085.Pressure1	pbmp1
BMP085.Temperature1	tbmp1
DHT22.Humidity1	d1h
DHT22.Temperature1	d1t
MPL3115A2.Presusre1	pmpl1
MPL3115A2.Temperature1	tmpl1
MPX5700.Pressure1	pmplx1

Πίνακας 2.1: Αντιστοίχιση των Data Point Elements του WinCC_OA με τα items του server

ύτερο pmplx2. Η διαδικασία αυτή, γίνεται για κάθε τιμή που θέλουμε να μετρήσουμε. Εισάγονται δηλαδή, τα στοιχεία για κάθε τιμή και πατάμε "Confirm" η διαδικασία επαναλαμβάνεται και για τους υπόλοιπους αισθητήρες. Αφού τελειώσουμε, πατάμε "Close" στα 2 παράθυρα και επιλέγουμε αισθητήρες για τα υπόλοιπα Arduino, αν είναι συνδεδεμένα περισσότερα του ενός. Τέλος, κλείνουμε όλα τα παράθυρα και ξαναπατάμε το κουμπί "Environmental Parameters" και στο παράθυρό μας αναγράφονται πλέον όλες οι τιμές που επιλέχθηκαν, όπως φαίνεται στην παρακάτω εικόνα [2.28].

Στη συνέχεια υπάρχει η δυνατότητα να χρησιμοποιηθεί οποιαδήποτε από τις υπόλοιπες λειτουργίες του παραθύρου, που έχουν περιγραφεί στην ενότητα 2.2.

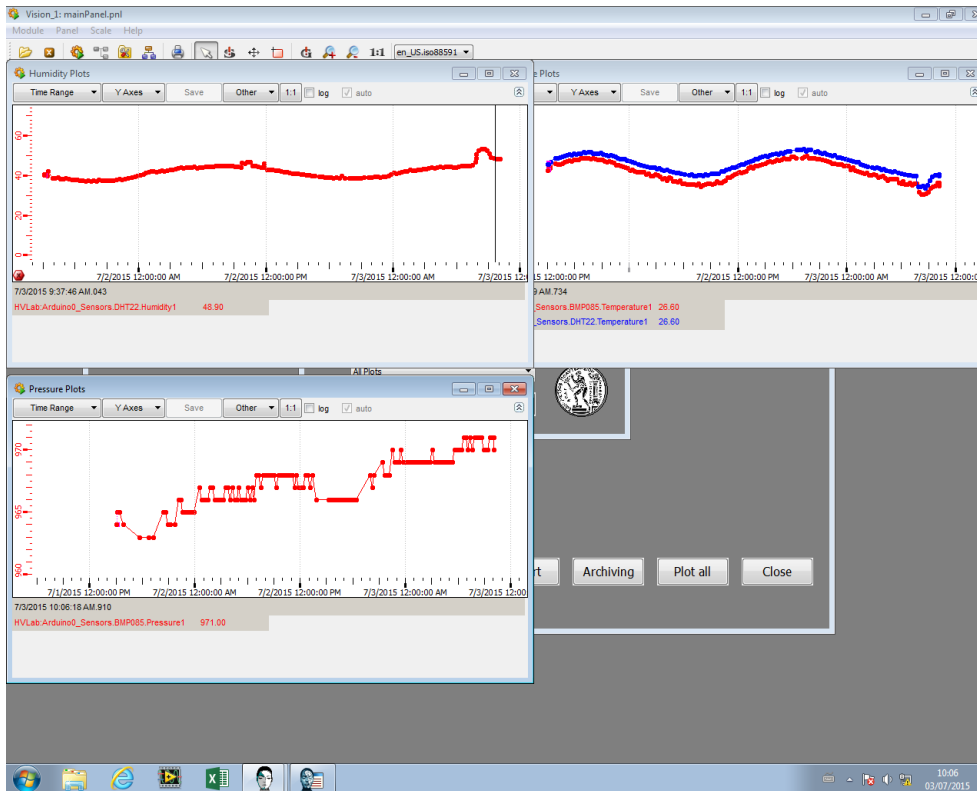


Σχήμα 2.28: Κύριο παράθυρο περιβαλλοντικών συνθηκών στο οποίο εμφανίζονται οι τιμές για τους αισθητήρες που δηλώθηκαν με τη διαδικασία που περιγράφεται στην ενότητα αυτή

Κεφάλαιο 3

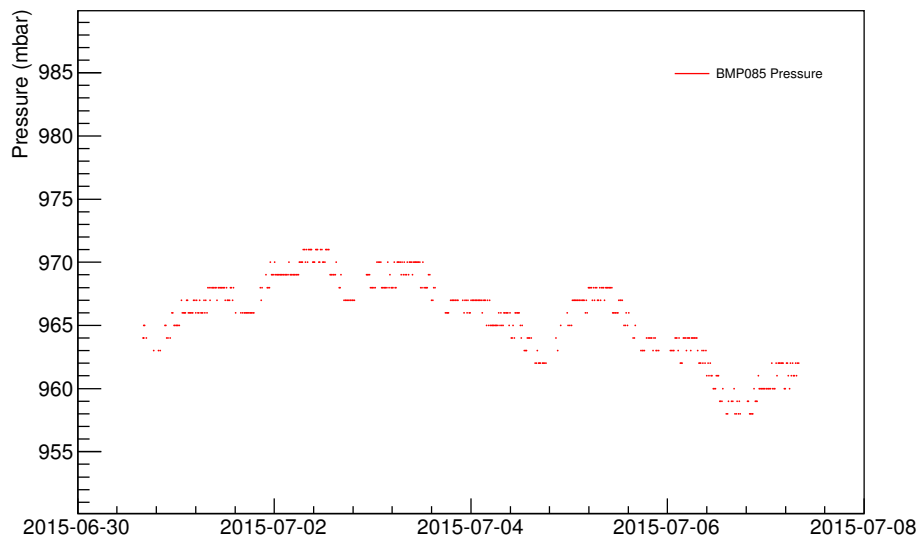
Συμπεράσματα

Κατά τη διάρκεια της εργασίας αυτής, δημιουργήθηκε ένα σύστημα μέτρησης με δυνατότητα καταγραφής και αποθήκευσης από απόσταση, με σκοπό την περαιτέρω επεξεργασία των περιβαλλοντικών συνθηκών πίεσης, θερμοκρασίας και υγρασίας. Το σύστημα αυτό βασίζεται στην μικροελεγκτική πλακέτα **Arduino** και στο σύστημα αυτομάτου ελέγχου **WinCC_OA**, το οποίο είναι υπεύθυνο για την καταγραφή και αποθήκευση των μετρήσεων. Χρησιμοποιήθηκαν 4 διαφορετικοί αισθητήρες **DHT22** (θερμοκρασία και υγρασία), **BMP085** (πίεση και θερμοκρασία) **MPL3115A2** (πίεση και θερμοκρασία), **MPX5700AP** (πίεση). Η συμπεριφορά όλων των αισθητήρων κρίνεται ικανοποιητική, στοιχεία για την μέτρηση της επίδοσης και της σταθερότητας τους αναλύονται στο κεφάλαιο 1, στην ενότητα 1.2. Το σύστημά μας, χρησιμοποιήθηκε με επιτυχία στο Test Beam που διεξήχθη από 31/6/2015 έως 15/7/2015 στις εγκαταστάσεις του **RD51 collaboration** στο **CERN**. Πιο συγκεκριμένα, χρησιμοποιήθηκε διάταξη που περιλάμβανε δύο αισθητήρες, έναν BMP085 και έναν DHT22. Για τις απαιτήσεις του συγκεκριμένου πειράματος, αποφασίστηκε να χρησιμοποιηθούν δύο **RJ45 connectors** ο ένας προσαρμοσμένος στο Arduino και ο άλλος σε μια πλακέτα πάνω στην οποία ήταν ενσωματωμένοι οι δύο αισθητήρες. Στην πρώτη που ακολουθεί [3.1], φαίνεται το project που αναπτύχθηκε κατά τη διάρκεια αυτής της εργασίας σε λειτουργία. Πιο συγκεκριμένα φαίνονται διαγράμματα πίεσης υγρασίας και θερμοκρασίας σε πραγματικό χρόνο δηλαδή κατά τη διάρκεια λήψης των μετρήσεων. Τα διαγράμματα αυτά άνοιξαν μέσω τις επιλογής "All Plots" στο παράθυρο επιλογής διαγραμμάτων του project μέτρησης περιβαλλοντικών συνθηκών. Στις επόμενες εικόνες [3.2], [3.3], [3.4] φαίνονται διαγράμματα πίεσης, θερμοκρασίας και υγρασίας αντίστοιχα με μετρήσεις που λήφθηκαν κατά τη διάρκεια του Test Beam.



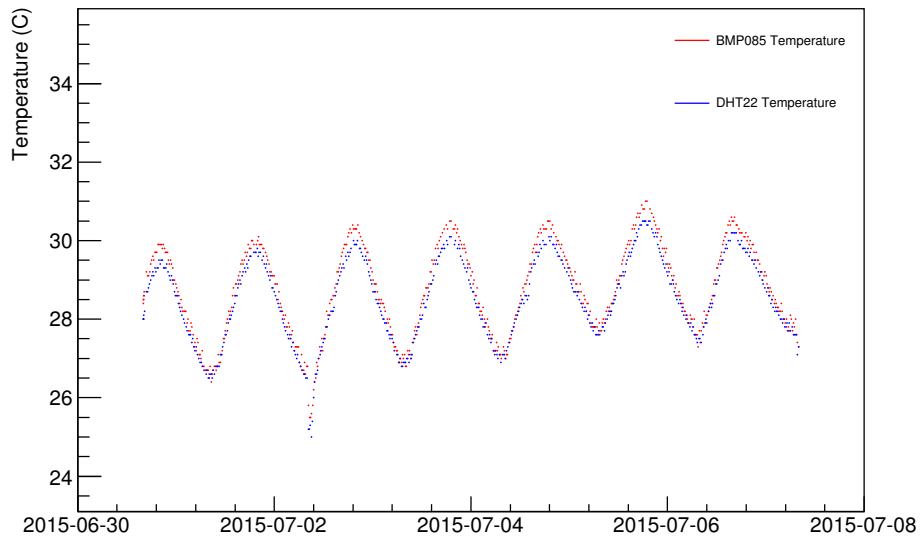
Σχήμα 3.1: Επιλογή "All Plots" στο παράθυρο επιλογής διαγραμμάτων του project μέτρησης περιβαλλοντικών συνθηκών κατά τη διάρκεια του Test Beam στις εγκαταστάσεις του RD51 collaboration στο CERN

Pressure vs Time



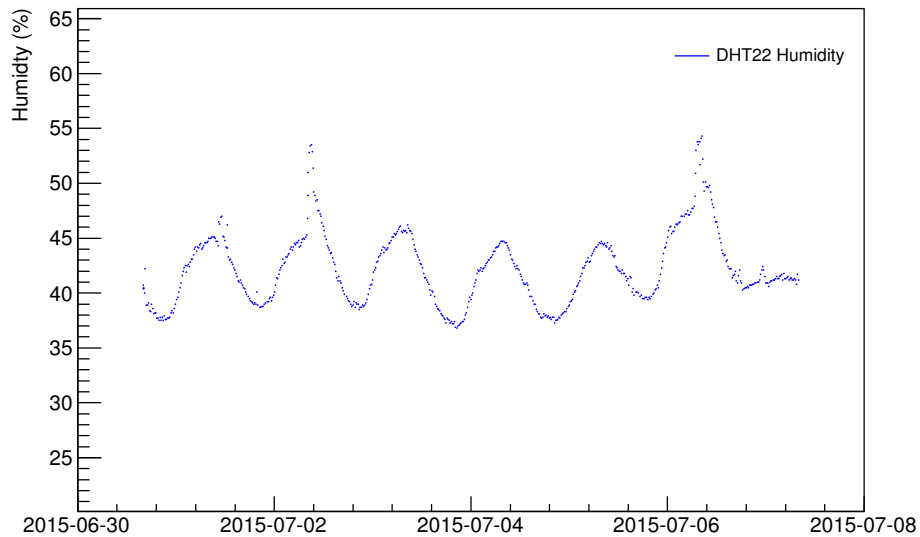
Σχήμα 3.2: Διάγραμμα πίεσης συναρτήσει του χρόνου για τον αισθητήρα BMP085 με δεδομένα που λήφθηκαν κατά τη διάρκεια του Test Beam στις εγκαταστάσεις του RD51 collaboration στο CERN

Temperature vs Time



Σχήμα 3.3: Διάγραμμα θερμοκρασίας συναρτήσει του χρόνου για τους αισθητήρες BMP085, DHT22 με δεδομένα που λήφθηκαν κατά τη διάρκεια του Test Beam στις εγκαταστάσεις του RD51 collaboration στο CERN

Humidity vs Time



Σχήμα 3.4: Διάγραμμα υγρασίας συναρτήσει του χρόνου για τον αισθητήρα DHT22 με δεδομένα που λήφθηκαν κατά τη διάρκεια του Test Beam στις εγκαταστάσεις του RD51 collaboration στο CERN

Βιβλιογραφία

- [1] *Arduino - ArduinoEthernetShield*. URL: <https://www.arduino.cc/en/Main/ArduinoEthernetShield>.
- [2] *Arduino - ArduinoMega2560*. URL: <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>.
- [3] *Arduino - Environment*. URL: <https://www.arduino.cc/en/Guide/Environment>.
- [4] *Arduino - Introduction*. URL: <https://www.arduino.cc/en/Guide/Introduction>.
- [5] *Arduino — Wikipedia, The Free Encyclopedia*. 2015. URL: <https://en.wikipedia.org/wiki/Arduino>.
- [6] John Boxall. *Arduino workshop: A Hands-On introduction with 65 projects*. No Starch Press, 2013. ISBN: 978-1-59327-448-1.
- [7] Atmel Corporation. *AVR4201: Pressure One (ATAVRSBPR1) Hardware Users Guide*. Rev. 8355A-AVR-12/10. 2010. URL: <http://www.farnell.com/datasheets/1509376.pdf>.
- [8] Element14. *XTRINSIC-SENSE Board*. Rev. 0.4, 01/09/2014. 2014. URL: <http://www.farnell.com/datasheets/1804142.pdf>.
- [9] Nick Gammon. *I2C - Two-Wire Peripheral Interface - for Arduino*. 2011. URL: <http://gammon.com.au/i2c>.
- [10] M Gonzalez-Berges. «The joint controls project framework». In: *arXiv preprint physics/0305128* (2003).
- [11] <https://opcfoundation.org/>.
- [12] <https://root.cern.ch/drupal/>.
- [13] <https://twiki.cern.ch/twiki/bin/view/MPGD/SloCSy>.
- [14] <https://www.adafruit.com/>.
- [15] <https://www.sparkfun.com/>.
- [16] <http://wikis.web.cern.ch/wikis/display/EN/WinCC-OA+Service>.
- [17] <http://www.st4makers.com/>.
- [18] *I²C — Wikipedia, The Free Encyclopedia*. 2015. URL: <https://en.wikipedia.org/wiki/I%C3%82%C2%B2C>.
- [19] K Karakostas, T Alexopoulos, and G Tsiopolitis. «A Slow control system for RD51 test facilities». In: *RD51 Note* 2011-011 ().

- [20] Aosong Electronics Co. Ltd. *Digital-output relative humidity and temperature sensor/module DHT22*. URL: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>.
- [21] *Object Linking and Embedding* — Wikipedia, The Free Encyclopedia. 2015. URL: https://en.wikipedia.org/wiki/Object_Linking_and_Embedding.
- [22] *Open Platform Communications* — Wikipedia, The Free Encyclopedia. 2015. URL: https://en.wikipedia.org/wiki/Open_Platform_Communications.
- [23] Fairchild Semiconductor. *BSS138 N-Channel Logic Level Enhancement Mode Field Effect Transistor*. Rev C(W). 2005. URL: <http://www.adafruit.com/datasheets/BSS138.pdf>.
- [24] Freescale Semiconductor. *Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated and Calibrated*. Rev 10, 10/2012. 2012. URL: http://cache.freescale.com/files/sensors/doc/data_sheet/MPX5700.pdf?pspl1=1.
- [25] Freescale Semiconductor. *Xtrinsic MPL3115A2 I²C Precision Altimeter*. MPL3115A2 Rev 3.0 12/2013. 2013. URL: http://cache.freescale.com/files/sensors/doc/data_sheet/MPL3115A2.pdf.
- [26] NXP Semiconductors. *I²C bus spicification and user manual*. Rev. 8v.6. 2014. URL: http://www.nxp.com/documents/user_manual/UM10204.pdf.
- [27] NXP Semiconductors. *Level shifting techniques in I²C bus design*. Application Note AN10441, Rev. 01. 2007. URL: <http://www.adafruit.com/datasheets/AN10441.pdf>.
- [28] Philips Semiconductors. *Bi-directional level shifter for I²C bus and other systems*. Application Note AN97055. 1997. URL: <http://www.adafruit.com/datasheets/an97055.pdf>.
- [29] Bosch Sensortec. *BMP085 Digital pressure sensor Data sheet*. Rev. 1.0 , Document number BST-BMP085-DS000-03. 2008. URL: https://www.adafruit.com/datasheets/BMP085_DataSheet_Rev.1.0_01July2008.pdf.
- [30] VAISALA. *HMP230 Series Transmitters User's Guide*. M210225en-B. 202. URL: http://www.vaisala.com/Vaisala%20Documents/User%20Guides%20and%20Quick%20Ref%20Guides/HMP230_UserGuide.pdf.
- [31] WIZnet. *W5100 Datasheet*. Version 1.1.6. 2008. URL: https://www.sparkfun.com/datasheets/DevTools/Arduino/W5100_Datasheet_v1_1_6.pdf.

- [32] Ανδρέου Δήμητρα. «Ανάπτυξη συστήματος μέτρησης θερμοκρασίας και υγρασίας βασισμένο σε Arduino και WinCC_OA». Προπτυχιακή διπλωματική εργασία. Εθνικό Μετσόβιο Πολυτεχνείο, 2015. URL: <http://dspace.lib.ntua.gr/handle/123456789/40468>.
- [33] Κωσταντίνος Καρακώστας. «Micromegas Telescope for the RD51 Test Beams». MA thesis. Εθνικό Μετσόβιο Πολυτεχνείο, 2010.
- [34] Παναγιώτης Μ. Παπάζογλου. *Ανάπτυξη εφαρμογών με το Arduino*. Εκδόσεις Τζιόλα, 2014. ISBN: 978-960-418-459-0.