

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

**ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ
ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**

**ΤΟΜΕΑΣ ΦΥΣΙΚΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΠΕΙΡΑΜΑΤΙΚΗΣ ΦΥΣΙΚΗΣ ΥΨΗΛΩΝ
ΕΝΕΡΓΕΙΩΝ**



**ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΜΕΤΡΗΣΗΣ
ΠΕΡΙΒΑΛΛΟΝΤΙΚΩΝ ΣΥΝΘΗΚΩΝ
ΒΑΣΙΣΜΕΝΟ ΣΕ ARDUINO YUN ΚΑΙ
WINCC_OA**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
ΤΟΥ
ΓΕΩΡΓΙΟΥ ΜΠΑΚΑ**

Επιβλέπων: Τσιπολίτης Γεώργιος
Καθηγητής ΕΜΠ

ΑΘΗΝΑ, ΜΑΪΟΣ 2016

Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός συστήματος μέτρησης και καταγραφής περιβαλλοντικών συνθηκών με χρήση της μικρο-υπολογιστικής πλακέτας Arduino YUN και κατάλληλων αισθητήρων. Πιο συγκεκριμένα, η χρήση των αισθητήρων αυτών γίνεται με σκοπό την παρακολούθηση πίεσης, θερμοκρασίας και υγρασίας και η αποθήκευση των τιμών αυτών για περαιτέρω ανάλυση. Η διαδικασία ξεκινά με το Arduino να λαμβάνει τιμές από τους αισθητήρες αυτούς. Στη συνέχεια τα δεδομένα αυτά στέλνονται μέσω ενός OPC Server στο WinCC_OA ώστε να γίνει η επεξεργασία τους. Το Project αυτό ενσωματώνεται στο ήδη υπάρχον Slow Control System (SLOCSY) του RD51 Collaboration του CERN. Η παρακολούθηση των περιβαλλοντικών συνθηκών στους ανιχνευτές κατά τη διάρκεια των test beams αποτελεί πολύ σημαντική διαδικασία το ίδιο σημαντική με την ήδη υπάρχουσα παρακολούθηση των τάσεων.

Abstract

The goal of this thesis is the development of a system that measures and records environmental conditions using a set of environmental sensors and the microcontroller Arduino YUN. More precisely, these sensors are used in order to measure temperature, pressure and humidity and to save these measurements for further analysis. The process starts with the Arduino Yun receiving data from the sensors. These data are sent via an OPC Server to the WinCC_OA where the data is analysed. This project incorporates to the Slow Control System (SLOCSY) of RD51 Collaboration of CERN. The Observation of the environmental parameters at the detectors during test beams is a very important procedure as it provides information for the smooth operation of the detectors.

Περιεχόμενα

Κατάλογος Σχημάτων	9
1 Μικρουπολογιστική Πλακέτα Arduino	13
1.1 Εισαγωγή	13
1.2 Arduino Yun	14
1.3 Αλληλεπίδραση με το Arduino	19
1.4 Σύνδεση Arduino Yun με wifi δίκτυο	22
2 Αισθητήρες	23
2.1 Αισθητήρας DHT22	23
2.1.1 Περιγραφή	23
2.2 Αισθητήρας BME280	26
2.2.1 Περιγραφή	26
2.2.2 Μετρήσεις	29
2.3 Αισθητήρας BMP085	31
2.3.1 Περιγραφή	31
2.4 I^2C Πρωτόκολλο (Inter-Integrated Circuit Protocol)	34
2.5 TCA9548A 1 to 8 I^2C Multiplexer Breakout	37
2.6 Συγκρίσεις Τιμών Αισθητήρων BME, BMP, DHT	38
3 OPC Server	41
3.1 Εισαγωγή	41
3.2 Arduino OPC Server	42
3.3 Εγκατάσταση και Χρήση Arduino OPC Server	44
4 Πλακέτα Τυπωμένου Κυκλώματος	47
4.1 Eagle Cadsoft	47
4.1.1 Εισαγωγή	47
4.1.2 Schematic Editor	48
4.1.3 Layout Editor	50
4.1.4 Autorouter	52

4.2	Boards στην Εργασία	53
4.2.1	BME και DHT Board	53
4.2.2	Main Arduino YUN/Mega Board	56
5	Ανάπτυξη Συστήματος με WinCC OA	61
5.1	WinCC OA	61
5.1.1	Εισαγωγή	61
5.1.2	Βασικά Χαρακτηριστικά Του WinCC OA	62
5.1.3	JCOP Framework	64
5.2	Οδηγίες Σύνδεσης του Arduino με το WinCC OA	65
5.2.1	Δημιουργία Καινούργιου project στο WinCC OA	65
5.3	Εγκατάσταση JCOP Framework Components	68
5.4	Slow Control System	70
5.4.1	Configuration	71
5.4.2	Archive	74
5.4.3	Export	75
5.4.4	All Plots	76
5.4.5	Real Time Data	77
6	Συμπεράσματα	79

Κατάλογος Σχημάτων

1.1	Μικροπολογιστική πλακέτα Arduino	13
1.2	Front and rear view of arduino yun	14
1.3	Indicating power, WLAN connection, WAN connection and USB.	16
1.4	Pin Mapping for ATmega32U4.	18
1.5	Arduino IDE main page	19
1.6	Σελίδα στο Yun wifi configuration	22
1.7	Σελίδα στο Yun wifi configuration	22
2.1	Ψηφιακός Αισθητήρας Πίεσης και Θερμοκρασίας	23
2.2	Τεχνικά χαρακτηριστικά του DHT22	24
2.3	Διαστάσεις του DHT22	24
2.4	Pin Mapping του DHT22	24
2.5	Arduino Yun -DHT22 connection	25
2.6	Αισθητήρας BME280	26
2.7	Γενικές Ηλεκτρικές Προδιαγραφές Αισθητήρα	27
2.8	Προδιαγραφές Παραμέτρων Υγρασίας	27
2.9	Προδιαγραφές Παραμέτρων Πίεσης	28
2.10	Προδιαγραφές Παραμέτρων Θερμοκρασίας	28
2.11	Διαγράμματα Πίεσης	29
2.12	Διαγράμματα θερμοκρασίας	30
2.13	Διαγράμματα Υγρασίας	30
2.14	Ψηφιακός Αισθητήρας BMP085	31
2.15	ATMEL AVR4201 με τον Αισθητήρα BMP085	31
2.16	Χαρακτηριστικά του Αισθητήρα BMP085	32
2.17	Pin Mapping για τον BMP085	33
2.18	Bidirectional Level Shifter	33
2.19	Απεικόνιση των γραμμών SDA, SCL κατά τη διάρκεια επικοινωνίας με χρήση του πρωτοκόλλου I^2C (7 bit)	34
2.20	Λογικός Πίνακας διαδικασίας λήψης δεδομένων από τον BMP085	35

2.21	Αναλυτικός πίνακας του memory map για τον αισθητήρα bme 280	36
2.22	Αναλυτικός πίνακας των registers για τον αισθητήρα bmp085	36
2.23	Πολυπλέκτης TCA9548A	37
2.24	Διαγράμμα Πίεσης των αισθητήρων BME και BMP	38
2.25	Διαγράμμα Θερμοκρασίας των αισθητήρων BME, BMP και DHT	38
2.26	Διαγράμμα Πίεσης των αισθητήρων BME και DHT	39
3.1	Διάγραμμα Επικοινωνίας με χρήση του πρωτοκόλλου OPC	41
3.2	Arduino OPC Server	43
3.3	Φάκελος Arduino OPC Server	44
3.4	Εκκίνηση Arduino OPC Server	45
3.5	Arduino OPC Server	45
3.6	Configuration Tab, Arduino OPC Server	46
4.1	Αρχική σελίδα EAGLE	48
4.2	Schematic Editor στο EAGLE	49
4.3	Παράδειγμα Schematic Editor	50
4.4	Μεταφορά στο Layout Editor	50
4.5	Layout Editor	51
4.6	Παράδειγμα Layout Editor	52
4.7	Παράθυρο Autorouter	53
4.8	Schematic του BME-DHT Board	54
4.9	Layout του BME-DHT Board μετά την χρήση του Autorouter	54
4.10	Schematic	56
4.11	Layout του BME-DHT Digital Board μετά την χρήση του Autorouter	56
4.12	Schematic του Arduino main Board	57
4.13	Layout του Arduino Main Board μετά την χρήση του Autorouter	57
4.14	Schematic του Arduino Main Board (Digital)	59
4.15	Layout του Arduino Main Board(Digital) μετά την χρήση του Autorouter	59
5.1	WinCC OA Data Structure	63
5.2	Επιλογή Distributed Project κατά τη διαδικασία δημιουργίας νέου project στο WinCC OA	65
5.3	Εισαγωγή Ονόματος κατά τη δημιουργία νέου project στο WinCC OA	66
5.4	Εισαγωγή System Name κατά τη δημιουργία νέου project στο WinCC OA	66

5.5	Διαδικασία δήλωσης του OPC Server στο configuration file του project	67
5.6	Διαδικασία δήλωσης του OPC DA Client	67
5.7	Άνοιγμα fwInstallation Panel με την επιλογή Module →Vision	68
5.8	Επιλογή fwInstallation και άνοιγμα του fwInstallation.pnl . . .	69
5.9	Παράθυρο fwInstallation Panel για την εγκατάσταση των components	69
5.10	Κύριο Παράθυρο Slow Control System	70
5.11	Κεντρικό Παράθυρο Environmental Parameters	71
5.12	Παράθυρο Configuration για το Arduino YUN	71
5.13	Configure Yun Panel	72
5.14	Configure Yun Panel(2)	72
5.15	Ορισμός αριθμού αισθητήρων για κάθε PCB	73
5.16	OPC Configuration Panel	73
5.17	Configuration OPC Panel(2)	74
5.18	Archiving Panel	74
5.19	Export Panel	75
5.20	All Plots Panel	76
5.21	Ardenvino Project Main Panel	77
5.22	All Plots Panel	77
5.23	Real Time Temperature Measurements	78
5.24	Real Time Pressure Measurements	78
5.25	Real Time Humidity Measurements	78
6.1	Θερμοκρασία από τους αισθητήρες BME και DHT κατά τη διάρκεια test beam	80
6.2	Πίεση από τον αισθητήρα BME κατά τη διάρκεια test beam . . .	80
6.3	Υγρασία από τους αισθητήρες BME και DHT κατά τη διάρκεια test beam	81
6.4	Ardenvino με χρήση Arduino YUN	81
6.5	Ardenvino με χρήση Arduino Mega	82
6.6	Δεδομένα που πάρθηκαν με χρήση ArdEnvino σε πραγματικό χρόνο	82

Κεφάλαιο 1

Μικροπολογιστική Πλακέτα Arduino

Η ενότητα αυτή περιλαμβάνει την γενική παρουσίαση της μικροπολογιστικής πλακέτας Arduino και πιο συγκεκριμένα της μικροπολογιστικής πλακέτας Arduino Yun που χρησιμοποιήθηκε στην παρούσα εργασία.

1.1 Εισαγωγή



Figure 1.1: Μικροπολογιστική πλακέτα Arduino

Το Arduino είναι μία μικροϋπολογιστική πλατφόρμα, η οποία βασισμένη σε έναν μικροελεγκτή μπορεί να χρησιμοποιηθεί για την κατασκευή ηλεκτρονικών συστημάτων τα οποία αλληλεπιδρούν με τον πραγματικό κόσμο. Αυτή η επικοινωνία μικροεπεξεργαστή και πραγματικού κόσμου γίνεται χρησιμοποιώντας αισθητήρες οι οποίοι μετρούν παραμέτρους από τον έξω κόσμο.

Η πλακέτα αποτελείται από έναν μικροελεγκτή και μία σειρά από επαφές εισόδου/εξόδου (I/O pins). Το εκάστοτε είδος του μικροελεγκτή, ο αριθμός και η λειτουργία των Pins έχουν σαν αποτέλεσμα την ύπαρξη μίας ολόκληρης γενιάς Arduino. Το arduino είναι ένα προϊόν που διατίθεται για το ευρύ

κοινό καθώς τα τελευταία χρόνια χρησιμοποιείται ευρέως σε τεχνολογικές εφαρμογές. Τα pins του arduino χωρίζονται σε ψηφιακά και αναλογικά. Επίσης το arduino εκτός από SPI Connections (Serial Peripheral Interface) δίνει την δυνατότητα στον χρήστη να μπορεί να επικοινωνεί με το arduino μέσω I^2C σύνδεσης (Inter-integrated Circuit). Παρατηρούμε συνεπώς ότι το Arduino αποτελεί μία εύχρηστη πλατφόρμα για ανάπτυξη απλών μέχρι και πολύ σύνθετων τεχνολογικών εφαρμογών.

1.2 Arduino Yun

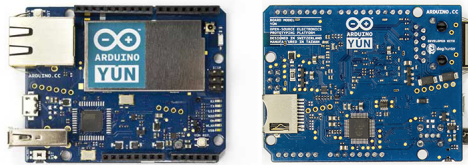


Figure 1.2: Front and rear view of arduino yun

Γενικά Στοιχεία

Στην εργασία αυτή χρησιμοποιήθηκε η πλακέτα Arduino Yun. Αυτό που ξεχωρίζει την πλακέτα αυτή από τις άλλες arduino πλακέτες είναι ότι μπορεί να έχει επικοινωνία με υπολογιστή/server κλπ χωρίς την χρήση καλωδίων, αλλά με σύνδεση στο διαδίκτυο μέσω wifi. Μπορεί κανείς δηλαδή να "φορτώσει" προγράμματα στο arduino yun μέσω wifi. Γενικότερα όμως το arduino yun αποτελείται από :

- Επεξεργαστή ATmega32U4
- Αποτελείται από 20 ψηφιακά (digital I/O) pins τα οποία χρησιμοποιούνται είτε σαν είσοδοι είτε σαν έξοδοι. Τα 7 από αυτά μπορούν επίσης να χρησιμοποιηθούν σαν έξοδοι διαμόρφωσης πλάτους παλμού (PWM = Pulse Width Modulation).
- 12 αναλογικά pins
- έναν χρυσταλλικό ταλαντωτή (clock) 16MHz

- Μία θύρα USB και μία θύρα Micro-USB (σαν υποδοχή ρεύματος)
- Μία θύρα Ethernet
- Ένα κουμπί επαναφοράς του επεξεργαστή
- Ένα κουμπί επαναφοράς του wifi configuration
- Ένα κουμπί καθολικής επαναφοράς του Arduino Yun

Παροχή Τάσης

Το Arduino μπορεί να τροφοδοτηθεί είτε μέσω θύρας USB είτε μέσω θύρας micro usb. Επίσης το arduino περιέχει pins τα οποία μπορούν να χρησιμοποιηθούν για την παροχή ρεύματος/τάσης σε άλλα εξωτερικά κυκλώματα.

- V_{in} : Παρέχει τάση ίση με την τάση εισόδου όταν το Arduino τροφοδοτείται από εξωτερική πηγή τάσης
- **5V**: Παρέχει τάση ίση με 5V
- **3.3V**: Παρέχει 3.3V τάση και ρεύμα 50mA (ενώ όλα τα υπόλοιπα Pins 40 ma)
- **GND**: Γειωμένα pins
- **IOREF**: Παρέχει την τάση αναφοράς με την οποία λειτουργεί η πλακέτα

Μνήμη

Το Arduino yun διαθέτει συνολικά 32kB μνήμη, εκ των οποίων τα 2.5kB είναι SRAM (Static Random Acces Memory) και το 1kB EEPROM (Electrically Erasable Programmable Read Only Memory).

Pin mapping

Καθένα από τα ψηφιακά pins μπορούν να χρησιμοποιηθούν είτε ως είσοδοι είτε ως έξοδοι ψηφιακών σημάτων, δηλαδή μπορούν να διαβάσουν ή και να στείλουν ψηφιακά σήματα. Η πλειοψηφία των Pins του Arduino εκτός από την λειτουργία εισόδου/εξόδου έχουν και επιπρόσθετες λειτουργίες. Πιο συγκεκριμένα:

- **Serial**: 0(RX) and 1 (TX). Τα Pins αυτά λειτουργούν σαν ζεύγος και μπορούν να δέχονται (RX) και να μεταφέρουν (TX) TTL σειριακά δεδομένα (transistor-transistor logic).

- **External Interrupts:** 3 (interrupt 0), 2 (interrupt 1), 0 (interrupt 2), 1 (interrupt 3) και 7 (interrupt 4). Τα pins αυτά μπορούν να ρυθμιστούν έτσι ώστε να διακόπτεται η λειτουργία του μικροελεγκτή και να ακολουθείται μία άλλη προγραμματισμένη ενέργεια η οποία κάθε φορά εξαρτάται από την συμπεριφορά της τιμής κάποιου από τα Pins.
- **PWM:** Τα pins 3, 5, 6, 9, 10, 11, και 13 προσφέρουν 8-bit διαμόρφωση πλάτους παλμού (Pulse Width Modulation).
- **TWI:** 2 (SDA) and 3 (SCL). Τα pins αυτά παρέχουν την δυνατότητα Two Wire Interface.
- **AREF:** Είναι το Pin το οποίο λειτουργεί ως σημείο αναφοράς για το εύρος μέτρησης των αναλογικών εισόδων του Arduino.
- **Analog pins:** A0 - A5, A6 - A11(στα digital pins 4,6,8,9,10,12). Το Yun έχει 12 analog pins τα οποία μπορούν να χρησιμοποιηθούν και ως digital i/o. Κάθε αναλογικό pin παρέχει 10-bit ανάλυση (δηλαδή μπορεί να μετρήσει 1024 διαφορετικές τιμές). Τα Pins αυτά είναι ρυθμισμένα να μετρούν τις τιμές αυτές σε ένα εύρος από 0 έως και 5V.
- **LED:** Στο arduino υπάρχει εκ κατασκευής ένα led το οποίο ρυθμίζεται από το Pin 13 του arduino. Όταν το pin 13 δίνει ρεύμα στο led, τότε το led ανάβει. Σε κάθε άλλη περίπτωση το led παραμένει σβηστό.
- **Other led:** Όπως φαίνεται και παρακάτω, υπάρχουν και άλλα led πάνω στην πλακέτα. Κάθε ένα από αυτά είναι υπεύθυνο να ενημερώνει τον χρήστη για μία λειτουργία όπως φαίνεται και στο σχέδιο.

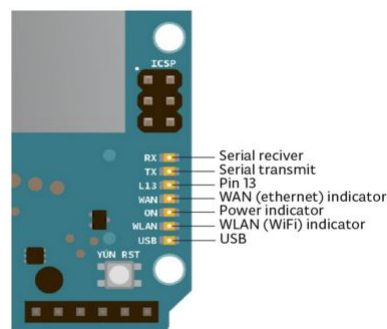


Figure 1.3: Indicating power, WLAN connection, WAN connection and USB.

Reset Buttons Εδώ θα γίνει περιγραφή των 3 reset buttons που βρίσκονται στο arduino yun.

- **YUN RST:** Το κουμπί αυτό όταν πατηθεί κάνει επαναφορά του μικροεπεξεργαστή AR9331. Η επαναφορά αυτή θα προκαλέσει την επανεκκίνηση του συστήματος Linux. Η επαναφορά θα έχει επίσης ως αποτέλεσμα όλα τα στοιχεία που είναι αποθηκευμένα στην RAM να χαθούν και κάθε πρόγραμμα που "τρέχει" στην πλακέτα να τερματίσει.
- **32U4 RST:** Το κουμπί αυτό όταν πατηθεί κάνει επαναφορά του μικροεπεξεργαστή ATmega32U4.
- **WLAN RST:** Αυτό το κουμπί έχει διπλή λειτουργία. Κυρίως χρησιμοποιείται για να επαναφέρει τις ρυθμίσεις wifi για το arduino yun στις αρχικές εργοστασιακές συνθήκες. Μία τέτοια επαναφορά βάζει το Yun σε κατάσταση access mode point (AP) και του εκχωρεί το IP address: 192.168.240.1. Έτσι μπορεί οποιοσδήποτε χρήστης να συνδεθεί μέσω wifi με το arduino που έχει SSID: "Arduino Yun-XXXXXXXXXXXX" όπου X είναι το MAC address που υπάρχει πάνω στο arduino. Μόλις υπάρχει σύνδεση με το Yun, ο χρήστης πληκτρολογεί στον web browser 192.168.240.1 ή "http://arduino.local" και μπορεί πλέον να συνδεθεί με το web panel του Arduino, στο οποίο μπορεί να συνδεθεί το arduino με άλλα wifi connections. Για την πρώτη λειτουργία αυτού του κουμπιού απαιτείται ο χρήστης να πιέσει το κουμπί παρατεταμένα για τουλάχιστον 5 δευτερόλεπτα. Αν ο χρήστης πιέσει το κουμπί για 30 δευτερόλεπτα, ενεργοποιείται η δεύτερη χρήση του συγκεκριμένου κουμπιού, η οποία είναι η επανεκκίνηση του περιβάλλοντος του linux. Αν γίνει αυτό, διαγράφονται όλα τα αρχεία και κάθε λογισμικό που είναι αποθηκευμένο στην flash memory του arduino που είναι συνδεδεμένη με τον μικροεπεξεργαστή AR9331.

Παρακάτω δίνεται το pin mapping για τον μικροεπεξεργαστή ATmega32U4.

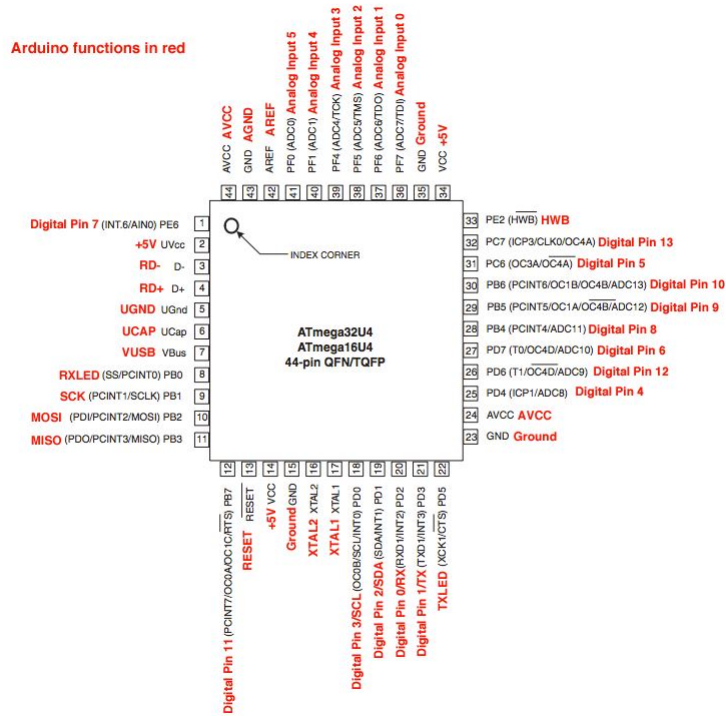


Figure 1.4: Pin Mapping for ATmega32U4.

1.3 Αλληλεπίδραση με το Arduino

Για την χρήση του Arduino , απαιτείται ο χρήστης να γράψει πρόγραμμα στο περιβάλλον εργασίας Arduino ή αλλιώς Arduino IDE (Integrated Development Environment). Το IDE διατίθεται δωρεάν στο διαδίκτυο από την Arduino και χρησιμοποιείται για το γράψιμο κώδικα όσο και για "φόρτωμα" (upload) του κώδικα στην πλακέτα. Παρακάτω δίνεται μία εικόνα του Arduino IDE.

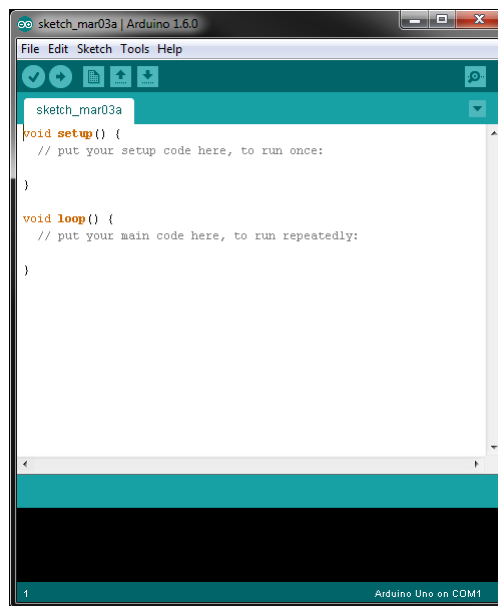


Figure 1.5: Arduino IDE main page

Δημιουργία Σκίτσων (Sketch)

Ο κώδικας που γράφει ο χρήστης στο IDE ονομάζεται σκίτσο (sketch). Αυτά γράφονται με τον επεξεργαστή κειμένου του IDE και σώζονται ως αρχεία με κατάληξη .ino. Τα σκίτσα, σώζονται σε έναν φάκελο ο οποίος ονομάζεται Sketchbook, και ανοίγει αυτόματα κατά την εκκίνηση του Arduino IDE. Στην περιοχή μηνυμάτων , εμφανίζονται πληροφορίες σχετικές με τις ενέργειες του χρήστη (όπως για παράδειγμα το σώσιμο ενός αρχείου). Στην κονσόλα εμφανίζονται πληροφορίες σχετικά με την μνήμη που καταναλώνει κάθε sketch όταν γίνεται upload στην πλακέτα, καθώς και λάθη που σχετίζονται με τον κώδικα όταν εκείνος δεν συντάσσεται σωστά (compilation errors).

Το Arduino IDE διαθέτει μία πληθώρα κουμπιών και μενού. Πιο αναλυτικά:

- **Verify:** Εδώ γίνεται έλεγχος του κώδικα για τυχόν συντακτικά λάθη χωρίς όμως να γίνεται upload του κώδικα στο Arduino
- **Upload:** Το κουμπί αυτό ελέγχει τον κώδικα για τυχόν συντακτικά λάθη, καθώς και κάνει upload τον κώδικα στο Arduino.
- **New:** Με το κουμπί αυτό ο χρήστης δημιουργεί ένα καινούργιο κενό sketch.
- **Open:** Με το κουμπί αυτό ο χρήστης μπορεί να περιηγηθεί στον φάκελο Sketchbook και να ανοίξει ένα από τα αποθηκευμένα σκίτσα.
- **Save:** Αποθήκευση παρόντος sketch
- **Serial Monitor:** Το κουμπί αυτό ανοίγει την σειριακή οθόνη (Serial Monitor) με την οποία μπορεί να επικοινωνεί το arduino και να εκτυπώνει να μεταδίδει και να δέχεται δεδομένα.

Ακόμα υπάρχουν τα menus : "File", "Edit", "Sketch", "Tools", "Help" τα οποία προσφέρουν ακόμα περισσότερες δυνατότητες στον χρήστη. Ο κώδικας ο οποίος είναι συμβατός με το Arduino είναι σε γλώσσα C και C++. Ο κώδικας που "φορτώνεται" στο Arduino μπορεί να αποτελείται από περισσότερα από ένα sketches ή αρχεία ή βιβλιοθήκες γραμμένα σε γλώσσα Arduino, C ή C++.

Φόρτωση Κώδικα (Uploading) στο Arduino

Για να γίνει η φόρτωση του προγράμματος στο Arduino, θα πρέπει να έχει επιλεγεί από το Tools menu του IDE, το είδος του Arduino που χρησιμοποιούμε, καθώς και η θύρα στην οποία είναι συνδεδεμένο το Arduino. Όταν πατηθεί το Upload, τα led TX, RX LED στην πλακέτα αναβοσβήνουν, πράγμα που υποδηλώνει ότι η πλακέτα στέλνει και λαμβάνει δεδομένα. Στην κονσόλα εμφανίζονται επίσης μηνύματα σχετικά με τη διαδικασία, δηλαδή το κατά πόσο αυτή ολοκληρώθηκε με επιτυχία ή όχι, και αν όχι εμφανίζονται τα σχετικά προβλήματα.

Βιβλιοθήκες

Οι βιβλιοθήκες είναι κώδικες ξεχωριστοί από τα σκίτσα. Μέσω αυτών το Arduino αποκτά περισσότερες δυνατότητες. Για παράδειγμα μία βιβλιοθήκη δίνει την δυνατότητα στο Arduino να επικοινωνεί με μία εξωτερική συσκευή, επεξεργασία δεδομένων κλπ. Επίσης μία βιβλιοθήκη δίνει την δυνατότητα στο arduino να επικοινωνεί με έναν αισθητήρα. Η βιβλιοθήκη καθορίζει πώς το arduino λαμβάνει και στέλνει μηνύματα. Οι βιβλιοθήκες δηλώνονται στην αρχή κάθε σκίτσου στο οποίο και χρησιμοποιούνται. Το Arduino αναγνωρίζει αυτές τις βιβλιοθήκες μόνο όταν είναι αποθηκευμένες στον υποφάκελο

libraries του φακέλου Sketchbook. Υπάρχουν βιβλιοθήκες που βρίσκονται ήδη μέσα στον φάκελο από την πρώτη κιόλας εγκατάσταση του Arduino IDE, ενώ μπορεί ο χρήστης να βρει βιβλιοθήκες διαθέσιμες στο διαδίκτυο ή ακόμα και να δημιουργήσει ο ίδιος τις δικές του.

Serial Monitor:

Όπως αναφέρθηκε και προηγουμένως πατώντας το κουμπί Serial Monitor, ανοίγει σε νέο παράθυρο η σειριακή οθόνη. Σε αυτή τυπώνονται δεδομένα από το Arduino εφόσον ο χρήστης έχει προγραμματίσει κατάλληλα το πρόγραμμα που έχει γίνει upload στο Arduino. Επίσης ο χρήστης μπορεί να στείλει δεδομένα μέσω του Serial Monitor.

1.4 Σύνδεση Arduino Yun με wifi δίκτυο

Στο σημείο αυτό, θα αναλυθεί ο τρόπος με τον οποίο ο χρήστης μπορεί να συνδέσει το Arduino με ένα δίκτυο Wifi, έτσι ώστε να μπορεί να "φορτώνει" προγράμματα χωρίς την επικοινωνία υπολογιστή και Arduino μέσω καλωδίου.

Αρχικά ο χρήστης τοποθετεί την θύρα USB στον υπολογιστή έτσι ώστε να γίνει η ρύθμιση της wifi σύνδεσης. Ο χρήστης περιμένει μέχρι να ανοίξει το LED λαμπάκι USB του Arduino (λευκό). Μόλις γίνει αυτό, ο χρήστης πρέπει να συνδεθεί στο ασύρματο δίκτυο που έχει δημιουργήσει το Arduino, με όνομα: Arduino Yun-XXXXXXXXXXXX" όπου X είναι το MAC address που υπάρχει πάνω στο arduino. Στη συνέχεια, ο χρήστης απαιτείται να ανοίξει έναν web browser και να πληκτρολογήσει ως διεύθυνση: "http://arduino.local" ή "192.168.240.1". Κάνοντάς το αυτό, εμφανίζεται η παρακάτω οθόνη:

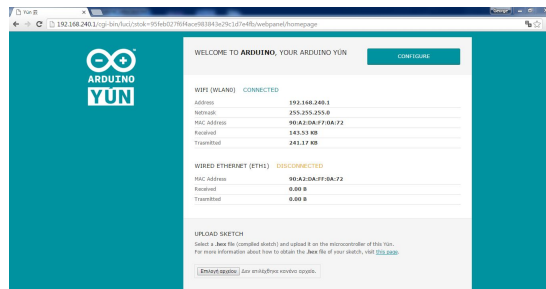


Figure 1.6: Σελίδα στο Yun wifi configuration

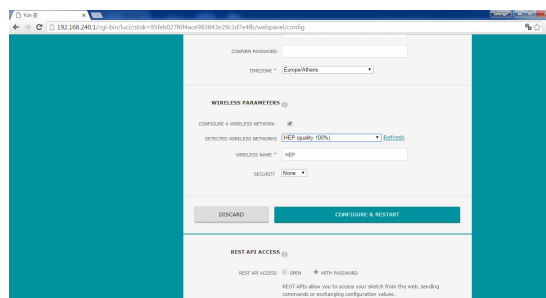


Figure 1.7: Σελίδα στο Yun wifi configuration

Ο χρήστης στη συνέχεια πρέπει να συνδέσει το Arduino με το δίκτυο στο οποίο είναι και εκείνος συνδεδεμένος. Πατώντας το κουμπί " " ο χρήστης περιμένει λίγη ώρα για να ολοκληρωθεί η διαδικασία. Μόλις ολοκληρωθεί η διαδικασία, ο χρήστης μπορεί πλέον να "φορτώσει" sketches από το Arduino IDE στο Arduino μέσω wifi. Για να πραγματοποιηθεί τελικά αυτή η διαδικασία, ο χρήστης πρέπει να επιλέξει "Tools"-> "Port" και από εκεί την IP του Arduino.

Κεφάλαιο 2

Αισθητήρες

Στην ενότητα αυτή παρουσιάζονται οι αισθητήρες που χρησιμοποιήθηκαν στην παρούσα εργασία. Πιο συγκεκριμένα, οι αισθητήρες οι οποίοι χρησιμοποιήθηκαν είναι οι εξής: DHT22, BME280 και BMP085. Σε αυτή την ενότητα λοιπόν, θα γίνει περιγραφή και ανάλυση του κάθε αισθητήρα ξεχωριστά, θα παρουσιαστούν αποτελέσματα που λήφθηκαν από τους συγκεκριμένους αισθητήρες από τον καθένα ξεχωριστά καθώς και αποτελέσματα που λήφθηκαν σε κοινή συνδεσμολογία.

2.1 Αισθητήρας DHT22

2.1.1 Περιγραφή

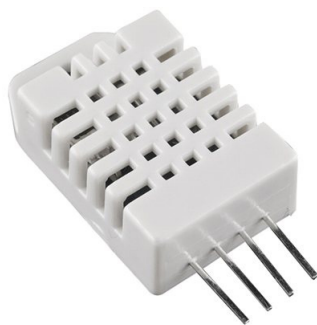


Figure 2.1: Ψηφιακός Αισθητήρας Πίεσης και Θερμοκρασίας

Ο αισθητήρας DHT22 ή αλλιώς AM2302, είναι ένας ψηφιακός αισθητήρας μέτρησης υγρασίας και θερμοκρασίας. Ο αισθητήρας αυτός αποτελεί ιδανικό μετρητή για εργασίες που σχετίζονται με Arduino καθώς μπορεί να μεταφερθεί

σε μεγάλες αποστάσεις, έχει πολύ χαμηλή κατανάλωση λειτουργίας και τέλος το κόστος του είναι πολύ μικρό.

Παρακάτω δίνονται τα τεχνικά χαρακτηριστικά με τα οποία λειτουργεί ο αισθητήρας DHT22.

Model	DHT22
Power supply	3.3-6V DC
Output signal	digital signal via single-bus
Sensing element	Polymer capacitor
Operating range	humidity 0-100%RH; temperature -40-80Celsius
Accuracy	humidity $\pm 2\%$ RH(Max $+5\%$ RH); temperature ± 0.5 Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
Repeatability	humidity $\pm 1\%$ RH; temperature ± 0.2 Celsius
Humidity hysteresis	$\pm 0.3\%$ RH
Long-term Stability	$\pm 0.5\%$ RH/year
Sensing period	Average: 2s
Interchangeability	fully interchangeable
Dimensions	small size 14*18*5.5mm; big size 22*28*5mm

Figure 2.2: Τεχνικά χαρακτηριστικά του DHT22

Επίσης δίνεται το Pin mapping του αισθητήρα και οι διαστάσεις του.

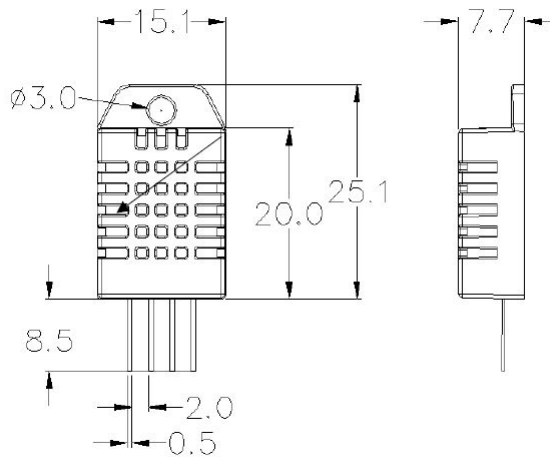


Figure 2.3: Διαστάσεις του DHT22

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

Figure 2.4: Pin Mapping του DHT22

Για την σύνδεση του αισθητήρα με το Arduino χρησιμοποιείται μία pull up αντίσταση 10kΩ καθώς και ένας πυκνωτής 100nF για φιλτράρισμα της τάσης

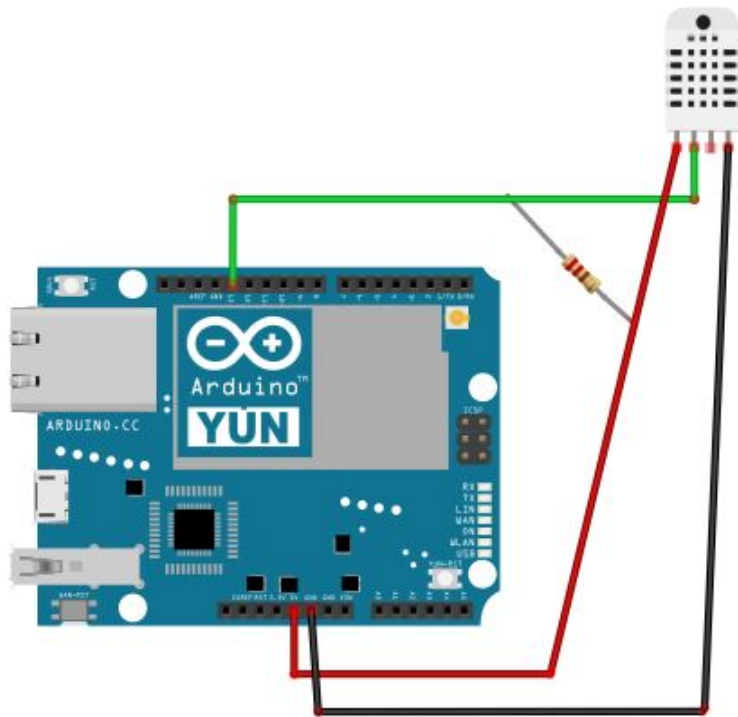


Figure 2.5: Arduino Yun -DHT22 connection

εισόδου όπως φαίνεται και παρακάτω.

2.2 Αισθητήρας BME280

2.2.1 Περιγραφή



Figure 2.6: Αισθητήρας BME280

Ο αισθητήρας BME280 είναι ένας ψηφιακός αισθητήρας πίεσης, θερμοκρασίας, υψόμετρου και υγρασίας. Ο αισθητήρας αυτός, είναι εξαιρετικά ακριβής, αρκετά οικονομικός και μπορεί να λειτουργήσει με είτε με SPI πρωτόκολλο (Standard peripheral Interface Protocol) είτε με χρήση του πρωτοκόλλου I^2C (inter-integrated protocol). Ο αισθητήρας διατίθεται από την Adafruit και είναι ιδανικός καθώς μπορούμε να πάρουμε μετρήσεις πίεσης, θερμοκρασίας και υγρασίας ταυτόχρονα, χρησιμοποιώντας έναν μόνο αισθητήρα.

Όπως αναφέρθηκε και προηγουμένως, η ακρίβεια των μετρήσεων του συγκεκριμένου αισθητήρα είναι πολύ μεγάλη. Γνωρίζουμε από τον κατασκευαστή, ότι η ακρίβεια μέτρησης υγρασίας είναι $\pm 3\%$, η ακρίβεια μέτρησης πίεσης είναι $\pm 1\text{hPa}$ και η ακρίβεια μέτρησης θερμοκρασίας είναι $\pm 1\text{ Celsius}$.

Παρακάτω δίνεται το Pin mapping για τον BME280.

1	2	3	4	5	6	7
Vin	3V ₀	GND	SCK	SD0	SD1	CS

Παρακάτω δίνονται τα χαρακτηριστικά λειτουργίας του BME280.

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Supply Voltage Internal Domains	V_{DD}	ripple max. 50 mVpp	1.71	1.8	3.6	V
Supply Voltage I/O Domain	V_{DDIO}		1.2	1.8	3.6	V
Sleep current	I_{DDSL}			0.1	0.3	μA
Standby current (inactive period of normal mode)	I_{DDSB}			0.2	0.5	μA
Current during humidity measurement	I_{DDH}	Max value at 85 °C		340		μA
Current during pressure measurement	I_{DDP}	Max value at -40 °C		714		μA
Current during temperature measurement	I_{DDT}	Max value at 85 °C		350		μA
Start-up time	$t_{startup}$	Time to first communication after both $V_{DD} > 1.58$ V and $V_{DDIO} > 0.65$ V			2	ms
Power supply rejection ratio (DC)	PSRR	full V_{DD} range			± 0.01 ± 5	%RH/V Pa/V
Standby time accuracy	$\Delta t_{standby}$			± 5	± 25	%

Figure 2.7: Γενικές Ηλεκτρικές Προδιαγραφές Αισθητήρα

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Operating range ³	R_H	For temperatures < 0 °C and > 60 °C see Figure 1	-40	25	85	°C
			0		100	%RH
Supply current	$I_{DD,H}$	1 Hz forced mode, humidity and temperature		1.8	2.8	μA
Absolute accuracy tolerance	A_H	20...80 %RH, 25 °C, including hysteresis		± 3		%RH
Hysteresis ⁴	H_H	10→90→10 %RH, 25 °C		± 1		%RH
Nonlinearity ⁵	NL_H	10→90 %RH, 25 °C		1		%RH
Response time to complete 63% of step ⁶	$\tau_{63\%}$	90→0 or 0→90 %RH, 25 °C		1		s
Resolution	R_H			0.008		%RH
Noise in humidity (RMS)	N_H	Highest oversampling, see chapter 3.6		0.02		%RH
Long term stability	ΔH_{stab}	10...90 %RH, 25 °C		0.5		%RH/year

Figure 2.8: Προδιαγραφές Παραμέτρων Υγρασίας

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Operating temperature range	T_A	operational	-40	25	+85	°C
		full accuracy	0		+65	
Operating pressure range	P	full accuracy	300		1100	hPa
Supply current	$I_{DD,LP}$	1 Hz forced mode, pressure and temperature, lowest power		2.8	4.2	μA
Temperature coefficient of offset ⁷	$TCOP$	25...65 °C, 900 hPa		±1.5		Pa/K
				±12.6		cm/K
Absolute accuracy pressure	$A_{P,full}$	300 ... 1100 hPa		±1.0		hPa
		0 ... 65 °C				
Relative accuracy pressure $V_{DD} = 3.3V$	A_{rel}	700 ... 900hPa 25 ... 40 °C		±0.12		hPa

Figure 2.9: Προδιαγραφές Παραμέτρων Πίεσης

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Operating range	T	Operational	-40	25	85	°C
		Full accuracy	0		65	°C
Supply current	$I_{DD,T}$	1 Hz forced mode, temperature measurement only		1.0		μA
Absolute accuracy temperature ⁸	$A_{T,25}$	25 °C		±0.5		°C
	$A_{T,full}$	0...65 °C		±1.0		°C
Output resolution	R_T	API output resolution		0.01		°C
RMS noise	N_T	Lowest oversampling		0.005		°C

Figure 2.10: Προδιαγραφές Παραμέτρων Θερμοκρασίας

2.2.2 Μετρήσεις

Στο σημείο αυτό θα παρουσιαστούν οι μετρήσεις που έγιναν για το καθορισμό της ακρίβειας των μετρήσεων των αισθητήρων αυτών. Αρχικά να σημειωθεί ότι η ακρίβεια των μετρήσεων είναι:

- **Θερμοκρασία:** ± 0.5 C
- **Πίεση:** ± 1 mbar
- **Υγρασία:** ± 3 %

Οι αισθητήρες αφέθηκαν για 11 ημέρες στο εργαστήριο με αποθήκευση τιμών κάθε χρονική στιγμή. Τα διαγράμματα που λήφθηκαν είναι τα παρακάτω 2.11 2.12 2.13:

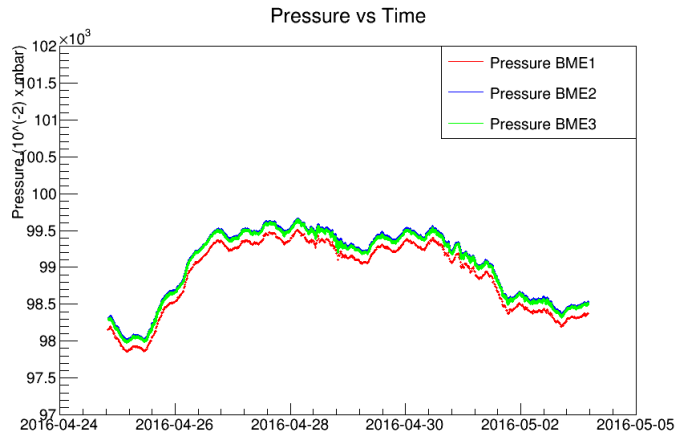


Figure 2.11: Διαγράμματα Πίεσης

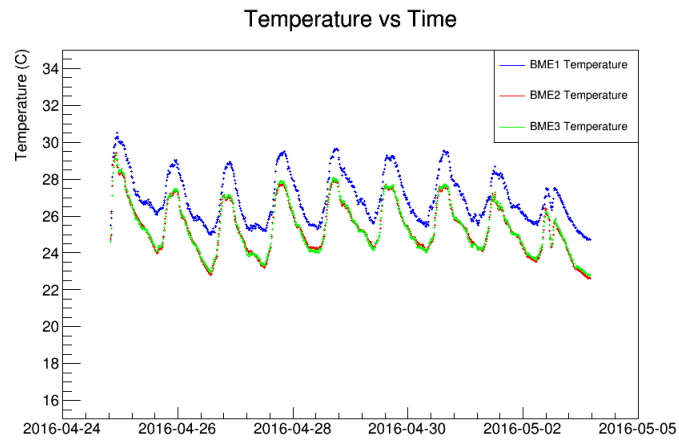


Figure 2.12: Διαγράμματα θερμοκρασίας

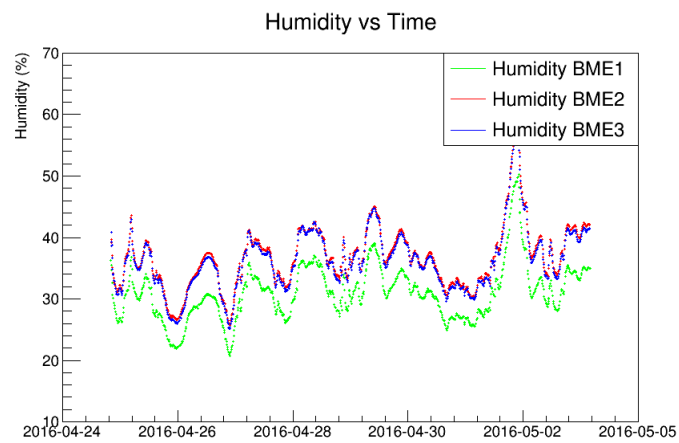


Figure 2.13: Διαγράμματα Υγρασίας

2.3 Αισθητήρας BMP085



Figure 2.14: Ψηφιακός Αισθητήρας BMP085

2.3.1 Περιγραφή

Ο BMP085 είναι ένας ψηφιακός αισθητήρας μέτρησης πίεσης, θερμοκρασίας και υψομέτρου. Πρόκειται για έναν εξαιρετικά ακριβή αισθητήρα, μικρό σε μέγεθος και με μικρή κατανάλωση ισχύος. Επίσης έχει την δυνατότητα σύνδεσης με I^2C πρωτόκολλο (Inter-Integrated Circuit protocol). Τα παραπάνω στοιχεία, καθιστούν τον αισθητήρα αυτό εξαιρετικό για χρήση σε εφαρμογές με μικροεπεξεργαστές, μικροελεγκτές όπως το Arduino. Ο συγκεκριμένος αισθητήρας προσφέρεται από αρκετές εταιρίες. Στην περίπτωση μας, χρησιμοποιήθηκε το sensor board AVR4201 της ATMEL, το οποίο φαίνεται παρακάτω στην εικόνα:

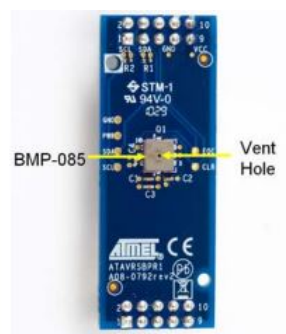


Figure 2.15: ATMEL AVR4201 με τον Αισθητήρα BMP085

Επίσης, στον παρακάτω πίνακα δίνονται μερικά χαρακτηριστικά του αισθητήρα αυτού:

Parameter	Symbol	Condition	Min	Typ	Max	Units
Operating temperature	T_A	operational	-40		+85	°C
		full accuracy	0		+65	
Supply voltage	V_{DD}	ripple max. 50mVpp	1.8	2.5	3.6	V
	V_{DDIO}		1.62	2.5	3.6	V
Supply current @ 1 sample / sec. 25°C	I_{DDLOW}	ultra low power mode		3		μA
	I_{DDSTD}	standard mode		5		μA
	I_{DDHR}	high resolution mode		7		μA
	I_{DDUHR}	ultra high res. mode		12		μA
Peak current	I_{peak}	during conversion		650	1000	μA
Standby current	I_{DDBEM}	at 25°C		0.1		μA
Serial data clock	f_{SCL}				3.4	MHz
Conversion time temperature	t_{C_temp}	standard mode		3	4.5	ms
Conversion time pressure	$t_{C_p_low}$	ultra low power mode		3	4.5	ms
	$t_{C_p_std}$	standard mode		5	7.5	ms
	$t_{C_p_hr}$	high resolution mode		9	13.5	ms
	$t_{C_p_uhr}$	ultra high res. mode		17	25.5	ms
Absolute accuracy pressure $V_{DD} = 3.3V$		700 ... 1100 hPa 0 ... +65 °C	-2.5	±1.0	+2.5	hPa
		300 ... 700 hPa 0 ... +65 °C	-3.0	±1.0	+3.0	hPa
		300 ... 1100 hPa -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 $V_{DD} = 3.3V$		700 ... 1100 hPa @ 25 °C		±0.2		hPa
		0 ... 65 °C @ p const.		±0.5		hPa
Noise in pressure		see table on page 10				
Absolute accuracy temperature $V_{DD} = 3.3V$		@ 25 °C	-1.5	±0.5	+1.5	°C
		0 ... +65 °C	-2.0	±1.0	+2.0	°C

Figure 2.16: Χαρακτηριστικά του Αισθητήρα BMP085

Τέλος να αναφερθεί ότι το I^2C πρωτόκολλο του Arduino λειτουργεί στα 5V ενώ του αισθητήρα στα 3.3V. Για τον λόγο αυτό χρησιμοποιήθηκε ένας **Bidirectional Level Shifter** κατασκευασμένος από την Adafruit (βλ εικόνα 2.17). Ο bidirectional level shifter είναι μία συσκευή που μετατρέπει τα 5V που στέλνει το Arduino σε 3.3V έτσι ώστε ο αισθητήρας να λειτουργεί κανονικά.

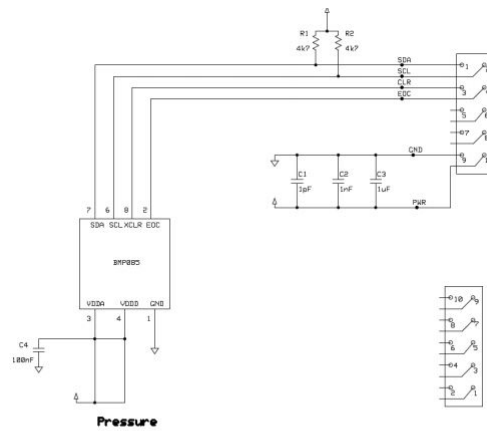


Table 5-1. Sensor I²C addresses.

Sensor	I ² C address
BMP085	0x77

Figure 2.17: Pin Mapping για τον BMP085

Από την άλλη, τα 3.3V που στέλνει ο αισθητήρας, μετατρέπονται σε 5V ώστε να μπορεί να τα διαβάσει το Arduino.



Figure 2.18: Bidirectional Level Shifter

2.4 I^2C Πρωτόκολο (Inter-Integrated Circuit Protocol)

Το πρωτόκολο αυτό δημιουργήθηκε από την εταιρία Phillips Semiconductor τη δεκαετία του '80. Χρησιμοποιείται συνήθως για την σύνδεση "αργών" περιφερειακών συσκευών με επεξεργαστές ή μικροελεγκτές. Ο μικροελεγκτής ονομάζεται **master** και όλες οι περιφερειακές συσκευές **slave**. Η συνδεσμολογία είναι αρκετά απλή και αυτό καθιστά την χρήση του συγκεκριμένου πρωτοκόλου αρκετά εύχρηστη. Πιο συγκεκριμένα η συνδεσμολογία αποτελείται από μόνο 2 γραμμές:

- **SDA (Serial Data Line)**
- **SCL (Serial Clock Line)**

Αυτό στην πράξη σημαίνει ότι ανεξάρτητα από τον αριθμό των slaves στο κύκλωμα, απαιτούνται μόνο δύο καλώδια για να επιτευχθεί επικοινωνία με όλους. Για την λειτουργία των slaves είναι προφανές ότι κάθε περιφερειακή μονάδα θα πρέπει να είναι συνδεδεμένη με μία τάση λειτουργίας και μία γείωση. Η γραμμή **SCL** ταλαντώνεται μεταξύ της κατάστασης 0 και 1 με μία συχνότητα η οποία ποικίλλει από τα 10 kbit/sec έως και 3.4 Mbit/sec. Ωστόσο, η πιο συνηθισμένη συχνότητα είναι 100 kbit/sec. Για κάθε στιγμή που η SCL βρίσκεται στην κατάσταση 1, η SDA αντιστοιχεί στο bit που διαβάζει ή στέλνει ο master.

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 την περαιτέρω εξήγηση της επικοινωνίας αυτής master-slave δίνεται η εικόνα 2.19

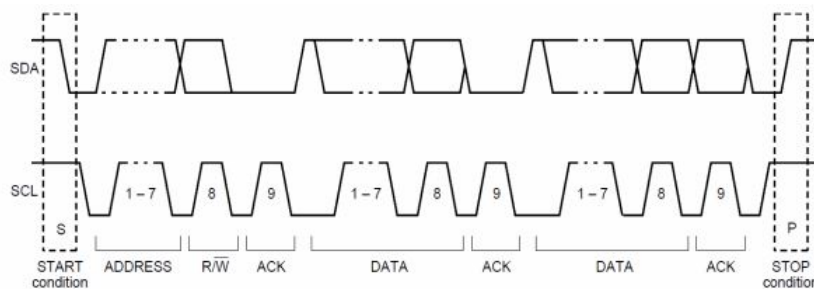


Figure 2.19: Απεικόνιση των γραμμών SDA, SCL κατά τη διάρκεια επικοινωνίας με χρήση του πρωτοκόλου I^2C (7 bit)

Στην αρχή και οι δύο γραμμές βρίσκονται σε υψηλή κατάσταση (high). Η επικοινωνία μεταξύ των δύο γραμμών ξεκινάει με το λεγόμενο **start condition** το οποίο, όπως φαίνεται και στο σχήμα 2.19 είναι όταν το SDA γίνεται

2.4. I²C ΠΡΩΤΟΚΟΛΟ (INTER-INTEGRATED CIRCUIT PROTOCOL) 35

ίσο με 0 (low) ενώ το SCL παραμένει υψηλό (high). Με την ενέργεια αυτή, ο master ενημερώνει όλους τους slaves ότι ξεκινάει η επικοινωνία.

Στη συνέχεια ο master στέλνει την διεύθυνση του slave με τον οποίο θέλει να επικοινωνήσει. Μετά από την λήψη του address από τον master, ο slave δέχεται 1 bit από τον master το οποίο υποδεικνύει στον slave την λειτουργία της γραμμής. Αν το bit αυτό είναι 0, τότε ο master γράφει στον slave. Αντίθετα αν το bit αυτό είναι 1, τότε ο master διαβάζει από τον slave.

Το τελευταίο bit που φαίνεται στην εικόνα παίζει τον ρόλο του "Ack" bit (Acknowledge bit). Με αυτό το bit, ο κάθε slave δηλώνει αν έχει όντως αυτή τη διεύθυνση ή όχι. Αυτό επιτυγχάνεται κατεβάζοντας την γραμμή SDA χαμηλά ή διατηρώντας τη γραμμή σταθερή. Αν δηλαδή η γραμμή κρατηθεί σταθερή στην κατάσταση 1 από τον slave, τότε θα σταλθεί από αυτόν ένα **negative acknowledge** (NAK) bit, ενώ αν η γραμμή αλλάξει στην κατάσταση 0, τότε θα σταλθεί ένα **acknowledge** (ACK) bit. Σε κάθε γραμμή υπάρχει μόνο ένας slave και συνεπώς ο master επικοινωνεί μόνο με έναν slave κάθε φορά.

Έπειτα, ο master ή ο slave, ανάλογα με την κατάσταση read/write που έχει δοθεί προηγουμένως, στέλνει ή διαβάζει τα δεδομένα. 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. Η διαδικασία λήψης μετρήσεων μέσω του I²C πρωτοκόλλου στον BMP085 για παράδειγμα, φαίνεται στο παρακάτω λογικό πίνακα: Στην συνέχεια δίνονται οι

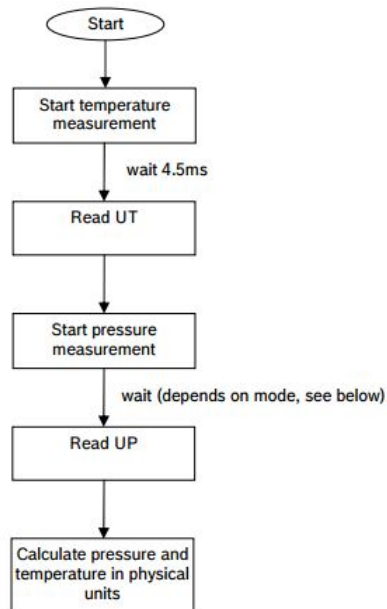


Figure 2.20: Λογικός Πίνακας διαδικασίας λήψης δεδομένων από τον BMP085

χάρτες μνήμης (memory maps) για τους αισθητήρες BMP085 και BME280.

Register Name	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset state
hum_lsb	0xFE	hum_lsb<7:0>								0x00
hum_msb	0xFD	hum_msb<7:0>								0x80
temp_xlsb	0xFC	temp_xlsb<7:4>				0	0	0	0	0x00
temp_lsb	0xFB	temp_lsb<7:0>								0x00
temp_msb	0xFA	temp_msb<7:0>								0x80
press_xlsb	0xF9	press_xlsb<7:4>				0	0	0	0	0x00
press_lsb	0xF8	press_lsb<7:0>								0x00
press_msb	0xF7	press_msb<7:0>								0x80
config	0xF5	t_sb[2:0]		filter[2:0]			spi3w_en[0]			0x00
ctrl_meas	0xF4	osrs_t[2:0]			osrs_p[2:0]			mode[1:0]		0x00
status	0xF3	measuring[0]				im_update[0]			0x00	
ctrl_hum	0xF2	osrs_h[2:0]								0x00
calib26..calib41	0xE1..0xF0	calibration data								individual
reset	0xE0	reset[7:0]								0x00
id	0xD0	chip_id[7:0]								0x60
calib00..calib25	0x88..0xA1	calibration data								individual

Registers:	Reserved registers do not change	Calibration data	Control registers	Data registers	Status registers	Chip ID	Reset
Type:	do not change	read only	read / write	read only	read only	read only	write only

Figure 2.21: Αναλυτικός πίνακας του memory map για τον αισθητήρα bme 280

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

Figure 2.22: Αναλυτικός πίνακας των registers για τον αισθητήρα bmp085

2.5 TCA9548A 1 to 8 I²C Multiplexer Break-out

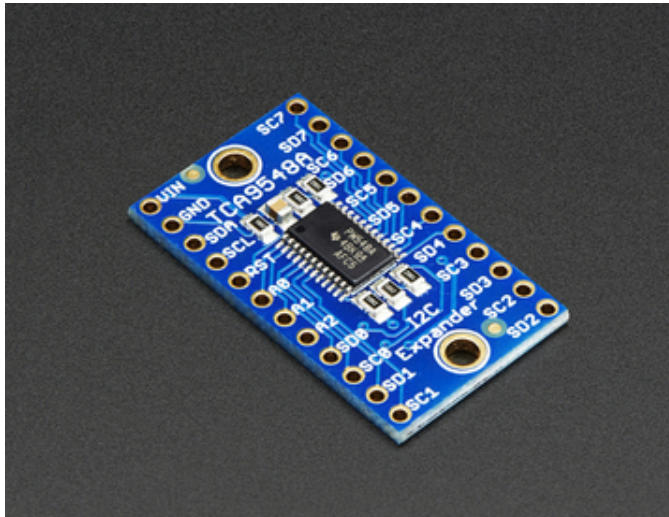


Figure 2.23: Πολυπλέκτης TCA9548A

Το εξάρτημα αυτό δεν αποτελεί έναν αισθητήρα. Ο πολυπλέκτης 8 καναλιών της Adafruit, μας δίνει την δυνατότητα να έχουμε πολλούς ίδιους αισθητήρες που χρησιμοποιούν I²C πρωτόκολλο συνδεδεμένους στο ίδιο Arduino. Οι αισθητήρες που λειτουργούν κάτω από αυτό το πρωτόκολλο, έχουν μία καθορισμένη διεύθυνση (Address) από τον κατασκευαστή τους. Αυτό σημαίνει ότι στην πραγματικότητα το Arduino δεν θα μπορούσε να διαβάσει τιμές από 4 ίδιους αισθητήρες που λειτουργούν κάτω από αυτό το πρωτόκολλο. Θα μπορούσαμε δηλαδή να χρησιμοποιούμε μόνο έναν αισθητήρα / Arduino κάθε φορά.

Ο πολυπλέκτης αυτός, δίνει την δυνατότητα στον χρήστη να χρησιμοποιήσει παραπάνω από μία διεύθυνση. Όλοι οι αισθητήρες συνδέονται με τον πολυπλέκτη και εκείνος τους προσδίδει μία άλλη διεύθυνση. Αυτό έχει σαν τελικό αποτέλεσμα, να μπορεί ο χρήστης να χρησιμοποιήσει πολλούς ίδιους αισθητήρες (όπως για παράδειγμα στην συγκεκριμένη εργασία για τους BME280). Ο πολυπλέκτης χρησιμοποιεί ο ίδιος I²C πρωτόκολλο υπό την διεύθυνση 0x70 και μπορεί να προσδώσει μία διεύθυνση από 0x71 έως και 0x77.

2.6 Συγκρίσεις Τιμών Αισθητήρων BME, BMP, DHT

Στο σημείο αυτό θα γίνει σύγκριση των τιμών που λήφθηκαν από τους αισθητήρες BME, BMP και DHT. Οι αισθητήρες αφέθηκαν 11 μέρες στο εργαστήριο και οι τιμές που λήφθηκαν ήταν για κάθε χρονική στιγμή.

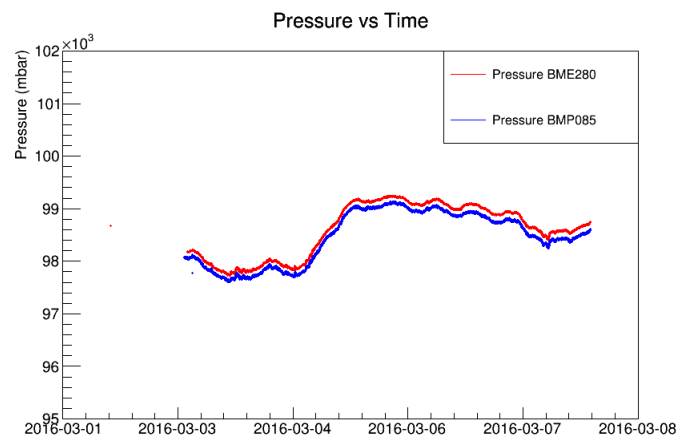


Figure 2.24: Διαγράμμα Πίεσης των αισθητήρων BME και BMP

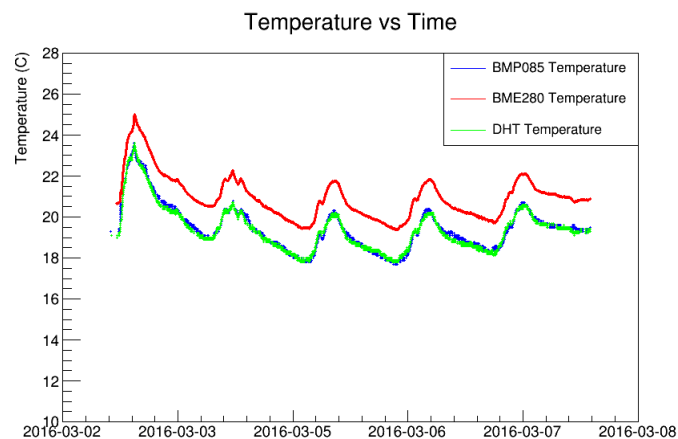


Figure 2.25: Διαγράμμα Θερμοκρασίας των αισθητήρων BME, BMP και DHT

Τελικά αποφασίστηκε να μην χρησιμοποιηθεί στην τελική εργασία ο αισθητήρας BMP καθώς οι τιμές που λήφθηκαν είχαν πολύ μικρή απόκλιση με τον BME. Σε συνδυασμό λοιπόν με την πολύπλοκη συνδεσμολογία του BMP (χρήση multiplexer αλλά και Level Shifter), πράγμα που θα ανέβαζε το συνολικό κόστος

της εργασίας, κρίθηκε σκόπιμο να μην χρησιμοποιηθεί ο BMP στο τελικό PCB της εργασίας.

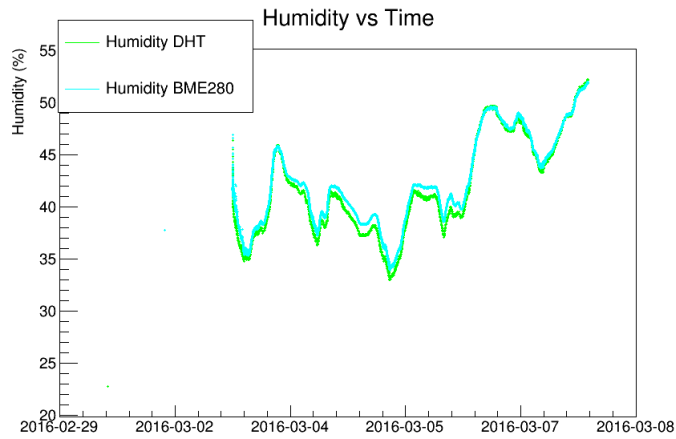


Figure 2.26: Διαγράμμα Πίεσης των αισθητήρων BME και DHT

Κεφάλαιο 3

OPC Server

Στην ενότητα αυτή παρουσιάζεται ο Arduino OPC (Open Platform Communication) Server. Θα γίνει δηλαδή αρχικά μία περιγραφή ενός συστήματος που χρησιμοποιεί το OPC πρωτόκολλο για επικοινωνία και στη συνέχεια θα περιγραφεί αναλυτικά ο OPC Server για Arduino καθώς και η μεθοδολογία εγκατάστασής και χρήσης του.

3.1 Εισαγωγή

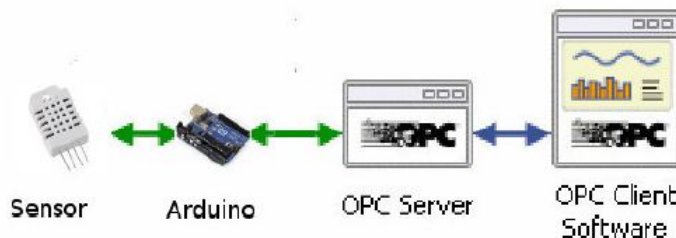


Figure 3.1: Διάγραμμα Επικοινωνίας με χρήση του πρωτοκόλλου OPC

Το **OPC**(Open Platform Communications) πρωτόκολλο πρόκειται για μία σειρά προδιαγραφών που επιτρέπουν την επικοινωνία μεταξύ λογισμικού(όπως για παράδειγμα το περιβάλλον των Windows) και εξωτερικών συσκευών. Η επικοινωνία πραγματοποιείται σε ζεύγη **διακομιστή (Server)** και **πελάτη (Client)**. Οι παραπάνω προδιαγραφές δημιουργήθηκαν για πρώτη φορά το 1996 με την ονομασία **OLE** (Object Linking and Embedding). Το OLE είναι ένα πρωτόκολλο που δημιουργήθηκε από την Microsoft για τη σύνδεση και ενσωμάτωση αρχείων και αντικειμένων και αυτός είναι και ο λόγος που

χρησιμοποιείται κυρίως από προγράμματα Windows. Το OLE είχε σκοπό την αλληλεπίδραση με το λογισμικό **SCADA** (Supervisory Logic Controller) με ένα PLC (Programmable Logic Controller), ενός δηλαδή μικροπολογιστή που ελέγχει μία συσκευή. Πλέον το πρωτόκολλο επικοινωνίας έχει εξελιχθεί τόσο, έτσι ώστε να χρησιμοποιείται ευρέως από συσκευές στον χώρο του αυτομάτου ελέγχου. Η πιο γνωστή μορφή του OPC είναι το **OPC Data Access (OPC DA)**, το οποίο χρησιμοποιείται για να διαβάζει και να γράφει δεδομένα σε πραγματικό χρόνο.

Το OPC είχε σχεδιασθεί για να παρέχει μία "γέφυρα" (bridge) σε εφαρμογές και hardware που είχαν λογισμικό Windows. Ο OPC Server για μία εξωτερική συσκευή προσφέρει σε έναν ή και περισσότερους servers χρησιμοποιώντας έναν ή και περισσότερους clients πρόσβαση στα δεδομένα της συσκευής αυτής. Ένας client έχει την δυνατότητα 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 μία μορφή, την οποία η εξωτερική συσκευή μπορεί να κατανοήσει. Με τον τρόπο αυτό, καθίσταται δυνατή η επικοινωνία του πελάτη και της συσκευής, με τον πελάτη να στέλνει εντολές μέσω του διακομιστή ή με τον διακομιστή να διαβάζει δεδομένα από τη συσκευή.

Το μεγαλύτερο πλεονέκτημα που έχει το OPC πρωτόκολλο είναι η δυνατότητα που δίνεται στον χρήστη να φτιάξει με πολύ απλό τρόπο έναν διακομιστή (Server) ή έναν πελάτη (Client) έχοντας στην διάθεσή του μόνο τα απαραίτητα στοιχεία για την σύνδεση διακομιστή και πελάτη. Οι προδιαγραφές είναι κοινές για όλους τους πελάτες και τους διακομιστές. Αυτό επιτρέπει σε οποιοδήποτε πελάτη να συνδεθεί με οποιοδήποτε διακομιστή. Το κόστος για τους κατασκευαστές συσκευών διατηρείται χαμηλό, καθώς απαιτείται μόνο ένας διακομιστής και έτσι η συσκευή μπορεί να επικοινωνήσει με οποιοδήποτε λογισμικό που παρέχει σύνδεση με OPC. Οι προμηθευτές λογισμικού συμπεριλαμβάνουν έναν πελάτη στο πρόγραμμά τους, δίνοντάς του τη δυνατότητα να συνδεθεί με οποιαδήποτε συσκευή διαθέτει OPC διακομιστή.

3.2 Arduino OPC Server

Για το Arduino, υπάρχει ένας διακομιστής, ο οποίος δημιουργήθηκε από τον **Ildefonso Martinez Marchena** και διατίθεται δωρεάν στο διαδίκτυο στην ιστοσελίδα: <http://www.st4makers.com/>. Ο διακομιστής αυτός είναι πλήρως συγχρονισμένος με τις προδιαγραφές OPC και μπορεί να συνδεθεί με οποιοδήποτε πρόγραμμα SCADA δίνοντας τη δυνατότητα στον χρήστη να χρησιμοποιεί το Arduino με μία πληθώρα προγραμμάτων αυτόματου ελέγχου.

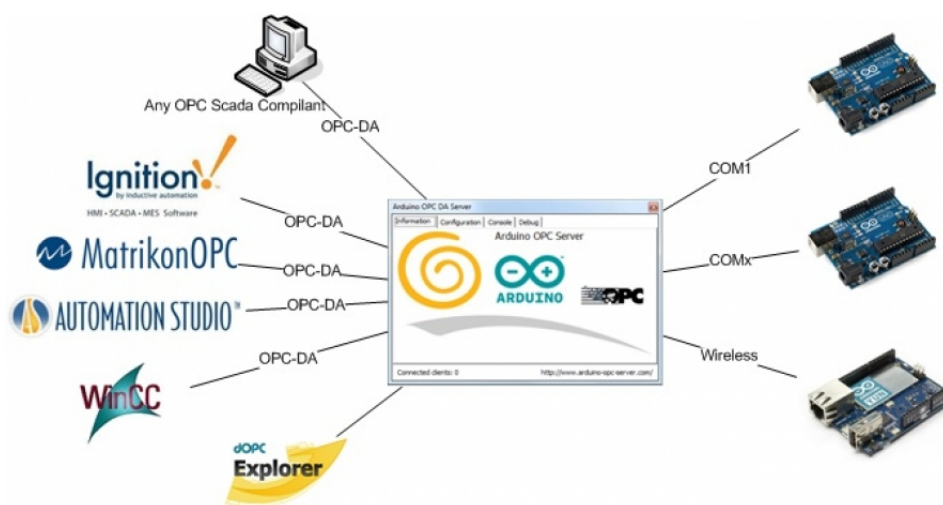


Figure 3.2: Arduino OPC Server

Μερικά χαρακτηριστικά του OPC Server αυτού είναι:

- Δυνατότητα ταυτόχρονης σύνδεσης με πολλές πλακέτες Arduino (YUN, UNO MEGA, Arduino Shields)
- Ταυτόχρονη χρήση σειριακής, Ethernet 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.

Ο OPC Server για Arduino πρόκειται για έναν εύκολο και δωρεάν τρόπο για χρήση του OPC server που επιτρέπει στο χρήστη να συνδέσει το Arduino με οποιαδήποτε OPC SCADA συμβατή εξωτερική συσκευή. Ο χρήστης μπορεί να στέλνει και να ανακτά δεδομένα από και προς SCADA HMI λογισμικό για να φτιάξει δωρεάν τεχνολογικές εφαρμογές. Για την χρήση του OPC Server για Arduino, ο χρήστης πρέπει απλώς να κατεβάσει και να προσθέσει στα Arduino Sketches την βιβλιοθήκη **OPC.h** η οποία υπάρχει διαθέσιμη δωρεάν στο διαδίκτυο. Με αυτό τον τρόπο ο χρήστης μπορεί να προσθέσει αντικείμενα στον OPC Server και αυτό του δίνει την δυνατότητα να διαβάζει και να στέλνει δεδομένα μεταξύ του Arduino και του προγράμματος SCADA (εδώ WinCC OA).

3.3 Εγκατάσταση και Χρήση Arduino OPC Server

Στην εργασία αυτή η χρήση του Arduino OPC Server είναι απαραίτητη, καθώς χωρίς αυτό η επικοινωνία Arduino και SCADA (εδώ το WinCC OA) θα ήταν αδύνατη.

Αρχικά, έγινε λήψη ενός zip αρχείου από την ιστοσελίδα που αναφέρθηκε προηγουμένως. Είναι σημαντικό ο χρήστης να κατεβάσει την πιο ενημερωμένη έκδοση για την αποφυγή τυχόν προβλημάτων. Πιο συγκεκριμένα, η έκδοση του Arduino OPC Server για την εργασία αυτή είναι η έκδοση 1.6. Μετά την λήψη του αρχείου, ο χρήστης πρέπει να κάνει εξαγωγή των συμπιεσμένων αρχείων σε έναν φάκελο. Ο φάκελος αυτός περιέχει διάφορα αρχεία. Μεταξύ αυτών και ένας φάκελος που περιέχει τις βιβλιοθήκες που απαιτούνται από το Arduino IDE. Ο χρήστης πρέπει να μεταφέρει τον φάκελο με τις βιβλιοθήκες στον φάκελο libraries του Arduino. Τα άλλα αρχεία στον φάκελο είναι τα παρακάτω:

- **ArduinoOPCServer** (executable)
- **changelog**
- **readme**

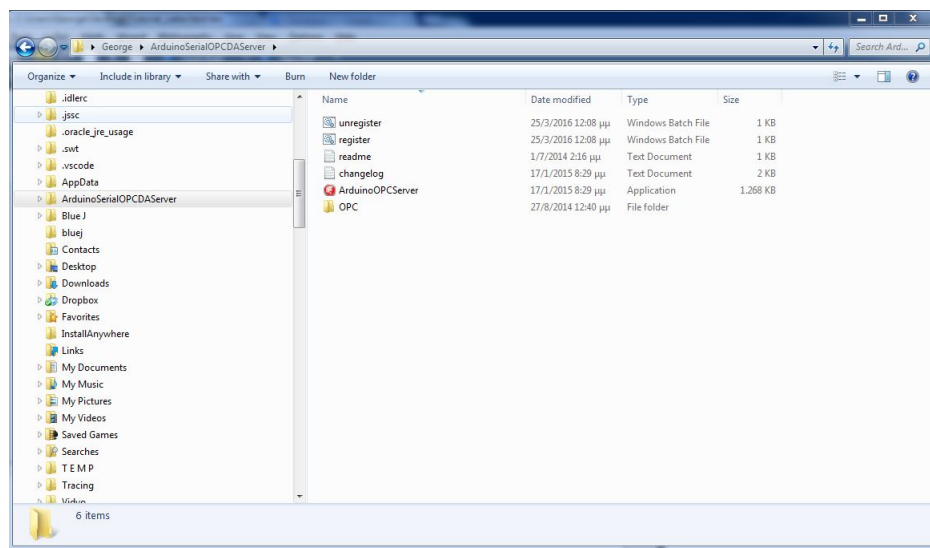


Figure 3.3: Φάκελος Arduino OPC Server

Κάνοντας δεξί κλικ στο ArduinoOPCServer.exe και κάνοντας κλικ στην επιλογή "Run as Administrator" γίνεται η εκκίνηση του server όπως φαίνεται

και παρακάτω στην εικόνα 3.4. Κατά την εκκίνηση δημιουργούνται δύο νέα αρχεία: "register.bat" και "unregister.bat". Ο χρήστης απαιτείται να εφαρμόσει τις απαραίτητες ρυθμίσεις στον server για την επικοινωνία με το Arduino.

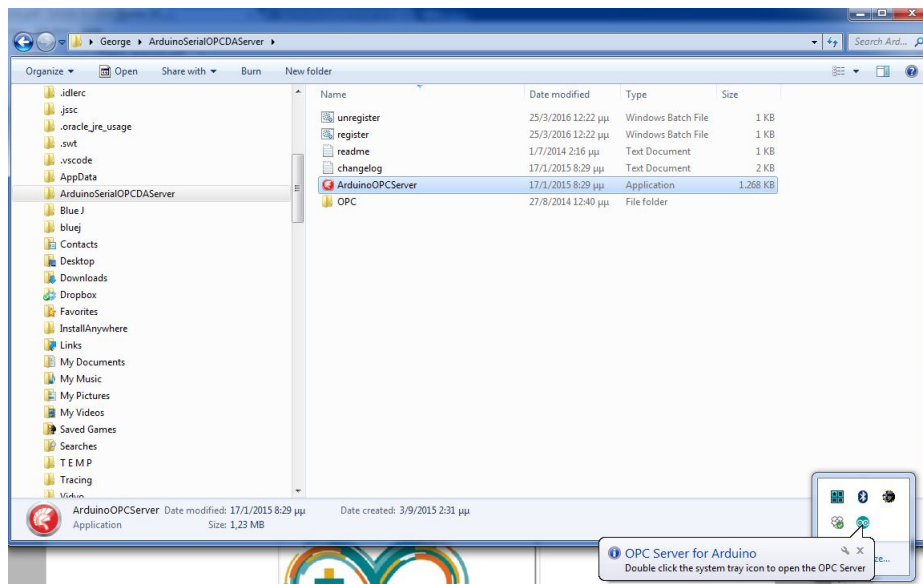


Figure 3.4: Εκκίνηση Arduino OPC Server

Κάνοντας διπλό κλικ στο εικονίδιο του server κάτω δεξιά στην οθόνη αναδύεται ένα νέο παράθυρο το οποίο αποτελεί κεντρικό παράθυρο του server.

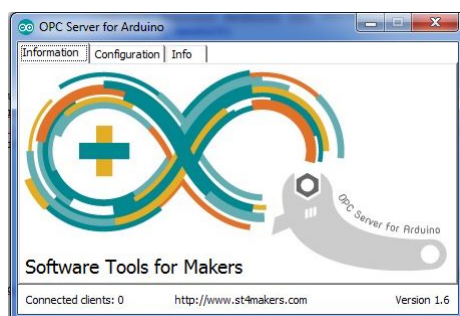


Figure 3.5: Arduino OPC Server

Στη συνέχεια ο χρήστης κάνει κλικ στο configuration tab στην συνέχεια ο χρήστης κάνει κλικ στην επιλογή "Arduino Yun". Εκεί δημιουργείται το ArduinoYun0 και στη συνέχεια εισάγεται η IP Address του Arduino. Η IP Address που δηλώνεται θα πρέπει να είναι ίδια με αυτή στο Arduino IDE

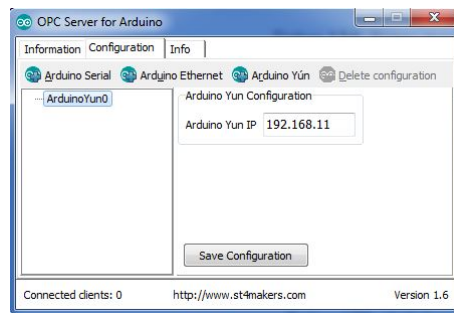


Figure 3.6: Configuration Tab, Arduino OPC Server

από το οποίο φορτώνεται το επιλεγμένο sketch. Στη συνέχεια ο χρήστης κάνει κλικ στο κουμπί "Save Configuration". Έπειτα στον φάκελο που είναι όλα τα αρχεία για τον server, ο χρήστης κάνει δεξί κλικ στο αρχείο register.bat και πατάει την επιλογή "Run as Administrator". Με αυτή την επιλογή ολοκληρώνεται η διαδικασία εγκατάστασης του server. Αναφέρεται ότι το Arduino YUN λαμβάνει την IP Address του δικτύου στο οποίο είναι συνδεδεμένο. Έτσι φορτώνοντας ένα sketch στο Arduino που κάνει χρήση των βιβλιοθηκών του Arduino OPC Server (OPC.h), το Arduino μπορεί να επικοινωνεί με τον Server.

Κεφάλαιο 4

Πλακέτα Τυπωμένου Κυκλώματος

Στο κεφάλαιο αυτό, παρουσιάζεται η σχεδίαση της πλακέτας τυπωμένου κυκλώματος (Personal Circuit Board) που χρησιμοποιήθηκε στην εργασία αυτή. Θα γίνει λεπτομερής παρουσίαση της διαδικασίας σχεδίασης των δύο πλακετών που χρησιμοποιήθηκαν. Οι 2 πλακέτες που χρησιμοποιήθηκαν σχεδιάστηκαν με την βοήθεια του Eagle, ενός προγράμματος της CadSoft ειδικά σχεδιασμένο για σχεδίαση και κατασκευή πλακετών τυπωμένου κυκλώματος (δηλαδή PCB Design).

4.1 Eagle Cadsoft

4.1.1 Εισαγωγή

Το EAGLE (Easy Applicable Graphical Layout Editor) είναι ένα ισχυρό λογισμικό για σχεδίαση πλακετών τυπωμένου κυκλώματος το οποίο απευθύνεται επαγγελματίες μηχανικούς αλλά και σε μαθητές καθώς είναι πολύ εύκολο στη χρήση του, αλλά ταυτόχρονα πολύ εξελιγμένο δίνοντας την δυνατότητα στο χρήστη για πολύπλοκους σχεδιασμούς.

Η απλότητα του λογισμικού αυτού, κάνει το λογισμικό αυτό πολύ εύχρηστο ακόμα και σε κόσμο νέο στο χώρο της σχεδίασης πλακετών τυπωμένου κυκλώματος. Η ελευθερία των σχεδιαστικών μέσων που παρέχει το EAGLE όπως οι εκτενής και ελεύθερες για χρήση βιβλιοθήκες με εξαρτήματα ηλεκτρονικών, ευκολύνουν ακόμα περισσότερο την άμεση εκμάθηση και χρήση του συγκεκριμένου λογισμικού.

Το λογισμικό είναι εύκολο στην εκμάθηση και στη χρήση του, καθώς παρέχει στο χρήστη τρία "πακέτα εργασιών" με απaráλλακτη διεπαφή χρήστη (User

Interface). Επίσης υπάρχουν εκατοντάδες βίντεο στο διαδίκτυο, μέσα από τα οποία μπορεί κανείς να μάθει πολύ άμεσα την χρήση του λογισμικού (tutorials). Τέλος το EAGLE παρέχει δωρεάν το εγχειρίδιο χρήσης του λογισμικού για όλους τους χρήστες. Το EAGLE είναι συμβατό με όλα τα λειτουργικά συστήματα (Windows, Linux, Mac) και είναι διαθέσιμο για 32 και 64 bit εκδόσεις. Επίσης περιέχει διάφορες επεκτάσεις του λογισμικού για πιο πολύπλοκους σχεδιασμούς καθώς και χιλιάδες βιβλιοθήκες έτοιμες για χρήση. Όπως θα δούμε και παρακάτω πιο αναλυτικά, ότι τα τρία πακέτα εργασιών που προσφέρει το EAGLE είναι τα εξής:

- **Schematic Editor**
- **Layout Editor**
- **Autorouter**

Αρχικά όμως πρέπει ο χρήστης να δημιουργήσει μία νέα εργασία (project) κατά την εκκίνηση του EAGLE. Η αρχική σελίδα που θα εμφανιστεί στον χρήστη είναι η εξής:

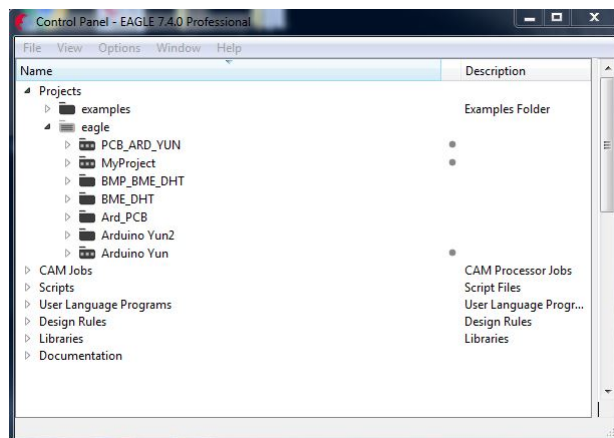


Figure 4.1: Αρχική σελίδα EAGLE

4.1.2 Schematic Editor

Επιλέγοντας το project που έχει δημιουργήσει ο χρήστης, κάνει κλικ στην επιλογή "New" και στην συνέχεια κλικ στην επιλογή "Schematic" .

Εδώ ο χρήστης μπορεί να δημιουργήσει το schematic του κυκλώματος. Μπορεί δηλαδή να τοποθετήσει από τις βιβλιοθήκες που διατίθενται τα διάφορα

ηλεκτρονικά εξαρτήματα που θέλει να χρησιμοποιήσει και να κάνει τις αντίστοιχες συνδέσεις μεταξύ τους. Το παράθυρο που αναδύεται στον χρήστη είναι το User Interface του schematic editor και είναι το παρακάτω:

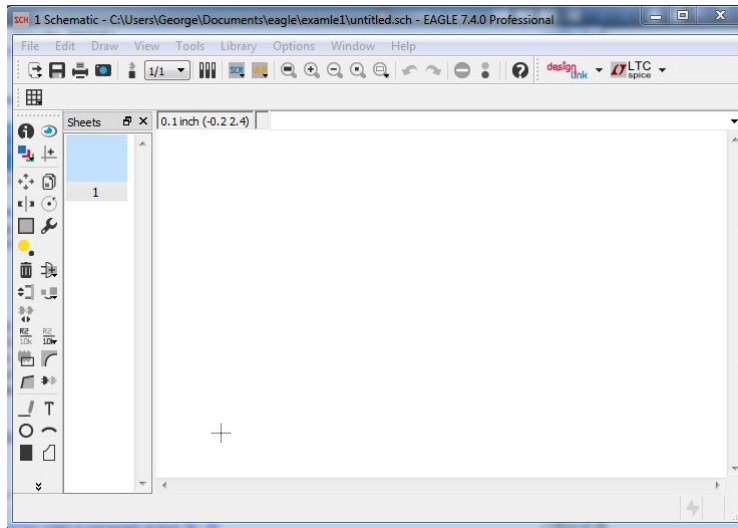


Figure 4.2: Schematic Editor στο EAGLE

Στην πραγματικότητα ο Schematic Editor επιτρέπει στον χρήστη να δημιουργήσει μία παρουσίαση του κυκλώματος που θέλει να σχεδιάσει, εύκολη στην ανάγνωση. Ο στόχος του schematic είναι να παρέχει έγγραφο για το σχέδιο, επιτρέποντας έτσι την δυνατότητα σε οποιονδήποτε να καταλάβει το σκοπό του σχεδίου. Το EAGLE παρέχεται στο χρήστη με χιλιάδες εξαρτήματα για την απλοποίηση του κάθε σχεδίου. Επίσης το EAGLE παρέχει την δυνατότητα ERC (Electrical Rule Check: Ηλεκτρικός έλεγχος σχεδίου). Άλλα χαρακτηριστικά του schematic είναι:

- Μέχρι και 999 φύλλα σε κάθε Schematic
- Προβολή εικόνων για κάθε φύλλο
- Απευθείας σύνδεση μεταξύ Schematic και Board.
- Αυτόματη παραγωγή του board
- Electrical Rule Check

Ο βασικός τρόπος με τον οποίο μπορεί κανείς να σχεδιάσει το schematic ξεκινά με την τοποθέτηση των εξαρτημάτων στο φύλλο εργασίας. Στη συνέχεια χρησιμοποιώντας το κουμπί "Nets" ή "Buses" μπορεί ο χρήστης να κάνει τις

απαραίτητες συνδέσεις μεταξύ των εξαρτημάτων. Τέλος ο χρήστης μπορεί να μετακινεί και να επεξεργάζεται κάθε εξάρτημα στο φύλλο εργασίας.

Πιο συγκεκριμένα ένα παράδειγμα ενός Schematic είναι το παρακάτω:

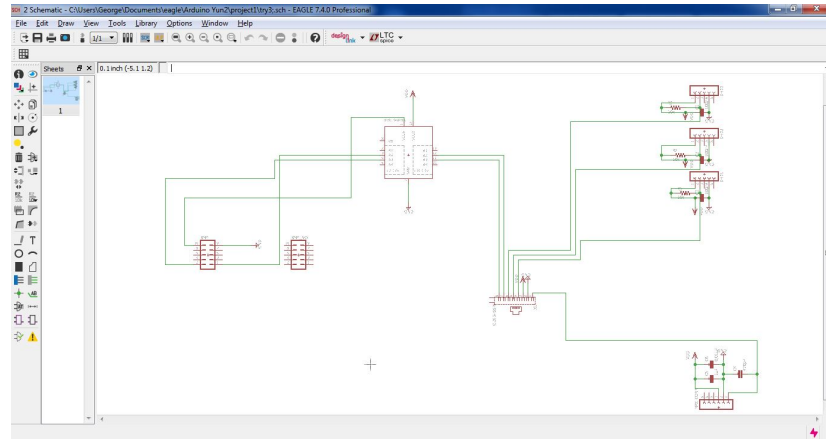


Figure 4.3: Παράδειγμα Schematic Editor

4.1.3 Layout Editor

Για την δημιουργία του layout editor μπορεί ο χρήστης να την εκκινήσει με δύο διαδικασίες. Είτε επιλέγοντας το project που έχει δημιουργήσει ο χρήστης, κάνει κλικ στην επιλογή "New" και στην συνέχεια κλικ στην επιλογή "Board", είτε όπως φαίνεται και παρακάτω στην εικόνα, κάνει κλικ στο κουμπί του φύλλου εργασίας του schematic "Generate/Switch to Board".

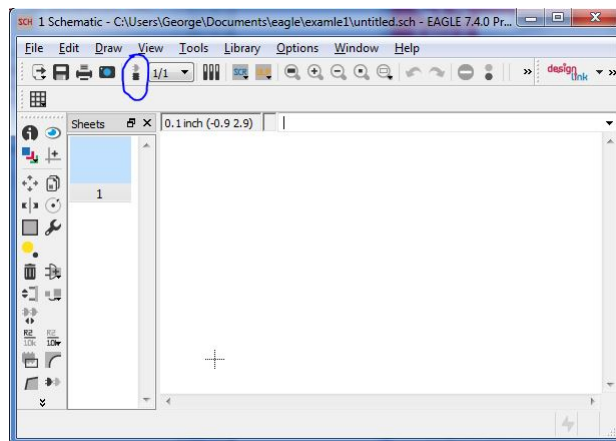


Figure 4.4: Μεταφορά στο Layout Editor

Ο Layout Editor είναι πρακτικά η διαδικασία μέσα από την οποία παράγεται το τελικό PCB Design. Η πλακέτα που παράγεται σε αυτό το φύλλο εργασίας, αντιπροσωπεύει το πραγματικό PCB που θα κατασκευαστεί. Ο χρήστης έχει τη δυνατότητα να εντοπίζει τα εξαρτήματα που είχε ήδη στο schematic, αφού το EAGLE επιτρέπει το forward/back annotation, μία διαδικασία που επιτρέπει την άμεση (Online) επικοινωνία μεταξύ schematic και layout editor. Το φύλλο εργασίας του layout editor φαίνεται στην εικόνα 4.5

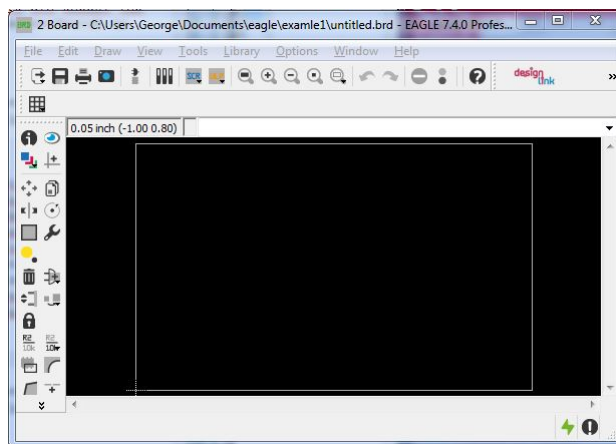


Figure 4.5: Layout Editor

Οι δυνατότητες που έχει ο χρήστης για την επεξεργασία της πλακέτας και των ηλεκτρονικών εξαρτημάτων είναι αρκετές. Μερικές από αυτές είναι:

- Δυνατότητα επιλογής διαστάσεων πλακέτας
- Υποστήριξη Blind και Buried vias
- Περιστροφή εξαρτημάτων (ακόμα και 0,1 μοίρες)
- Κλείδωμα εξαρτημάτων για να μην μπορούν να μετακινηθούν
- Τοποθέτηση κειμένου με οποιοδήποτε προσανατολισμό
- Οι τροχιές των γραμμών μπορούν να είναι ακόμα και καμπύλες.
- Design Rule Check, δηλαδή έλεγχος για τυχόν προβλήματα που μπορούν να παρουσιάζονται στην πλακέτα.
- Παραγωγή δεδομένων που απαιτούνται για την κατασκευή της σχεδιαζόμενης πλακέτας (drills, μέγεθη γραμμών κα) μέσω του CAM Processor.

Πιο συγκεκριμένα ένα παράδειγμα ενός Board είναι το παρακάτω:

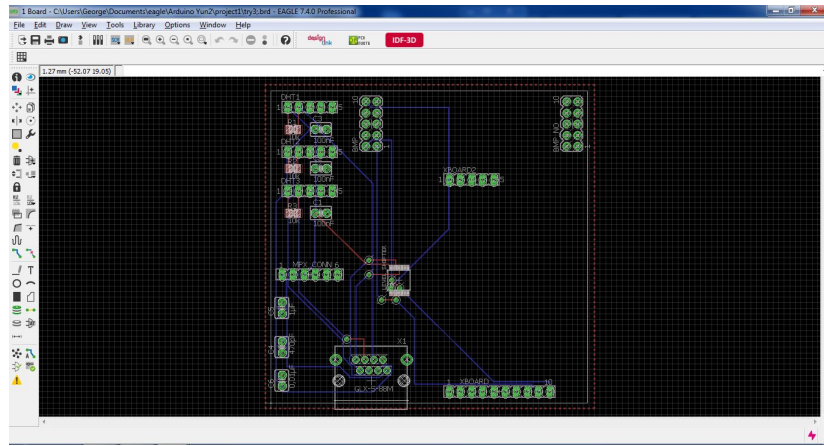


Figure 4.6: Παράδειγμα Layout Editor

4.1.4 Autorouter

Η λειτουργία αυτή που παρέχει το EAGLE είναι ένα χρήσιμο εργαλείο για την τοποθέτηση ιχνών χαλκού (copper traces) μεταξύ των εξαρτημάτων. Ο autorouter μπορεί να καθοδηγηθεί μεταβάλλοντας τους παράγοντες κόστους έτσι ώστε η πορεία που έχουν τα ίχνη χαλκού (δηλαδή οι γραμμές επικοινωνίας του pcb) να είναι χρήσιμα και λειτουργικά. Ο χρήστης έχει τη δυνατότητα να επιλέξει εκείνος ποιο από τα προτεινόμενα routes του έχει δημιουργήσει η λειτουργία αυτή, κρίνοντας εκείνος ποιο είναι το πιο αποδοτικό και οικονομικό. Ορισμένα χαρακτηριστικά του Autorouter είναι:

- Routing Grid (πλέγμα οδήγησης) στα 0,02mm
- Αλγόριθμος RipupAndRetry, που δίνει στο χρήστη τη δυνατότητα να σβήσει το προηγούμενο σχεδιασμένο route και να δημιουργήσει ένα νέο
- Χρήση των σχεδιαστικών κανόνων που χρησιμοποιήθηκαν στο Layout Editor
- Δυνατότητα επιλογής όχι μόνο Auto αλλά και Manual Route
- Δυνατότητα κατασκευής έως και πλακέτα 16 στρωμάτων με ίχνη χαλκού (16 layered board).

Στο Layout Editor οι συνδέσεις που έχουν προέλθει από το Schematic εμφανίζονται με air-wires. Εκεί λοιπόν έγκειται η ανάγκη για χρήση του Autorouter. Η λειτουργία αυτή πραγματοποιεί το "τύπωμα" των ενώσεων μεταξύ των εξαρτημάτων πάνω στην πλακέτα που σχεδιάζει ο χρήστης.

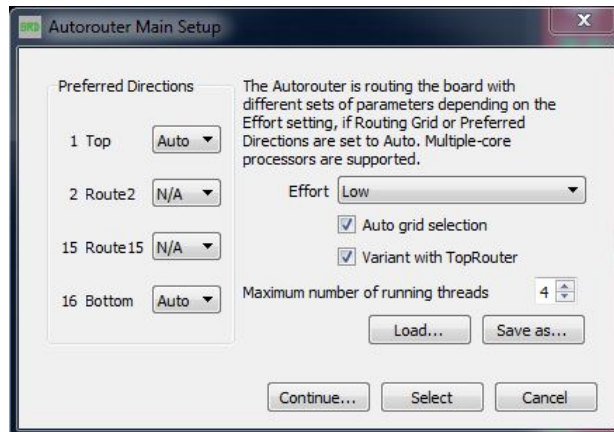


Figure 4.7: Παράθυρο Autorouter

4.2 Boards στην Εργασία

Στο σημείο αυτό θα παρουσιαστούν η πλακέτα πάνω στην οποία βρίσκονται οι αισθητήρες (4 πλακέτες αυτού του είδους) και το κεντρικό PCB που περιέχει το Arduino και πάνω στο οποίο γίνεται η σύνδεση με τους αισθητήρες.

4.2.1 BME και DHT Board

Στο σημείο αυτό, θα δούμε αναλυτικά την διαδικασία μέσα από την οποία δημιουργήθηκε η πλακέτα που περιέχει τους αισθητήρες. Σκοπός της συγκεκριμένης πλακέτας είναι να στέλνει τα δεδομένα που διαβάζονται από τους αισθητήρες BMP και DHT στο Arduino. Για την επικοινωνία της πλακέτας αυτής με το Arduino χρησιμοποιείται RJ45 connector (Ethernet καλώδιο). Κάθε Πλακέτα δηλαδή, έχει:

- 1 σειρά 7 female pins για τον BME
- 1 σειρά 5 female pins για τον DHT
- 1 θύρα RJ45 (GLX-S-88M) για την επικοινωνία κάθε πλακέτας με το Arduino

Παρακάτω δίνεται το Schematic για το Board 4.8:

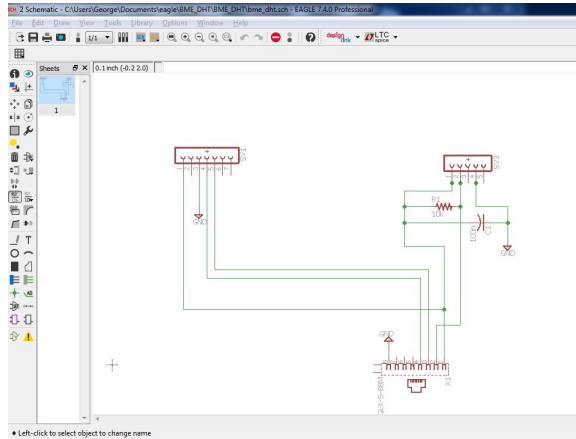


Figure 4.8: Schematic του BME-DHT Board

Και το Layout του Board 4.9:

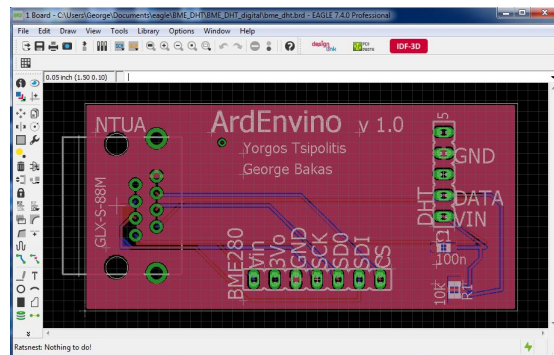


Figure 4.9: Layout του BME-DHT Board μετά την χρήση του Autorouter

Καθένα από αυτές τις πλακέτες, περιέχει δηλαδή 2 αισθητήρες. Όπως έχει αναφερθεί και προηγουμένως, ο BME παρέχει πληροφορίες για πίεση, θερμοκρασία και υγρασία χρησιμοποιώντας μόνο ένα καλώδιο σύνδεσης για τάση λειτουργίας, ένα για γείωση και δύο καλώδια σύνδεσης (λόγω I^2C πρωτοκόλλου). Ο DHT παρέχει πληροφορίες για υγρασία και θερμοκρασία χρησιμοποιώντας ένα καλώδιο σύνδεσης για τάση λειτουργίας, ένα για γείωση και ένα για αποστολή δεδομένων.

Η αντιστοιχία των καλωδίων αυτών στις υποδοχές της θύρας του RJ45 είναι :

- **1:** VCC
- **2:** Data DHT
- **3:** SD0 (BME Data)
- **4:** SCK (BME Clock)
- **5:** -
- **6:** -
- **7:** -
- **8:** GND (Ground)

Για την εξασφάλιση της λειτουργίας του συνολικού προϊόντος, κρίθηκε κατάλληλο εκτός από την κατασκευή ενός PCB που να χρησιμοποιεί I^2C πρωτόκολλο, να κατασκευαστεί και μία αντίστοιχη σειρά από PCB, τα οποία θα συνδέονται με το κεντρικό PCB που θα περιέχει το Arduino, τα οποία θα συνδέονται με SPI σύνδεση στα ψηφιακά (digital) Pins του Arduino. Στην περίπτωση αυτή, η αντιστοιχία των καλωδίων στις υποδοχές της θύρας RJ45 είναι:

- **1:** VCC
- **2:** Data DHT
- **3:** SCK
- **4:** SD0
- **5:** SDI
- **6:** CS
- **7:** -
- **8:** GND (Ground)

Παρακάτω δίνονται το schematic και το board layout για την προαναφερθείσα πλακέτα.

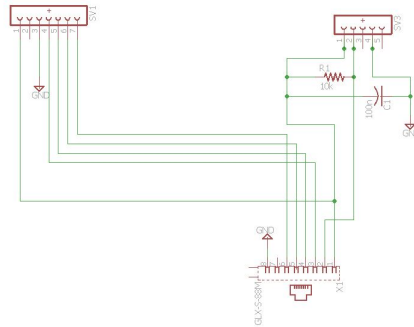


Figure 4.10: Schematic

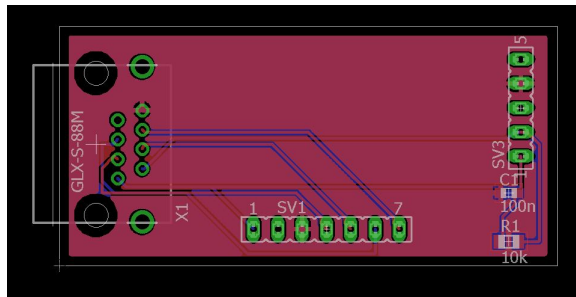


Figure 4.11: Layout του BME-DHT Digital Board μετά την χρήση του Autorouter

4.2.2 Main Arduino YUN/Mega Board

Στο σημείο αυτό, θα δούμε αναλυτικά την διαδικασία μέσα από την οποία δημιουργήθηκε η πλακέτα που περιέχει το Arduino και τις θύρες σύνδεσης με τα υπόλοιπα boards. Σκοπός της συγκεκριμένης πλακέτας είναι να δέχεται τα δεδομένα από κάθε πλακέτα που περιέχει αισθητήρες και να τα στέλνει με διαδικασίες που αναφέρθηκαν προηγουμένως στο WinCC OA για περαιτέρω επεξεργασία. Αυτό που κάνει την χρήση της πλακέτας δυναμική, είναι η δυνατότητα που δίνει στο χρήστη να επιλέγει ποιο Arduino θέλει εκείνος να χρησιμοποιήσει κάθε φορά. Ο χρήστης μπορεί να επιλέξει ανάμεσα στο Arduino YUN και στο MEGA. Με χρήση του YUN ο χρήστης μπορεί να στέλνει δεδομένα χρησιμοποιώντας wifi ενώ με χρήση του MEGA τα δεδομένα στέλνονται μέσω UTP καλωδίου. Όπως και πριν για να έχουμε σίγουρη επιτυχία στην εργασία, φτιάχτηκαν 2 πλακέτες. Η μία χρησιμοποιούσε I^2C πρωτόκολλο για όλες τις συνδέσεις των BME αισθητήρων. Η πλακέτα αυτή περιέχει:

- 2 σειρές male pins για την σύνδεση της πλακέτας με το Arduino.
- 5 θύρες RJ45 καλωδίων

- 2 σειρές 12 female pins για τον multiplexer
- 2 σειρές 18 male pins για σύνδεση με Arduino Mega
- 1 σειρά 3 male pins για σύνδεση του I^2C πρωτοκόλλου με το Arduino Mega

Παρακάτω δίνεται το Schematic του Board 4.12 και το layout του Arduino main board 4.13

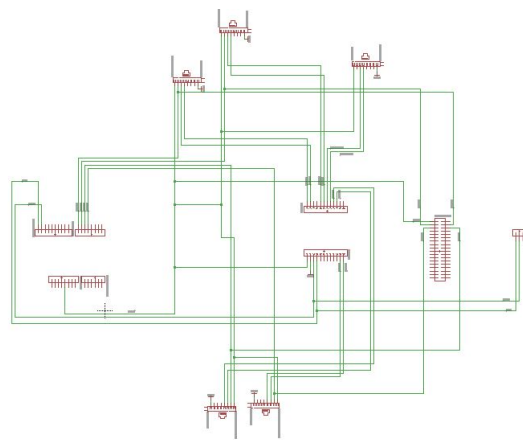


Figure 4.12: Schematic του Arduino main Board

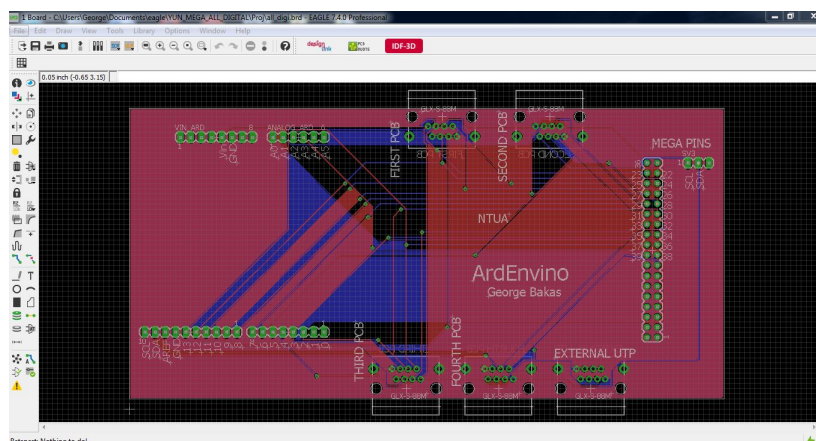


Figure 4.13: Layout του Arduino Main Board μετά την χρήση του Autorouter

Κάθε θύρα RJ45 της πλακέτας που χρησιμοποιεί τον TCA Multiplexer στέλνει τα δεδομένα είτε στο Arduino μέσω του multiplexer (BME) είτε απευθείας στο Arduino (DHT). Κάθε RJ45 θύρα έχει τις αντίστοιχες συνδέσεις με τις θύρες του multiplexer και του Arduino.

Για τους DHT τα δεδομένα στέλνονται στο Arduino YUN ως εξής:

- **RJ45 up left:** DHT1 στο digital pin 7
- **RJ45 up right:** DHT2 στο digital pin 6
- **RJ45 down left:** DHT3 στο digital pin 5
- **RJ45 down right:** DHT4 στο digital pin 4

Για τους DHT τα δεδομένα στέλνονται στο Arduino Mega ως εξής:

- **RJ45 up left:** DHT1 στο digital pin 22
- **RJ45 up right:** DHT2 στο digital pin 23
- **RJ45 down left:** DHT3 στο digital pin 24
- **RJ45 down right:** DHT4 στο digital pin 25

Αντίστοιχα δίνονται οι συνδέσεις των BME με τον Multiplexer που στέλνει τα δεδομένα στις θύρες SDA και SCL του Arduino (Όμοια για YUN και MEGA).

- **RJ45 up left:** BME SCK και SD0 στα pins SC7, SD7
- **RJ45 up right:** BME SCK και SD0 στα pins SC5, SD5
- **RJ45 down left:** BME SCK και SD0 στα pins SC3, SD3
- **RJ45 down right:** BME SCK και SD0 στα pins SC1, SD1

Στην αντίστοιχη άλλη πλακέτα που δημιουργήθηκε, όλες οι συνδέσεις των BME είναι ψηφιακές. Οι DHT συνδέονται με αναλογικό τρόπο στα Analog pins του Arduino YUN ενώ στο MEGA συνδέονται με τα ψηφιακά Pins. Επίσης στην πλακέτα αυτή τοποθετήθηκε στο κέντρο μία LCD οθόνη η οποία μπορεί να συνδεθεί με το αντίστοιχο Arduino μέσω I^2C και να παρουσιάζει διάφορες τιμές που παίρνουν οι αισθητήρες κατά τη λειτουργία τους.

Για τους DHT τα δεδομένα στέλνονται στο Arduino YUN ως εξής:

- **RJ45 up left:** DHT1 στο analog pin A0

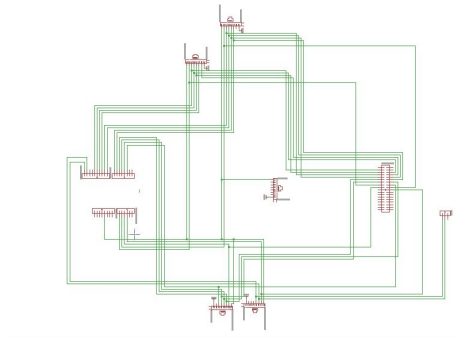


Figure 4.14: Schematic του Arduino Main Board (Digital)

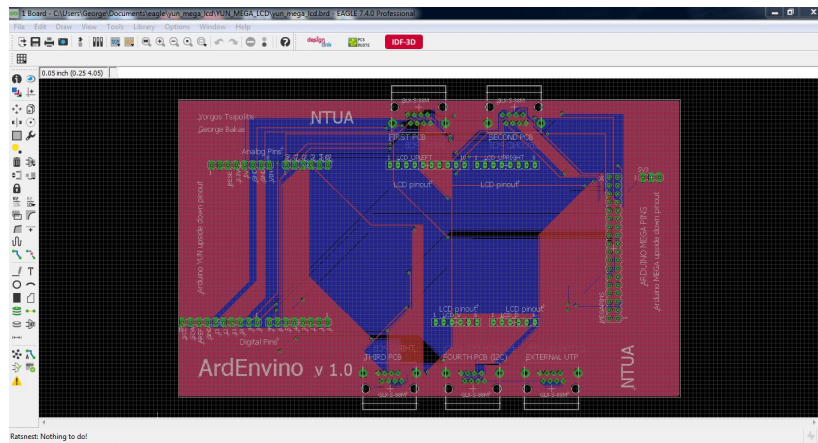


Figure 4.15: Layout του Arduino Main Board(Digital) μετά την χρήση του Autorouter

- **RJ45 up right:** DHT2 στο analog pin A1
- **RJ45 down left:** DHT3 στο analog pin A2
- **RJ45 down right:** DHT4 στο analog pin A3

Για τους DHT τα δεδομένα στέλνονται στο Arduino MEGA ως εξής:

- **RJ45 up left:** DHT1 στο digital pin 35
- **RJ45 up right:** DHT2 στο digital pin 36
- **RJ45 down left:** DHT3 στο digital pin 37
- **RJ45 down right:** DHT4 στο digital pin 38

Για τους BME τα δεδομένα στέλνονται στο Arduino YUN ως εξής:

- **RJ45 up left:** BME1 στα digital pins 13,12,11,10
- **RJ45 up right:** BME2 στα digital pins 9,8,7,6
- **RJ45 down left:** BME3 στα digital pins 5,4,3,2
- **RJ45 down right:** BME4 με χρήση I^2C πρωτοκόλλου

Για τους BME τα δεδομένα στέλνονται στο Arduino MEGA ως εξής:

- **RJ45 up left:** BME1 στα digital pins 23,24,25,26
- **RJ45 up right:** BME2 στα digital pins 27,28,29,30
- **RJ45 down left:** BME3 στα digital pins 31,32,33,34
- **RJ45 down right:** BME4 με χρήση I^2C πρωτοκόλλου

Όλα τα παραπάνω ήρθαν σε σύνδεση μεταξύ τους με χρήση καλωδίων UTP. Με αυτό τον τρόπο, πραγματοποιήθηκε επικοινωνία μεταξύ του main board και των τεσσάρων PCB με τους αισθητήρες.

Κεφάλαιο 5

Ανάπτυξη Συστήματος με WinCC OA

Στο κεφάλαιο αυτό, παρουσιάζεται το WinCC OA, ένα πρόγραμμα που χρησιμοποιείται για την κατασκευή συστημάτων αυτομάτου ελέγχου. Γίνεται αναφορά στο SLOW Control SYstem (SLOCSY) και στις νέες λειτουργίες που προστέθηκαν κατά την εργασία αυτή. Τέλος περιγράφεται με αναλυτικό τρόπο η διαδικασία σύνδεσης του Arduino με το WinCC OA και το Slow Control System.

5.1 WinCC OA

5.1.1 Εισαγωγή

Το WinCC OA ανήκει στην κατηγορία των προγραμμάτων **SCADA** (Supervising Control and Data Acquisition) τα οποία είναι συστήματα βιομηχανικού αυτόματου ελέγχου. Το κυριότερο χαρακτηριστικό αυτών των προγραμμάτων είναι ότι αποτελούνται από έναν 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 τοπικές αυτές μονάδες και συλλέγει δεδομένα. Τα δεδομένα αυτά μπορούν με τη σειρά τους να σταλούν σε άλλους σταθμούς μέσα σε ένα τοπικό δίκτυο LAN (Local Area Network) ή σε σημεία εκτός δικτύου μέσω κάποιου συστήματος τηλεπικοινωνίας.

Στον τομέα της Φυσικής Υψηλών Ενεργειών κρίνεται αναγκαίος ο τακτικός έλεγχος των ανιχνευτών. Για το λόγο αυτό, γίνεται χρήση των συστημάτων SCADA. Παρά την πληθώρα των συστημάτων SCADA, έχει επικρατήσει το WinCC OA για την κατασκευή συστημάτων ελέγχου. Το WinCC OA έχει φιλικό προς το χρήστη περιβάλλον, η παραμετροποίηση του είναι πολύ εύκολη

διαδικασία και δίνει στο χρήστη τη δυνατότητα εύκολης σύνδεσης εξωτερικών συσκευών με το λογισμικό OPC για συλλογή δεδομένων και περαιτέρω επεξεργασία τους. Τα παραπάνω είναι λίγοι από τους λόγους για τους οποίους το WinCC OA χρησιμοποιείται ευρέως έναντι των υπόλοιπων συστημάτων SCADA.

5.1.2 Βασικά Χαρακτηριστικά Του WinCC OA

Στη συνέχεια παρουσιάζονται ορισμένα από τα βασικά χαρακτηριστικά του WinCC OA.

- **Γραφικό Περιβάλλον (Graphics Editor (GED))**

Στο γραφικό περιβάλλον του WinCC OA, ο χρήστης έχει τη δυνατότητα να δημιουργήσει μία σειρά από παράθυρα (panels) τα οποία μπορεί να εμπλουτίσει με γραφικά στοιχεία που υπάρχουν έτοιμα στο WinCC OA, όπως κουμπιά, πλαίσια, πίνακες κλπ. Η λειτουργία του κάθε αντικειμένου πάνω στο παράθυρο (panel) καθώς και του παραθύρου καθορίζεται από κώδικα που γράφει ο χρήστης.

- **Κείμενα Ελέγχου (Control Scripts)**

Ο χρήστης έχει τη δυνατότητα να δημιουργήσει τα δικά του control scripts έτσι ώστε αυτά να εκτελούνται από κάποιο panel ή σαν ανεξάρτητη διαδικασία. Τα scripts γράφονται σε μία γλώσσα παρόμοιας δομής και λογικής με την C. Ωστόσο, η γλώσσα αυτή παρέχει πολύ περισσότερες λειτουργίες και συναρτήσεις με την απλή C. Με τον τρόπο αυτό, ο χρήστης μπορεί να κάνει το WinCC OA να συμπεριφέρεται με συγκεκριμένο τρόπο ανάλογα με τις παραμέτρους που εκείνος θέτει και έτσι το σύστημα να γίνεται πιο λειτουργικό σε σχέση με τις αρχικές λειτουργίες που είχε.

- **Βάση Δεδομένων (Database)**

Το WinCC OA διαθέτει μία εσωτερική βάση δεδομένων όπου μπορεί να αποθηκεύει δεδομένα που διαβάζει από τις εξωτερικές συσκευές με τις οποίες είναι συνδεδεμένο. Τα δεδομένα ομαδοποιούνται σε datapoints (DP) ενός ήδη προκαθορισμένου είδους Datapoint (Data Point Type, DPT). Το είδος των δεδομένων, περιγράφει τη δομή των δεδομένων ενός συγκεκριμένου είδους συσκευής. Τα datapoints περιέχουν τις τα δεδομένα και τις πληροφορίες για μία συγκεκριμένη συσκευή. Κάθε τιμή ή πληροφορία μίας συγκεκριμένης συσκευής αντιπροσωπεύεται από ένα datapoint Element (DPE) που περιέχεται σε ένα DP.

Τα παραπάνω θα γίνουν πιο κατανοητά μέσα από ένα παράδειγμα. Έστω ότι ο χρήστης χρησιμοποιεί ένα Arduino που έχει συνδεδεμένους έναν αισθητήρα

πίεσης και θερμοκρασίας και έναν αισθητήρα θερμοκρασίας και υγρασίας. Το Arduino μπορεί να αποτελεί ένα DataPoint Type. Έστω λοιπόν ότι έχει δημιουργηθεί ένα Data Point με όνομα Arduino0, είδους Arduino. Το datapoint αυτό θα ακολουθεί τη δομή του DPT Arduino. Αυτό σημαίνει ότι θα περιέχει πληροφορίες και τιμές για τους δύο αισθητήρες που συνδέονται με το Arduino. Αν ο χρήστης συνδέσει ένα άλλο Arduino με τους ίδιους αισθητήρες, θα πρέπει να δημιουργήσει ένα νέο DP ίδιου τύπου, ενώ αν συνδέσει ένα Arduino με διαφορετικούς αισθητήρες θα πρέπει να δημιουργήσει ένα νέο DataPoint Type. Τέλος κάθε αισθητήρας θα πρέπει να έχει κάποια Data-Point Elements τα οποία θα περιέχουν τις τιμές που στέλνουν οι αισθητήρες κάθε φορά. Για παράδειγμα ο ένας αισθητήρας θα έχει 2 DPE's για υγρασία και θερμοκρασία, ενώ ο άλλος αισθητήρας θα έχει 2 DPE's για πίεση και θερμοκρασία. Η παρακάτω αναπαράσταση αποτυπώνει γραφικά το παραπάνω παράδειγμα:

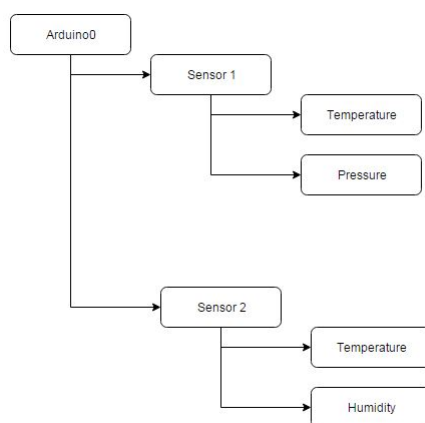


Figure 5.1: WinCC OA Data Structure

- **Σύνδεση με Εξωτερικές Συσκευές**

Όπως αναφέρθηκε και προηγουμένως, το WinCC OA δίνει τη δυνατότητα σύνδεσής του με εξωτερικές συσκευές. Έτσι ο χρήστης μπορεί να ζητήσει πληροφορίες από τις εξωτερικές συσκευές που χρησιμοποιεί, όπως για παράδειγμα σε αυτή την εργασία ο χρήστης ζητά να μάθει τις τιμές των περιβαλλοντικών συνθηκών που μετρούν οι αισθητήρες που είναι συνδεδεμένοι με το Arduino. Στη συνέχεια ο χρήστης είναι εκείνος ο οποίος επιλέγει τον τρόπο που τα δεδομένα αυτά θα διαχειριστούν, έχοντας τη δυνατότητα να τα αποθηκεύσει στη βάση δεδομένων του WinCC OA. Εφόσον οι εξωτερικές συσκευές το επιτρέπουν, ο χρήστης έχει τη δυνατότητα ακόμα και να χειρίζεται τις εξωτερικές

συσκευές μέσω του WinCC OA. Ο τρόπος επικοινωνίας ποικίλει ανάλογα με το πρωτόκολλο που χρησιμοποιεί κάθε εξωτερική συσκευή για την επικοινωνία της. Στην εργασία αυτή για παράδειγμα, το Arduino χρησιμοποιεί το πρωτόκολλο OPC για την επικοινωνία του με το WinCC OA. Το WinCC OA διαθέτει δηλαδή έναν OPC Client, ο οποίος συνδέεται με τον OPC Server και έτσι το WinCC OA έχει τη δυνατότητα αλληλεπίδρασης με το Arduino.

5.1.3 JCOP Framework

Στις περιπτώσεις που χρησιμοποιείται το WinCC OA είναι πολύ συνηθισμένη και η χρήση του **JCOP Framework** (Joint COntrols Project Framework). Το JCOP είναι μία σειρά από panels, control scripts και γενικότερα εργαλείων για το WinCC OA τα οποία δημιουργήθηκαν από κοινού από 4 ομάδες των πειραμάτων του LHC (Large Hadron Colider) με την επιπλέον συνεισφορά άλλων φυσικών και μηχανικών στο CERN. Σκοπός της δημιουργίας των εργαλείων αυτών είναι η παροχή κατευθυντήριων γραμμών για να δημιουργούνται ανεξάρτητα προγράμματα τα οποία να μπορούν ωστόσο όταν χρειαστεί να χρησιμοποιηθούν μαζί.

Το JCOP αποτελεί στην πραγματικότητα μία σειρά από εργαλεία τα οποία έχει στη διάθεσή του ο χρήστης και κάνουν τη σύνδεση των διαφόρων συσκευών με το WinCC OA εφικτή. Τα εργαλεία που παρέχονται από το JCOP χρησιμοποιούνται ως βασική δομή για τη δημιουργία projects.

5.2 Οδηγίες Σύνδεσης του Arduino με το WinCC OA

Στο υποκεφάλαιο αυτό, θα γίνει αναλυτική περιγραφή της διαδικασίας σύνδεσης του Arduino με το WinCC OA.

Αρχικά σε κάθε περίπτωση θα πρέπει ο χρήστης να "ανεβάσει" (upload) πρόγραμμα στο Arduino, τέτοιο που να δημιουργεί τουλάχιστον ένα αντικείμενο (item) στον server. Κάθε αντικείμενο που δημιουργείται ουσιαστικά αποτελεί μία μεταβλητή που θέλει να έχει πρόσβαση στο WinCC OA. Για την μεταξύ τους επικοινωνία θα πρέπει να γίνει εγκατάσταση του server όπως αναφέρθηκε στην παράγραφο 3.3. Επόμενο βήμα είναι να δημιουργηθεί project στο WinCC OA. Στο WinCC OA ο χρήστης μπορεί να επιλέξει ανάμεσα από δύο επιλογές. Η μία είναι να προσθέσει τα panels που θέλει να δημιουργήσει στο ήδη υπάρχον πρόγραμμα, ενώ η δεύτερη επιλογή είναι να δημιουργήσει ένα καινούργιο project και απλά να χρησιμοποιήσει τα panels από το project αυτό.

5.2.1 Δημιουργία Καινούργιου project στο WinCC OA

Πριν όμως δούμε τη διαδικασία με τον οποία δημιουργήθηκε αυτή η εργασία, θα δειχθεί ο τρόπος με τον οποίο ο χρήστης δημιουργεί ένα νέο δικό του project. Αρχικά γίνεται εκκίνηση του WinCC OA καθώς και του project administrator και στην ο χρήστης επιλέγει "New Project" και στη συνέχεια "Distributed Project", όπως φαίνεται και στην εικόνα 5.2.

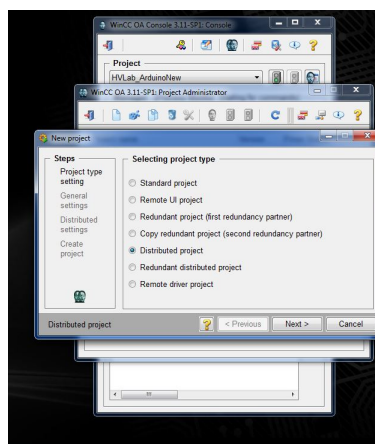


Figure 5.2: Επιλογή Distributed Project κατά τη διαδικασία δημιουργίας νέου project στο WinCC OA

Ο χρήστης μετά εισάγει το επιθυμητό όνομα για το project, το μέρος στο οποίο θα αποθηκευτεί και τελικά γίνεται εισαγωγή ενός System Name για το project, όπως για παράδειγμα dist 1 (βλ. 5.4).

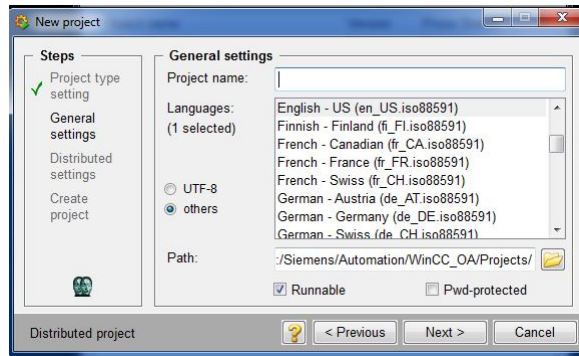


Figure 5.3: Εισαγωγή Ονόματος κατά τη δημιουργία νέου project στο WinCC ΟΑ

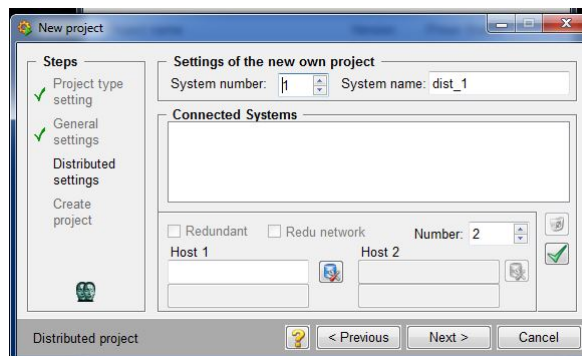


Figure 5.4: Εισαγωγή System Name κατά τη δημιουργία νέου project στο WinCC ΟΑ

Πατώντας "OK" στο τελικό παράθυρο, δημιουργείται το project. Επόμενο βήμα είναι να συνδεθεί το project που δημιουργήθηκε με τον OPC Server. Επιλέγεται από τον χρήστη το project που μόλις δημιουργήθηκε και ο χρήστης κάνει click στην επιλογή "edit config file". Αφού επιλεγθεί το "config file" στο παράθυρο που αναδύεται ο χρήστης πρέπει να προσθέσει:

```
[opc_7] server = "ArduinoOPCServer" "ArduinoOPCServer.1"
```

και στη συνέχεια ο χρήστης πατά "save", "exit" όπως φαίνεται και στην εικόνα 5.5

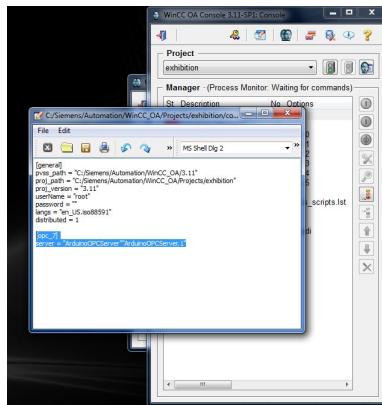


Figure 5.5: Διαδικασία δήλωσης του OPC Server στο configuration file του project

Στη συνέχεια ο χρήστης πατάει το κουμπί "append new manager". Έτσι ανοίγει ένα παράθυρο στο οποίο επιλέγεται "OPC DA Client". Στο επόμενο παράθυρο στο πεδίο Options εισάγεται -num 7 και στο Start Mode επιλέγεται manual και "OK" (βλ. εικόνα 5.6).

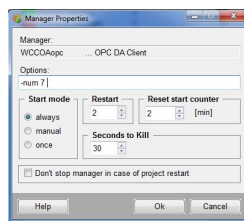


Figure 5.6: Διαδικασία δήλωσης του OPC DA Client

Για την ολοκλήρωση της διαδικασίας ο χρήστης πρέπει να εκκινήσει τον manager που μόλις δημιουργήθηκε. Στη συνέχεια ο χρήστης πρέπει να ανοίξει τον GEDI και τον PARA. Στη συνέχεια επιλέγεται η επιλογή "Internal Datapoints" και στο DPT "_DriverCommon" ο χρήστης κάνοντας δεξί κλικ επιλέγει "Create Datapoint". Στο παράθυρο που αναδύεται ο χρήστης πληκτρολογεί το όνομα "_Driver7" και επιλέγει "OK". Η ίδια διαδικασία επαναλαμβάνεται για το DPT "_Statistics_Driver_Configs", αυτή τη φορά όμως εισάγεται σαν όνομα το "_Stat_Configs_driver7".

5.3 Εγκατάσταση JCOP Framework Components

Στο υποκεφάλαιο αυτό θα γίνει περιγραφή της εγκατάστασης του JCOP Framework στο project αυτό. Αρχικά ο χρήστης επιλέγει το αρχείο **fwInstaller.zip** και το εξάγει στο φάκελο του project που έχει δημιουργήσει. Στη συνέχεια απαιτείται να γίνει εξαγωγή του αρχείου **framework.zip** σε οποιοδήποτε σημείο του υπολογιστή. Έπειτα πρέπει να γίνει εκκίνηση του project στο WinCC OA και να ανοιχθεί το **fwInstallation** panel. Για να πραγματοποιηθεί η παραπάνω διαδικασία, ο χρήστης πρέπει αρχικά αφού ανοίξει το project, να ενεργοποιήσει το GEDI (Graphics EDItor) και να ακολουθήσει τα εξής βήματα όπως φαίνεται και στις εικόνες.

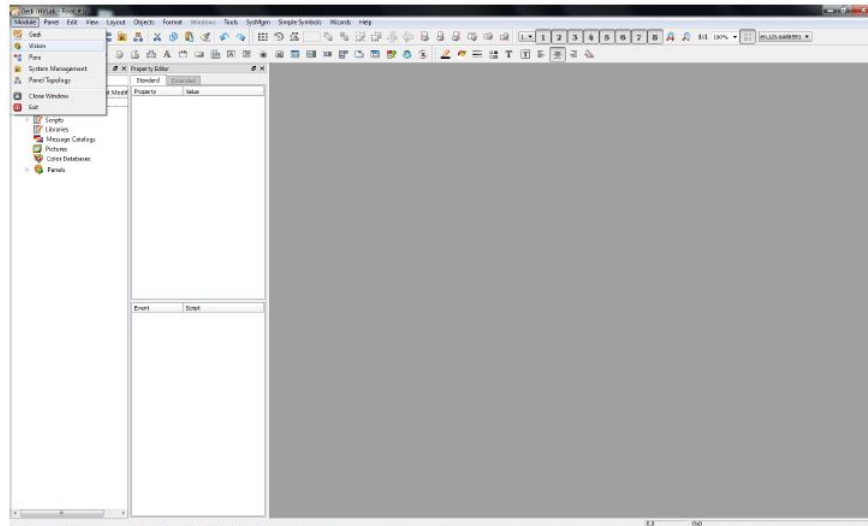


Figure 5.7: Άνοιγμα fwInstallation Panel με την επιλογή Module →Vision

Στο πάνω αριστερά μέρος του GEDI ο χρήστης επιλέγει Module →Vision. Στη συνέχεια ανοίγει το παράθυρο που φαίνεται στην εικόνα 5.8 και από εκεί επιλέγεται ο φάκελος **fwInstallation** και ανοίγεται το **fwInstallation.pnl**.

Όταν ανοίξει το **fwInstallation panel** ζητά από το χρήστη να επιλέξει έναν φάκελο, όπου θα αποθηκευτούν τα components. Αφού δημιουργηθεί ένας τέτοιος φάκελος (αν δεν έχει ήδη δημιουργηθεί) πρέπει στη συνέχεια να δηλωθεί ο φάκελος στον οποίο είχαν εξαχθεί τα περιεχόμενα του **framework.zip** αρχείου. Ο πίνακας Available components συμπληρώνεται από όλα τα components τα οποία μπορούν να εγκατασταθούν στο project. Γίνεται επιλογή των επιθυμητών components από τον χρήστη. Στο project αυτό χρειάστηκαν τα **fwAccessControl**, **fwCaen**, **fwCore**, **fwTrending**, **fwXML**. Αφού γί-

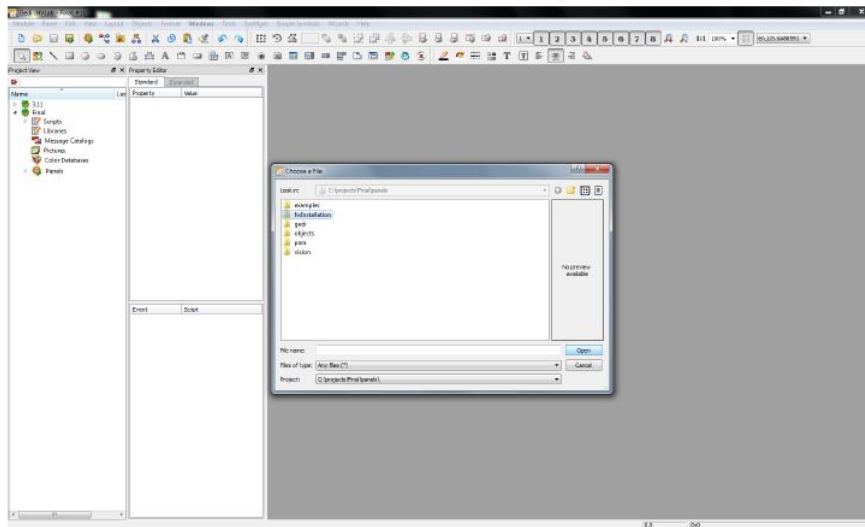


Figure 5.8: Επιλογή fwInstallation και άνοιγμα του fwInstallation.pnl

νουν οι επιλογές των components ο χρήστης κάνει κλικ στο κουμπί "Install" και πατά "OK". Στα δύο επόμενα παράθυρα επιλέγεται το "Activate Driver" και "Restart Project".

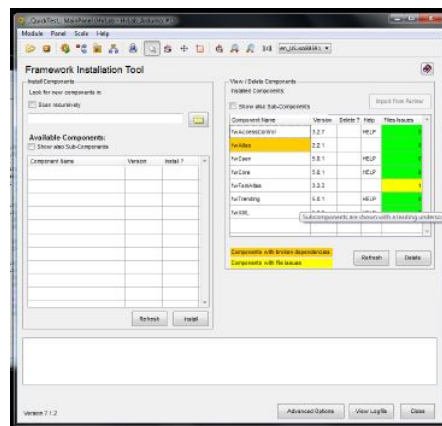


Figure 5.9: Παράθυρο fwInstallation Panel για την εγκατάσταση των components

5.4 Slow Control System

Το PVSS αποτελεί μία προηγούμενη έκδοση του WinCC ΟΑ που είχε σκοπό την παρακολούθηση και συλλογή δεδομένων από τους ανιχνευτές. Στο PVSS υπήρχε ένα project το οποίο είχε αναπτυχθεί στα πλαίσια της μεταπτυχιακής διπλωματικής εργασίας "Micromegas Telescope for te RD51 Test Beams". Το project αυτό μεταφέρθηκε από το PVSS στο WinCC ΟΑ στα πλαίσια της διπλωματικής εργασίας "Ανάπτυξη Συστήματος Μέτρησης Θερμοκρασίας και Υγρασίας βασισμένο σε Arduino και WinCC ΟΑ". Το κύριο παράθυρο του συστήματος φαίνεται παρακάτω στην εικόνα 5.10.

Channel	Min	Max	Value	Unit	Min	Max	Value	Unit	Min	Max	Value	Unit	Min	Max	Value	Unit
1	1.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
2	1.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
3	1.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
4	2.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
5	1.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
6	2.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
7	2.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
8	1.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
9	0.8	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
10	0.8	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
11	0.8	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
12	1.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
13	0.5	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
14	0.8	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
15	0.8	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
16	0.8	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
17	1.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
18	0.8	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
19	1.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
20	0.8	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
21	1.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
22	0.8	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
23	1.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
24	0.8	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
25	1.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
26	0.8	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
27	1.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
28	0.8	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
29	1.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA
30	0.8	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA	0.0	0.000	0.000	mA

Figure 5.10: Κύριο Παράθυρο Slow Control System

Στο υπόλοιπο κομμάτι του υποκεφαλαίου αυτού, θα γίνει παρουσίαση του project της εργασίας αυτής στο WinCC ΟΑ, η οποία στηρίχτηκε στην διπλωματική εργασία "Ανάπτυξη Συστήματος Μέτρησης Περιβαλλοντικών Συνθηκών Βασισμένο σε Arduino και WinCC_OA".

Ανοίγοντας το main panel, ο χρήστης κάνει κλικ στην επιλογή "Environmental Parameters". Εκεί εμφανίζεται το παράθυρο της εικόνας 5.11 όπου ο χρήστης και επιλέγει "Arduino YUN". Στη συνέχεια ο χρήστης ανακατευθύνεται στο main configuration panel για το Arduino Yun. Ο χρήστης πρέπει αρχικά να κάνει κλικ στο κουμπί Configuration και εμφανίζεται το παράθυρο

στο οποίο ο χρήστης θα καθορίσει τις ρυθμίσεις για το Arduino Yun. Στα πεδία αυτά δηλαδή θα καθορίσει τον αριθμό των PCB που θα χρησιμοποιήσει, τον αριθμό των αισθητήρων σε κάθε PCB, καθώς και τα ονόματα των αντικειμένων (items) που θα στέλνει το Arduino έτσι ώστε το WinCC OA να επικοινωνεί και να λαμβάνει δεδομένα από το Arduino.

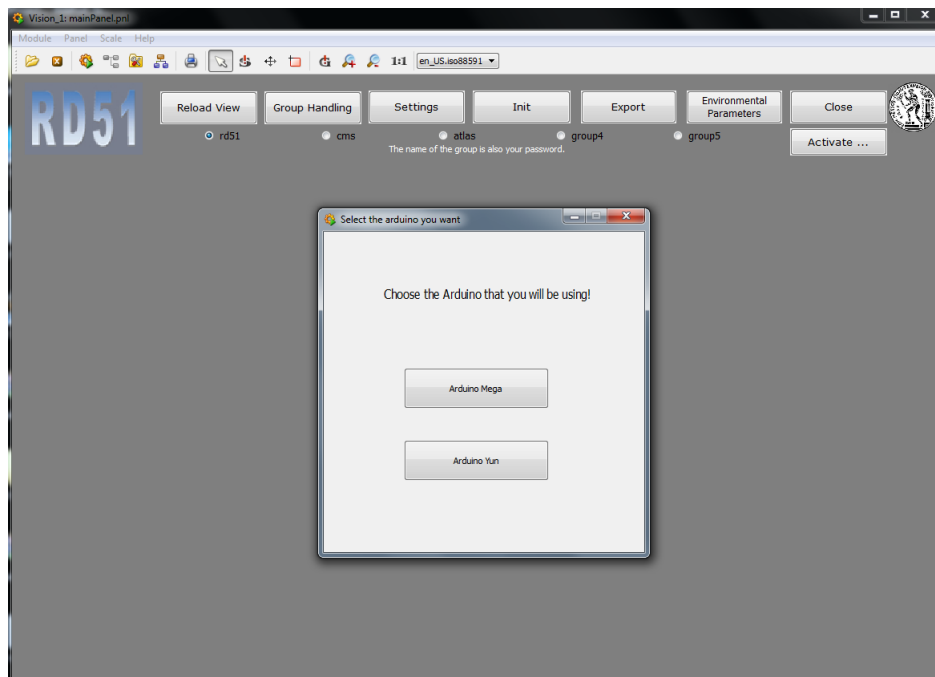


Figure 5.11: Κεντρικό Παράθυρο Environmental Parameters

5.4.1 Configuration

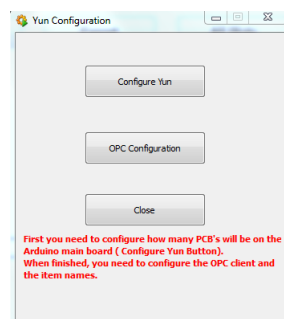


Figure 5.12: Παράθυρο Configuration για το Arduino YUN

Ας δούμε όμως λίγο πιο αναλυτικά τον τρόπο με τον οποίο γίνονται οι ρυθμίσεις. Πρώτα από όλα, ο χρήστης πρέπει να καθορίσει τον αριθμό των PCB και των αισθητήρων που θα χρησιμοποιήσει. Συνεπώς κάνει κλικ στην επιλογή "Configure Yun". Εκεί εμφανίζεται ένα νέο πάνελ (βλ 5.13. Εκεί ο χρήστης δηλώνει τον αριθμό των PCB που θα χρησιμοποιήσει. Στη συνέχεια στο panel 5.14 ο χρήστης δηλώνει το PCB για το οποίο θέλει να καθορίσει τον αριθμό των αισθητήρων. Αν δηλαδή ο χρήστης έχει δηλώσει ότι θα έχει μόνο 1 PCB, τότε θα καθορίσει τον αριθμό των αισθητήρων για το PCB1 όπως φαίνεται στην εικόνα 5.15.

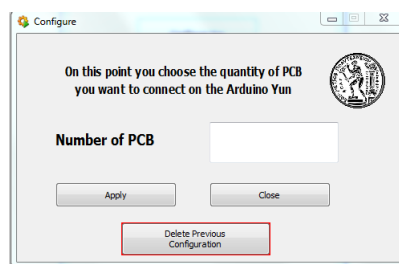


Figure 5.13: Configure Yun Panel

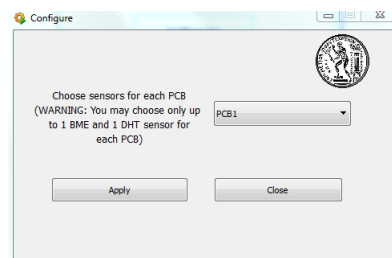


Figure 5.14: Configure Yun Panel(2)

Η διαδικασία δήλωσης των PCB έχει ολοκληρωθεί με τον τρόπο που αναφέρθηκε παραπάνω. Στη συνέχεια πρέπει ο χρήστης να καθορίσει τις ρυθμίσεις για τη σύνδεση των τιμών που στέλνει το Arduino με τα αντίστοιχα DPE (DataPoint Elements) του WinCC ΟΑ. Στο αρχικό λοιπόν Configuration Panel του Arduino YUN, ο χρήστης κάνει κλικ στην επιλογή "OPC Configuration". Εκεί διαλέγει με τον ίδιο τρόπο όπως και πριν, για ποιο PCB θα επιλέξει να κάνει ρυθμίσεις.

Στη συνέχεια αναδύεται το παράθυρο 5.17 για το PCB που έχει επιλέξει ο χρήστης. Στο panel αυτό, ο χρήστης πρέπει να καθορίσει το όνομα των αντικειμένων που θέλει να διαβάσει. Τα αντικείμενα που θέλει να διαβάσει ο χρήστης είναι τα αντικείμενα που στέλνει το Arduino και περιέχουν τις τιμές για τις περιβαλλοντικές συνθήκες που μελετά ο χρήστης.

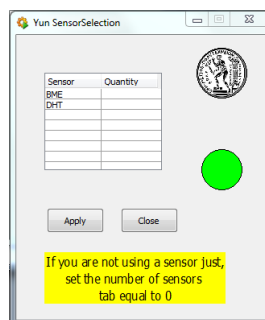


Figure 5.15: Ορισμός αριθμού αισθητήρων για κάθε PCB

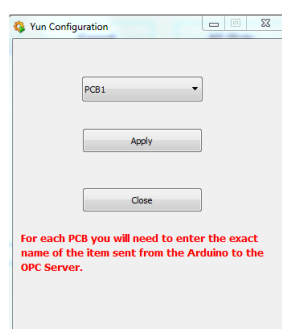


Figure 5.16: OPC Configuration Panel

Ο server είναι προεπιλεγμένος ArduinoOPCServer. Στην επιλογή "Driver Number" ο χρήστης πληκτρολογεί "7". Στην επιλογή "Group" επιλέγει "ArduinoGroup". Στο DP Element διαλέγει την τιμή για την οποία θέλει να καθορίσει το όνομά της και στην επιλογή "OPC Item Name" πληκτρολογεί το όνομα του αντικειμένου. Ο καθορισμός του ονόματος γίνεται με την παρακάτω λογική.

Οι μεταβλητές από τους BME αισθητήρες είναι της μορφής **h_bmei**, **p_bmei**, **t_bmei** και για τους DHT της μορφής **hum_i**, **tem_i** όπου το **i** είναι ο αριθμός του PCB που έχει επιλεγεί κάθε φορά. Συνεπώς αν για παράδειγμα ο χρήστης καθορίζει τα ονόματα των αντικειμένων για το 1ο PCB θα έχουμε:

- **h_bme1**: Η τιμή της υγρασίας που στέλνει ο BME από το 1ο PCB
- **p_bme1**: Η τιμή της πίεσης που στέλνει ο BME από το 1ο PCB
- **t_bme1**: Η τιμή της θερμοκρασίας που στέλνει ο BME από το 1ο PCB
- **hum1**: Η τιμή της υγρασίας που στέλνει ο DHT από το 1ο PCB
- **tem1**: Η τιμή της θερμοκρασίας που στέλνει ο DHT από το 1ο PCB

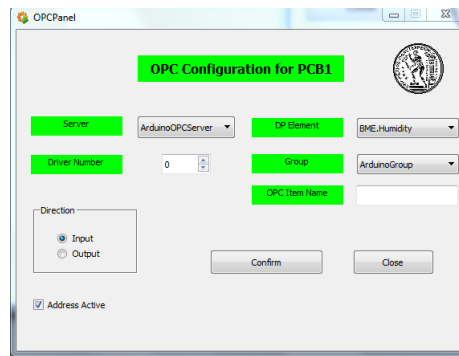


Figure 5.17: Configuration OPC Panel(2)

Μετά τον καθορισμό των ονομάτων για κάθε PCB που χρησιμοποιείται, ο χρήστης κάνει κλικ στην επιλογή "Confirm".

5.4.2 Archive

Στο σημείο αυτό θα γίνει περιγραφή του "Archive Panel". Στο αρχικό main panel ο χρήστης κάνει κλικ στην επιλογή "Archive". Έτσι εμφανίζεται το παράθυρο στο οποίο ο χρήστης επιλέγει για ποιο PCB θέλει να καθορίσει τις ρυθμίσεις για το Archiving και ανακατευθύνεται στο παράθυρο 5.18 στο οποίο γίνονται οι ρυθμίσεις για το archiving του αντίστοιχου PCB.

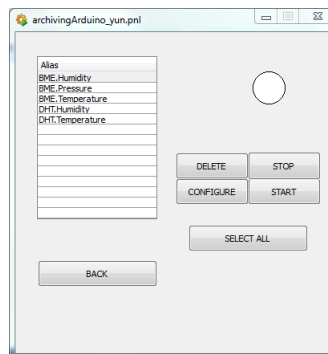


Figure 5.18: Archiving Panel

Οι ρυθμίσεις για το archiving είναι προκαθορισμένες. Οι τιμές λαμβάνονται ανά 5 λεπτά ή αν αλλάξει η τιμή τους. Για τη θερμοκρασία και την υγρασία το όριο αλλαγής τιμής είναι 1 βαθμός ενώ για την πίεση το όριο αλλαγής τιμής είναι 50 mbar. Για τον καθορισμό όλων των μεταβλητών ο χρήστης κάνει κλικ στην επιλογή "Select all", στη συνέχεια κλικ στο "Configure", μετά "Start"

και το archiving έχει ξεκινήσει. Αν ο χρήστης θέλει να σταματήσει τη διαδικασία μπορεί να το κάνει αυτό πατώντας το κουμπί "STOP" ενώ μπορεί ακόμα και να διαγράψει τις ρυθμίσεις που έχουν γίνει κάνοντας κλικ την επιλογή "DELETE". Τα δεδομένα που λαμβάνονται από την διαδικασία αυτή αποθηκεύονται στην βάση δεδομένων (database) του WinCC OA. Στο επόμενο υποκεφάλαιο θα αναλυθεί ο τρόπος με τον οποίο ο χρήστης μπορεί να πάρει τα δεδομένα (export) και να τα επεξεργαστεί.

5.4.3 Export

Στο σημείο αυτό θα γίνει αναλυτική περιγραφή του Export Panel. Στο main panel, ο χρήστης κάνει κλικ στην επιλογή Export. Και εδώ όπως και πριν ο χρήστης επιλέγει το PCB για το οποίο θέλει να κάνει εξαγωγή των αποθηκευμένων δεδομένων και αναδύεται το παράθυρο 5.20.

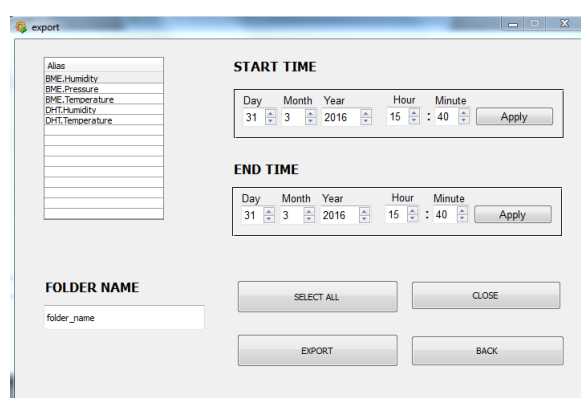


Figure 5.19: Export Panel

Ο χρήστης κάνοντας κλικ στην επιλογή "SELECT ALL" επιλέγει όλες τις μεταβλητές για το PCB που έχει επιλέξει. Στη συνέχεια στο "START TIME" επιλέγει την χρονική στιγμή από την οποία θέλει να ξεκινήσει το αρχείο εξαγωγής, και με την επιλογή "END TIME" την τελική στιγμή που θέλει στο export αρχείο. Στο πεδίο "FOLDER NAME" πληκτρολογεί το όνομα του φακέλου μέσα στον οποίο θα υπάρχουν τα δεδομένα με τις τιμές που έχουν ληφθεί για τη χρονική διάρκεια που έχει καθορίσει ο χρήστης. Πατώντας το κουμπί "EXPORT" δημιουργείται ο ζητούμενος φάκελος.

5.4.4 All Plots

Κάνοντας κλικ στην επιλογή "All Plots" του αρχικού main panel, ο χρήστης ανακατευθύνεται στο panel 5.9 στο οποίο επιλέγει το PCB που θέλει να ασχοληθεί και για ποιες τιμές θέλει να γίνονται διαγράμματα.

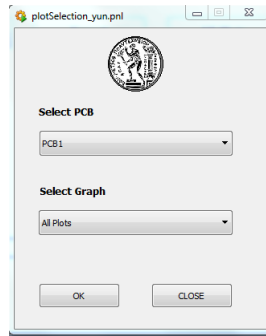


Figure 5.20: All Plots Panel

5.4.5 Real Time Data

Παρακάτω δίνονται διάφορα screenshots που λήφθηκαν κατά τη διάρκεια μετρήσεων στο test beam τον Μάιο στο RD51 Collaboration στο CERN.

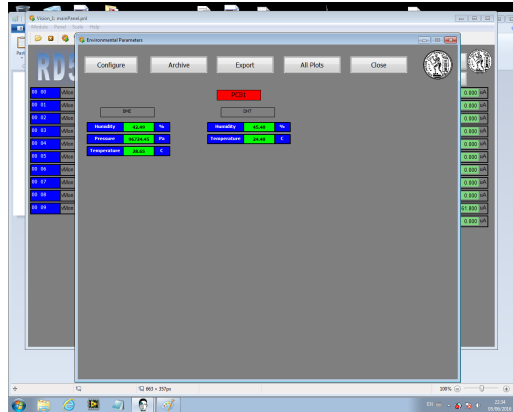


Figure 5.21: Ardenvino Project Main Panel

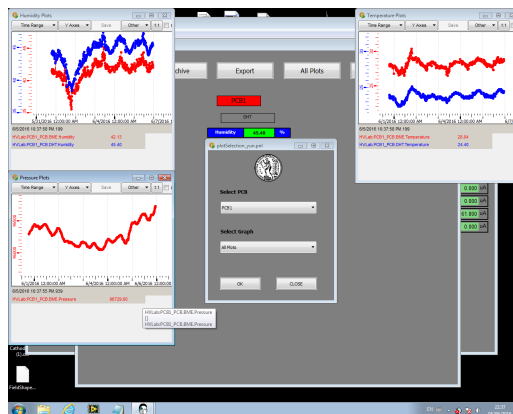


Figure 5.22: All Plots Panel

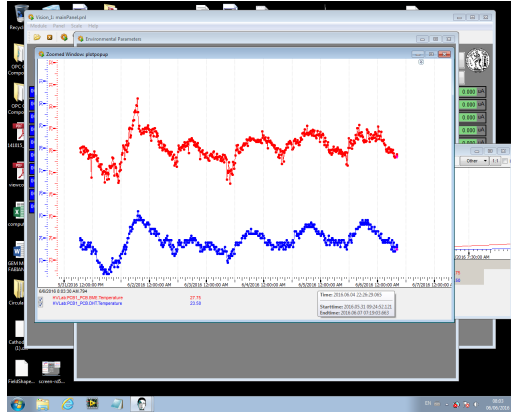


Figure 5.23: Real Time Temperature Measurements

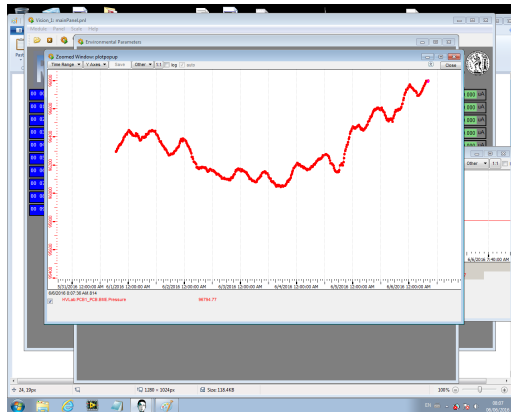


Figure 5.24: Real Time Pressure Measurements

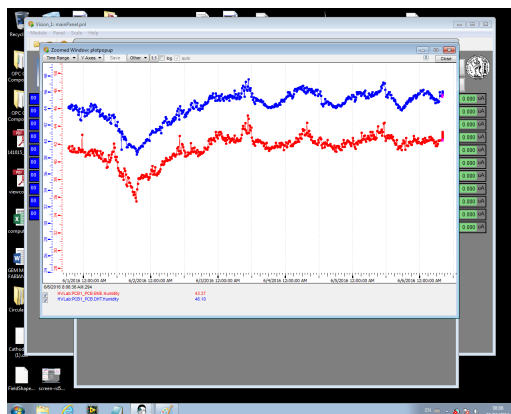


Figure 5.25: Real Time Humidity Measurements

Κεφάλαιο 6

Συμπεράσματα

Η εργασία αυτή, δημιουργήθηκε με σκοπό τη μέτρηση, καταγραφή, αποθήκευση και επεξεργασία από απόσταση των περιβαλλοντικών συνθηκών πίεσης, θερμοκρασίας και υγρασίας. Το σύστημα που δημιουργήθηκε βασίζεται στην μικροελεγκτική πλακέτα **Arduino YUN** που είναι υπεύθυνη για την καταγραφή των περιβαλλοντικών συνθηκών και την αποστολή τους στον OPC Server και στο σύστημα αυτόματου ελέγχου **WinCC_OA** το οποίο είναι υπεύθυνο για την καταγραφή και αποθήκευση των μετρήσεων. Στην εργασία χρησιμοποιήθηκαν 2 αισθητήρες. Ο πρώτος **DHT 22** είχε τη δυνατότητα καταγραφής της υγρασίας και της θερμοκρασίας με χρήση SPI πρωτοκόλλου, ενώ ο δεύτερος **BME280** κατέγραφε πίεση, θερμοκρασία και υγρασία κάνοντας χρήση I^2C πρωτοκόλλου. Η περιγραφή της συμπεριφοράς των αισθητήρων σε πραγματικές συνθήκες μέτρησης έγινε στο δεύτερο κεφάλαιο. Ο DHT 22 δίνει στο χρήστη μια σχετικά ικανοποιητική τιμή (relative scale) ενώ ο BME280 λαμβάνει μετρήσεις με πολύ μεγάλη ακρίβεια για πίεση, θερμοκρασία και υγρασία.

Αρχικά στο RD51 Collaboration του CERN, χρησιμοποιήθηκε ένα προσωρινό μοντέλο (project board) το οποίο είχε μία οθόνη LCD, έναν αισθητήρα BME 280 και έναν αισθητήρα DHT 22. Οι μετρήσεις έγιναν κατά τη διάρκεια του test beam και τα αποτελέσματα που πάρθηκαν για 5 ημέρες δίνονται από τα παρακάτω διαγράμματα:

Δίνονται στη συνέχεια φωτογραφίες από χρήση του τελικού προϊόντος **Ardenvino** κατά τη χρήση του στο εργαστήριο.

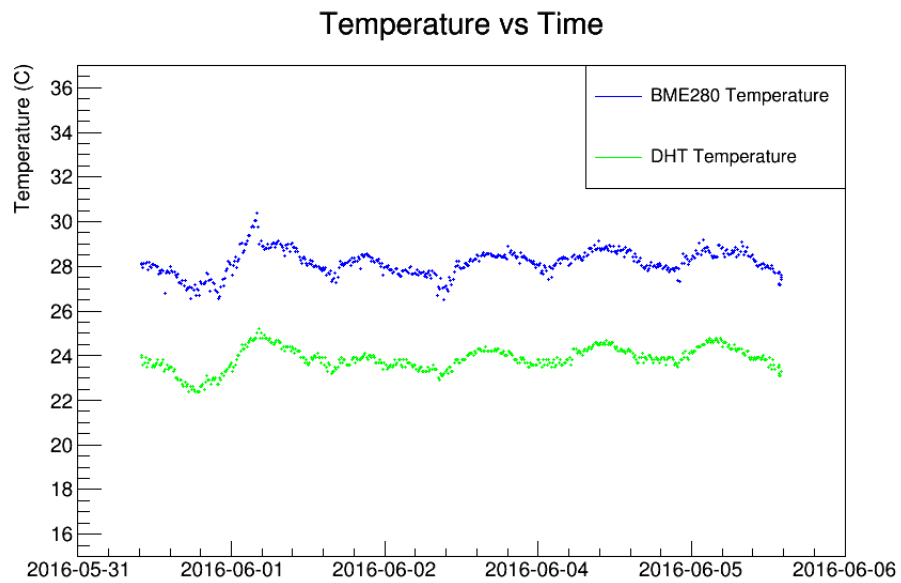


Figure 6.1: Θερμοκρασία από τους αισθητήρες BME και DHT κατά τη διάρκεια test beam

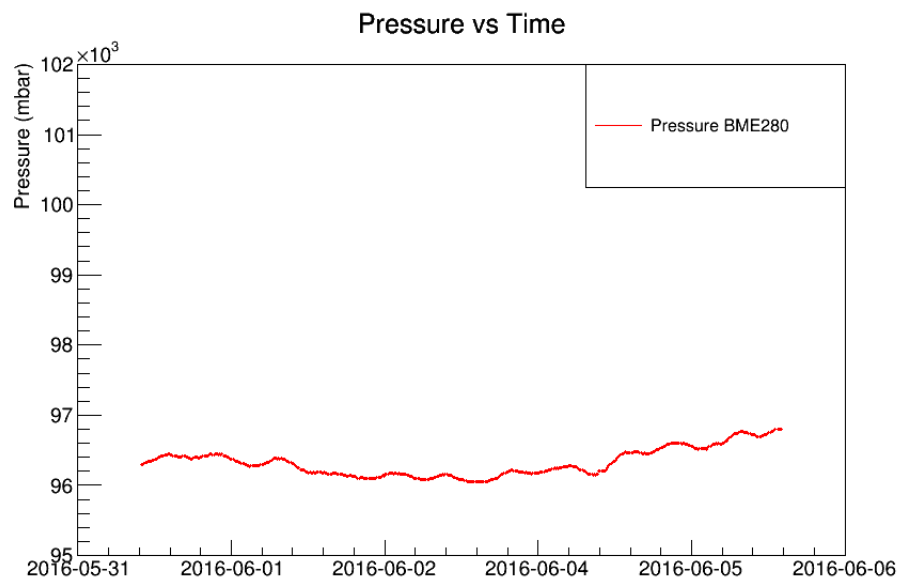


Figure 6.2: Πίεση από τον αισθητήρα BME κατά τη διάρκεια test beam

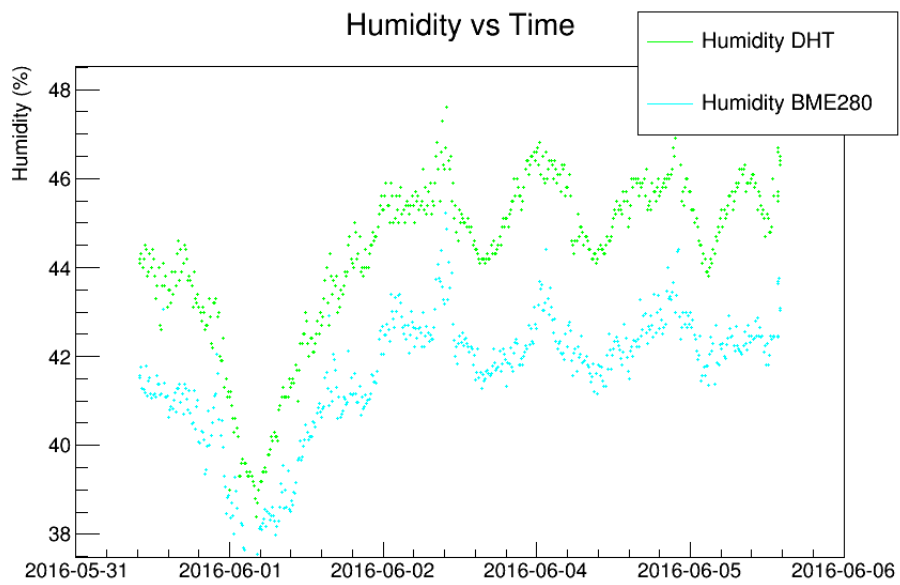


Figure 6.3: Υγρασία από τους αισθητήρες BME και DHT κατά τη διάρκεια test beam

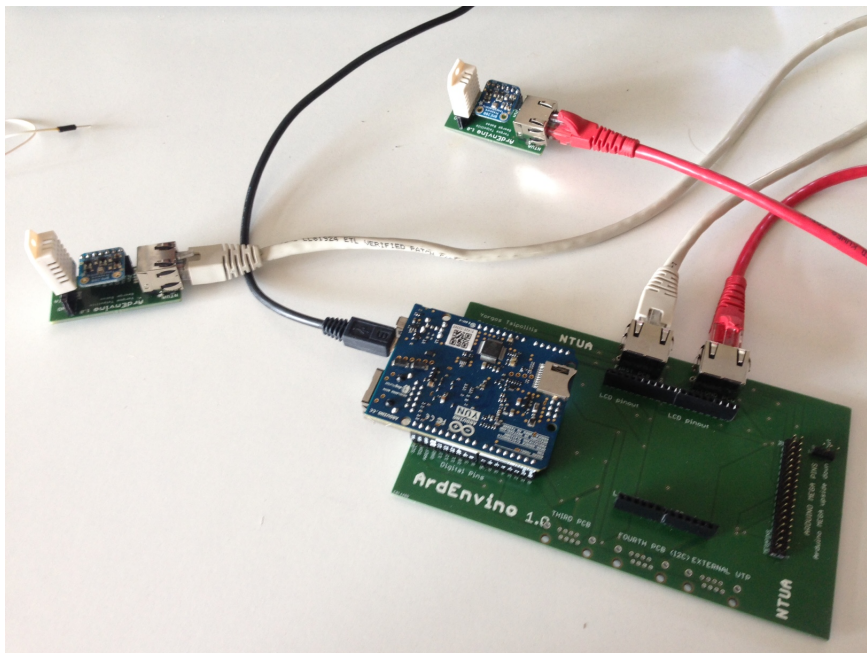


Figure 6.4: Ardenvino με χρήση Arduino YUN

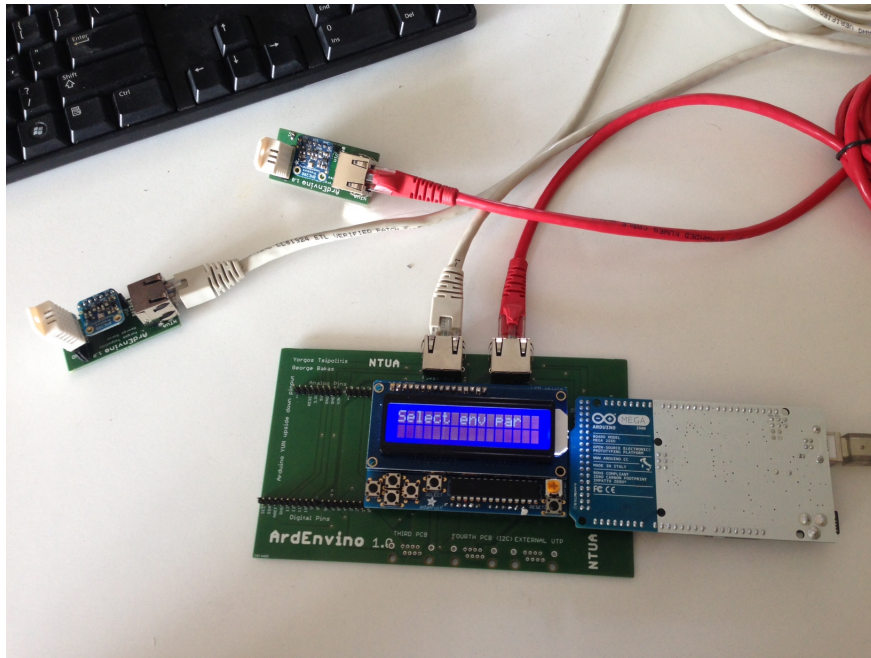


Figure 6.5: Ardenvino με χρήση Arduino Mega

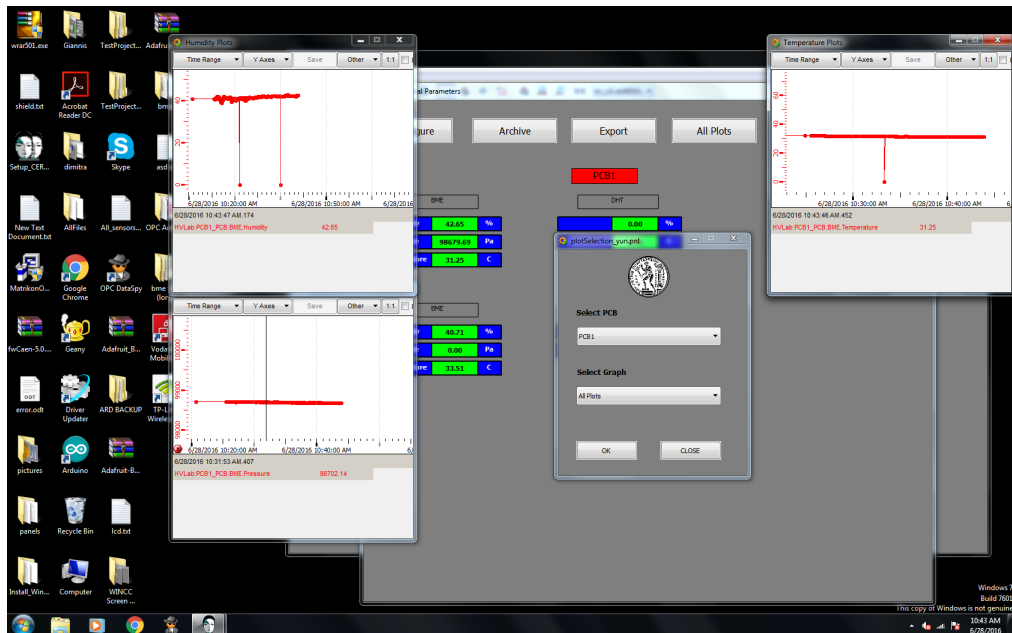


Figure 6.6: Δεδομένα που πάρθηκαν με χρήση ArdEnvino σε πραγματικό χρόνο

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 - ArduinoYun. URL: <https://www.arduino.cc/en/Main/ArduinoBoardYun>
- [6] Arduino — Wikipedia, The Free Encyclopedia. 2015. URL: <https://en.wikipedia.org/wiki/Arduino>.
- [7] Nick Gammon. I2C - Two-Wire Peripheral Interface - for Arduino. 2011. URL: <http://gammon.com.au/i2c>.
- [8] M Gonzalez-Berges. «The joint controls project framework». In: arXiv preprint physics/0305128 (2003).
- [9] Adafruit- BME 280 Sensor: URL:<https://learn.adafruit.com/adafruit-bme280-humidity-barometric-pressure-temperature-sensor-breakout/overview>
- [10] Adafruit - TCAA9548A: URL:<https://learn.adafruit.com/adafruit-tca9548a-1-to-8-i2c-multiplexer-breakout/wiring-and-test>
- [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] CadSoft EAGLE PCB Design : URL: <http://www.cadsoftusa.com/>
- [20] Adafruit - BMP 085 Sensor: URL: <https://learn.adafruit.com/bmp085/using-the-bmp085>

[21] Open Platform Communications — Wikipedia, The Free Encyclopedia.2015. URL: https://en.wikipedia.org/wiki/Open_Platform_Communications.

[22] 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>.

[23] Παναγιώτης Μ. Παπάζογλου. Ανάπτυξη Εφαρμογών με το Arduino. Εκδόσεις Τζιόλα, 2014. ISBN: 978-960-418-459-0.

[24] Παπακριβόπουλος Γιάννης, "Ανάπτυξη συστήματος μέτρησης θερμοκρασίας και υγρασίας βασισμένο σε Arduino και WinCC_OA". Προπτυχιακή διπλωματική εργασία. Εθνικό Μετσόβιο Πολυτεχνείο, 2015