

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ
ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΟΜΕΑΣ ΦΥΣΙΚΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΠΕΙΡΑΜΑΤΙΚΗΣ ΦΥΣΙΚΗΣ ΥΨΗΛΩΝ
ΕΝΕΡΓΕΙΩΝ



**Ανάπτυξη συστήματος λήψης
περιβαλλοντικών συνθηκών βασισμένο σε
Raspberry Pi και WinCC OA**

Πτυχιακή Εργασία

της

Άννας Ζαχαροπούλου

Αθήνα, Ιούνιος 2017

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ
ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΟΜΕΑΣ ΦΥΣΙΚΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΠΕΙΡΑΜΑΤΙΚΗΣ ΦΥΣΙΚΗΣ ΥΨΗΛΩΝ
ΕΝΕΡΓΕΙΩΝ



**Ανάπτυξη συστήματος λήψης
περιβαλλοντικών συνθηκών βασισμένο σε
Raspberry Pi και WinCC OA**

Πτυχιακή Εργασία

της

Άννας Ζαχαροπούλου

Επιβλέπων:

Γεώργιος Τσιπολίτης
Καθηγητής ΕΜΠ

Τριμελής Επιτροπή

Γ. Τσιπολίτης
Καθηγητής
ΕΜΠ
(επιβλέπων)

Κ. Κουσουρής
Επικ. Καθηγητής
ΕΜΠ

Χ. Μάρκου
Ερευνητής Α
ΕΚΕΦΕ «ΔΗΜΟΚΡΙΤΟΣ»

Αθήνα, Ιούνιος 2017

ANNA ΖΑΧΑΡΟΠΟΥΛΟΥ

Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών

© Άννα Ζαχαροπούλου, 2017

Με επιφύλαξη παντός δικαιώματος-All rights reserved

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται η παρούσα σημείωση. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς το συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτή τη διατριβή εκφράζουν το συγγραφέα και δεν πρέπει να θεωρηθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Στα πειράματα Φυσικής Υψηλών Ενεργειών (ΦΥΕ) ο αυτόματος έλεγχος και η παρακολούθηση των τιμών των περιβαλλοντικών παραμέτρων καθίστανται σημαντικά στοιχεία, για τη σωστή και άρτια διεκπεραίωση των πειραμάτων. Αυτό οφείλεται στο γεγονός, πως η μεταβολή στην τιμή της πίεσης, της θερμοκρασίας και της υγρασίας επηρεάζει τη λειτουργία και κατ'επέκταση το αποτέλεσμα των ανιχνευτικών διατάξεων. Κατά τη διάρκεια των test beams στο RD51 Collaboration του CERN, η δυνατότητα λήψης, εμφάνισης και καταγραφής των περιβαλλοντικών συνθηκών από απόσταση είναι καθοριστική. Η άμεση παρατήρηση και ο έλεγχος των παραμέτρων αυτών, αποσκοπούν στην πρόληψη, στην αποτροπή καθώς και στη μετέπειτα ανάλυση της μεταβολής του gain των Gas Chambers, που χρησιμοποιούνται.

Σκοπός της παρούσας διπλωματικής εργασίας ήταν η ανάπτυξη ενός συστήματος μέτρησης και καταγραφής των περιβαλλοντικών παραμέτρων με χρήση του μικροϋπολογιστή Raspberry Pi 3+ και κατάλληλου αισθητήρα. Η λήψη μετρήσεων θερμοκρασίας, πίεσης και υγρασίας από τους αισθητήρες μέσω του Raspberry Pi, αποσκοπεί στη δυνατότητα παρακολούθησης των τιμών τους, καθώς και στην αποθήκευσή τους για περαιτέρω ανάλυση. Η διαδικασία ξεκινάει με το Raspberry Pi να λαμβάνει τιμές από τους αισθητήρες. Στη συνέχεια, τα δεδομένα αυτά μέσω του πρωτοκόλλου HTTP αποστέλλονται σε web server, με σκοπό την παρακολούθησή τους. Επιπλέον, βάσει του πρωτοκόλλου επικοινωνίας OPC UA τα δεδομένα αποστέλλονται στο πρόγραμμα SCADA WinCC OA με σκοπό την παρακολούθηση αλλά και την αποθήκευσή τους, ώστε να επεξεργαστούν. Η επικοινωνία μεταξύ Raspberry Pi και του WinCC OA πραγματοποιείται μέσω ενός OPC UA Server, ο οποίος στήθηκε στα πλαίσια της παρούσας διπλωματικής για αυτόν ακριβώς το σκοπό.

Abstract

At High-Energy Physics experiments, automatic control and monitoring of environmental parameters have a great effect on the proper and thorough execution of experiments. This is due to the fact that changes in the values of pressure, temperature and humidity affect the function and thus the result of the detectors. During test beams at RD51 CERN's Collaboration, the ability to record and display remotely environmental conditions is critical. The immediate observation and control of these parameters are aimed at preventing, deterring and subsequently explaining the change in the Gas Chambers' gain.

The aim of this thesis was the development of a system that measures and records environmental conditions using the microcomputer Raspberry Pi 3+ and the appropriate environmental sensors. The reason of receiving temperature, pressure and humidity measurements from Raspberry Pi is to monitor these values as well as store them for further analysis. The process begins with Raspberry Pi receiving values from the sensors. Thereafter, these data are sent to a web server through HTTP protocol for monitoring. In addition, based on OPC UA communication protocol, data are sent to the SCADA program WinCC OA, for monitoring and storage in order to process them. The communication between Raspberry Pi and WinCC OA is carried out through an OPC UA Server, which was implemented within the scope of this thesis for this very purpose.

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κύριο Γεώργιο Τσιπολίτη, καθηγητή του Εθνικού Μετσόβιου Πολυτεχνείου, που με εμπιστεύτηκε και μου έδωσε την ευκαιρία να ασχοληθώ με το συγκεκριμένο θέμα, καθώς και για τη συνεχή βοήθεια και καθοδήγηση που μου παρείχε όλο αυτό το διάστημα. Στη συνέχεια, θα ήθελα να ευχαριστήσω τους Γιάννη Παπακρισόπουλο και Γιώργο Μπάκα, καθώς η βοήθειά τους ήταν καθοριστική για τη διεκπεραίωση της διπλωματικής μου εργασίας. Επιπλέον, ευχαριστώ από καρδιάς την οικογένειά μου και ιδιαιτέρως τον πατέρα μου, Γιώργο, για την υπομονή που έδειξε, αλλά κυρίως για την στήριξη και την πολύτιμη βοήθεια, που μου παρείχε καθόλη τη διάρκεια της εξέλιξης της εργασίας. Τέλος, τους φίλους και δικούς μου ανθρώπους, που με στηρίζουν και με υποστηρίζουν όλα αυτά τα χρόνια.

Περιεχόμενα

Κατάλογος Σχημάτων	1
1 Εισαγωγή	5
2 Raspberry Pi	9
2.1 Εισαγωγή	9
2.1.1 Αναδρομή στα μοντέλα Raspberry Pi	11
2.2 Raspberry Pi 3 Model B+	12
2.2.1 Γενικά Τεχνικά Χαρακτηριστικά	12
2.2.2 GPIO (General Purpose Input/Output)	13
2.3 Προσεκρόπιση λογισμικού (Software) του Raspberry Pi	15
2.3.1 Operating System (OS) Raspbian	15
2.3.2 Γλώσσα προγραμματισμού Python	15
2.3.3 Apache HTTP	16
2.3.3.1 PHP και HTML	17
3 Εγκατάσταση και Ρυθμίσεις Raspberry Pi (Setup)	19
3.1 Εγγραφή εικόνας Raspbian	19
3.2 Εκκίνηση Raspberry Pi	22
3.2.1 Έμμεση πρόσβαση και διαχείριση της λειτουργίας από Windows μέσω Secure Shell (SSH)	22
3.2.1.1 Γραφικό περιβάλλον - VNC Server	28
3.3 Εγκατάσταση και Ενεργοποίηση Apache Server	32
4 Πρωτόκολλο I^2C (Inter-integrated Circuit)	35
4.1 Σειριακή μεταφορά δεδομένων	35
4.1.1 Περιγραφή του πρωτοκόλλου επικοινωνίας I^2C	36
4.2 Raspberry Pi και I^2C	39
4.2.1 Ρυθμίσεις ενεργοποίησης(Configuration)	39
5 Αισθητήρας BME280	43
5.1 Περιγραφή	43
5.1.1 Προδιαγραφές λειτουργίας	45

5.1.2	Λειτουργία αισθητήρα BME280	47
5.1.2.1	Η Φυσική του αισθητήρα BME280	50
5.2	Λήψη δεδομένων με το Raspberry Pi	52
5.2.1	Αποστολή δεδομένων σε web server	57
6	OPC Server	61
6.1	Πρωτόκολλο επικοινωνίας OPC	61
6.1.1	Design	62
6.2	OPC Unified Architecture (UA)	63
6.2.1	Δομή OPC UA Server - Address Space	65
6.3	OPC UA Raspberry Pi Server	68
6.3.1	Δομή OPC UA Raspberry Pi Server	69
7	WinCC OA	71
7.1	Εισαγωγή	71
7.1.1	Βασικά Χαρακτηριστικά Του WinCC OA	71
7.1.2	JCOP Framework	74
7.2	Σύνδεση Raspberry Pi με το WinCC OA	74
7.2.1	Δημιουργία καινούργιου project στο WinCC OA	75
7.2.2	Σύνδεση WinCC OA με OPC UA Server	77
7.3	Εγκατάσταση JCOP Framework Components	84
7.4	Raspberry Pi WinCC OA Project	86
7.4.1	Archive	87
7.4.2	Export	90
7.4.3	All plots - Real Time Data	93
8	Συμπεράσματα	95
	Βιβλιογραφία	101

Κατάλογος Σχημάτων

1.1	Linear dependence: Micromegas Gain with Pressure	6
1.2	Linear dependence: Micromegas Gain with Temperature	6
1.3	Micromegas Gain and Normalized Pressure with Time	7
2.1	Μοντέλα και γενιές Raspberry Pi	11
2.2	Microcomputer Raspberry Pi 3 Model B+	12
2.3	Υποδοχές του Raspberry Pi 3 Model B+	14
2.4	General Pins Input/Output (GPIO)	14
3.1	Εκτέλεση της εφαρμογής SD Formatter	20
3.2	Ενεργοποίηση της προσαρμογής μεγέθους	20
3.3	Τέλος διαδικασίας, πληροφορίες για τη χωρητικότητα	21
3.4	Περιβάλλον εφαρμογής Etcher - Επιλογή .img αρχείου και Drive	21
3.5	Ολοκλήρωση της εγγραφής της εικόνας Raspbian	22
3.6	Ανίχνευση της διεύθυνσης IP του Raspberry Pi με το πρόγραμμα Advanced IP Scanner	23
3.7	Εισαγωγή της διεύθυνσης IP του Raspberry Pi στο πρόγραμμα PuTTY	24
3.8	Εισαγωγή username: pi και password: raspberry κατά την εκκίνηση του Raspberry Pi	25
3.9	Μενού ρυθμίσεων των λειτουργιών του Raspberry Pi	26
3.10	Ενεργοποίηση του VNC Server από τη γραμμή εντολών	29
3.11	Σύνδεση στο VNC Viewer	30
3.12	Γραφικό περιβάλλον του Raspberry Pi	31
3.13	Apache2 test web page	32
4.1	Οι δύο γραμμές SCL και SDA του πρωτοκόλλου επικοινωνίας I^2C	36
4.2	Πρωτόκολλο επικοινωνίας I^2C : Συνθήκες έναρξης και τερματισμού	37
4.3	Απεικόνιση των γραμμών SDA και SCL κατά τη διάρκεια επικοινωνίας με χρήση του πρωτοκόλλου I^2C (7 bit)	38
4.4	Interfacing Options - Configure Connection to peripheral	40
4.5	I2C- Enable-Disable automatic loading	40
4.6	Ενεργοποίηση ARM I2C interface	41
4.7	Αποτέλεσμα εκτέλεσης της εντολής 'sudo i2cdetect -y 1'	42

5.1	Πλακέτα αισθητήρα BME280	43
5.2	Γενικές προδιαγραφές αισθητήρα BME280	45
5.3	Προδιαγραφές παραμέτρου θερμοκρασίας	45
5.4	Προδιαγραφές παραμέτρου πίεσης	46
5.5	Προδιαγραφές παραμέτρου υγρασίας	46
5.6	Χρονικό διάγραμμα λήψης μετρήσεων με τη λειτουργία forced mode συναρτήσε του ρεύματος	47
5.7	Χρονικό διάγραμμα λήψης μετρήσεων με τη λειτουργία normal mode συναρτήσε του ρεύματος	47
5.8	Εναλλαγή μεταξύ των καταστάσεων λειτουργίας του αισθητήρα BME280	48
5.9	Διάγραμμα λήψης μετρήσεων για τις τρεις παραμέτρους	48
5.10	Βηματική απόκριση του φίλτρου για τις διάφορες δυνατές τιμές της coefficient του φίλτρου.	49
5.11	Memory map του αισθητήρα BME280	50
5.12	Συνδεσμολογία αισθητήρα BME280 με το Raspberry Pi - I2C address 0x77	52
5.13	Συνδεσμολογία δυο αισθητήρων BME280 με το Raspberry Pi	53
5.14	Εντολή ανίχνευσης 'i2cdetect -y 1'- I2C addresses 0x76 - 0x77	54
5.15	Διάγραμμα θερμοκρασίας συναρτήσε του χρόνου	55
5.16	Διάγραμμα πίεσης συναρτήσε του χρόνου	56
5.17	Διάγραμμα υγρασίας συναρτήσε του χρόνου	56
5.18	Αρχική σελίδα	58
5.19	Εμφάνιση δεδομένων από 1 BME280	58
5.20	Εμφάνιση δεδομένων από 2 BME280	59
6.1	Δομή πρωτοκόλλου OPC	61
6.2	Δομή λήψης και διαχείρισης δεδομένων από hardware σε σύστημα SCADA μέσω OPC Server	62
6.3	OPC UA vs Classic OPC	64
6.4	OPC UA Security	65
6.5	OPC UA Server - Object Model	66
6.6	OPC UA Server - Node Model	66
6.7	Ο OPC UA Server στο Raspberry Pi	70
7.1	Δομή DataPoint Type Raspberry_Pi	74
7.2	Δημιουργία νέου Distributed project στο WinCC OA	75
7.3	Επιλογή ονόματος κατά τη δημιουργία νέου project στο WinCC OA	76
7.4	Επιλογή System Name για το νέο project στο WinCC OA	76
7.5	Δήλωση του OPC UA Server στο configuration file	77
7.6	Δημιουργία OPC UA Client Manager	78
7.7	System Manager επιλογή OPC Driver	79
7.8	Configuration του OPC UA Client στο WinCC OA	80
7.9	Configuration Periphery address (DataPoint Element) των στοιχείων του DataPoint	80

7.10 Configuration of OPC UA Client	81
7.11 Configuration of Periphery address of OPC UA	82
7.12 Configuration of subscription for OPC UA Client	83
7.13 Περιεχόμενο OPC UA Server	83
7.14 Επιλογή φακέλου fwInstallation στο οποίο βρίσκεται το fwInstallation.pnl	84
7.15 Εκκίνηση fwInstallation.pnl	85
7.16 Components που εγκαταστάθηκαν	85
7.17 Panel επιλογής αριθμού αισθητήρων	86
7.18 Main Panel για έναν BME280	87
7.19 Main Panel για δυο BME280	87
7.20 Configure Archiving	88
7.21 Archiving parameters for 1 BME280	88
7.22 Archiving parameters for 2 BME280's	89
7.23 Configure Export	90
7.24 Export parameters for 1 BME280	91
7.25 Export parameters for 2 BME280's	92
7.26 Επιλογή της μεταβλητής που θα εμφανιστεί το γράφημα των τιμών της real-time	93
7.27 Live plots	93
7.28 Live plot θερμοκρασίας	94
7.29 Live plot πίεσης	94
7.30 Live plot υγρασίας	94
8.1 Διάγραμμα θερμοκρασίας συναρτήσει του χρόνου κατά τη διάρκεια δύο εβδομάδων 18/5-6/1.	96
8.2 Διάγραμμα πίεσης συναρτήσει του χρόνου κατά τη διάρκεια δύο εβδομάδων 18/5-6/1.	96
8.3 Διάγραμμα υγρασίας συναρτήσει του χρόνου κατά τη διάρκεια δύο εβδομάδων 18/5-6/1.	97
8.4 Διάγραμμα differential θερμοκρασίας συναρτήσει του χρόνου κατά τη διάρκεια δύο εβδομάδων 18/5-6/1.	97
8.5 Διάγραμμα differential πίεσης συναρτήσει του χρόνου κατά τη διάρκεια δύο εβδομάδων 18/5-6/1.	98
8.6 Διάγραμμα differential υγρασίας συναρτήσει του χρόνου κατά τη διάρκεια δύο εβδομάδων 18/5-6/1.	98
8.7 Διάταξη που περιλαμβάνει το Raspberry Pi με τους δύο αισθητήρες BME280.	99
8.8 Διάταξη που περιλαμβάνει το Raspberry Pi, τους δυο αισθητήρες και οθόνη αφής 7".	100

Κεφάλαιο 1

Εισαγωγή

Τίτλος της συγκεκριμένης πτυχακής εργασίας είναι ‘Ανάπτυξη συστήματος λήψης περιβαλλοντικών συνθηκών βασισμένο σε Raspberry Pi και WinCC OA’, στο κεφάλαιο αυτό γίνεται μια μικρή εισαγωγή στο κίνητρο, το οποίο μας ωθεί στον τομέα Πειραματικής Φυσικής Υψηλών Ενεργειών (ΦΥΕ) να στήσουμε συστήματα λήψης, ελέγχου και διαχείρισης περιβαλλοντικών δεδομένων.

Στα πειράματα ΦΥΕ ο αυτόματος έλεγχος και η παρακολούθηση των μεταβολών στις τιμές των περιβαλλοντικών παραμέτρων είναι απαραίτητος εφόσον επηρεάζει τη λειτουργία και κατ’ επέκταση το αποτέλεσμα των ανιχνευτικών διατάξεων. Επομένως, για την άρτια και σωστή διεκπεραίωση των πειραμάτων καθίστανται σημαντικά στοιχεία.

Κατά τη διάρκεια των test beams στο RD51 Collaboration στο CERN, η δυνατότητα λήψης, εμφάνισης και αποθήκευσης των δεδομένων, για τις συνθήκες που επικρατούν στο χώρο, όπου βρίσκονται οι ανιχνευτικές διατάξεις, αλλά από απόσταση είναι καθοριστική. Η άμεση παρατήρηση και ο έλεγχος των μεταβολών στις τιμές της πίεσης και της θερμοκρασίας κυρίως, αποσκοπούν στην πρόληψη, στην αποτροπή, καθώς και στη μετέπειτα ανάλυση της μεταβολής του gain των Gas Chambers, που χρησιμοποιούνται.

Στις εικόνες 1.1 και 1.2, φαίνεται η γραμμική μεταβολή του gain ενός Micromegas Gas Chamber συναρτήσει της πίεσης και της θερμοκρασίας αντίστοιχα. Από τα διαγράμματα αυτά, φαίνεται πως η εξάρτηση του gain με την πίεση είναι αντιστρόφως ανάλογη, ενώ αντίθετα με τη θερμοκρασία είναι ανάλογη, με την αύξηση της θερμοκρασίας δηλαδή αυξάνεται το gain ενός Micromegas. Στην εικόνα 1.3, φαίνεται πως μεταβάλλεται το gain ενός Micromegas Gas Chamber ως προς τη ατμοσφαιρική πίεση συναρτήσει του χρόνου. Είναι εμφανές πως όταν η ατμοσφαιρική πίεση αυξάνεται το gain μειώνεται και αντίστροφα όταν η ατμοσφαιρική πίεση μειώνεται τότε το gain αυξάνεται.

Επομένως, λόγω της εξάρτησης από τις περιβαλλοντικές παραμέτρους, δημιουργείται η ανάγκη ανάπτυξης συστημάτων ελέγχου από απόσταση των τιμών των περιβαλλοντικών παραμέτρων. Στα πλαίσια αυτά δημιουργήθηκε το συγκεκριμένο project, όπου η λήψη των τιμών πραγματοποιείται από τον μικροϋπολογιστή Rasp-

berry Pi και ο έλεγχος των δεδομένων από το πρόγραμμα SCADA WinCC ΟΑ.

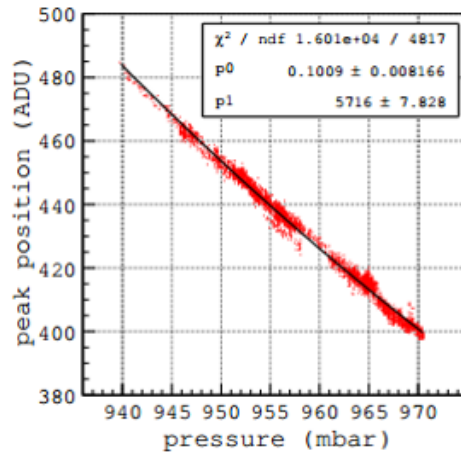


Figure 1.1: Γραμμική μεταβολή του gain ενός Micromegas με την πίεση

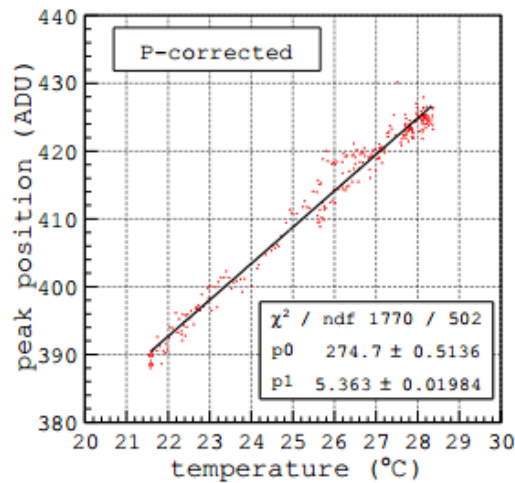


Figure 1.2: Γραμμική μεταβολή του gain ενός Micromegas με τη θερμοκρασία

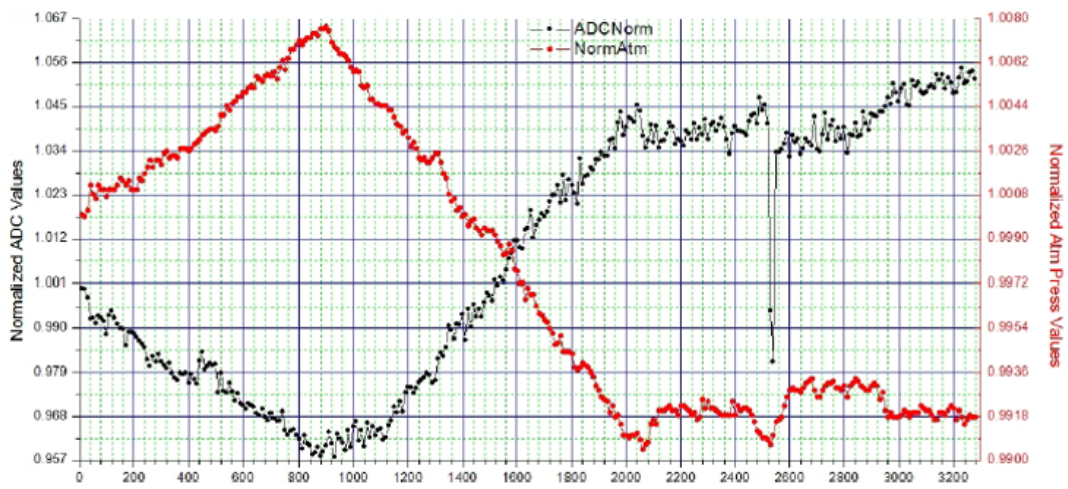


Figure 1.3: Μεταβολή του gain ενός Micromegas και της κανονικοποιημένης ατμοσφαιρικής πίεσης συναρτήσει του χρόνου

Κεφάλαιο 2

Raspberry Pi

Στο κεφάλαιο αυτό θα γίνει αναλυτική περιγραφή του μικροϋπολογιστή Raspberry Pi. Αρχικά, γίνεται μια σύντομη εισαγωγή και στη συνέχεια αναλύονται τα βασικά του χαρακτηριστικά σε τεχνικό επίπεδο (hardware), καθώς και τα προγράμματα λογισμικού που χρησιμοποιήθηκαν (software), όπως το λειτουργικό του σύστημα Raspbian, που επιλέχθηκε, η βασική γλώσσα προγραμματισμού, που χρησιμοποιήθηκε, για την υλοποίηση του project, Python. Τέλος γίνεται αναφορά σε Apache, PHP και HTML που επίσης χρησιμοποιήθηκαν για την αποστολή δεδομένων σε webserver.

2.1 Εισαγωγή

Το Raspberry Pi είναι ένας πλήρης ηλεκτρονικός υπολογιστής μεγέθους πιστωτικής κάρτας, ο οποίος κατασκευάστηκε από τη Raspberry Pi Foundation αρχικά για εκπαιδευτικούς σκοπούς και είναι εμπνευσμένος από το BBC Micro του 1981. Σχεδιάστηκε από μια ομάδα εργαζομένων στο τμήμα Computer Laboratory του πανεπιστημίου του Cambridge, καθώς έβλεπαν πως το ενδιαφέρον και οι γνώσεις των φοιτητών για την πληροφορική είχαν μειωθεί. Η ομάδα αυτή είχε ως στόχο να δημιουργήσει έναν ηλεκτρονικό υπολογιστή χαμηλού κόστους, η χρήση του οποίου από μαθητές δευτεροβάθμιας εκπαίδευσης αλλά και από φοιτητές θα είχε ως αποτέλεσμα την πρακτική εξάσκηση αυτών με τον προγραμματισμό αλλά και την ταυτόχρονη εξοικείωση τους με hardware συστήματα. Χάρη στο μικρό του μέγεθος, τη χαμηλή του τιμή και βασικά τις πολλές δυνατότητες διασύνδεσης και επέκτασης που προσφέρει, έγινε γρήγορα ευρέως αποδεκτό και χρησιμοποιείται από software και hardware μηχανικούς, καθώς και γενικότερα από ανθρώπους που ασχολούνται με projects τα οποία απαιτούν πολλά παραπάνω από έναν απλό μικροελεγκτή, όπως οι συσκευές Arduino.

Λειτουργεί όπως ένας κλασικός ηλεκτρονικός υπολογιστή και ο χρήστης μπορεί να κάνει κάθε ενέργεια που κάνει στο desktop ή το laptop του, όπως για παράδειγμα να πλοηγηθεί στο ίντερνετ, να παρακολουθήσει βίντεο υψηλής ανάλυσης, να χρησιμοποιήσει κειμενογράφο και να προγραμματίσει κώδικα.

Στις πλακέτες του μικροϋπολογιστή, χρησιμοποιείται η μέθοδος **SoC** (System on a Chip), βάσει της οποίας όλα τα απαραίτητα ηλεκτρονικά για τη λειτουργία του, τοποθετούνται συμπιεσμένα σε ένα πολύ μικρό πακέτο, σε ένα και μόνο chip, αντίθετα με τους κοινούς υπολογιστές οι οποίοι έχουν ξεχωριστά chip για CPU, GPU, USB controller, RAM κλπ. Με τον τρόπο αυτό η υπολογιστική ισχύς συγκεντρώνεται σε τόσο μικρό μέγεθος και με τόσο χαμηλό κόστος. Αυτό αποτελεί και το πιο εντυπωσιακό στοιχείο του.

Βασιζόμενοι στις δυνατότητες και τα χαρακτηριστικά του, οι εφαρμογές του Raspberry Pi είναι απεριόριστες. Κατ' αρχάς, η διασύνδεσή του με οθόνη, πληκτρολόγιο και ποντίκι, παρέχει έναν πλήρη ηλεκτρονικό υπολογιστή, ο οποίος υποστηρίζει διάφορα λειτουργικά συστήματα. Τα βασικότερα από αυτά είναι αρκετές διανομές Linux με προτεινόμενη το OS Raspbian, καθώς και τα Ubuntu MATE, Snappy Ubuntu και άλλα. Επίσης, υποστηρίζει και άλλα και διανομές άλλων λειτουργικών συστημάτων όπως για παράδειγμα του RISC(RISC OS PI) αλλά και μια έκδοση των Windows 10 (Windows 10 IoT Core) η οποία αναπτύχθηκε ειδικά για χρήση στον μικροϋπολογιστή Raspberry Pi. Το λειτουργικό σύστημα, που χρησιμοποιήθηκε, είναι το Raspbian, το οποίο αποθηκεύεται σε micro SD card.

Μια από τις πιο διαδεδομένες πρακτικές εφαρμογές του Raspberry Pi είναι ως Media Center PC, όπως και για gaming, για παράδειγμα όσοι παίζουν Minecraft, μπορούν να χρησιμοποιούν το Raspberry Pi αποκλειστικά για αυτό, με την ειδική έκδοση που έχει κυκλοφορήσει Minecraft Pi edition.

Επίσης, συνδέοντας έναν εξωτερικό σκληρό δίσκο πάνω, ο χρήστης μπορεί να μετατρέψει το Raspberry Pi σε ένα αποκλειστικό σύστημα για κατέβασμα Torrent χωρίς να σπαταλά πόρους του προσωπικού του ηλεκτρονικού υπολογιστή. Ο χρήστης έχει τη δυνατότητα ταυτόχρονα με το σκληρό δίσκο, να διαμορφώσει ένα προσωπικό Cloud service. Με αυτόν τον τρόπο, είναι σε θέση να έχει πρόσβαση στα αρχεία του μέσω ίντερνετ, από οπουδήποτε στον κόσμο, παρακάμπτοντας υπηρεσίες όπως το Dropbox ή το Google Drive. Αξίζει να σημειωθεί, πως είναι απόλυτα εφικτό, να στηθεί ένας οικιακός Web Server, όπου ο χρήστης θα έχει τη δυνατότητα να ανεβάζει τις προσωπικές του σελίδες. Συνδυάζοντας τις κατάλληλες περιφερειακές συσκευές (πχ αισθητήρες) και προγραμματίζοντας ο χρήστης μπορεί να διεκπαιρεώσει απεριόριστες εφαρμογές χρησιμοποιώντας τον μικροϋπολογιστή Raspberry Pi.

Μια πολύ διαδεδομένη πλέον χρήση του μικροϋπολογιστή είναι η κατασκευή **clusters**. Ο χρήστης συνδέοντας από δύο και πάνω Raspberry Pi's μεταξύ τους εκμεταλεύεται την υπολογιστική τους ισχύ κατασκευάζοντας ένα low cost cluster. Υπάρχουν παραδείγματα στα οποία χρήστες έχουν συνδέσει πολύ περισσότερα, έως και 96 για το heavy lifting των Big Data. Ως computer cluster ορίζεται ένα σύμπλεγμα υπολογιστών, που αποτελείται από μια σειρά συνδεδεμένων υπολογιστών, οι οποίοι λειτουργούν μαζί, έτσι ώστε να μπορούν να θεωρούνται ως ένα ενιαίο σύστημα. Τα παράλληλα προγράμματα που λειτουργούν σε έναν από αυτούς τους υπολογιστές χρησιμοποιούν την υπολογιστική ισχύ όλων των υπολογιστών του cluster.

2.1.1 Αναδρομή στα μοντέλα Raspberry Pi

Από το 2012, που κυκλοφόρησε η πρώτη γενιά Raspberry Pi, έχουν κυκλοφορήσει πολλά μοντέλα, οι τιμές των οποίων κυμαίνονται στα \$30-35. Οι βασικές τους διαφορές είναι στον επεξεργαστή, την εσωτερική τους μνήμη RAM, τα pins εισόδου και εξόδου (I/O), τις ικανότητες γενικής εξωτερικής διασύνδεσης που προσφέρουν κ.α. Πιο αναλυτικά, το 2009 με τη συνεργασία των Pete Lomas και David Braben δημιουργήθηκε το φιλανθρωπικό ίδρυμα Raspberry Pi Foundation και τρία χρόνια αργότερα κυκλοφόρησε το πρώτο Raspberry Pi, με τα Model A, Model A+ και Model B. Τα μοντέλα αυτά διέθεταν επεξεργαστή ARMv6k στα 70 MHz, 256MB RAM, κάρτα γραφικών Broadcom VideoCore IV, και καταναλωτή 1 έως 3.5 watt, ενώ η αποθήκευση των δεδομένων γινόταν σε κάρτες SD, SDHC και MicroSD. Το Raspberry Pi Model B, μέσα σε 2 χρόνια από τη στιγμή που βγήκε στη μαζική παραγωγή πούλησε πάνω από δυο εκατομμύρια κομμάτια.

Στη συνέχεια κυκλοφόρησαν οι εκδόσεις Model B rev 2 και Model B+ με 512MB RAM. Τον Οκτώβρη του 2014 οι συνολικές πωλήσεις προσέγγιζαν τα 4 εκατομμύρια. Το Φεβρουάριο του 2015 κυκλοφόρησε το Raspberry Pi Generation 2 Model B, το οποίο είχε RAM 1 GB, έξι φορές πιο ισχυρό και γρήγορο επεξεργαστή με τον τετραπύρηννο Cortex-A7 (ARMv7) και διπύρηννη κάρτα γραφικών Broadcom videoCoreIV. Το Νοέμβριο του επόμενου χρόνου κυκλοφόρησε η μέχρι τότε πιο μικρή έκδοση, το Raspberry Pi Zero, το οποίο είχε το μισό μέγεθος από το συνηθισμένο Raspberry Pi, με 512MB RAM και επεξεργαστή ARM1176JF-S στα 1000 MHz, το οποίο κόστιζε μόλις \$5.

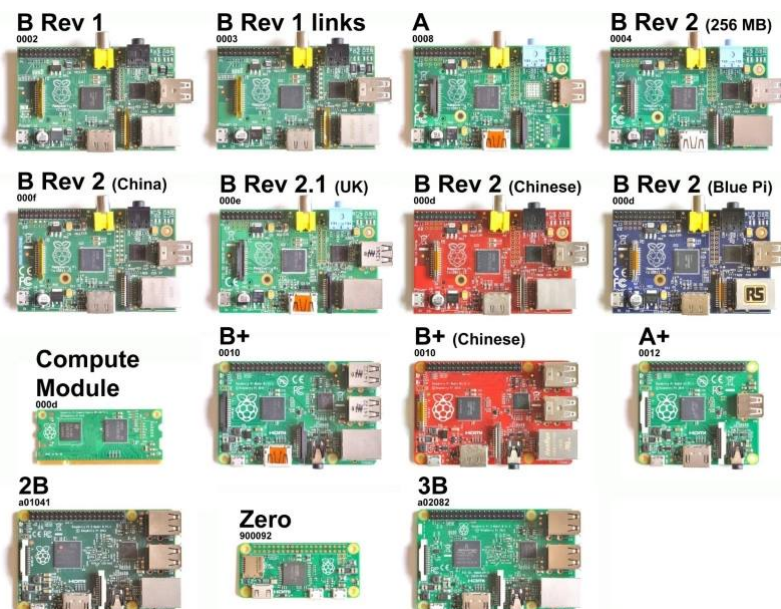


Figure 2.1: Μοντέλα και γενιές Raspberry Pi

Το Raspberry Pi Generation 3 Model B, που χρησιμοποιήθηκε για την υλοποίηση του project της παρούσας διπλωματικής εργασίας, κυκλοφόρησε στις αρχές του 2016. Ήρθε εξοπλισμένο με ακόμα ταχύτερο επεξεργαστή ARM Cortex-A53 στα 1200MHz, 1GB RAM, ενσωματωμένο 802.11n Wireless LAN adapter και κάρτα γραφικών Broadcom VideoCore IV χρονισμένη στα 250MHz, συχνότητα υψηλότερη από κάθε προηγούμενη γενιά, ενώ η μέση κατανάλωση ενέργειας του μοντέλου αυτού είναι στα 4W.

Τέλος, αξίζει τέλος να σημειωθεί πως το Φεβρουάριο του 2017 κυκλοφόρησε το Raspberry Pi Zero W το οποίο είναι παρόμοιο με το Raspberry Pi Zero, παρέχει όμως τη δυνατότητα σύνδεσης με Wi-Fi και Bluetooth στοιχίζοντας μόνο \$10.

2.2 Raspberry Pi 3 Model B+

Όπως αναφέρθηκε, για τη διεκπαιρέωση της παρούσας πτυχιακής εργασίας χρησιμοποιήθηκε το Raspberry Pi 3 Model B+. Το μοντέλο αυτό κυκλοφόρησε το Φεβρουάριο του 2016 και αποτελεί την τρίτη γενιά Raspberry Pi. Έχει διαστάσεις 85x56x17 mm και κοστίζει \$35.

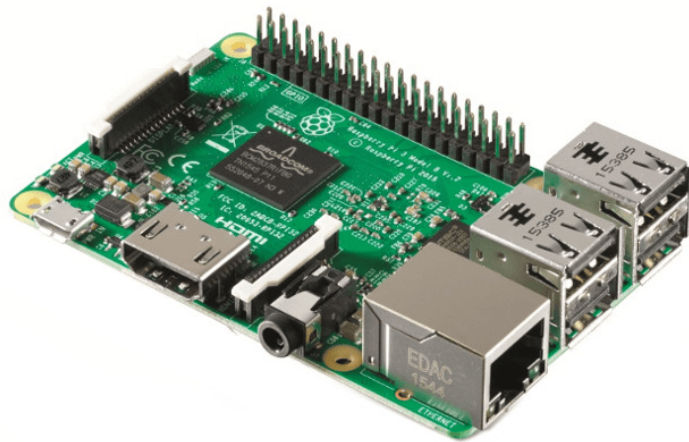


Figure 2.2: Microcomputer Raspberry Pi 3 Model B+

2.2.1 Γενικά Τεχνικά Χαρακτηριστικά

Οι τεχνικές προδιαγραφές του συγκεκριμένου μοντέλου είναι οι εξής:

- επεξεργαστής Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53

- 802.11 b/g/n Wireless LAN
- Bluetooth 4.1 (Bluetooth Classic and Low Energy)
- GPU: Dual Core VideoCore IV[®] Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated OpenVG
- RAM 1GB LPDDR2

Επίσης, όπως φαίνεται και στην εικόνα 2.3 έχει διάφορες θύρες επέκτασης:

- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot
- VideoCore IV 3D graphics core

Τέλος, όσον αφορά την τροφοδοσία και την κατανάλωση ενέργειας, το Raspberry Pi για τη λειτουργία του χρειάζεται τροφοδοσία 5V. Ο πυρήνας του επεξεργαστή χρειάζεται 1,2V, τα οποία παίρνει από ενσωματωμένο διακόπτη στον αντίστοιχο σύνδεσμο του επεξεργαστή Broadcom BCM2837. Με αυτόν τον τρόπο, αποτρέπονται οι καταστροφικές συνέπειες που θα επέφερε στην πλακέτα τροφοδοσία μεγαλύτερη των 5V, καθώς η τροφοδοσία παρέχεται κατευθείαν στις HDMI και USB θύρες. Επίσης, διαθέτει δίοδο προστασίας και ημιαγωγικό διακόπτη. Η κατανάλωση του Raspberry Pi διαφέρει ανάλογα με τα μοντέλα και τις λειτουργίες που εκτελεί κάθε φορά. Η μέση τιμή της κατανάλωσης είναι περίπου 3-4W στα 5V τροφοδοσίας. Η ανεπάρκεια στην τροφοδοσία, που είναι δυνατόν να προκύψει, οφείλεται σε διάφορους λόγους που έχουν να κάνουν με τις λειτουργίες και τα περιφερειακά που χρησιμοποιεί. Οι βασικότερες πηγές κατανάλωσης είναι η χρήση γραφικού περιβάλλοντος (GUI), USB συσκευές και η σύνδεση Ethernet και τέλος η GPU.

2.2.2 GPIO (General Purpose Input/Output)

Ένα πολύ σημαντικό χαρακτηριστικό του Raspberry Pi είναι η σειρά από τα GPIO pins που βρίσκονται στην άκρη της πλακέτας. Αυτά τα pins αποτελούν το μέσο διασύνδεσης του Raspberry Pi με τον 'έξω κόσμο', καθώς μέσω αυτών των εισόδων/

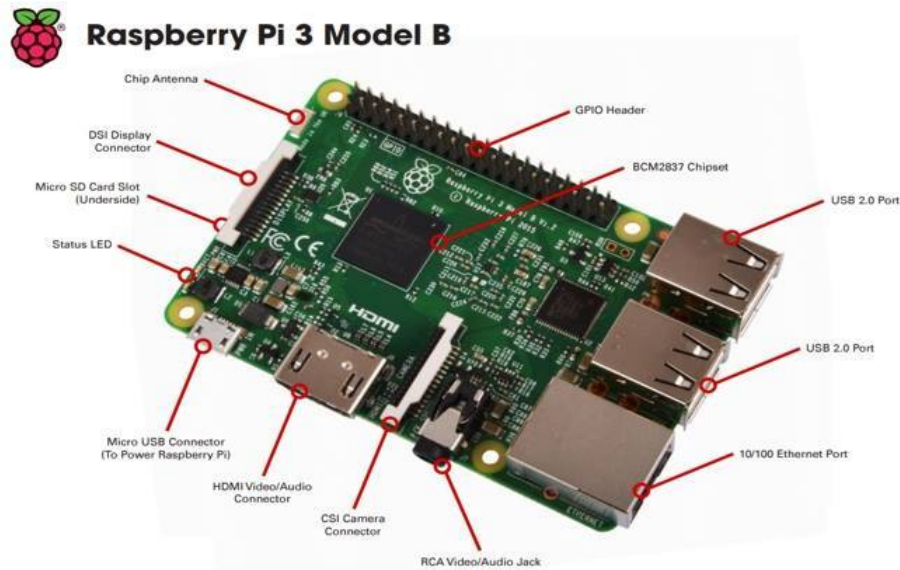


Figure 2.3: Υποδοχές του Raspberry Pi 3 Model B+

εξόδων μπορούν να εκτελεστούν πάρα πολλά projects, όπως η λήψη περιβαλλοντολογικών παραμέτρων από έναν ή περισσότερους digital αισθητήρες. Από τα 40 συνολικά pins τα 26 είναι digital GPIO pins και τα υπόλοιπα είναι 4 pins τροφοδοσίας, 8 pins γείωσης και 2 pins ID EEPROM. Υπάρχουν 2 τρόποι αρίθμησης των pins :

- GPIO numbering: όπως βλέπει ο υπολογιστής τα GPIO pins
- Physical numbering: ξεκινώντας την αρίθμηση των pins από το πιο κοντινό στην SD card

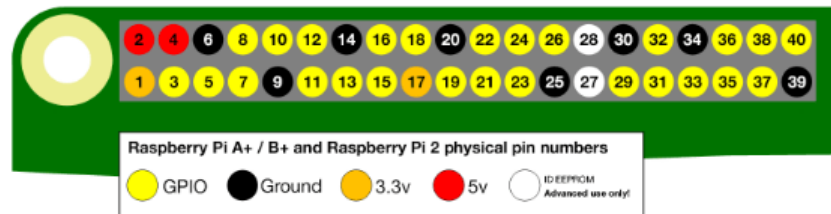


Figure 2.4: General Pins Input/Output (GPIO)

2.3 Προεσκόπηση λογισμικού (Software) του Raspberry Pi

Με τον όρο λογισμικό (Software) ορίζεται η συλλογή προγραμμάτων των ηλεκτρονικών υπολογιστών, διαδικασιών και οδηγιών, που εκτελούν ορισμένες εργασίες σε ένα υπολογιστικό σύστημα. Στην ενότητα αυτή γίνεται αναφορά στο Software του Raspberry Pi, στο λειτουργικό του σύστημα (Raspbian) και στη γλώσσα προγραμματισμού Python, με την οποία υλοποιήθηκε ο προγραμματισμός του project και ο OPC Server, ο οποίος θα αναλυθεί σε επόμενο κεφάλαιο. Τέλος, γίνεται αναφορά σε Apache, PHP και HTML που χρησιμοποιήθηκαν για τη δημιουργία απλής ιστοσελίδας, στην οποία θα εμφανίζονται δεδομένα που συλλέγει το Raspberry Pi από τους αισθητήρες.

2.3.1 Operating System (OS) Raspbian

Ως λειτουργικό σύστημα (Operating System (OS)) ενός υπολογιστή ορίζεται ένα σύνολο βασικών προγραμμάτων το οποίο ελέγχει τη συνολική λειτουργία του, χρησιμοποιείται ως υπόβαθρο, για την εκτέλεση όλων των υπόλοιπων προγραμμάτων και είναι υπεύθυνο για τη διαχείριση και το συντονισμό των εργασιών. Επίσης, χρησιμοποιείται για τη διαχείριση των περιφερειακών συσκευών και την εξασφάλιση επικοινωνίας μεταξύ του χρήστη και του υπολογιστή. Αποτελείται από ένα σύνολο μηχανισμών μέσω των οποίων επιτυγχάνεται αυτόματη διαχείριση των πόρων ενός υπολογιστή και ελεγχόμενη κατανομή τους στις εκτελούμενες εφαρμογές, έτσι ώστε οι τελευταίες να είναι σε θέση να προσπελάσουν εύκολα τους πόρους και τις συσκευές του συστήματος, χωρίς να χρειάζεται να γνωρίζουν με ακρίβεια τη δομή του υποκείμενου υλικού, ώστε να μπορούν να εκτελούνται ταυτόχρονα πολλαπλές εφαρμογές, χωρίς να έρχονται σε διένεξη μεταξύ τους ή με τον υπολογιστή.

Το **Raspbian** είναι το προτεινόμενο και πιο διαδεδομένο OS του Raspberry Pi. Είναι ένα ελεύθερο λειτουργικό σύστημα βασισμένο στη διανομή Debian των Linux. Κυκλοφόρησε τον Ιούνιο του 2012 και παραμένει σε συνεχή εξέλιξη, καθώς αποτελεί ένα συνολικό open source project. Έρχεται με πάνω από 35.000 πακέτα, τα οποία είναι ήδη μεταγλωττισμένα και σε έτοιμο format 'εικόνας' για εγκατάσταση στο Raspberry Pi (Raspbian Image). Η εικόνα Raspbian είναι ένα αρχείο στο οποίο είναι αποθηκευμένος ο install manager του λειτουργικού και εγγράφεται σε micro SD card, η οποία στη συνέχεια χρησιμοποιείται για την εκκίνηση (boot) και τη μετέπειτα λειτουργία του Raspberry Pi.

2.3.2 Γλώσσα προγραμματισμού Python

Το Raspberry Pi παρέχει τη δυνατότητα προγραμματισμού σε οποιαδήποτε γλώσσα προγραμματισμού μπορεί να μεταγλωττίσει ο ARM μικροεπεξεργαστής που διαθέτει. Για την υλοποίηση αυτού του project ο προγραμματισμός έγινε σε Python και πιο συγκεκριμένα χρησιμοποιήθηκε η έκδοση 2.7.

Η Python αποτελεί μια υψηλού επιπέδου γλώσσα προγραμματισμού, η οποία δημιουργήθηκε το 1990 από τον Ολλανδό Guido van Rossum. Παρότι είναι μια πολύ ισχυρή γλώσσα, είναι εύκολη στο να γράψει και να διαβάσει ο χρήστης το συντακτικό του κώδικα. Τέλος, επιτρέπει στους προγραμματιστές να προγραμματίσουν σε λιγότερες γραμμές κώδικα από ότι θα χρειαζόταν σε άλλες γλώσσες όπως η Java ή η C++. Αρχικά ξεκίνησε ως γλώσσα για scripts σε Linux. Τα προγράμματα που είναι γραμμένα σε Python μοιάζουν με shell scripts, καθώς τα αρχεία περιέχουν εντολές, τις οποίες ο υπολογιστής εκτελεί απο πάνω προς τα κάτω.

Επίσης, η Python διαθέτει μια τεράστια συλλογή βιβλιοθηκών και αρκετές υλοποιημένες συναρτήσεις, οι οποίες επιταχύνουν τη διαδικασία προγραμματισμού. Υπάρχουν βιβλιοθήκες για σχεδόν ό,τι μπορεί να χρειαστεί ο χρήστης, όπως για web frameworks, game programming, rendering graphics, GUI interfaces, scientific computing κλπ. Όσον αφορά το συντακτικό, ένα ιδιαίτερο χαρακτηριστικό της γλώσσας, που αξίζει να σημειωθεί, καθώς τη διαφοροποιεί από άλλες, είναι η χρήση κενών διαστημάτων (tabs) για τον διαχωρισμό των συντακτικών δομών του προγράμματος, σε αντίθεση με το συντακτικό που ακολουθείται σε άλλες γλώσσες, όπου για τον ίδιο σκοπό χρησιμοποιούνται ειδικά σύμβολα (πχ αγκύλες).

Στο λειτουργικό σύστημα Raspbian είναι ήδη προεγκατεστημένες δυο εκδόσεις της Python, Python 2 και Python 3, όπως επίσης και το περιβάλλον ανάπτυξης (editor) IDLE.

2.3.3 Apache HTTP

Ο Apache HTTP ή απλά Apache είναι ένα διακομιστής-εξυπηρετητής (server) του παγκόσμιου ιστού (web). Αποτελεί έναν από τους σημαντικότερους web servers, καθώς λειτουργεί σε διάφορες πλατφόρμες (Windows, Linux και Mac OS) και είναι λογισμικό ανοιχτού κώδικα (open source).

Κάθε φορά, που ένας χρήστης του διαδικτύου επισκέπτεται έναν ιστότοπο το πρόγραμμα πλοήγησης (browser) επικοινωνεί με έναν server μέσω του πρωτοκόλλου επικοινωνίας HTTP, ο οποίος παράγει τις ιστοσελίδες και τις αποστέλλει πίσω στον browser. Το πρωτόκολλο επικοινωνίας HTTP (HyperText Transfer Protocol) αποτελεί το κύριο πρωτόκολλο που χρησιμοποιείται απο τους browsers του Παγκόσμιου Ιστού (World Wide Web, **www**) για τη μεταφορά δεδομένων ανάμεσα σε ένα διακομιστή (server) και έναν πελάτη (client)

Client είναι ο browser, ενώ μια εφαρμογή που εκτελείται σε έναν υπολογιστή που φιλοξενεί έναν ιστότοπο είναι ο server. Ο client υποβάλλει ένα μήνυμα αίτησης βάσει του πρωτοκόλλου HTTP στο server, ο οποίος είτε παρέχει πόρους, όπως αρχεία HTML και άλλο περιεχόμενο ή εκτελεί άλλες λειτουργίες για λογαριασμό του client και επιστρέφει ένα μήνυμα απάντησης στον client. Η απάντηση περιέχει πληροφορίες για την κατάσταση ολοκλήρωσης σχετικά με το αίτημα ή το περιεχόμενο που ζητήθηκε στο μήνυμα.

Αν και το HTTP πρωτόκολλο σχεδιάστηκε για χρήση στον Ιστό, υποστηρίζει λειτουργίες που είναι πιο γενικές απ' ότι απαιτείται. Οι λειτουργίες αυτές ονομάζονται

2.3. ΠΡΟΕΣΚΟΠΙΣΗ ΛΟΓΙΣΜΙΚΟΥ (SOFTWARE) ΤΟΥ RASPBERRY PI 17

μέθοδοι. Κάθε μήνυμα αίτησης αποτελείται από μία ή περισσότερες γραμμές κειμένου ASCII. Η πρώτη λέξη της πρώτης γραμμής της αίτησης είναι το όνομα της ζητούμενης μεθόδου. Στη συνέχεια αναφέρονται επιγραμματικά οι ενσωματωμένες μέθοδοι αίτησης του πρωτοκόλλου HTTP:

- **GET:** Η μέθοδος GET ζητά από το διακομιστή να στείλει τη σελίδα και τη διαβάσει. Η πιο συνηθής μορφή της μεθόδου GET είναι η εξής: 'GET filename HTTP,' όπου το 'filename' προσδιορίζει το όνομα του πόρου που πρέπει να προσκομιστεί.
- **HEAD:** Η μέθοδος HEAD ζητά μόνο την κεφαλίδα του μηνύματος, χωρίς τη πραγματική σελίδα. Η μέθοδος αυτή μπορεί να χρησιμοποιηθεί για τη συλλογή πληροφοριών, ή απλώς και μόνο για τον έλεγχο εγκυρότητας μιας διεύθυνσης URL.
- **POST:** Η μέθοδος POST χρησιμοποιείται κατά την υποβολή φορμών. Όπως και η μέθοδος GET, η POST περιέχει μια διεύθυνση URL αλλά αντί να ανακτά τη σελίδα μεταφέρει δεδομένα στον διακομιστή, ο οποίος τα επεξεργάζεται ανάλογα με το URL.
- **PUT:** Η μέθοδος PUT είναι η αντίστροφη της GET, δηλαδή αντί να διαβάσει τη σελίδα, τη γράφει.
- **DELETE:** Η μέθοδος DELETE καταργεί τη σελίδα
- **TRACE:** Η μέθοδος TRACE χρησιμοποιείται για αποσφαλμάτωση. Η μέθοδος αυτή είναι χρήσιμη, όταν η επεξεργασία των αιτήσεων δεν γίνεται σωστά και ο πελάτης θέλει να δει ποια αίτηση έλαβε πραγματικά ο διακομιστής.
- **CONNECT:** Η μέθοδος CONNECT επιτρέπει στον χρήστη να πραγματοποιήσει σύνδεση με έναν διακομιστή Ιστού μέσω μιας ενδιάμεσης συσκευής.
- **OPTIONS:** Η μέθοδος OPTIONS παρέχει έναν τρόπο, ώστε ο πελάτης να στέλνει ερωτήματα στον διακομιστή σχετικά με μια σελίδα και να λαμβάνει τις μεθόδους και τις κεφαλίδες που μπορούν να χρησιμοποιηθούν.

2.3.3.1 PHP και HTML

Οι γλώσσες PHP και HTML αλληλεπιδρούν μεταξύ τους: η PHP μπορεί να παράξει HTML, και η HTML μπορεί να μεταφέρει πληροφορίες στην PHP.

Η **PHP** (PHP : Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP επεξεργάζεται από έναν συμβατό web server, όπως ο Apache, ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο. Το περιεχόμενο αυτό, είτε θα σταλεί στον web browser σε μορφή κώδικα **HTML**, ή θα επεξεργαστεί τις εισόδους, που του δόθηκαν, χωρίς να προβάλλει έξοδο στον χρήστη, αλλά θα τις μεταβιβάσει σε κάποιο άλλο PHP script.

Η PHP αποτελεί μια από τις πιο διαδεδομένες τεχνολογίες του World Wide Web και χρησιμοποιείται από πληθώρα εφαρμογών και ιστοτόπων. Το γεγονός αυτό αποτελεί απόρροια της ευκολίας που παρουσιάζει ο προγραμματισμός με αυτή, αλλά και στο γεγονός πως είναι μια γλώσσα, η οποία βρίσκεται σχεδόν σε κάθε διακομιστή.

Από την άλλη, η **HTML** (HyperText Markup Language) αποτελεί την κύρια γλώσσα σήμανσης για τις ιστοσελίδες και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Η HTML δεν είναι γλώσσα προγραμματισμού. Δηλαδή, μέσα από την HTML δεν μπορούμε να κατασκευάσουμε προγράμματα (ακόμα και η αλληλεπίδραση client-server πολλές φορές καθίσταται δύσκολη), αλλά μόνο να δώσουμε στον web browser τις οδηγίες για το τι θα εμφανίζει στο χρήστη.

Κεφάλαιο 3

Εγκατάσταση και Ρυθμίσεις Raspberry Pi (Setup)

Το Raspberry Pi αποτελεί, όπως αναφέρθηκε και αναλύθηκε στο Κεφάλαιο 2 έναν ολοκληρωμένο ηλεκτρονικό υπολογιστή μεγέθους πιστωτικής κάρτας και επομένως απαιτείται ένα set up πριν τη χρήση του. Στο κεφάλαιο αυτό, περιγράφεται αυτή η διαδικασία, αρχικά η εγγραφή της εικόνας Raspbian σε κάρτα micro SD 32 Mb και στη συνέχεια η ενεργοποίηση και διαμόρφωση των βασικών λειτουργιών του. Επίσης, περιγράφεται πως είναι δυνατή η έμμεση πρόσβαση και η διαχείριση του Raspberry Pi από ηλεκτρονικό υπολογιστή με OS Windows μέσω Putty και VNC Server.

3.1 Εγγραφή εικόνας Raspbian

Το πρώτο βήμα για την εκκίνηση της λειτουργίας του Raspberry Pi αποτελεί η εγγραφή της εικόνας Raspbian σε micro SD card. Η Raspberry Pi Foundation έχει παράξει και κυκλοφορήσει τις δικές της προτεινόμενες εικόνες Raspbian τις οποίες μπορεί ο χρήστης να κατεβάσει από την επίσημη ιστοσελίδα, του [rasbian.org](http://www.raspbian.org) `"http://www.raspbian.org/RaspbianImage"`. Επίσης, υπάρχουν στην αγορά έτοιμες SD cards οι οποίες είναι έτοιμες για χρήση με εγγεγραμμένη ήδη την εικόνα Raspbian. Αυτές, ο χρήστης τις βρίσκει στην ιστοσελίδα: `'Raspbian Preinstalled SD cards'`. Στο project της παρούσας εργασίας ακολουθήθηκε η εξής διαδικασία εγγραφής της εικόνας Raspbian σε micro SD card 32GB.

Αρχικά, πριν την εγγραφή της εικόνας πρέπει να διασφαλιστεί το γεγονός πως η micro SD κάρτα είναι πλήρως κενή και δεν περιέχει τμήματα τα οποία αποκόπτον μέρος της χωρητικότητας της, έτσι ώστε η επικείμενη εγγραφή της εικόνας να πραγματοποιηθεί σωστά και επιτυχημένα. Η μέγιστη δυνατή χωρητικότητα αποδίδεται μετά από εφαρμογή του προγράμματος **'SD Formatter'**, το οποίο αποτελεί ένα εργαλείο, που λαμβάνεται από την επίσημη ιστοσελίδα του SD Formatter. Οι εικόνες 3.1, 3.2 και 3.3 που ακολουθούν, παρουσιάζουν τη διαδικασία Format SD card 4GB, όχι των 32GB που χρησιμοποιήθηκε.

20ΚΕΦΑΛΑΙΟ 3. ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΡΥΘΜΙΣΕΙΣ RASPBERRY PI (SETUP)

Μετά την εγκατάστασή του, γίνεται εισαγωγή της SD card στον υπολογιστή. Ακολουθεί η εκτέλεση της εφαρμογής, όπου ανοίγει το παράθυρο της εικόνας 3.1 , στο σημείο αυτό ο χρήστης πρέπει να προσέξει το Drive, που θα επιλέξει να κάνει το format. Στη συνέχεια πρέπει να γίνει ενεργοποίηση της προσαρμογής μεγέθους, στο πεδίο 'format size adjustment' επιλέγει 'ON', όπως φαίνεται στην εικόνα 3.2.

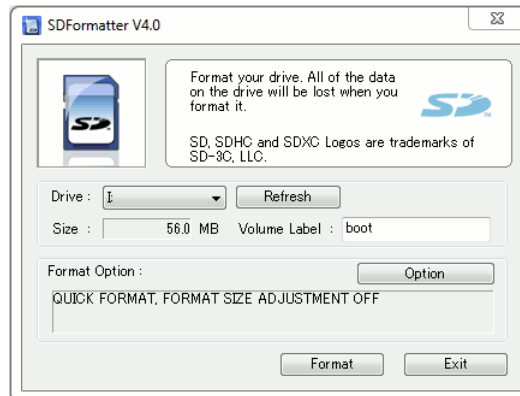


Figure 3.1: Εκτέλεση της εφαρμογής SD Formatter. Ο χρήστης πρέπει να επιλέξει το σωστό Drive στο οποίο θα κάνει format.

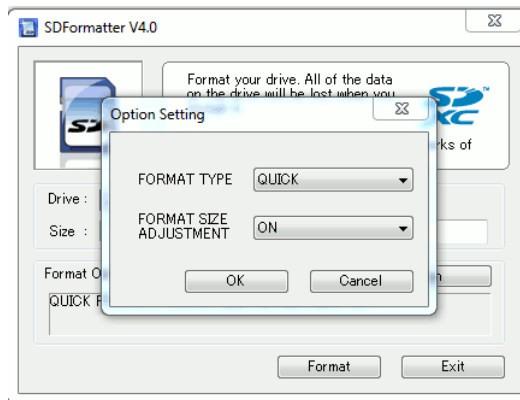


Figure 3.2: Στη συνέχεια ο χρήστης ενεργοποιεί την προσαρμογή μεγέθους, στο πεδίο 'format size adjustment' επιλέγει 'ON'

Πατώντας το κουμπί 'Format' ξεκινά η διαδικασία, η οποία διαρκεί μερικά δευτερόλεπτα. Όταν τελειώσει, εμφανίζεται το παράθυρο της εικόνας 3.3, το οποίο παρέχει τις πληροφορίες για τη χωρητικότητα της micro SD card, που ενδιαφέρουν το χρήστη.

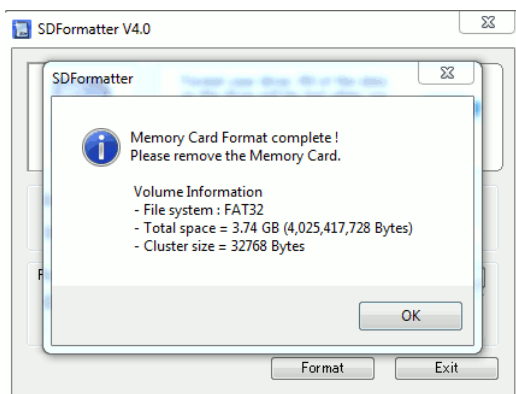


Figure 3.3: Τέλος διαδικασίας, στο παράθυρο που αναδύεται εμφανίζονται οι πληροφορίες για τη χωρητικότητα της micro SD card, που ενδιαφέρουν το χρήστη.

Αφού ολοκληρωθεί η προετοιμασία της micro SD, ο χρήστης κατεβάζει την εικόνα Raspbian από την επίσημη ιστοσελίδα 'raspbian.org' σε ένα αρχείο .zip, το οποίο στη συνέχεια κάνει extract. Στα αρχεία που περιέχει βρίσκεται και ένα αρχείο τύπου .img, το οποίο θα εγγραφεί στην κάρτα micro SD με το πρόγραμμα Etcher. Η διαδικασία εγγραφής του αρχείου περιγράφεται από τις εικόνες 3.4 και 3.5, που ακολουθούν.

Αρχικά, γίνεται επιλογή του κατάλληλου αρχείου εικόνας και του Drive στον οποίο θα γίνει η εγγραφή της. Επομένως, γίνεται επιλογή του αρχείου τύπου .img και της micro SD card. Τέλος, ο χρήστης επιλέγει 'Flash Image'.

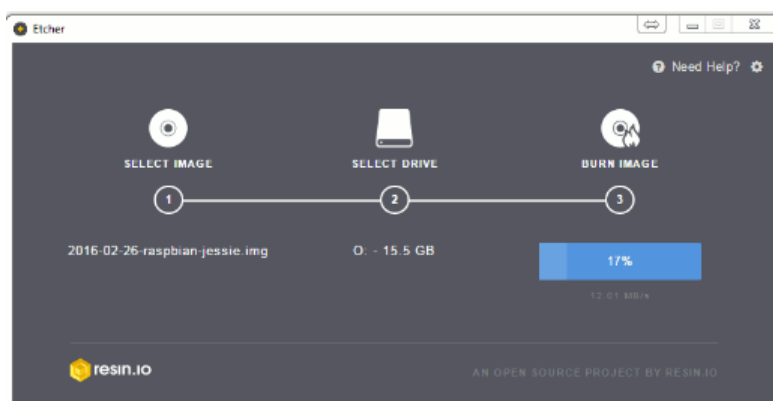


Figure 3.4: Περιβάλλον εφαρμογής Etcher. Ο χρήστης επιλέγει το αρχείο .img της εικόνας Raspbian και η micro SD card, που αποτελεί το Drive στο οποίο θα πραγματοποιηθεί η εγγραφή.

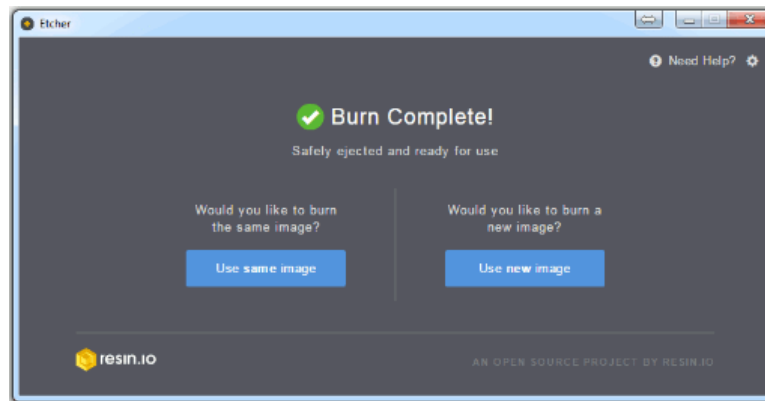


Figure 3.5: Ολοκλήρωση της εγγραφής της εικόνας Raspbian

3.2 Εκκίνηση Raspberry Pi

Αφού ολοκληρωθεί η διαδικασία εγγραφής της εικόνας Raspbian, όπως περιγράφηκε και εισαχθεί η micro SD card στην κατάλληλη υποδοχή του Raspberry Pi (εικόνα 2.3), ακολουθεί η εκκίνηση του Raspberry Pi 3(boot). Πρώτο βήμα είναι η τροφοδοσία του με micro USB που παρέχει τουλάχιστον 2 A (για τις παλαιότερες εκδόσεις απαιτείται τροφοδοσία 0,7A) στα 5V και στη συνέχεια σύνδεση του μέσω της θύρας Ethernet στο τοπικό δίκτυο είτε απευθείας στο router ή μέσω του προσωπικού υπολογιστή του χρήστη.

3.2.1 Έμμεση πρόσβαση και διαχείριση της λειτουργίας από Windows μέσω Secure Shell (SSH)

Για την πρώτη εκκίνηση, αλλά και για τη μετέπειτα χρήση του Raspberry Pi χωρίς τη χρήση οθόνης και πληκτρολογίου, αλλά μέσω υπολογιστή με λογισμικό Windows ακολουθείται η διαδικασία που περιγράφεται στη συνέχεια.

Απαραίτητα εργαλεία είναι τα προγράμματα: Advanced IP Scanner με το οποίο γίνεται ανίχνευση της διεύθυνσης IP που λαμβάνει το Raspberry Pi, καθώς και το πρόγραμμα PuTTY. Το τελευταίο αποτελεί SSH client.

Ο **Secure Shell (SSH) server** αποτελεί βασικό component των Linux, το οποίο επιτρέπει στο χρήστη να διαχειριστεί ουσιαστικά τη γραμμή εντολών (terminal) άλλου υπολογιστή από τον υπολογιστή του, χρησιμοποιώντας έναν SSH client, όπως για παράδειγμα το πρόγραμμα PuTTY. Το πρόγραμμα αυτό εδραιώνει την επικοινωνία μεταξύ του Raspberry Pi με τον άλλον υπολογιστή, στον οποίο είναι εγκατεστημένο. Το μόνο που χρειάζεται είναι η διεύθυνση IP του Raspberry, για να την εισχωρήσει στο PuTTY. Αξίζει να σημειωθεί πως η επικοινωνία πραγματοποιείται μόνο όταν το Raspberry Pi και ο υπολογιστής του χρήστη βρίσκονται συνδεδεμένοι στο ίδιο τοπικό δίκτυο.

Στις πιο πρόσφατες εκδόσεις του Raspbian η πρόσβαση στο SSH γίνεται

αυτόματα εξ' ορισμού, ενώ στις παλαιότερες έπρεπε να γίνει πρώτα το configuration του SSH χρησιμοποιώντας το Raspi Config. Το χαρακτηριστικό αυτό, διευκολύνει την πρώτη εκκίνηση του Raspberry Pi, καθώς επιτρέπει στο χρήστη να το εκκινήσει για πρώτη φορά με τη διαδικασία που ακολουθεί, χωρίς να το συνδέσει σε οθόνη και πληκτρολόγιο.

Αφού ο χρήστης συνδέσει το Raspberry Pi με το micro USB για τροφοδοσία και καλώδιο Ethernet για σύνδεση στο δίκτυο, ανοίγει το πρόγραμμα Advanced IP Scanner. Αυτό εμφανίζει όλες τις συσκευές, οι οποίες είναι συνδεδεμένες σε συγκεκριμένο δίκτυο, οπότε ο χρήστης βρίσκει με αυτόν τον τρόπο τη διεύθυνση IP του Raspberry Pi.

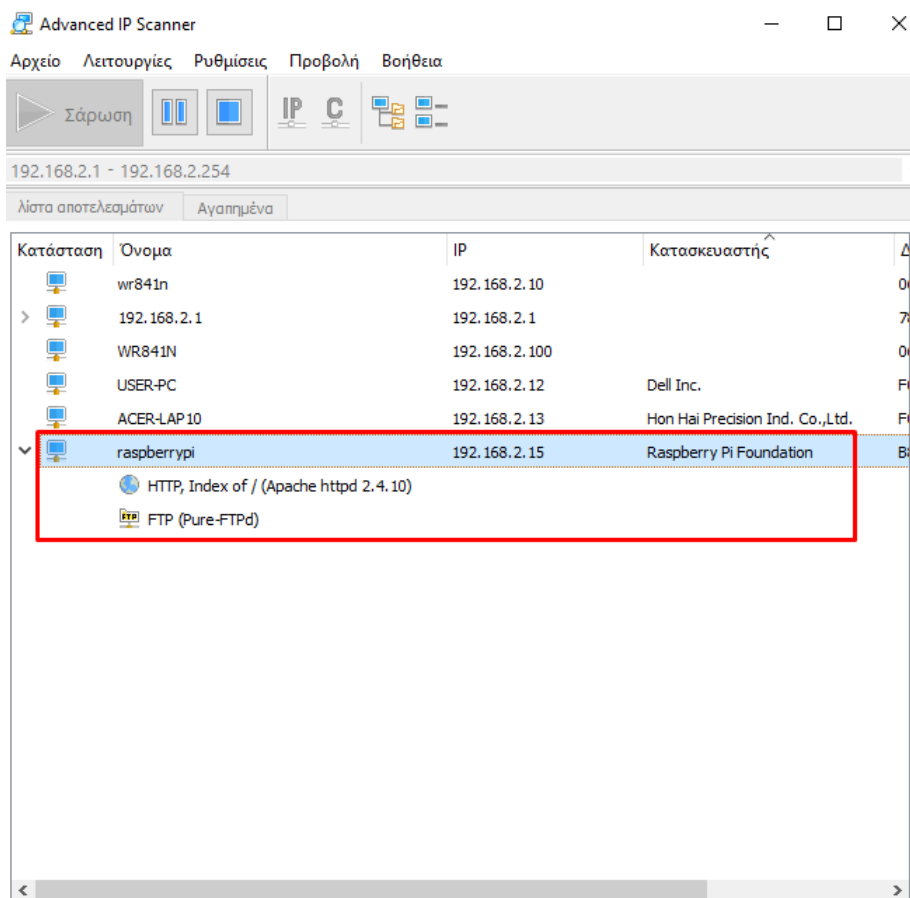


Figure 3.6: Για το πρώτο boot του Raspberry Pi, ο χρήστης αφού το συνδέσει με Ethernet στον προσωπικό του υπολογιστή ή απευθείας στο router για λήψη IP διεύθυνσης, την ανιχνεύει με το πρόγραμμα Advanced IP Scanner, το οποίο εμφανίζει όλες τις διευθύνσεις IP που λαμβάνουν οι συσκευές που είναι συνδεδεμένες στο τοπικό δίκτυο.

24ΚΕΦΑΛΑΙΟ 3. ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΡΥΘΜΙΣΕΙΣ RASPBERRY PI (SETUP)

Στη συνέχεια γράφοντας τη διεύθυνση αυτή στο κατάλληλο πεδίο στο PuTTY και πατώντας 'OPEN' ξεκινάει η σύνδεση.

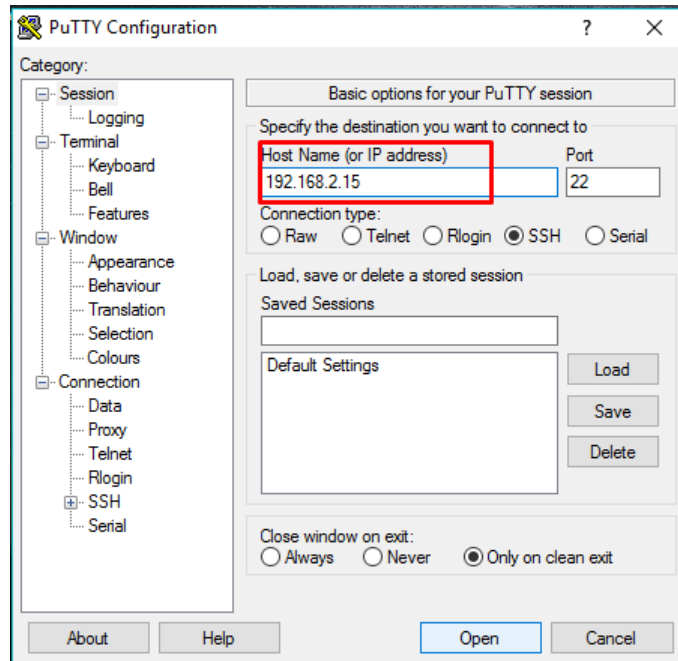


Figure 3.7: Εισαγωγή της διεύθυνσης IP του Raspberry Pi στο πρόγραμμα PuTTY, το οποίο αποτελεί SSH Client και δίνει στο χρήστη τη δυνατότητα να διαχειρίζεται τη γραμμή εντολών του Raspberry Pi (terminal) από το δικό του προσωπικό υπολογιστή χωρίς τη διασύνδεση του Raspberry Pi με οθόνη και πληκτρολόγιο

Εάν η σύνδεση μέσω SSH έχει εδραιωθεί σωστά τότε θα ανοίξει το terminal του Raspberry Pi, όπου στη γραμμή εντολών ο χρήστης πρέπει να δηλώσει το όνομα του user και τον κωδικό πρόσβασης του Raspberry Pi. Εφόσον γίνεται εκκίνηση για πρώτη φορά ο χρήστης πρέπει να πληκτρολογήσει:

```
login as: pi  
pi@localhost's password: raspberry
```

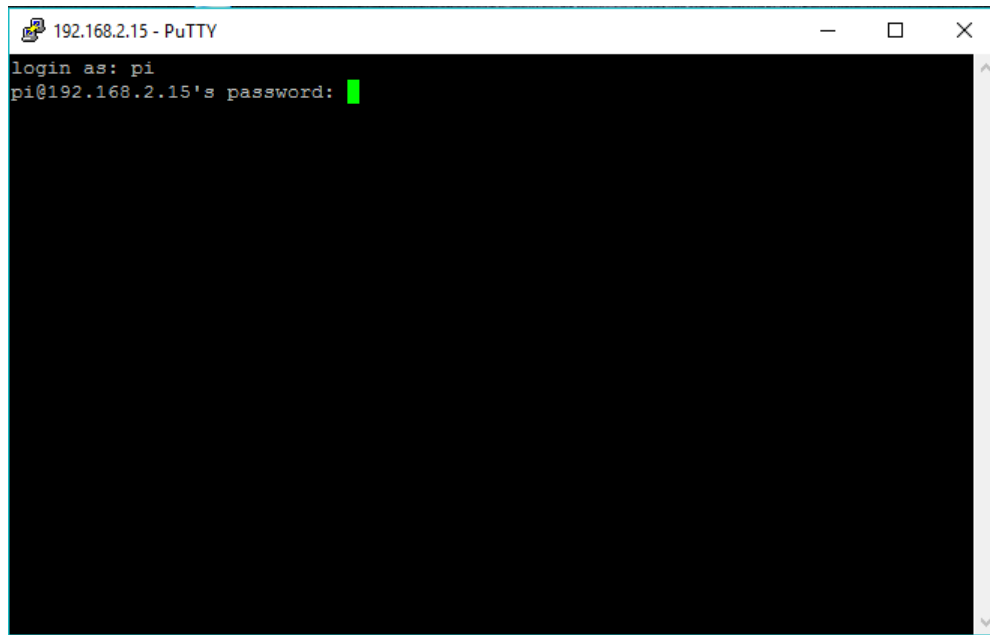


Figure 3.8: Εισαγωγή username: pi και password: raspberry κατά την εκκίνηση του Raspberry Pi

Το Raspberry Pi, έχει ενεργοποιηθεί. Στη συνέχεια, πρέπει να πραγματοποιηθούν κάποιες ρυθμίσεις. Εισάγοντας στη γραμμή εντολών `sudo raspi-config` ανοίγει το παράθυρο με το μενού ρυθμίσεων των λειτουργιών. (Raspberry Pi Software Configuration Tool (raspi-config)).

26ΚΕΦΑΛΑΙΟ 3. ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΡΥΘΜΙΣΕΙΣ RASPBERRY PI (SETUP)

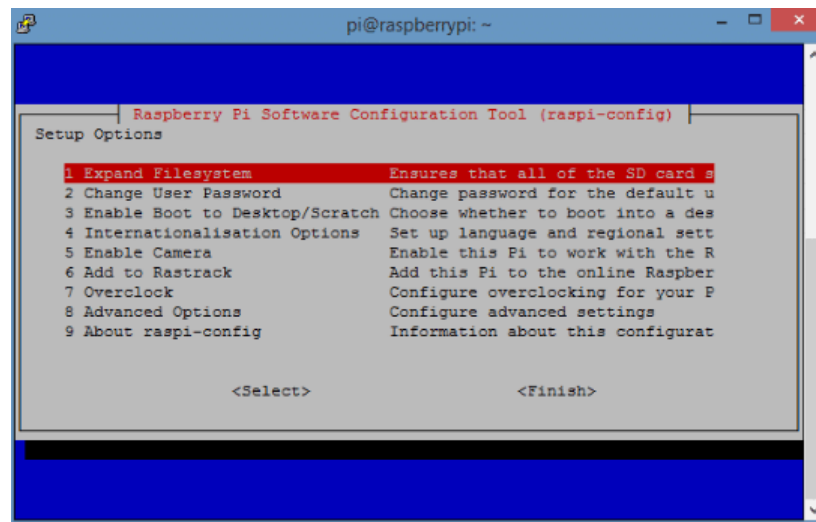


Figure 3.9: Μετά την εισαγωγή username και password, ο χρήστης πρέπει ανοίγει το Μενού ρυθμίσεων των λειτουργιών του Raspberry Pi (Raspberry Pi Software Configuration Tool (raspi-config) με την εντολή ‘sudo raspi-config’.

Στο μενού αυτό ο χρήστης μπορεί να τροποποιήσει τις βασικές ρυθμίσεις του Raspberry Pi. Η πρώτη ρύθμιση, που πρέπει να πραγματοποιηθεί είναι η επέκταση της μνήμης, έτσι ώστε να χρησιμοποιείται όλος ο αποθηκευτικός χώρος της micro SD κάρτας που εισήγαγε ο χρήστης. Αυτό πραγματοποιείται επιλέγοντας την πρώτη επιλογή του μενού των ρυθμίσεων ‘Expand Filesystem’, όπως φαίνεται και στην εικόνα 3.9 .

Το Raspberry Pi 3 έχει ενσωματωμένο 802.11n Wireless LAN αντάπτορα, επομένως ο χρήστης δε χρειάζεται επιπλέον να προμηθευτεί ξεχωριστά Wi-Fi dongle. Επομένως, το επόμενο βήμα είναι η ενεργοποίηση του Wi-Fi. Υπάρχουν δύο τρόποι, ο πρώτος είναι να δηλώσει από πριν τα δίκτυα στα οποία θα συνδέεται και τον κωδικό τους, ενώ ο δεύτερος είναι να τροποποιήσει τις ρυθμίσεις και το Raspberry Pi να συνδέεται αυτόματα στο Wi-Fi κάθε φορά που τροφοδοτείται με ρεύμα και ξεκινά η λειτουργία του και να συμπληρώνεται αυτόματα το αρχείο του πρώτου τρόπου.

Για τον πρώτο τρόπο, ο χρήστης στη γραμμή εντολών του Raspberry Pi μέσω του PuTTY γράφει:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

και στο αρχείο που ανοίγει συμπληρώνει το όνομα του δικτύου και τον κωδικό ασφαλείας του εάν αυτό έχει:

```
update_config=1  
country=GR
```

```
network= {  
  ssid="networkname1"  
  key_mgmt="password"  
}
```

```
network= {  
  ssid="networkname2"  
  key_mgmt=NONE  
}
```

Για το δεύτερο τρόπο, ο χρήστης γράφει:

```
sudo nano /etc/network/interfaces
```

και στο αρχείο που ανοίγει:

```
# interfaces(5) file used by ifup(8) and ifdown(8)  
# Please note that this file is written to be used with dhcpcd  
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'  
# Include files from /etc/network/interfaces.d:  
source-directory /etc/network/interfaces.d  
  
auto lo  
  
iface lo inet loopback  
iface eth0 inet dhcp  
  
allow-hotplug wlan0  
auto wlan0  
iface wlan0 inet manual  
  
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf  
  
allow-hotplug wlan1  
iface wlan1 inet manual  
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf  
  
iface default inet dhcp
```

Με τον τρόπο αυτό το Raspberry Pi συνδέεται αυτόματα στο Wi-Fi.

3.2.1.1 Γραφικό περιβάλλον - VNC Server

Μέσα από το πρόγραμμα PuTTY, το οποίο αποτελεί SSH Client, ο χρήστης έχει πρόσβαση μόνο στη γραμμική εντολών του Raspberry Pi. Υπάρχει δυνατότητα να έχει πρόσβαση και στο γραφικό περιβάλλον του (Graphical User Interface GUI) από το δικό του προσωπικό υπολογιστή, χωρίς δηλαδή να χρειάζεται πάλι να συνδέσει το Raspberry Pi με οθόνη και πληκτρολόγιο. Η δυνατότητα αυτή του δίνεται μέσα από το σύστημα **VNC**.

Το VNC (Virtual Network Computing) είναι ένα γραφικό σύστημα κοινής χρήσης της επιφάνειας εργασίας ενός υπολογιστή, το οποίο χρησιμοποιεί το πρωτόκολλο **Remote Frame Buffer protocol (RFB)** για το χειρισμό άλλου απομακρυσμένου υπολογιστή. Μεταδίδει τα συμβάντα του πληκτρολογίου και του ποντικιού από έναν υπολογιστή στον άλλο, μεταδίδοντας την ενημερωμένη κάθε στιγμή οθόνη προς την άλλη κατεύθυνση, μέσω ενός δικτύου. Το σύστημα αυτό είναι ανεξάρτητο του λογισμικού, στο οποίο Server και Client είναι εγκατεστημένοι.

Πρακτικά αυτό δίνει τη δυνατότητα στο χρήστη να ξεκινήσει προγράμματα και να κάνει διάφορες ρυθμίσεις στο Raspberry Pi από το δικό του υπολογιστή σε γραφικό περιβάλλον (GUI) και όχι από τη γραμμική εντολών στο PuTTY. Το οποίο παραμένει αναγκαίο για την εγκατάσταση του VNC Server αλλά και για την εκκίνηση του κάθε φορά.

Στο Raspbian repository είναι διαθέσιμο το **VNC Connect**, επομένως για την εγκατάστασή του ο χρήστης πρέπει στη γραμμική εντολών στο PuTTY, να δώσει τις ακόλουθες εντολές:

```
sudo apt-get update
sudo apt-get install realvnc-vnc-server realvnc-vnc-viewer
```

και στη συνέχεια να επανεκινήσει (reboot) το Raspberry Pi. Στον υπολογιστή με λογισμικό Windows με τον οποίο θα χειρίζεται το Raspberry Pi, πρέπει να εγκαταστήσει το πρόγραμμα **VNC Viewer**, το οποίο αποτελεί VNC client.

Για να εκκινήσει ο VNC Server, αρχικά πρέπει να δημιουργηθεί το αρχείο **vnc.sh** το οποίο θα εκτελείται για την εκκίνηση:

```
sudo nano vnc.sh
```

στο οποίο ο χρήστης πρέπει να εισάγει τα εξής:

```
#!/bin/sh
vncserver :0 -geometry 1920x1080 -depth 24 -dpi 96
```

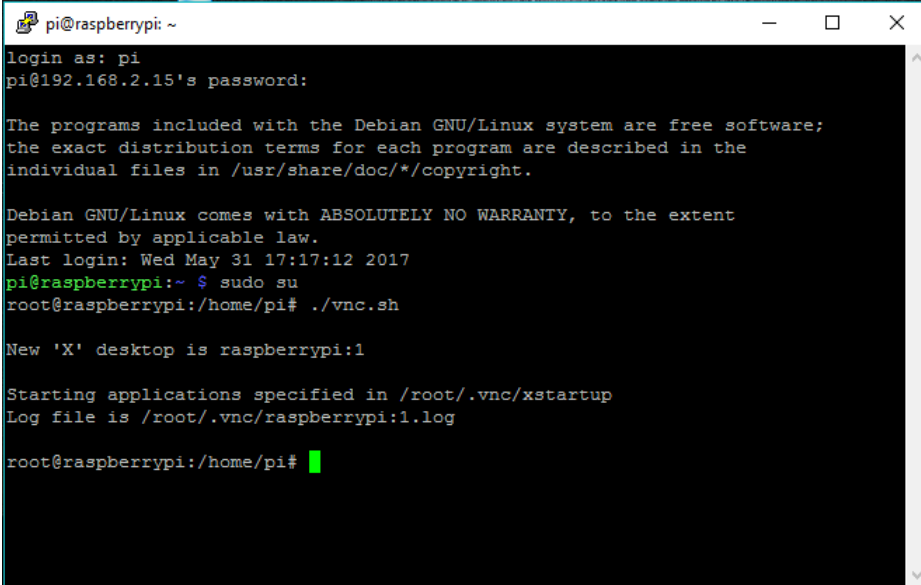
Στη συνέχεια αφού σώσει το αρχείο, να το κάνει εκτελέσιμο με την εντολή:

```
sudo chmod +x vnc.sh
```

Τέλος, εκτελώντας το αρχείο:

```
./vnc.sh
```

ο VNC Server ενεργοποιείται και ο χρήστης με τη διεύθυνση IP που έχει λάβει το Raspberry Pi, μπορεί να συνδεθεί στο VNC Viewer και να το χειριστεί μέσα από το GUI. Αφού γράψει τη διεύθυνση IP του Raspberry Pi θα του ζητηθεί κωδικός, όπου θα συμπληρώσει 'raspberrypi', όπως και όταν συνδέεται μέσω SSH.



```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.2.15's password:  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Wed May 31 17:17:12 2017  
pi@raspberrypi:~ $ sudo su  
root@raspberrypi:/home/pi# ./vnc.sh  
  
New 'X' desktop is raspberrypi:1  
  
Starting applications specified in /root/.vnc/xstartup  
Log file is /root/.vnc/raspberrypi:1.log  
  
root@raspberrypi:/home/pi#
```

Figure 3.10: Ο χρήστης πρέπει να εγκαταστήσει αρχικά τον vncserver στο Raspberry Pi και στη συνέχεια να δημιουργήσει ένα εκτελέσιμο αρχείο './vnc.sh' το οποίο εκτελεί για να χειριστεί το Raspberry Pi μέσα από το γραφικό περιβάλλον του από τον vnc client (Vnc Viewer) που έχει εγκαταστήσει στον προσωπικό του υπολογιστή με λογισμικό Windows.

30ΚΕΦΑΛΑΙΟ 3. ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΡΥΘΜΙΣΕΙΣ RASPBERRY PI (SETUP)

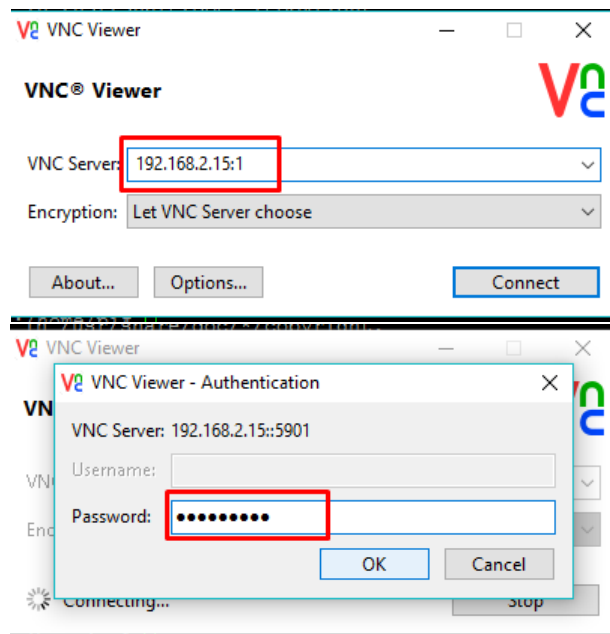


Figure 3.11: Αφού ενεργοποιήσει ο χρήστης τον vnc server στο Raspberry Pi, εικόνα 3.10 εκτελεί το πρόγραμμα VNC Viewer που αποτελεί vnc client. Στο παράθυρο που ανοίγει εισάγει τη διεύθυνση IP του Raspberry Pi και στη συνέχεια αφού πατήσει Connect εισάγει στο πεδίο Password 'raspberrry'.

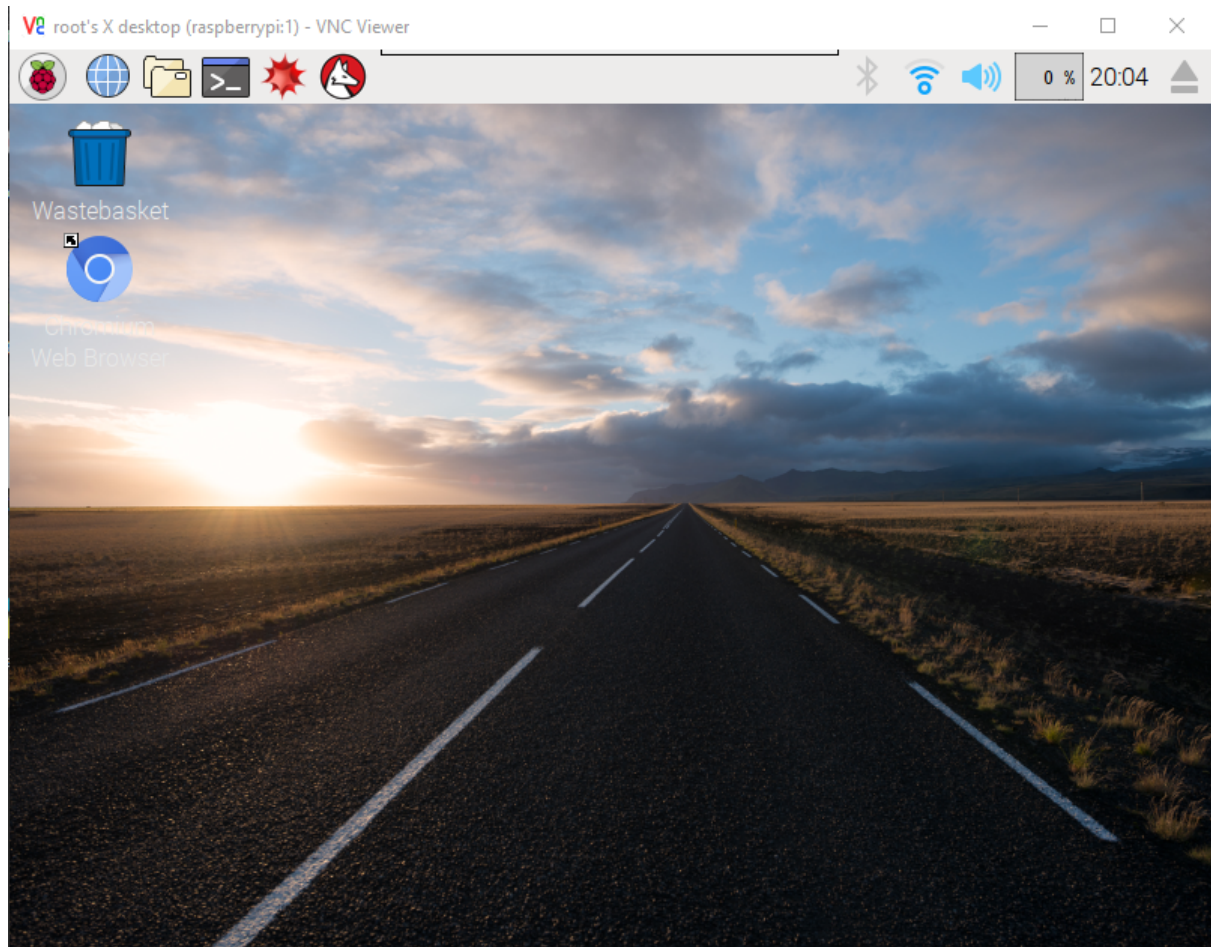


Figure 3.12: Στην εικόνα φαίνεται το γραφικό περιβάλλον του Raspberry Pi, στο οποίο έχει πρόσβαση ο χρήστης μέσα από τον προσωπικό του υπολογιστή, μέσα από το VNC Viewer, χωρίς να συνδέσει οθόνη. Αυτό είναι το περιβάλλον στο οποίο θα έχει πρόσβαση στην περίπτωση που συνδέσει οθόνη.

3.3 Εγκατάσταση και Ενεργοποίηση Apache Server

Για την αποστολή των δεδομένων που συλλέγει το Raspberry Pi από τους αισθητήρες σε webserver, εγκαταστάθηκε στο Raspberry Pi Apache Server. Ο server αυτός, όπως περιγράφηκε στο κεφάλαιο 2, αποτελεί από τους πλέον διαδεδομένους web servers. Η διαδικασία εγκατάστασης είναι αυτή που περιγράφεται στην παρούσα ενότητα.

Στη γραμμή εντολών ο χρήστης γράφει τις ακόλουθες εντολές:

```
sudo apt-get update
sudo apt-get install apache2 -y
```

Εξ' ορισμού ο Apache τοποθετεί ένα test αρχείο HTML (index.html) στο φάκελο web (/var/www/html). Για να δει ο χρήστης αυτήν την test ιστοσελίδα αρκεί να γράψει στο browser του Raspberry Pi: **http://localhost/** ή του προσωπικού του υπολογιστή: : **http://192.168.x.x/**, όπου 192.168.x.x είναι η διεύθυνση IP του Raspberry Pi.

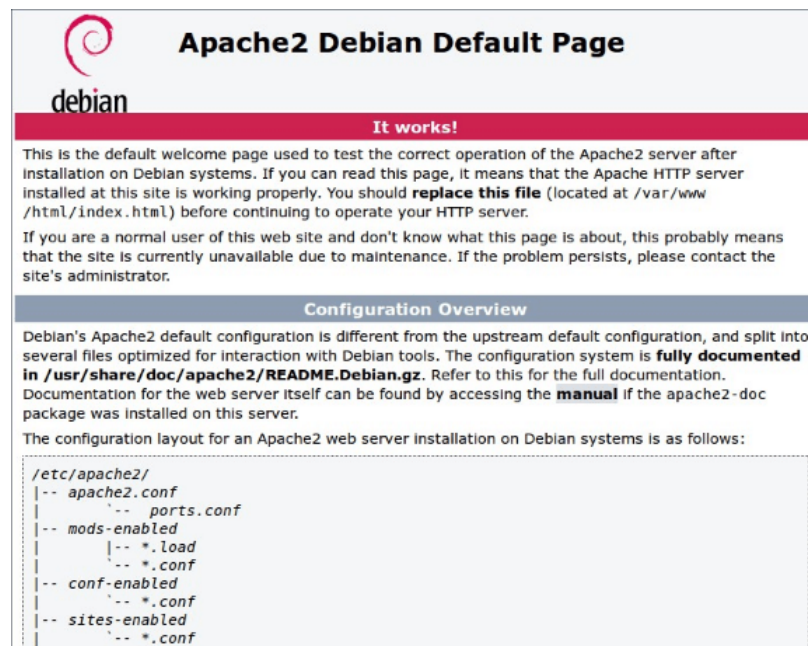


Figure 3.13: Με την εγκατάσταση του Apache τοποθετείται ένα test αρχείο HTML (index.html) στο φάκελο web (/var/www/html). Ο χρήστης έχει πρόσβαση σε αυτήν την test ιστοσελίδα είτε απο το browser του Raspberry Pi (http://localhost/) είτε του προσωπικού του υπολογιστή (http://192.168.x.x/), όπου 192.168.x.x είναι η διεύθυνση IP του Raspberry Pi. Στην εικόνα φαίνεται η test ιστοσελίδα.

Τέλος, για τη δημιουργία της ιστοσελίδας στην οποία θα εμφανίζονται οι τιμές των παραμέτρων θερμοκρασίας, πίεσης και υγρασίας που συλλέγει το Raspberry Pi από τους αισθητήρες BME280 είναι απαραίτητη η εγκατάσταση του πακέτου της γλώσσας PHP. Η εγκατάσταση του πακέτου της γλώσσας και το configuration με τον Apache server γίνεται με την εξής εντολή:

```
sudo apt-get install php5 libapache2-mod-php5 -y
```


Κεφάλαιο 4

Πρωτόκολλο I^2C (Inter-integrated Circuit)

Το πρωτόκολλο αυτό αποτελεί ένα πρωτόκολλο επικοινωνίας και στην πράξη είναι ένας σειριακός δίαυλος (I^2C bus) επικοινωνίας, ο οποίος δημιουργήθηκε από την εταιρία Phillips Semiconductor (τώρα NXP) τη δεκαετία το '80 και πλέον χρησιμοποιείται κατά κόρον από τους περισσότερους αν όχι όλους IC κατασκευαστές. Χρησιμοποιείται για τη σύνδεση περιφερειακών συσκευών μικρής ταχύτητας σε motherboard, embedded systems, κινητά τηλέφωνα ή άλλες ηλεκτρονικές συσκευές. Αξίζει να σημειωθεί, πως δεν χρησιμοποιείται μόνο για την επικοινωνία συσκευών, που βρίσκονται πάνω σε ένα τυπωμένο κύκλωμα, αλλά και για την επικοινωνία συσκευών που συνδέονται με καλώδια. Στο κεφάλαιο, που θα ακολουθεί, περιγράφεται το πρωτόκολλο αυτό και ο τρόπος με τον οποίο πραγματοποιείται η επικοινωνία των συσκευών που συνδέονται. Τέλος, περιγράφεται πως έγινε το configuration στο Raspberry Pi.

4.1 Σειριακή μεταφορά δεδομένων

Το I^2C bus χρησιμοποιείται για την επικοινωνία και τη μεταφορά δεδομένων μεταξύ μιας συσκευής **master** (μικροελεγκτές ή επεξεργαστές) με μια ή περισσότερες ψηφιακές συσκευές **slave**. Κάθε μια από τις δεύτερες έχει τη δική της ξεχωριστή I^2C **διεύθυνση** (I^2C **address**), η οποία μπορεί να είναι είτε 7 bit είτε 10 bit και εξαρτάται από την εκάστοτε συσκευή. Οι master συσκευές δε χρειάζεται να έχουν διεύθυνση, εκτός αν στο I^2C bus είναι συνδεδεμένες πολλές master συσκευές. Η βασική I^2C επικοινωνία γίνεται με μεταφορά 8 bits. Από αυτά τα 8 bits τα πρώτα 7 αποτελούν την I^2C address της slave συσκευής. Το επόμενο bit δηλαδή το 8^ο, χρησιμοποιείται για να δηλώσει η συσκευή master αν θα διαβάσει ή θα γράψει από/στη slave. Για παράδειγμα, αν το bit αυτό έχει οριστεί στο 1 (high) τότε η συσκευή master θα διαβάσει από τη συσκευή slave.

Η χρήση του συγκεκριμένου πρωτοκόλλου καθίσταται αρκετά εύκολη σε

σχέση με άλλα (πχ SPI) καθώς η συνδεσμολογία του είναι αρκετά απλή. Πιο συγκεκριμένα η συνδεσμολογία αποτελείται από 2 μόνο γραμμές:

- **SDA** (Serial Data Line - Γραμμή δεδομένων)
- **SCL** (Serial Clock Line - Γραμμή ρολογιού)

Η γραμμή SCL ταλαντώνεται μεταξύ της κατάστασης **0 (low)** και **1 (high)**, παράγοντας τους παλμούς του ρολογιού, όπως φαίνεται στην εικόνα 4.1. Για κάθε στιγμή που η γραμμή SCL βρίσκεται στην κατάσταση 1, η SDA αντιστοιχεί στο bit που διαβάζει ή στέλνει ο master. Η master συσκευή είναι αυτή που ελέγχει και οδηγεί τη γραμμή SCL, ενώ οι συσκευές slave είναι αυτές που ανταποκρίνονται στις συσκευές master. Η γραμμή SCL ταλαντώνεται 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.

Μια slave συσκευή δεν μπορεί να ξεκινήσει μια μεταφορά δεδομένων πάνω στο I^2C bus, μόνο η master συσκευή μπορεί. Και οι δύο συσκευές μπορούν να μεταφέρουν δεδομένα στο I^2C bus, αλλά μόνο οι master συσκευές ελέγχουν τη μεταφορά. Οι δυο αυτές γραμμές συνδέονται σε όλες τις συσκευές, που υπάρχουν πάνω στο I^2C bus. Πρακτικά, αυτό σημαίνει, πως ανεξάρτητα από τον αριθμό των slave συσκευών απαιτούνται 2 μόνο καλώδια για τη μεταφορά δεδομένων και την επικοινωνία με όλους. Για τη λειτουργία των slave συσκευών υπάρχουν προφανώς άλλα δυο καλώδια, ένα τροφοδοσίας και ένα γείωσης.

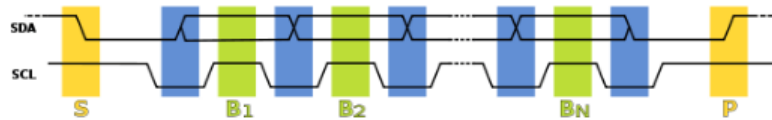


Figure 4.1: Η συνδεσμολογία για τη χρήση του πρωτοκόλλου I^2C αποτελείται από 2 γραμμές, τη γραμμή του ρολογιού SCL και τη γραμμή των δεδομένων. Η γραμμή SCL ταλαντώνεται μεταξύ της κατάστασης **0 (low)** και **1 (high)**, παράγοντας τους παλμούς του ρολογιού. Για κάθε στιγμή που η γραμμή SCL βρίσκεται στην κατάσταση 1, η SDA αντιστοιχεί στο bit που διαβάζει ή στέλνει ο master. Η συσκευή master ελέγχει τη γραμμή του ρολογιού, ενώ οι συσκευές slave ανταποκρίνονται στους παλμούς αυτούς. Και οι δύο μπορούν να μεταφέρουν δεδομένα στο I^2C bus αλλά μόνο οι master ελέγχουν την επικοινωνία. Οι 2 γραμμές είναι συνδεδεμένες με όλες τις συσκευές που υπάρχουν στο bus και αυτό πρακτικά σημαίνει πως ανεξάρτητα από τον αριθμό των συσκευών, για την επικοινωνία τους απαιτούνται μόνο 2 καλώδια.

4.1.1 Περιγραφή του πρωτοκόλλου επικοινωνίας I^2C

Στην αρχή και οι δυο γραμμές SCL και SDA βρίσκονται στην κατάσταση high. Η επικοινωνία ξεκινάει από τη master συσκευή, όταν στείλει τη **Συνθήκη Έναρξης**

(Start Condition) με την οποία ενημερώνει όλους τους slaves, ότι ξεκινάει η επικοινωνία και τερματίζεται όταν στείλει τη **Συνθήκη Τερματισμού (Stop Condition)**.

Όπως φαίνεται και στην εικόνα 4.2 η συνθήκη Start σηματοδοτείται όταν η γραμμή SDA γίνει low, ενώ η SCL παραμένει high. Από την άλλη, η συνθήκη Stop σηματοδοτείται όταν η γραμμή SDA από low γίνει high, με την SCL να παραμένει high. Μεταξύ των δυο αυτών συνθηκών πραγματοποιείται η επικοινωνία και η μεταφορά δεδομένων μεταξύ master-slave. Ο δίαυλος I^2C (bus) θεωρείται κατειλημένος μετά τη συνθήκη Start και ελεύθερος μετά τη συνθήκη Stop.

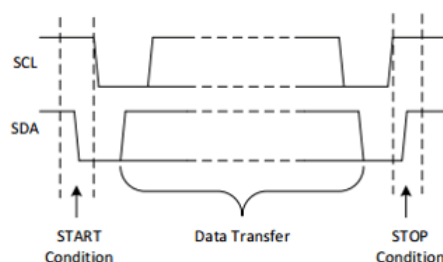


Figure 4.2: Πρωτόκολλο επικοινωνίας I^2C : Συνθήκες έναρξης και τερματισμού. Αρχικά και οι δυο γραμμές βρίσκονται στην κατάσταση high και η επικοινωνία ξεκινάει από τη συσκευή master όταν αυτή στείλει την συνθήκη START και τερματίζεται όταν στείλει τη συνθήκη STOP. Η συνθήκη Start σηματοδοτείται όταν η γραμμή SDA γίνει low, ενώ η SCL παραμένει high. Η συνθήκη Stop σηματοδοτείται όταν η γραμμή SDA από low γίνει high, με την SCL να παραμένει high.

Μετά τη συνθήκη Start, ο master στέλνει την I^2C διεύθυνση του slave με τον οποίο θέλει να επικοινωνήσει (7 bit). Η μεταφορά δεδομένων γίνεται σε ακολουθίες των 8 bit, τα οποία τοποθετούνται στη γραμμή SDA, ξεκινώντας από το πιο σημαντικό bit (MSB). Το επιπλέον bit λέγεται **bit R/W** και χρησιμεύει για να πληροφορήσει τη slave συσκευή εάν η master πρόκειται να γράψει (να στείλει δεδομένα) ή να διαβάσει (να λάβει δεδομένα) από αυτήν. Όπως έχει ήδη αναφερθεί, αν το bit αυτό είναι μηδέν η master συσκευή πρόκειται να γράψει στη slave, ενώ αν είναι 1 πρόκειται να διαβάσει. Όταν μεταφέρονται δεδομένα, η γραμμή SCL πάλλεται από την κατάσταση high στην κατάσταση low, ενώ η γραμμή SDA πρέπει να παραμένει σταθερή και να μην αλλάζει όσο η γραμμή SCL είναι high.

Για κάθε 8 bit δεδομένων που μεταφέρονται, η συσκευή slave στέλνει πίσω ένα bit επιβεβαίωσης (**Acknowledge bit**), με το οποίο δηλώνει αν έχει όντως αυτή τη διεύθυνση ή όχι. Έτσι στην πραγματικότητα απαιτούνται 9 παλμοί ρολογιού, για τη μεταφορά των 8 bit κάθε byte δεδομένων. Το τελευταίο αυτό bit καθορίζεται από την κατάσταση της γραμμής SDA, από το αν θα μεταβεί στην κατάσταση low ή αν θα παραμείνει σταθερή στην κατάσταση high. Αν η γραμμή SDA διατηρηθεί high από τη συσκευή slave, τότε θα σταλεί από αυτόν ένα **negative acknowledge (NAK)** bit, ενώ αν η γραμμή αλλάξει στην κατάσταση low, τότε θα σταλθεί ένα **acknowledge**

(ACK) bit. Στην εικόνα 4.3 αναπαριστάται όλη η διαδικασία της επικοινωνίας.

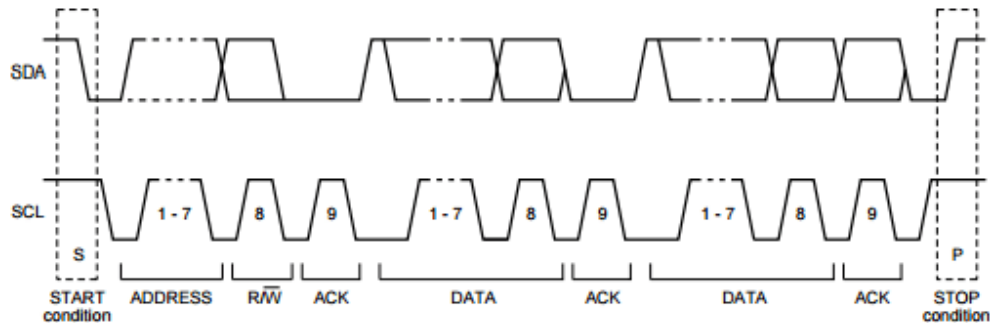


Figure 4.3: Για την επικοινωνία απαιτούνται ουσιαστικά 9 bit. Μετά την αποστολή της συνθήκης START από τη συσκευή master, η ίδια στέλνει τα 7 επόμενα bit που αποτελούν την I^2C διεύθυνση της συσκευής slave με την οποία θέλει να επικοινωνήσει και η SDA παραμένει high. Ακολουθεί το bit (R/W) και τέλος το ACK ή NACK bit επιβεβαίωσης που εξαρτάται από την κατάσταση της SDA. Η ίδια διαδικασία ακολουθείται μέχρις ότου η master συσκευή τερματίσει την επικοινωνία με την αποστολή της συνθήκης STOP. Κατά τη μεταφορά δεδομένων η SDA πρέπει να παραμένει σταθερή και να μην αλλάζει όσο η SCL είναι σε κατάσταση high.

4.2 Raspberry Pi και I²C

Συνοπτικά, σύμφωνα με όσα αναφέρθηκαν, το I²C bus επιτρέπει τη σύνδεση πολλών συσκευών (slaves) στο Raspberry Pi, το οποίο είναι προφανώς η master συσκευή. Κάθε μία από τις slave συσκευές έχει τη δική της διαφορετική I²C address. Οι διευθύνσεις αυτές δίνουν τη δυνατότητα στο χρήστη, να ξέρει με ποιές συσκευές επικοινωνεί το Raspberry Pi, εφόσον αν δεν είναι σωστά συνδεδεμένες, δεν τις ανιχνεύει δηλαδή μέσω του πρωτοκόλλου, δεν είναι δυνατή η επικοινωνία μεταξύ τους. Για να πραγματοποιηθούν αυτά πρέπει ο χρήστης να ακολουθήσει τη διαδικασία που περιγράφεται στην επόμενη παράγραφο.

4.2.1 Ρυθμίσεις ενεργοποίησης(Configuration)

Στο Raspberry Pi αυτό το πρωτόκολλο επικοινωνίας είναι απενεργοποιημένο και πρέπει να ενεργοποιηθεί από το χρήστη. Το πρώτο βήμα είναι η εκτέλεση της εντολής:

```
sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

Στο αρχείο που ανοίγει, υπάρχει ένα σχόλιο και δυο ακόμα γραμμές:

```
#blacklist spi and i2c by default (many users don't need them)
```

```
blacklist spi-bcm2708
```

```
blacklist i2c-bcm2708
```

Πρέπει να γίνει σχολιασμός της δεύτερης γραμμής (`blacklist i2c-bcm2708`) και να σωθεί το τροποποιημένο αρχείο:

```
#blacklist spi and i2c by default (many users don't need them)
```

```
blacklist spi-bcm2708
```

```
#blacklist i2c-bcm2708
```

Στη συνέχεια πρέπει να γίνει εγκατάσταση και ενεργοποίηση της επιλογής I²C του πυρήνα (Enable Kernel I2C Module). Ο πιο εύκολος τρόπος είναι μέσω του **raspi-config**, που περιγράφεται με τις εικόνες που ακολουθούν. Αρχικά εκτελείται η εντολή:

```
sudo raspi-config
```

και ανοίγει το παράθυρο, Raspberry Pi Configuration Tool (raspi-config) που φαίνεται στην εικόνα 4.4, όπου επιλέγει την επιλογή Interfacing Options - Configure Connection to peripheral και εμφανίζεται το μενού επιλογών της εικόνας 4.5, όπου

επιλέγει I2C- Enable-Disable automatic loading. Τέλος, οδηγείται στο παράθυρο της εικόνας 4.6 όπου επιλέγει <Yes> και έτσι γίνεται η ενεργοποίηση ARM I2C interface.

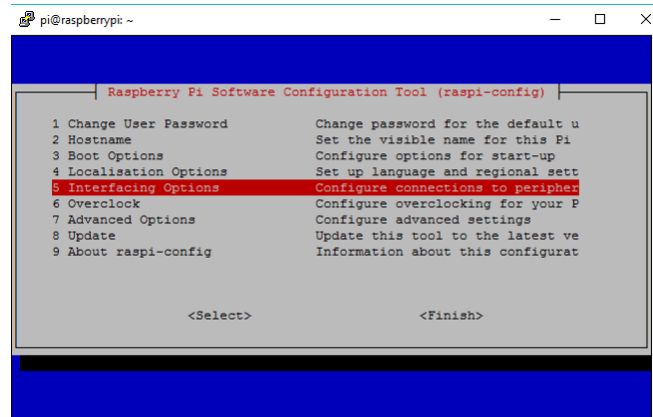


Figure 4.4: Στο παράθυρο που ανοίγει μετά την εκτέλεση της εντολής, ο χρήστης από το μενού Raspberry Pi Configuration Tool (raspi-config) επιλέγει Interfacing Options - Configure Connection to peripheral.

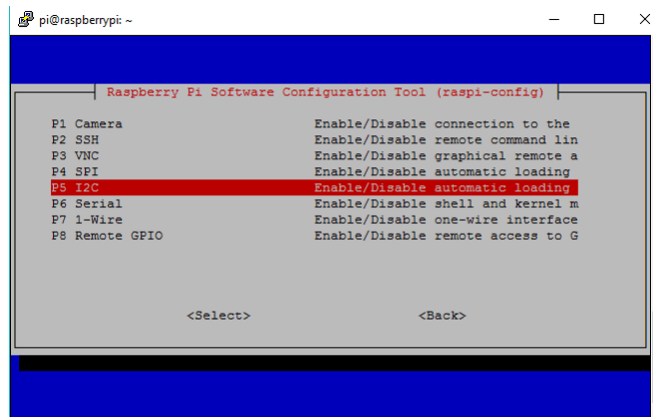


Figure 4.5: Στο μενού των επιλογών που εμφανίζεται γίνεται επιλογή του I2C- Enable-Disable automatic loading.

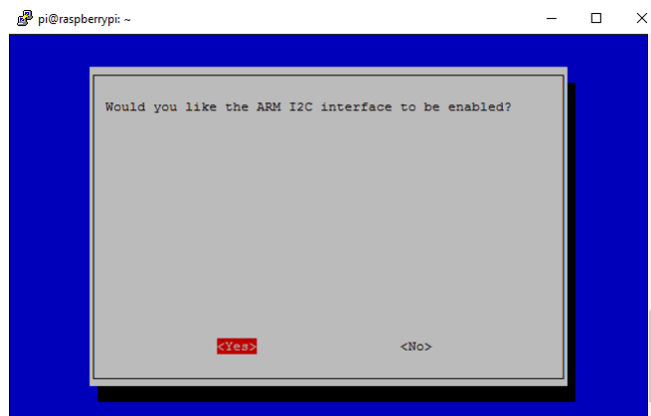


Figure 4.6: Ενεργοποίηση ARM I2C interface

Τέλος, πρέπει να εγκατασταθούν τα απαραίτητα εργαλεία (i2c-tools) εκτελώντας τις ακόλουθες εντολές:

```
sudo apt-get install -y python-smbus
sudo apt-get install -y i2c-tools
```

Αφού πραγματοποιηθούν όλα αυτά, το Raspberry Pi επανεκκινείται και έπειτα γίνεται έλεγχος αν αναγνωρίζει όλες τις συνδεδεμένες slave συσκευές, έτσι ώστε στη συνέχεια να επικοινωνήσει με αυτές. Η ανίχνευση πραγματοποιείται εκτελώντας την εντολή:

```
sudo i2cdetect -y 1
```



```

pi@raspberrypi: ~
root@raspberrypi:/home/pi# i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  76 77
root@raspberrypi:/home/pi#

```

Figure 4.7: Αποτέλεσμα εκτέλεσης της εντολής ‘sudo i2cdetect -y 1’ ενώ είναι συνδεδεμένοι με το Raspberry Pi δυο αισθητήρες BME280 με I^2C διευθύνσεις 0x77 και 0x76

Ο αριθμός 1 δηλώνει το **I2C-port** στο οποίο πραγματοποιείται η επικοινωνία και εξαρτάται από το μοντέλο Raspberry Pi το οποίο χρησιμοποιείται. Τα μοντέλα Rev 2 χρησιμοποιούν το I2C-port = 1, ενώ τα μοντέλα Rev 1 Pi χρησιμοποιούν το I2C-port = 0. Η διαφορά έγκειται στο ότι οι εκδόσεις Rev 1 και Rev 2 διαφέρουν στο σήμα που δέχονται στα pins 3 και 5 του GPIO. Το Raspberry Pi 3 Model B+ είναι Rev 2 άρα χρησιμοποιείται το port 1.

Κεφάλαιο 5

Αισθητήρας BME280

Στο κεφάλαιο αυτό γίνεται περιγραφή του αισθητήρα BME280, ο οποίος χρησιμοποιήθηκε για την πραγματοποίηση του project της παρούσας διπλωματικής.

5.1 Περιγραφή

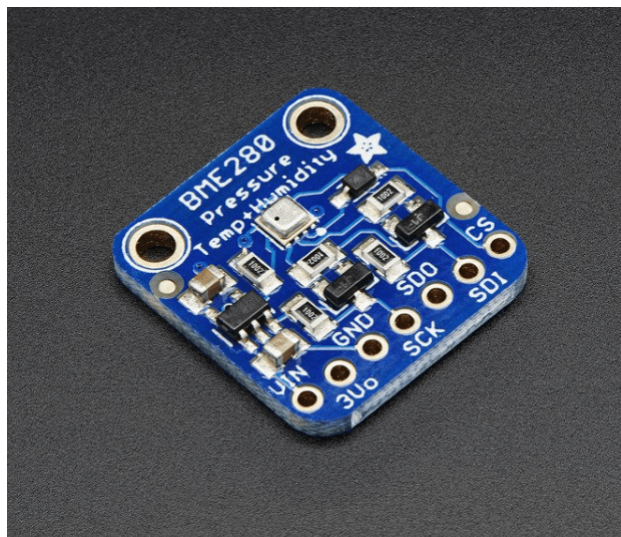


Figure 5.1: Ο αισθητήρας BME280, κατασκευάζεται από την Bosch και η πλακέτα του, που φαίνεται στην εικόνα διατίθεται από την Adafruit.

Ο αισθητήρας BME280, κατασκευάζεται από την Bosch και η πλακέτα που φαίνεται στην εικόνα 5.1 διατίθεται από την Adafruit. Ο BME280 είναι ένας σχετικά οικονομικός ψηφιακός αισθητήρας πίεσης, υγρασίας και θερμοκρασίας, ο οποίος παρέχει μετρήσεις με μεγάλη ακρίβεια και μπορεί να λειτουργήσει είτε με το πρωτόκολλο

SPI (Standard peripheral Interface Protocol) είτε με το πρωτόκολλο I^2C , που αναλύθηκε στο κεφάλαιο 4.

Αρχικά σχεδιάστηκε για mobile εφαρμογές, για παράδειγμα ενσωμάτωση σε smartphones και smartwatches νέας τεχνολογίας, καθώς και gps. Χρησιμοποιείται, επίσης, ευρέως στον τομέα Internet of Things (IoT) και σε εφαρμογές αυτοματισμού και ελέγχου τόσο σε σπίτια όσο και βιομηχανικές εγκαταστάσεις (έλεγχος θέρμανσης και εξαερισμού, προειδοποίηση σχετικά με την ξηρότητα του αέρα ή τις υψηλές θερμοκρασίες κτλ). Τα βασικά πλεονεκτήματα, που παρέχει ο αισθητήρας αυτός, είναι πρωτίστως η δυνατότητα που δίνει στο χρήστη να λαμβάνει τρεις τιμές για θερμοκρασία, πίεση και υγρασία ταυτόχρονα χρησιμοποιώντας μόνο έναν αισθητήρα, οποίος έχει πολύ μικρό μέγεθος και σχετικά χαμηλή τιμή. Τέλος, όπως αναφέρθηκε, ο συγκεκριμένος αισθητήρας παρέχει αρκετά μεγάλη ακρίβεια μετρήσεων, χαμηλά επίπεδα θορύβου και αρκετά γρήγορη απόκριση. Πιο αναλυτικά, όπως δίνεται από τον κατασκευαστή, η ακρίβεια κάθε τιμής είναι:

- Θερμοκρασία: $\pm 1^\circ C$
- Πίεση: $\pm 1hPa$
- Υγρασία: $\pm 3\%$

Στον πίνακα που ακολουθεί δίνεται το Pin mapping της πλακέτας του αισθητήρα ξεκινώντας από τα αριστερά προς στα δεξιά, όπως φαίνεται στην εικόνα 5.1

1	2	3	4	5	6	7
Vin	$3V_0$	GND	SCK	SD0	SD1	CS

Table 5.1: Pin mapping BME280

Πιο αναλυτικά τα pins

- **SCK** : **C**lock **L**ine pin: είσοδος (input) στην πλακέτα του αισθητήρα
- **SDO**: **S**erial **D**ata **O**ut (Master In Slave Out pin) αποστολή δεδομένων από τον αισθητήρα προς στη συσκευή master
- **SDI**: **S**erial **D**ata **I**n (Master Out Slave In pin) αποστολή δεδομένων από τη master συσκευή προς τον αισθητήρα
- **CS**: **C**hip **S**elect pin είσοδος (input) στην πλακέτα του αισθητήρα για χρήση SPI πρωτοκόλλου επικοινωνίας

Αξίζει να σημειωθεί πως στην πλάκέτα του αισθητήρα περιέχεται κύκλωμα level shifting έτσι ώστε να είναι δυνατόν και ασφαλές να χρησιμοποιηθεί λογική τάση εισόδου (Vin) είτε 3V είτε 5V.

5.1.1 Προδιαγραφές λειτουργίας

Στους πίνακες, που ακολουθούν, αποτυπώνονται οι γενικές ηλεκτρικές προδιαγραφές λειτουργίας του αισθητήρα, αλλά και οι προδιαγραφές κάθε παραμέτρου ξεχωριστά, όπως δίνονται από τον κατασκευαστή.

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 5.2: Γενικές προδιαγραφές αισθητήρα BME280

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 ⁹	$A_{T,25}$	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 5.3: Προδιαγραφές παραμέτρου θερμοκρασίας

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	TCO_P	25...65 °C, 900 hPa		±1.5		Pa/K
				±12.6		cm/K
Absolute accuracy pressure	$A_{P,full}$	300 ... 1100 hPa 0 ... 65 °C		±1.0		hPa
Relative accuracy pressure $V_{DD} = 3.3V$	A_{rel}	700 ... 900hPa 25 ... 40 °C		±0.12		hPa

Figure 5.4: Προδιαγραφές παραμέτρου πίεσης

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 5.5: Προδιαγραφές παραμέτρου υγρασίας

5.1.2 Λειτουργία αισθητήρα BME280

Ο αισθητήρας παρέχει 3 διαφορετικές καταστάσεις λειτουργίας (**sensor modes**) και η εναλλαγή ανάμεσα στα modes γίνεται από την επιλογή της ρύθμισης mode[1:0], όπως φαίνεται στην εικόνα 4.8:

- **Sleep mode:** είναι η επιλεγμένη κατάσταση μετά την εκκίνηση, η κατανάλωση ισχύος είναι στο ελάχιστο, καθώς δεν πραγματοποιείται καμία διαδικασία και όλοι οι καταχωρητές (registers) είναι προσβάσιμοι.
- **Forced mode:** πραγματοποιείται μια μόνο μέτρηση κατόπιν αιτήματος. Όταν τελειώσει η μέτρηση, ο αισθητήρας επιστρέφει στο sleep mode και το αποτέλεσμα της μέτρησης μπορεί να ανακτηθεί από τα data registers. Για την επόμενη μέτρηση αυτή η λειτουργία πρέπει να επιλεγθεί ξανά και συνιστάται για εφαρμογές, που απαιτούν χαμηλό χρόνο δειγματοληψίας,

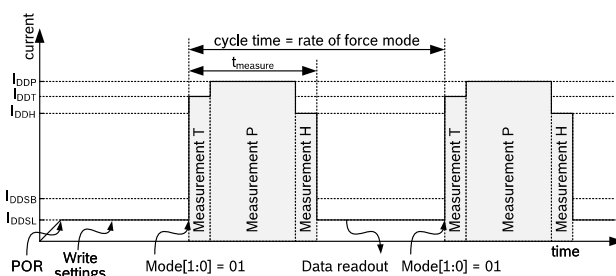


Figure 5.6: Χρονικό διάγραμμα λήψης μετρήσεων με τη λειτουργία forced mode συναρτήσει του ρεύματος

- **Normal mode:** αυτόματες εναλλαγές μιας μέτρησης και μιας περιόδου αναστολής λειτουργίας. Η λειτουργία αυτή συνιστάται, όταν χρησιμοποιείται το ενσωματωμένο IIR φίλτρο.

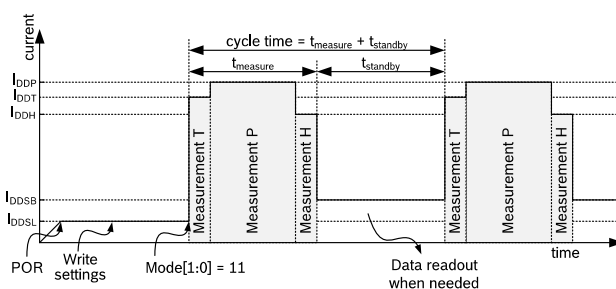


Figure 5.7: Χρονικό διάγραμμα λήψης μετρήσεων με τη λειτουργία normal mode συναρτήσει του ρεύματος

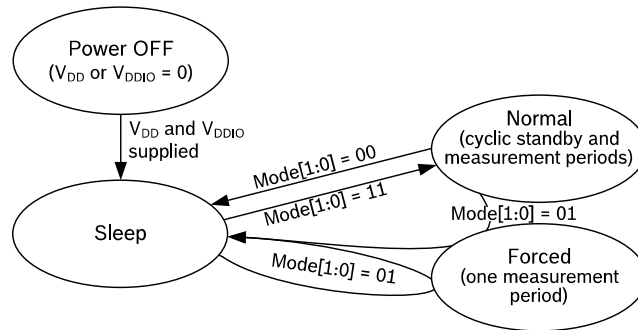


Figure 5.8: Ο αισθητήρας παρέχει 3 καταστάσεις λειτουργίας, sleep mode, forced mode και normal mode. Η εναλλαγή των καταστάσεων λειτουργίας γίνεται με την επιλογή της ρύθμισης mode[1:0] από το χρήστη.

Προκειμένου να προσαρμοσθούν ο ρυθμός των δεδομένων, ο θόρυβος, ο χρόνος απόκρισης και η κατανάλωση με τις ανάγκες του χρήστη παρέχεται ποικιλία στους τρόπους δειγματοληψίας (oversampling modes), στις καταστάσεις του φίλτρου και στο ρυθμό λήψης, από όπου ο χρήστης μπορεί να επιλέξει.

Η περίοδος λήψης μετρήσεων του αισθητήρα BME280 αποτελείται από μια μέτρηση θερμοκρασίας, μία πίεσης και μία υγρασίας με προεπιλεγμένο εύρος δειγματοληψίας. Στην εικόνα 4.9 φαίνεται το διάγραμμα της ροής λήψης μετρήσεων για τις τρεις παραμέτρους.

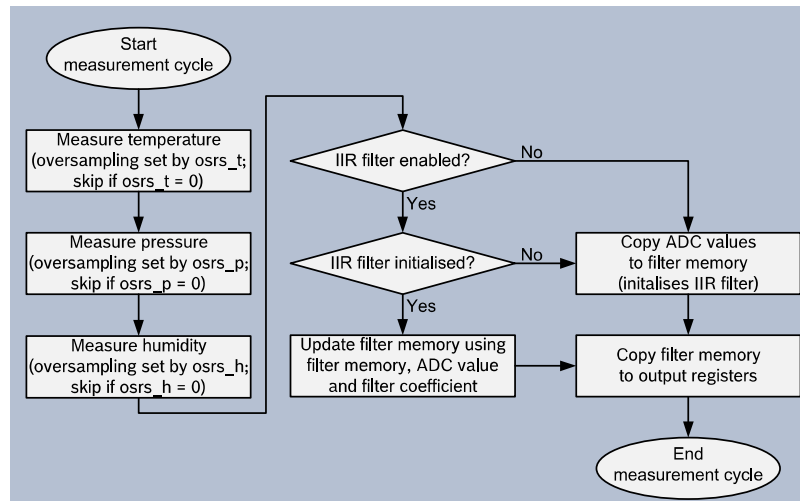


Figure 5.9: Διάγραμμα λήψης μετρήσεων για τις τρεις παραμέτρους

Μετά την περίοδο μέτρησης, οι μετρήσεις της πίεσης και της θερμοκρασίας μπορούν να περάσουν από ένα προαιρετικό αναδρομικό φίλτρο (IIR filter), το οποίο αφαιρεί τις βραχυπρόθεσμες (short-term) δακυμάνσεις στις τιμές της πίεσης. Για

τις τιμές της υγρασίας μέσα στον αισθητήρα κάτι τέτοιο δε χρειάζεται, καθώς δεν κυμαίνονται ταχέως και δεν απαιτούν φιλτράρισμα χαμηλής διέλευσης. Ωστόσο, η περιβαλλοντική πίεση υπόκειται σε πολλές short-term αλλαγές. Για την καταστολή αυτών των διαταραχών στα δεδομένα εξόδου, ο αισθητήρας περιλαμβάνει ένα IIR φίλτρο, το οποίο μειώνει αποτελεσματικά το εύρος ζώνης (bandwidth) των σημάτων εξόδου της θερμοκρασίας και της πίεσης και αυξάνει το resolution των δεδομένων στα 20bit. Η έξοδος του φίλτρου δίνεται από τη σχέση που ακολουθεί.

$$\text{data_filtered} = \frac{\text{data_filtered_old} \cdot (\text{filter_coefficient} - 1) + \text{data_ADC}}{\text{filter_coefficient}}$$

Στη σχέση αυτή, `data_filtered` είναι η νέα τιμή της μνήμης του φίλτρου και η τιμή, που θα καταχωρηθεί στους register του αισθητήρα, `data_filtered_old` είναι η τιμή που προέρχεται από τη μνήμη την εκάστοτε στιγμή, `data_ADC` (ADC, Analog to Digital Conversion) η εκάστοτε ADC τιμή. Στο IIR φίλτρο μπορούν να χρησιμοποιηθούν διάφορες σταθερές φίλτρου (coefficients), οι οποίες καθυστερούν την απόκριση στις εισόδους του αισθητήρα. Ο χρόνος απόκρισης, όταν το φίλτρο είναι ενεργοποιημένο εξαρτάται από τον αριθμό των δεδομένων και τη συχνότητα των δεδομένων εξόδου. Στην εικόνα που ακολουθεί φαίνεται η βηματική απόκριση του φίλτρου για τις διάφορες δυνατές τιμές της coefficient του φίλτρου.

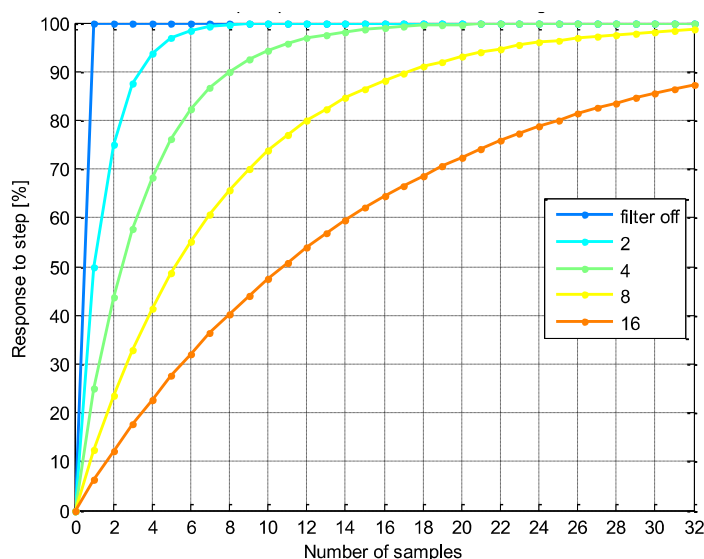


Figure 5.10: Βηματική απόκριση του φίλτρου για τις διάφορες δυνατές τιμές της coefficient του φίλτρου.

Η μέτρηση για κάθε μία από τις παραμέτρους μπορεί να ενεργοποιηθεί αλλά και να παραλειφθεί. Όταν είναι ενεργοποιημένες, υπάρχουν διάφορες επιλογές για το εύρος της δειγματοληψίας και την υπερδειγματοληψία (oversampling), η οποία

μπορεί και να μειώσει το θόρυβο στις μετρήσεις. Η μέτρηση της υγρασίας ελέγχεται από τη ρύθμιση **osrs_h[2:0]**, της πίεσης από **osrs_p[2:0]** και της θερμοκρασίας από **osrs_t[2:0]**. Τέλος, όσον αφορά το resolution των μετρήσεων κάθε παραμέτρου, της υγρασίας είναι καθορισμένη στα 16 bit ενώ της πίεσης και της θερμοκρασίας εξαρτάται από το εάν είναι ενεργοποιημένο το IIR φίλτρο. Στην περίπτωση, που είναι ενεργοποιημένο το resolution, είναι 20 bit ενώ αντίθετα είναι $16+(\text{osrs}_p-1)$ και $16+(\text{osrs}_t-1)$, αντίστοιχα.

Τέλος παρουσιάζεται το **Memory map** του αισθητήρα BME280 για κάθε register.

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]		model[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:		read only	read / write	read only	read only	read only	write only

Figure 5.11: Memory map του αισθητήρα BME280

5.1.2.1 Η Φυσική του αισθητήρα BME280

Ο αισθητήρας BME280 είναι ένας αισθητήρας **MEMS** (Micro-Electro-Mechanical System) και συγκεκριμένα ένας από τους λίγους MEMS αισθητήρες υγρασίας. Η τεχνολογία MEMS είναι η τεχνολογία, στην οποία βασίζεται η κατασκευή ηλεκτρονικών συσκευών σε κλίμακα micro. Η λήψη των τιμών για τις τρεις παραμέτρους βασίζεται σε συγκεκριμένες τεχνικές της επιστήμης της Φυσικής, οι οποίες περιγράφονται περιληπτικά στη συνέχεια.

Το βασικό στοιχείο, που απαιτείται για τη λήψη των μετρήσεων κυρίως της πίεσης και της υγρασίας και κατ' επέκταση της θερμοκρασίας, είναι μια κοιλότητα με θύρα (έξοδος) στον περιβάλλοντα αέρα, ώστε να είναι δυνατή η απορρόφηση μορίων νερού από τον περιβάλλοντα αέρα.

Ο αισθητήρας υγρασίας, που βρίσκεται ενσωματωμένος στο κύκλωμα του αισθητήρα BME280, βασίζεται στην ανίχνευση των μεταβολών της χωρητικότητας ενός διηλεκτρικού υλικού. Πιο συγκεκριμένα, περιλαμβάνει ένα ζεύγος ηλεκτροδίων, τα οποία συζευγμένα μεταξύ τους, σχηματίζουν έναν πυκνωτή και διαχωρίζονται από ένα διηλεκτρικό υλικό, η διηλεκτρική σταθερά του οποίου μεταβάλλεται συναρτήσει της απορροφούμενης υγρασίας. Το διηλεκτρικό στρώμα σχηματίζεται από υλικά, όπως τα πολυμερή, τα οποία είναι διαμορφωμένα έτσι ώστε το στρώμα αυτό να

απορροφά και να συγκρατεί μόρια νερού, σε συγκεντρώσεις που είναι ανάλογες της περιβάλλουσας υγρασίας. Τα μόρια του νερού μεταβάλλουν τη διηλεκτρική σταθερά του πολυμερούς, προκαλώντας έτσι τη μεταβολή της χωρητικότητας μεταξύ των δυο ηλεκτροδίων. Με αυτόν τον τρόπο, η τιμή της υγρασίας μπορεί να καθοριστεί από τη μέτρηση της μεταβολής της χωρητικότητας μεταξύ των δύο ηλεκτροδίων.

Η λήψης της τιμής της θερμοκρασίας γίνεται από thermistors συνδεδεμένα σε κύκλωμα γέφυρας, τα οποία είναι ενσωματωμένα στο κύκλωμα του BME280 και η δομή τους περιλαμβάνει τον πυκνωτή του αισθητήρα υγρασίας. Επομένως και η τιμή της θερμοκρασίας εξαρτάται από τη μεταβολή της χωρητικότητας μεταξύ των ηλεκτροδίων. Τα βασικά πλεονεκτήματα από τη χρήση και τη μελέτη των μεταβολών των στοιχείων, όπως ο πυκνωτής, είναι ο χαμηλός θόρυβος, η υψηλή ακρίβεια και η χαμηλή κατανάλωση ισχύος κατά τη μέτρηση της υγρασίας και επομένως και της θερμοκρασίας.

Ο αισθητήρας της πίεσης από την άλλη, λαμβάνει τιμές σύμφωνα με το πιεζοηλεκτρικό φαινόμενο. Σύμφωνα με το φαινόμενο αυτό, ένα πιεζοηλεκτρικό υλικό παράγει ηλεκτρικό ρεύμα, όταν υποστεί μια μηχανική παραμόρφωση (τάση) και αντίθετα σε ένα τέτοιο υλικό προκύπτει μηχανική παραμόρφωση, όταν ασκηθεί σε αυτό ηλεκτρικό πεδίο. Ο αισθητήρας πίεσης, που είναι ενσωματωμένος στο κύκλωμα του BME280, περιλαμβάνει μια εύκαμπτη μεμβράνη, αιωρούμενη πάνω από μια κοιλότητα σε ένα υπόστρωμα ημιαγωγού, η οποία παραμορφώνεται από τη διαφορά πίεσης μεταξύ των δυο πλευρών, λόγω της απορρόφησης των μορίων του νερού. Τα ευαίσθητα στοιχεία πιεζοηλεκτρικού τύπου, σχηματίζουν γέφυρα Wheatstone, συνδέονται με τις επιφάνειες της μεμβράνης και επιτρέπουν έτσι την ανίχνευση του βαθμού παραμόρφωσης και επομένως την τιμή της πίεσης.

5.2 Λήψη δεδομένων με το Raspberry Pi

Ο αισθητήρας λειτουργεί και με τα δυο πρωτόκολλα επικοινωνίας SPI και I^2C . Στην παρούσα εργασία χρησιμοποιήθηκε το πρωτόκολλο I^2C , στο οποίο, όπως αναφέρθηκε και στο κεφάλαιο 4, κάθε slave συσκευή έχει μια μοναδική I^2C διεύθυνση. Στην προκειμένη περίπτωση ο αισθητήρας BME έχει εξ' ορισμού τη διεύθυνση **0x77**.

Ο αισθητήρας παρέχει τη δυνατότητα να αλλάξει διεύθυνση I^2C από 0x77 σε **0x76** γειώνοντας το pin SDI (εικόνα 5.13). Το γεγονός αυτό είναι αρκετά σημαντικό βάσει των πλεονεκτημάτων του αισθητήρα, καθώς μπορούμε να χρησιμοποιήσουμε δύο αισθητήρες BME280 για τη λήψη μετρήσεων ταυτόχρονα, ο ένας με διεύθυνση 0x77 και ο δεύτερος με διεύθυνση 0x76.

Η συνδεσμολογία των αισθητήρων BME280 με το Raspberry Pi για τις δύο δυνατές περιπτώσεις I2C addresses φαίνεται στους πίνακες και τις αντίστοιχες εικόνες που ακολουθούν:

- Ένας αισθητήρας BME280 με I2C address 0x77

Vin	GND	SCK	SDI
3V3	GND	PIN 5	PIN 3

Table 5.2: Συνδεσμολογία BME280 - I2C address 0x77

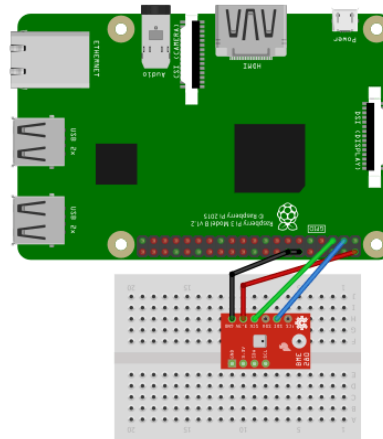


Figure 5.12: Συνδεσμολογία αισθητήρα BME280 με το Raspberry Pi - I2C address 0x77

- Δύο αισθητήρες BME280 με I2C addresses 0x76 - 0x77

Vin	GND	SCK	SD0	SDI
3V3	GND	PIN 5	GND	PIN 3

Table 5.3: Συνδεσμολογία BME280 - I2C address 0x76

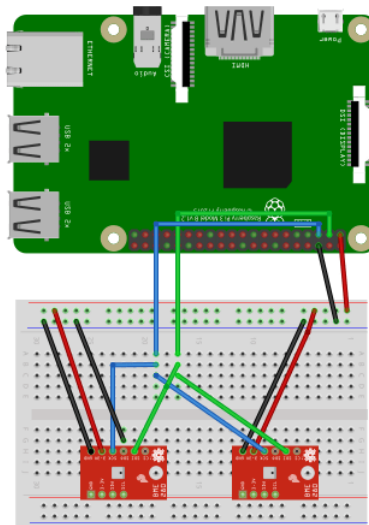


Figure 5.13: Συνδεσμολογία δυο αισθητήρων BME280 με το Raspberry Pi, ο πρώτος αριστερά έχει I2C address 0x76 ενώ ο δεύτερος 0x77

Στην εικόνα 5.14 που ακολουθεί φαίνεται το αποτέλεσμα της εκτέλεσης της εντολής ανίχνευσης διευθύνσεων I^2C , `i2cdetect -y 1` στην περίπτωση που είναι συνδεδεμένοι και οι δυο αισθητήρες.

```

pi@raspberrypi: ~
root@raspberrypi:/home/pi# i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  -- 76 77
root@raspberrypi:/home/pi#

```

Figure 5.14: Εντολή ανίχνευσης ‘`i2cdetect -y 1`’ I2C addresses 0x76 - 0x77

Για τη λήψη μετρήσεων χρησιμοποιήθηκε η βιβλιοθήκη της **Adafruit** σε Python 2.7. Αφού πραγματοποιηθεί η λήψη της από το ‘github.com’ και ακολουθήσουν οι εντολές `install` και `build`, κάνοντας **import** την συγκεκριμένη βιβλιοθήκη σε ένα αρχείο `python` και δηλώνοντας τον αισθητήρα είναι δυνατή η λήψη μετρήσεων. Ακολουθεί ένα παράδειγμα `python` αρχείου για τη λήψη και εμφάνιση δεδομένων θερμοκρασίας, πίεσης και υγρασίας ανα 1 λεπτό για 15 λεπτά.

```

1  from Adafruit_BME280 import *
2  import time
3  sensor = BME280(mode=BME280_OSAMPLE_8)
4  for i in range(1, 16, 1):
5      degrees = sensor.read_temperature()
6      pascals = sensor.read_pressure()
7      mbar=pascals/100
8      humidity = sensor.read_humidity()
9      print ( 'Temp      = {0:0.3 f} deg C'.format(degrees))
10     print ( 'Pressure  = {0:0.2 f} Pa'.format(mbar))
11     print ( 'Humidity  = {0:0.2 f} '.format(humidity))
12     time.sleep(60)
13

```

Listing 5.1: BME280 example

Στα διαγράμματα που ακολουθούν φαίνονται οι τιμές των παραμέτρων θερμοκρασίας, πίεσης και υγρασίας κάθε BME280 συναρτήσει του χρόνου, για μια εβδομάδα συνεχούς λειτουργίας και λήψης δεδομένων.

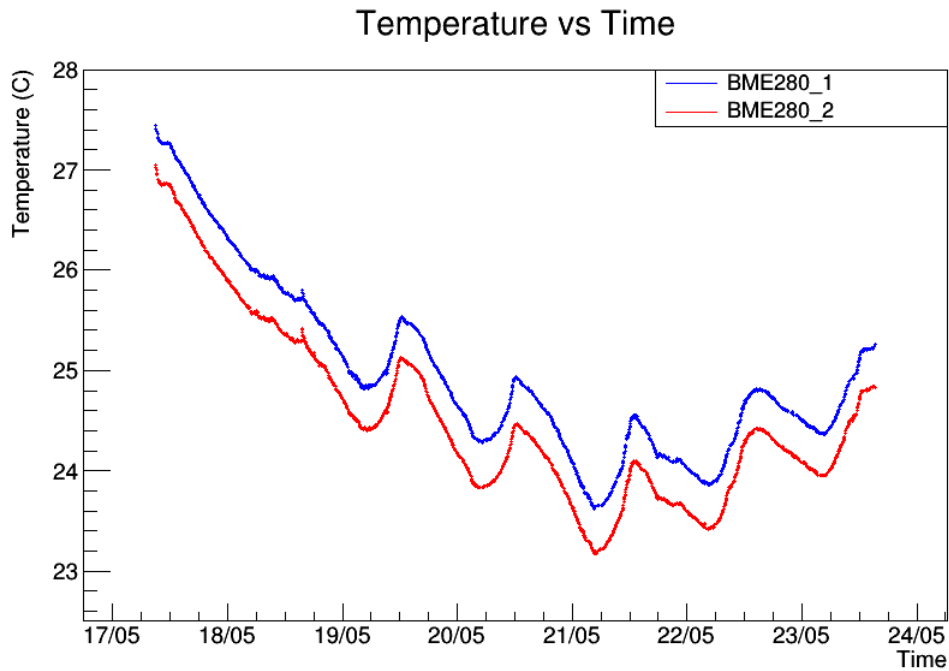


Figure 5.15: Διάγραμμα θερμοκρασίας συναρτήσει του χρόνου

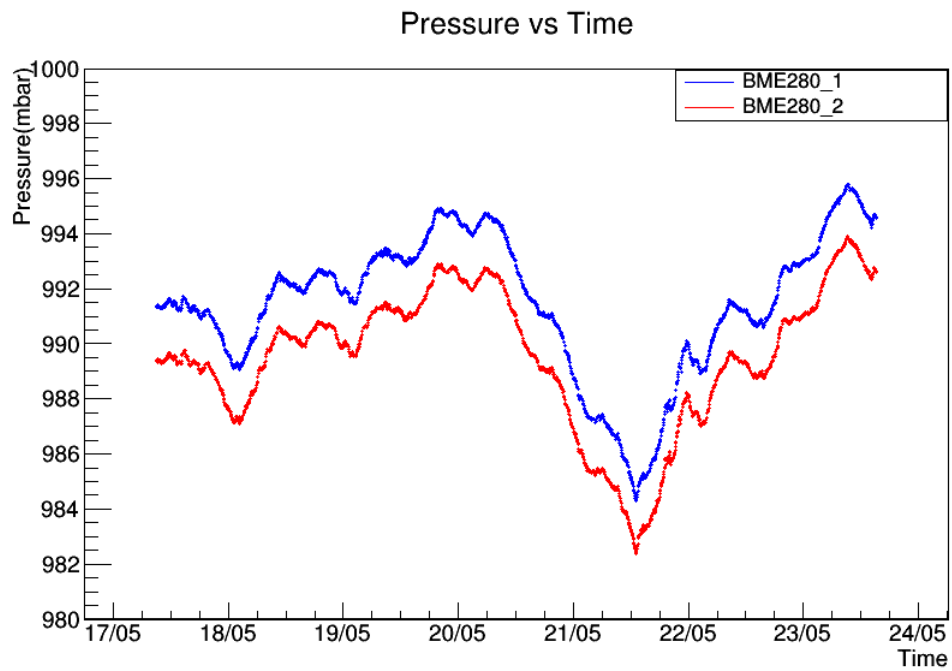


Figure 5.16: Διάγραμμα πίεσης συναρτήσει του χρόνου

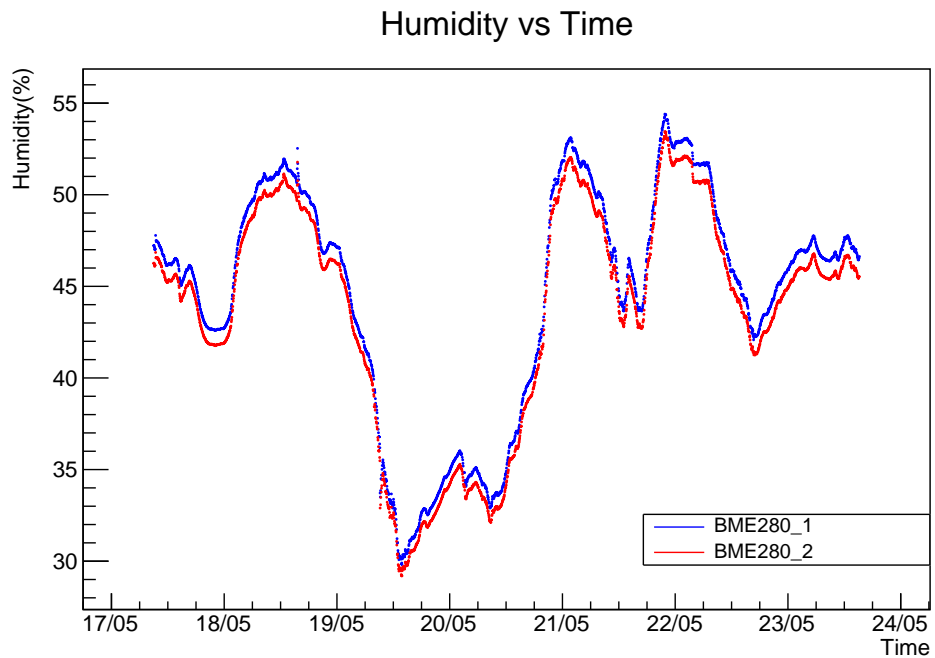


Figure 5.17: Διάγραμμα υγρασίας συναρτήσει του χρόνου

5.2.1 Αποστολή δεδομένων σε web server

Στο κεφάλαιο 2 πραγματοποιήθηκε μια περιληπτική αναφορά στον Apache HTTP ή απλά Apache web server, στη γλώσσα προγραμματισμού για τη δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο PHP, καθώς και στη γλώσσα σήμανσης ιστοσελίδων HTML. Στο κεφάλαιο 3 περιγράφηκε η διαδικασία που ακολουθείται για το configuration του web server και της PHP στο Raspberry Pi. Στην ενότητα που ακολουθεί παρατίθεται η ιστοσελίδα που δημιουργήθηκε με αυτά τα εργαλεία σε συνδυασμό με την γλώσσα προγραμματισμού Python, για την εμφάνιση των δεδομένων που συλλέγει ο μικροϋπολογιστής από τους αισθητήρες BME280. Αξίζει να σημειωθεί, πως η ιστοσελίδα αυτή δημιουργήθηκε για εμφάνισή της μέσω του Chromium. Το Chromium είναι συμβατό το browser Google Chrome και αποτελεί τον προτεινόμενο web browser για χρήση στο Raspberry Pi, .

Ο χρήστης μπορεί να επισκευθεί την ιστοσελίδα είτε από το δικό του προσωπικό υπολογιστή, γνωρίζοντας τη διεύθυνση IP που έχει λάβει το Raspberry Pi, πληκτρολογώντας στην αναζήτηση του browser **http://192.168.x.x/file_name.php**. Επίσης, μπορεί να επισκευθεί την ιστοσελίδα από το browser του Raspberry Pi, είτε μέσω VNC Viewer είτε απευθείας με σύνδεση οθόνης. Η αναζήτηση γίνεται πληκτρολογώντας **http://localhost/file_name.php**, όπου file_name.php και στις δύο περιπτώσεις είναι το όνομα του αρχείου php που περιέχει τον κώδικα της ιστοσελίδας και βρίσκεται στο directory /var/www/html.

Ο χρήστης αρχικά επιλέγει τον αριθμό των αισθητήρων BME280 που είναι συνδεδεμένοι με το Raspberry Pi και στη συνέχεια πατάει το κουμπί **Submit**. Έτσι ανάλογα με την επιλογή του ανοίγει η αντίστοιχη σελίδα, που εμφανίζει σε πίνακα τις παραμέτρους με τις αντίστοιχες τιμές. Οι σελίδες αυτές, ανανεώνονται αυτόματα κάθε 2 λεπτά. Τέλος ο χρήστης έχει την επιλογή **Return** με την οποία επιστρέφει στην αρχική ιστοσελίδα.

Οι δυνατότητες του μικροϋπολογιστή σε συνδυασμό με τις απεριόριστες δυνατότητες που παρέχουν τα εργαλεία προγραμματισμού (ισχυρές γλώσσες υψηλού επιπέδου, βάσεις δεδομένων κλπ) και το διαδίκτυο, αποτελούν ένα σημαντικό εργαλείο για το χρήστη, ώστε να δημιουργήσει ένα πλήρες οικιακό Weather Station και όχι μόνο με απεριόριστες δυνατότητες εμφάνισης, αποθήκευσης και ανάκτησης των δεδομένων.

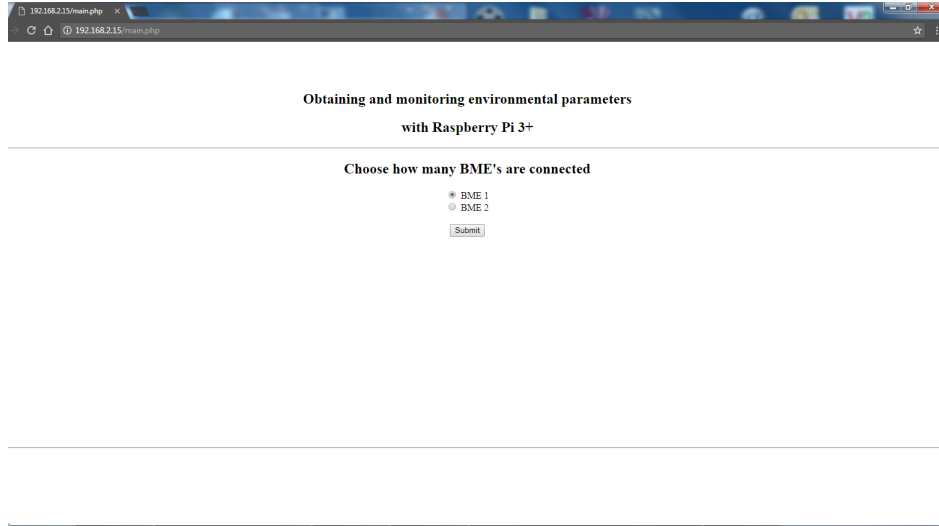


Figure 5.18: Αρχική σελίδα

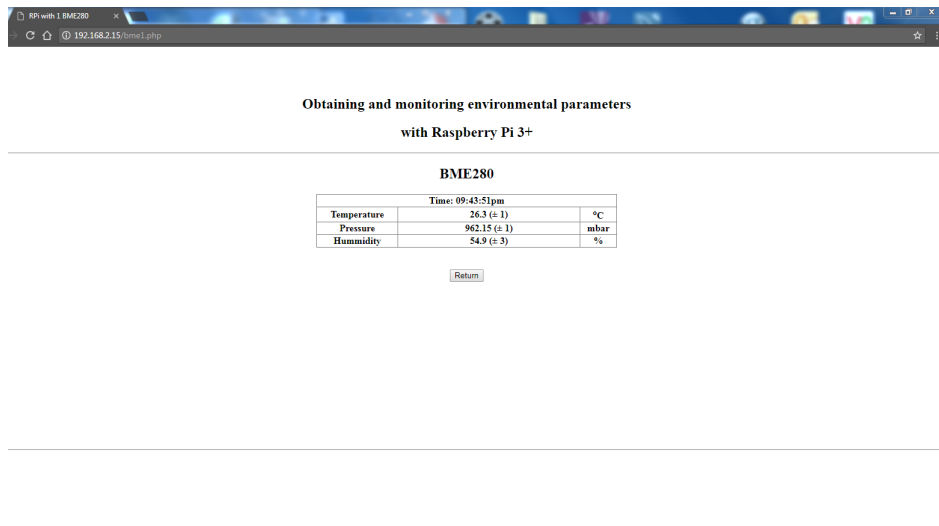


Figure 5.19: Ο χρήστης επιλέγει 1 BME280 και με το Submit επιστρέφει η σελίδα όπου εμφανίζει τα δεδομένα που συλλέγονται από έναν αισθητήρα.

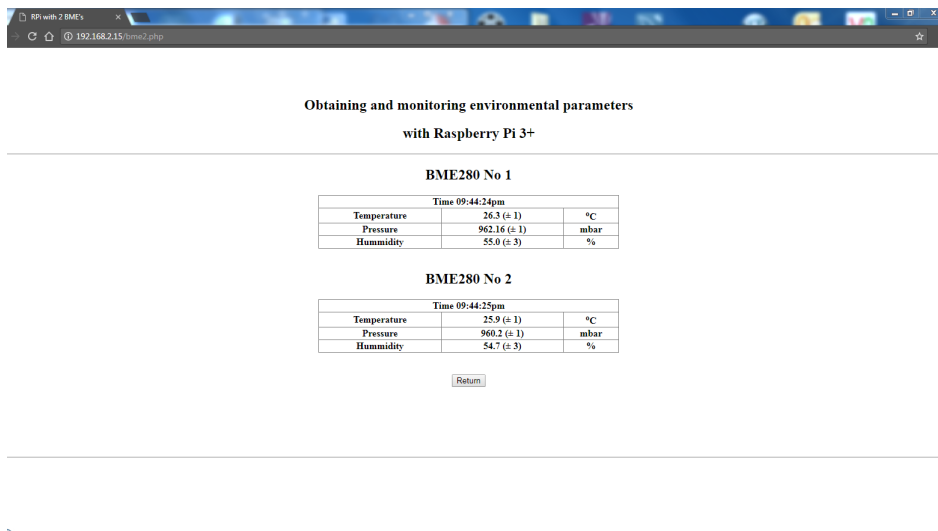


Figure 5.20: Ο χρήστης επιλέγει 2 BME280 και με το Submit επιστρέφει η σελίδα όπου εμφανίζει τα δεδομένα που συλλέγονται από δυο αισθητήρες.

Κεφάλαιο 6

OPC Server

Στο κεφάλαιο αυτό γίνεται περιγραφή του πρωτοκόλλου και των προδιαγραφών επικοινωνίας **OPC** (Open Platform Communication) και συγκεκριμένα αναλύεται το πρωτόκολλο OPC UA. Τέλος, παρουσιάζεται ο OPC UA Raspberry Pi Server, που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας.

6.1 Πρωτόκολλο επικοινωνίας OPC

OPC (Open Platform Communication) καλείται μια σειρά προδιαγραφών και πρωτοκόλλων, που επιτρέπουν την επικοινωνία μεταξύ λογισμικού (πχ Windows) και εξωτερικών συσκευών, η οποία πραγματοποιείται σε ζεύγη διακομιστή (**Server**) και πελάτη (**Client**). Οι προδιαγραφές δημιουργήθηκαν για πρώτη φορά το 1996 και ονομάζονταν **OLE** (Object Linking and Embedding) **for Process Control (OPC)**. Οι προδιαγραφές OPC βασίστηκαν επίσης στα πρωτόκολλα COM και DCOM. Όλα αυτά τα πρωτόκολλα και οι προδιαγραφές αναπτύχθηκαν από τη Microsoft για την 'οικογένεια' λογισμικών Microsoft Windows, με στόχο τη σύνδεση και ενσωμάτωση αρχείων και αντικειμένων. Στην εικόνα που ακολουθεί φαίνεται η δομή και η λογική του πρωτοκόλλου επικοινωνίας OPC.

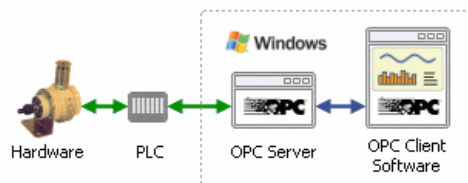


Figure 6.1: Δομή πρωτοκόλλου OPC

Το OLE είχε σκοπό την αλληλεπίδραση με το λογισμικό **SCADA** (Supervisory Control and Data Acquisition) με ένα **PLC** (Programmable Logic Controller) που ελέγχει μια συσκευή, εικόνα 5.2. Ο όρος SCADA (Supervising Control and Data Acquisition) περιγράφει μια κατηγορία συστημάτων βιομηχανικού αυτομάτου ελέγχου και τηλεμετρίας. Το χαρακτηριστικό των συστημάτων SCADA είναι ότι αποτελούνται από τοπικούς ελεγκτές, που ελέγχουν επί μέρους στοιχεία και μονάδες μιας εγκατάστασης, συνδεδεμένους σε ένα κεντρικό Master Station, ο οποίος μπορεί κατόπιν να επικοινωνεί τα δεδομένα που συλλέγει από την εγκατάσταση σε ένα πλήθος από σταθμούς εργασίας σε τοπικό LAN ή και να μεταδίδει τα δεδομένα της εγκατάστασης σε μακρινά σημεία μέσω κάποιου συστήματος τηλεπικοινωνίας, πχ μέσω του ενσύρματου τηλεφωνικού δικτύου ή μέσω κάποιου ασύρματου δικτύου.

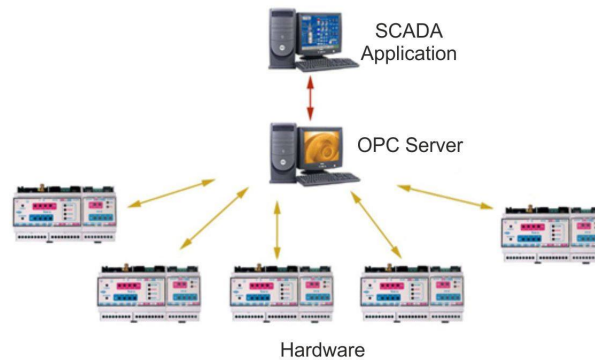


Figure 6.2: Δομή λήψης και διαχείρισης δεδομένων από hardware σε σύστημα SCADA μέσω OPC Server

Οι προδιαγραφές OPC εξελίχθηκαν αρκετά από το OLE, και καθορίζουν πλέον ένα συγκεκριμένο και ευρύ σύνολο αντικειμένων, interfaces, και μεθόδων για χρήση σε εφαρμογές ελέγχου και αυτοματοποίησης. Η πιο γνωστή μορφή του OPC και η πιο διαδεδομένη μέχρι τώρα, είναι το **OPC Data Access (OPC DA)**, το οποίο χρησιμοποιείται για να γράφει και να διαβάζει σε πραγματικό χρόνο (real time data). Πέρα από το OPC DA υπάρχουν και οι προδιαγραφές OPC Historical Data Access (**HDA**) που παρέχουν τη δυνατότητα πρόσβασης και ανάκτησης αρχειοθετημένων δεδομένων. Όλα αυτά τα πρωτόκολλα και οι προδιαγραφές διατηρούνται και υποστηρίζονται από τον οργανισμό **OPC Foundation**.

6.1.1 Design

Όπως αναφέρθηκε το πρωτόκολλο OPC πραγματοποιείται σε ζεύγη Server και Client και ουσιαστικά σχεδιάστηκε για να παρέχει μια 'γέφυρα' (bridge) μεταξύ των εφαρμογών με λογισμικό Windows και του hardware των διαδικασιών ελέγχου και αυτοματισμού. Τα πρότυπα αυτά ορίζουν συνεπείς μεθόδους πρόσβασης σε δεδομένα, που συλλέγονται από τις εξωτερικές συσκευές που ελέγχουν το hardware. Οι μέθοδοι εί-

να συνεπείς καθώς είναι ανεξάρτητες από τον τύπο και την πηγή των δεδομένων. Έτσι δηλαδή, ένας OPC Server για την εξωτερική συσκευή, που συλλέγει τα δεδομένα από το hardware, παρέχει τις ίδιες μεθόδους και εντολές τόσο σε έναν ή και ,περισσότερους OPC Clients ώστε να έχουν πρόσβαση σε αυτά τα δεδομένα όσο και σε κάθε άλλον OPC Server.

OPC Client μπορεί να είναι ένα οποιοδήποτε πρόγραμμα, το οποίο θέλει να επικοινωνήσει με μία συσκευή. Ο ρόλος του OPC Server είναι να μετατρέπει το αίτημα αυτό του πελάτη σε κατάλληλη μορφή, την οποία η εξωτερική συσκευή μπορεί να κατανοήσει. Με τον τρόπο αυτό, καθίσταται δυνατή η επικοινωνία του client και της συσκευής, με τον client να στέλνει εντολές μέσω του server ή με το server να διαβάζει δεδομένα από την εκάστοτε εξωτερική συσκευή και να τα επικοινωνεί στον Client.

Το μεγαλύτερο πλεονέκτημα που εμφανίζει το πρωτόκολλο επικοινωνίας OPC, είναι η ευκολία, που δίνει στο χρήστη να κατασκευάσει με πολύ απλό τρόπο ένα Server ή ένα Client, έχοντας στη διαθεσή του μόνο τα απαραίτητα στοιχεία για τη διασύνδεση μεταξύ τους. Οι προδιαγραφές είναι κοινές για όλους τους Client και Server, γεγονός το οποίο επιτρέπει σε οποιοδήποτε Client, να συνδεθεί με οποιοδήποτε Server. Το κόστος για τους κατασκευαστές διατηρείται χαμηλό, καθώς απαιτείται ένας μόνο Server και έτσι η συσκευή μπορεί να επικοινωνήσει με οποιοδήποτε λογισμικό που παρέχει σύνδεση με OPC. Οι προμηθευτές λογισμικού συμπεριλαμβάνουν έναν Client στο πρόγραμμα τους, δίνοντάς του τη δυνατότητα να συνδεθεί με οποιαδήποτε συσκευή διαθέτει OPC Server.

Από την άλλη όμως, εμφανίζει και αρκετά προβλήματα όσον αφορά την ασφάλεια (security) αλλά κυρίως την εξάρτησή του από τις πλατφόρμες της Microsoft και την αναγκαιότητα ανάπτυξής τους σε λογισμικό Windows. Δυνατότητα επικοινωνίας δηλαδή, βάσει του κλασσικού πρωτοκόλλου OPC υπάρχει μόνο σε περιβάλλοντα ανάπτυξης Windows, Server και Client πρέπει να είναι εγκατεστημένοι σε αυτό. Το γεγονός αυτό σε συνδυασμό με τις όλο και αυξανόμενες απαιτήσεις για χρήση του πρωτοκόλλου OPC σε εφαρμογές αυτοματισμού και ελέγχου σε διαφορετικά περιβάλλοντα ανάπτυξης αποτέλεσε πρόβλημα, καθώς έθετε σημαντικούς περιορισμούς. Λύση σε αυτό το πρόβλημα δόθηκε το 2008 με την ανάπτυξη του πρωτοκόλλου επικοινωνίας **OPC UA (Unified Architecture)**. Το πρωτόκολλο αυτό χρησιμοποιήθηκε στην παρούσα διπλωματική εργασία και αναλύεται στο κεφάλαιο που ακολουθεί.

6.2 OPC Unified Architecture (UA)

Το **OPC UA (Unified Architecture)** αποτελεί ένα πρωτόκολλο επικοινωνίας 'machine to machine' για βιομηχανικές εφαρμογές αυτοματισμού και ελέγχου. Επικοινωνία 'machine to machine' καλείται η απευθείας ενσύρματη και ασύρματη επικοινωνία μεταξύ συσκευών. Συμπεριλαμβάνονται βιομηχανικές συσκευές, που επιτρέπουν σε αισθητήρες και γενικότερα άλλα όργανα μέτρησης, να επικοινωνήσουν τα δεδομένα, που καταγράφουν (πχ θερμοκρασία) σε λογισμικό εφαρμογής που μπορεί

να τα χρησιμοποιήσει. Τέτοια επικοινωνία επιτυγχάνεται με ένα απομακρυσμένο δίκτυο (remote network) των συσκευών αυτών που μεταδίδουν πληροφορίες σε έναν κεντρικό κόμβο για ανάλυση.

Είναι η εξέλιξη του κλασσικού μέχρι τώρα πρωτοκόλλου OPC, το οποίο αναπτύχθηκε από την OPC Foundation. Εμφανίζει όμως σημαντικές διαφορές από τον προκάτοχό του, καθώς αποτελεί μια βελτιωμένη έκδοσή του, η οποία ήρθε να δώσει λύση στα διάφορα προβλήματα που αντιμετωπίστηκαν με τη χρήση αυτού. Η αρχιτεκτονική του πρωτοκόλλου OPC UA είναι υπηρεσιοστραφής αρχιτεκτονική (Service-Oriented Architecture (SOA)) και βασίζεται σε διαφορετικά λογικά επίπεδα. Η αρχιτεκτονική αυτή επεκτείνει την ασφάλεια (security) και τη λειτουργικότητα (functionality) του προκάτοχου του, του οποίου η αρχιτεκτονική όπως αναφέρθηκε βασίζεται σε τεχνολογίες COM/DCOM.

Επίσης, το OPC UA υποστηρίζει δύο πρωτόκολλα επικοινωνίας: ένα δυαδικό (binary) πρωτόκολλο: **opc.tcp://Server** και ένα πρωτόκολλο εφαρμογών διαδικτύου (Web Service (SOAP)): **http://Server**. Το πρώτο προσφέρει την καλύτερη απόδοση και απαιτεί τους λιγότερους πόρους (δεν απαιτείται XML Parser, SOAP και HTTP, το οποίο είναι σημαντικό για τις ενσωματωμένες συσκευές). Επίσης, προσφέρει την καλύτερη διαλειτουργικότητα, επιτρέπει την εύκολη ενεργοποίηση διαμέσου ενός τοίχου προστασίας (firewall), χρησιμοποιώντας μόνο μία αυθαίρετα επιλέξιμη θύρα TCP. Από την άλλη, το πρωτόκολλο Web Service (SOAP) υποστηρίζεται καλύτερα από τα διαθέσιμα εργαλεία (π.χ. από τα περιβάλλοντα Java ή .NET) και χρησιμοποιεί συγκεκριμένες θύρες HTTP/HTTPS. Στη συνέχεια, αναλύονται οι καινοτομίες, που προσφέρει το πρωτόκολλο OPC UA έναντι του κλασσικού:

- **Ανεξάρτητο από την πλατφόρμα επικοινωνίας**

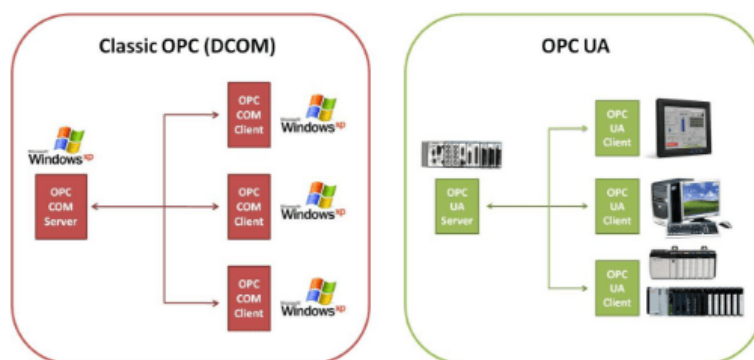


Figure 6.3: OPC UA vs Classic OPC

Δεν υπάρχει πλέον αυστηρός περιορισμός και εξάρτηση από λειτουργικά συστήματα της Microsoft (Windows), καθώς το πρωτόκολλο OPC UA παρακάμπτει την εξάρτηση αυτή. OPC UA Server και Client σε διαφορετικά λειτουργικά συστήματα μπορούν να επικοινωνήσουν.

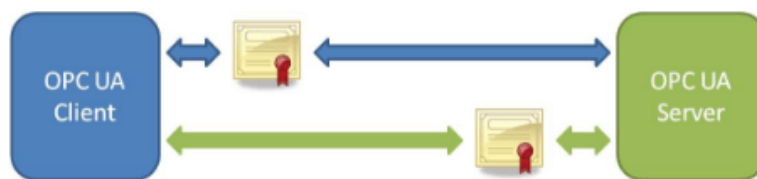


Figure 6.4: OPC UA Security

- **Διευρυμένη ασφάλεια**

Ένα από τα σημαντικότερα οφέλη της εξάλειψης της εξάρτησης από την τεχνολογία COM/DCOM είναι τα διευρυμένα χαρακτηριστικά ασφαλείας. Τα κλασικά συστήματα OPC βασίζονται στη δύσκολη και περίπλοκη διαμόρφωση του DCOM για την παροχή ασφαλείας μεταξύ των διαδικασιών καθώς, οι προγραμματιστές πρέπει να χρησιμοποιούν λίστες Access Control, που είναι αποθηκευμένες σε ρυθμίσεις DCOM για να διαμορφώσουν τις ρυθμίσεις ασφαλείας για κάθε στοιχείο. Αυτό συχνά οδηγεί στην αδυναμία της ασφαλείας και συνεπώς σε μεγάλα κενά ασφαλείας στο δίκτυο. Σε αντίθεση, το OPC UA χρησιμοποιεί τις τυποποιημένες τεχνολογίες ιστού ως βάση ασφαλείας που περιλαμβάνει ταυτόχρονα τις δυνατότητες ελέγχου ταυτότητας και κρυπτογράφησης για την προστασία δεδομένων. Συγκεκριμένα, υποστηρίζει πρότυπα κρυπτογραφίας δημόσιου κλειδιού (Public-Key Cryptography Standards PK CS12) για την παροχή των ιδιωτικών κλειδιών X.509 και των αρχείων πιστοποιητικών, που περιέχουν δημόσια κλειδιά. Τόσο ο Server όσο και ο Client μπορούν να επιλέξουν ποιο ζεύγος δημόσιων κλειδιών και ιδιωτικών κλειδιών θα χρησιμοποιήσουν. Επιπλέον, ο χρήστης μπορεί να ενεργοποιήσει μία από τις δύο πολιτικές ασφαλείας: Basic256 και Basic128Rsa15, οι οποίες είναι οι βάσεις για τον αλγόριθμο υπογραφής ή κρυπτογράφησης των δεδομένων μεταξύ του Client και Server.

Ως άμεσο αποτέλεσμα του τυποποιημένου μοντέλου ασφαλείας, το OPC UA επιτρέπει την εύκολη ενσωμάτωση σε προϋπάρχοντα δίκτυα πληροφορικής (IT networks), γεγονός που περιορίζει το κόστος διαμόρφωσης. Το πρωτόκολλο OPC UA μπορεί να επικοινωνεί δεδομένα μέσω οποιασδήποτε τυποποιημένης θύρας TCP ή HTTP. Έτσι, μέσω αυτής της τυποποίησης, το OPC UA μπορεί να συνδεθεί με ασφάλεια σε ένα VPN και μέσω firewall για να επιτρέψει την απρόσκοπτη σύνδεση Server-Client.

6.2.1 Δομή OPC UA Server - Address Space

Σύμφωνα με όλες τις προδιαγραφές του πρωτοκόλλου OPC UA που περιγράφηκαν μέχρι τώρα, ως **AddressSpace** ενός OPC UA Server ορίζεται το σύνολο αντικειμένων (objects) που επικοινωνεί στους Clients, τα οποία αποτελούν δεδομένα ενός real-time συστήματος. Ο πρωταρχικός στόχος του OPC UA AddressSpace είναι να παρέχει έναν τυπικό τρόπο στους Servers να επικοινωνούν τα objects στους Clients. Το OPC UA Object Model έχει σχεδιαστεί για να ικανοποιεί αυτόν ακριβώς το στόχο, ορίζοντας

τη δομή των objects με μεταβλητές (variables) και μεθόδους (methods), η δομή του οποίου φαίνεται στην εικόνα 6.5.

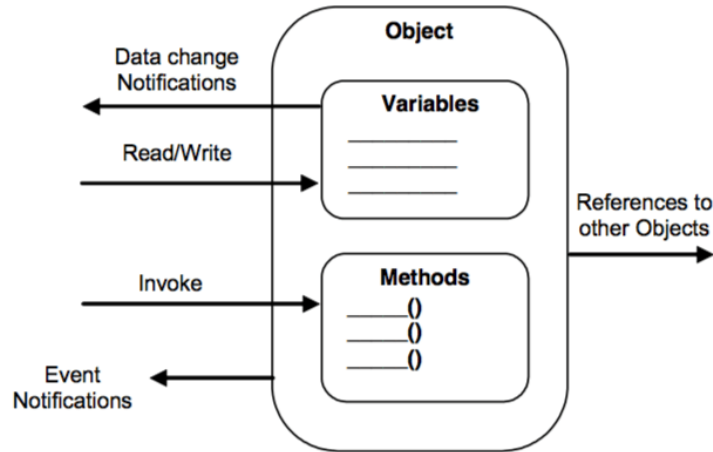


Figure 6.5: OPC UA Server - Object Model

Κάθε ένα από τα στοιχεία αυτού του Object Model στο AddressSpace σε κάθε Server αναπαριστάται ως κόμβος (**node**), οι οποίοι κόμβοι περιγράφονται από χαρακτηριστικά (**Attributes**) και διασυνδέονται μεταξύ τους με αναφορές (**References**). Η εικόνα 6.6 περιγράφει το Node Model.

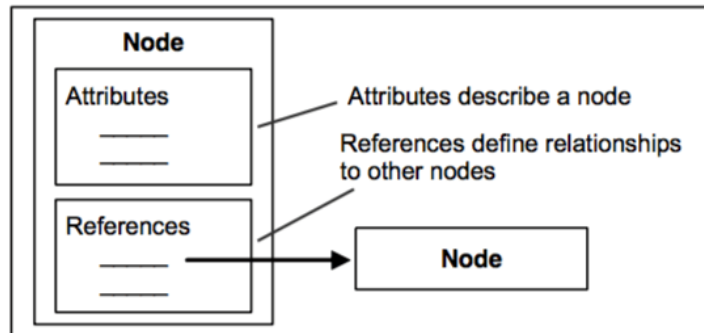


Figure 6.6: OPC UA Server - Node Model

Για τη δημιουργία του AddressSpace πρέπει να ενσωματωθούν τα nodes και να διασυνδεθούν με References. Η ενσωμάτωση των nodes απαιτεί την ανάθεση των κατάλληλων τιμών (values) στα Attributes. Στο Address Space Model κάθε ένα από τα nodes προσδιορίζει ένα NodeClass και κάθε NodeClass αναπαριστά ένα διαφορετικό στοιχείο του Object Model. Τα NodeClasses αποτελούνται από τα Attributes και References, που πρέπει να ενσωματωθούν για να καθοριστεί ένα node στο AddressSpace. Το Address Space Model παρέχει ένα πολύ συγκεκριμένο

σύνολο NodeClasses, το οποίο είναι αυστηρά καθορισμένο και μη επεκτάσιμο. Κάθε ένα από τα NodeClasses παρέχει συγκεκριμένη λειτουργία και αντιπροσωπεύει καλά καθορισμένες πληροφορίες κατά την εκτέλεση.

Αναφορικά τα 8 NodeClasses του Address Space Model είναι:

- **View:** αποτελεί ένα υποσύνολο από nodes στο AddressSpace
- **Object:** περιγράφει συστήματα, εξαρτήματα συστημάτων, real-world και software object
- **Variable:** χρησιμοποιείται ως holder δεδομένων διεργασίας σε πραγματικό χρόνο (παρέχει τιμή)
- **Method:** αποτελεί συνάρτηση η εμβέλεια της οποίας περιορίζεται από ένα συγκεκριμένο Object
- **ObjectType:** προσδιορίζει και περιγράφει το Object
- **DataType:** καθορίζει απλούς και σύνθετους τύπους δεδομένων για την τιμή του Variable
- **ReferenceType:** καθορίζει τη διασύνδεση των nodes
- **VariableType:** προσδιορίζει και περιγράφει το Variable

Τέλος, αξίζει να σημειωθεί πως κάθε node στο AddressSpace ενός OPC UA Server χαρακτηρίζεται από NodeId, που επιτρέπει στους Clients να αναγνωρίζουν/ταυτοποιούν και να έχουν έτσι πρόσβαση σε κάθε node. Η δομή του NodeId είναι η εξής:

- **namespaceIndex:** αριθμητική τιμή που ταυτοποιεί το namespace
- **identifierType:** καθορίζει τον τύπο του NodeId, το format και το εύρος του. Ανάλογα με τις ανάγκες της εφαρμογής μπορεί να είναι ένας από τους έξι τύπους:
 - NUMERIC: αριθμητικό
 - STRING: χαρακτήρας
 - GUID: Globally Unique Identifier
 - OPAQUE: namespace specific format
- **identifier:** μοναδικό αναγνωριστικό στοιχείο στο πλαίσιο του namespace

Ως namespace ορίζεται ένα URI (Unique Resource Identifier), το οποίο καθορίζει την αρχή ονοματολογίας (naming), που είναι υπεύθυνη για την αντιστοίχιση του αναγνωριστικού στοιχείου με το NodeId. Το URI κάθε NodeId καθορίζεται από αριθμητικές τιμές στον OPC UA Server με σκοπό την ομαλή μεταφορά και ταυτοποίηση δεδομένων στον Client.

6.3 OPC UA Raspberry Pi Server

Κατά τη διάρκεια της παρούσας διπλωματικής εργασίας δημιουργήθηκε η ανάγκη ύπαρξης OPC Server έτσι ώστε να γίνει χρήση του πρωτοκόλλου OPC με OPC Client το πρόγραμμα SCADA WinCC. Οι δυνατότητες του μικροϋπολογιστή Raspberry Pi σε συνδυασμό με τα πλεονεκτήματα που παρέχει το πρωτόκολλο OPC UA, οδήγησαν στο να στηθεί OPC UA Server σε Python στο Raspberry Pi, ο οποίος βάσει αυτού του πρωτοκόλλου επικοινωνεί τα δεδομένα των αισθητήρων BME στο WinCC το οποίο παρέχει OPC UA Client. Το project 'FreeOpcUa' παρέχει μια εφαρμογή εξ' ολοκλήρου σε γλώσσα προγραμματισμού Python (συμβατό με τις εκδόσεις 2 και 3) με υψηλού επιπέδου κλάσεις OPC UA Server και OPC UA Client, και παραδείγματα τα οποία είναι δυνατόν να χρησιμοποιηθούν ως έχει ή να προσαρμοστούν στις ανάγκες κάθε χρήστη.

Η εγκατάσταση του πακέτου FreeOpcUa και των βιβλιοθηκών που αυτό παρέχει έγινε με την εντολή:

```
sudo pip install freeopcua
```

Επειδή για την ολοκλήρωση του project της παρούσας διπλωματικής εργασίας χρησιμοποιήθηκε η έκδοση Python 2.7 ακολουθώντας τις οδηγίες εγκατάστασης, έπρεπε να εγκατασταθούν επιπλέον τα πακέτα: enum34, trollius(asyncio) και futures (concurrent.futures). Έτσι, ακολούθησαν οι εντολές:

```
sudo pip install enum34
sudo pip install trollius(asyncio)
sudo pip install futures(concurrent.futures)
```

Κα στη συνέχεια sudo build για κάθε πακέτο.

Όπως αναφέρθηκε προηγουμένως, το OPC UA εφαρμόζει πρότυπα πρωτόκολλα δικτύου, συμπεριλαμβανομένου του ελέγχου ταυτότητας με πιστοποίηση και κρυπτογράφηση των δεδομένων. Για το λόγο αυτό έπρεπε να εγκατασταθεί στο Raspberry Pi το πακέτο 'cryptography' ώστε να λειτουργεί σωστά ο OPC UA Server. Αυτό έγινε με την εξής ακολουθία εντολών:

```
sudo apt-get install build-essential libssl-dev
libffi-dev python-dev
sudo pip install cryptography
sudo build cryptography
```

Η πρώτη εντολή διασφαλίζει την εγκατάσταση των απαιτούμενων εξαρτήσεων για το πακέτο cryptography στο Debian και στη συνέχεια γίνεται η εγκατάστασή του.

Μετά την εγκατάσταση της βιβλιοθήκης `freeopcua` με χρήση `ftp server` έγινε λήψη του πακέτου `python-opcua` από το GitHub στο Raspberry Pi και η δημιουργία OPC UA Server, ο οποίος επικοινωνεί τα δεδομένα που λαμβάνει από τους αισθητήρες BME280, κάνοντας `import` τη βιβλιοθήκη της Adafruit.

Βασική προϋπόθεση για να πληρεί τις προδιαγραφές του πρωτοκόλλου που περιγράφηκαν πρέπει να συμπληρωθούν στον κώδικα τα εξής:

```
1 server = Server()
2 server.set_endpoint("opc.tcp://localhost/freeopcua/server/")
3 server.set_server_name("RaspberryPi")
4 server.load_certificate("certificate-example.der")
5 server.load_private_key("private-key-example.pem")
6 uri = "http://examples.freeopcua.github.io"
7 idx = server.register_namespace(uri)
```

6.3.1 Δομή OPC UA Raspberry Pi Server

Βάσει των όσων έχουν αναφερθεί, η δομή του OPC UA Server είναι συγκεκριμένη. Αρχικά δημιουργείται ένα Object για τον αισθητήρα

```
1 objects = server.get_objects_node()
2 obj = objects.add_object(idx, "BME280")
```

Στη συνέχεια προστίθενται οι μεταβλητές (Variables), που αποτελούνται από τα στοιχεία που μετράει ο αισθητήρας, θερμοκρασία, πίεση, υγρασία:

```
1 temp= obj.add_variable(idx, "temperature", round(degrees,2))
2 pres= obj.add_variable(idx, "pressure", round(mbar,2))
3 hum= obj.add_variable(idx, "humidity", round(humidity,2))
```

Η μέθοδος που χρησιμοποιείται για να θέσει τη real-time τιμή, που λαμβάνει ο αισθητήρας σε κάθε Variable είναι:

```
1 variable.set_value(sensor_value)
```

Τέλος, οι συναρτήσεις έναρξης και λήξης του Server είναι οι εξής:

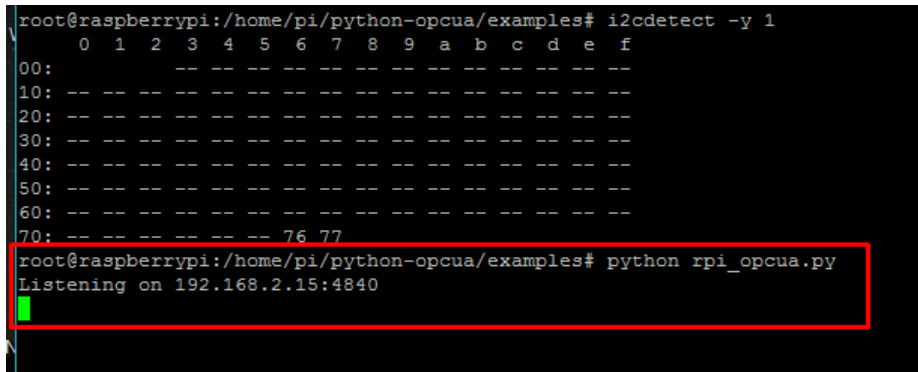
```
1 server.start()
2 server.stop()
```

Τέλος, να σημειωθεί, πως ο OPC UA Raspberry Pi Server είναι δυναμικός server. Δε χρειάζεται ο χρήστης να τροποποιήσει τον κώδικα ανάλογα με τον αριθμό των αισθητήρων που χρησιμοποιούνται κάθε φορά. Όπως έχει αναλυθεί σε προηγούμενο κεφάλαιο, κάθε αισθητήρας βάσει του πρωτοκόλλου επικοινωνίας I2C έχει μια μοναδική διεύθυνση. Στη συγκεκριμένη περίπτωση, ο ένας αισθητήρας BME280 έχει διεύθυνση 0x77 και ο δεύτερος θα έχει 0x76. Έτσι, εκτελώντας εσωτερικά στον κώδικα του server ένα python script, που ανιχνεύει τις I2C διευθύνσεις, ο OPC UA Server επικοινωνεί τα δεδομένα από έναν ή δυο αισθητήρες αντίστοιχα.

Όταν εκτελείται το python script το οποίο περιέχει τον κώδικα του OPC UA Server που υλοποιήθηκε, με την εντολή

```
python rpi_opcua.py
```

ο server είναι έτοιμος. Οι Clients θα συνδεθούν σε αυτόν χρησιμοποιώντας τη διεύθυνση URL (TCP/IP) του **opc.tcp://ip_address:4840**, όπως φαίνεται και στην εικόνα 6.7.



```
root@raspberrypi:/home/pi/python-opcua/examples# i2cdetect -y 1
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  76 77
root@raspberrypi:/home/pi/python-opcua/examples# python rpi_opcua.py
Listening on 192.168.2.15:4840
```

Figure 6.7: Ο OPC UA Server στο Raspberry Pi είναι έτοιμος. Ο χρήστης εκτελεί την εντολή 'python rpi_opcua.py' και ο server είναι σε θέση να συνδεθούν σε αυτόν οι OPC UA Clients και να τους επικοινωνήσει τα δεδομένα που συλλέγει από έναν ή δυο αισθητήρες BME280

Κεφάλαιο 7

WinCC OA

7.1 Εισαγωγή

Το WinCC OA ανήκει στην κατηγορία των προγραμμάτων **SCADA** (Supervising Control and Data Acquisition), τα οποία είναι συστήματα βιομηχανικού αυτόματου ελέγχου. Όπως έχει αναφερθεί στο κεφάλαιο 6, βασικό χαρακτηριστικό αυτών των προγραμμάτων είναι ότι αποτελούνται από έναν κεντρικό σταθμό εργασίας (Master Station) και πολλούς επιμέρους τοπικούς σταθμούς. Οι τοπικοί σταθμοί 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 Πειραματικής Φυσικής Υψηλών Ενεργειών γίνεται ευρεία χρήση των προγραμμάτων SCADA, καθώς ο τακτικός έλεγχος των ανιχνευτικών διατάξεων καθίσταται αναγκαίος. Για τη δημιουργία συστημάτων αυτόματου ελέγχου παρά την πληθώρα προγραμμάτων SCADA, έχει επικρατήσει το WinCC OA. Το γεγονός αυτό είναι απόρροια κυρίως των εξής στοιχείων, αρχικά το περιβάλλον του WinCC OA είναι φιλικό προς το χρήστη, η παραμετροποίηση είναι πολύ εύκολη διαδικασία και τέλος δίνει στο χρήστη τη δυνατότητα εύκολης σύνδεσης εξωτερικών συσκευών για τη συλλογή δεδομένων και την περαιτέρω επεξεργασία αυτών μέσω των πρωτοκόλλων OPC και OPC UA.

7.1.1 Βασικά Χαρακτηριστικά Του WinCC OA

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

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

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 κάθε αντικειμένου πάνω στα panels καθώς και οι λειτουργίες κάθε panel καθορίζονται από κώδικα, που γράφει ο χρήστης στα κείμενα ελέγχου (control scripts).

- **Κείμενα Ελέγχου(Control Scripts)**

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

- **Σύνδεση με Εξωτερικές Συσκευές**

Το WinCC OA δίνει τη δυνατότητα σύνδεσής με εξωτερικές συσκευές, έτσι ο χρήστης έχει τη δυνατότητα να εξάγει πληροφορίες από τις συσκευές που χρησιμοποιεί, όπως για παράδειγμα στην παρούσα εργασία ο χρήστης έχει τη δυνατότητα να εξάγει πληροφορίες για τις τιμές των περιβαλλοντικών παραμέτρων, που μετρούν οι αισθητήρες, οι οποίοι είναι συνδεδεμένοι με το Raspberry Pi. 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. Το Raspberry Pi, για παράδειγμα, χρησιμοποιεί το πρωτόκολλο OPC UA για την επικοινωνία του με το WinCC OA, καθώς το τελευταίο διαθέτει έναν OPC UA Client, ο οποίος συνδέεται με τον OPC UA Server του Raspberry Pi. Με αυτόν τον τρόπο, το Raspberry Pi επικοινωνεί τα δεδομένα, που συλλέγει στο WinCC OA. Τέλος, ο χρήστης είναι αυτός που επιλέγει τον τρόπο με τον οποίο τα δεδομένα αυτά θα διαχειριστούν, εφόσον έχει τη δυνατότητα να τα αποθηκεύσει στη βάση δεδομένων του WinCC OA.

- **Βάση Δεδομένων (Database)**

Το WinCC OA διαθέτει μία εσωτερική βάση δεδομένων, στην οποία αποθηκεύονται τα δεδομένα, που συλλέγονται από τις εξωτερικές συσκευές, με τις οποίες αυτό επικοινωνεί βάσει συγκεκριμένου πρωτοκόλλου. Τα δεδομένα ομαδοποιούνται σε **Datapoints (DP)** ενός ήδη προκαθορισμένου είδους Datapoint, το **Data Point Type (DPT)**.

Το είδος των δεδομένων (DPT) περιγράφει τη δομή δεδομένων ενός συγκεκριμένου είδους συσκευής. Τα Datapoints, που δημιουργούνται, περιέχουν τα

δεδομένα και τις πληροφορίες για μία συγκεκριμένη συσκευή, τα οποία αντιπροσωπεύονται από ένα **Datapoint Element (DPE)**, που περιέχεται στο DP.

Για κάθε DPE πέρα από την τιμή (value) υπάρχουν περαιτέρω πληροφορίες που μπορεί ο χρήστης να αποθηκεύσει. Οι επιπλέον αυτές πληροφορίες αποτελούν τα configs κάθε DPE, τα οποία ο χρήστης πρέπει να εισάγει ανάλογα με τις απαιτήσεις του project. Ένα από αυτά είναι ο καθορισμός της αντίστοιχης περιφεριακής διεύθυνσης διασύνδεσης (periphery address), που εξαρτάται από το πρωτόκολλο βάσει του οποίου γίνεται η επικοινωνία. Ένα ακόμα είναι η οριοθέτηση ενός εύρους τιμών ‘ασφαλούς λειτουργίας’, έτσι ώστε ο χρήστης να ειδοποιείται, όταν η τιμή είναι εκτός αυτού του ορίου (alarm). Τέλος, μπορεί επιπλέον να καθοριστεί πότε και αν η τιμή θα αρχειοθετείται ή όχι (archive), καθώς και αν θα εξομαλύνεται η τιμή της (smoothe).

Συγκεκριμένα, το Raspberry Pi είναι συνδεδεμένο με 2 αισθητήρες BME280, οι οποίοι μετράνε θερμοκρασία, πίεση και υγρασία. Το Raspberry_Pi αποτελεί ένα DataPoint Type. Έστω ότι έχει δημιουργηθεί ήδη ένα DataPoint με όνομα RPi, το οποίο είναι τύπου Raspberry_Pi. Το DP δηλαδή ακολουθεί τη δομή του DPT Raspberry_Pi. Αυτό σημαίνει, πως θα παρέχει πληροφορίες και τιμές για τους αισθητήρες με τους οποίους είναι συνδεδεμένο. Σε περίπτωση που ο χρήστης προσθέσει ένα επιπλέον Raspberry Pi, που λαμβάνει τιμές από τους ίδιους αισθητήρες, θα πρέπει να δημιουργήσει ακόμα ένα DP τύπου Raspberry_Pi. Στην περίπτωση, όμως που προσθέσει ένα Raspberry Pi, που λαμβάνει τιμές από διαφορετικούς αισθητήρες, πρέπει να δημιουργήσει ένα καινούργιο DPT.

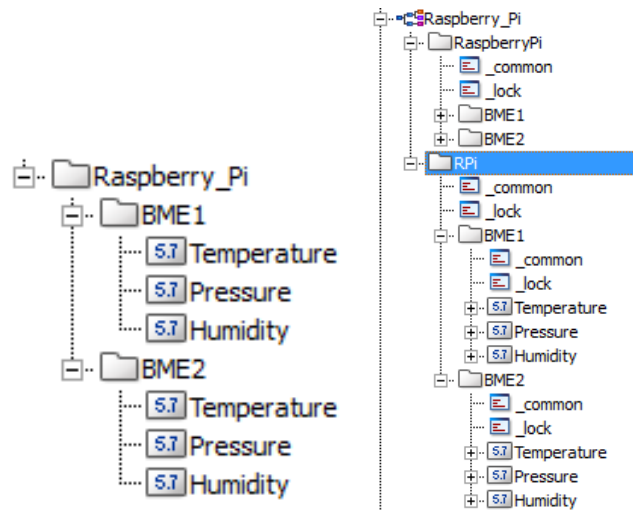


Figure 7.1: Στις εικόνες εμφανίζεται η δομή του DataPoint Type Raspberry_Pi, το οποίο περιγράφει τη δομή δεδομένων της συσκευής, άρα περιέχει ένα node για κάθε αισθητήρα BME280 και ένα για Temperature, Pressure και Humidity. Όταν δημιουργηθεί ένα DataPoint με όνομα RPi, το οποίο είναι τύπου Raspberry_Pi, θα ακολουθεί τη δομή του DPT Raspberry_Pi. Σε περίπτωση που ο χρήστης προσθέσει ένα επιπλέον Raspberry Pi, που λαμβάνει τιμές από τους ίδιους αισθητήρες, θα πρέπει να δημιουργήσει ακόμα ένα DP τύπου Raspberry_Pi.

7.1.2 JCOP Framework

Στις περιπτώσεις που χρησιμοποιείται το WinCC OA είναι πολύ συνηθισμένη και η χρήση του **JCOP Framework** (Joint COntrols Project Framework). Το JCOP αποτελεί 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 να χρησιμοποιηθούν μαζί όταν χρειαστεί. Το JCOP αποτελεί στην πραγματικότητα μία σειρά από εργαλεία τα οποία έχει στη διάθεσή του ο χρήστης και χρησιμοποιούνται ως βασική δομή για τη δημιουργία projects.

7.2 Σύνδεση Raspberry Pi με το WinCC OA

Η επικοινωνία μεταξύ Raspberry Pi και WinCC πραγματοποιείται μέσω του πρωτοκόλλου OPC UA. Στο προηγούμενο κεφάλαιο 6 έγινε περιγραφή του OPC UA Server, που στήθηκε στο Raspberry Pi. Ο Server αυτός επικοινωνεί τα δεδομένα, που

συλλέγει το Raspberry Pi από τους αισθητήρες BME280. Στη συνέχεια, ανάλογα τον αριθμό των αισθητήρων, δημιουργεί τα αντίστοιχα Objects και καταχωρεί τα δεδομένα σε variables, τα οποία είναι σε θέση να αποστέλλει σε κάθε OPC UA Client που θα συνδεθεί σε αυτόν. Το WinCC παρέχει OPC UA Client, ο οποίος με τη διαδικασία που περιγράφεται στη συνέχεια συνδέεται με τον Server.

Αρχικά, πρέπει ο Server να είναι σε θέση να 'ακούσει' το αίτημα του Client, να είναι δηλαδή ενεργός, εικόνα 6.7, ώστε να μπορεί να ανιχνευθεί. Επομένως, αφού ενεργοποιηθεί ο OPC UA Server στο RASpberry Pi, στη συνέχεια πραγματοποιείται η διαδικασία του configuration του OPC UA Client στο WinCC OA, σε καινούργιο ή σε ήδη υπάρχον project. Στο υποκεφάλαιο, που ακολουθεί, περιγράφεται η διαδικασία δημιουργίας καινούργιου project στο WinCC OA και το configuration του OPC UA Client και τέλος πως συνδέεται το Raspberry Pi με το WinCC OA.

7.2.1 Δημιουργία καινούργιου project στο WinCC OA

Αρχικά ο χρήστης ανοίγει το **WinCC OA Console** και το **WinCC OA Project Administrator** (πρέπει να τα εκτελέσει σαν administrator). Στο δεύτερο επιλέγει **New Project** και στη συνέχεια **Distributed Project**, όπως φαίνεται στην εικόνα 7.2.

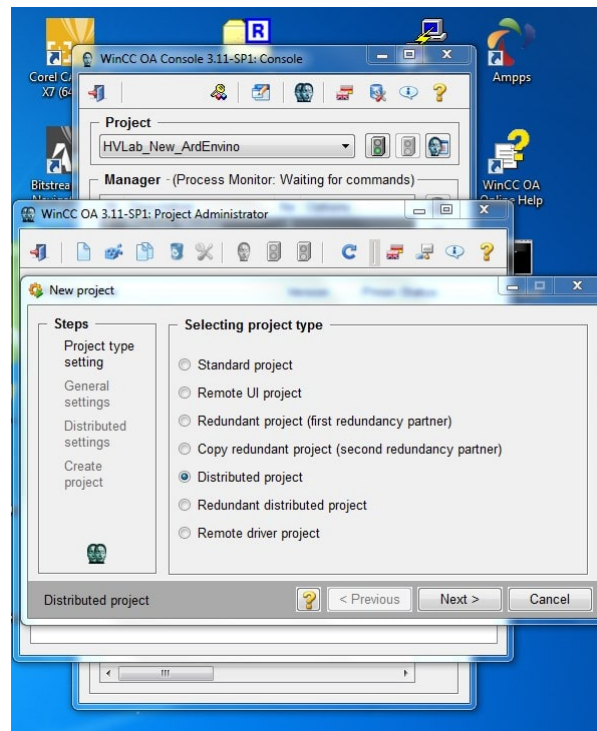


Figure 7.2: Δημιουργία νέου Distributed project στο WinCC OA

Έπειτα, επιλέγει το όνομα του νέου project, το φάκελο στον οποίο θα αποθηκευτεί (εικόνα 7.3) και τέλος το **System Name** του project (εικόνα 7.4).

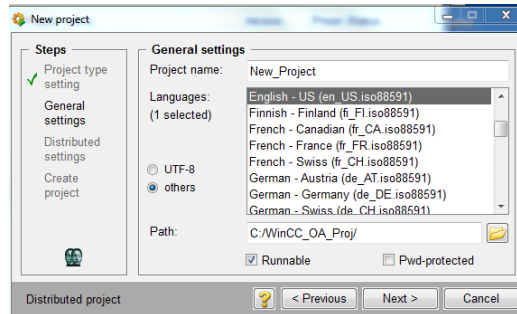


Figure 7.3: Επιλογή ονόματος κατά τη δημιουργία νέου project στο WinCC OA

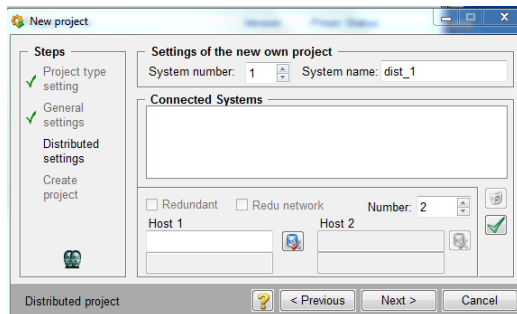


Figure 7.4: Επιλογή System Name για το νέο project στο WinCC OA

Τέλος, πατώντας 'Ok' ο χρήστης, δημιουργεί το νέο project. Επόμενο βήμα είναι η σύνδεση του WinCC OA με τον OPC UA Server του Raspberry Pi, πρέπει να γίνει δηλαδή το configuration του OPC UA Client του WinCC.

7.2.2 Σύνδεση WinCC OA με OPC UA Server

Στη συνέχεια, το project που δημιουργήθηκε, με τη διαδικασία που περιγράφηκε, πρέπει να συνδεθεί με τον OPC UA Server. Ο χρήστης ανοίγει αυτό το project από το WinCC OA Console και πατάει την επιλογή 'Edit config file'. Στο παράθυρο, που ανοίγει, επιλέγει το config file του project και εισάγει στο τέλος τις εξής γραμμές:

```
[opcua_10]  
server="RaspberryPi"
```

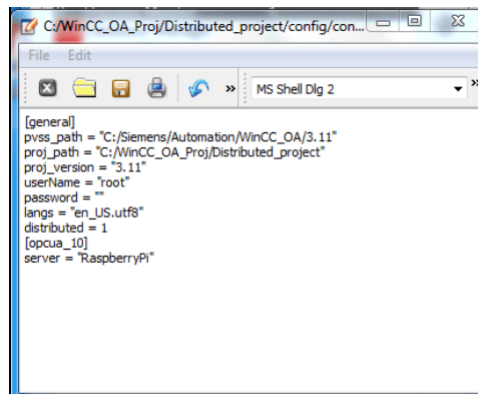


Figure 7.5: Ο χρήστης πρέπει να εισάγει στο configuration file τον OPC UA Server 'RaspberryPi' στο num-10, το οποίο ανοίγει από το παράθυρο Console.

Στη συνέχεια πατάει "Save" και "Exit", ώστε να αποθηκευθεί το configuration. Το επόμενο βήμα είναι να εισαχθεί ο OPC UA manager στο WinCC ΟΑ Console. Ο χρήστης κάνοντας δεξί κλικ, επιλέγει την επιλογή 'Append a new manager' και στο καινούργιο παράθυρο, που θα ανοίξει, θα επιλέξει την επιλογή : WCCOΑορcua ... OPC UA Client , εικόνα 7.6

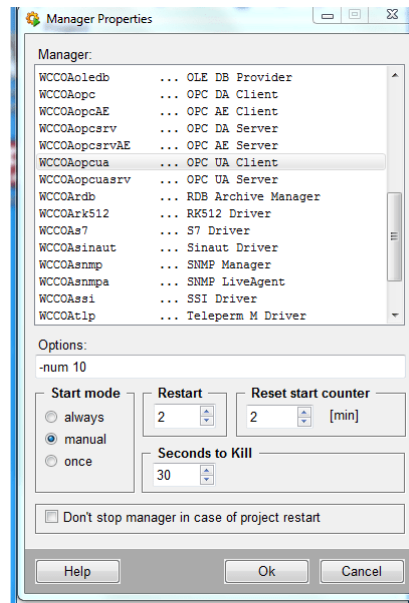


Figure 7.6: Δημιουργία OPC UA Client Manager

Στο πεδίο 'Options' συμπληρώνει '-num 10' και στο πεδίο 'Start Mode' συμπληρώνει 'manual', ώστε ο χρήστης να είναι αυτός, που θα ξεκινάει τον manager και να μην ξεκινάει αυτόματα κάθε φορά που θα γίνεται εκκίνηση του project. Στη συνέχεια, πρέπει να εκκινήσει το project ('Start project') και τον manager, που μόλις δημιουργήθηκε, κάνοντας δεξί κλικ και επιλέγοντας 'Manager Start'.

Για την ολοκλήρωση της διαδικασίας, αφού ανοίξει ο GEDI, ο χρήστης πρέπει να ανοίξει τον PARA, όπου επιλέγει το πεδίο 'Internal Datapoints' και στο DPT '_DriverCommon' κάνοντας δεξί κλικ επιλέγει '**Create datapoint**' και το ονομάζει '_Driver10'. Την ίδια διαδικασία ακολουθεί για τη δημιουργία datapoints στα DPTs: '_OPCUA' και '_Statistics_DriverConfigs'

Αφού έχουν δημιουργηθεί τα απαραίτητα datapoints, ο χρήστης επιλέγει να ανοίξει από το GEDI το 'System Manager'. Στο παράθυρο, που ανοίγει, επιλέγει την επιλογή 'OPC Driver' και στη συνέχεια 'OPC UA Client'.

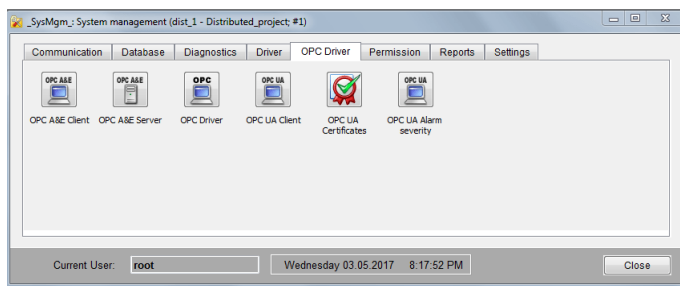


Figure 7.7: Στο GEDI ο χρήστης επιλέγει να ανοίξει το System Manager, στη συνέχεια τον OPC Driver και τέλος επιλέγει OPC UA Client.

Εκεί επιλέγει 'Create' και στο παράθυρο, που ανοίγει, κάνει CONFIGURE τον OPC UA Client, που θα συνδεθεί στον OPC UA Server 'Raspberry Pi', εισάγοντας το URL του Server: 'opc.tcp:ipaddress:4840' και τον ενεργοποιεί επιλέγοντας την επιλογή 'Active'.

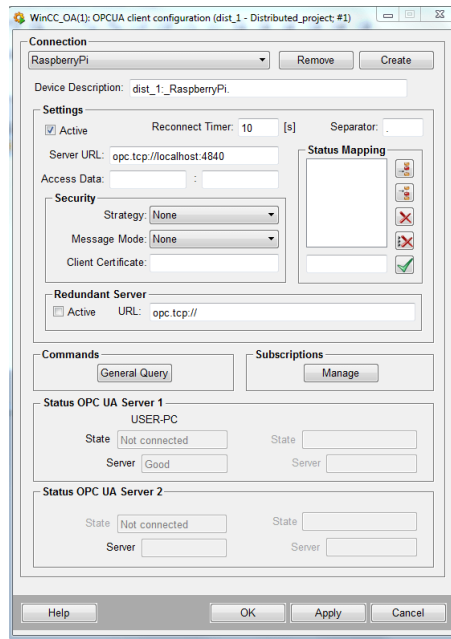


Figure 7.8: Configuration του OPC UA Client στο WinCC OA

Στη συνέχεια, ο χρήστης πρέπει να συνδέσει τα στοιχεία του DataPoint, με τις τιμές των παραμέτρων που στέλνει ο server για κάθε παράμετρο. Επομένως, ανοίγει το PARA και στο Temperature του BME1 για παράδειγμα, κάνοντας δεξί κλικ επιλέγει Insert Config και στη συνέχεια Periphery address, το οποίο αποτελεί ένα DataPoint Element.

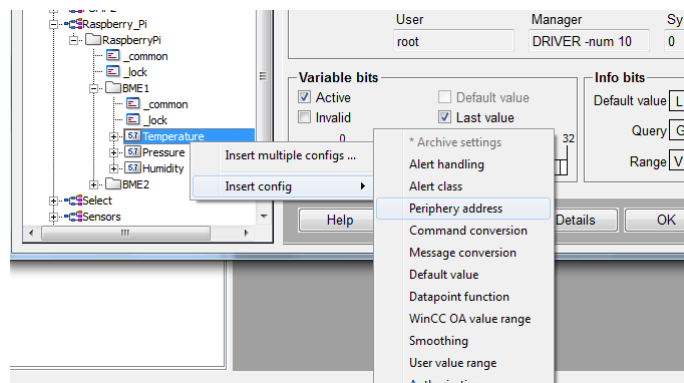


Figure 7.9: Configuration Periphery address (DataPoint Element) των στοιχείων του DataPoint

Έπειτα πρέπει να γίνει το configuration του OPC UA Client. Επιλέγοντας το config_address του Temperature ανοίγει το παράθυρο, που φαίνεται στην εικόνα 7.10, όπου στο πεδίο Driver type επιλέγει OPCUA CLIENT και πατάει Configure.

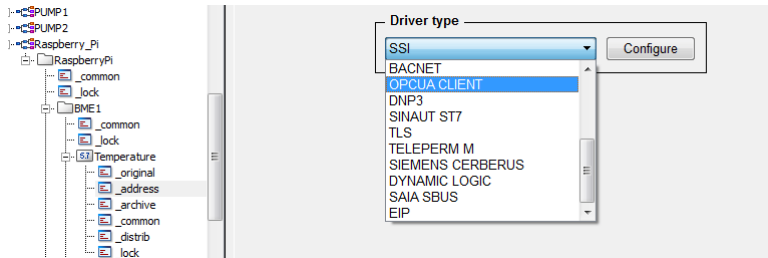


Figure 7.10: Configuration of OPC UA Client

Τότε ανοίγει το παράθυρο Periphery OPCUA.

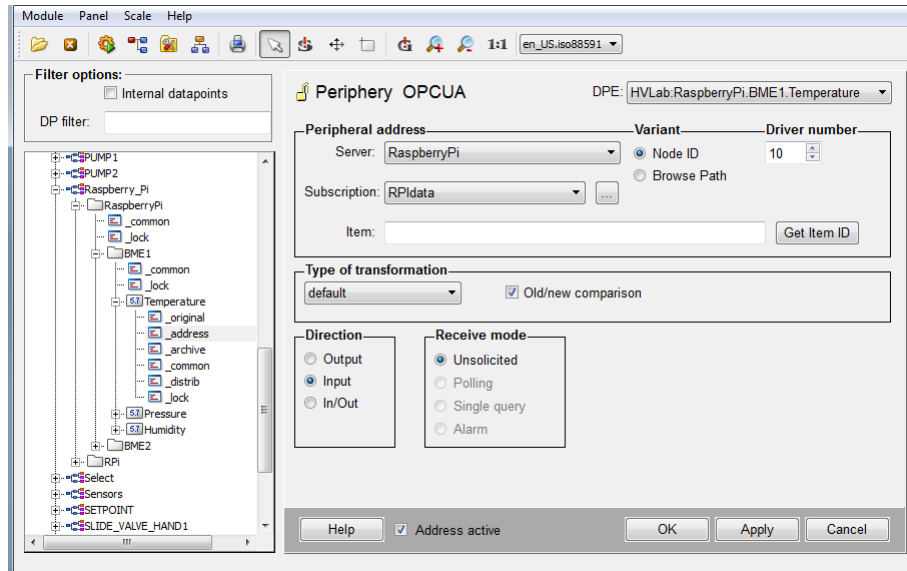


Figure 7.11: Configuration of Periphery address of OPC UA

Στο πεδίο Server ο χρήστης επιλέγει RaspberryPi, στο πεδίο Driver number συμπληρώνει 10 και τέλος στο πεδίο Subscription επιλέγει τις τρεις τελείες στα δεξιά και στο παράθυρο, που ανοίγει, δημιουργεί (create) μια νέα εγγραφή (subscription) για τον OPC UA Server RaspberryPi. Αφού δημιουργηθεί το subscription RPIdata ο χρήστης πατάει Apply και OK.

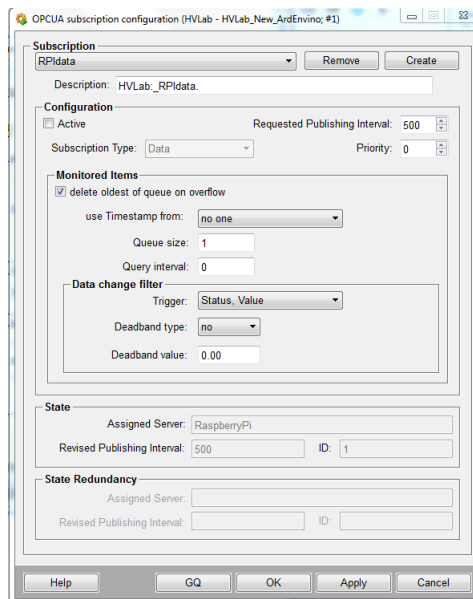


Figure 7.12: Configuration of subscription for OPC UA Client

Τέλος, στο πεδίο Item επιλέγει Get Item ID και στο παράθυρο, που αναδύεται, εμφανίζεται το περιεχόμενο του OPC UA Server RaspberryPi, εικόνα 7.13. Εκεί

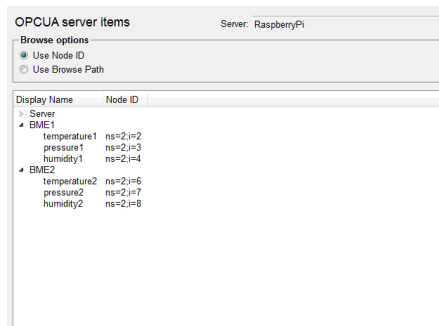


Figure 7.13: Περιεχόμενο OPC UA Server

επιλέγει το Object του server BME1 και τέλος το Variable temperature1. Την ίδια διαδικασία ακολουθεί και για τα υπόλοιπα στοιχεία του DataPoint.

7.3 Εγκατάσταση JCOP Framework Components

Στην ενότητα 6.1.2 έγινε αναφορά στο JCOP Framework και την σημασία της χρήσης του. Στην ενότητα αυτή γίνεται περιγραφή της διαδικασίας εγκατάστασής του στο project. Αρχικά ο χρήστης επιλέγει το αρχείο **fwInstaller.zip** και το εξάγει στο φάκελο του project, που έχει δημιουργήσει. Έπειτα, πρέπει να μεταφέρει το αρχείο **framework.zip** σε οποιοδήποτε σημείο του υπολογιστή (για παράδειγμα στην επιφάνεια εργασίας). Στη συνέχεια, γίνεται εκκίνηση του project στο WinCC OA και πρέπει να ανοιχθεί το fwInstallation panel. Για να πραγματοποιηθεί αυτό, ο χρήστης πρέπει, αφού ανοίξει το project, να ενεργοποιήσει το GEDI (Graphics EDItor) και να ακολουθήσει τη διαδικασία που περιγράφεται στις εικόνες, που ακολουθούν.

Στο πάνω αριστερά μέρος του GEDI ο χρήστης επιλέγει Module και στη συνέχεια Vision και ανοίγει το παράθυρο, που φαίνεται στην εικόνα 7.14, όπου γίνεται επιλογή του φακέλου fwInstallation και ανοίγεται το fwInstallation.pnl.

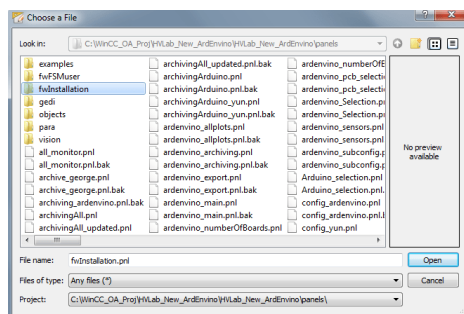


Figure 7.14: Ο χρήστης επιλέγει το φάκελο στον οποίο περιέχεται το αρχείο fwInstallation.pnl

Όταν ανοίξει το **'fwInstallation panel'**, εικόνα 7.15, ζητείται από το χρήστη να επιλέξει το φάκελο, στον οποίο θα αποθηκευτούν τα επιλεγμένα 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 αρχείου **framework.zip**. Ο πίνακας Available components συμπληρώνεται από όλα τα components, τα οποία μπορούν να εγκατασταθούν στο project. εικόνα 7.16. Στη συνέχεια, ο χρήστης επιλέγει τα components, που επιθυμεί. Στο project χρειάστηκαν τα εξής: **fwAccessControl**, **fwCaen**, **fwCore**, **fwTrending**, **fwXML**. Τέλος, αφού επιλεγθούν τα components ο χρήστης κάνει κλικ στο κουμπί "Install" και πατά "OK". Στα δύο επόμενα παράθυρα επιλέγεται το "Activate Driver" και "Restart Project".

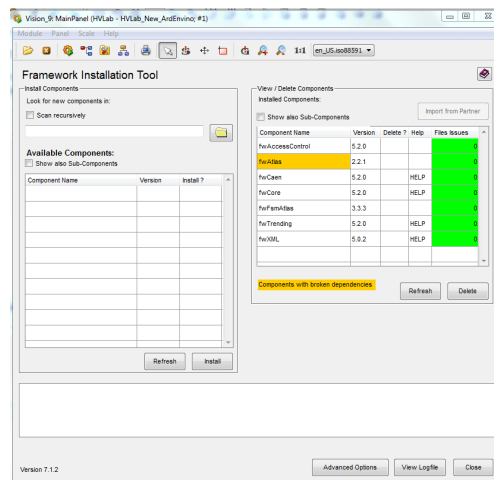


Figure 7.15: Πραγματοποιείται εκκίνηση του panel 'fwInstallation.pnl' και στη συνέχεια ζητείται από το χρήστη να επιλέξει το φάκελο, στον οποίο θα αποθηκευτούν τα επιλεγμένα components. Αφού δημιουργηθεί ένας τέτοιος φάκελος (αν δεν έχει ήδη δημιουργηθεί), πρέπει στη συνέχεια να δηλωθεί ο φάκελος στον οποίο είχαν εξαχθεί τα περιεχόμενα του framework.zip αρχείου.

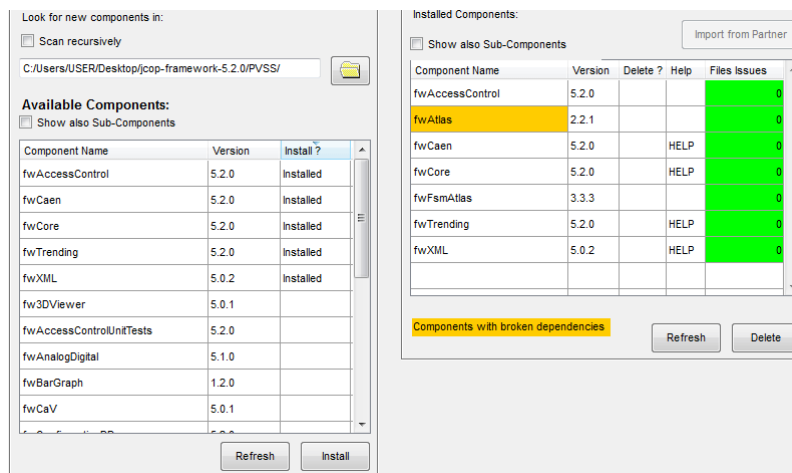


Figure 7.16: Ο πίνακας Available components συμπληρώνεται από όλα τα components τα οποία μπορούν να εγκατασταθούν στο project. Στη συνέχεια, ο χρήστης επιλέγει τα components που επιθυμεί. Στο project χρειάστηκαν τα εξής: fwAccessControl, fwCaen, fwCore, fwTrending, fwXML. Τέλος, αφού επιλεγούν τα components ο χρήστης κάνει κλικ στο κουμπί "Install" και πατά "OK". Στα δύο επόμενα παράθυρα επιλέγεται το "Activate Driver" και "ReSTART Project".

7.4 Raspberry Pi WinCC OA Project

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

Αφού ο χρήστης ολοκληρώσει την εγκατάσταση των απαραίτητων components, μπορεί να ξεκινήσει την προεπισκόπηση του project εκτώντας το panel επιλογής του αριθμού αισθητήρων. Έτσι εμφανίζεται το παράθυρο της εικόνας 7.17, όπου ο χρήστης καλείται να επιλέξει τον αριθμό των αισθητήρων, που έχει χρησιμοποιήσει (έναν ή δυο).

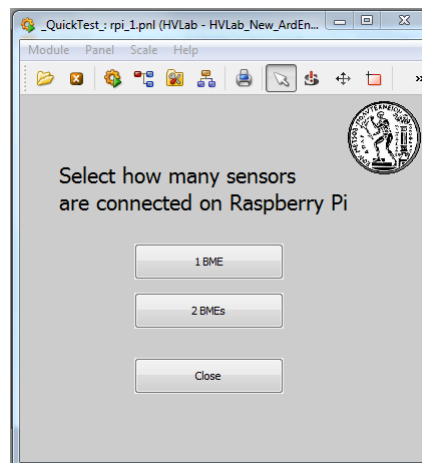


Figure 7.17: Panel επιλογής αριθμού αισθητήρων

Ανάλογα με την επιλογή του θα εμφανιστεί το αντίστοιχο main panel για το Raspberry Pi, όπου θα εμφανίζει τις τιμές των παραμέτρων, που συλλέγονται από έναν (εικόνα 7.18) ή δύο (εικόνα 7.19) αισθητήρες. Στο main panel σε κάθε περίπτωση εμφανίζονται τρία κουμπιά, κάθε ένα από τα οποία εκτελεί μια συγκεκριμένη διεργασία. Τα κουμπιά αυτά και οι αντίστοιχες διεργασίες, που εκτελούνται, περιγράφονται στις ενότητες που ακολουθούν.

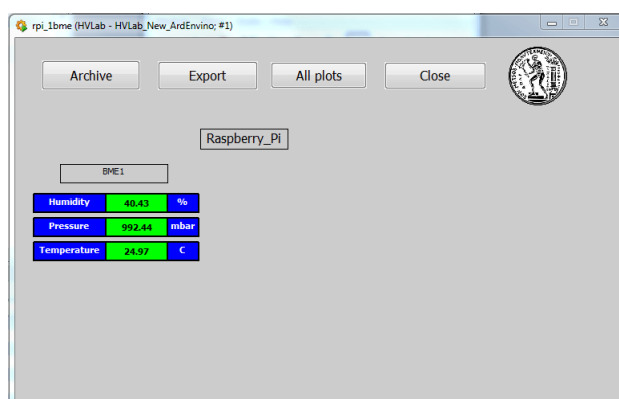


Figure 7.18: Main Panel για έναν BME280



Figure 7.19: Main Panel για δυο BME280

7.4.1 Archive

Στην ενότητα αυτή γίνεται περιγραφή του "Archive Panel", στο οποίο ο χρήστης επιλέγει τις παραμέτρους για τις οποίες θα καθορίσει τις ρυθμίσεις αρχειοθέτησης τους (archiving), την αποθήκευσή τους δηλαδή στη βάση δεδομένων. Στο main panel του Raspberry Pi, ο χρήστης κάνοντας κλικ στην επιλογή "Archive", εμφανίζεται το παράθυρο της εικόνας 7.20.

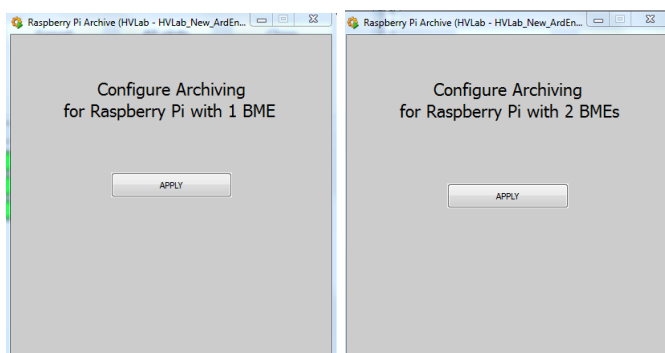


Figure 7.20: Configure Archiving

Εκεί επιλέγει "APPLY" για κάθε περίπτωση και στη συνέχεια ανάλογα με τον αριθμό των αισθητήρων, που έχει επιλέξει, ανοίγει το αντίστοιχο παράθυρο, όπως φαίνεται στις εικόνες 7.21 και 7.22.

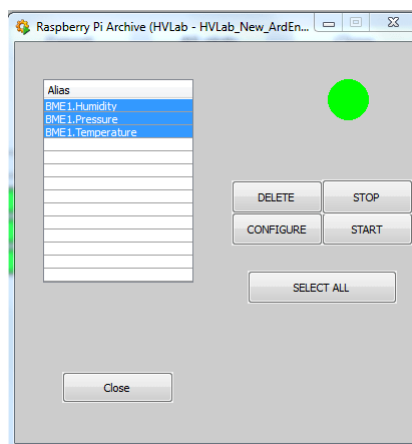


Figure 7.21: Archiving Panel για τις τιμές των μεταβλητών, που συλλέγονται από έναν BME280. Ο χρήστης έχει τη δυνατότητα να επιλέξει ποιες από όλες τις μεταβλητές θέλει να αποθηκευτούν, ενώ του δίνεται και η δυνατότητα να τις επιλέξει όλες με το κουμπί "SELECT ALL", στη συνέχεια κάνει κλικ στο "CONFIGURE" και τέλος στο "START", ώστε να ξεκινήσει το archiving. Επίσης, έχει τη δυνατότητα να σταματήσει τη διαδικασία πατώντας το "STOP" ή να διαγράψει τις ρυθμίσεις, που έχουν γίνει με το "DELETE". Οι τιμές των μεταβλητών αποθηκεύονται στη βάση δεδομένων.

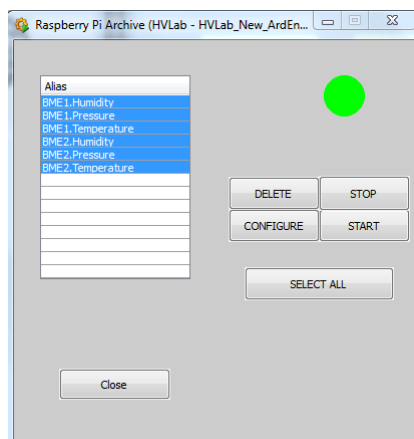


Figure 7.22: Archiving Panel για τις τιμές των μεταβλητών που συλλέγονται από δύο BME280's. Ο χρήστης έχει τις ίδιες επιλογές με την περίπτωση, που συλλέγονται τιμές από έναν BME280, εικόνα 6.21. Οι τιμές των μεταβλητών αποθηκεύονται στη βάση δεδομένων.

Οι ρυθμίσεις για το archiving είναι συγκεκριμένες και προκαθορισμένες. Οι τιμές λαμβάνονται κάθε 3 λεπτά ή αν αλλάξει η τιμή τους κατά 1 μονάδα για τη θερμοκρασία και την υγρασία και 50 mbar για την πίεση. Ο χρήστης μπορεί να επιλέξει ποιές από όλες τις μεταβλητές θέλει να αποθηκευτούν, ενώ του δίνεται και η δυνατότητα να τις επιλέξει όλες κάνοντας κλικ στην επιλογή "SELECT ALL", στη συνέχεια κλικ στο "CONFIGURE" και τέλος στο "START", ώστε να ξεκινήσει το archiving. Επίσης, έχει τη δυνατότητα να σταματήσει τη διαδικασία πατώντας το "STOP" ή να διαγράψει τις ρυθμίσεις, που έχουν γίνει με το "DELETE" (έχοντας πριν επιλέξει τις μεταβλητές για τις οποίες επιθυμεί να διαγράψει τις ρυθμίσεις).

Οι τιμές των επιλεγμένων παραμέτρων, που λαμβάνονται από την διαδικασία αυτή, αποθηκεύονται στην βάση δεδομένων (database) του WinCC OA. Στην επόμενη ενότητα θα αναλυθεί ο τρόπος, με τον οποίο ο χρήστης μπορεί να εξάγει τα αποθηκευμένα δεδομένα (export) και να τα επεξεργαστεί.

7.4.2 Export

Στην ενότητα αυτή γίνεται περιγραφή του "Export Panel", στο οποίο ο χρήστης επιλέγει τις παραμέτρους των οποίων τις τιμές θέλει να εξάγει, ώστε να τις επεξεργαστεί. Απαραίτητη προϋπόθεση είναι, να έχει πραγματοποιηθεί προηγουμένως archiving, ώστε οι τιμές αυτές να έχουν αποθηκευτεί στη βάση δεδομένων. Στο main panel του Raspberry Pi ο χρήστης επιλέγοντας το "Export" εμφανίζεται ένα νέο παράθυρο, όπου επιλέγει "APPLY" και αναδύεται αυτόματα το Export Panel ανάλογα με τον αριθμό των αισθητήρων, που είχε επιλέξει αρχικά, εικόνες 7.24 και 7.25.

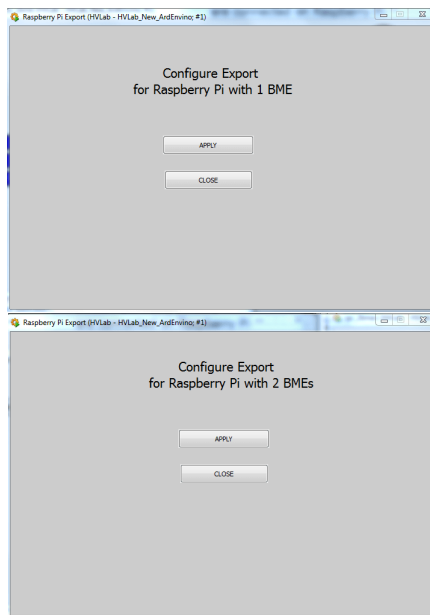


Figure 7.23: Configure Export

Ο χρήστης μπορεί να επιλέξει τις μεταβλητές, των οποίων τις τιμές θέλει να εξαγάγει σε αρχείο τύπου ".dat". Πατώντας το κουμπί "SELECT ALL" μπορεί να τις επιλέξει όλες. Αν τις επιλέξει όλες, οι τιμές αυτών θα αποθηκευτούν ξεχωριστά σε διαφορετικό αρχείο στον ίδιο όμως φάκελο. Στη συνέχεια, επιλέγει τη χρονική στιγμή, από την οποία επιθυμεί να ξεκινήσει το αρχείο εξαγωγής στο πεδίο "START TIME" και αντίστοιχα τη χρονική στιγμή, που επιθυμεί να σταματήσει, στο πεδίο "END TIME". Στο πεδίο "FOLDER NAME" εισάγει το όνομα του φακέλου, που θα περιέχει τα αρχεία τύπου .dat με τις τιμές κάθε μεταβλητής για κάθε χρονική στιγμή στο χρονικό διάστημα, που έχει ορίσει ο χρήστης. Τέλος, πατώντας "EXPORT", δημιουργείται ο φάκελος.

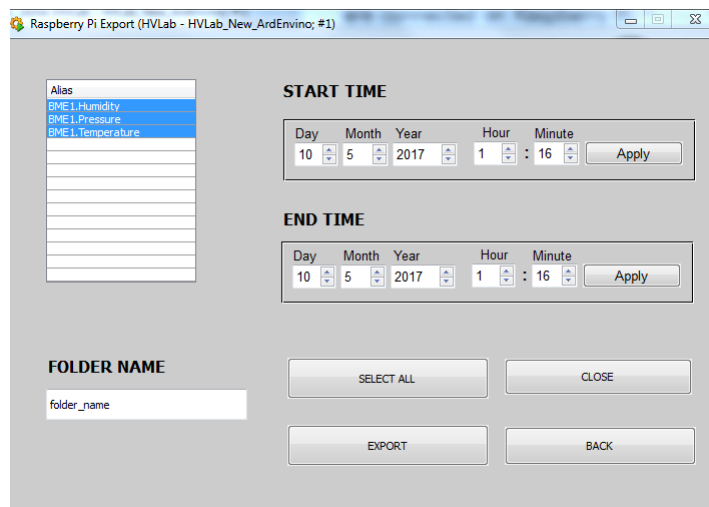


Figure 7.24: Export Panel για την εξαγωγή των τιμών των μεταβλητών σε αρχείο τύπου '.dat', οι οποίες συλλέγονται από έναν αισθητήρα BME280. Ο χρήστης επιλέγει, ποιες μεταβλητές επιθυμεί, έχει όμως πάλι την επιλογή να τις επιλέξει όλες. Οι τιμές θα αποθηκευτούν σε ξεχωριστά .dat αρχεία. Στο πεδίο 'START TIME', επιλέγει τη χρονική στιγμή, από την οποία επιθυμεί να ξεκινήσει το αρχείο εξαγωγής και αντίστοιχα επιλέγει τη χρονική στιγμή, που επιθυμεί να σταματήσει, στο πεδίο 'END TIME'. Στο πεδίο 'FOLDER NAME' εισάγει το όνομα του φακέλου, που θα περιέχει τα αρχεία τύπου .dat. Τέλος, πατώντας 'EXPORT' δημιουργείται ο φάκελος.

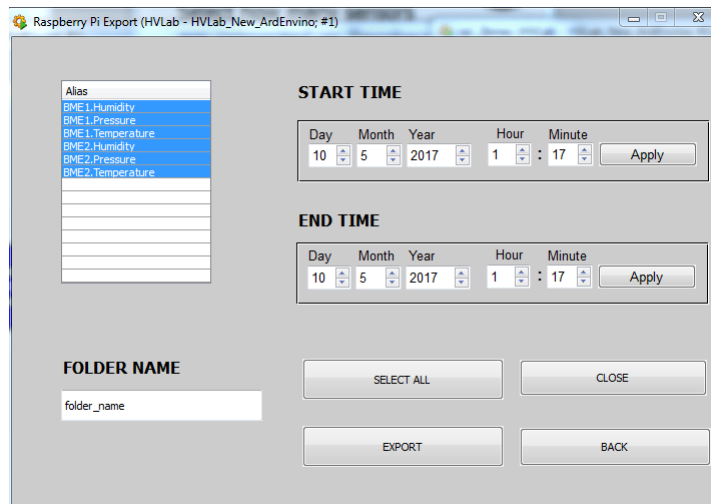


Figure 7.25: Export Panel για την εξαγωγή των τιμών των μεταβλητών σε αρχείο τύπου '.dat', οι οποίες συλλέγονται από δύο αισθητήρες BME280. Ο χρήστης έχει τις ίδιες επιλογές για την εξαγωγή των τιμών που έχουν αποθηκευτεί στη βάση δεδομένων, με την περίπτωση που συλλέγονται τιμές από έναν BME280, εικόνα 6.24

7.4.3 All plots - Real Time Data

Το τελευταίο κουμπί, που υπάρχει στο main panel του Raspberry Pi είναι το "All plots". Όταν το πατήσει ο χρήστης, ανοίγει ένα νέο παράθυρο, στο οποίο επιλέγει για ποια μεταβλητή επιθυμεί, να εμφανίσει τη γραφική παράσταση των τιμών της σε πραγματικό χρόνο (real-time plots). Επίσης, υπάρχει η επιλογή "All plots", όπου εμφανίζει και τις τρεις γραφικές παραστάσεις για τις τιμές της θερμοκρασίας, της υγρασίας και της πίεσης σε πραγματικό χρόνο, 7.27.

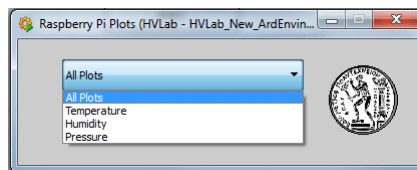


Figure 7.26: Επιλογή της μεταβλητής που θα εμφανιστεί το γράφημα των τιμών της real-time

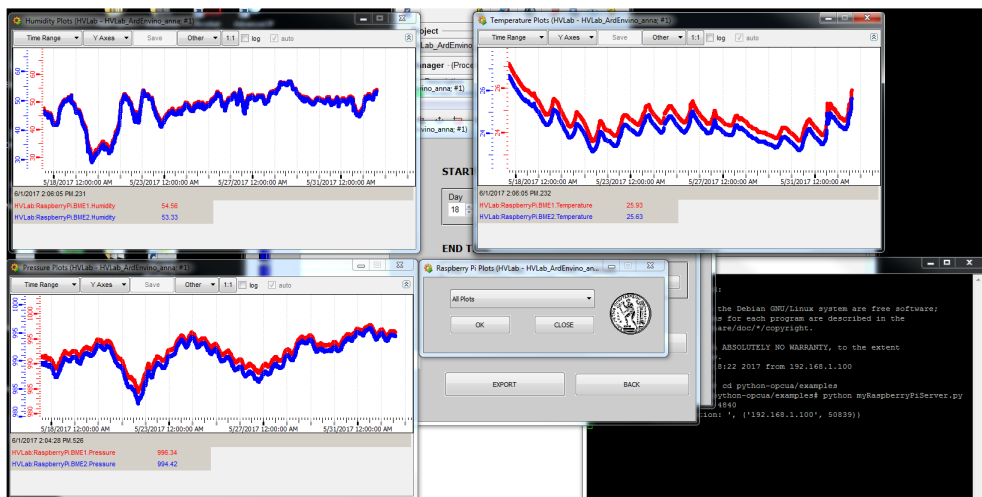


Figure 7.27: Διάγραμματα λήψης τιμών των παραμέτρων θερμοκρασίας, πίεσης και υγρασίας σε πραγματικό χρόνο.

Στις εικόνες 7.28, 7.29 και 7.30 φαίνονται τα live plots για κάθε μια από τις παραμέτρους, της θερμοκρασίας, της πίεσης και της υγρασίας

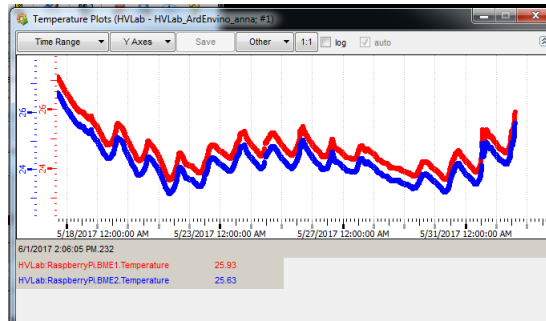


Figure 7.28: Διάγραμμα λήψης τιμών της παραμέτρου θερμοκρασίας σε πραγματικό χρόνο.

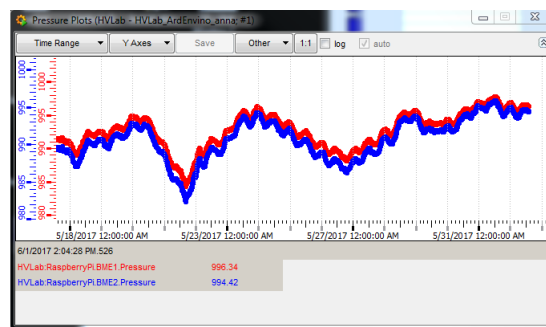


Figure 7.29: Διάγραμμα λήψης τιμών της παραμέτρου πίεσης σε πραγματικό χρόνο.

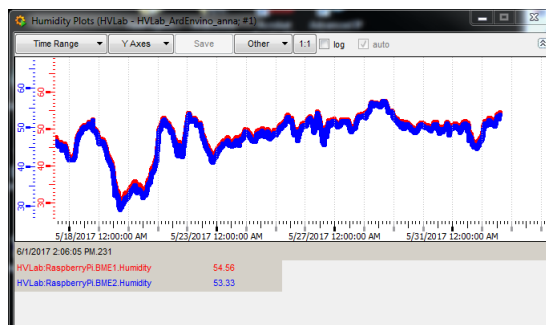


Figure 7.30: Διάγραμμα λήψης τιμών της παραμέτρου υγρασίας σε πραγματικό χρόνο.

Κεφάλαιο 8

Συμπεράσματα

Για τους σκοπούς της παρούσας πτυχιακής εργασίας δημιουργήθηκε ένα σύστημα μέτρησης, καταγραφής, αποθήκευσης και επεξεργασίας από απόσταση των περιβαλλοντικών παραμέτρων θερμοκρασίας, πίεσης και υγρασίας. Το σύστημα αυτό βασίζεται στο μικροϋπολογιστή Raspberry Pi και το πρόγραμμα SCADA WinCC OA. Το πρώτο είναι υπεύθυνο για τη λήψη και την αποστολή των δεδομένων στο WinCC OA βάσει του πρωτοκόλλου επικοινωνίας OPC UA. Το WinCC από την άλλη είναι υπεύθυνο για την καταγραφή και την αποθήκευση των δεδομένων με σκοπό την μετέπειτα επεξεργασία τους. Η επικοινωνία μεταξύ Raspberry Pi και του προγράμματος SCADA γεφυρώνεται μέσω ενός OPC UA Server, ο οποίος στήθηκε στα πλαίσια της πτυχιακής εργασίας για αυτόν τον σκοπό. Για τη λήψη των περιβαλλοντικών παραμέτρων χρησιμοποιήθηκαν δυο ψηφιακοί αισθητήρες BME280, οι οποίοι λαμβάνουν τιμές θερμοκρασίας, πίεσης και υγρασίας, και συνδέονται με το Raspberry Pi βάσει του πρωτοκόλλου επικοινωνίας I^2C .

Τα διαγράμματα που φαίνονται στις εικόνες 8.1, 8.2, 8.3, δημιουργήθηκαν από τα αρχεία που εξήχθησαν από το WinCC OA μέσω του panel 'Export'. Στα διαγράμματα αυτά παρουσιάζονται οι τιμές των περιβαλλοντικών παραμέτρων, από μετρήσεις που έγιναν από δυο αισθητήρες BME280 στο εργαστήριο, κατά τη διάρκεια δύο εβδομάδων 18/5-6/1.

Στα διαγράμματα φαίνεται πως οι τιμές των δυο αισθητήρων εμφανίζουν μια μικρή απόκλιση. Εφόσον όμως, δεν μας ενδιαφέρει το absolute value των παραμέτρων αλλά το gradient value, για να δούμε κατά πόσο όντως έχουν απόκλιση οι τιμές που των 2 BME's δημιουργήσαμε τα διαγράμματα των differential values για κάθε παράμετρο, τα οποία φαίνονται στις εικόνες που ακολουθούν 8.4, 8.7, 8.8.

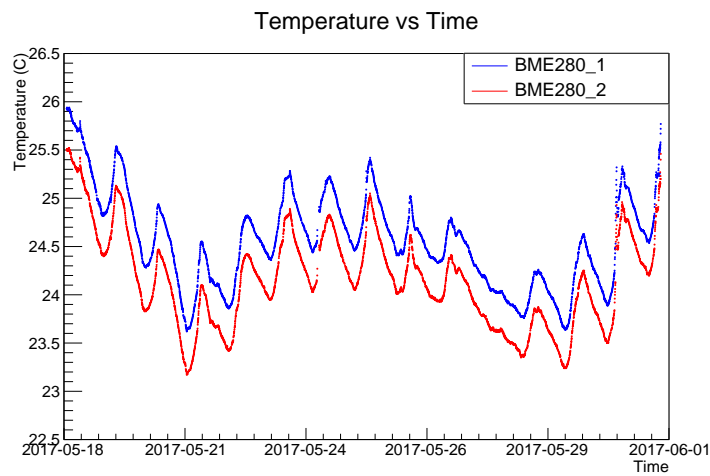


Figure 8.1: Διάγραμμα θερμοκρασίας συναρτήσει του χρόνου κατά τη διάρκεια δύο εβδομάδων 18/5-6/1.

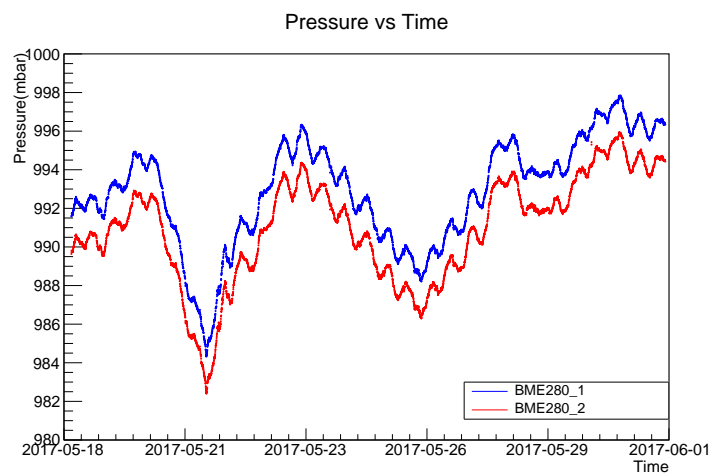


Figure 8.2: Διάγραμμα πίεσης συναρτήσει του χρόνου κατά τη διάρκεια δύο εβδομάδων 18/5-6/1.

Τέλος παρατίθενται εικόνες της διάταξης του Raspberry Pi συνδεδεμένο με τους δυο αισθητήρες BME280 εικόνα 7.27 και με οθόνη αφής 7", στην οποία γίνεται φαίνεται η ιστοσελίδα παρακολούθησης των τιμών των δύο αισθητήρων εικόνα ??

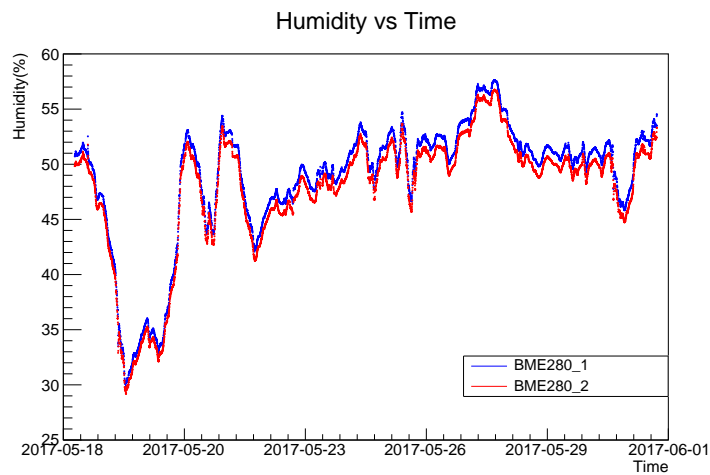


Figure 8.3: Διάγραμμα υγρασίας συναρτήσει του χρόνου κατά τη διάρκεια δύο εβδομάδων 18/5-6/1.

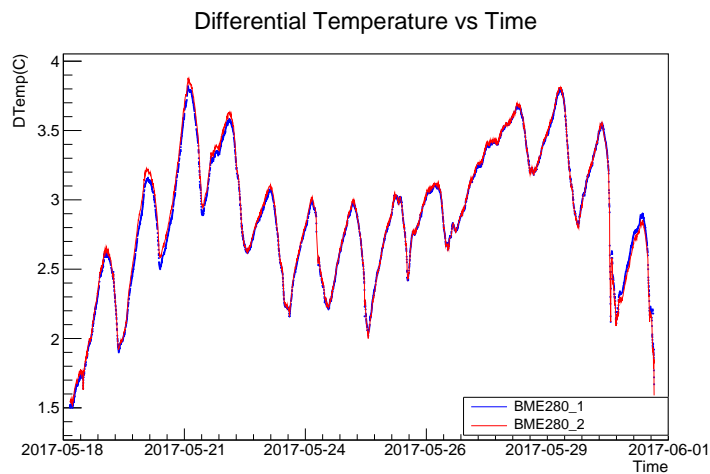


Figure 8.4: Διάγραμμα differential θερμοκρασίας συναρτήσει του χρόνου κατά τη διάρκεια δύο εβδομάδων 18/5-6/1.

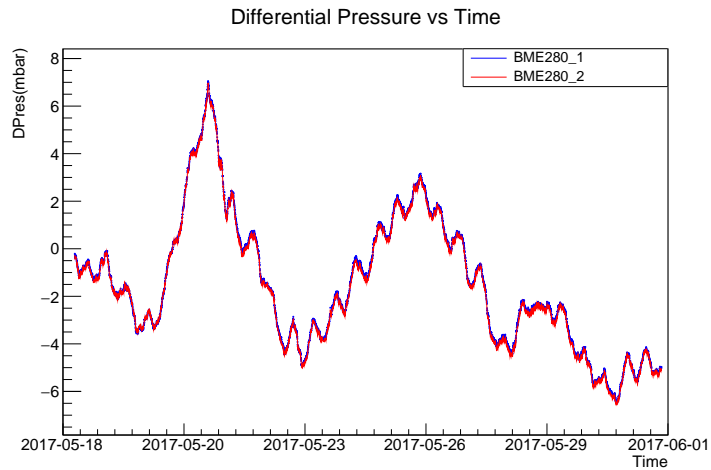


Figure 8.5: Διάγραμμα differential πίεσης συναρτήσει του χρόνου κατά τη διάρκεια δύο εβδομάδων 18/5-6/1.

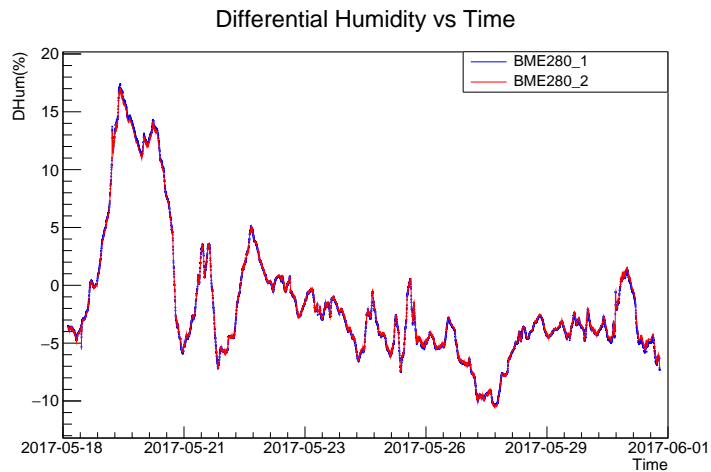


Figure 8.6: Διάγραμμα differential υγρασίας συναρτήσει του χρόνου κατά τη διάρκεια δύο εβδομάδων 18/5-6/1.

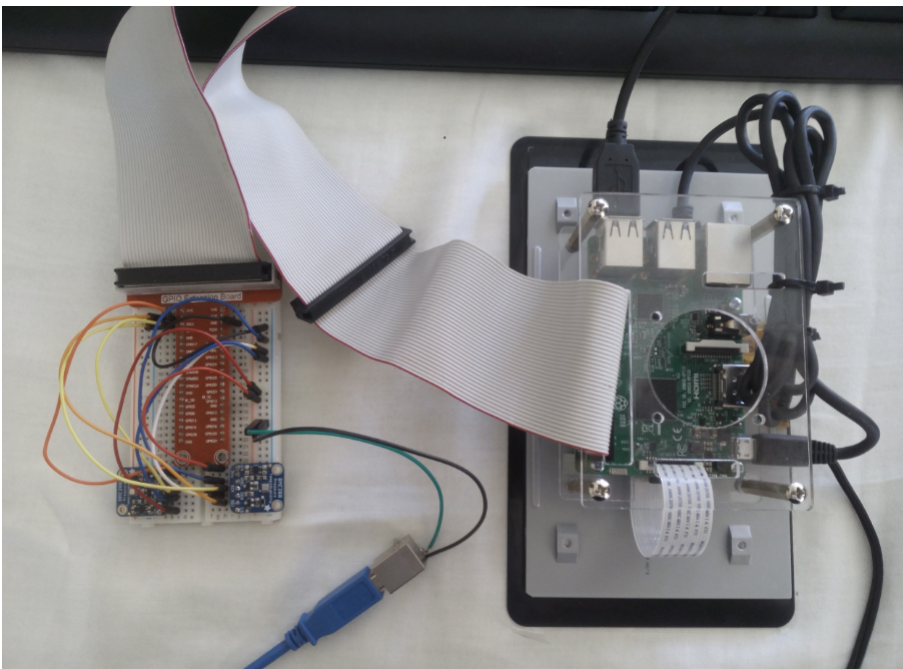


Figure 8.7: Διάταξη που περιλαμβάνει το Raspberry Pi με τους δύο αισθητήρες BME280.

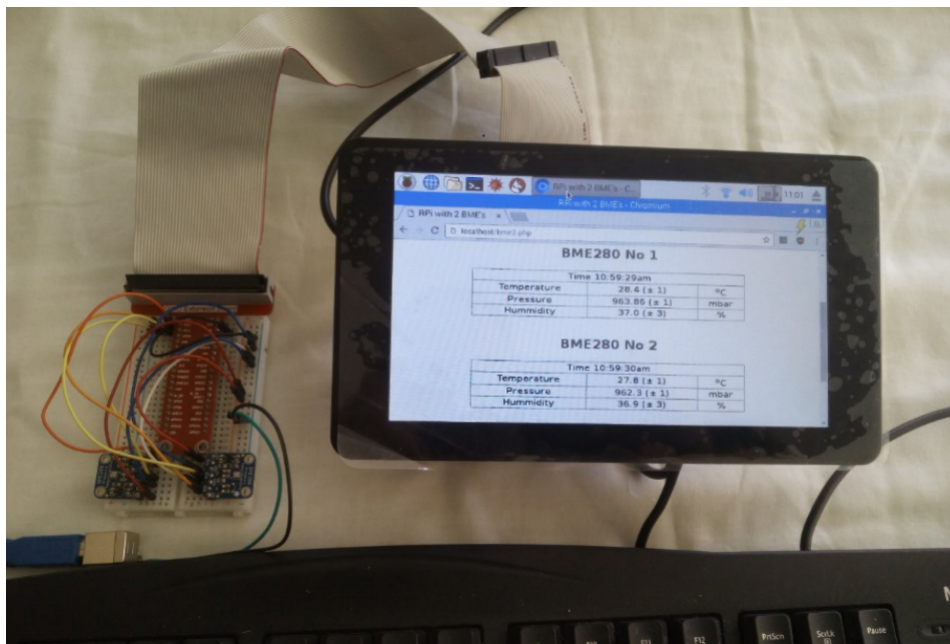


Figure 8.8: Διάταξη που περιλαμβάνει το Raspberry Pi συνδεδεμένο με τους Raspberry Pi καθώς και οθόνη αφής 7", στην οποία φαίνεται η ιστοσελίδα παρακολούθησης των τιμών των παραμέτρων των δύο αισθητήρων .

Βιβλιογραφία

- [1] <https://www.raspberrypi.org/>
- [2] <http://rasbian.org/>
- [3] <http://wikis.web.cern.ch/wikis/display/EN/WinCC-OA+Service>.
- [4] I2C — Wikipedia, The Free Encyclopedia.2015. URL: <https://en.wikipedia.org/wiki/I%C3%82%C2%B2C>.
- [5] Open Platform Communications — Wikipedia, The Free Encyclopedia.2015. URL: https://en.wikipedia.org/wiki/Open_Platform_Communications.
- [6] <https://opcfoundation.org/>.
- [7] OPC UA — Wikipedia, The Free Encyclopedia URL: https://en.wikipedia.org/wiki/OPC_Unified_Architecture
- [8] Apache HTTP — Wikipedia, The Free Encyclopedia URL: https://en.wikipedia.org/wiki/Apache_HTTP_Server
- [9] PHP — Wikipedia, The Free Encyclopedia URL: <https://en.wikipedia.org/wiki/Php>
- [10] Adafruit- BME 280 Sensor: URL:<https://learn.adafruit.com/adafruitbme280-humidity-barometric-pressure-temperature-sensor-breakout/overview>
- [11] T.Waber et al, Fabrication and characterization of a piezoresistive humidity sensor with a stress-free package, 2014
- [12] Delvina Japhet & Kelvin Ndai, Comparisons of thermal responses of different temperature sensors, 2016
- [13] Gaia Maria Berruti, Radiation tolerant fiber optic humidity sensors for High Energy Physics Applications, CERN Thesis, 2015
- [14] Bosch Sensortec, BME280 Combined humidity and pressure sensor, Final Datasheet
- [15] Patent US 20140116122 A1, Combined pressure and humidity sensor

- [16] Patent US 9540229 B2, Packaged sensor assembly
- [17] Patent US 9476779 B2, Sensor with an embedded thermistor for precise local temperature measurement
- [18] Patent US 9022644 B1, Micromachined thermistor and temperature measurement circuitry, and method of manufacturing and operating same
- [19] C. Adlof et al, Environmental study of a Micromegas detector, 2009
- [20] Konstantinos S. Karakostas, MicroMegas Detector for Calorimetry in ILL. Προπτυχιακή διπλωματική εργασία. Εθνικό Μετσόβιο Πολυτεχνείο, 2008
- [21] Παπακριβόπουλος Ιωάννης, Ανάπτυξη συστήματος μέτρησης θερμοκρασίας και υγρασίας βασισμένο σε Arduino και WinCC_OA. Προπτυχιακή διπλωματική εργασία. Εθνικό Μετσόβιο Πολυτεχνείο, 2015
- [22] Μπάκας Γιώργος, Ανάπτυξη συστήματος μέτρησης περιβαλλοντικών συνθηκών βασισμένο σε Arduino YUN και WinCC_OA. Προπτυχιακή διπλωματική εργασία. Εθνικό Μετσόβιο Πολυτεχνείο, 2016
- [23] Κωνσταντίνος Καραϊσκος, Σχεδίαση και κατασκευή ψηφιακού καταγραφέα στις πλατφόρμες Raspberry Pi και Arduino για σύνδεση σε εφαρμογές μικρών ανεμογεννητριών, Προπτυχιακή διπλωματική εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, 2016
- [24] Χρυσόστομος Σ. Χρίστου, Πλατφόρμα για Απομακρυσμένο Έλεγχο Οικίας μέσω Εφαρμογής Android, Προπτυχιακή διπλωματική εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, 2016