



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Πλατφόρμα ανάπτυξης εφαρμογών ευφυούς ελέγχου οικίας
(domotics)**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΧΡΗΣΤΟΣ Ε. ΕΠΙΤΡΟΠΑΚΗΣ

Επιβλέπων : Ευστάθιος Δ. Συκάς

Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2014



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Πλατφόρμα ανάπτυξης εφαρμογών ευφυούς ελέγχου οικίας
(domotics)**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΧΡΗΣΤΟΣ Ε. ΕΠΙΤΡΟΠΑΚΗΣ

Επιβλέπων : Ευστάθιος Δ. Συκάς

Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την

.....
Ευστάθιος Συκάς
Καθηγητής Ε.Μ.Π.

.....
Μιχαήλ Θεολόγου
Καθηγητής Ε.Μ.Π.

.....
Γεώργιος Στασινόπουλος
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2014

.....
ΧΡΗΣΤΟΣ Ε. ΕΠΙΤΡΟΠΑΚΗΣ

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

Copyright © ΧΡΗΣΤΟΣ ΕΠΙΤΡΟΠΑΚΗΣ, 2014

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη πλατφόρμας που βασίζεται στο Raspberry Pi για τον έλεγχο και την αυτοματοποίηση των ηλεκτρονικών συστημάτων μίας οικίας. Ακόμη, η καταγραφή στοιχείων για μεγάλο χρονικό διάστημα για την εξαγωγή στατιστικών συμπερασμάτων που θα συμβάλουν στον αποτελεσματικότερο έλεγχο καθώς και στην εξοικονόμηση ενέργειας.

Πιο συγκεκριμένα, παρουσιάζεται και αναπτύσσεται όλο το λογισμικό για την καταγραφή θερμοκρασιών από αισθητήρες 1-wire καθώς και για τον έλεγχο ηλεκτρικών συσκευών από τα GPIO του Raspberry Pi. Επίσης, γίνεται αναλυτική παρουσίαση για το πως ο αναγνώστης μπορεί να στήσει εξ αρχής ένα καινούριο Raspberry Pi και έναν Web server(που στη συγκεκριμένη Εργασία έγινε σε Virtual Machine)

Για την Εργασία αυτή χρησιμοποιήθηκε η τεχνολογία του MQTT για την μεταφορά των μηνυμάτων από τον μικροελεγκτή στον server, το RRDTool για την δημιουργία βάσεων δεδομένων που χαρακτηρίζονται από την ευκολία στην δημιουργία και ανάγνωσή τους και οι βιβλιοθήκες του Highcharts για την εμφάνιση των αποτελεσμάτων από τον server.

Οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν είναι η php για την ανάπτυξη των ιστοσελίδων της διαχείρισης του Raspberry Pi καθώς και του Virtual Machine και η python για την υλοποίηση των προγραμμάτων που τρέχουν στα δύο μηχανήματα.

Λέξεις Κλειδιά

domotics, Έξυπνο Σπίτι, Raspberry Pi, Web Server, 1-wire, MQTT, RRDTool, Highcharts, Temperature, GPIO.

Abstract

The purpose of this thesis is the development of a platform based on the Raspberry Pi to control and automate the electronic systems of a house. Furthermore, recording data for a long time to draw statistical conclusions will contribute to more effective control and energy saving.

More particularly, all software for recording temperatures of 1-wire sensors and controlling electrical devices with GPIO of Raspberry Pi is presented and developed. Also, there is a detailed presentation of how the reader can set up a brand new Raspberry Pi and a Web server (which in this thesis was done in Virtual Machine).

In this thesis we used technology of the MQTT for transferring messages from the microcontroller to the server, the RRDTool to create databases that are characterized by the ease of creating and reading and Highcharts libraries for displaying the results on the server.

Php was the programming language for the development of the website for the remote administration of Raspberry Pi and for the websites of the Virtual Machine and python for the development of programs that run on both machines.

Keywords

domotics, Smart Home, Raspberry Pi, Web Server, 1-wire, MQTT, RRDTool, Highcharts, Temperature, GPIO.

Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμότερες ευχαριστίες μου στον καθηγητή και επιβλέποντα της συγκεκριμένης εργασίας κ. Ευστάθιο Συκά για την άριστη συνεργασία και τις πολύτιμες πληροφορίες που μου προσέφερε καθ' όλη τη διάρκεια της. Επίσης, θα ήθελα να ευχαριστήσω τον κ. Αλέπη Ευθύμιο, λέκτορα του Πανεπιστημίου Πειραιώς για την καθοδήγηση σε όλη την διάρκεια των σπουδών μου. Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου και την Στέλλα για την συνεχή στήριξη που μου προσφέρουν όλα αυτά τα χρόνια.

Περιεχόμενα

1.1 Πρόλογος - Εισαγωγή	11
1.2 Σκοπός της Εργασίας	12
2.1 Home automation	13
Internet of Things	13
Home automation	13
Πλεονεκτήματα και μειονεκτήματα	15
Ιστορική αναδρομή	16
Η έννοια "Domotics"	17
Τα απαραίτητα στοιχεία για τον οικιακό αυτοματισμό	18
Εφαρμογές	18
Τα πρωτόκολλα	21
Κόστος	22
3. Το υλικό	
I. Raspberry Pi	23
II. SWE0a Temperature Sensor	27
1-wire	28
III. RPI3 I2C to 1-Wire Host Adapter	30
IV. SWE2b Sensor Connection Module	31
V. Εικονικό μηχάνημα που φιλοξενεί τον server	31
4. Το λογισμικό	
I. Λειτουργικό σύστημα	32
II. Λογισμικό	33
OWFS	33
MQTT	34
RRDTool	36
HighCharts	37
5. Στήσιμο μηχανημάτων	
I. Raspberry Pi	38
II. Virtual Machine	42
6. Η web εφαρμογή	
I. Raspberry Pi	43
II. Virtual Machine Web Server	50
Παράρτημα	
I. Raspberry Pi	57
II. Virtual Machine	76
ΒΙΒΛΙΟΓΡΑΦΙΑ	105

Εικόνες

Εικόνα 2.1. Έξυπνο σπίτι	14
Εικόνα 2.2. Το σκάφος με απομακρυσμένο έλεγχο του Nikola Tesla	16
Εικόνα 3.1. Σύγκριση του Raspberry Pi Model A, Model B και Model B+	23
Εικόνα 3.2. Raspberry Pi Model B	24
Εικόνα 3.3. GPIO Model B	24
Εικόνα 3.4. Raspberry Pi με οθόνη αφής	25
Εικόνα 3.5. Camera Module για το Raspberry Pi	26
Εικόνα 3.6. Αισθητήρας θερμοκρασίας SWE0a	27
Εικόνα 3.7. Αισθητήρας θερμοκρασίας DS18B20	27
Εικόνα 3.8. Java Ring με ενσωματωμένο iButton	28
Εικόνα 3.9. Διαδικασίες Reset, αποστολής λογικού "1" και λογικού "0"	29
Εικόνα 3.10. RPI3 I2C to 1-Wire Host Adapter	30
Εικόνα 3.11. DS2482-800 8 Channel I2C to 1-Wire Master IC	30
Εικόνα 3.12. SWE2b Sensor Connection Module	31
Εικόνα 4.1. Παράδειγμα σύνδεσης συσκευών Slave και Master	33
Εικόνα 4.2. Παράδειγμα ανταλλαγής μηνυμάτων μέσω broker	35
Εικόνα 5.1. Raspberry Pi configuration	39
Εικόνα 5.2. Αποτέλεσμα εντολής <code>sudo i2cdetect -y 1</code>	40
Εικόνα 5.3. Apache web server	40
Εικόνα 5.4. Έκδοση Python	41
Εικόνα 6.1. Η αρχική σελίδα Index	43
Εικόνα 6.2. Αλλαγή username και password	43
Εικόνα 6.3. Σελίδα top.html που αποτελεί το top frame	43
Εικόνα 6.4. Σελίδα menu.html που αποτελεί το αριστερό frame	44
Εικόνα 6.5. Διαχείριση IP του Raspberry Pi	45
Εικόνα 6.6. Επιλογή χρονικών διαστημάτων	46
Εικόνα 6.5. Επιλογή αισθητήρων	46
Εικόνα 6.6. Επιλεγμένος αισθητήρας	47
Εικόνα 6.7. Επιλογή GPIO	47
Εικόνα 6.8. Προεπιλεγμένα GPIO	48
Εικόνα 6.9. Εμφάνιση τελευταίων μετρήσεων	49

Σχήματα

Raspberry pi :

index.php	57
home.htm	58
top.html	59
menu.html	59
configure.php	60
cron.php	63
sensors.php	64
gpio.php	65
results.php	69
changelogin.php	70
reboot.php	71
shutdown.php	72
sensors.py	73
revertvalue.py	74

Virtual Machine:

index.php	76
hour.php	76
3hour.php	80
day.php	85
week.php	89
month.php	94
year.php	99
job.py	103

1.1 Πρόλογος - Εισαγωγή

Η ραγδαία ανάπτυξη της τεχνολογίας, η μείωση του κόστους αγοράς εξελιγμένων ηλεκτρονικών συσκευών καθώς και το μικρό μέγεθός τους οδηγεί όλο και περισσότερο στη χρησιμοποίησή τους για την διευκόλυνσή της καθημερινότητας του σύγχρονου ανθρώπου. Συγκεκριμένα, η χρήση τέτοιων κατάλληλων μικροσυσκευών σε ένα σπίτι συνδεδεμένων σε ένα τοπικό δίκτυο δημιουργεί ένα "Έξυπνο σπίτι".

Το "Έξυπνο σπίτι" χρησιμοποιεί μικροελεγχτές που αυτοματοποιούν την επίβλεψη των συνθηκών της οικίας αλλά και την αλλαγή της κατάστασης άλλων συσκευών (Home automation). Συγκεκριμένα, οι μικροελεγχτές μπορούν να μετρούν παραμέτρους που σχετίζονται με το χώρο της οικίας όπως για παράδειγμα την θερμοκρασία ή την υγρασία ενός δωματίου ή εξωτερικού χώρου καθώς επίσης να ελέγχουν και να μεταβάλλουν την κατάσταση άλλων συσκευών, δηλαδή μπορούν αυτόματα να ενεργοποιήσουν ή να απενεργοποιήσουν μία ηλεκτρονική συσκευή. Έχοντας τον κατάλληλο εξοπλισμό, ένα σπίτι δίνει την δυνατότητα στον χρήστη να γνωρίζει ανά πάσα στιγμή την κατάσταση που επικρατεί στο χώρο καθώς και να ελέγχει οποιαδήποτε συσκευή είναι συνδεδεμένη στο δίκτυο.

Η δυνατότητα αυτή του χρήστη να ελέγχει διαρκώς την οικία του δεν αποσκοπεί μόνο στη διευκόλυνση του αλλά και στην όσο το δυνατόν χαμηλότερη κατανάλωση ενέργειας. Αυτό μπορεί να επιτευχθεί είτε εξολοκλήρου κατά την κρίση του χρήστη, δηλαδή να μεταβάλλει χειροκίνητα την κατάσταση των συσκευών, είτε ορίζοντας συγκεκριμένες τιμές-κατώφλια που μόλις ξεπεραστούν να μεταβάλλεται αυτόματα η λειτουργία τους.

Η δημιουργία ενός τέτοιου σπιτιού μπορεί να γίνει είτε με έτοιμα ενσωματωμένα συστήματα που όμως θα πρέπει να έχει προβλεφτεί κατά την κατασκευή του κτηρίου είτε με τη χρήση μιας πλατφόρμας που επιτρέπει τη σύνδεση άλλων συσκευών και εξαρτημάτων σε αυτή. Μια τέτοια πλατφόρμα είναι το Raspberry pi που χρησιμοποιήθηκε σε αυτή την εργασία.

1.2 Σκοπός της Εργασίας

Σκοπός αυτής της εργασίας είναι η ανάπτυξη μία πλατφόρμας εφαρμογών που θα εξασφαλίζει τον ευφυή έλεγχο ενός σπιτιού. Με τη χρήση αυτή της πλατφόρμας ο χρήστης θα μπορεί εύκολα και από παντού να ελέγχει και να διαχειρίζεται την κατάσταση που επικρατεί στην οικία. Θα μπορούσαμε να διακρίνουμε την εργασία μας σε δύο σκέλη:

- Το πρώτο σκέλος, αφορά τη γενικότερη διαχείριση της πλατφόρμας μέσα από μια ιστοσελίδα. Πιο αναλυτικά, ο χρήστης θα μπορεί να μεταβάλλει τα χαρακτηριστικά της σύνδεσης της πλατφόρμας στο τοπικό δίκτυο (IP address, Netmask, Default Gateway) καθώς και να επιλέξει τα στοιχεία των οποίων οι μετρήσεις τον ενδιαφέρουν (αισθητήρες και Είσοδοι-Έξοδοι I/O).
- Το δεύτερο σκέλος, περιλαμβάνει τη μέτρηση και την καταγραφή των θερμοκρασιακών δεδομένων και της κατάστασης λειτουργίας των συσκευών βάσει των αισθητήρων και των θυρών (I/O) που έχουν επιλεγθεί. Τα δεδομένα θα αποθηκεύονται σε μία βάση δεδομένων και θα αναρτούνται σε μορφή διαγραμμάτων σε μια ιστοσελίδα.

Τέλος, η συγκεκριμένη πλατφόρμα θα μπορούσε να επεκταθεί και σε πιο σύνθετες εφαρμογές απλά προσθέτοντας άλλα εξαρτήματα είτε μεταβάλλοντας τα ήδη υπάρχοντα (αντικατάσταση ενσύρματου αισθητήρα με ασύρματο, πρόσθεση αισθητήρα για μέτρηση υγρασίας και άλλων χαρακτηριστικών).

2. Home automation

Internet of Things

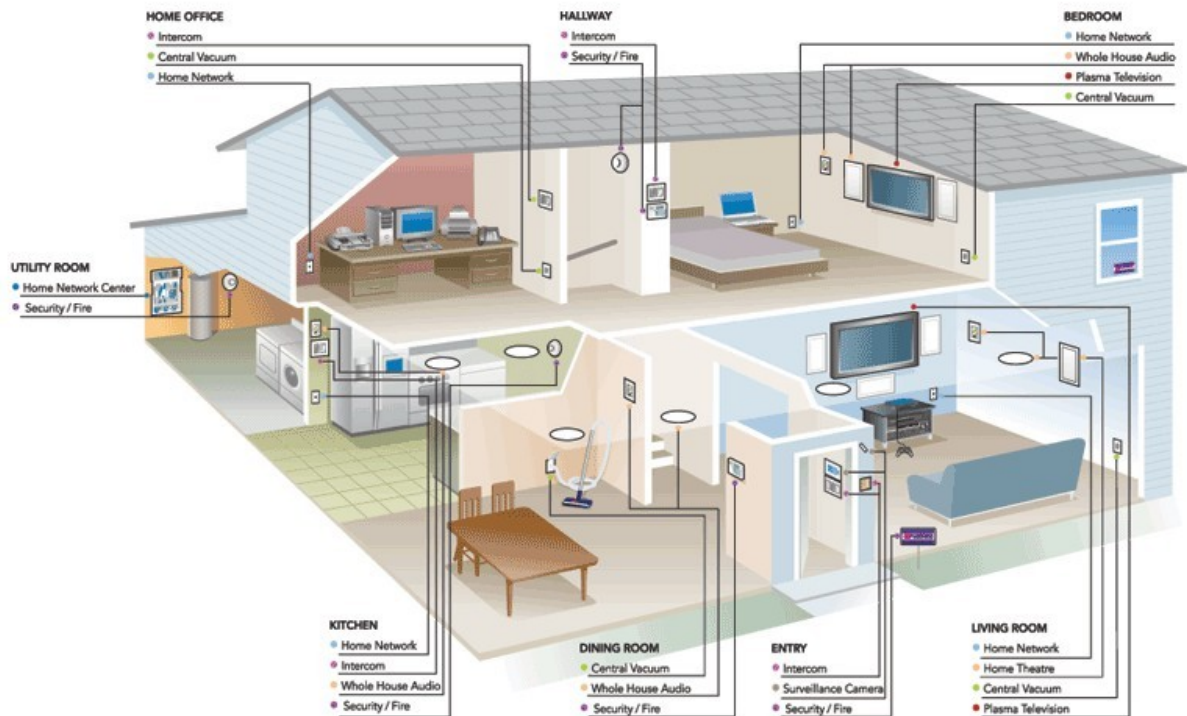
Το Internet of Things (IoT) είναι ένα δίκτυο αντικειμένων τα οποία μπορεί να είναι άνθρωποι ή συσκευές. Το κάθε αντικείμενο έχει ένα μοναδικό αναγνωριστικό και έχει την ικανότητα να επικοινωνεί με τα άλλα αντικείμενα μέσω του δικτύου χωρίς την παρέμβαση του ανθρώπου. Το IoT παρουσιάζει μεγάλη εξέλιξη από τότε που ο μεγαλύτερος όγκος της πληροφορίας μεταδίδεται ασύρματα.

Ένα αντικείμενο, στο Internet of Things, μπορεί να είναι ένα εμφύτευμα για την παρακολούθηση της καρδιάς ενός ανθρώπου, μέχρι ένα αυτοκίνητο που έχει ενσωματωμένους αισθητήρες για να ειδοποιεί τον οδηγό όταν η πίεση των ελαστικών είναι χαμηλή - ή οποιοδήποτε άλλο φυσικό ή τεχνητό αντικείμενο που μπορεί να εκχωρηθεί μια διεύθυνση IP και να έχει τη δυνατότητα να μεταφέρει δεδομένα μέσω ενός δικτύου. Μέχρι στιγμής, το Internet of things είναι πιο στενά συνδεδεμένο με το machine-to-machine (M2M) στον τομέα των κατασκευών και ενέργειας, πετρελαίου και φυσικού αερίου επιχειρήσεις κοινής ωφέλειας. Οι συσκευές που κατασκευάζονται με δυνατότητες επικοινωνίας M2M συχνά αναφέρονται ως έξυπνες. Τέτοιες συσκευές χρησιμοποιούνται ευρέως στο οικιακό αυτοματισμό.[28]

Home Automation

Το Home automation (αυτοματοποίηση οικίας) αναφέρεται στη χρήση των ηλεκτρονικών υπολογιστών και της τεχνολογίας των πληροφοριών για τον έλεγχο των οικιακών συσκευών και των στοιχείων της οικίας (όπως παράθυρα ή το φωτισμό). Τα συστήματα μπορούν να είναι από απομακρυσμένος έλεγχος του φωτισμού μέχρι πολύπλοκα δίκτυα υπολογιστών και μικροελεγκτών με διάφορους βαθμούς νοημοσύνης και αυτοματισμού. Το home automation έχει υιοθετηθεί για λόγους ευκολίας στη χρήση, ασφάλειας και εξοικονόμησης ενέργειας.

Στις σύγχρονες κατασκευές σε ανεπτυγμένες χώρες, τα περισσότερα σπίτια έχουν ενσύρματο δίκτυο για το ηλεκτρικό ρεύμα, το τηλέφωνο, το σήμα της τηλεόρασης, και το κουδούνι της πόρτας. Πολλές εργασίες του σπιτιού είχαν αυτοματοποιηθεί με την ανάπτυξη εξειδικευμένων αυτοματοποιημένων συσκευών που πλέον θεωρούνται δεδομένες σε ένα καθημερινό σπίτι. Όπως για παράδειγμα, τα πλυντήρια που έχουν αναπτυχθεί για να αποφευχθεί το πλύσιμο των ρούχων στο χέρι, καθώς και θερμοσίφωνες που αναπτύχθηκαν για την αποφυγή του ζεστάματος του νερού με άλλους τρόπους.



Εικόνα 2.1. Έξυπνο σπίτι

Η χρήση φυσικού αερίου, υγρών καυσίμων, καθώς και η χρήση της ηλεκτρικής ενέργειας αυξάνει την αυτοματοποίηση στον τομέα της θέρμανσης. Η ανάπτυξη των θερμοστατών επιτρέπεται τον αυτοματοποιημένο έλεγχο της θέρμανσης, και της ψύξης.

Καθώς ο αριθμός των ελεγχόμενων συσκευών στην οικία ανεβαίνει, η διασύνδεση και επικοινωνία μεταξύ των συσκευών παίζει πολύ σημαντικό ρόλο. Για παράδειγμα, ένας καυστήρας μπορεί να στείλει ένα μήνυμα προειδοποίησης όταν χρειάζεται καθαρισμό, ή ένα ψυγείο όταν χρειάζεται σέρβις. Επίσης, αν το σύστημα συναγερμού είναι ενεργοποιημένο, με το home automation θα μπορούσε να καλέσει τον ιδιοκτήτη, ή έναν αριθμό έκτακτης ανάγκης, εάν ανιχνευτεί ένας εισβολέας.

Σε απλές εγκαταστάσεις, ο αυτοματισμός μπορεί να είναι τόσο απλός όσο η ενεργοποίηση των φώτων όταν ένα άτομο εισέρχεται στο δωμάτιο. Σε πιο προηγμένες εγκαταστάσεις, τα δωμάτια μπορεί να καταλαβαίνουν όχι μόνο την παρουσία ενός ατόμου στο εσωτερικό, αλλά και ποιο είναι το πρόσωπο αυτό και να ρυθμίσει τις κατάλληλες παραμέτρους όπως είναι ο φωτισμός, η θερμοκρασία, τα επίπεδα έντασης της μουσικής ή σταθμών της τηλεόρασης, λαμβάνοντας υπόψη την ημέρα της εβδομάδας, την ώρα της ημέρας, και άλλους παράγοντες.

Άλλες αυτοματοποιημένες εργασίες μπορεί να περιλαμβάνουν μείωση του κλιματισμού ή της θέρμανσης, όταν το σπίτι είναι άδειο, και την αποκατάσταση της κανονικής ρύθμισης όταν ένα άτομο είναι έτοιμο να επιστρέψει. Πιο εξελιγμένα συστήματα μπορούν να διατηρούν έναν κατάλογο των προϊόντων που χρησιμοποιούνται, σημειώνοντας

τη χρήση τους μέσω των κωδικών τους, και να προετοιμάσουν μια λίστα με ψώνια ή ακόμα και να παραγγείλουν αυτόματα όταν εξαντλούνται.

Το Home automation μπορεί επίσης να παρέχει μια απομακρυσμένη διεπαφή για τις οικιακές συσκευές ή και για το ίδιο το σύστημα αυτοματισμού, για να παρέχουν έλεγχο και παρακολούθηση σε ένα smartphone ή σε μία Web σελίδα.

Ένα παράδειγμα της απομακρυσμένης παρακολούθησης οικιακού αυτοματισμού θα μπορούσε να ενεργοποιείται όταν ένας ανιχνευτής καπνού ανιχνεύσει μια κατάσταση πυρκαγιάς ή καπνού, κάνοντας όλα τα φώτα στο σπίτι να αναβοσβήνουν για να ειδοποιήσουν για την πιθανή έκτακτη ανάγκη. Αν το σπίτι είναι εξοπλισμένο με ένα home theater, το σύστημα οικιακού αυτοματισμού μπορεί να κλείσει όλα τα εξαρτήματα ήχου και βίντεο για να μην αποσπάται η προσοχή, ή για να γίνει μια ηχητική ανακοίνωση. Το σύστημα θα μπορούσε επίσης να καλέσει τον ιδιοκτήτη του σπιτιού στο κινητό του τηλέφωνο για να τον ειδοποιήσει, ή καλέσει την πυροσβεστική ή την εταιρεία security.

Πλεονεκτήματα και μειονεκτήματα

Πλεονεκτήματα

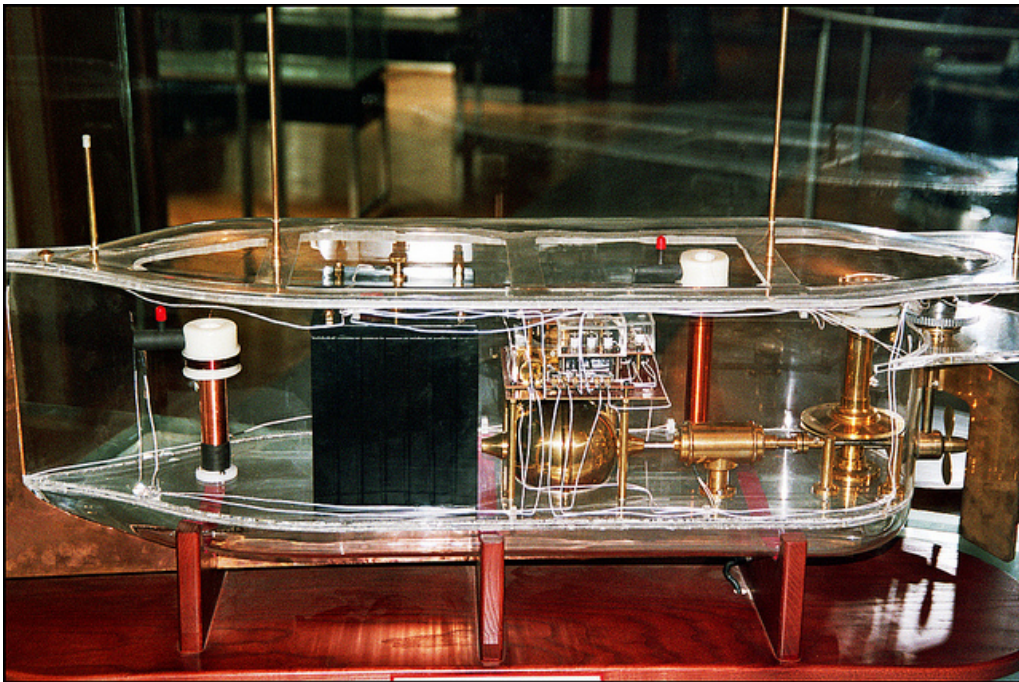
- Μείωση της κατανάλωσης ενέργειας μέσω του ευφυούς ελέγχου.
- Απόκτηση άνεσης μέσω έξυπνου ελέγχου: για παράδειγμα, μπορεί να παραχθεί σε ένα κουμπί ένας φωτισμού χωρίς να χρειάζεται να ενεργοποιηθούν ξεχωριστά πολλές λάμπες
- Προστασία από τις ληστές με την προσομοίωση της παρουσίας ατόμων στο σπίτι.
- Ασφάλεια για τους χρήστες με την προειδοποίηση όταν υπάρχουν καταστάσεις ανάγκης.
- Παρακολούθηση από μια εξωτερική υπηρεσία ασφαλείας με την αυτόματη μετάδοση των στοιχείων του συναγερμού.

Μειονεκτήματα

- Υψηλότερο αρχικό κόστος σε σύγκριση με την απλή εγκατάσταση του κτιρίου. Πολλές φορές όμως το κόστος αποσβένεται από την εξοικονόμηση ενέργειας κατά τη λειτουργία.
- Όταν υπάρχει υψηλή πολυπλοκότητα χρειάζεται εκπαίδευση για την σωστή λειτουργία των συστημάτων.
- Αυξημένη πιθανότητα δυσλειτουργίας του κτηρίου σε σύγκριση με την απλή εγκατάσταση που μπορεί να οφείλεται σε αστοχία υλικού ή σε σφάλματα του λογισμικού.

Ιστορική αναδρομή

Το Home automation αποτελούσε σενάριο επιστημονικής φαντασίας για πολλά χρόνια, αλλά εφαρμόζεται στην πραγματικότητα από τις αρχές του 20ου αιώνα, μετά την ευρεία είσοδο της ηλεκτρικής ενέργειας στο σπίτι, καθώς και την ταχεία πρόοδο της τεχνολογίας των πληροφοριών. Οι πρώτες συσκευές απομακρυσμένου ελέγχου άρχισαν να εμφανίζονται στα τέλη του 1800. Για παράδειγμα, ο Nikola Tesla κατοχύρωσε την ευρεσιτεχνία για τον απομακρυσμένο έλεγχο των σκαφών και οχημάτων το 1898.



Εικόνα 2.2. Το σκάφος με απομακρυσμένο έλεγχο του Nikola Tesla

Η εμφάνιση των ηλεκτρικών οικιακών συσκευών ξεκίνησε μεταξύ 1915 και 1920 όπου έγινε η αντικατάσταση των οικιακών εργασιών με φθηνές, μηχανικές συσκευές. Ο ηλεκτρισμός στα σπίτια όμως ήταν ακόμα σε πολύ πρώιμο στάδιο και έτσι οι συσκευές αυτές πολυτέλεια και εμφανίζονταν μόνο στα πιο εύπορα νοικοκυριά.

Παρόμοιες ιδέες με τα σύγχρονα συστήματα οικιακού αυτοματισμού πρωτοακούστηκαν κατά τη διάρκεια του World's Fairs της δεκαετίας του 1930. Σε εκθέσεις στο Σικάγο (1934), τη Νέα Υόρκη (1939) και (1964-1965), απεικονίζονται ηλεκτροδοτούμενα και αυτοματοποιημένα σπίτια. Το 1966 ο Jim Sutherland, ένας μηχανικός που εργαζόταν για την Westinghouse Electric, ανέπτυξε ένα σύστημα οικιακού αυτοματισμού που ονομαζόταν "ECHO IV". Τα πρώτα "καλωδιωμένα σπίτια" χτίστηκαν από αμερικανούς ερασιτέχνες κατά τη διάρκεια της δεκαετίας του 1960, αλλά περιορίζονταν από την τεχνολογία της εποχής. Ο όρος "έξυπνο σπίτι" επινοήθηκε για πρώτη φορά από την αμερικανική ένωση των Housebuilders το 1984.

Με την εφεύρεση του μικροελεγκτή, το κόστος του ηλεκτρονικού ελέγχου μειώθηκε ραγδαία. Οι κατασκευαστές υιοθέτησαν τεχνολογίες απομακρυσμένου και ευφυούς ελέγχου και έτσι αναπτύχθηκε ακόμα περισσότερο η ιδέα του "Έξυπνου σπιτιού".

Μέχρι το τέλος της δεκαετίας του 1990, το "domotics" χρησιμοποιούνταν συνήθως για να περιγράψει οποιοδήποτε σύστημα στο οποίο οι δραστηριότητες στο σπίτι συνδέονταν με τον τομέα της πληροφορικής και της τηλεματικής. Η φράση είναι ένας νεολογισμός που σχηματίζεται από το Domus (στα λατινικά σημαίνει σπίτι) και την πληροφορική, και αναφέρεται στην εφαρμογή των ηλεκτρονικών υπολογιστών και των ρομπότ για οικιακές συσκευές. Παρακάτω παρουσιάζεται αναλυτικότερα η έννοια domotics. Η έννοια "Domotique" εισήχθη αρχικά στη Γαλλία στη δεκαετία του 1980 και κατά τη διάρκεια της δεκαετίας του 1990 εισήχθη στην Ισπανία και την Ιταλία ως «Domótica», και αναφέρεται στον οικιακό αυτοματισμό.

Παρά το ενδιαφέρον οικιακού αυτοματισμού, από τα τέλη της δεκαετίας του 1990 δεν υπήρχε αξιοσημείωτη ανάπτυξη, αφού τέτοια συστήματα χρησιμοποιούνται ακόμη και σήμερα από ερασιτέχνες ή πολύ πλούσιους. Η έλλειψη ενός ενιαίου και απλουστευμένου πρωτόκολλου καθώς και το υψηλό κόστος της εγκατάστασης απωθεί τους καταναλωτές από την κατασκευή έξυπνων σπιτιών.

Αν και υπάρχουν ακόμη πολλά περιθώρια ανάπτυξης, σύμφωνα με την ABI Research, 1,5 εκατομμύρια συστήματα οικιακού αυτοματισμού έχουν εγκατασταθεί στις ΗΠΑ το 2012, και προβλέπεται να παρατηρηθεί απότομη αύξηση στα 8 εκατομμύρια μέχρι το 2017.

Η έννοια "Domotics"

Ο Γάλλος δημοσιογράφος Bruno de Latour επινόησε τον όρο domotic το 1984. Η έννοια Domotic έχει εισαχθεί πρόσφατα στο λεξιλόγιο ως σύνθετη λέξη από τις λατινικές λέξεις domus, που όπως αναφέραμε και προηγουμένως σημαίνει σπίτι και informatics που σημαίνει πληροφορική, και αναφέρεται σε ευφυείς οικίες, δηλαδή εκεί όπου εφαρμόζεται η χρήση των τεχνολογιών της αυτοματοποίησης και της επιστήμης των υπολογιστών.

Ο όρος καλύπτει ένα ευρύ φάσμα εφαρμογών της τεχνολογίας των πληροφοριών για τα προβλήματα του οικιακού αυτοματισμού. Domotics είναι η μελέτη της υλοποίησης ευφυούς περιβάλλοντος στο σπίτι.

Τα "ψηφιακά σπίτια" περιλαμβάνουν οικιακό αυτοματισμό, πολυμέσα, τηλεπικοινωνίες, ηλεκτρονικό εμπόριο, κλπ, μέσω οικιακών δικτύων και αυτοματισμό σπιτιού που σημαίνει ότι τα συστήματα του σπιτιού μπορούν να "μιλούν" μεταξύ τους για βελτιωμένη άνεση, την αποτελεσματικότητα και την ασφάλεια.

Τα απαραίτητα στοιχεία για τον οικιακό αυτοματισμό

Τα στοιχεία ενός συστήματος οικιακού αυτοματισμού μπορεί να είναι: αισθητήρες (όπως για τη θερμοκρασία, το φως της ημέρας, ή την ανίχνευση κίνησης), ελεγκτές (όπως ένα γενικής χρήσης προσωπικού υπολογιστή ή ένα ειδικό ελεγκτή αυτοματισμού), ενεργοποιητές, (όπως μηχανοκίνητες βαλβίδες, διακόπτες φωτός και συσκευών), δίαυλοι (ενσύρματη ή ασύρματη)? και διεπαφές (ανθρώπου-μηχανής και / ή μηχανής-μηχανής).

Απαιτείται μία ή περισσότερες διεπαφές ανθρώπου-μηχανής ή / και μηχανής-προς-μηχανή, έτσι ώστε οι κάτοικοι του σπιτιού να μπορούν να αλληλεπιδρούν με το σύστημα για την παρακολούθηση και τον έλεγχο του. Αυτό μπορεί να αποτελείται από ένα εξειδικευμένο τερματικό σταθμό ή, πλέον στις μέρες μας, μπορεί να είναι μια εφαρμογή που τρέχει σε ένα έξυπνο κινητό τηλέφωνο ή σε ένα tablet ή ακόμα και σε μία web σελίδα. Οι συσκευές μπορούν να επικοινωνούν μέσω μεμονωμένης καλωδίωσης, ή συνδέοντας τις όλες σε ένα ενσύρματο δίκτυο, ή ασύρματα, χρησιμοποιώντας ένα ή περισσότερα πρωτόκολλα. Έτσι μπορούμε να χρησιμοποιήσουμε ένα κεντρικό χειριστήριο καθώς επίσης μπορούμε να έχουμε και μεμονωμένο χειρισμό κάθε συσκευής στο σπίτι.

Εφαρμογές

Συστήματα ψύξης θέρμανσης και εξαερισμού

Τα συστήματα κλιματισμού, θέρμανσης και εξαερισμού (HVAC) μπορεί να περιλαμβάνουν έλεγχο της θερμοκρασίας και της υγρασίας, καθώς και την ανανέωση του αέρα όταν κρίνεται απαραίτητο. Ένας θερμοστάτης που διαχειρίζεται από μία web σελίδα επιτρέπει στον ιδιοκτήτη σπιτιού τον έλεγχο των συστημάτων θέρμανσης και κλιματισμού του κτιρίου εξ αποστάσεως. Το σύστημα μπορεί αυτόματα να ανοίξει και να κλείσει τα παράθυρα για την ψύξη του σπιτιού.

Φωτισμός

Συστήματα ελέγχου φωτισμού μπορούν να χρησιμοποιηθούν για τον έλεγχο των οικιακών φώτων. Τα φώτα μπορεί να ελεγχθούν με χρονοδιακόπτη, ή να καταλαβαίνουν αυτόματα τότε υπάρχει κάποιο άτομο σε ένα δωμάτιο. Οι ηλεκτρονικά ελεγχόμενες λάμπες μπορούν επίσης να ελέγχονται ως προς τη φωτεινότητα ή το χρώμα για να παρέχουν διαφορετικά επίπεδα φωτός για διάφορες εργασίες. Ο φωτισμός μπορεί να ελέγχεται εξ αποστάσεως μέσω ενός ασύρματου ελέγχου ή μέσω του internet. Ο φυσικός φωτισμός (φως της ημέρας) μπορεί να χρησιμοποιηθεί για να ελέγχει αυτόματα τη σκίαση ενός παράθυρου και τις κουρτίνες για να γίνεται όσο το δυνατόν καλύτερη αξιοποίηση του φυσικού φωτός για την εξοικονόμηση ενέργειας.

Οπτικοακουστικά ερεθίσματα

Αυτή η κατηγορία περιλαμβάνει την αναπαραγωγή του βίντεο καθώς και την αναπαραγωγή και την ένταση ήχου. Πολλαπλές πηγές ήχου ή βίντεο μπορεί να έχουν επιλεγθεί να είναι διαθέσιμες σε ένα ή περισσότερα δωμάτια και μπορεί να συνδέονται με το φωτισμό και τις κουρτίνες για να παρέχουν την ανάλογη ατμόσφαιρα.

Σκίαση

Ο αυτόματος έλεγχος στα πατζούρια και στις κουρτίνες μπορεί να χρησιμοποιηθεί για:

- προσομοίωση παρουσίας
- εξασφάλιση ιδιοτηκότητας
- έλεγχο της θερμοκρασίας
- έλεγχος φωτεινότητας
- Ασφάλεια (στην περίπτωση των παντζουριών)

Ασφάλεια

Ένα σύστημα ασφαλείας με ενσωματωμένο ένα σύστημα οικιακού αυτοματισμού μπορεί να παρέχει πρόσθετες υπηρεσίες, όπως η εξ αποστάσεως παρακολούθηση των καμερών ασφαλείας μέσω του Διαδικτύου, ή το κεντρικό κλείδωμα όλων των θυρών και των παραθύρων περιμετρικά.

Με τον οικιακό αυτοματισμό, ο χρήστης μπορεί να επιλέξει και να παρακολουθεί τις κάμερες ζωντανά από μια πηγή Internet στο σπίτι ή την επιχείρησή τους. Τα συστήματα ασφαλείας μπορεί να περιλαμβάνουν αισθητήρες κίνησης, που θα ανιχνεύουν κάθε είδους ανεπιθύμητη κίνηση και ενημερώνουν τον χρήστη μέσω του συστήματος ασφαλείας ή μέσω κινητού τηλεφώνου.

Το σύστημα αυτοματισμού μπορεί να προσομοιώσει την παρουσία μας στο σπίτι με την αυτόματη ρύθμιση του φωτισμού ή των πατζουριών. Μπορούν επίσης να ενσωματωθούν συστήματα ανίχνευσης πυρκαγιάς, διαρροής αερίου, μονοξειδίου του άνθρακα, ή ακόμα και διαρροές νερού. Ακόμη, τα συστήματα συναγερμού επιτρέπουν σε κάποιον τραυματία στο σπίτι να καλέσει εύκολα βοήθεια.

Ενδοεπικοινωνία

Ένα σύστημα ενδοεπικοινωνίας μπορεί να επιτρέψει την επικοινωνία μέσω ενός μικροφώνου και ηχείου μεταξύ των διαφορετικών δωματίων. Η ένταξη της ενδοεπικοινωνίας στο τηλέφωνο, ή του συστήματος θυροτηλεόρασης στην τηλεόραση, μπορεί να επιτρέψει στους χρήστες να βλέπουν την κάμερα της πόρτας αυτόματα.

Καταγραφείς στοιχείων (Loggers)

Μια ξεχωριστή κατηγορία εφαρμογών που η εφαρμογή της δεν περιορίζεται μόνο στον οικιακό αυτοματισμό αλλά εμφανίζεται και σε επαγγελματικούς χώρους είναι η καταγραφή στοιχείων του χώρου μέσω αισθητήρων για τη μελλοντική επεξεργασία τους και εξαγωγή κρίσιμων συμπερασμάτων.

Στον οικιακό αυτοματισμό μπορεί να περιλαμβάνουν την καταγραφή της θερμοκρασίας σε όλους τους χώρους εντός και εκτός του σπιτιού. Ακόμη, μπορεί να υπάρχουν καταγραφείς για την κατανάλωση του ρεύματος από κάθε συσκευή ξεχωριστά ή και από κάθε διαφορετικό χρήστη του σπιτιού. Επίσης, χρησιμοποιούνται για την καταγραφή στοιχείων όπως το πότε άνοιξε μια πόρτα ή ένα παράθυρο ή ακόμα και το πόσες ώρες μιας ημέρας ήταν οι κάτοικοι του σπιτιού μέσα σε αυτό. Τέλος, θα μπορούσε να χρησιμοποιηθεί για την αναλυτική καταγραφή της λειτουργίας όλων των συσκευών του σπιτιού έτσι ώστε ο ιδιοκτήτης να έχει τον πλήρη έλεγχό τους.

Εκτός από τον οικιακό αυτοματισμό τέτοιοι καταγραφείς μπορεί να εμφανίζονται εσωτερικά σε μία συσκευή για να παρέχουν στοιχεία για τον πλήρη έλεγχο της ίδιας καθώς επίσης και σε μία πληθώρα εφαρμογών όπως είναι

- παρακολούθηση μη επανδρωμένου μετεωρολογικού σταθμού (όπως ταχύτητα κατεύθυνση ανέμου, η θερμοκρασία, η σχετική υγρασία, ηλιακή ακτινοβολία).
- παρακολούθηση μη επανδρωμένου φράγματος νερού (όπως το επίπεδο του νερού, το βάθος του νερού, η ροή του νερού, το pH του νερού, η αγωγιμότητα του νερού).
- παρακολούθηση επιπέδου υγρασίας του εδάφους.
- παρακολούθηση της πίεσης ενός αερίου.
- offshore πλατφόρμες για την παρακολούθηση των συνθηκών του περιβάλλοντος.
- παρακολούθηση της οδικής κυκλοφορίας.
- παρακολούθηση της θερμοκρασίας ή της υγρασίας κατά τη διάρκεια της μεταφοράς τροφίμων
- μέτρηση της διακύμανσης της έντασης του φωτός.
- παρακολούθηση της διαδικασίας συντήρησης και την αντιμετώπισης προβλημάτων εφαρμογών.
- παρακολούθηση της διαδικασίας για την εξακρίβωση του τρόπου χρήσης μιας συσκευής.
- παρακολούθηση του περιβάλλοντος.
- Έλεγχος Οχημάτων (συμπεριλαμβανομένων των δοκιμών σύγκρουσης)

- καταγραφή κατανάλωσης ενέργειας ανάλογα με το τον χρήστη που λειτουργεί τη συσκευή
- παρακολούθηση της θερμοκρασίας, της υγρασίας και της ενέργειας για την μελέτη της αποτελεσματικότητας της θέρμανσης και του κλιματισμού.
- παρακολούθηση των στοιχείων ενός πλοίου όπως είναι οι τάσεις των τοιχωμάτων ανάλογα με το φορτίο ή και η κατανάλωση των καυσίμων

Ειδικότερα, οι καταγραφείς θερμοκρασίας μπορούν να φανούν χρήσιμοι για την εξοικονόμηση χρημάτων και ενέργειας σε εγκαταστάσεις ή αποθήκες τροφίμων όπου οι συνθήκες παίζουν πολύ σημαντικό ρόλο στην καλή διατήρηση των εμπορευμάτων. Ακόμη, μπορούν να χρησιμοποιηθούν για την διαπίστωση της καλής και συνεχούς λειτουργίας συσκευών και την αποφυγή της υπερθέρμανσης και της βλάβης σε κάποια από αυτές. Τέλος, με τέτοιους καταγραφείς μπορούμε να ελέγχουμε τα στατιστικά στοιχεία στη διάρκεια του χρόνου για την μέγιστη δυνατή απόδοση των συσκευών μας.

Τα πρωτόκολλα

Τα πρωτόκολλα επικοινωνίας αποτελούν ένα από τα σοβαρότερα ζητήματα για τον οικιακό αυτοματισμό καθώς σε περίπτωση που οι συσκευές που είναι εγκατεστημένες στην οικία δεν υποστηρίζουν κοινά πρωτόκολλα η επικοινωνία μεταξύ τους θα είναι αδύνατη. Έτσι, θα πρέπει να έχουμε προβλέψει έτσι ώστε να υπάρχει συμβατότητα μεταξύ των έξυπνων συσκευών που θα επιλέξουμε.

Τα περισσότερα πρωτόκολλα έχουν φτιαχτεί για ασύρματη επικοινωνία μεταξύ των συσκευών καθώς έτσι αποφεύγονται τα πολλά καλώδια και η δυσκολία εγκατάστασης σε ένα ήδη υπάρχον σπίτι. Τα πιο διαδεδομένα πρωτόκολλα που υποστηρίζουν ασύρματη επικοινωνία είναι το Z-Wave, ZigBee.

Παρακάτω παρατίθενται οι ιδιότητες από κάποια από τα γνωστότερα πρωτόκολλα επικοινωνίας του οικιακού αυτοματισμού.

Protocol ♦	Power Line ♦	Radio-Frequency ♦	Data Rate ♦	Available API? ♦	Open Source ♦	Needs Neutral Wire? ♦
C-Bus	no	yes	3500 bps	yes	no	n/a (uses category-5 UTP)
EnOcean	no	902 MHz (North America)	9600 bps	yes	?	?
Insteon	yes	yes	13,165 bps	yes	no	Usually
KNX	yes	yes	9600 bps	yes	no	no ^[citation needed]
UPB	yes	no	480 bps	no	no	no
X10	yes	no	20 bps	yes	no	Sometimes
Zigbee	no	2.4 GHz (worldwide) 915 MHz (Americas and Australia) 868 MHz (Europe)	20-250k bps	yes	no	Usually
Z-Wave	no	yes	250k bps	yes	no	Usually (Z-wave list that need Neutral [Ⓞ])

Κόστος

Οι δαπάνες περιλαμβάνουν κυρίως εξοπλισμό, εξαρτήματα, καθώς και το κόστος της εγκατάστασης. Επίσης, σαν πάγιες δαπάνες θα πρέπει να θεωρηθούν το κόστος της ηλεκτρικής ενέργειας για να λειτουργεί το σύστημα ελέγχου, το κόστος συντήρησης για τα συστήματα ελέγχου και τη δικτύωση, συμπεριλαμβανομένων αντιμετώπιση προβλημάτων, καθώς και ενδεχόμενο κόστος της αναβάθμισης καθώς αλλάζουν τα πρότυπα. Η αυξημένη πολυπλοκότητα μπορεί επίσης να αυξήσει το κόστος συντήρησης για δικτυωμένες συσκευές.

Επίσης για την ασφάλεια του συστήματος ελέγχου μπορεί να απαιτείται ένα υψηλό ποσό, ειδικά εάν το σύστημα ελέγχου εκτείνεται πέρα από το σπίτι, και σε ασύρματα δίκτυα ή στο διαδίκτυο.[27]

2. Το υλικό

I. Raspberry Pi

Το Raspberry Pi είναι ένας υπολογιστής στο μέγεθος μίας πιστωτικής κάρτας. Τα βασικά μοντέλα παραγωγής είναι δύο, το Model A και το Model B ενώ πλέον είναι διαθέσιμη και μία βελτιωμένη έκδοση του Model B που είναι το Model B+. Οι διαφορές που υπάρχουν μεταξύ του Model A, του Model B και του Model B+ παρουσιάζονται στον παρακάτω πίνακα [1]

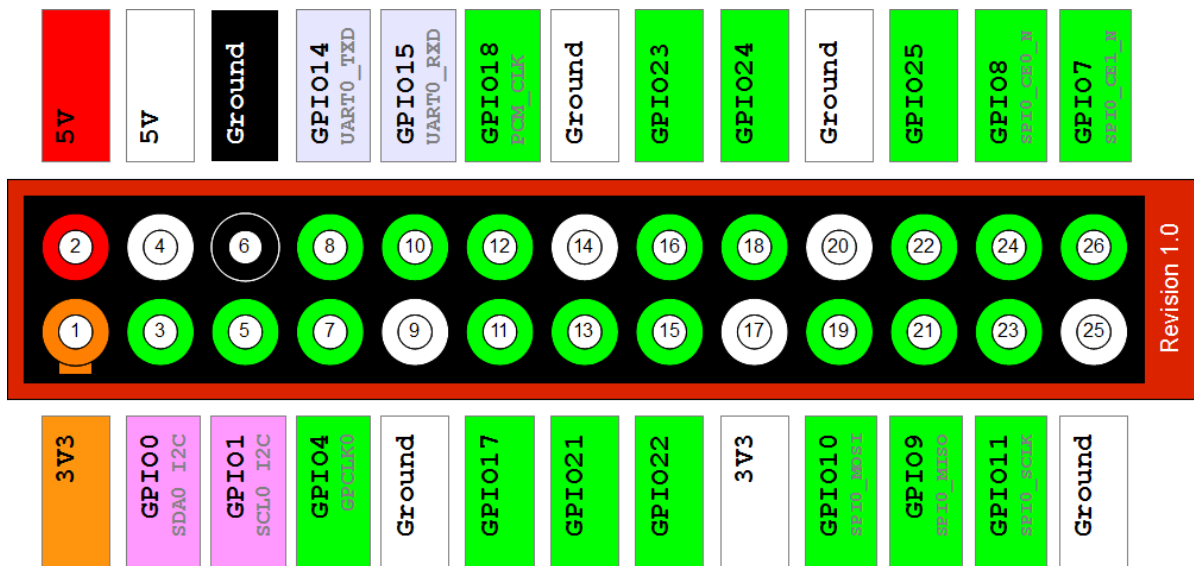
	Model A	Model B	Model B+
Target price:	US\$25	US\$35 ^{[29][30]}	
SoC:	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM, and single USB port) ^{[1][31]}		
CPU:	700 MHz ARM1176JZF-S core (ARM11 family, ARMv6 instruction set) ^[1]		
GPU:	Broadcom VideoCore IV @ 250 MHz ^{[32][33]} OpenGL ES 2.0 (24 GFLOPS) MPEG-2 and VC-1 (with license ^[34]), 1080p30 h.264/MPEG-4 AVC high-profile decoder and encoder ^[1]		
Memory (SDRAM):	256 MB (shared with GPU)	512 MB (shared with GPU) as of 15 October 2012	
USB 2.0 ports: ^[11]	1 (direct from BCM2835 chip)	2 (via the on-board 3-port USB hub) ^[35]	4 (via the on-board 5-port USB hub) ^{[35][36]}
Video input:	15-pin MIPI camera interface (CSI) connector, used with the Raspberry Pi Camera or Raspberry Pi NoIR Camera. ^[37]		
Video outputs:	Composite video (PAL and NTSC) (in Models A and B, via RCA jack; in Model B+, via 3.5 mm TRRS jack shared with audio out), HDMI (rev 1.3 & 1.4), ^[40] MIPI display interface (DSI) for raw LCD panels ^{[41][42]} 14 HDMI resolutions from 640×350 to 1920×1200 plus various PAL and NTSC standards. ^[37]		
Audio outputs:	analog audio via 3.5 mm phone jack, HDMI, and, as of revision 2 boards, I ² S audio ^[44] (also potentially for audio input)		
Onboard storage: ^[11]	SD / MMC / SDIO card slot (3.3 V with card power only)		MicroSD ^[29]
Onboard network: ^[11]	None	10/100 Mbit/s Ethernet (8P8C) USB adapter on the third/fifth port of the USB hub ^[35]	
Low-level peripherals:	8× GPIO, ^[45] plus the following, which can also be used as GPIO: UART, I ² C bus, SPI bus with two chip selects, I ² S audio ^[46] +3.3 V, +5 V, ground ^{[32][47]}		17× GPIO plus the same specific functions, and HAT ID bus
Power ratings:	300 mA (1.5 W) ^[48]	700 mA (3.5 W)	600 mA (3.0 W) ^[29]
Power source:	5 V via MicroUSB or GPIO header		
Size:	85.60 mm × 56 mm (3.370 in × 2.205 in) – not including protruding connectors		
Weight:	45 g (1.6 oz)		

Εικόνα 2.1. Σύγκριση του Raspberry Pi Model A, Model B και Model B+



Εικόνα 2.2. Raspberry Pi Model B

Τα Raspberry Pi Model A και Model B έχουν 26 pins ενώ του Model B+ έχει 40 pins που χρησιμοποιούνται για Είσοδο-Εξοδο και ονομάζονται GPIO (General Purpose Input Output).



Εικόνα 2.3. GPIO Model B

Αρχικά σχεδιάστηκε για την ευκολότερη εκμάθηση και κατανόηση του προγραμματισμού υπολογιστών σε μαθητές καθώς η τιμή του είναι εξαιρετικά χαμηλή. Επειδή το Raspberry Pi λειτουργεί όπως ένας κανονικός υπολογιστής, χωρίς βέβαια να έχει την ίδια απόδοση με έναν υπολογιστή τελευταίας τεχνολογίας, με τις επιπλέον δυνατότητες

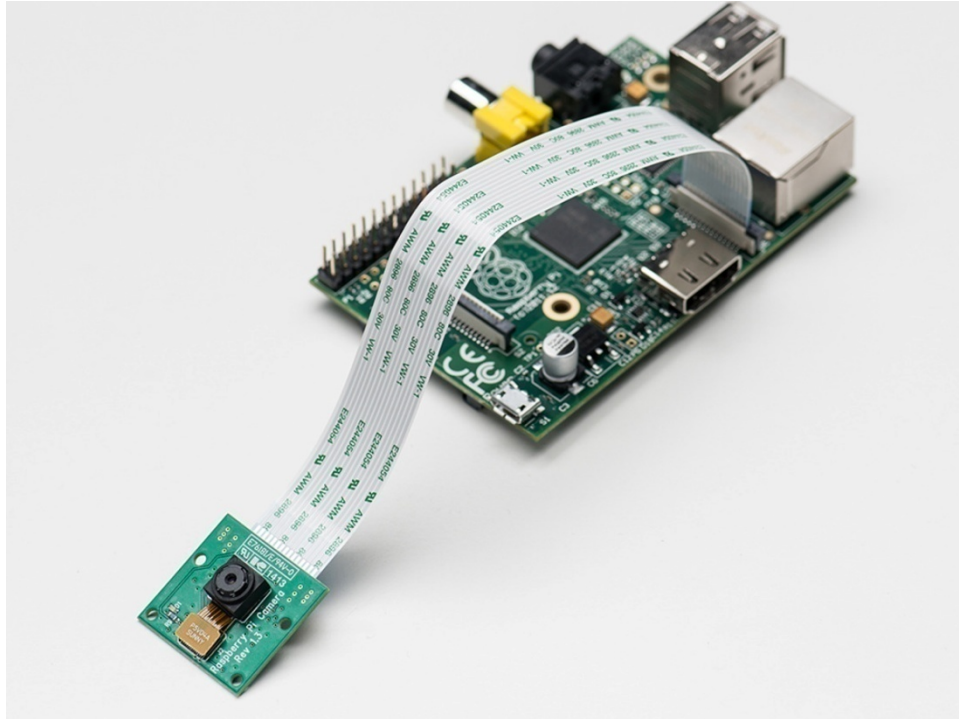
που δίνουν τα GPIO μπορούν να πραγματοποιηθούν διάφορες εφαρμογές όπως Media Center, Arcade Machine, Smart TV, Robotics, RPi cloud server, RPi Μετεωρολογικός Σταθμός, για αυτοματισμούς σπιτιού, Baby MONITOR, για αποθήκευση δικτύου, Streaming internet radio, Mini web server, FTP server, Proxy server, Firewall και πολλά άλλα.[2]

Η δυνατότητα σύνδεσης πληκτρολογίου και ποντικιού μέσω USB καθώς και οθόνης μέσω HDMI το κάνουν να μην υπολείπεται σε τίποτα ενός Η/Υ. Επιπλέον, η δυνατότητα σύνδεσης οθόνης αφής καθώς και κεραίας wifi μπορούν να κάνουν το Raspberry Pi να χρησιμοποιείται σαν ένα tablet.



Εικόνα 2.4. Raspberry Pi με οθόνη αφής

Ακόμη, είναι πολύ διαδεδομένη η χρήση του πρόσθετου της κάμερας που επιτρέπει την παρακολούθηση χώρων εξ αποστάσεως αλλά και την λήψη και επεξεργασία απλών φωτογραφιών.

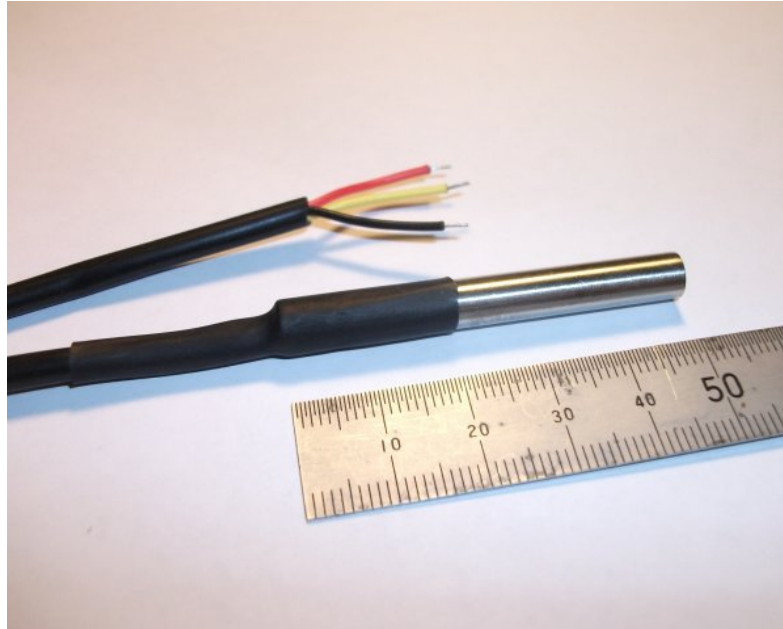


Εικόνα 2.5. Camera Module για το Raspberry Pi

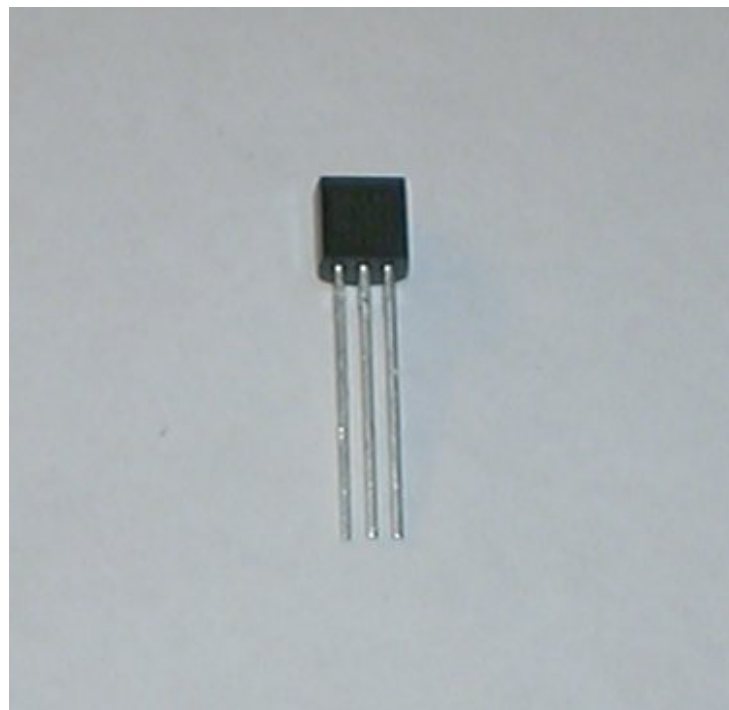
Επίσης, το Raspberry Pi μπορεί να χρησιμοποιηθεί εύκολα με ενσωματωμένα συστήματα καθώς από το 2014 η Raspberry Pi Foundation ξεκίνησε το Compute Module που μετατρέπει το Raspberry Pi σε ένα SODIMM 200-pin module.[3]

II. SWE0a Temperature Sensor

Αισθητήρας DS18B20 που περικλείεται από ανοξείδωτο ατσάλι. Το μαύρο καλώδιο αποτελεί τη γείωση, το κόκκινο καλώδιο την τάση 5V και το κίτρινο καλώδιο το δίαυλο των δεδομένων.[7]



Εικόνα 2.6. Αισθητήρας θερμοκρασίας SWE0a



Εικόνα 2.7. Αισθητήρας θερμοκρασίας DS18B20

1-wire

Το 1-Wire είναι ένα σύστημα διαύλου επικοινωνίας συσκευών που έχει σχεδιαστεί από την Dallas Semiconductor Corp., και παρέχει τα δεδομένα χαμηλής ταχύτητας, της σήμανσης, και την εξουσία πάνω από ένα ενιαίο σήμα. Το 1-Wire είναι παρόμοιο με το PC, αλλά με χαμηλότερους ρυθμούς δεδομένων και μεγαλύτερη εμβέλεια. Συνήθως χρησιμοποιείται για την επικοινωνία με μικρές φθηνές συσκευές, όπως ψηφιακά θερμομέτρα και άλλους αισθητήρες μέτρησης καιρικών χαρακτηριστικών. Ένα δίκτυο των συσκευών 1-Wire με ένα κεντρικό ελεγκτή ονομάζεται MicroLAN.

Ένα ιδιαίτερο χαρακτηριστικό του διαύλου είναι η δυνατότητα χρησιμοποίησης μόνο δύο καλωδίων: των δεδομένων και της γείωσης. Για να επιτευχθεί αυτό, οι συσκευές 1-wire περιλαμβάνουν ένα πυκνωτή 800 pF για την αποθήκευση φορτίο, και την φόρτισή του όταν η γραμμή δεδομένων είναι ανενεργή.

Ανάλογα με τη λειτουργία, οι συσκευές 1-wire είναι διαθέσιμες και ως μεμονωμένα εξαρτήματα σε ένα ολοκληρωμένο κύκλωμα, και σε ορισμένες περιπτώσεις, είναι φορητές, ονομάζονται iButton και μοιάζουν με μια μπαταρία ρολογιού. Οι κατασκευαστές παράγουν επίσης συσκευές πιο περίπλοκες από ότι ένα μεμονωμένο εξάρτημα που χρησιμοποιούν το δίαυλο 1-wire για να επικοινωνούν.



Εικόνα 2.8. Java Ring με ενσωματωμένο iButton

Οι συσκευές 1-Wire μπορεί να είναι ένα από τα πολλά στοιχεία ενός κυκλώματος της πλακέτας κάποιου προϊόντος, ή ένα μεμονωμένο εξάρτημα σε συσκευές όπως αισθητήρες θερμοκρασίας. Κάποια συστήματα συλλογής δεδομένων και ελέγχου συνδέονται με συσκευές 1-Wire με τη χρήση καλωδίων Modular connector (καλώδιο ethernet, τηλεφωνικό) ή με CAT-5 καλώδιο, με τις συσκευές να τοποθετείται στα ανάλογα βύσματα.

Συστήματα αισθητήρων μπορούν να κατασκευαστεί με την σύνδεση πολλών στοιχείων 1-Wire. Κάθε στοιχείο περιέχει όλα τη λογική που απαιτείται για να λειτουργήσει ο δίαυλος 1-Wire. Για παράδειγμα, μπορούμε να έχουμε αισθητήρες τάσης και έντασης, καταγραφείς θερμοκρασίας και υγρασίας. Αυτά μπορεί να συνδεθούν με έναν υπολογιστή χρησιμοποιώντας ένα μετατροπέα διαύλων. Το USB, η σειριακή RS-232, και η παράλληλη θύρα είναι διαδεδομένες λύσεις για τη σύνδεση του MicroLan στον κεντρικό υπολογιστή.

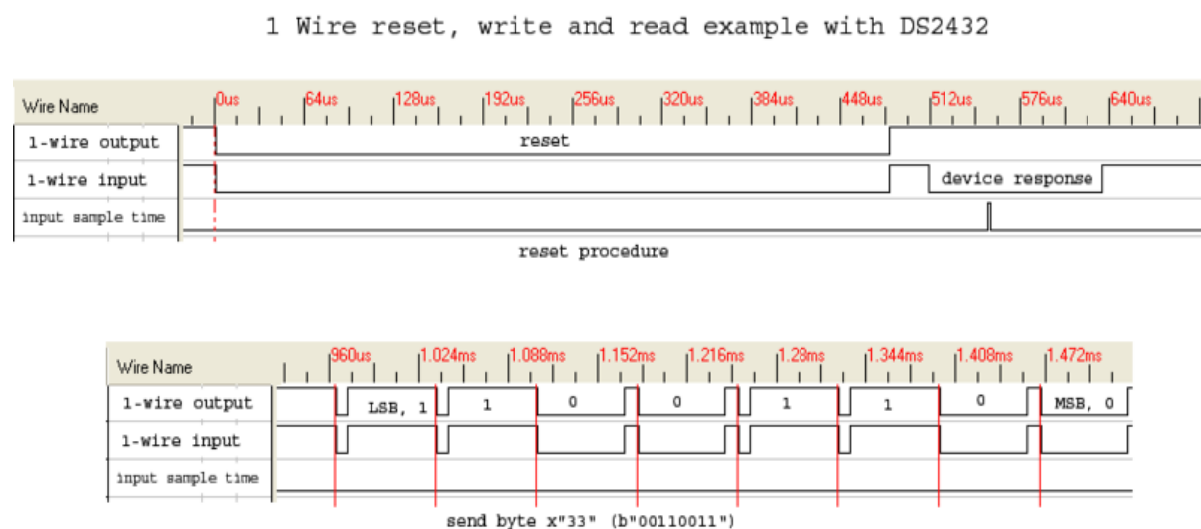
Το iButton (επίσης γνωστό ως το Κλειδί του Ντάλας) είναι ένα μηχανικό πρότυπο συσκευασίας που τοποθετεί ένα συστατικό 1-Wire μέσα σε ένα μικρό ανοξείδωτο ατσάλινο "κουμπί" παρόμοια με μια στρογγυλή μπαταρία ρολογιού. Τα iButtons που συνδέονται με τα συστήματα διαύλων 1-Wire μέσω των υποδοχών με τις επαφές που αγγίζουν το "καπάκι" και "βάση" του κουμπιού.

Κάθε τσιπ 1-Wire έχει ένα μοναδικό κωδικό που αποτελεί και το αναγνωριστικό του. Αυτό το χαρακτηριστικό καθιστά το τσιπ, ειδικά σε ένα iButton, κατάλληλο για χρήση ως ένα κλειδί για να ανοίξει μια κλειδαριά, την απενεργοποίηση συστημάτων συναγερμού, τον έλεγχο ταυτότητας των χρηστών του συστήματος του υπολογιστή, κλπ.

Πολλές συσκευές μπορούν να μοιράζονται τον ίδιο διάυλο. Κάθε συσκευή στο διάυλο έχει ένα μοναδικό σειριακό αριθμό 64-bit. Το λιγότερο σημαντικό byte του σειριακού αριθμού είναι ένας αριθμός 8-bit που λέει τον τύπο της συσκευής. Το πιο σημαντικό byte είναι ένα πρότυπο (για το διάυλο 1-wire) 8-bit CRC. Ο χώρος διευθύνσεων 64-bit αναζητείται ως ένα δυαδικό δέντρο, επιτρέποντας τον εντοπισμό έως και 75 συσκευών ανά δευτερόλεπτο.

Για να γίνει μία καινούρια σύνδεση ο κεντρικός ελεγκτής ξεκινά μια εκπομπή με ένα παλμό επαναφοράς, η οποία στέλνει σήμα 0 στο καλώδιο για τουλάχιστον 480 μs. Αυτό επαναφέρει κάθε slave συσκευή στο διάυλο. Μετά από αυτό, οποιαδήποτε slave συσκευή, αν υπάρχει, δείχνει ότι υπάρχει με ένα παλμό παρουσίας: κρατάει το διάυλο στο 0 για τουλάχιστον 60 μs αφού τον έχει απελευθερώσει ο κεντρικός ελεγκτής.

Για να σταλεί ένα λογικό "1", ο κεντρικός ελεγκτής του διαύλου στέλνει ένα πολύ σύντομο (1-15 μs) παλμό λογικού "0". Για να σταλεί ένα λογικό "0", στέλνει ένα παλμό λογικού "0" 60 μs όπως φαίνεται παρακάτω.[9]



Εικόνα 2.9. Διαδικασίες Reset, αποστολής λογικού "1" και λογικού "0"

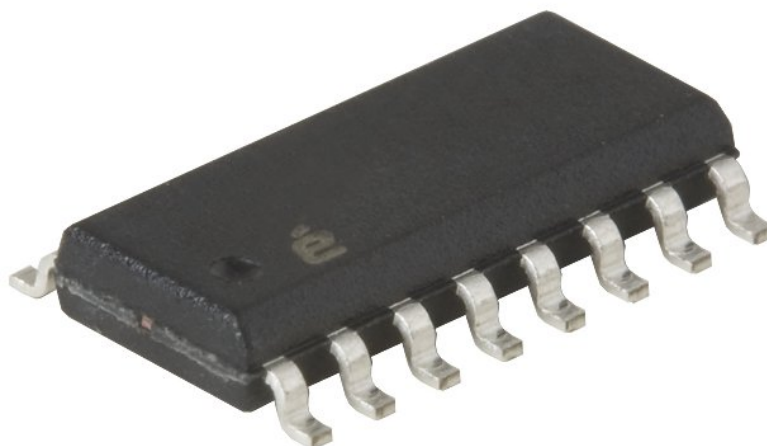
III. RPI3 I2C to 1-Wire Host Adapter

Μετατροπέας 8 καναλιών I2C σε 1-wire. Τα δύο κανάλια (bus.0 και bus.1) συνδέονται με τις δύο ενσύρματες υποδοχές RJ45 (CN2 και CN3) στην αριστερή πλευρά της μονάδας (πάνω από την τροφοδοσία και την κάρτα SD). Τα άλλα έξι κανάλια τοποθετημένα σε δύο ομάδες των τριών και συνδέονται με τις ενσύρματες RJ45 (CN4 και CN5) στο κάτω μέρος της μονάδας (πάνω από την υποδοχή HDMI).



Εικόνα 2.10. RPI3 I2C to 1-Wire Host Adapter

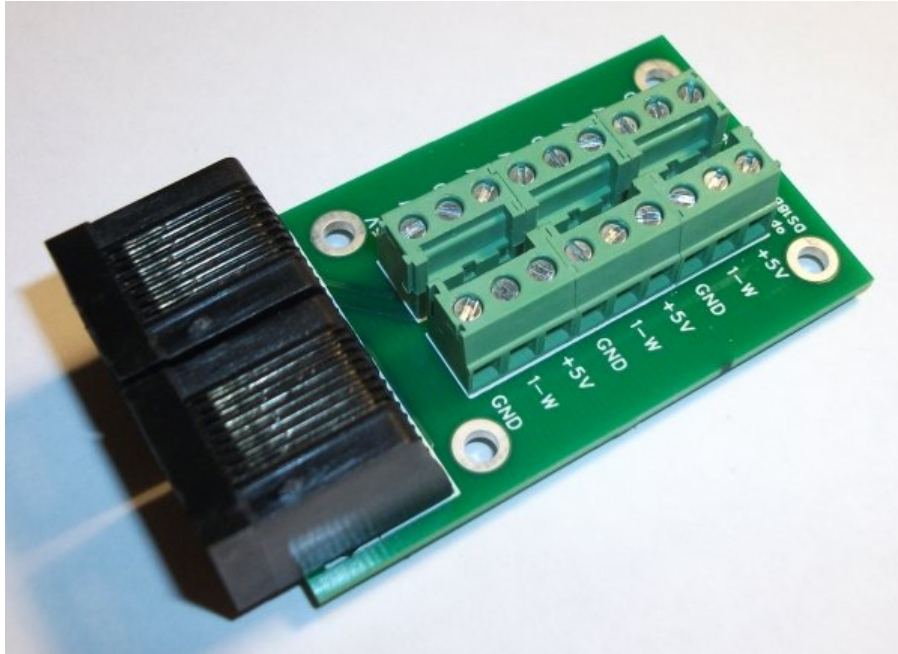
Το βασικό στοιχείο του πρόσθετου είναι το ολοκληρωμένο DS2482-800 που αποτελεί τον μετατροπέα.[8]



Εικόνα 2.11. DS2482-800 8 Channel I2C to 1-Wire Master IC

IV. SWE2b Sensor Connection Module

Αποτελείται δύο υποδοχές RJ45 και 6 κλέμες σε παράλληλη σύνδεση που επιτρέπουν τη σύνδεση των παραπάνω αισθητήρων SWE0a σε 1-Wire δίκτυο μέσω του RPI3 I2C to 1-Wire Host Adapter.[10]



Εικόνα 2.12. SWE2b Sensor Connection Module

V. Εικονικό μηχανήμα που φιλοξενεί τον server

Το εικονικό μηχανήμα έχει στηθεί με VMware και τα τεχνικά του χαρακτηριστικά είναι τα παρακάτω:

- Επεξεργαστής Intel(R) Xeon(R) CPU E5504 @ 2.00GHz
- Μνήμη RAM 1 GB
- Σκληρός δίσκος 64 GB

3. Το λογισμικό

I. Λειτουργικό σύστημα

Υπάρχουν πολλά διαθέσιμα λειτουργικά συστήματα για το Raspberry Pi όπως είναι το Archlinux ARM, το Pidora (Fedora Remix), το Raspbmc και το OPENELEC, το RISC OS καθώς και το Raspbian που αποτελεί και την προτεινόμενη έκδοση για εφαρμογές γενικής χρήσης. Το Raspberry pi δεν μπορεί να έχει τα Windows σαν λειτουργικό σύστημα.

Όλα τα παραπάνω είναι παραλλαγές των γνωστών λειτουργικών που τρέχουν και σε υπολογιστές που έχουν βελτιστοποιηθεί και προσαρμοστεί για τις ανάγκες του Raspberry Pi.

Το Pidora είναι μια διανομή Linux για το Raspberry Pi. Περιέχει τα πακέτα λογισμικού από το λειτουργικό Fedora (συγκεκριμένα, το Fedora ARM δευτεροβάθμιας έργο αρχιτεκτονικής) που έχουν μεταγλωττιστεί ειδικά για την αρχιτεκτονική ARMv6 που χρησιμοποιείται στο Raspberry Pi, πακέτα τα οποία έχουν γραφτεί ειδικά ή έχουν τροποποιηθεί για το Raspberry Pi, και το λογισμικό που παρέχεται από την Raspberry Pi Foundation για την πρόσβαση στη συσκευή.

Το Raspbmc είναι μια διανομή Linux βασισμένη στο Debian που φέρνει το XBMC στο Raspberry Pi. Το Raspberry Pi έχει αρκετή δύναμη για να χειριστεί την αναπαραγωγή πολυμέσων, καθιστώντας το ιδανικό συστατικό για την δημιουργία ενός HTPC (Home Theater PC), που μπορούσε να αποτελείται από πολύ πιο δαπανηρές πλατφόρμες που θα παρήγαγαν το ίδιο αποτέλεσμα.

Το Raspbian είναι ένα ελεύθερο λειτουργικό σύστημα βασισμένο στο Debian και βελτιστοποιημένο για το υλικό του Raspberry Pi. Αποτελεί το ενδεδειγμένο λειτουργικό όταν το Raspberry Pi πρόκειται να χρησιμοποιηθεί για γενική χρήση. Έχει περίπου 35.000 πακέτα προ-μεταγλωττισμένο λογισμικό σε κατανοητή μορφή για εύκολη εγκατάσταση.

Η προτεινόμενη εγκατάσταση για αρχαίους από την Raspberry Pi Foundation είναι το NOOBS γίνεται με απλή αντιγραφή στην κάρτα SD και περιέχει όλα τα παραπάνω λειτουργικά. Στο πρώτο άνοιγμα του Raspberry Pi αποφασίζει ο χρήστης ποιο από τα λειτουργικά θέλει να εγκαταστήσει και τα υπόλοιπα διαγράφονται αυτόματα από την κάρτα.[4]

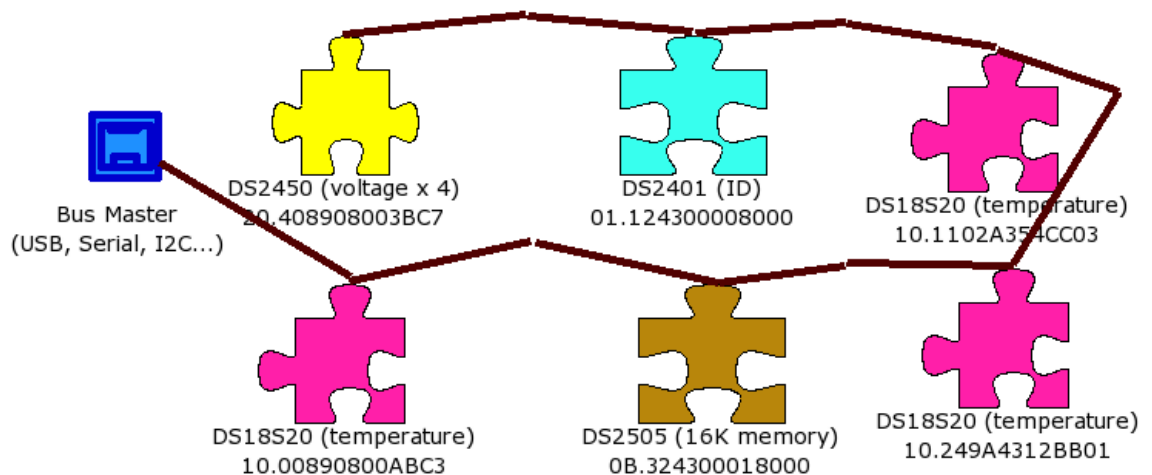
II. Λογισμικό

OWFS

Το OWFS (1-wire Filesystem) είναι μία σουίτα προγραμμάτων που κάνουν τον δίαυλο του 1-wire και τις συσκευές που είναι συνδεδεμένες σε αυτόν εύκολα προσβάσιμες. Το OWFS δημιουργεί ένα εικονικό σύστημα αρχείων που το όνομα του κάθε φακέλου μέσα στον κατάλογο είναι το ID της συσκευής που είναι συνδεδεμένη στο 1-wire. Για παράδειγμα, μπορούμε να δούμε ένα directory της μορφής 10.001122334/temperature που σημαίνει ότι βλέπουμε από τη συσκευή 1-wire με ID 10.001122334 την παράμετρο της θερμοκρασίας.

Για τη σύνδεση των συσκευών (slaves) χρειάζεται και ένας host (master bus) όπως περιγράφηκε και στο Υλικό.

- Το bus master διαβάζει και θέτει την τάση της γραμμής
- Το bus master ξεκινά κάθε επικοινωνία
- Το master bus συνδέεται με τον υπολογιστή μέσω
 - USB
 - σειριακή θύρα
 - δίκτυο
 - απευθείας
 - παράλληλη θύρα
 - I2C



Εικόνα 3.1. Παράδειγμα σύνδεσης συσκευών Slave και Master

Οι βασικές εντολές του OWFS για τη γραμμή εντολών είναι:

- **owdir** - Επιστρέφει μία λίστα με όλες τις συσκευές που είναι συνδεδεμένες στο δίκτυο 1-wire.
- **owread** - Επιστρέφει την τιμή για μία ιδιότητα μία; συσκευής που είναι συνδεδεμένη στο δίκτυο 1-wire.

- **owwrite** - Επιφέρει αλλαγή σε μία ιδιότητα μια; συσκευής που είναι συνδεδεμένη στο δίκτυο 1-wire.
- **owget** - Αρχικά εκτελεί την εντολή `owdir` σαν το όρισμα που δόθηκε να ήταν `directory` και σε περίπτωση που αποτύχει εκτελείται η εντολή `owread` με το όρισμα που δόθηκε σαν ιδιότητα.[5]

MQTT

Το MQTT (Message Queue Telemetry Transport) είναι ένα πρωτόκολλο ανταλλαγής μηνυμάτων πάνω από TCP/IP. Χρησιμοποιείται κυρίως για απομακρυσμένες συνδέσεις όπου το εύρος ζώνης του δικτύου είναι περιορισμένο. Η προεπιλεγμένη θύρα είναι η 1883. Το πρωτόκολλο αυτό στηρίζεται στο `publish/subscribe`. Μία συσκευή κάνει `publish` τα δεδομένα σε κάποιο `topic` και όσες συσκευές θέλουν να λαμβάνουν αυτά τα δεδομένα κάνουν `subscribe` στο συγκεκριμένο `topic`. Για να γίνει αυτό χρειάζεται ένας `broker`, όπως είναι το `RSMB`, `WebSphere MQ`, `HiveMQ`, `Apache Apollo`, `Apache ActiveMQ`, `RabbitMQ`, `MQTT.js`, `mosca`, `2lemetry`, `GnatMQ`, ενώ γνωστότερο είναι το `MOSQUITTO`, που είναι υπεύθυνο για την μεταφορά αυτών των μηνυμάτων στις εγγεγραμμένες συσκευές. Η γνωστότερη εφαρμογή που χρησιμοποιεί το MQTT είναι το `Facebook Messenger`[14]

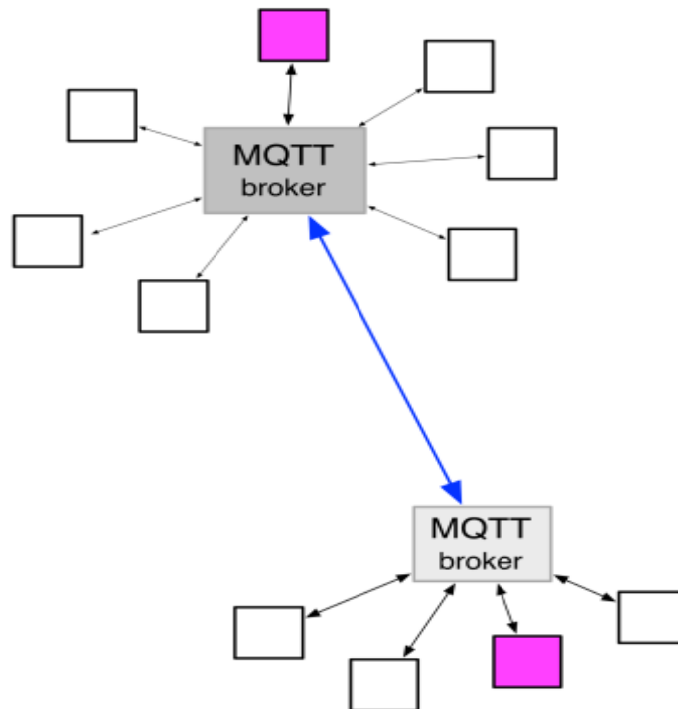
Το MQTT ορίζει τρία επίπεδα του `Quality of Service (QoS)`. Το `QoS` καθορίζει πόσο πολύ ο `broker / client` θα προσπαθήσει να εξασφαλίσει ότι το μήνυμα έχει ληφθεί. Τα μηνύματα μπορούν να αποσταλούν σε οποιοδήποτε επίπεδο `QoS`, και οι πελάτες μπορούν να προσπαθήσουν να εγγραφούν σε `topic` σε κάθε επίπεδο `QoS`. Αυτό σημαίνει ότι ο πελάτης επιλέγει το μέγιστο `QoS` που θα λάβει. Για παράδειγμα, εάν ένα μήνυμα που δημοσιεύεται στο `QoS 2` και ένας πελάτης είναι συνδρομητής με `QoS 0`, το μήνυμα θα πρέπει να παραδοθεί σε αυτό τον πελάτη με `QoS 0`. Αν ένας δεύτερος πελάτης έχει επίσης εγγραφή στο ίδιο θέμα, αλλά και με `QoS 2` θα λάβει το ίδιο μήνυμα αλλά με `QoS 2`. Για ένα δεύτερο παράδειγμα, αν ένας πελάτης είναι συνδρομητής με `QoS 2` και ένα μήνυμα δημοσιεύτηκε στις `QoS 0`, ο πελάτης θα λάβει για `QoS 0`.

Τα υψηλότερα επίπεδα `QoS` είναι πιο αξιόπιστα, αλλά έχουν μεγαλύτερη καθυστέρηση και χρειάζονται μεγαλύτερο εύρος ζώνης.

- 0: Ο `broker / client` θα παραδώσει το μήνυμα μία φορά, χωρίς καμία επιβεβαίωση.
- 1: Ο `broker / client` θα παραδώσει το μήνυμα, τουλάχιστον μία φορά, με την επιβεβαίωση που απαιτείται.
- 2: Ο `broker / client` θα παραδώσει το μήνυμα ακριβώς μια φορά, χρησιμοποιώντας μια χειραψία τεσσάρων σταδίων.

Το `subscribe` μιας συσκευής μπορεί να είναι διαρκές, δηλαδή να μην χάνονται τα μηνύματα που γίνονται `publish` όταν η συσκευή δεν είναι συνδεδεμένη στο `topic`. Αυτό

γίνεται με την επιλογή `cleansession = False` και με τον ορισμό ενός `id` για την συγκεκριμένη `session` και με `QoS` μεγαλύτερο του 0. Έτσι ο `broker` "θυμάται" μέχρι ποιά μήνυμα έχει σταλεί στο συγκεκριμένο `id` και αποστέλλει από το ακριβώς επόμενο. Επίσης, τα μηνύματα αυτά αποθηκεύονται σε μία ουρά όπου η προεπιλεγμένη χωρητικότητα της είναι 100 μηνύματα και μπορεί να αλλάξει προσθέτοντας το `max_queued_messages` με μία παράμετρο.[13]



Εικόνα 3.2. Παράδειγμα ανταλλαγής μηνυμάτων μέσω broker

Υπάρχει πρόσθετο για την γλώσσα `python` όπου γίνεται ευκολότερη η σύνδεση, το `publish` και το `subscribe` των συσκευών στα επιθυμητά `topic`. [11] Για την χρήση του πρόσθετου φτιάχνουμε έναν `Client` με την εντολή :

```
client = mosquitto.Mosquitto ( client_id="", clean_session=True, userdata=None, protocol=MQTTv311)
```

και στη συνέχεια κάνουμε `subscribe` στα επιθυμητά `topic` με την εντολή

```
client.subscribe(topic, qos=0)
```

και `publish` με την εντολή

```
publish(topic, payload=None, qos=0, retain=False)
```

RRDTool

Το RRDTool (Round-Robin Database Tool) είναι ένα εργαλείο για τη δημιουργία βάσεων δεδομένων που έχουν σταθερό μέγεθος. Χρησιμοποιείται συνήθως για την αποθήκευση στοιχείων όπως θερμοκρασία, υγρασία, κατανάλωση κ.α. σε σχέση με το χρόνο.[16]

Πιο χρήσιμη ιδιότητα του RRDTool είναι το RRA (Round-Robin Archive) καθώς εκεί αποθηκεύονται ομαδοποιημένα όλα τα δεδομένα ανάλογα με τις ρυθμίσεις που έχουμε επιλέξει. Σε μία βάση δεδομένων μπορούν να χρησιμοποιηθούν περισσότερα του ενός RRA και κάθε ένα να τροφοδοτεί το επόμενο όταν οι μετρήσεις ξεπερνούν σε αριθμό το κατώφλι που έχουμε ορίσει. Για παράδειγμα, με τον κώδικα `rrdtool create subdata.rrd -s 10 DS:ds0:GAUGE:300:0:U RRA:AVERAGE:0.5:30:3600 RRA:AVERAGE:0.5:90:1200 RRA:AVERAGE:0.5:360:1200 RRA:MAX:0.5:360:1200 RRA:AVERAGE:0.5:8640:600 RRA:MAX:0.5:8640:600` δημιουργούμε μία βάση δεδομένων που συλλέγει δεδομένα κάθε 10 δευτερόλεπτα και αποθηκεύει μέσους όρους για ανά 5 λεπτά, 15 λεπτά, 1 ώρα και 1 ημέρα, καθώς και τα μέγιστο ανά 1 ώρα και 1 ημέρα.

Όταν δημιουργήσουμε μία βάση δεδομένων μετά δεν γίνεται να αλλάξουμε τις παραμέτρους που έχουμε ορίσει. Μπορούμε όμως να τις μεταφέρουμε σε καινούργιο αρχείο με την εντολή `rrdtool resize filename rra-numGROW|SHRINK rows`. Το νέο `.rrd` αρχείο, με τα τροποποιημένα RRAs, αποθηκεύεται στο `resize.rrd` στον τρέχοντα κατάλογο. Το αρχικό `.rrd` αρχείο δεν μεταβάλλεται.

Με την εντολή `rrdtool fetch filename CF [--resolution|-r resolution] [--start|-s start] [--end|-e end]` μπορούμε να εμφανίσουμε τα στοιχεία της βάσης δεδομένων που θέλουμε θέτοντας ως παραμέτρους από τότε μέχρι τότε θέλουμε και με τι ανάλυση.

Ακόμη, μπορούμε να χρησιμοποιήσουμε την εντολή `rrdtool {update | updatev} filename [--template|-t ds-name[:ds-name]...] [--daemon address] [--]N|timestamp:value[:value...] at-timestamp@value[:value...] [timestamp:value[:value...] ...]` για να προσθέσουμε στοιχεία στη βάση δεδομένων που θέλουμε. Πρώτη στήλη είναι το timestamp και χρησιμοποιώντας το γράμμα N για αυτή τη στήλη λαμβάνεται αυτόματα το timestamp της στιγμής που προστίθεται το στοιχείο. [15]

Τέλος, με την εντολή `rrdtool graph|graphv filename [option ...] [data definition ...] [data calculation ...] [variable definition ...] [graph element ...] [print element...]` μπορούμε να εμφανίσουμε τα στοιχεία της βάσης δεδομένων σε γραφήματα.[17]

HighCharts

Το Highcharts είναι έτοιμες βιβλιοθήκες με κώδικα γραμμένο σε javascript για την παρουσίαση δεδομένων σε μορφή γραφικών παραστάσεων σε μία ιστοσελίδα. Υπάρχουν πολλές διαφορετικές επιλογές γραφημάτων ανάλογα με το τι θέλουμε να παρουσιάσουμε. Εύκολα με μερικές αλλαγές στον υπάρχοντα κώδικα μπορούμε να προσαρμόσουμε τις έτοιμες βιβλιοθήκες για την παρουσίαση των δικών μας δεδομένων.[18]

Το Highcharts βασίζεται αποκλειστικά σε native browser technologies (τα προγράμματα γίνονται μεταγλωττίζονται εσωτερικά στον browser) και δεν απαιτεί plugins από την πλευρά του client όπως Flash ή Java. Επιπλέον, δεν χρειάζεται να εγκαταστήσουμε τίποτα στον server μας. Δεν χρησιμοποιεί καθόλου PHP ή ASP.NET. Το Highcharts χρειάζεται μόνο δύο αρχεία JS για να τρέξει: Τον πυρήνα highcharts.js και είτε το jQuery, MooTools ή Prototype framework. Ένα από αυτά τα πλαίσια είναι πιθανότατα ήδη σε χρήση στην ιστοσελίδα μας.

Το Highcharts υποστηρίζει διαγράμματα με γραμμές, περιοχές, στήλες, μπάρες, πίτες, διασποράς, και γραφήματα πολικών τύπων. Πολλά από αυτά μπορούν να συνδυαστούν σε ένα διάγραμμα.

Μπορούμε ακόμα να συγκρίνουμε τις μεταβλητές που δεν είναι στην ίδια κλίμακα - για παράδειγμα τη θερμοκρασία έναντι των βροχοπτώσεων και την πίεση του αέρα. Το Highcharts επιτρέπει να ορίσουμε έναν άξονα y για κάθε σειρά (ή έναν άξονα x), αν θέλετε να συγκρίνετε σύνολα δεδομένων των διαφορετικών κατηγοριών. Κάθε άξονας μπορεί να τοποθετηθεί δεξιά ή αριστερά, πάνω ή κάτω στο διάγραμμα. Όλες οι επιλογές μπορούν να ρυθμιστούν ξεχωριστά, συμπεριλαμβανομένης της αναστροφής, το styling και τη θέση.

Το 75% του συνόλου των διαγραμμάτων με ένα άξονα X και Y έχουν ένα άξονα X με ημερομηνία-ώρα. Ως εκ τούτου το Highchart είναι πολύ έξυπνος για τις αξίες του χρόνου. Με χιλιοστά του δευτερολέπτου, το Highcharts καθορίζει πού θα τοποθετηθούν τα διαστήματα, έτσι ώστε να σηματοδοτούν πάντα την αρχή του μήνα ή την εβδομάδα, τα μεσάνυχτα και το μεσημέρι, την πλήρη ώρα κ.λπ.

Ακόμα, μπορούμε να ενεργοποιήσουμε την εξαγωγή των διαγραμμάτων, έτσι ώστε οι χρήστες να μπορούν να εξάγουν τα γραφήματα σε PNG, JPG, PDF ή μορφή SVG με το πάτημα ενός κουμπιού, ή να εκτυπώνουν το γράφημα απευθείας από την ιστοσελίδα. [19]

Τέλος, στα διαγράμματα του Highcharts μπορεί να γίνει ζουμ έτσι ώστε να εξασφαλίζεται μεγαλύτερη ακρίβεια στην ανάγνωση τους. Στην επιλογή αυτή μπορούμε να καθορίσουμε μέχρι πόσα δεδομένα θα εμφανίζονται στην οθόνη. [20]

4. Στήσιμο μηχανημάτων

I. Raspberry Pi

Το Raspberry Pi χρησιμοποιήθηκε για την μέτρηση των θερμοκρασιών μέσω των αισθητήρων 1-wire, για τον έλεγχο των GPIO, για την παρουσίαση των αποτελεσμάτων καθώς και της διαχείρισης του σε ιστοσελίδα που σερβίρει το ίδιο. Παρακάτω παρουσιάζεται αναλυτικά κάθε βήμα για την εγκατάσταση και την ρύθμιση όλων των απαραίτητων παραμέτρων για να έχουμε το τελικό αποτέλεσμα.

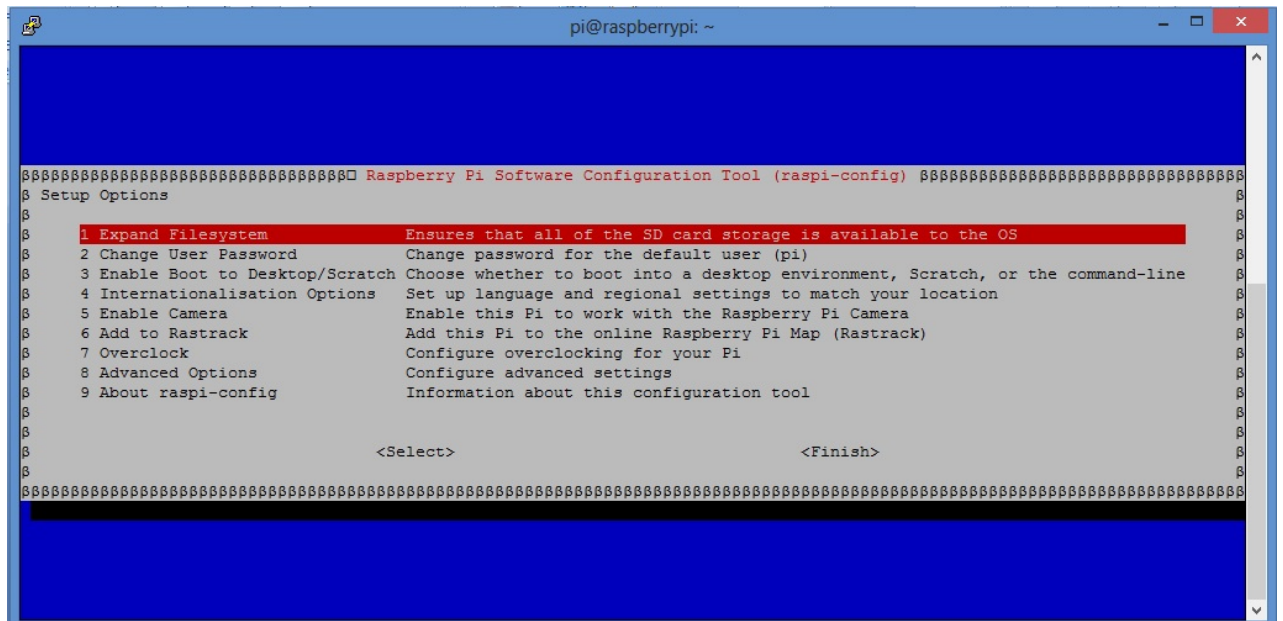
Το λειτουργικό που επιλέχθηκε είναι το can καθώς είναι πολύ κοντά στην έκδοση Debian για σταθερούς υπολογιστές και έτσι μπορεί χρησιμοποιηθεί εύκολα και χωρίς περαιτέρω μελέτη. Για να εγκαταστήσουμε το λειτουργικό Raspbian στο Raspberry Pi θα πρέπει αρχικά να αντιγράψουμε το αρχείο wheezy-raspbian.img, που είναι διαθέσιμο στην επίσημη ιστοσελίδα του raspberry pi, στην κάρτα SD που διαθέτουμε.

Βάζουμε την κάρτα σε έναν υπολογιστή που στην συγκεκριμένη περίπτωση τρέχει Windows 8 και με το SDFormatter κάνουμε format στην κάρτα έχοντας την επιλογή size adjustment στο ON έτσι να γίνει προσαρμογή στο μέγεθος της κάρτας. Έπειτα, με το Win32DiskImager αντιγράφουμε το αρχείο image στην κάρτα. Οι δύο παραπάνω διαδικασίες μπορούν να γίνουν και με άλλα αντίστοιχα προγράμματα που χρησιμοποιούνται για την διαμόρφωση καρτών SD.

Στη συνέχεια, βάζουμε την κάρτα SD στο Raspberry pi και αφού το έχουμε συνδέσει στο τοπικό μας δίκτυο το ανάβουμε. Οι αρχικές ρυθμίσεις του Raspberry Pi είναι η απόκτηση IP να γίνεται μέσω dhcp client και η δυνατότητα σύνδεσης ssh να είναι ενεργοποιημένη. Έτσι, βρίσκουμε την IP address του Raspberry pi, από τις συνδεδεμένες συσκευές στο Router, που από εδώ και στο εξής θα την αναφέρουμε ως <Pi IP>.

Συνδεόμαστε στην <Pi IP> με ssh με ένα πρόγραμμα όπως το PuTTY. Αν όλα έχουν πάει καλά η σύνδεση είναι επιτυχής και εμφανίζεται στον υπολογιστή μας ένα παράθυρο με την λέξη "login as:". Το προκαθορισμένο username και password είναι pi και raspberry αντίστοιχα.

Η πρώτη ρύθμιση που θα πρέπει να κάνουμε στο Raspberry Pi είναι η επέκταση της μνήμης (expand filesystem) έτσι ώστε να είναι ορατή από το λειτουργικό σύστημα όλο το μέγεθος της κάρτας SD. Για να εισέλθουμε στις ρυθμίσεις του Raspberry Pi εκτελούμε την εντολή **sudo raspi-config** (Επισημαίνεται ότι για τον χρήστη pi δεν χρειάζεται κωδικός για να εκτελεστεί μία εντολή με sudo). Με την εκτέλεση αυτής της εντολής βλέπουμε τις παρακάτω επιλογές:



Εικόνα 4.1. Raspberry Pi configuration

Επιλέγουμε το 1 Expand Filesystem και πατάμε Enter. Μετά το τέλος της διαδικασίας αυτής μπορούμε να έχουμε πρόσβαση σε όλο το μέγεθος της κάρτας SD. Εκτελώντας την εντολή **df -h** μπορούμε να δούμε το διαθέσιμο τον χρησιμοποιημένο καθώς και τον συνολικό χώρο στην κάρτα.

Στη συνέχεια εκτελούμε την εντολή **sudo apt-get update && sudo apt-get upgrade** (απαντάμε με YES στο ερώτημα αν θέλουμε να συνεχίσουμε) έτσι ώστε να γίνουν όλες οι τελευταίες αναβαθμίσεις που είναι διαθέσιμες για το Raspberry pi καθώς επίσης και την εντολή **sudo apt-get clean** για να διαγραφούν τα προσωρινά αρχεία της εγκατάστασης.

Στο δεύτερο βήμα, θα εγκαταστήσουμε τους 1-wire αισθητήρες. Ανοίγουμε και πάλι τις ρυθμίσεις του Raspberry Pi επιλέγουμε Advanced Options και εκεί επιλέγουμε το A6 I2C και επιλέγουμε Yes για να ενεργοποιήσουμε την αυτόματη φόρτωση των drivers για το I2C. Αφού έχουμε προσαρμόσει σωστά τα εξαρτήματα που έχουν περιγραφεί παραπάνω θα πρέπει να εγκαταστήσουμε το owfs. Για να το πετύχουμε αρχικά ενεργοποιούμε την μονάδα (module) **i2c-bcm2708** με τις εντολές **sudo modprobe i2c-bcm2708** και **sudo modprobe i2c-dev**. Για να φορτώνονται κάθε φορά που ξεκινάει το Raspberry pi αυτά τα module προσθέτουμε στο τέλος του αρχείου **/etc/modules** τη λέξη **"i2c-bcm2708"**. Επίσης, εκτελούμε την εντολή **sudo apt-get install i2c-tools** για την εγκατάσταση των εργαλείων για το I2C. Αν όλα έχουν πάει καλά με την εκτέλεση της εντολής **sudo i2cdetect -y 1** θα έχουμε ένα αποτέλεσμα σαν το παρακάτω όπου φαίνονται οι θύρες που είναι καταλυμένες από το I2C.

```
pi@raspberrypi ~ $ sudo i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  1c  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  68  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
pi@raspberrypi ~ $
```

Εικόνα 4.2. Αποτέλεσμα εντολής `sudo i2cdetect -y 1`

Στη συνέχεια, για να εγκαταστήσουμε το `owfs` εκτελούμε την εντολή `sudo apt-get install owfs ow-shell` και διαμορφώνουμε το αρχείο `/etc/owfs.conf` ως εξής: Αλλάζουμε την γραμμή "server: FAKE = DS18S20,DS2405" σε "server: device=/dev/i2c-1". Κάνουμε επανεκκίνηση στον `owfs` server με την εντολή `sudo service owserver restart` μπορούμε να εκτελέσουμε οποιαδήποτε εντολή θέλουμε από αυτές που έχουν αναφερθεί παραπάνω στο κομμάτι του `owfs`.^[6]

Στο επόμενο βήμα θα εγκαταστήσουμε τον Apache server καθώς και την γλώσσα `php5`. Αρχικά εκτελούμε την εντολή `sudo apt-get install apache2 -y` και στη συνέχεια την εντολή `sudo apt-get install php5 libapache2-mod-php5 -y`. Για να δοκιμάσουμε αν έχει εγκατασταθεί ο Apache server μπορούμε να χτυπήσουμε την `<Pi IP>` από έναν browser ενός υπολογιστή στο δίκτυο και να λάβουμε την προκαθορισμένη σελίδα του apache που φαίνεται παρακάτω.



It works!

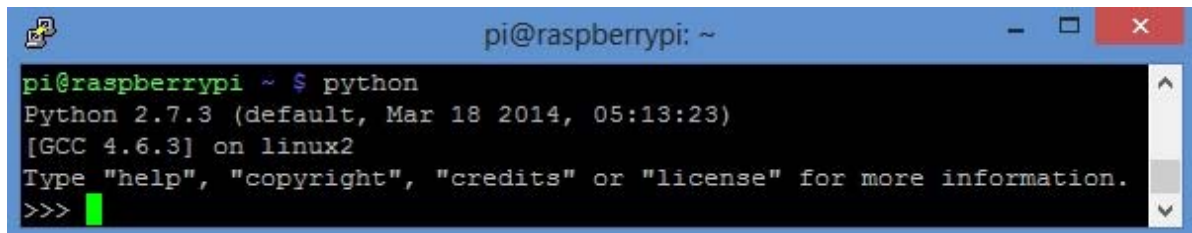
This is the default web page for this server.

The web server software is running but no content has been added, yet.

Εικόνα 4.3. Apache web server

Ο προκαθορισμένος χρήστης για τον apache server είναι ο `www-data`, έτσι θα πρέπει να τον προσθέσουμε στους `sudoers` για να μπορεί να εκτελέσει τις εντολές που θα χρειαστούμε στη συνέχεια. Για να το επιτύχουμε αυτό εισάγουμε στο αρχείο `/etc/sudoers` την γραμμή `www-data ALL=(ALL) NOPASSWD:ALL` (επισημαίνεται ότι με αυτόν τον τρόπο ο χρήστης `www-data` μπορεί να εκτελέσει όλες τις εντολές με `sudo` χωρίς κάποιον κωδικό). Ακόμα εκτελούμε την εντολή `chown -R www-data /etc/network/interfaces` έτσι ώστε ο χρήστης `www-data` να μπορεί να αλλάξει το αρχείο αυτό που θα μας χρειαστεί για την επιλογή του τρόπου λήψης IP από την ιστοσελίδα.

Στο τέταρτο βήμα θα εγκαταστήσουμε την γλώσσα python καθώς και όλα τα απαραίτητα πρόσθετα για την χρησιμοποίηση της για τα GPIO και για το MQTT. Εκτελούμε τις εντολές **sudo apt-get install python python-dev libjpeg-dev libfreetype6-dev python-setuptools python-pip** , **sudo easy_install -U distribute** για να κάνουμε update στο distribute της python και στη συνέχεια εγκαθιστούμε το πρόσθετο για GPIO με την εντολή **sudo pip install RPi.GPIO**. Εκτελούμε την εντολή python για να δούμε αν η γλώσσα έχει εγκατασταθεί και παίρνουμε το παρακάτω αποτέλεσμα:



```
pi@raspberrypi: ~
pi@raspberrypi ~ $ python
Python 2.7.3 (default, Mar 18 2014, 05:13:23)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Εικόνα 4.4. Έκδοση Python

Στο επόμενο βήμα θα εγκαταστήσουμε έναν ftp server έτσι ώστε να μπορούμε εύκολα να διαχειριζόμαστε τα αρχεία που υπάρχουν στο Raspberry Pi. Εκτελούμε την εντολή **sudo apt-get install vsftpd** και εγκαθιστούμε το vsftpd. Στη συνέχεια κάνουμε τις εξής αλλαγές στο αρχείο **/etc/vsftpd.conf** : Αλλάζουμε την γραμμή **anonymous_enable=YES** σε **anonymous_enable=NO** για να απενεργοποιήσουμε την anonymous πρόσβαση, σβήνουμε το # από τις γραμμές **local_enable=YES write_enable=YES** έτσι ώστε να μην λαμβάνονται σαν σχόλια.

Για την εγκατάσταση του no ip εκτελούμε τις εντολές **mkdir /home/pi/noip** και **cd /home/pi/noip** για την δημιουργία και την αλλαγή του directory. Στη συνέχεια, εκτελούμε τις εντολές **wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz** , **tar vzx noip-duc-linux.tar.gz** , **cd noip-2.1.9-1** , **sudo make** , **sudo make install**. Για να τρέξει το no ip στο Raspberry Pi εκτελούμε την εντολή **sudo /usr/local/bin/noip2**.

Σαν τελευταίο βήμα θα εγκαταστήσουμε το Mosquitto MQTT Broker. Εκτελούμε την εντολή **sudo apt-get install mosquitto mosquitto-clients python-mosquitto** για να εγκαταστήσουμε το Mosquitto MQTT Broker, το Mosquitto Client καθώς και τα πρόσθετα για του mosquitto για την python. Τέλος, εκτελούμε την εντολή **sudo iptables -A INPUT -p tcp -m tcp --dport 1883 -j ACCEPT** έτσι ώστε τα πακέτα tcp που έρχονται στην πόρτα 1883 (προκαθορισμένη πόρτα του MQTT) να μην απορρίπτονται.

II. Virtual Machine

Το Virtual Machine χρησιμοποιήθηκε για την αποθήκευση των δεδομένων σε βάση δεδομένων RRDTool και για την παρουσίαση τους σε μια ιστοσελίδα με τη μορφή γραφημάτων με τη βοήθεια του HighCharts.

Συνδεόμαστε όπως και προηγουμένως με ssh στην IP του Virtual Machine που από εδώ και στο εξής θα αναφέρουμε ως <VM IP>. Εγκαθιστούμε με τον ίδιο τρόπο όπως προηγουμένως το **apache web server**, την **php**, την **python** και το **mosquitto MQTT Broker** και **Client**.

Για την εγκατάσταση του RRDTool εκτελούμε τις εντολές **apt-get install libpango1.0-dev libxml2-dev** για την εγκατάσταση κάποιων απαραίτητων αρχείων, **wget http://oss.oetiker.ch/rrdtool/pub/rrdtool-1.4.7.tar.gz** , **tar -zxvf rrdtool-1.4.7.tar.gz** για την λήψη και την αποσυμπίεση του rrdtool 1.4.7 , **cd rrdtool-1.4.7** , **mkdir /tmp/rrdbuild** , **export BUILD_DIR=/tmp/rrdbuild** , **mkdir /opt/rrdtool-1.4.7** , **export INSTALL_DIR=/opt/rrdtool-1.4.7** , **./configure -prefix=\$INSTALL_DIR** , **make** και **make install** . Αφού έχουμε εγκαταστήσει το RRDTool θα πρέπει να εγκαταστήσουμε και το πρόσθετο της php και της python για το RRDTool έτσι ώστε να είναι ευκολότερο να το χρησιμοποιήσουμε μέσα σε αρχεία κώδικα.

Για να εγκαταστήσουμε το πρόσθετο για το RRDTool για την python εκτελούμε την εντολή **sudo pip install python-rrdtool** . Αφού εγκαταστήσουμε το πρόσθετο κάνουμε **import rrdtool** στο πρόγραμμα που θέλουμε να το χρησιμοποιήσουμε. Για να εγκαταστήσουμε το πρόσθετο για την php εκτελούμε τις εντολές **aptitude install librrd-dev php-pear php5-dev pecl install rrd** , **echo "extension=rrd.so" >> /etc/php5/conf.d/rrd.ini** και **service apache2 restart** και μπορούμε να χρησιμοποιήσουμε τις εντολές του RRDTool για την php.

Τέλος, για να χρησιμοποιήσουμε το HighCharts για την παρουσίαση των διαγραμμάτων εκτελούμε τις εντολές **wget http://code.highcharts.com/zips/Highcharts-4.0.4.zip** , **unzip Highcharts-4.0.4.zip** και **mv ./ Highcharts-4.0.4 /var/www/html** για να μπορούμε να δούμε δημιουργήσουμε τα δικά μας γραφήματα και να τα εμφανίσουμε μέσα από τον **apache web server**.

5. Η web εφαρμογή

I. Raspberry Pi

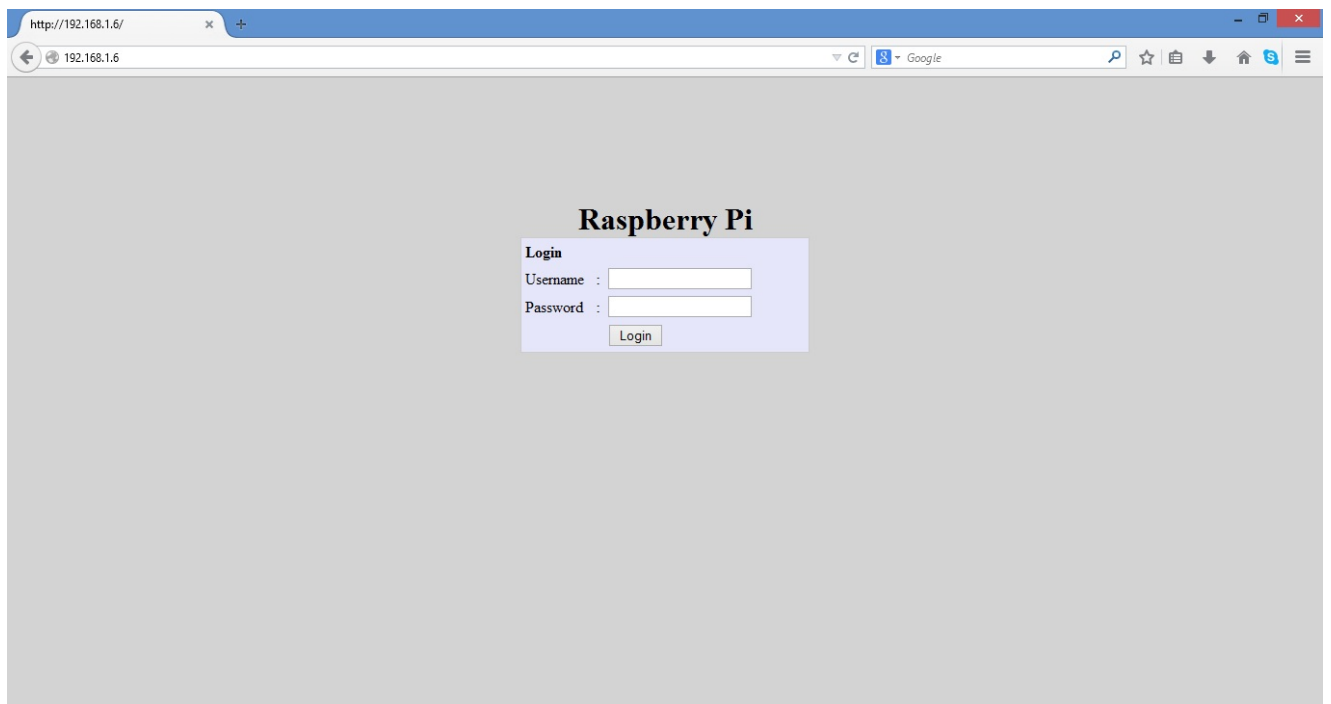
Οι σελίδες στο Raspberry Pi αφορούν τη διαχείρισή του και τη διαχείριση των εξαρτημάτων όπως είναι οι επιλογή των αισθητήρων και των GPIO και ο έλεγχος τους. Αρχικά, για να εισέλθει ο χρήστης σε αυτή τη σελίδα θα πρέπει να περάσει από τη σελίδα επιβεβαίωσης των στοιχείων username και password. Τα προεπιλεγμένα στοιχεία είναι

Username: admin

Password: admin

Ο χρήστης μπορεί να αλλάξει τα στοιχεία αυτά από τη στιγμή που έχει εισέλθει στη σελίδα changelogin.php

index.php



Εικόνα 5.1. Η αρχική σελίδα Index

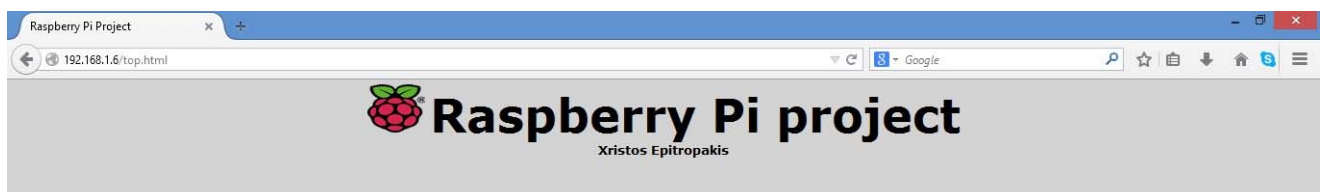
changelogin.php



Εικόνα 5.2. Αλλαγή username και password

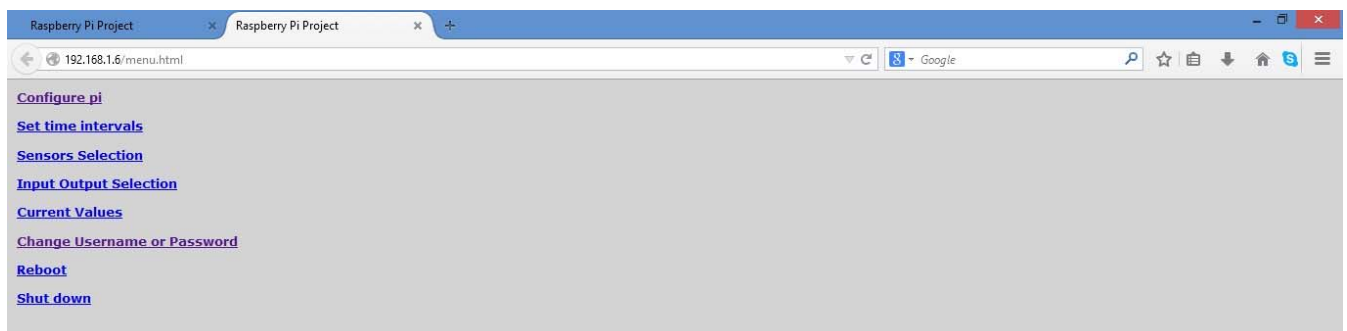
Μόλις γίνει η σωστή εισαγωγή των στοιχείων μεταφερόμαστε στη σελίδα home που περιλαμβάνει 3 frames. Το frame top.html , το frame menu.html και το κεντρικό frame όπου ανοίγουν όλες οι σελίδες.

top.html



Εικόνα 5.3. Σελίδα top.html που αποτελεί το top frame

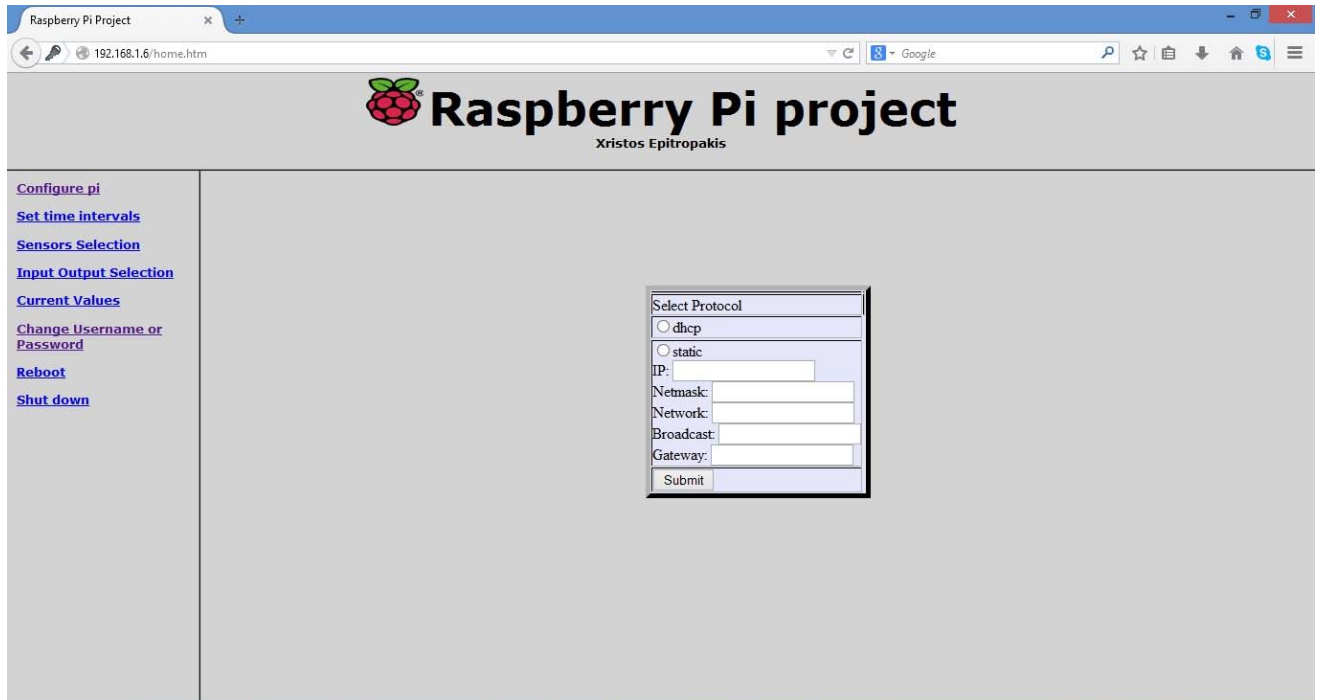
menu.html



Εικόνα 5.4. Σελίδα menu.html που αποτελεί το αριστερό frame

Η πρώτη σελίδα που σερβίρεται στο κεντρικό frame είναι η configure.php.

configure.php



Εικόνα 5.5. Διαχείριση IP του Raspberry Pi

Στη σελίδα αυτή ο χρήστης επιλέγει τον τρόπο με τον οποίο το Raspberry Pi αποκτά την IP. Μπορεί να επιλέξει dhcp ή static. Αν επιλεγεί το dhcp η απόδοση της IP γίνεται από τον dhcp client, ενώ αν επιλεγεί static, ο χρήστης, θα πρέπει να ορίσει την IP, Netmask, Network, Broadcast, Gateway.

Η επόμενη σελίδα είναι η cron.php όπου ο χρήστης επιλέγει κάθε πόσα λεπτά θα λαμβάνεται μέτρηση από τους αισθητήρες και τα GPIO. Το ελάχιστο χρονικό διάστημα είναι 1 λεπτό και το μέγιστο 1 ώρα.

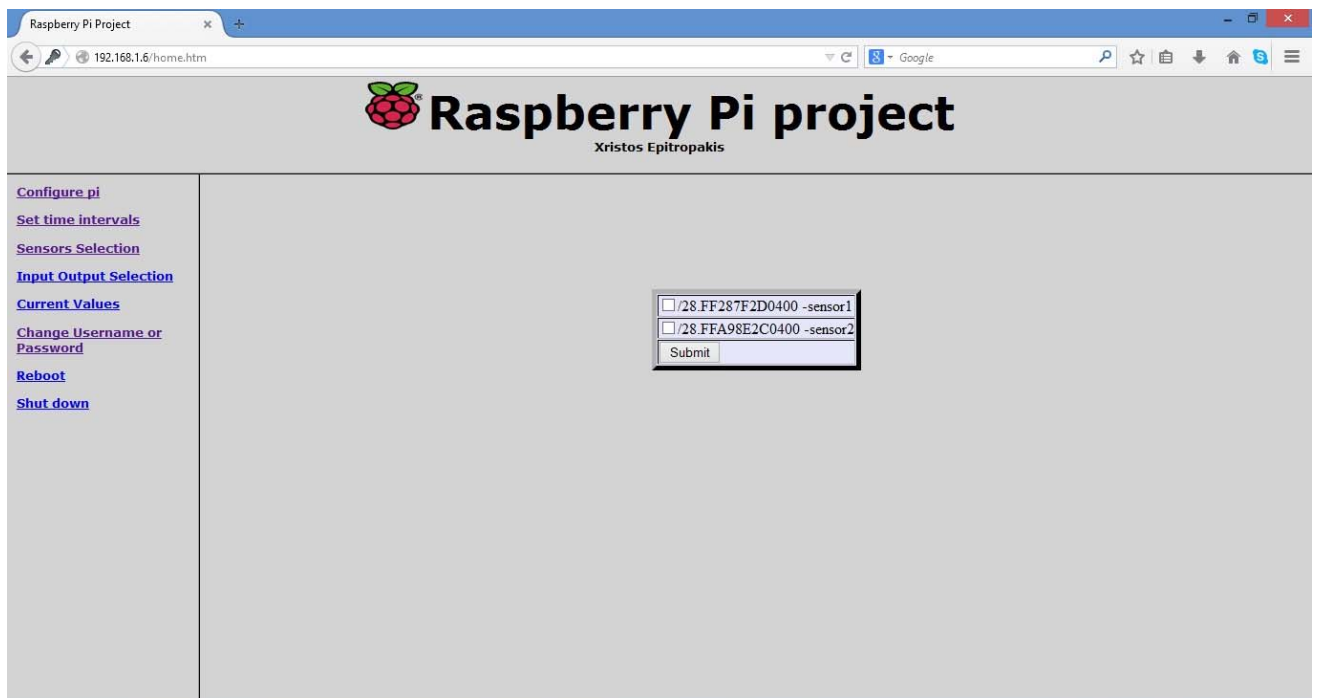
cron.php



Εικόνα 5.6. Επιλογή χρονικών διαστημάτων

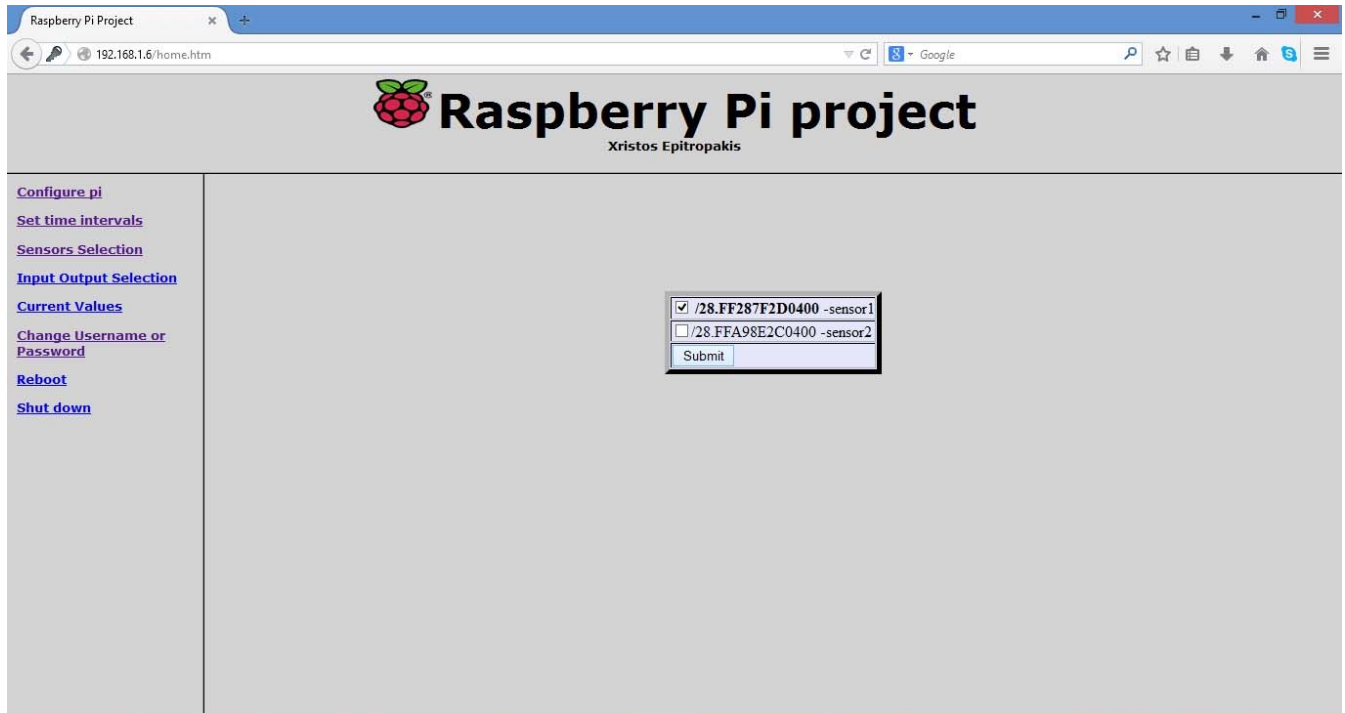
Στην επόμενη σελίδα `sensors.php` ο χρήστης επιλέγει από όλους τους διαθέσιμους αισθητήρες ποιούς θα χρησιμοποιήσει για τις μετρήσεις του.

sensors.php



Εικόνα 5.5. Επιλογή αισθητήρων

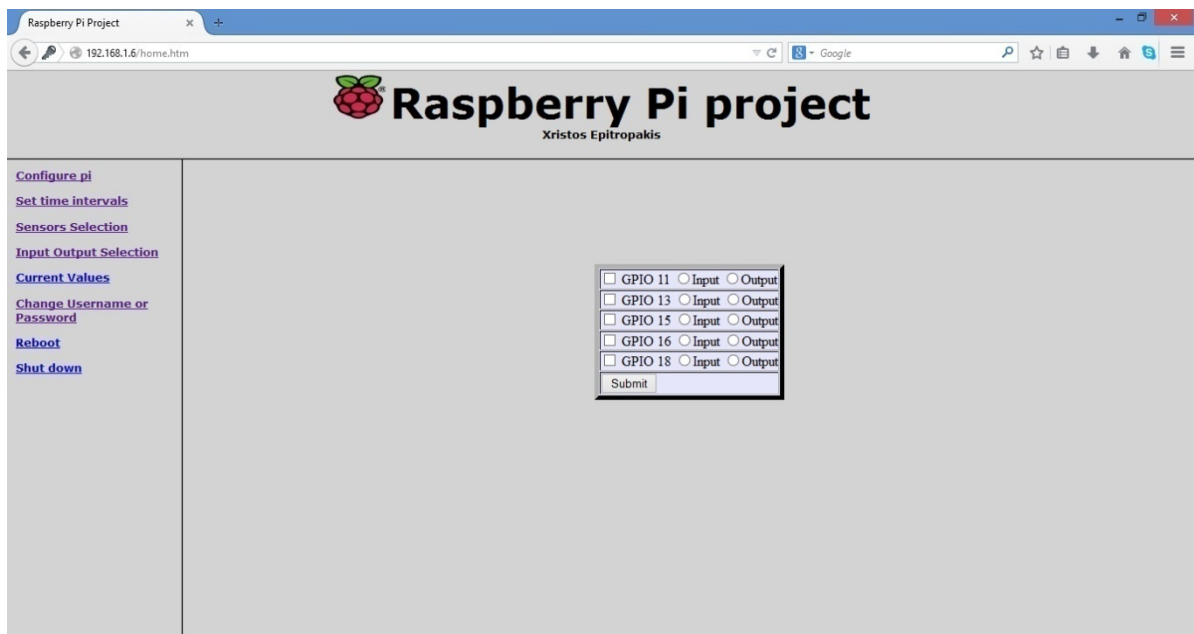
Σε περίπτωση που έχουμε επιλέξει κάποιον αισθητήρα, αυτός εμφανίζεται με **bold** καθώς επίσης είναι προεπιλεγμένος όπως φαίνεται παρακάτω.



Εικόνα 5.6. Επιλεγμένος αισθητήρας

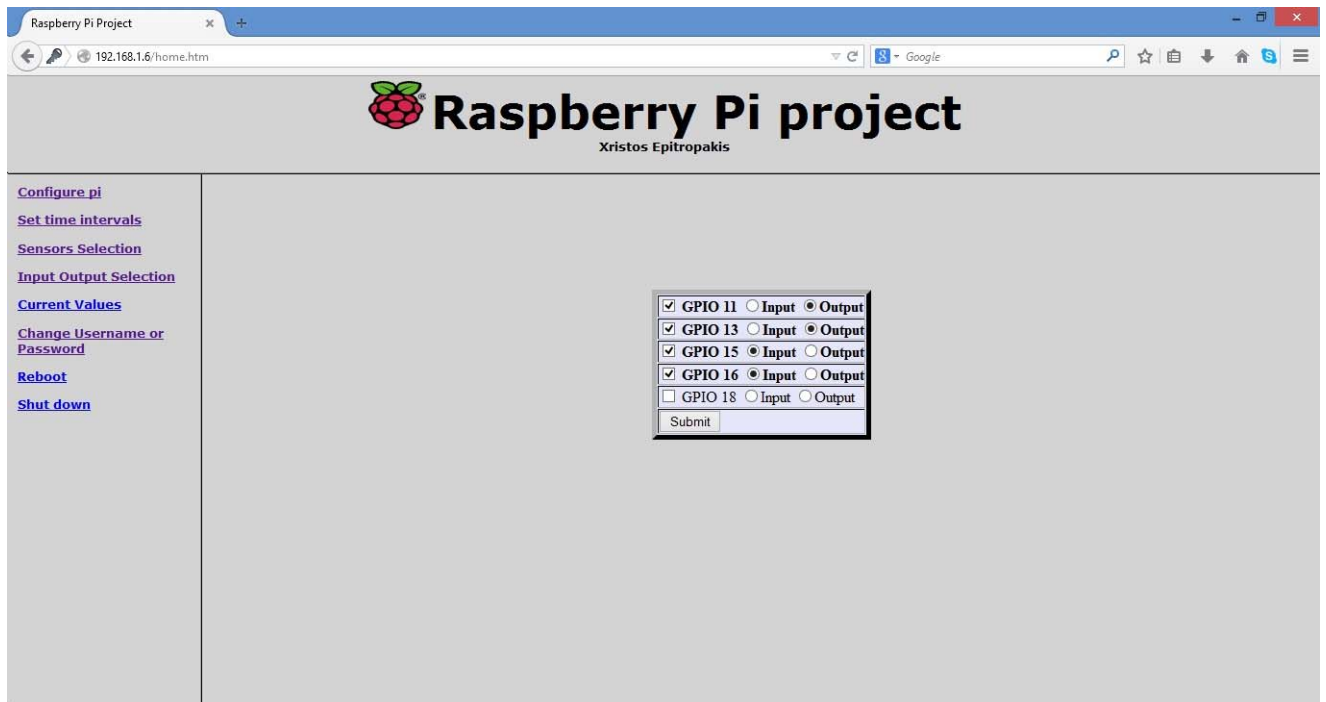
Στη συνέχεια, έχουμε την επιλογή των GPIO αν θα είναι είσοδος ή έξοδος.

gpio.php



Εικόνα 5.7. Επιλογή GPIO

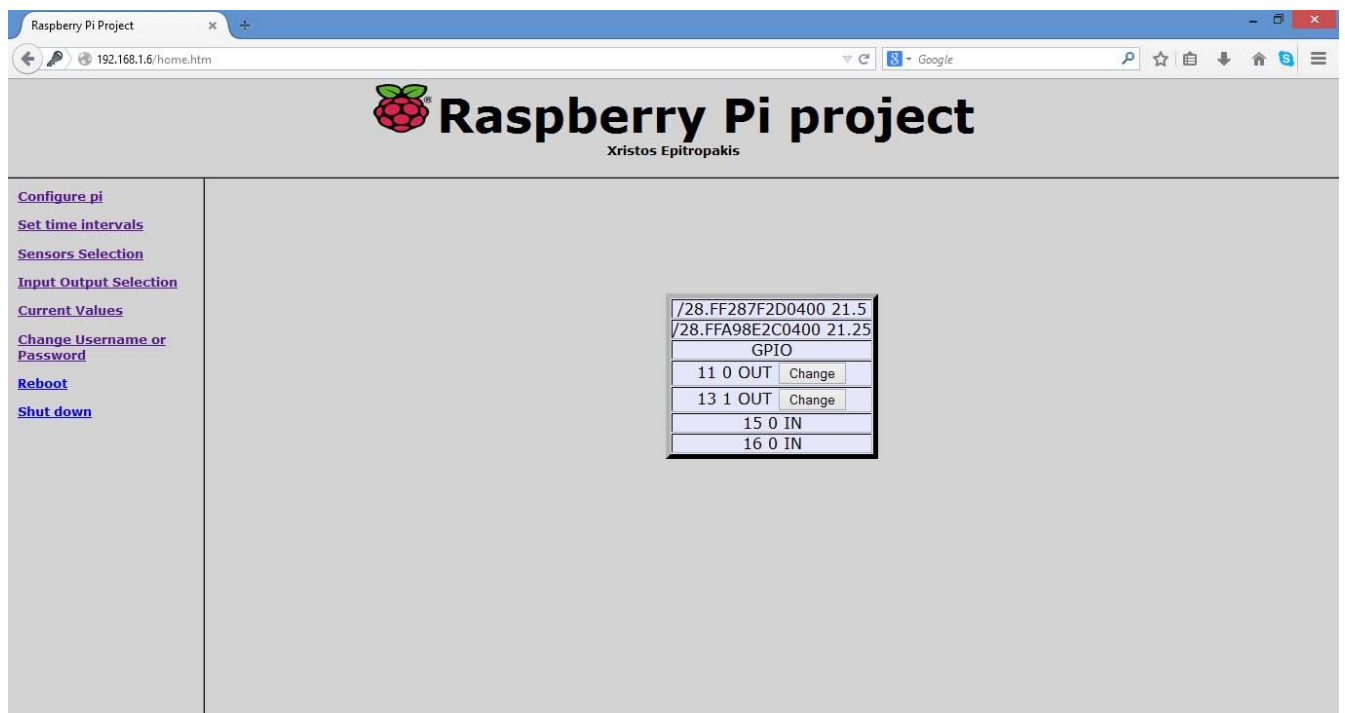
Όπως και προηγουμένως, όταν έχουμε κάνει κάποια επιλογή φαίνεται με **bold** και είναι προεπιλεγμένα τα στοιχεία που έχουμε επιλέξει



Εικόνα 5.8. Προεπιλεγμένα GPIO

Στην επόμενη σελίδα μπορούμε να δούμε τα στοιχεία που έχουμε επιλέξει καθώς και την τελευταία μέτρηση που έχει γίνει. Ακόμη, αν έχουμε επιλέξει για ένα GPIO να είναι έξοδος μπορούμε να αλλάξουμε την τιμή του από ON σε OFF και το αντίστροφο.

results.php



The screenshot shows a web browser window with the title "Raspberry Pi Project" and the URL "192.168.1.6/home.htm". The page features the Raspberry Pi logo and the text "Raspberry Pi project" and "Xristos Epitropakis". On the left side, there is a navigation menu with the following links: [Configure pi](#), [Set time intervals](#), [Sensors Selection](#), [Input Output Selection](#), [Current Values](#), [Change Username or Password](#), [Reboot](#), and [Shut down](#). The main content area displays a table of sensor data and GPIO status:

/28.FF287F2D0400	21.5
/28.FFA98E2C0400	21.25
GPIO	
11 0 OUT	<input type="button" value="Change"/>
13 1 OUT	<input type="button" value="Change"/>
15 0 IN	
16 0 IN	

Εικόνα 5.9. Εμφάνιση τελευταίων μετρήσεων

Τέλος, ο χρήστης μπορεί να επιλέξει να κάνει επανεκκίνηση ή και να απενεργοποίηση το Raspberry Pi πατώντας στο Reboot και Shutdown στο menu.

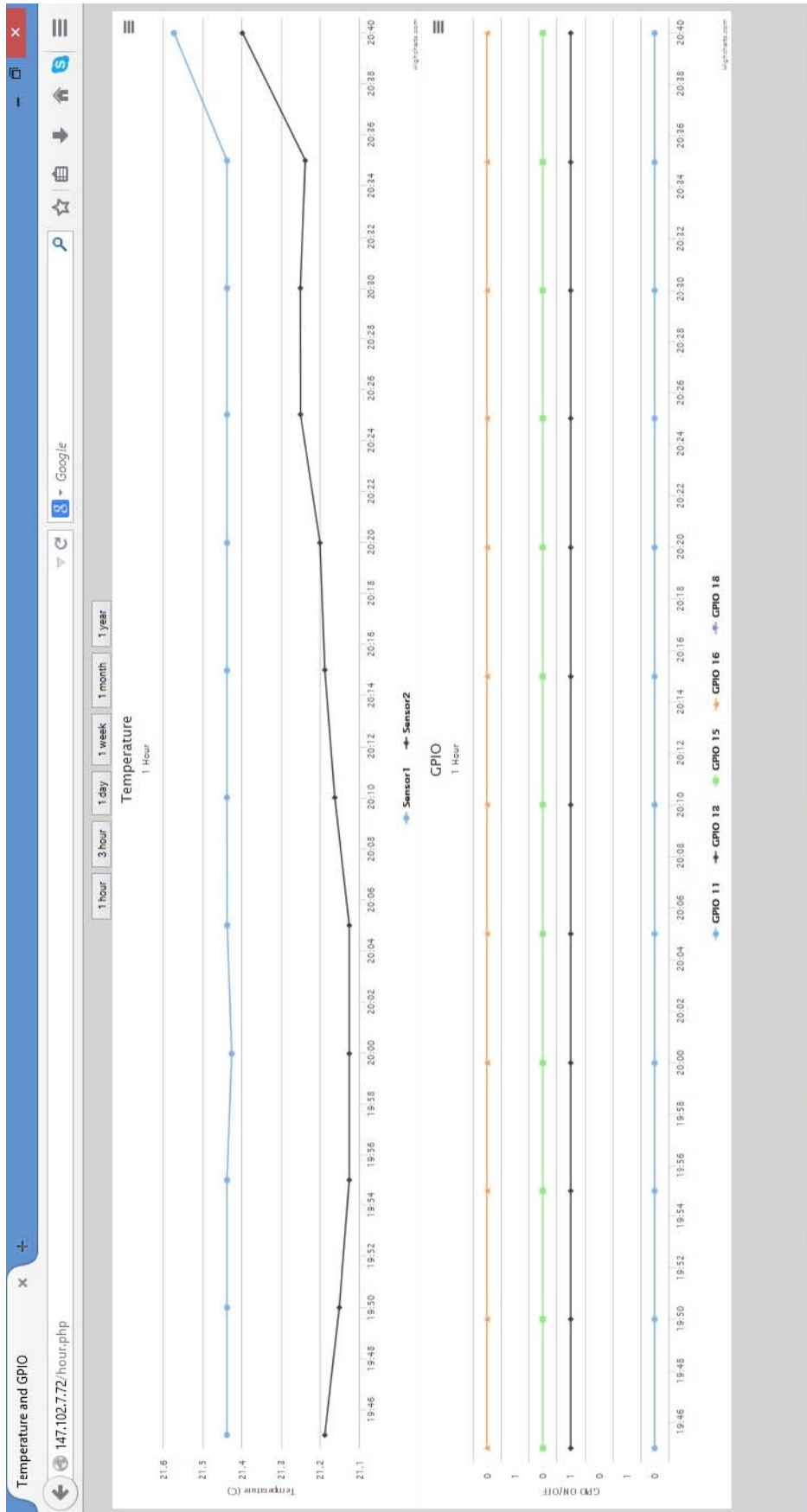
II. Virtual Machine Web Server

Ο χρήστης μπορεί μέσω των σελίδων που σερβίρει ο web server του Virtual Machine να επιλέξει πόσες από τις τελευταίες μετρήσεις θα δει. Μπορεί να επιλέξει την τελευταία ώρα, τις τελευταίες 3 ώρες, την τελευταία μέρα, την τελευταία εβδομάδα, τον τελευταίο μήνα, και τον τελευταίο χρόνο.

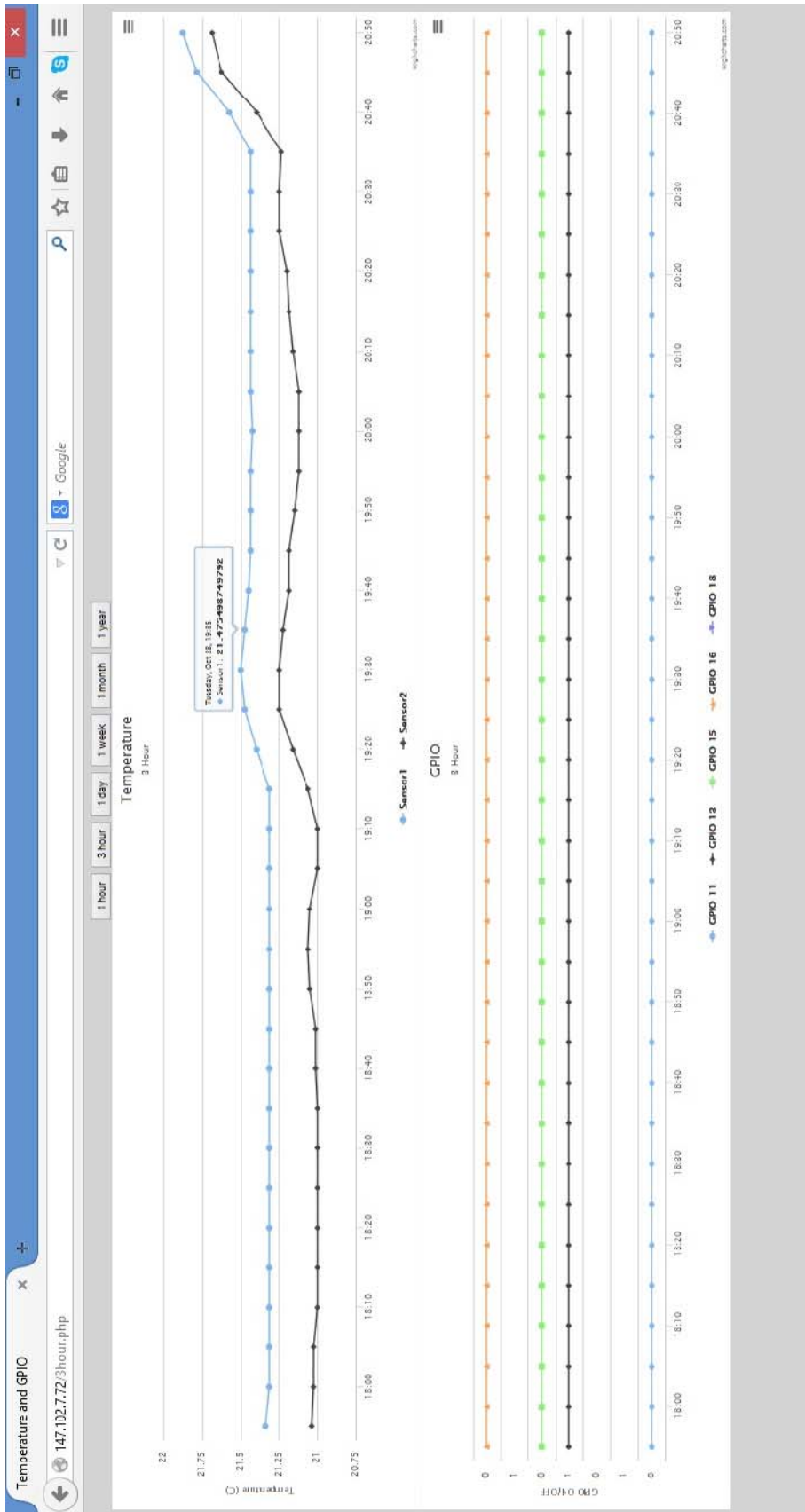
Σε όλες τις σελίδες ο χρήστης μπορεί να κάνει ζουμ στις μετρήσεις που θέλει να δει με μεγαλύτερη ακρίβεια. Επίσης, μπορεί να επιλέξει ποιά στοιχεία θέλει να βλέπει και ποια όχι για να είναι πιο ευανάγνωστη η γραφική παράσταση.

Παρακάτω φαίνονται όλες οι σελίδες που είναι διαθέσιμες για το χρήστη.

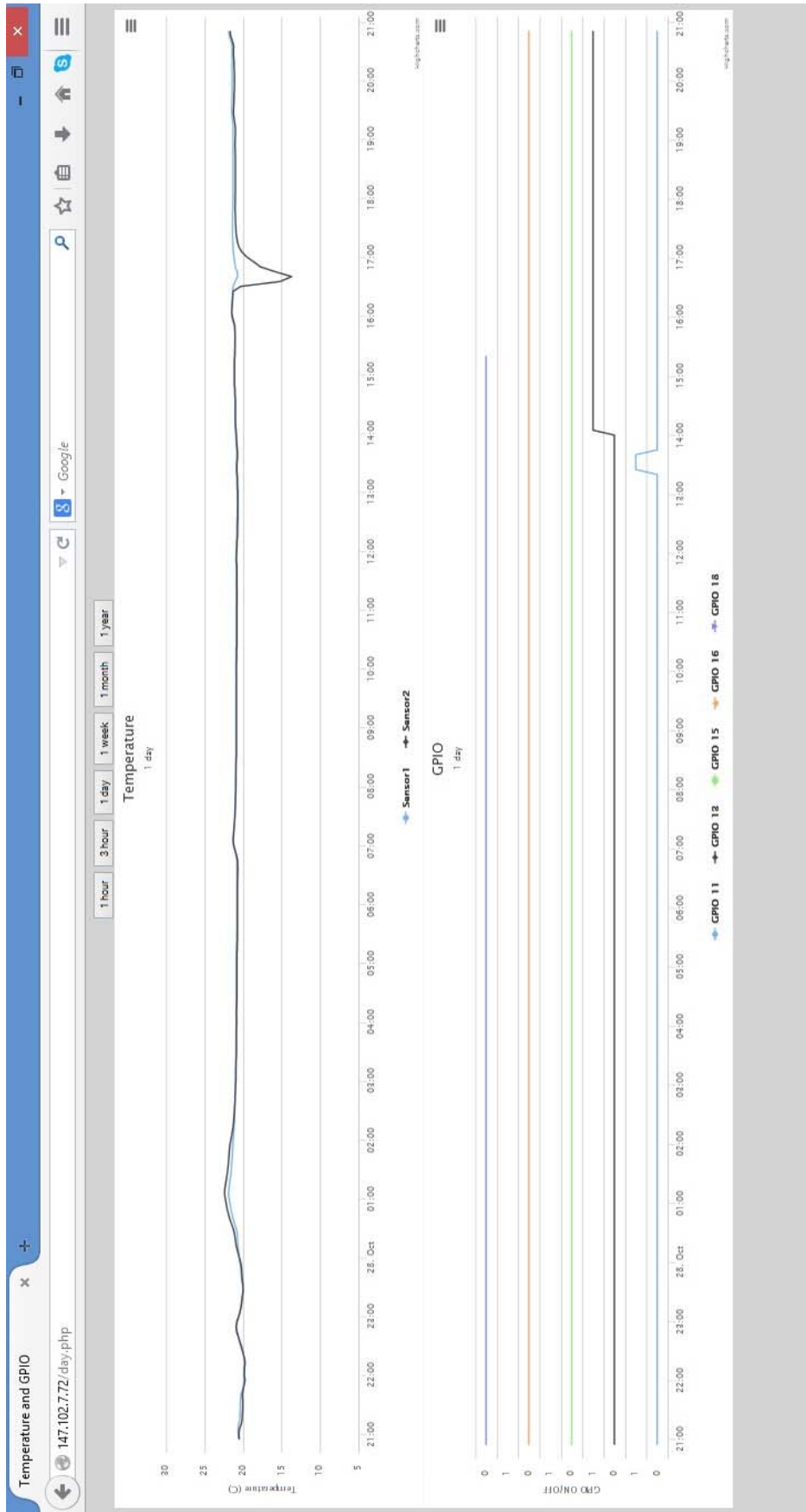
hour.php

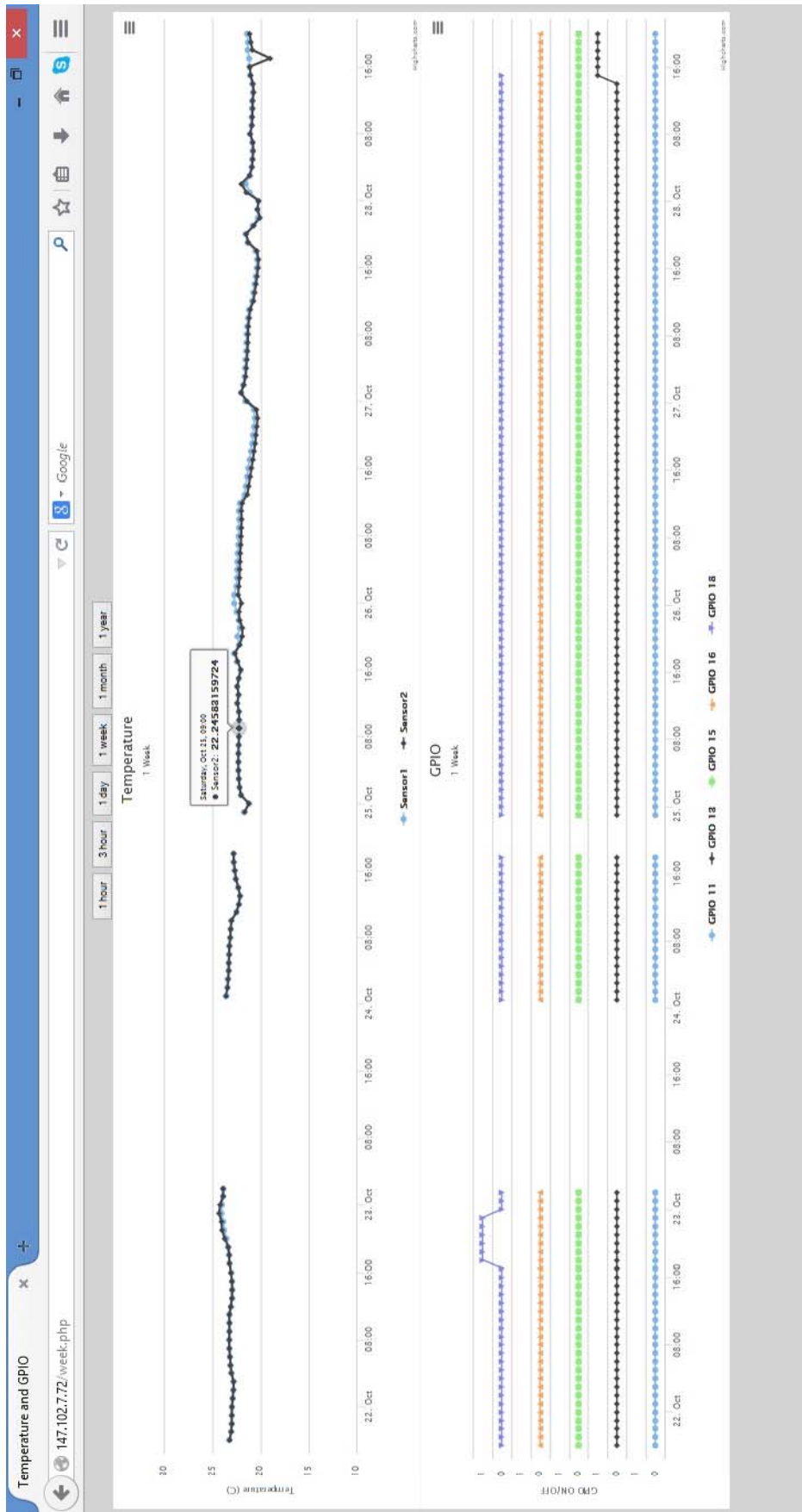


3hour.php

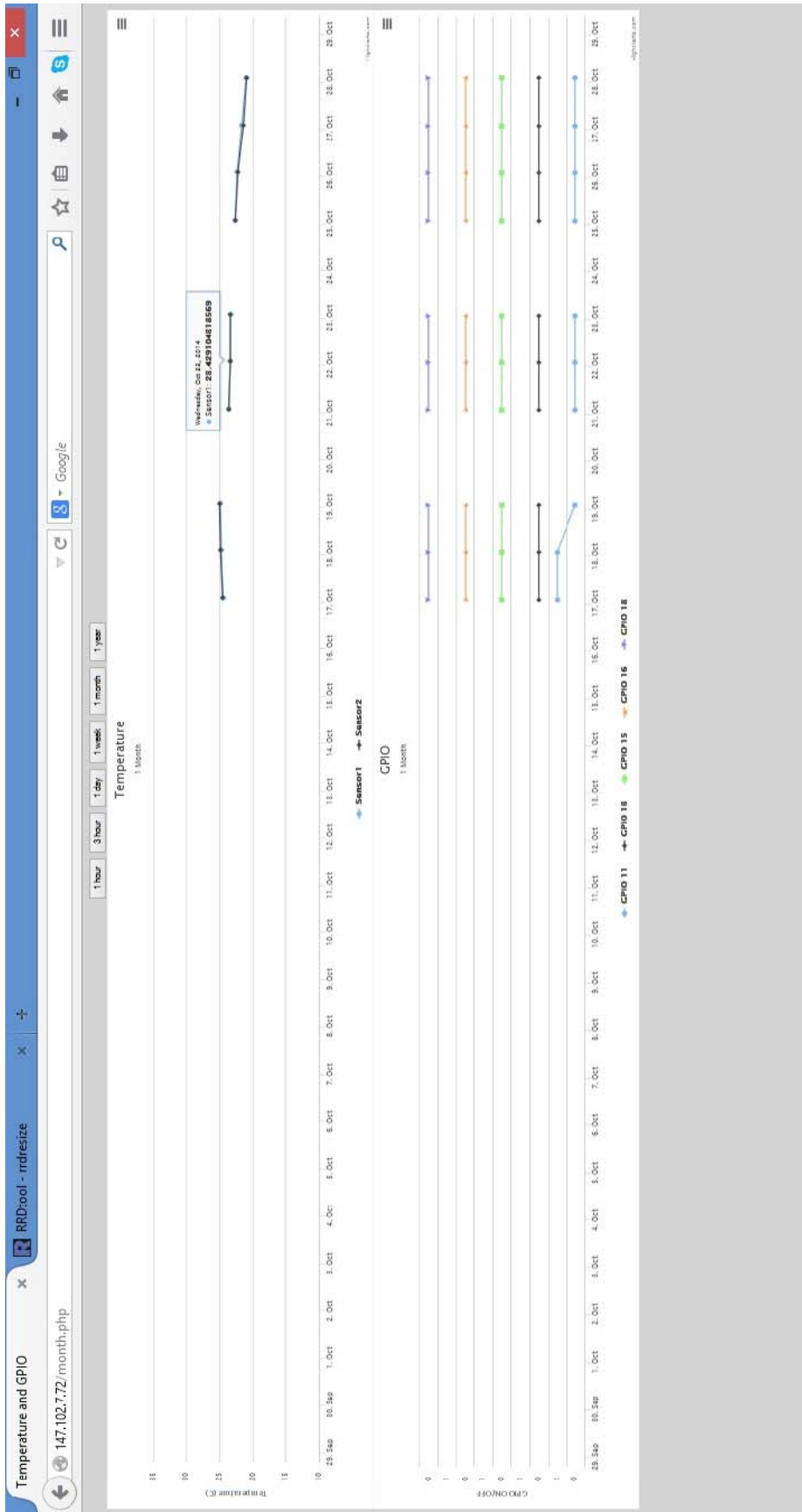


day.php

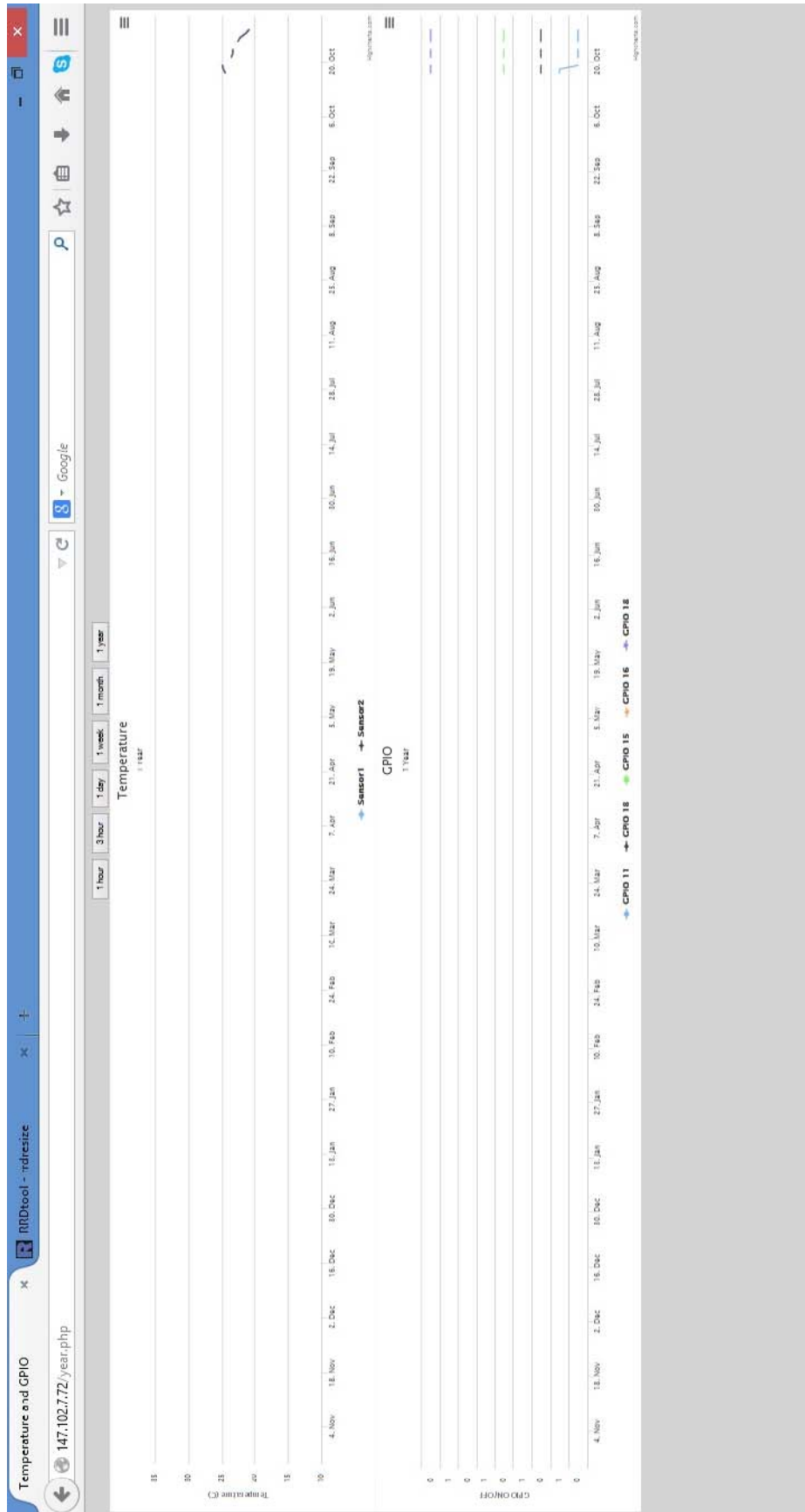




month.php



year.php



Παράρτημα

I. Raspberry Pi

Περιεχόμενα φακέλου /var/www [21]

index.php

```
<!DOCTYPE HTML> <html>
  <head>
    <style> .error {color: #FF0000;,
      font-size:1pt;
    }
  </style>
</head>
<body bgcolor="#D3D3D3">
<?php $Err='';
  if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $myusername=test_input($_POST['myusername']);
    $mypassword=test_input($_POST['mypassword']);
    $handle = fopen("pass.txt", "r");
    if ($handle) {
      while (($line = fgets($handle)) !== false) {
        $buffer[]=$line;
      }
    } else {
      echo "Pi Error...";
    }
    fclose($handle);
    $username=$buffer[0];
    $password=$buffer[1];
    if(((strcmp(trim($username),trim($myusername))==0) &&
(strcmp(trim($password),trim($mypassword))==0))){
      header("location:home.htm");
    } else {
      $Err='Wrong Username or Password';
    }
  }
  function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
  }
  ?>
  <br><br>
  <br><br>
  <br><br>
  <table width="300" border="0" align="center"
cellpadding="0" cellspacing="1" bgcolor="#CCCCCC">
```

```

        <tr>
            <form name="form1" method="post"
action="index.php">
            <font size="6"><center><b>Raspberry
Pi</b></center></font>
            <td>
                <table width="100%" border="0"
cellpadding="3" cellspacing="1" bgcolor="#E6E6FA">
                    <tr> <td colspan="3"><strong>Login</strong></td> </tr>
                    <tr><td width="78">Username</td>
                        <td width="6">:</td>
                        <td width="294"><input
name="myusername" type="text" id="myusername"></td>
                    </tr><tr><td>Password</td><td>:</td>
                        <td><input name="mypassword"
type="password" id="mypassword"></td>
                    </tr><tr><td>&nbsp;</td>
                        <td>&nbsp;</td>
                        <td><input type="submit"
name="Submit" value="Login"></td>
                    </tr>
                    <tr>
                        <td colspan="2" style="text-align: right;"><span
class="error"> <?php echo $Err;?></span>
                        </td>
                    </tr>
                </table>
            </td>
        </form>
    </tr>
</table>
</body>
</html>

```

home.htm

```

<html>
<head>
<title>Raspberry Pi Project</title>
</head>

<frameset rows="100,*" frameborder="10" border="2"
framespacing="0">
    <frame name="topNav" src="top.html">
<frameset cols="200,*" frameborder="10" border="2"
framespacing="0">
    <frame name="menu" src="menu.html" marginheight="0"
marginwidth="0" scrolling="auto" noresize>
    <frame name="content" src="configure.php"
marginheight="0" marginwidth="0" scrolling="auto" noresize>

```

```
<noframes>
<p>Your browser doesn't support frames...</p>
</noframes>

</frameset>
</frameset>
</html>
```

top.html

```
<html>
<head>
<title>Raspberry Pi Project</title>
<style type="text/css">
body {
    font-family:verdana,arial,sans-serif;
    font-size:10pt;
    margin:5px;
    text-align:center;
    background-color:#D3D3D3;
}
</style>
</head>
<body>


<font size="10"><b>Raspberry Pi project</b></font>
<br>
<b>Xristos Epitropakis</b>
</body>
</html>
```

menu.html

```
<html>
<head>
<title>Raspberry Pi Project</title>
<style type="text/css">
body {
    font-family:verdana,arial,sans-serif;
    font-size:10pt;
    margin:10px;
    background-color:#D3D3D3;
}
</style>
</head>
<body>
```

```

<p><a href="configure.php" target="content"><b>Configure
pi</b></a></p>
<p><a href="cron.php" target="content"><b>Set time
intervals</b></a></p>
<p><a href="sensors.php" target="content"><b>Sensors
Selection</b></a></p>
<p><a href="gpio.php" target="content"><b>Input Output
Selection</b></a></p>
<p><a href="results.php" target="content"><b>Current
Values</b></a></p>
<p><a href="changelogin.php" target="content"><b>Change
Username or Password</b></a></p>
<p><a href="reboot.php"
target="content"><b>Reboot</b></a></p>
<p><a href="shutdown.php" target="content"><b>Shut
down</b></a></p>
</body>
</html>

```

configure.php [22]

```

<!DOCTYPE HTML> <html>
  <head>
    <style>
      .error {color: #FF0000;,
        font-size:1pt;
      }
    </style>
  </head> <body bgcolor="#D3D3D3"> <?php function
test_input($data) {
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
}
$ip = $ipErr = $netmaskErr = $networkErr = $broadcastErr =
$gatewayErr =
$netmask = $network = $broadcast = $gateway = $radioErr = "";
$ok=true;
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  if (!empty($_POST["inet"])) {
    $inet=$_POST["inet"];
    if (trim($inet)=="static"){
      $long = ip2long($_POST["ip"]);
      if ($long == -1 || $long === FALSE){
        $ipErr = "*Invalid IP";
        $ok=false;
      }else {
        $ip = test_input($_POST["ip"]);
      }
    }
    $long = ip2long($_POST["netmask"]);

```

```

if ($long == -1 || $long === FALSE){
    $netmaskErr = "*Invalid Netmask";
    $ok=false;
}else {
    $netmask = test_input($_POST["netmask"]);
}
$long = ip2long($_POST["network"]);
if ($long == -1 || $long === FALSE){
    $networkErr = "*Invalid Network";
    $ok=false;
}else {
    $network = test_input($_POST["network"]);
}
$long = ip2long($_POST["broadcast"]);
if ($long == -1 || $long === FALSE){
    $broadcastErr = "*Invalid Broadcast";
    $ok=false;
}else {
    $broadcast = test_input($_POST["broadcast"]);
}
$long = ip2long($_POST["gateway"]);
if ($long == -1 || $long === FALSE){
    $gatewayErr = "*Invalid Gateway";
    $ok=false;
}else {
    $gateway = test_input($_POST["gateway"]);
}
if($ok){
    $data = file("/etc/network/interfaces");
    $Delete = "iface eth0 inet static";
    $Deletel = "iface eth0 inet dhcp";
    $count=0;
    foreach($data as $line) {
        if($count!=0) {
            $count++;
            $line='';
            if ($count==6) {
                $count=0;
                $line="address $ip\nnetmask
$netmask\nnetwork $network\nbroadcast $broadcast\ngateway
$gateway\n";
            }
        }
        if((trim($line) == $Delete)){
            $count=1;
        }
        if((trim($line) == $Deletel)){
            $line="iface eth0 inet
static\naddress $ip\nnetmask $netmask\nnetwork
$network\nbroadcast $broadcast\ngateway $gateway\n";
        }
    }
}

```

```

        $out[] = $line;
    }
    file_put_contents("/etc/network/interfaces",
$out);
}
} else {
    $data = file("/etc/network/interfaces");
    $Delete = "iface eth0 inet static";
    $count=0;
    foreach($data as $line) {
        if($count!=0) {
            $count++;
            $line='';
            if ($count==6) {
                $count=0;
            }
        }
        if((trim($line) == $Delete)){
            $count=1;
            $line="iface eth0 inet dhcp\n";
        }
        $out[] = $line;
    }
    file_put_contents("/etc/network/interfaces",
$out);
}
}
else {
    $radioErr='*Please Select a Protocol';
}
}
?>
<br><br>
<br><br>
<br><br>
<form method="post" action="configure.php">
    <table width=auto border="5" align="center"
cellpadding="0" cellspacing="1" bgcolor="#E6E6FA">
        <tr><td><center><span class="error"> <?php echo
$radioErr;?></span><td></tr>
        <tr><td>Select Protocol <td></tr>
        <tr><td><input type="radio" name="inet"
value="dhcp">dhcp</td><tr>
        <tr><td><input type="radio" name="inet"
value="static">static<br>
        IP: <input type="text" name="ip">
        <span class="error"> <?php echo $ipErr;?></span><br>
        Netmask: <input type="text" name="netmask">
        <span class="error"> <?php echo
$netmaskErr;?></span><br>

```

```

        Network: <input type="text" name="network">
        <span class="error"> <?php echo
$networkErr;?></span><br>
        Broadcast: <input type="text" name="broadcast">
        <span class="error"> <?php echo
$broadcastErr;?></span><br>
        Gateway: <input type="text" name="gateway">
        <span class="error"> <?php echo
$gatewayErr;?></span></td></tr>
        <tr><td><input type="submit" name="submit"
value="Submit"></center></td></tr>
    </table>
</form>
</body>
</html>

```

cron.php

```

<!DOCTYPE HTML>
<html>
    <head>
        <style>
            .error {color: #FF0000;,
                font-size:1pt;
            }
        </style>
    </head>
    <body bgcolor="#D3D3D3">
        <?php function test_input($data) {
            $data = trim($data);
            $data = stripslashes($data);
            $data = htmlspecialchars($data);
            return $data;
        }?>
        <br><br>
        <br><br>
        <br><br>
        <center>
        <?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $mycron=$_POST["crontime"];
    if (is_numeric($mycron)) {
        if ($mycron>0 && $mycron <61){
            $myfile = fopen("/home/pi/diploma/crontab.txt",
"w") or die("Unable to open file!");
            if ($mycron!=60){
                $mycron=intval($mycron);
                $txt = "*/$mycron * * * * sudo python
/home/pi/diploma/sensors.py\n";

```

```

        } else {
            $txt = "* */1 * * * sudo python
/home/pi/diploma/sensors.py\n";
        }
        fwrite($myfile, $txt);
        fclose($myfile);
        exec('sudo crontab
/home/pi/diploma/crontab.txt');
    } else {
        ?> <span class="error"> <?php echo "*Insert a
value between 1-60";?> </span> <?php
    }
    } else {
        ?> <span class="error"> <?php echo "*Insert an
integer"; ?> </span> <?php
    }
} ?>
<form method="post" action="cron.php">
    Set how often to take values (from 1 minute to 60):<br>
    <input type="text" name="crontime"><br>
    <input type="submit" name="submit" value="Submit">
</form>
</center>
</body>
</html>

```

sensors.php [23]

```

<!DOCTYPE HTML>
<html>
    <head>
    </head>
    <body bgcolor="#D3D3D3"> <?php function test_input($data) {
        $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        return $data;
    }
    $mysensors = "";
    $ok=true;
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $mysensors=$_POST["mysensors"];
        $myfile = fopen("/home/pi/diploma/sensors.txt", "w") or
die("Unable to open file!");
        $txt=implode("\n", $mysensors);
        fwrite($myfile, $txt);
        fclose($myfile);
    }
    ?>
    <br><br>

```



```

<center>
<form method="post" action="sensors.php">
  <?php $output = shell_exec("owdir");
    $owdir=explode ("/bus.7", $output);
    $count=0;
    $sensors=$owdir[0];
    if ($sensors!=NULL){
      $sensa=explode("\n", $sensors);
      array_pop($sensa);
    }
    <table width=auto border="5" align="center"
cellpadding="0" cellspacing="1" bgcolor="#E6E6FA">
  <?php
$mystring=file_get_contents("/home/pi/diploma/sensors.txt");
  foreach($sensa as $line) {
    $count++;
    ?>
    <tr><td>
    <?php $pos = strpos($mystring, $line);
    ?>
    <input type="checkbox" name="mysensors[]"
id="mysensors" <?php if ($pos !== false){?>checked<?php } ?>
value="<?php echo $line;?>"><?php if ($pos !== false){?> <b>
<?php } echo $line;?> </b> <?php
    if (trim($line)=="/28.FF287F2D0400") {
      echo "-sensor1";
    }
    elseif (trim($line)=="/28.FFA98E2C0400"){
      echo "-sensor2";
    }
    }?></td></tr><br>
    <?php echo '<br>';
  }
  ?>
  <tr><td><input type="submit" name="submit"
value="Submit"></td></tr>
</table>
</form>
</center>
  <?php
  } else { ?> 0 Sensors detected... <?php } ?>
</body>
</html>

```

gpio.php

```

<!DOCTYPE HTML> <html>
  <head>
  <style>
    .error {color: #FF0000;,
      font-size:1pt;
    }

```

```

        </style>
    </head>
<body bgcolor="#D3D3D3">
<?php function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
$gpio1Err = $gpio2Err = $gpio3Err = $gpio4Err = $gpio5Err = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $myfile = fopen("/home/pi/diploma/gpio.txt", "w") or
die("Unable to open file!");
    if (!empty($_POST["gpio1"])) {
        $mygpio1=$_POST["gpio1"];
        if (!empty($_POST["rgpio1"])) {
            $myrgpio1=$_POST["rgpio1"];
            $txt="$mygpio1 $myrgpio1\n";
            fwrite($myfile, $txt);
        } else {
            $gpio1Err = "*Select input or output";
        }
    }
    if (!empty($_POST["gpio2"])) {
        $mygpio2=$_POST["gpio2"];
        if (!empty($_POST["rgpio2"])) {
            $myrgpio2=$_POST["rgpio2"];
            $txt="$mygpio2 $myrgpio2\n";
            fwrite($myfile, $txt);
        } else {
            $gpio2Err = "*Select input or output";
        }
    }
    if (!empty($_POST["gpio3"])) {
        $mygpio3=$_POST["gpio3"];
        if (!empty($_POST["rgpio3"])) {
            $myrgpio3=$_POST["rgpio3"];
            $txt="$mygpio3 $myrgpio3\n";
            fwrite($myfile, $txt);
        } else {
            $gpio3Err = "*Select input or output";
        }
    }
    if (!empty($_POST["gpio4"])) {
        $mygpio4=$_POST["gpio4"];
        if (!empty($_POST["rgpio4"])) {
            $myrgpio4=$_POST["rgpio4"];
            $txt="$mygpio4 $myrgpio4\n";
            fwrite($myfile, $txt);
        } else {
            $gpio4Err = "*Select input or output";
        }
    }
    if (!empty($_POST["gpio5"])) {

```

```

        $mygpio5=$_POST["gpio5"];
        if (!empty($_POST["rgpio5"])) {
            $myrgpio5=$_POST["rgpio5"];
            $txt="$mygpio5 $myrgpio5\n";
            fwrite($myfile, $txt);
        } else {
            $gpio5Err = "*Select input or output";
        }
    }
    fclose($myfile);
}
?>
<br><br>
<br><br>
<br><br>
<center>
<table width=auto border="5" align="center"
cellpadding="0" cellspacing="1" bgcolor="#E6E6FA">
    <form method="post" action="gpio.php">
        <?php
$mystring=file_get_contents("/home/pi/diploma/gpio.txt");
    $gp="11";
    $pos = strpos($mystring, $gp);
?>
        <tr><td><input type="checkbox" name="gpio1"
value="11" <?php if ($pos !== false){ ?> checked <?php } ?> >
<?php if ($pos !== false){?> <b> <?php } ?> GPIO 11
        <?php $gp="11 IN";
        $pos = strpos($mystring, $gp);?>
        <input type="radio" name="rgpio1" value="IN"<?php if
($pos !== false){ ?> checked <?php } ?>>Input
        <?php $gp="11 OUT";
        $pos = strpos($mystring, $gp);?>
        <input type="radio" name="rgpio1" value="OUT"<?php if
($pos !== false){ ?> checked <?php } ?>>Output<span
class="error"><?php echo $gpio1Err;?></span></td></tr>
<?php $gp="13";
    $pos = strpos($mystring, $gp);
?>
        <tr><td><input type="checkbox" name="gpio2"
value="13" <?php if ($pos !== false){ ?> checked <?php } ?>>
<?php if ($pos !== false){?> <b> <?php } ?>GPIO 13
        <?php $gp="13 IN";
        $pos = strpos($mystring, $gp);?>
        <input type="radio" name="rgpio2" value="IN"<?php if
($pos !== false){ ?> checked <?php } ?>>Input
        <?php $gp="13 OUT";
        $pos = strpos($mystring, $gp);?>
        <input type="radio" name="rgpio2" value="OUT"<?php if
($pos !== false){ ?> checked <?php } ?>>Output <span
class="error"> <?php echo $gpio2Err;?></span></td></tr>
<?php $gp="15";
    $pos = strpos($mystring, $gp);

```

```

?>
    <tr><td><input type="checkbox" name="gpio3"
value="15" <?php if ($pos !== false){ ?> checked <?php } ?>>
<?php if ($pos !== false){?> <b> <?php } ?>GPIO 15
    <?php $gp="15 IN";
    $pos = strpos($mystring, $gp);?>
    <input type="radio" name="rgpio3" value="IN"<?php if
($pos !== false){ ?> checked <?php } ?>>Input
    <?php $gp="15 OUT";
    $pos = strpos($mystring, $gp);?>
    <input type="radio" name="rgpio3" value="OUT"<?php if
($pos !== false){ ?> checked <?php } ?>>Output <span
class="error"> <?php echo $gpio3Err;?></span></td></tr>
<?php $gp="16";
    $pos = strpos($mystring, $gp);
?>
    <tr><td><input type="checkbox" name="gpio4"
value="16" <?php if ($pos !== false){ ?> checked <?php } ?>>
<?php if ($pos !== false){?> <b> <?php } ?>GPIO 16
    <?php $gp="16 IN";
    $pos = strpos($mystring, $gp);?>
    <input type="radio" name="rgpio4" value="IN"<?php if
($pos !== false){ ?> checked <?php } ?>>Input
    <?php $gp="16 OUT";
    $pos = strpos($mystring, $gp);?>
    <input type="radio" name="rgpio4" value="OUT"<?php if
($pos !== false){ ?> checked <?php } ?>>Output <span
class="error"> <?php echo $gpio4Err;?></span></td></tr>
<?php $gp="18";
    $pos = strpos($mystring, $gp);
?>
    <tr><td><input type="checkbox" name="gpio5"
value="18" <?php if ($pos !== false){ ?> checked <?php } ?>>
<?php if ($pos !== false){?> <b> <?php } ?>GPIO 18
    <?php $gp="18 IN";
    $pos = strpos($mystring, $gp);?>
    <input type="radio" name="rgpio5" value="IN"<?php if
($pos !== false){ ?> checked <?php } ?>>Input
    <?php $gp="18 OUT";
    $pos = strpos($mystring, $gp);?>
    <input type="radio" name="rgpio5" value="OUT"<?php if
($pos !== false){ ?> checked <?php } ?>>Output <span
class="error"> <?php echo $gpio5Err;?></span></td></tr>
    <tr><td><input type="submit" name="submit"
value="Submit"></td></tr>
</table>
</form>
</center>
</body>
</html>

```

results.php

```
<!DOCTYPE HTML> <html>
  <head>
    <style>
      body {
        font-family: verdana, arial, sans-serif;
        font-size: 10;
        margin: 10px;
        text-align: center;
      }
    </style>
  </head>
  <body bgcolor="#D3D3D3">
    <?php function test_input($data) {
      $data = trim($data);
      $data = stripslashes($data);
      $data = htmlspecialchars($data);
      return $data;
    }
    $mysensors = ""; $ok=true;
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
      $change=$_POST["change"];
      exec("sudo python /home/pi/diploma/revertvalue.py
$change");
    }
    <br><br>
    <br><br>
    <br><br>
    <form method="post" action="results.php">
      <?php
        $handle = fopen("/home/pi/diploma/values.txt",
"r");
        if ($handle) {
          while (($line = fgets($handle)) !== false) {
            $buffer[]=$line;
          }
        } else {
          echo "Pi Error...";
        }
        fclose($handle);
        ?> <?php
        $gpio=false;
        if (!empty($buffer)){
          ?><table width="auto" border="5"
align="center" cellpadding="0" cellspacing="1"
bgcolor="#E6E6FA">
            <?php foreach($buffer as $line) {
              ?><tr><td><?php
                echo $line;
```

```

        if ($gpio){
            if ((substr($line,5,3))=="OUT"){
                $valu=substr($line,0,2);
                $valu=intval($valu);
                ?><button name="change"
type="submit" value="<?php echo $valu ?>">Change</button>
                <?php }
            }
            if ((trim($line))=="GPIO") {
                $gpio=true;
            }
            ?>
        </tr></td>

<?php } ?> </table> <?php
} else {
    echo "Storing new values... Wait a
second..";
    ?> <meta http-equiv="refresh" content="2">
<?php
} ?>
</form>
</body>
</html>

```

changelogin.php

```

<!DOCTYPE HTML> <html>
    <head>
    </head> <body bgcolor="#D3D3D3"> <?php if
($ _SERVER["REQUEST_METHOD"] == "POST") {
    $myusername=test_input($_POST['myusername']);
    $mypassword=test_input($_POST['mypassword']);
    $handle = fopen("pass.txt", "w");
    if ($handle) {
        fwrite($handle, $myusername);
        fwrite($handle, "\n");
        fwrite($handle, $mypassword);
    } else {
        echo "Pi Error...";
    }
    fclose($handle);
    header("location:admin.php");
}
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);

```

```

    $data = htmlspecialchars($data);
    return $data;
}
?>
    <br><br>
    <br><br>
    <br><br>
    <table width="300" border="0" align="center"
cellpadding="0"
cellspacing="1" bgcolor="#CCCCCC">
    <tr>
        <form name="form1" method="post"
action="changelogin.php">
        <td>
            <table width="100%" border="0"
cellpadding="3" cellspacing="1" bgcolor="#E6E6FA">
                <tr>
                    <td
colspan="3"><strong>Changing Login</strong></td>
                </tr>
                <tr>
                    <td width="78">New
Username</td>
                    <td width="6">:</td>
                    <td width="294"><input
name="myusername" type="text" id="myusername"></td>
                </tr>
                <tr>
                    <td>New Password</td>
                    <td>:</td>
                    <td><input
name="mypassword" type="password" id="mypassword"></td>
                </tr>
                <tr>
                    <td>&nbsp;</td>
                    <td>&nbsp;</td>
                    <td><input type="submit"
name="Submit" value="Submit"></td>
                </tr>
            </table>
        </td>
    </form>
    </tr>
</table>
</body> </html>

```

reboot.php

```

<!DOCTYPE HTML>
<html>

```

```
<body bgcolor="#D3D3D3">
<center><b>System is Rebooting.....</b></center>
<?php $output = shell_exec("sudo shutdown -r 0");?>
</body>
</html>
```

shutdown.php

```
<!DOCTYPE HTML>
<html>
<body>
<center><b>System is Shutting Down.....</b></center>
<?php $output = shell_exec("sudo shutdown -h 0");?>
</body>
</html>
```

pass.txt

```
admin
admin
```


Περιεχόμενα φακέλου /home/pi/diploma [25]

sensors.py

```
import mosquitto
import os,time,datetime
import RPi.GPIO as GPIO
def on_publish(mosq, obj, mid):
    pass
cli = mosquitto.Mosquitto()
cli.on_publish = on_publish
cli.connect("127.0.0.1")
sens = open('/home/pi/diploma/sensors.txt', 'r')
ts = time.time()
sd = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
st = datetime.datetime.fromtimestamp(ts).strftime('%H:%M')
path='/home/pi/diploma/logfiles/'+sd+'/'
if not os.path.exists(path):
    os.mkdir(path)
valname=path+st
val= open(valname,'w')
cur=open('/home/pi/diploma/values.txt','w')
oksen=1
for line in sens:
    sens='owread '+line.rstrip()+'/temperature; echo'
    s1=os.popen(sens).read()
    s1=float(s1.rstrip())
    cli.publish("raspberry/sensors",str(ts)+'
'+line.rstrip()+ ' '+str(s1),1)
    val.write(line.rstrip()+ ' '+str(s1)+'\n')
    cur.write(line.rstrip()+ ' '+str(s1)+'\n')
    oksen=0
if oksen:
    cur.write('No sensor selected\n')
cur.write('GPIO\n')
gpio = open('/home/pi/diploma/gpio.txt', 'r')

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
okgpio=1
for line in gpio:
    okgpio=0
    GPIO.setmode(GPIO.BOARD)
    nline=line[0:2]
    ninout=line[3:]
    nline=int(nline)
    if (ninout.rstrip() == 'IN'):
        inout=GPIO.IN
    else:
        inout=GPIO.OUT
    GPIO.setup(nline, inout)
```

```

        cli.publish("raspberrypi/gpio",str(ts)+' '+str(nline)+'
'+str(GPIO.input(nline))+' '+ninout.rstrip(),1)
        val.write(str(nline)+' '+str(GPIO.input(nline))+'
'+ninout.rstrip()+'\n')
        cur.write(str(nline)+' '+str(GPIO.input(nline))+'
'+ninout.rstrip()+'\n')
if okgpio:
        cur.write('No GPIO selected\n')

```

revertvalue.py [26]

```

import sys
import RPi.GPIO as GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
mygpio=int(sys.argv[1])
GPIO.setup(mygpio,GPIO.OUT)
myin=GPIO.input(mygpio)
if myin:
        GPIO.output(mygpio,False)
        myin2=0
else:
        GPIO.output(mygpio,True)
        myin2=1
file=open('/home/pi/diploma/values.txt','r')
str1="%d %d OUT"%(mygpio,myin)
str2="%d %d OUT"%(mygpio,myin2)
data=file.read().replace(str1,str2)
file.close()
file=open('/home/pi/diploma/values.txt','w')
file.write(data)
file.close()
myin=GPIO.input(mygpio)

```

gpio.txt

```

11 OUT
13 OUT
16 IN
18 IN

```

sensors.txt

```

/28.FF287F2D0400
/28.FFA98E2C0400

```

values.txt

```
/28.FF287F2D0400 21.5  
/28.FFA98E2C0400 21.25  
GPIO  
11 0 OUT  
13 0 OUT  
16 0 IN  
18 0 IN
```

crontab.txt

```
*/1 * * * * sudo python /home/pi/diploma/sensors.py
```

II. Virtual Machine

Περιεχόμενα φακέλου /var/www/html

index.php

```
<!DOCTYPE html>
<html>
<body>
<?php header("location:hour.php");?>

</body>
</html>
```

hour.php

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <title>Temperature and GPIO</title>
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery
.min.js"></script>
    <style type="text/css">
      ${demo.css}
    </style>
  <?php $ret = rrd_fetch
("/home/christos/diploma/28.FF287F2D0400.rrd", array(
"AVERAGE", "-s", "-1h", "-e", "-300" ));
$start=$ret[start];
$start+=300;
foreach ($ret[data][sensor] as $time=>$key ) {
  if (trim($key) != "NAN"){
    $mydata[]=$key;
  }
  else {
    $mydata[]='null';
  }
}
$myfinal=implode(" , ", $mydata);
$ret2 = rrd_fetch
("/home/christos/diploma/28.FFA98E2C0400.rrd", array(
"AVERAGE", "-s", "-1h", "-e", "-300" ));
foreach($ret2[data][sensor] as $time=>$key ) {
  if (trim($key) != "NAN"){
    $mydata2[]=$key;
  }
  else {
```

```

        $mydata2[]='null';
    }
}
$myfinal2=implode(" , ",$mydata2);
$ret3 = rrd_fetch ("/home/christos/diploma/11.rrd", array(
"AVERAGE", "-s", "-1h", "-e", "-300" ));
foreach ($ret3[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata3[]=1;
        }
        else {
            $mydata3[]=0;
        }
    }
    else {
        $mydata3[]='null';
    }
}
$myfinal3=implode(" , ",$mydata3);
$ret4 = rrd_fetch ("/home/christos/diploma/13.rrd", array(
"AVERAGE", "-s", "-1h", "-e", "-300" ));
foreach ($ret4[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata4[]=3;
        }
        else {
            $mydata4[]=2;
        }
    }
    else {
        $mydata4[]='null';
    }
}
$myfinal4=implode(" , ",$mydata4);
$ret5 = rrd_fetch ("/home/christos/diploma/15.rrd", array(
"AVERAGE", "-s", "-1h", "-e", "-300" ));
foreach ($ret5[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata5[]=5;
        }
        else {
            $mydata5[]=4;
        }
    }
    else {
        $mydata5[]='null';
    }
}
}

```

```

$myfinal5=implode(" , ", $mydata5);
$ret6 = rrd_fetch ( "/home/christos/diploma/16.rrd", array(
"AVERAGE", "-s", "-1h", "-e", "-300" ) );
foreach ( $ret6[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata6[]=7;
        }
        else {
            $mydata6[]=6;
        }
    }
    else {
        $mydata6[]='null';
    }
}
$myfinal6=implode(" , ", $mydata6);
$ret7 = rrd_fetch ( "/home/christos/diploma/18.rrd", array(
"AVERAGE", "-s", "-1h", "-e", "-300" ) );
foreach ( $ret7[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata7[]=9;
        }
        else {
            $mydata7[]=8;
        }
    }
    else {
        $mydata7[]='null';
    }
}
$myfinal7=implode(" , ", $mydata7); ?>
<script type="text/javascript">
$(function () {
    $('#container').highcharts({
        chart: {
            zoomType: 'x'
        },
        title: {
            text: 'Temperature'
        },
        subtitle: {
            text: '1 Hour'
        },
        xAxis: {
            type: 'datetime',
            maxZoom: 600 * 1000
        },
        yAxis: {
            title: {

```

```

        text: 'Temperature (C)'
    }},
    series: [{
        name: 'Sensor1',
        data: [<?php echo $myfinal;?>],
        pointStart: (<?php echo $start+7200;?>)*1000,
        pointInterval: 300*1000
    },{
        name: ' Sensor2',
        data: [<?php echo $myfinal2;?>],
        pointStart: (<?php echo $start+7200;?>)*1000,
        pointInterval: 300*1000
    }
    ]
});
});
$(function () {
    $('#container2').highcharts({
        chart: {
            zoomType: 'x'
        },
        title: {
            text: 'GPIO'
        },
        subtitle: {
            text: '1 Hour'
        },
        xAxis: {
            type: 'datetime',
            maxZoom: 600 * 1000
        },
        yAxis: {
            title: {
                text: 'GPIO ON/OFF'
            }
        },
        categories:[0,1,0,1,0,1,0,1,0,1]
    },
    series: [{
        name: 'GPIO 11',
        data: [<?php echo $myfinal3;?>],
        pointStart: (<?php echo $start;?> +7200)*1000,
        pointInterval: 300*1000
    },{
        name: 'GPIO 13',
        data: [<?php echo $myfinal4;?>],
        pointStart: (<?php echo $start;?> +7200)*1000,
        pointInterval: 300*1000
    },{
        name: 'GPIO 15',
        data: [<?php echo $myfinal5;?>],
        pointStart: (<?php echo $start;?> +7200)*1000,

```

```

        pointInterval: 300*1000
    },{
name: 'GPIO 16',
    data: [<?php echo $myfinal6;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 300*1000
    },{
name: 'GPIO 18',
    data: [<?php echo $myfinal7;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 300*1000
    }
    ]
});
</script>
</head>
<body bgcolor="#D3D3D3">
    <script src="./js/highcharts.js"></script>
    <script src="./js/modules/exporting.js"></script>
    <center>
        <button onclick="location.href='hour.php'">1
hour</button>
        <button onclick="location.href='3hour.php'">3
hour</button>
        <button onclick="location.href='day.php'">1
day</button>
        <button onclick="location.href='week.php'">1
week</button>
        <button onclick="location.href='month.php'">1
month</button>
        <button onclick="location.href='year.php'">1
year</button>
    </center>
    <div id="container" style="min-width: 310px; height:
400px; margin: 0 auto"></div>
    <div id="container2" style="min-width: 310px; height:
400px; margin: 0 auto"></div>
</body>
</html>

```

3hour.php

```

<!DOCTYPE HTML> <html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
        <title>Temperature and GPIO</title>

```



```

        <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery
.min.js"></script>
        <style type="text/css">
            ${demo.css}
        </style>
        <?php $ret = rrd_fetch
("/home/christos/diploma/28.FF287F2D0400.rrd", array(
"AVERAGE", "-s", "-3h", "-e", "-300" ) );
$start=$ret[start];
$start+=300;
foreach ( $ret[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        $mydata[]=$key;
    }
    else {
        $mydata[]='null';
    }
}
$myfinal=implode(" , ", $mydata);
$ret2 = rrd_fetch
("/home/christos/diploma/28.FFA98E2C0400.rrd", array(
"AVERAGE", "-s", "-3h", "-e", "-300" ) );
foreach ( $ret2[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        $mydata2[]=$key;
    }
    else {
        $mydata2[]='null';
    }
}
$myfinal2=implode(" , ", $mydata2);
$ret3 = rrd_fetch ("/home/christos/diploma/11.rrd", array(
"AVERAGE", "-s", "-3h", "-e", "-300" ) );
foreach ( $ret3[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata3[]=1;
        }
        else {
            $mydata3[]=0;
        }
    }
    else {
        $mydata3[]='null';
    }
}
$myfinal3=implode(" , ", $mydata3);
$ret4 = rrd_fetch ("/home/christos/diploma/13.rrd", array(
"AVERAGE", "-s", "-3h", "-e", "-300" ) );
foreach ( $ret4[data][sensor] as $time=>$key ) {

```

```

    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata4[]=3;
        }
        else {
            $mydata4[]=2;
        }
    }
    else {
        $mydata4[]='null';
    }
}
$myfinal4=implode(" , ",$mydata4);
$ret5 = rrd_fetch ("/home/christos/diploma/15.rrd", array(
"AVERAGE", "-s", "-3h", "-e", "-300" ));
foreach ( $ret5[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata5[]=5;
        }
        else {
            $mydata5[]=4;
        }
    }
    else {
        $mydata5[]='null';
    }
}
$myfinal5=implode(" , ",$mydata5);
$ret6 = rrd_fetch ("/home/christos/diploma/16.rrd", array(
"AVERAGE", "-s", "-3h", "-e", "-300" ));
foreach ( $ret6[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata6[]=7;
        }
        else {
            $mydata6[]=6;
        }
    }
    else {
        $mydata6[]='null';
    }
}
$myfinal6=implode(" , ",$mydata6);
$ret7 = rrd_fetch ("/home/christos/diploma/18.rrd", array(
"AVERAGE", "-s", "-3h", "-e", "-300" ));
foreach ( $ret7[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata7[]=9;
        }
    }
}

```

```

    }
    else {
        $mydata7[]=8;
    }
}
else {
    $mydata7[]='null';
}
}
$myfinal7=implode(" , ", $mydata7); ?>
<script type="text/javascript"> $(function () {
    $('#container').highcharts({
        chart: {
            zoomType: 'x'
        },
        title: {
            text: 'Temperature'
        },
        subtitle: {
            text: '3 Hour'
        },
        xAxis: {
            type: 'datetime',
            maxZoom: 600 * 1000
        },
        yAxis: {
            title: {
                text: 'Temperature (C)'
            }
        },
        series: [{
            name: 'Sensor1',
            data: [<?php echo $myfinal;?>],
            pointStart: (<?php echo $start;?> +7200)*1000,
            pointInterval: 300*1000
        },{
            name: ' Sensor2',
            data: [<?php echo $myfinal2;?>],
            pointStart: (<?php echo $start;?> +7200)*1000,
            pointInterval: 300*1000
        }
    ]
});
});
$(function () {
    $('#container2').highcharts({
        chart: {
            zoomType: 'x'
        },
        title: {
            text: 'GPIO'
        },
    },

```

```

        subtitle: {
            text: '3 Hour'
        },
        xAxis: {
            type: 'datetime',
            maxZoom: 600 * 1000
        },
        yAxis: {
            title: {
                text: 'GPIO ON/OFF'
            }
        },
        categories:[0,1,0,1,0,1,0,1,0,1]
    },
    series: [{
        name: 'GPIO 11',
        data: [<?php echo $myfinal3;?>],
        pointStart: (<?php echo $start;?> +7200)*1000,
        pointInterval: 300*1000
    },{
        name: 'GPIO 13',
        data: [<?php echo $myfinal4;?>],
        pointStart: (<?php echo $start;?> +7200)*1000,
        pointInterval: 300*1000
    },{
        name: 'GPIO 15',
        data: [<?php echo $myfinal5;?>],
        pointStart: (<?php echo $start;?> +7200)*1000,
        pointInterval: 300*1000
    },{
        name: 'GPIO 16',
        data: [<?php echo $myfinal6;?>],
        pointStart: (<?php echo $start;?> +7200)*1000,
        pointInterval: 300*1000
    },{
        name: 'GPIO 18',
        data: [<?php echo $myfinal7;?>],
        pointStart: (<?php echo $start;?> +7200)*1000,
        pointInterval: 300*1000
    }
    ]
    });
});
</script>
</head>
<body bgcolor="#D3D3D3">
    <script src="./js/highcharts.js"></script>
    <script src="./js/modules/exporting.js"></script>
    <center>
        <button onclick="location.href='hour.php'">1
hour</button>

```

```

        <button onclick="location.href='3hour.php'">3
hour</button>
        <button onclick="location.href='day.php'">1
day</button>
        <button onclick="location.href='week.php'">1
week</button>
        <button onclick="location.href='month.php'">1
month</button>
        <button onclick="location.href='year.php'">1
year</button>
    </center>
    <div id="container" style="min-width: 310px; height:
400px; margin: 0 auto"></div>
    <div id="container2" style="min-width: 310px; height:
400px; margin: 0 auto"></div>
</body>
</html>

```

day.php

```

<!DOCTYPE HTML> <html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <title>Temperature and GPIO</title>
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery
.min.js"></script>
    <style type="text/css">
      ${demo.css}
    </style>
  <?php $ret = rrd_fetch
( "/home/christos/diploma/28.FF287F2D0400.rrd", array(
"AVERAGE", "-s", "-1d", "-e", "-300" ) );
$start=$ret[start];
$start+=300;
foreach ( $ret[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        $mydata[]=$key;
    }
    else {
        $mydata[]='null';
    }
}
$myfinal=implode(" , ", $mydata);
$ret2 = rrd_fetch
( "/home/christos/diploma/28.FFA98E2C0400.rrd", array(
"AVERAGE", "-s", "-1d", "-e", "-300" ) );
foreach ( $ret2[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){

```

```

        $mydata2[]=$key;
    }
    else {
        $mydata2[]='null';
    }
}
$myfinal2=implode(" , ", $mydata2);
$ret3 = rrd_fetch ("/home/christos/diploma/11.rrd", array(
"AVERAGE", "-s", "-1d", "-e", "-300" ));
foreach ( $ret3[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata3[]=1;
        }
        else {
            $mydata3[]=0;
        }
    }
    else {
        $mydata3[]='null';
    }
}
$myfinal3=implode(" , ", $mydata3);
$ret4 = rrd_fetch ("/home/christos/diploma/13.rrd", array(
"AVERAGE", "-s", "-1d", "-e", "-300" ));
foreach ( $ret4[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata4[]=3;
        }
        else {
            $mydata4[]=2;
        }
    }
    else {
        $mydata4[]='null';
    }
}
$myfinal4=implode(" , ", $mydata4);
$ret5 = rrd_fetch ("/home/christos/diploma/15.rrd", array(
"AVERAGE", "-s", "-1d", "-e", "-300" ));
foreach ( $ret5[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata5[]=5;
        }
        else {
            $mydata5[]=4;
        }
    }
    else {

```

```

        $mydata5[]='null';
    }
}
$myfinal5=implode(" , ",$mydata5);
$ret6 = rrd_fetch ("/home/christos/diploma/16.rrd", array(
"AVERAGE", "-s", "-1d", "-e", "-300" ));
foreach ( $ret6[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata6[]=7;
        }
        else {
            $mydata6[]=6;
        }
    }
    else {
        $mydata6[]='null';
    }
}
$myfinal6=implode(" , ",$mydata6);
$ret7 = rrd_fetch ("/home/christos/diploma/18.rrd", array(
"AVERAGE", "-s", "-1d", "-e", "-300" ));
foreach ( $ret7[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata7[]=9;
        }
        else {
            $mydata7[]=8;
        }
    }
    else {
        $mydata7[]='null';
    }
}
$myfinal7=implode(" , ",$mydata7); ?>
<script type="text/javascript"> $(function () {
    $('#container').highcharts({
        chart: {
            zoomType: 'x'
        },
        title: {
            text: 'Temperature'
        },
        subtitle: {
            text: '1 day'
        },
        xAxis: {
            type: 'datetime',
            maxZoom: 12 * 600 * 1000
        },

```

```

yAxis: {
    title: {
        text: 'Temperature (C)'
    },
    minRange: 15
},
series: [{
    name: 'Sensor1',
    data: [<?php echo $myfinal;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 300*1000
}, {
    name: ' Sensor2',
    data: [<?php echo $myfinal2;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 300*1000
}
]
});
$(function () {
    $('#container2').highcharts({
        chart: {
            zoomType: 'x'
        },
        title: {
            text: 'GPIO'
        },
        subtitle: {
            text: '1 day'
        },
        xAxis: {
            type: 'datetime',
            maxZoom: 12 * 600 * 1000
        },
        yAxis: {
            title: {
                text: 'GPIO ON/OFF'
            }
        },
        categories:[0,1,0,1,0,1,0,1,0,1]
    },
    series: [{
        name: 'GPIO 11',
        data: [<?php echo $myfinal3;?>],
        pointStart: (<?php echo $start;?> +7200)*1000,
        pointInterval: 300*1000
    }, {
        name: 'GPIO 13',
        data: [<?php echo $myfinal4;?>],
        pointStart: (<?php echo $start;?> +7200)*1000,
        pointInterval: 300*1000
    }
}
}
});

```



```

    },{
    name: 'GPIO 15',
    data: [<?php echo $myfinal5;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 300*1000
    },{
    name: 'GPIO 16',
    data: [<?php echo $myfinal6;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 300*1000
    },{
    name: 'GPIO 18',
    data: [<?php echo $myfinal7;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 300*1000
    }
    ]
    });
});
</script>
</head>
<body bgcolor="#D3D3D3">
    <script src="./js/highcharts.js"></script>
    <script src="./js/modules/exporting.js"></script>
    <center>
        <button onclick="location.href='hour.php'">1
hour</button>
        <button onclick="location.href='3hour.php'">3
hour</button>
        <button onclick="location.href='day.php'">1
day</button>
        <button onclick="location.href='week.php'">1
week</button>
        <button onclick="location.href='month.php'">1
month</button>
        <button onclick="location.href='year.php'">1
year</button>
    </center>
    <div id="container" style="min-width: 310px; height:
400px; margin: 0 auto"></div>
    <div id="container2" style="min-width: 310px; height:
400px; margin: 0 auto"></div>
</body>
</html>

```

week.php

```

<!DOCTYPE HTML> <html>
<head>

```

```

        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
        <title>Temperature and GPIO</title>
        <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery
.min.js"></script>
        <style type="text/css">
        ${demo.css}
        </style>
<?php $ret = rrd_fetch
("/home/christos/diploma/28.FF287F2D0400.rrd", array(
"AVERAGE", "-s", "-1w", "-e", "-3600" ));
$start=$ret[start];
$start+=3600;
foreach ( $ret[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        $mydata[]=$key;
    }
    else {
        $mydata[]='null';
    }
}
$myfinal=implode(" , ", $mydata);
$ret2 = rrd_fetch
("/home/christos/diploma/28.FFA98E2C0400.rrd", array(
"AVERAGE", "-s", "-1w", "-e", "-3600" ));
foreach ( $ret2[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        $mydata2[]=$key;
    }
    else {
        $mydata2[]='null';
    }
}
$myfinal2=implode(" , ", $mydata2);
$ret3 = rrd_fetch ("/home/christos/diploma/11.rrd", array(
"AVERAGE", "-s", "-1w", "-e", "-3600" ));
foreach ( $ret3[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata3[]=1;
        }
        else {
            $mydata3[]=0;
        }
    }
    else {
        $mydata3[]='null';
    }
}
$myfinal3=implode(" , ", $mydata3);

```

```

$ret4 = rrd_fetch ( "/home/christos/diploma/13.rrd", array(
"AVERAGE", "-s", "-1w", "-e", "-3600" ) );
foreach ( $ret4[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata4[]=3;
        }
        else {
            $mydata4[]=2;
        }
    }
    else {
        $mydata4[]='null';
    }
}
$myfinal4=implode(" , ",$mydata4);
$ret5 = rrd_fetch ( "/home/christos/diploma/15.rrd", array(
"AVERAGE", "-s", "-1w", "-e", "-3600" ) );
foreach ( $ret5[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata5[]=5;
        }
        else {
            $mydata5[]=4;
        }
    }
    else {
        $mydata5[]='null';
    }
}
$myfinal5=implode(" , ",$mydata5);
$ret6 = rrd_fetch ( "/home/christos/diploma/16.rrd", array(
"AVERAGE", "-s", "-1w", "-e", "-3600" ) );
foreach ( $ret6[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata6[]=7;
        }
        else {
            $mydata6[]=6;
        }
    }
    else {
        $mydata6[]='null';
    }
}
$myfinal6=implode(" , ",$mydata6);
$ret7 = rrd_fetch ( "/home/christos/diploma/18.rrd", array(
"AVERAGE", "-s", "-1w", "-e", "-3600" ) );
foreach ( $ret7[data][sensor] as $time=>$key ) {

```

```

if (trim($key) != "NAN"){
    if ($key>0.5) {
        $mydata7[]=9;
    }
    else {
        $mydata7[]=8;
    }
}
else {
    $mydata7[]='null';
}
}
$myfinal7=implode(" , ", $mydata7); ?>
<script type="text/javascript"> $(function () {
    $('#container').highcharts({
        chart: {
            zoomType: 'x'
        },
        title: {
            text: 'Temperature'
        },
        subtitle: {
            text: '1 Week'
        },
        xAxis: {
            type: 'datetime',
            maxZoom: 12 * 600 * 1000
        },
        yAxis: {
            title: {
                text: 'Temperature (C)'
            },
            minRange: 15
        },
        series: [{
            name: 'Sensor1',
            data: [ <?php echo $myfinal;?> ],
            pointStart: ( <?php echo $start;?> +7200)*1000,
            pointInterval: 12*300*1000
        }, {
            name: ' Sensor2',
            data: [ <?php echo $myfinal2;?> ],
            pointStart: ( <?php echo $start;?> +7200)*1000,
            pointInterval: 12*300*1000
        }
    ]
});
});
$(function () {
    $('#container2').highcharts({
        chart: {
            zoomType: 'x'

```

```

    },
    title: {
        text: 'GPIO'
    },
    subtitle: {
        text: '1 Week'
    },
    xAxis: {
        type: 'datetime',
        maxZoom: 12 * 600 * 1000
    },
    yAxis: {
        title: {
            text: 'GPIO ON/OFF'
        }
    },
    categories:[0,1,0,1,0,1,0,1,0,1]
},
series: [{
    name: 'GPIO 11',
    data: [<?php echo $myfinal3;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 12*300*1000
}, {
    name: 'GPIO 13',
    data: [<?php echo $myfinal4;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 12*300*1000
}, {
    name: 'GPIO 15',
    data: [<?php echo $myfinal5;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 12*300*1000
}, {
    name: 'GPIO 16',
    data: [<?php echo $myfinal6;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 12*300*1000
}, {
    name: 'GPIO 18',
    data: [<?php echo $myfinal7;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 12*300*1000
}
]
});
});
</script>
</head>
<body bgcolor="#D3D3D3">
<script src="./js/highcharts.js"></script>
<script src="./js/modules/exporting.js"></script>

```

```

        <center>
            <button onclick="location.href='hour.php'">1
hour</button>
            <button onclick="location.href='3hour.php'">3
hour</button>
            <button onclick="location.href='day.php'">1
day</button>
            <button onclick="location.href='week.php'">1
week</button>
            <button onclick="location.href='month.php'">1
month</button>
            <button onclick="location.href='year.php'">1
year</button>
        </center>
        <div id="container" style="min-width: 310px; height:
400px; margin: 0 auto"></div>
        <div id="container2" style="min-width: 310px; height:
400px; margin: 0 auto"></div>
    </body>
</html>

```

month.php

```

<!DOCTYPE HTML> <html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
        <title>Temperature and GPIO</title>
        <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery
.min.js"></script>
        <style type="text/css">
            ${demo.css}
        </style>
    <?php $ret = rrd_fetch
( "/home/christos/diploma/28.FF287F2D0400.rrd", array(
"AVERAGE", "-s", "-1m", "-e", "-3600" ) );
$start=$ret[start];
$start+=86400;
foreach ( $ret[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        $mydata[]=$key;
    }
    else {
        $mydata[]='null';
    }
}
$myfinal=implode(" , ", $mydata);

```

```

$ret2 = rrd_fetch
( "/home/christos/diploma/28.FFA98E2C0400.rrd", array(
"AVERAGE", "-s", "-1m", "-e", "-3600" ) );
foreach ( $ret2[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        $mydata2[]=$key;
    }
    else {
        $mydata2[]='null';
    }
}
$myfinal2=implode(" , ", $mydata2);
$ret3 = rrd_fetch ( "/home/christos/diploma/11.rrd", array(
"AVERAGE", "-s", "-1m", "-e", "-3600", "-r", "86400" ) );
foreach ( $ret3[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata3[]=1;
        }
        else {
            $mydata3[]=0;
        }
    }
    else {
        $mydata3[]='null';
    }
}
$myfinal3=implode(" , ", $mydata3);
$ret4 = rrd_fetch ( "/home/christos/diploma/13.rrd", array(
"AVERAGE", "-s", "-1m", "-e", "-3600", "-r", "86400" ) );
foreach ( $ret4[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata4[]=3;
        }
        else {
            $mydata4[]=2;
        }
    }
    else {
        $mydata4[]='null';
    }
}
$myfinal4=implode(" , ", $mydata4);
$ret5 = rrd_fetch ( "/home/christos/diploma/15.rrd", array(
"AVERAGE", "-s", "-1m", "-e", "-3600", "-r", "86400" ) );
foreach ( $ret5[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata5[]=5;
        }
        else {

```

```

        $mydata5[]=4;
    }
}
else {
    $mydata5[]='null';
}
}
$myfinal5=implode(" , ",$mydata5);
$ret6 = rrd_fetch ("/home/christos/diploma/16.rrd", array(
"AVERAGE", "-s", "-1m", "-e", "-3600", "-r", "86400" ));
foreach ( $ret6[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata6[]=7;
        }
        else {
            $mydata6[]=6;
        }
    }
}
else {
    $mydata6[]='null';
}
}
$myfinal6=implode(" , ",$mydata6);
$ret7 = rrd_fetch ("/home/christos/diploma/18.rrd", array(
"AVERAGE", "-s", "-1m", "-e", "-3600", "-r", "86400" ));
foreach ( $ret7[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata7[]=9;
        }
        else {
            $mydata7[]=8;
        }
    }
}
else {
    $mydata7[]='null';
}
}
$myfinal7=implode(" , ",$mydata7); ?>
<script type="text/javascript">
$(function () {
    $('#container').highcharts({
        chart: {
            zoomType: 'x'
        },
        title: {
            text: 'Temperature'
        },
        subtitle: {
            text: '1 Month'
        }
    });
});

```



```

    },
    xAxis: {
      type: 'datetime',
      maxZoom: 12 * 600 * 1000
    },
    yAxis: {
      title: {
        text: 'Temperature (C)'
      },
      minRange: 15
    },
  },
  series: [{
    name: 'Sensor1',
    data: [<?php echo $myfinal;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 24*12*300*1000
  },{
    name: ' Sensor2',
    data: [<?php echo $myfinal2;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 24*12*300*1000
  }
  ]
  });
});
$(function () {
  $('#container2').highcharts({
    chart: {
      zoomType: 'x'
    },
    title: {
      text: 'GPIO'
    },
    subtitle: {
      text: '1 Month'
    },
    xAxis: {
      type: 'datetime',
      maxZoom: 12 * 600 * 1000
    },
    yAxis: {
      title: {
        text: 'GPIO ON/OFF'
      },
      categories:[0,1,0,1,0,1,0,1,0,1]
    },
    series: [{
      name: 'GPIO 11',
      data: [<?php echo $myfinal3;?>],
      pointStart: (<?php echo $start;?> +7200)*1000,
      pointInterval: 24*12*300*1000
    }
  ]
});

```

```

    },{
    name: 'GPIO 13',
    data: [<?php echo $myfinal4;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 24*12*300*1000
    },{
    name: 'GPIO 15',
    data: [<?php echo $myfinal5;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 24*12*300*1000
    },{
    name: 'GPIO 16',
    data: [<?php echo $myfinal6;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 24*12*300*1000
    },{
    name: 'GPIO 18',
    data: [<?php echo $myfinal7;?>],
    pointStart: (<?php echo $start;?> +7200)*1000,
    pointInterval: 24*12*300*1000
    }
    ]
    });
});
</script>
</head>
<body bgcolor="#D3D3D3">
<script src="./js/highcharts.js"></script>
<script src="./js/modules/exporting.js"></script>
<center>
<button onclick="location.href='hour.php'">1
hour</button>
<button onclick="location.href='3hour.php'">3
hour</button>
<button onclick="location.href='day.php'">1
day</button>
<button onclick="location.href='week.php'">1
week</button>
<button onclick="location.href='month.php'">1
month</button>
<button onclick="location.href='year.php'">1
year</button>
</center>
<div id="container" style="min-width: 310px; height:
400px; margin: 0 auto"></div>
<div id="container2" style="min-width: 310px; height:
400px; margin: 0 auto"></div>
</body>
</html>

```

year.php

```
<!DOCTYPE HTML> <html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <title>Temperature and GPIO</title>
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery
.min.js"></script>
    <style type="text/css">
      ${demo.css}
    </style>
  <?php $ret = rrd_fetch
( "/home/christos/diploma/28.FF287F2D0400.rrd", array(
"AVERAGE", "-s", "-1y", "-e", "-86400" ) );
$start=$ret[start];
$start+=86400;
foreach ( $ret[data][sensor] as $time=>$key ) {
  if (trim($key) != "NAN"){
    $mydata[]=$key;
  }
  else {
    $mydata[]='null';
  }
}
$myfinal=implode(" , ", $mydata);
$ret2 = rrd_fetch
( "/home/christos/diploma/28.FFA98E2C0400.rrd", array(
"AVERAGE", "-s", "-1y", "-e", "-86400" ) );
foreach ( $ret2[data][sensor] as $time=>$key ) {
  if (trim($key) != "NAN"){
    $mydata2[]=$key;
  }
  else {
    $mydata2[]='null';
  }
}
$myfinal2=implode(" , ", $mydata2);
$ret3 = rrd_fetch ( "/home/christos/diploma/11.rrd", array(
"AVERAGE", "-s", "-1y", "-e", "-86400" ) );
foreach ( $ret3[data][sensor] as $time=>$key ) {
  if (trim($key) != "NAN"){
    if ($key>0.5) {
      $mydata3[]=1;
    }
    else {
      $mydata3[]=0;
    }
  }
  else {
    $mydata3[]='null';
  }
}
```

```

    }
}
$myfinal3=implode(" , ",$mydata3);
$ret4 = rrd_fetch ("/home/christos/diploma/13.rrd", array(
"AVERAGE", "-s", "-1y", "-e", "-86400" ));
foreach ( $ret4[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata4[]=3;
        }
        else {
            $mydata4[]=2;
        }
    }
    else {
        $mydata4[]='null';
    }
}
$myfinal4=implode(" , ",$mydata4);
$ret5 = rrd_fetch ("/home/christos/diploma/15.rrd", array(
"AVERAGE", "-s", "-1y", "-e", "-86400" ));
foreach ( $ret5[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata5[]=5;
        }
        else {
            $mydata5[]=4;
        }
    }
    else {
        $mydata5[]='null';
    }
}
$myfinal5=implode(" , ",$mydata5);
$ret6 = rrd_fetch ("/home/christos/diploma/16.rrd", array(
"AVERAGE", "-s", "-1y", "-e", "-86400" ));
foreach ( $ret6[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata5[]=5;
        }
        else {
            $mydata5[]=4;
        }
    }
    else {
        $mydata6[]='null';
    }
}
$myfinal6=implode(" , ",$mydata6);

```

```

$ret7 = rrd_fetch ( "/home/christos/diploma/18.rrd", array(
"AVERAGE", "-s", "-1y", "-e", "-86400" ) );
foreach ( $ret7[data][sensor] as $time=>$key ) {
    if (trim($key) != "NAN"){
        if ($key>0.5) {
            $mydata7[]=9;
        }
        else {
            $mydata7[]=8;
        }
    }
    else {
        $mydata7[]='null';
    }
}
$myfinal7=implode(" , ", $mydata7); ?>
<script type="text/javascript"> $(function () {
    $('#container').highcharts({
        chart: {
            zoomType: 'x'
        },
        title: {
            text: 'Temperature'
        },
        subtitle: {
            text: '1 Year'
        },
        xAxis: {
            type: 'datetime',
            maxZoom: 24 * 12* 600 * 1000
        },
        yAxis: {
            title: {
                text: 'Temperature (C)'
            },
            minRange: 15
        },
        series: [{
            name: 'Sensor1',
            data: [<?php echo $myfinal;?>],
            pointStart: (<?php echo $start;?> +7200)*1000,
            pointInterval: 24*12*300*1000
        },{
            name: ' Sensor2',
            data: [<?php echo $myfinal2;?>],
            pointStart: (<?php echo $start;?> +7200)*1000,
            pointInterval: 24*12*300*1000
        }
    ]
});
});

```

```

$(function () {
    $('#container2').highcharts({
        chart: {
            zoomType: 'x'
        },
        title: {
            text: 'GPIO'
        },
        subtitle: {
            text: '1 Year'
        },
        xAxis: {
            type: 'datetime',
            maxZoom: 24 * 12 * 600 * 1000
        },
        yAxis: {
            title: {
                text: 'GPIO ON/OFF'
            }
        },
        categories:[0,1,0,1,0,1,0,1,0,1]
    },
    series: [{
        name: 'GPIO 11',
        data: [<?php echo $myfinal3;?>],
        pointStart: (<?php echo $start;?> +7200)*1000,
        pointInterval: 24*12*300*1000
    },{
        name: 'GPIO 13',
        data: [<?php echo $myfinal4;?>],
        pointStart: (<?php echo $start;?> +7200)*1000,
        pointInterval: 24*12*300*1000
    },{
        name: 'GPIO 15',
        data: [<?php echo $myfinal5;?>],
        pointStart: (<?php echo $start;?> +7200)*1000,
        pointInterval: 24*12*300*1000
    },{
        name: 'GPIO 16',
        data: [<?php echo $myfinal6;?>],
        pointStart: (<?php echo $start;?> +7200)*1000,
        pointInterval: 24*12*300*1000
    },{
        name: 'GPIO 18',
        data: [<?php echo $myfinal7;?>],
        pointStart: (<?php echo $start;?> +7200)*1000,
        pointInterval: 24*12*300*1000
    }
    ]
    });
});
</script>

```

```

</head>
<body bgcolor="#D3D3D3">
    <script src="./js/highcharts.js"></script>
    <script src="./js/modules/exporting.js"></script>
    <center>
        <button onclick="location.href='hour.php'">1
hour</button>
        <button onclick="location.href='3hour.php'">3
hour</button>
        <button onclick="location.href='day.php'">1
day</button>
        <button onclick="location.href='week.php'">1
week</button>
        <button onclick="location.href='month.php'">1
month</button>
        <button onclick="location.href='year.php'">1
year</button>
    </center>
    <div id="container" style="min-width: 310px; height:
400px; margin: 0 auto"></div>
    <div id="container2" style="min-width: 310px; height:
400px; margin: 0 auto"></div>
</body>
</html>

```

Περιεχόμενα φακέλου /home/christos/diploma

job.py

```

import sys;
sys.path.append('/usr/lib/python2.7/dist-packages/')
import mosquitto
import rrdtool

def on_disconnect():
    reconnect()

def on_message(mosq, obj, msg):
    mydata=msg.payload
    mysplitdata=mydata.split(' ')
    if (mysplitdata[1][:1])=='/':
        mynewdata=mysplitdata[1][1:]
    else:
        mynewdata=mysplitdata[1]
    ret = rrdtool.update(mynewdata+'.rrd',mysplitdata[0]+' ':' +
mysplitdata[2]);
    if ret:
        print rrdtool.error()
def on_publish(mosq, obj, mid):
    pass

```

```
cli =
mosquitto.Mosquitto(client_id="Xristos",clean_session=False)
cli.on_message = on_message
cli.on_publish = on_publish
cli.disconnect = on_disconnect
cli.connect("e107222.ddns.net")

cli.subscribe("raspberrypi/sensors",1)
cli.subscribe("raspberrypi/gpio",1)

cli.loop_forever(timeout=1.0, max_packets=1)
```


ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1]http://en.wikipedia.org/wiki/Raspberry_Pi
- [2] <http://www.raspberrypi.org/>
- [3]<http://www.raspberrypi.org/documentation/>
- [4]<http://www.raspberrypi.org/downloads/>
- [5]<http://owfs.org/>
- [6]http://www.sheepwalkelectronics.co.uk/RPI2_software.php
- [7]<http://www.sheepwalkelectronics.co.uk/SWE0a.php>
- [8]http://www.sheepwalkelectronics.co.uk/RPI3_information.php
- [9]<http://en.wikipedia.org/wiki/1-Wire>
- [10]<http://www.sheepwalkelectronics.co.uk/SWE2b.php>
- [11]<http://mosquitto.org/documentation/python/>
- [12]<http://mosquitto.org/>
- [13]<http://mqtt.org/>
- [14]<http://en.wikipedia.org/wiki/MQTT>
- [15]<http://unix2date.com/>
- [16]<http://en.wikipedia.org/wiki/RRDtool>
- [17]<http://oss.oetiker.ch/rrdtool/>
- [18]<http://www.highcharts.com/>
- [19]<http://www.highcharts.com/docs/>
- [20]<http://api.highcharts.com/highcharts>
- [21]<http://www.w3schools.com/php/>
- [22]<http://en.wikipedia.org/wiki/PHP>
- [23]<http://php.net/manual/en/>
- [24]<http://stackoverflow.com/>
- [25]<https://www.python.org/>

[26]<http://www.tutorialspoint.com/python/>

[27]http://en.wikipedia.org/wiki/Home_automation

[28] http://en.wikipedia.org/wiki/Internet_of_Things