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

Internet of Things: Έξυπνος Θερμοστάτης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ραφτοπούλου Μαρία

Επιβλέπων: Συκάς Ευστάθιος
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2016

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

Internet of Things: Έξυπνος Θερμοστάτης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ραφτοπούλου Μαρία

Επιβλέπων: Συκάς Ευστάθιος
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την ……..

.....	.....	.....
Συκάς Ευστάθιος	Θεολόγου Μιχαήλ	Στασινόπουλος Γεώργιος
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2016

.....

ΜΑΡΙΑ ΡΑΦΤΟΠΟΥΛΟΥ

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

Περίληψη

Η ραγδαία ανάπτυξη στην έρευνα για το Internet of Things έχει ένος εύρος από εφαρμογές στην καθημερινή μοντέρνα ζωή και μία από τις πιο σπουδαίες είναι το έξυπνο σπίτι. Πιο συγκεκριμένα, οι έξυπνοι θερμοστάτες έχουν κατασκευαστεί για να προσφέρουν άνεση και για να βελτιστοποιήσουν τους ήδη υπάρχοντες θερμοστάτες.

Σκοπός της παρούσας διπλωματικής εργάσιας είναι η κατασκευή ενός έξυπνου θερμοστάτη και για την υλοποίησή του θα χρησιμοποιηθεί ένα Raspberry Pi το οποίο θα επιτρέπει στο χρήστη να παρακολουθεί και να ελέγχει τη θερμοκρασία του σπιτιού του μέσω ενός γραφικού περιβάλλοντος και απομακρυσμένα.

Αρχικά, κατασκευάστηκε ένα θερμόμετρο με τη βοήθεια ενός αισθητήρα θερμοκρασίας και αναπτύχθηκε μία διαδικτυακή εφαρμογή όπου ο χρήστης μπορεί να παρακολουθεί διάφορα στατιστικά και να εισάγει κάποιο εβδομαδιαίο πρόγραμμα με τις επιθυμητές θερμοκρασίες. Στη συνέχεια, λόγω των υψηλών θερμοκρασιών που υπάρχουν στην Αθήνα, αποφασίστηκε ότι η συνέχεια θα γίνει σε προσομοιώσεις.

Το επόμενο βήμα, ήταν να μελετηθούν διάφορα θερμικά μοντέλα που προσομοιώνουν κάποιο κτίριο και η μοντελοποίηση ενός απλού θερμοστάτη on-off και ενός θερμοστάτη που χρησιμοποιεί ελεγκτή pid. Οι διάφορες προσομοιώσεις που έγιναν για τους δύο θερμοστάτες στα διάφορα μοντέλα έδηξαν το πιο κατάλληλο μοντέλο.

Έπειτα, αναπτύχθηκε κάποιος έξυπνος αλγόριθμος και προσομοιώθηκε στο μοντέλο που είχε επιλεγεί πιο πριν. Τέλος, έγιναν διάφορες συγκρίσεις ανάμεσα στους διάφορους θερμοστάτες που μοντελοποιήθηκαν και αναφέρθηκε η βελτιστοποίηση που προσφέρει ο έξυπνος θερμοστάτης.

Λέξεις Κλειδιά

Internet of Things, Έξυπνο Σπίτι, Έξυπνος Θερμοστάτης, Raspberry Pi, Γραφικό περιβάλλον χρήστη, αισθητήρας θερμοκρασίας, θερμόμετρο, on-off θερμοστάτης, PID ελεγκτής, Μοντέλα για τη θερμική δυναμική κτιρίων, RRDTool, OpenWeatherMaps, HighCharts, Bootstrap, No-ip

Abstract

The rapidly growing research about Internet of Things has a range of applications in the everyday modern life but one of its greatest is Smart Home. Specifically, smart thermostats have been developed in order to offer comfort and to optimize the current thermostats.

The objective of this thesis is the development of a smart thermostat and for its implementation a Raspberry Pi is going to be used which will allow the user to monitor and control the temperature of his or her house via a graphical user interface (GUI) remotely.

Firstly, a thermometer was implemented with the help of a temperature sensor and a web application was developed which will let the user to monitor a range of statistics and to provide a weekly schedule of the wanted temperatures. Then, due to high temperatures that are observed in Athens, it was decided that the project will continue in simulations.

The next step, was to study different thermal models which describe buildings and to model a simple on-off thermostat and a thermostat which uses a PID controller. The simulations that were made for the two thermostats on the models, indicated the model the most appropriate.

Moreover, a smart algorithm has been developed and it was simulated on the model which was chosen before. Finally, a comparison was made between the thermostats that were modelled and then the optimization that the smart thermostat offers, was reported.

Keywords

Internet of Things, Smart Home, Smart Thermostats, Raspberry Pi, Graphical User Interface, temperature sensor, thermometer, on-off thermostat, PID controller, Models for the Heat Dynamics of buildings, RRDTool, OpenWeatherMaps, HighCharts, Bootstrap, No-ip

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω τις θερμότερες ευχαριστίες μου στον καθηγητή της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών και επιβλέποντα της συγκεκριμένης εργασίας κ. Ευστάθιο Συκά για τη δυνατότητα που μου προσέφερε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα. Επίσης, θα ήθελα να τον ευχαριστήσω για τον πολύτιμο χρόνο που μου έχει αφιερώσει και για την καθοδήγηση και τις πληροφορίες που μου έχει δώσει. Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου για τη συνεχή υποστήριξη τους.

Πίνακας Περιεχομένων

Πίνακας Εικόνων

Πίνακας Παραστάσεων

ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ

Η σύγχρονη ζωή είναι επηρεασμένη από όλα τα τεχνολογικά επιτεύγματα και μέρα με τη μέρα παρατηρείται η ανάπτυξη νέων τεχνολογιών με βασικό στόχο την απλοποίηση της καθημερινής ζωής. Ξεκινώντας από τις πιο απλές και βασικές ανάγκες του σύγχρονου ανθρώπου και καταλήγοντας σε πολυτελείς ανέσεις, παρατηρούμε ότι διατίθενται στην αγορά μια μεγάλη γκάμα από προϊόντα. Κάποια παραδείγματα βασικών αναγκών είναι η τηλεόραση, ο ηλεκτρονικός υπολογιστής ή ακόμα και ένα ψυγείο ενώ συνεχίζοντας σε πιο σύγχρονες ανάγκες είναι τα smart meters και οι έξυπνοι θερμοστάτες.

Πιο κάτω θα αναφερθούμε στην ανάγκη για κατασκευή συσκευών όπως των πιο πάνω και την τάση που υπάρχει σε αυτές τις τεχνολογίες σήμερα.

1.1. Πρόλογος

Ο σύγχρονος άνθρωπος έχει αυξημένες ανάγκες λόγω περιορισμένου ελευθέρου χρόνου και αυξημένων υποχρεώσεων και για αυτό το λόγο κάθε συσκευή που μπορεί να τον βοηθήσει στο να βελτιστοποιήσει την καθημερινότητά του είναι προτέρημα. Βασιζόμενοι σε αυτή την ιδέα, έχουν δημιουργηθεί διάφορες συσκευές και εφαρμογές που με την πάροδο του χρόνου διεισδύουν όλο και πιο πολύ στην καθημερινή ζωή μέχρι που θεωρούνται αναγκαίες. Ένα αξιόλογο παράδειγμα τέτοιας συσκευής είναι τα smart meter[1] που χρησιμοποιούνται από τις αρχές παροχής ενέργειας για να κοστολογήσουν την ενέργεια ανάλογα με την ώρα της μέρας και την περίοδο του χρόνου και έτσι ο καταναλωτής να πληρώνει ακριβώς το ποσό που του αναλογεί. Η συγκεκριμένη συσκευή εμφανίζει ραγδαία ανάπτυξη τα τελευταία 10 χρόνια σε μεγάλες χώρες του κόσμου και ένα μέρος αυτού οφείλεται στο ότι ο χρήστης έχει τη δυνατότητα να ενημερώνεται για την κατανάλωση που έχει κάνει και για το κόστος που θα επιβαρυνθεί με αποτέλεσμα να μπορεί να προσπαθήσει να ελαχιστοποιήσει την κατανάλωσή του και αντίστοιχα το κόστος όσο θέλει. Το τελικό αποτέλεσμα άρα επωφελή και διευκολύνει τον καταναλωτή

Κατανοώντας λοιπόν ότι είναι αναγκαία η κατασκευή τέτοιων τεχνολογικών συσκευών και λαμβάνοντας υπόψιν ότι σε πολυάριθμες χώρες του κόσμου χρησιμοποιούνται θερμοστάτες για να διατηρούν το σπίτι σε επιθυμητές θερμοκρασίες, αναπτύχθηκαν οι έξυπνοι θερμοστάτες. Η διαφορά ενός έξυπνου θερμοστάτη από ένα απλό είναι ότι δίνει στον χρήστη μια περεταίρω ελευθερία στο να παρατηρεί δεδομένα αλλά κυρίως ότι διατηρεί την επιθυμητή θερμοκρασία του σπιτιού πιο σταθερή και με την πάροδο του χρόνου, έχει την δυνατότητα τα μαθαίνει το πρόγραμμα του χρήστη και να κάνει κάποιες βελτιστοποιήσεις από μόνος του, ώστε να γίνεται ακόμα πιο ακριβές στις θερμοκρασίες που επιθυμεί ο χρήστης κάθε στιγμή.

1.2. Έξυπνοι Θερμοστάτες

Στην αγορά σήμερα υπάρχει μια πολυποίκιλη γκάμα από θερμοστάτες που μπορεί να προμηθευτεί ο καθ' ένας για το σπίτι του. Κάθε θερμοστάτης έχει διαφορετικά χαρακτηριστικά και δυνατότητες και αναλόγως των δυνατοτήτων υπάρχει και το αντίστοιχο κόστος. Ένας από τους πιο έξυπνους θερμοστάτες που κυκλοφορούν στην αγορά και προσφέρει πολλές δυνατότητες στο χρήστη είναι ο Nest της εταιρίας Google[2]. Ωστόσο αν και πολλές άλλες εταιρίες διαθέτουν αξιόλογους θερμοστάτες, όλοι οι θερμοστάτες κυμαίνονται σε αρκετά υψηλές τιμές.

Η χρήση των έξυπνων θερμοστατών, λόγω των δυνατοτήτων που προσφέρουν, έχει αυξηθεί ραγδαία τόσο στη Βόρειο Αμερική όσο και στην Ευρώπη. Κατά τη διάρκεια του έτους 2015, παρατηρήθηκε αύξηση κατά 81% στον συγκεκριμένο τομέα ενώ πιο συγκεκριμένα, η αύξηση στη Βόρειο Αμερική ήταν 78% ενώ στην Ευρώπη έφτασε στι 90%. Παρόλο των μεγάλων ποσοστών ανάπτυξης, ο αριθμών των σπιτιών που χρησιμοποιούν έξυπνους θερμοστάτες είναι ακόμα πολύ μικρός και έτσι προβλέπεται ότι για την επόμενη πενταετία θα υπάρχει αύξηση του ποσοστού κατά 54.5% ανά έτος με αποτέλεσμα μέχρι το 2020 να έχουν εγκατασταθεί 51.1 εκατομμύρια έξυπνοι θερμοστάτες.[3]

Πιο κάτω υπάρχουν διάφορα χαρακτηριστικά που μπορεί να έχει ένας έξυπνος θερμοστάτης:

- Λειτουργία κλιματισμού και θέρμανσης
- Αποδοχή ημερήσιου ή εβδομαδιαίου προγράμματος χρήστη
- Εκμάθηση και βελτιστοποίηση προγράμματος χρήστη
- Ελαχιστοποίηση κατανάλωσης ενέργειας με ένδειξη κόστους κατανάλωσης
- Χρήση Wi-Fi για ενσωμάτωση στο προσωπικό δίκτυο
- Χρήση web-app ή mobile-app για απομακρυσμένη χρήση
- Δυνατότητα αλλαγής θερμοκρασίας για τη συγκεκριμένη στιγμή
- Ενημέρωση για πρόβλεψη καιρού
- Δυνατότητα πλοήγησης σε παλιότερα δεδομένα
- Υπολογισμός κόστους κατανάλωσης ενέργειας λόγω κλιματισμού
- Χρήση κωδικών πρόσβασης για ασφάλεια
- Ενδείξεις τυχών σφαλμάτων
- Ωραίος σχεδιασμός
- Εύκολος στη χρήση

Ένας έξυπνος θερμοστάτης λοιπόν είναι ένας συνδυασμός αισθητήρων και έξυπνων αλγορίθμων, οι οποίοι αλληλοεπιδρούν μεταξύ τους και μαζί με την πρόσβαση στο διαδίκτυο, μπορούν να προσφέρουν μεγάλη άνεση στη καθημερινή ζωή.

Ένα μειονέκτημα των έξυπνων θερμοστατών είναι ότι μπορούν να υποβαθμίσουν την ασφάλεια του σπιτιού αφού μπορούν εύκολα να δεχτούν επιθέσεις, όσον αφορά την ασφάλεια τους (hacking). Αποτέλεσμα των συγκεκριμένων επιθέσεων είναι η κλοπή του προγράμματος χρήστη και άρα να μπορεί να γνωστοποιηθεί το πότε βρίσκεται ή όχι στο σπίτι ο ιδιοκτήτης.

1.3. Internet of Things (IoT)

Η διασύνδεση διαφόρων φυσικών αντικειμένων με τη βοήθεια ηλεκτρονικής, λογισμικού και αισθητήρων αναφέρεται ως Internet of Things (IoT)[4]. Κάθε συσκευή και μπορεί να λειτουργήσει ανεξάρτητα με το δικό της υπολογιστικό σύστημα αλλά ταυτόχρονα με τη βοήθεια του διαδικτύου, μπορεί να επικοινωνήσει με οποιαδήποτε άλλη συσκευή χρειαστεί, οπουδήποτε και αν βρίσκεται αυτή. Το IoT έχει αρχίσει να αυξάνεται ραγδαία και υπολογίζεται ότι μέχρι το 2020 θα απαρτίζεται από 50 δισεκατομμύρια αντικείμενα.

Εικόνα 1.1 - Internet of Things

Το IoT, μπορεί να βρει εφαρμογές σε διάφορους τομείς, από τον τομέα της υγείας μέχρι και τον τομέα της μετακίνησης, του περιβάλλοντος και του σπιτιού. Ένα παράδειγμα στο τομέα της μετακίνησης είναι τα smart traffic signals[5] που έχουν σαν στόχο να ελαχιστοποιήσουν την κυκλοφοριακή συμφόρηση. Με τη βοήθεια ενός κεντρικού υπολογιστή, κάθε ψηφιακός σηματοδότης λαμβάνει σήματα με πληροφορίες όπως τη διεύθυνση και τη φορά κάθε οχήματος και με τη χρήση δεκτών οπτικών ινών βίντεο γίνονται οι διάφορες αλλαγές. Τα πρώτα smart traffic signals έχουν υλοποιηθεί στο Pittsburgh των ΗΠΑ και έχουν θετικά αποτελέσματα.

Η εφαρμογή του IoT στον οικιακό τομέα, αναφέρεται ως smart homes, αφού δίνει τη δυνατότητα σε διάφορες οικιακές συσκευές να ελέγχονται απομακρυσμένα και να ανταλλάζουν πληροφορίες. Οι έξυπνοι θερμοστάτες λοιπόν είναι μέρος των smart homes και κάνουν χρήση της έννοιας IoT.

1.4. Σκοπός

Σκοπός αυτής της εργασίας είναι η υλοποίηση ενός έξυπνου θερμοστάτη, ο οποίος θα έχει χαμηλό κόστος και θα προσφέρει αρκετά από τα χαρακτηριστικά των εμπορικών έξυπνων θερμοστατών. Πιο συγκεκριμένα τα χαρακτηριστικά του θα είναι τα εξής:

- Χαμηλό κόστος στο υλικό (Υπολογιστικό σύστημα και αισθητήρες)
- Λειτουργία Θέρμανσης
- Εβδομαδιαίο πρόγραμμα χρήστη: Ο χρήστης θα μπορεί να εισαγάγει την επιθυμητή θερμοκρασία που θα θέλει να έχει το σπίτι του σε συγκεκριμένα χρονικά διαστήματα. Κάθε μέρα της εβδομάδας μπορεί να έχει διαφορετικό πρόγραμμα ή διαφορετικά σε αριθμό χρονικά διαστήματα, ώστε ο χρήστης να έχει πλήρη έλεγχο και να προσαρμόζεται πιο εύκολα στις επιθυμίες του.
- Εκμάθηση προγράμματος χρήστη: Ανάπτυξη αλγορίθμων που θα προσαρμόζουν το χρόνο λειτουργίας της θέρμανσης ώστε να πετυχαίνετε πιο γρήγορα η επιθυμητή θερμοκρασία.
- Σταθερή θερμοκρασία: Η επιθυμητή θερμοκρασία δεν θα δέχεται διακύμανσης πέραν των $\pm 0.5℃$.
- Σύνδεση στο προσωπικό δίκτυο του σπιτιού.
- Ανάπτυξη διαδικτυακής εφαρμογής για απομακρυσμένο έλεγχο και παρατήρηση των δεδομένων.
- Υποστήριξη δελτίου πρόβλεψης καιρού για τις επόμενες 5 ημέρες.
- Χρήση κωδικών πρόσβασης για λόγους ασφάλειας
- Εύκολος στη χρήση
- Μικρός σε μέγεθος

Για να υλοποιηθούν τα πιο πάνω, η εργασία χωρίσθηκε σε 4 διακριτά μέρη τα οποία αναλύονται πιο κάτω:

- Στο πρώτο μέρος, έγινε η επιλογή του κατάλληλου υλικού(υπολογιστικού συστήματος, αισθητήρων, κλπ), ώστε να τηρεί τις προϋποθέσεις. Αφού τελείωσε η επιλογή, ακολούθησε η συνδεσμολογία και τέλος ο έλεγχος της ορθής λειτουργίας, ότι δηλαδή αποθηκεύονται με επιτυχία σε βάσεις δεδομένων τόσο οι θερμοκρασίες του δωματίου, όσο και οι θερμοκρασίες του περιβάλλοντος.
- Στο δεύτερο μέρος, δημιουργήθηκε ένας web server ο οποίος θα φιλοξενεί τη διαδικτυακή εφαρμογή. Μετά αναπτύχθηκε και η εφαρμογή κατάλληλα ώστε να πληρεί τα χαρακτηριστικά που τέθηκαν πιο πάνω, χρησιμοποιώντας μία βάση

δεδομένων για την αποθήκευση του προγράμματος χρήστη και διάφορα διαγράμματα για την παρουσίαση κάποιων πληροφοριών.

- Στο τρίτο μέρος, μελετήθηκαν διάφοροι ελεγκτές ώστε να βρεθεί ο κατάλληλος που θα μπορούσε να χρησιμοποιηθεί για να σταθεροποιεί γρήγορα την θερμοκρασία και τελικά να μην δίνει μεγάλες αποκλίσεις από την επιθυμητή. Επίσης, λόγω των υψηλών θερμοκρασιών του καλοκαιριού στην Αθήνα και λόγω του ότι ο θερμοστάτης θα πραγματοποιεί μόνο λειτουργία θέρμανσης, δεν θα ήταν δυνατό να πραγματοποιηθούν πειράματα σε πραγματικά δεδομένα. Για αυτό το λόγο, μελετήθηκαν διάφορα θερμικά μοντέλα σπιτιού και έγιναν πειράματα σε αυτά τόσο για ένα απλό θερμοστάτη on-off, όσο και για θερμοστάτη που χρησιμοποιεί ελεγκτή pid. Μετά το πέρας λοιπόν αυτών των πειραμάτων, ήμασταν σε θέση να έχουμε ένα θερμοστάτη που λειτουργεί σωστά και με ακρίβεια.
- Στο τέταρτο μέρος, αναπτύχθηκε αλγόριθμος ο οποίος μαθαίνει το πρόγραμμα του χρήστη με την πάροδο του χρόνου και έτσι μπορεί να βελτιστοποιήσει το θερμοστάτη. Λόγω αυτής του της γνώσης, ο θερμοστάτης είναι σε θέση να γνωρίζει πόσο πιο γρήγορα πρέπει να ανοίξει ή να κλείσει τη θέρμανση στην αρχή του κάθε διαστήματος που υπάρχει επιθυμητή θερμοκρασία αντίστοιχα. Έτσι, πετυχαίνει την επιθυμητή θερμοκρασία με μεγαλύτερη ακρίβεια και χωρίς καθυστερήσεις . Για τη λειτουργία του αλγορίθμου έγινε προσομοίωση χρησιμοποιώντας δεδομένα από κάποιες κρύες μέρες του χειμώνα.

ΚΕΦΑΛΑΙΟ 2. ΤΕΧΝΟΓΝΩΣΙΑ

Σε αυτό το κεφάλαιο αναφέρονται οι διάφορες τεχνολογίες οι οποίες μελετήθηκαν καθ' όλη τη διάρκεια της εργασίας. Αρχικά αναφέρονται τα διάφορα υλικά που θα μπορούσαν να χρησιμοποιηθούν και μετά περιγράφεται ο απλός θερμοστάτης και οι διάφοροι ελεγκτές που θα μπορούσαν να υλοποιηθούν για να βελτιστοποιούν τον θερμοστάτη. Στη συνέχεια αναλύονται τα θερμικά μοντέλα σπιτιού που μελετήθηκαν και τέλος γίνεται μία μικρή αναφορά για την ανάγκη ύπαρξης γραφικού περιβάλλοντος.

2.1. Υπολογιστικά Συστήματα

Για την υλοποίηση της εργασίας χρειάζεται να χρησιμοποιηθεί ένα υπολογιστικό σύστημα όπου θα αποθηκεύονται τα δεδομένα, θα δέχεται τις εισόδους και θα λειτουργεί σαν web server για την ανάπτυξη της web εφαρμογής. Τα δύο πιο διαδεδομένα υπολογιστικά συστήματα για ανάπτυξη διαφόρων εφαρμογών είναι το Arduino και το Raspberry Pi.

Το Arduino[6] είναι ένας single-board μικροελεκτής με διάφορες εισόδους και εξόδους και μνήμη, ο οποίος μπορεί να προγραμματιστεί. Υποστηρίζει I^2C serial bus και pins εισόδων εξόδων, τα οποία του επιτρέπουν να συνδέεται με διάφορα άλλα επιπρόσθετα εξαρτήματα και να εκτελούν λειτουργίες παράλληλα. Ακόμα, το προγραμματιστικό του περιβάλλον είναι απλό και έτσι προτιμάται από αρχάριους προγραμματιστές. Τέλος το λογισμικό του είναι ανοιχτού κώδικα και μπορούν να βρεθούν ήδη έτοιμες εργασίες ή βοήθεια από την κοινότητα.

Το Raspberry Pi[7] είναι ένας single-board υπολογιστής, ο οποίος έχει ενσωματωμένη μνήμη, επεξεργαστή, εισόδους/εξόδους και διάφορα άλλα χαρακτηριστικά που του επιτρέπουν να λειτουργεί σαν ολοκληρωμένος υπολογιστής και χρησιμοποιείται η τεχνολογία SoC(System on Chip), δηλαδή όλα τα απαραίτητα στοιχεία είναι ενσωματωμένα στο ίδιο chip. Όπως και στο Arduino, το λογισμικό είναι ανοιχτού κώδικα και υπάρχει μεγάλη υποστήριξη από την κοινότητα. Η διαφοροποίηση ανάμεσα στα δύο, είναι ότι το Raspberry Pi έχει λειτουργικό σύστημα και μεγαλύτερη υπολογιστική ισχύ που του επιτρέπει να υλοποιεί μεγαλύτερης πολυπλοκότητας projects καθώς μπορεί να συνδεθεί σε οθόνη και σε πληκτρολόγιο.

Λόγω της πολυπλοκότητας της υλοποίησης ενός έξυπνου θερμοστάτη, είναι καταλληλότερη η χρήση του Raspberry Pi. Στην αγορά υπάρχουν διαθέσιμα διάφορα μοντέλα, τα οποία διαφοροποιούνται στα χαρακτηριστικά λειτουργίας τους. Κάποια από αυτά τα χαρακτηριστικά είναι τα πιο κάτω:

- Ταχύτητα και Πυρήνες του επεξεργαστή
- Μέγεθος μνήμης (SDRAM)
- Αριθμός USB ports
- Έξοδοι βίντεο
- Μέγεθος μνήμης για αποθήκευση δεδομένων
- Τρόπος σύνδεσης στο διαδίκτυο
- Αριθμός GPIO (General Purpose Input Output)
- Κόστος αγοράς

Συγκρίνοντας τα διάφορα μοντέλα και τα χαρακτηριστικά τους, επιλέγηκε για την εργασία το Raspberry Pi 2.

2.2. Αισθητήρες Θερμοκρασίας

Στην αγορά υπάρχει μια πολύ μεγάλη γκάμα από αισθητήρες θερμοκρασίας που μπορούν να χρησιμοποιηθούν. Κάθε αισθητήρας χαρακτηρίζεται από το εύρος θερμοκρασιών που μπορεί να μετρήσει, από την ακρίβεια που προσφέρει και από το είδος της εξόδου. Τα δύο είδη εξόδων που υπάρχουν είναι η αναλογική και η ψηφιακή.

Στη παρούσα εργασία, θα χρησιμοποιηθεί το Raspberry Pi 2 το οποίο δεν έχει ενσωματωμένο Analog to Digital Convertor (ADC) και για αυτό το λόγο θα χρησιμοποιηθεί κάποιος αισθητήρας που παράγει ψηφιακή έξοδο. Ο αισθητήρας που επιλέγηκε τελικά είναι ο DS18B20 της εταιρίας Dallas Semiconductor, ο οποίος χρησιμοποιεί και τεχνολογία 1-wire.

2.2.1. 1-wire

Το 1-wire[8] είναι ένα σύστημα διαύλου επικοινωνίας συσκευών το οποίο έχει σχεδιαστεί από την εταιρία Dallas Semiconductor Corp και προσφέρει χαμηλής ταχύτητας δεδομένα, σήμανση και εξουσία πάνω από μόνο ένα σήμα, με αποτέλεσμα να λειτουργεί σαν το I2C αλλά για χαμηλότερο ρυθμό δεδομένων και μεγαλύτερο εύρος. Το κυριότερο χαρακτηριστικό της συγκεκριμένης τεχνολογίας είναι η δυνατότητα χρήσης μόνο των καλωδίων των δεδομένων και της γείωσης. Για να επιτύχει αυτή τη δυνατότητα, χρησιμοποιείται ένας πυκνωτής 800pF για να αποθηκεύει το φορτίο και για να τροφοδοτεί τη συσκευή σε περιόδους που η γραμμή δεδομένων είναι ανενεργή.

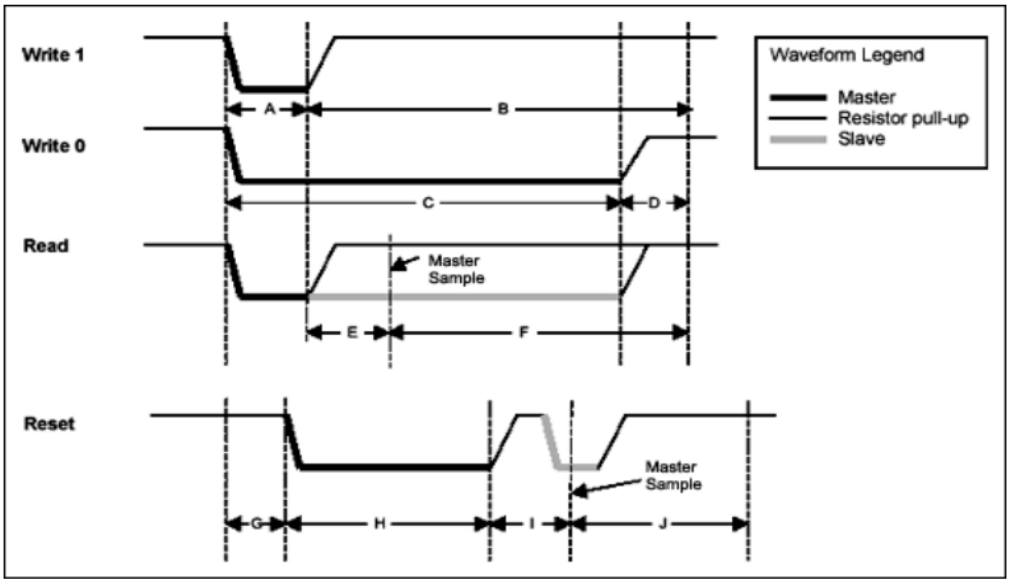
Ανάλογα με την λειτουργία, οι συσκευές 1-wire είναι διαθέσιμες σαν ένα μεμονωμένο εξάρτημα ή σε φορητή μορφή που ονομάζεται iButton και μοιάζει με μπαταρία ρολογιού. Ακόμα υπάρχουν και πιο πολύπλοκες συσκευές που χρησιμοποιούν δίαυλο επικοινωνίας 1-wire. Η διασύνδεση διαφόρων συσκευών 1-wire, μπορεί να δημιουργήσει κάποιο σύστημα αισθητήρων, όπου η κάθε συσκευή θα περιλαμβάνει όλη την λογική που απαιτείται για να χρησιμοποιήσει τον 1-wire δίαυλο. Όλες οι συσκευές μπορούν να συνδεθούν στον ηλεκτρονικό υπολογιστή μέσω ενός μετατροπέα διαύλου ή απευθείας σε μικροελεκτές. Το δίκτυο που δημιουργείται από αυτές τις συσκευές και χρησιμοποιεί την μία σαν κύρια (master) συσκευή, ονομάζεται MicroLan.

Σε κάθε MicroLan ο master είναι είτε ο ηλεκτρονικός υπολογιστής είτε ο μικροελεκτής και είναι ο υπεύθυνος για τη δραστηριότητα στο δίαυλο, όπου μπορούν να συνδεθούν πολλές συσκευές. Για την ταυτοποίηση της κάθε συσκευής χρησιμοποιείται ένας 64-bit σειριακός αριθμός και μπορούν να ανιχνεύονται έως και 75 συσκευές ανά δευτερόλεπτο. Ακόμα, κάθε συσκευή πρέπει να χρησιμοποιεί μια pullup αντίσταση, για να ανεβάζει την τάση στο δίαυλο στα 3 ή 5Volts ή και για να της παρέχει ισχύ.

Για να αρχίσει η διαδικασία επικοινωνίας, στέλνει ο master ένα παλμό reset στα 0Volts διάρκειας 480μs προς όλους τους slaves και μετά όσοι slaves θέλουν να δείξουν την παρουσία τους, στέλνουν ένα παλμό στα 0Volts για 60μs. Στην περίπτωση που ο master θέλει να στείλει «1», χρησιμοποιεί ένα παλμό διάρκειας 1-15μs στα 0Volts, ενώ όταν θέλει να στείλει «0» χρησιμοποιεί παλμό διάρκειας 60μs στα 0Volts. Οι slaves αντιλαμβάνονται την πτώση του παλμού μετά από 30μs στα 0Volts. Για την περίπτωση που ο master

λαμβάνει δεδομένα, στέλνει 1-15μs παλμό στα 0Volts για να ξεκινήσει το κάθε bit. Αν το αποστελλόμενο bit είναι «1» τότε ο master δεν κάνει τίποτα και ο δίαυλος τίθεται σε τάση ενώ αν είναι «0» η γραμμή δεδομένων τίθεται στα 0Volts για 60μs.

Πιο κάτω φαίνονται παραδείγματα των πιο πάνω παλμών:



Letter	A	B	C	D	E	F	G	H	I	J
Time(μs)	6	64	60	10	9	55	0	480	70	410

Εικόνα 2.1 – Παλμοί 1-wie

2.3. Θερμοστάτες

Ο θερμοστάτης[9] είναι μια συσκευή ανίχνευσης θερμοκρασίας κάποιου συστήματος και έχει ως σκοπό τη διατήρηση του συστήματος σε μια συγκεκριμένη θερμοκρασία, το setpoint. Για να πετύχει το σκοπό του ο θερμοστάτης ανοιγοκλείνει κάποια συσκευή θέρμανσης ή ψύξης ή ρυθμίζει τη ροή θερμότητας. Οι θερμοστάτες έχουν ευρεία χρήση και μπορούν να βρεθούν σε συστήματα θέρμανσης και κλιματισμού, σε φούρνους κουζίνας και στο ψυγείο.

Σε ένα θερμοστάτη, υπάρχουν ενσωματωμένοι αισθητήρες για την μέτρηση της ακριβούς θερμοκρασίας για να μπορεί να γίνει η σύγκριση με την επιθυμητή. Ωστόσο, η χρήση της θέρμανσης ή του κλιματισμού δεν γίνεται αναλογικά με την διαφορά των δύο θερμοκρασιών αλλά λειτουργεί με πλήρες φορτίο μέχρι να πετύχει την επιθυμητή θερμοκρασία και μετά σταματά να λειτουργεί. Ακόμα, πολλές φορές υπάρχει όριο στη συχνότητα αλλαγής κατάστασης on/off των συσκευών θέρμανσης και κλιματισμού που ελέγχουν, ούτως ώστε να μην προκληθεί κάποια ζημιά.

Για την ανίχνευση της θερμοκρασίας, κάθε θερμοστάτης μπορεί να χρησιμοποιεί διαφορετικό αισθητήρα. Κάποιες τεχνολογίες αισθητήρων που χρησιμοποιούνται σήμερα είναι οι διμεταλλικοί μηχανικοί ή ηλεκτρικοί αισθητήρες, οι διαστελλόμενες παλέτες

κεριού, τα ηλεκτρονικά θερμίστορ και οι συσκευές ημιαγωγών και τα ηλεκτρικά θερμοζεύγη.

Οι διμεταλλικοί μηχανικοί θερμοστάτες χρησιμοποιούνται συνήθως σε συστήματα που αφορούν πρόσημο νερό και κεντρικά συστήματα θέρμανσης που χρησιμοποιούν ατμό. Με καθαρά μηχανικό τρόπο, ανιχνεύεται ο ατμός ή το ζεστό νερό του καλοριφέρ και ρυθμίζεται κατάλληλα η ροή.

Οι μηχανικοί θερμοστάτες παλέτας κεριού, χρησιμοποιούνται στην αυτοκινητοβιομηχανία για να διατηρήσουν σταθερή τη θερμοκρασία της μηχανής. Μέσα σε ένα θάλαμο υπάρχει μία παλέτα κεριού η οποία διαστέλλεται και λιώνει σε μία συγκεκριμένη θερμοκρασία. Όταν αυτή η θερμοκρασία ξεπεραστεί, αυτή η διαστολή του θαλάμου προκαλεί κίνηση μιας ράβδου που ανοίγει μια βαλβίδα και ελέγχει το κύκλωμα κλιματισμού της μηχανής. Επίσης, οι συγκεκριμένοι θερμοστάτες χρησιμοποιούνται σε θερμοστατικές βαλβίδες μίξης, οι οποίες είναι υπεύθυνες για τον έλεγχο μίξης του ζεστού και κρύου νερού.

Οι διμεταλλικοί ηλεκτρικοί θερμοστάτες χρησιμοποιούνται για συστήματα κεντρικού κλιματισμού με νερό και ατμό. Χρησιμοποιώντας τη διαστολή 2 μετάλλων μετρούν τη θερμοκρασία του αέρα και αποφασίζουν αν το σύστημα θα πρέπει να είναι ανοιχτό ή κλειστό. Τυπικά, το σύστημα είναι σε κατάσταση λειτουργίας όταν η θερμοκρασία αέρα από τον αισθητήρα είναι μικρότερη της επιθυμητής, ενώ το σύστημα σταματάει να λειτουργεί όταν ξεπεραστεί η επιθυμητή θερμοκρασία. Ωστόσο, το συγκεκριμένο σύστημα λειτουργεί με υστέρηση κάποιων βαθμών ώστε να μην γίνεται υπερβολικό ανοιγοκλείσιμο το συστήματος. Πιο σύγχρονοι θερμοστάτες, αντικατέστησαν τα δύο μέταλλα που χρησιμοποιούνταν για την ανίχνευση θερμοκρασίας αέρα με ηλεκτρονικούς αισθητήρες.

Πιο σύγχρονοι θερμοστάτες είναι οι ψηφιακοί, οι οποίοι χρησιμοποιούν θερμίστορ ή συσκευές ημιαγωγών για την μέτρηση της θερμοκρασίας. Επίσης παρέχουν συνήθως μια οθόνη όπου υπάρχει ένδειξη της θερμοκρασίας δωματίου και διαφόρων άλλων ρυθμίσεων που μπορεί να ελέγξει ο χρήστης. Η συνεχής ανάπτυξη των θερμοστατών έχει επιφέρει την κατασκευή προγραμματιζόμενων θερμοστατών όπου ο χρήστης μπορεί να εισάγει κάποιους κανόνες για την λειτουργεία της θέρμανσης ή του κλιματισμού.

2.4. Θεωρία Ελέγχου

Ο ελεγκτής[10] (Cotrollers) είναι μια συσκευή σε μορφή μικροεπεξεργαστή ή υπολογιστή η οποία, παρακολουθεί και αλλοιώνει φυσικά τις συνθήκες λειτουργίας ενός δυναμικού συστήματος. Το σύστημα δέχεται κάποιες εισόδους, input variables, οι οποίες επηρεάζουν την τρέχουσα κατάσταση λειτουργίας αφού μεταβάλλουν κάποιες άλλες μεταβλητές του συστήματος, τις output variables.

Τα συστήματα μπορούν να χωριστούν σε δύο κατηγορίες: Ανοικτού βρόχου (Open-Loop) και Κλειστού βρόχου (Closed-Loop).

Τα συστήματα ανοικτού βρόχου χαρακτηρίζονται από την μη ανατροφοδότηση δεδομένων πίσω στον ελεγκτή μετά την επίδρασή του. Δηλαδή, μια μεταβλητή εισόδου δίνεται στον ελεγκτή και αυτός αποφασίζει με ποιό τρόπο θα επηρεάσει το σύστημα. Η μεταβολή αυτή

του συστήματος που προκάλεσε ο ελεγκτής, αλλάζει την μεταβλητή που θέλουμε να χειριστούμε. Παράδειγμα ενός τέτοιου συστήματος μπορεί να είναι η παρακολούθηση της πληρότητας ενός δωματίου.

Τα συστήματα κλειστού βρόχου παρακολουθούν συνεχώς την μεταβλητή που θέλουν να ελέγξουν και την συγκρίνουν με τη μεταβλητή αναφοράς (reference variable). Το αποτέλεσμα της σύγκρισης αυτής καθορίζει το πώς θα μεταβληθεί η πρώτη ώστε να φτάσει στη τιμή της μεταβλητής αναφοράς. Αναφερόμαστε στη τιμή της εισόδου ως x και στη μεταβλητή αναφοράς ως w. Η διαφορά w-x είναι η απόκλιση-σφάλμα και αναφέρεται ως e. Επίσης υπάρχει και η μεταβλητή παρεμβολή z, η οποία είναι όλοι οι παράγοντες που μπορούν να μεταβάλουν την x και βρίσκονται στο σύστημα.

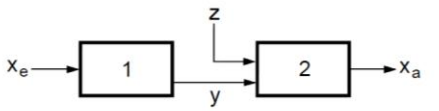


Fig. 1-2 Block diagram of an open-loop control system (general)

1 Controlling equipment (open-loop control function)
2 Controlled system
x_e Input variable (control variable)
x_a Output variable
y Manipulated variable
z Interference variable

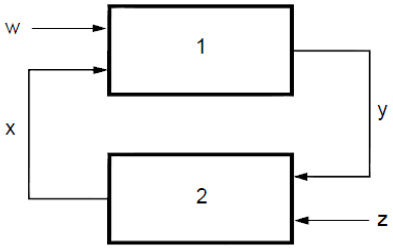


Fig. 1-6 Control loop (general)

1 Controlling equipment
2 Controlled system
w Reference variable (setpoint)
x Controlled variable (actual value)
y Manipulated variable
z Interference variable(s)

Εικόνα 2.2 – (Αριστερά) Open-Loop Control System (Δεξιά) Closed-Loop Control System

Ένα απλό παράδειγμα συστήματος κλειστού βρόχου μπορεί να είναι η σταθεροποίηση της θερμοκρασίας ενός δωματίου. Κάθε φορά δίνεται σαν είσοδος x η θερμοκρασία του δωματίου και συγκρίνεται με την επιθυμητή θερμοκρασία w, υπολογίζοντας έτσι την τιμή e. Στην περίπτωση που e=0, δεν χρειάζεται να γίνει καμία μεταβολή από τον ελεγκτή, στη περίπτωση που e>0 (w>x) πρέπει ο ελεγκτής να ανοίξει τη θέρμανση ενώ αντίθετα στη περίπτωση που e<0 (w<x), ο ελεγκτής πρέπει να κλείσει τη θέρμανση. Αυτή η απόφαση ενσωματώνεται στην μεταβλητή y, η οποία οδηγείται στο σύστημα θέρμανσης για να κάνει τη σωστή λειτουργία.

Τα συστήματα κλειστού βρόχου χωρίζονται σε δύο καταστάσεις λειτουργίας: την σταθερή κατάσταση λειτουργίας (steady-state condition) όπου η μεταβλητή εισόδου είναι ίση με την επιθυμητή, και την δυναμική λειτουργία. Η δυναμική λειτουργία είναι η κατάσταση όπου υπάρχει ο ελεγκτής αντιδρά με ένα συγκεκριμένο τρόπο σε κάποια μεταβολή, δηλαδή κάνει μία δυναμική απόκριση(dynamic response).

Τα συστήματα που θα αναφερθούν στην συγκεκριμένη εργασία, είναι συστήματα κλειστού βρόχου.

Το ελεγχόμενο σύστημα, είναι το σύστημα το οποίο ελέγχεται από τον ελεγκτή και χωρίζεται σε δύο κατηγορίες. Η πρώτη είναι τα μη αυτοελεγχόμενα συστήματα(non-self-regulating controlled systems) ή μη στατικά(astatic) ή ολοκληρωμένα

συστήματα(IntegralSystems). Η δεύτερη κατηγορία είναι τα αυτοελεγχόμενα συστήματα(self-regulating controlled systems) ή στατικά(static) ή αναλογικά συστήματα(proportional systems).

Στη πρώτη κατηγορία εντάσσονται τα συστήματα που δεν μπορούν να βρεθούν σε μια νέα σταθερή κατάσταση, εάν πιο πριν απόκλιναν από την σταθερή κατάσταση που ήταν. Για παράδειγμα, υπάρχει ένας κουβάς με νερό που χάνει και γεμίζει με τον ίδιο ρυθμό νερό και άρα βρίσκεται σε σταθερή κατάσταση. Στη περίπτωση που ο ρυθμός εισροής νερού αυξηθεί, το νερό στο κουβά θα αρχίσει να αυξάνεται και μετά το πέρας κάποιου χρονικού διαστήματος θα υπάρχει υπερχείλιση.

Στα αυτοελεγχόμενα συστήματα, η μεταβλητή x πάντα τείνει να φτάσει σε μία νέα κατάσταση, αφού πιο πριν έχει δεχθεί μια διαφορά Δy. Για παράδειγμα, είναι γνωστό ότι αν το y δεχτεί μία μεταβολή ίση με +1, το x τότε αυξηθεί κατά +2. Έτσι αν σε κάποια άλλη στιγμή υπάρξει μεταβολή του y κατά +3, αναμένουμε ότι το x θα αυξηθεί κατά +6. Η σχέση των δύο μεταβλητών ονομάζεται απόκριση σταθερής κατάστασης(steady-state response) όταν είμαστε σε σταθερή κατάσταση λειτουργίας. Ορίζεται επίσης ο συντελεστής μεταφοράς (transfer coefficient) ως $K_s = \frac{\Delta x}{\Delta y} = tg_a$. Η χαρακτηριστική παράσταση ονομάζεται χαρακτηριστική σταθερής κατάστασης του ελεγχόμενου συστήματος (steady-state characteristic of the controlled system) ή χαρακτηριστική ελέγχου (control characteristic) και φαίνεται στην εικόνα 2.3. Είναι σημαντικό να σημειωθεί ότι τέτοια συστήματα στη φύση είναι σπάνια αφού δεν παρατηρείται συχνά αυτή η γραμμικότητα (π.χ. αν θέλουμε να ελέγξουμε θερμοκρασία, έχει μη γραμμική συμπεριφορά).

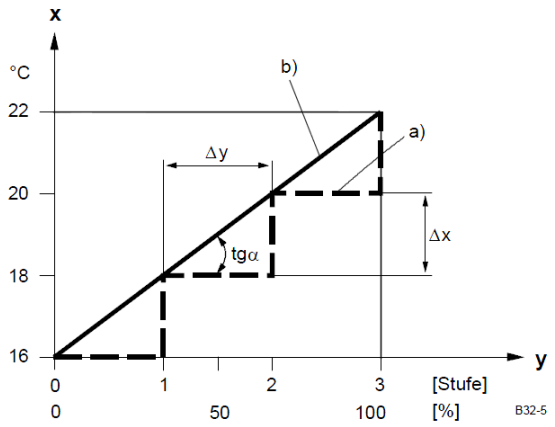


Fig. 2-5 Control characteristic of the electric room heater as per Fig. 2-3

a) 3-position control
b) Infinitely variable control

Εικόνα 2.2 - Χαρακτηριστική σταθερής κατάστασης του ελεγχόμενου συστήματος

Στη περίπτωση που αναφερόμαστε στη δυναμική απόκριση αυτοελεγχόμενων συστημάτων, αναφερόμαστε στη χρονική καθυστέρηση που υφίσταται μέχρι η μεταβλητή x να φτάσει στην επιθυμητή τιμή. Η χρονική αυτή καθυστέρηση οφείλεται σε αποθήκευση ενέργειας στη μορφή θερμικής απόκρισης στα κατασκευαστικά μέρη όπως είναι οι βαλβίδες και η μάζα των αγωγών. Κάποια παραδείγματα αποθήκευσης ενέργειας είναι η θέρμανση του νερού μέσα σε ένα λέβητα και η θέρμανση του δωματίου από το καλοριφέρ. Όσο πιο πολλά

στοιχεία αποθήκευσης υπάρχουν σε ένα σύστημα, τόσο πιο περίπλοκο γίνεται. Για την αξιολόγηση της δυναμικής απόκρισης, μπορεί να χρησιμοποιηθεί η μέθοδος μέσω της βηματικής απόκρισης. Γνωρίζοντας ότι η μεταβλητή y παρέχει πληροφορίες για την απόκριση του συστήματος ως προς το σημείο αναφοράς και η x για την απόκριση του συστήματος, μπορούμε να μεταβάλουμε τη τιμή του y με μία βηματική συνάρτηση και να λάβουμε μία βηματική απόκριση για το πώς θα επηρεαστεί το σύστημα. Η μέθοδος της βηματικής απόκρισης φαίνεται στην εικόνα 2.4.

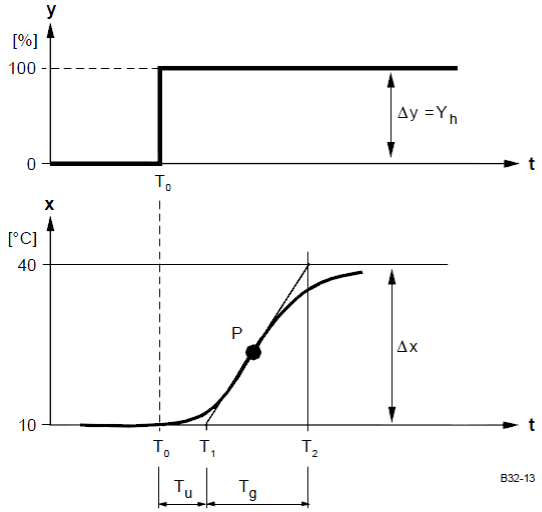


Fig. 2-15 Determining the delay time T_U and balancing time T_g of a multiple storage element system

P Inflexion point
T_u Delay time
T_g Balancing time

Εικόνα 2.3 - Ελεγχόμενο σύστημα με νεκρό χρόνο και στοιχεία αποθήκευσης

Νεκρός χρόνος (dead time) είναι ο χρόνος που μεσολαβεί χωρίς κάποια μεταβολή του συστήματος, γιατί το ελεγχόμενο σύστημα βρίσκεται κάποια απόσταση μακριά από το σημείο εφαρμογής της μεταβλητής y. Συστήματα που μπορεί να εμπεριέχουν νεκρό χρόνο μπορεί να είναι μια βαλβίδα που ελέγχει τη ροή νερού σε μία δεξαμενή, η οποία βρίσκεται μακριά από την βαλβίδα και το νερό πρέπει να διανύσει κάποια απόσταση μέσα σε σωληνώσεις. Η τιμή του νεκρού χρόνου εξαρτάται από την απόσταση αυτή. [1B]

2.5. Ελεγκτές

Η λειτουργία ενός ελεγκτή είναι να μεταβάλλει αυτόματα τη τιμή μεταβλητής y, η οποία θα επηρεάσει την διαφορά e που έχει δημιουργηθεί ανάμεσα στην επιθυμητή τιμή και στην τιμή αναφοράς από κάποιο εξωτερικό παράγοντα. Για μία τέτοια λειτουργία, ο ελεγκτής χρειάζεται να έχει κάποιο αισθητήρα για να μετράει συνέχεια την τιμή της μεταβλητής που θέλουμε να ελέγχουμε και να την συγκρίνει συνεχώς με τη μεταβλητή αναφοράς ώστε να κάνει τις απαραίτητες ενέργειες για να μηδενιστεί το σφάλμα ανάμεσα σε αυτές τις δύο τιμές.

Οι ελεγκτές μπορούν να κατηγοριοποιηθούν σε ομάδες ανάλογα με:

- την εφαρμογή που θα χρησιμοποιηθούν
- τον τύπο της ελεγχόμενης μεταβλητής x
- τη βοηθητική ενέργεια που χρειάζεται
- την ενεργό διεύθυνση
- τη μεταβλητή ελέγχου y
- τη χαρακτηριστική απόκρισης

Η κατηγορία που αφορά τον τύπο της ελεγχόμενης μεταβλητής x λειτουργεί ως εξής: Αν η ελεγχόμενη μεταβλητή για παράδειγμα είναι θερμοκρασίας, όπως και στην περίπτωσή μας, πρέπει να χρησιμοποιηθεί ένας ελεγκτής θερμοκρασίας ενώ αν απαιτείται να ελεγκτεί η υγρασία, θα πρέπει να χρησιμοποιηθεί ελεγκτής υγρασίας κοκ.

Η κατηγορία βοηθητικής ενέργειας χωρίζεται σε δύο υποκατηγορίες: στους ελεγκτές που λειτουργούν από μόνοι τους (self-actuated controllers) και στους ελεγκτές που χρειάζονται επιπρόσθετη ενέργεια (powered controllers). Στην πρώτη υποκατηγορία εντάσσονται οι ελεγκτές που χρησιμοποιούν την ενέργεια από τον μηχανισμό μέτρησης της ελεγχόμενης μεταβλητής(π.χ. αισθητήρας) για να λειτουργήσουν το ελεγχόμενο σύστημα. Τέτοιοι ελεγκτές συνήθως χρησιμοποιούνται για μέτρηση θερμοκρασίας και πίεσης. Στην δεύτερη υποκατηγορία εντάσσονται οι ελεγκτές που δεν μπορούν να αντλήσουν αρκετή ενέργεια από τον μηχανισμό μέτρησης της ελεγχόμενης μεταβλητής και χρειάζονται επιπρόσθετη ενέργεια για να ελέγξουν το σύστημα. Ανάλογα με το είδος της επιπρόσθετης ενέργειας που χρησιμοποιούν αυτοί οι ελεγκτές, διακρίνονται σε ηλεκτρικούς, πνευματικούς, υδραυλικούς, κλπ.

Ανάλογα, και η κατηγορία που σχετίζεται με τη μεταβλητή ελέγχου χωρίζεται σε τρείς υποκατηγορίες ανάλογα με το είδος του σήματος εξόδου:

- Συνεχούς λειτουργίας ελεγκτές (Continuous-action controllers): Υπολογίζουν συνεχώς την διαφορά e και μπορούν να χρησιμοποιήσουν οποιαδήποτε τιμή μπορεί να λάβει η μεταβλητή y για να επαναφέρουν το σύστημα στη σταθερή κατάσταση.
- Διακοπτόμενης λειτουργίας ελεγκτές (Discontinuous-action controllers): Μπορούν να χρησιμοποιήσουν μόνο συγκεκριμένες τιμές της μεταβλητής y. Για παράδειγμα η έξοδος του ελεγκτή μπορεί να είναι μόνο 0 ή 1, δηλαδή ON/OFF.
- Ημησυνεχόμενοι ελεγκτές (Quasi-continuous controllers): Η τιμή του σήματος εξόδου μπορεί να έχει οποιαδήποτε τιμή μπορεί να λάβει αλλά η συχνότητα με την οποία θα λαμβάνεται η απόφαση έχει μια προκαθορισμένη τιμή. Ένα παράδειγμα είναι οι ελεγκτές δύο καταστάσεων με συγκεκριμένο πλάτος παλμού (Ton/Toff = 0…100%).

Ένας ελεγκτής πρέπει να λάβει απόφαση για το πόσο γρήγορα, κατά πόσο και για πόσο θα πρέπει να μεταβάλει τη λειτουργία του συστήματος ελέγχου για να αντιμετωπίσει μία διαφορά e. Για να περιγραφεί αυτή η λειτουργία χρησιμοποιείται η σταθερής κατάστασης και η δυναμικής κατάστασης απόκριση. Στην απόκριση σταθερής κατάστασης αναφέρεται η σχέση ανάμεσα στις μεταβλητές x και y και ορίζεται ο συντελεστής μεταφοράς (transfer

coefficient) ως $K_R = \frac{\Delta y}{\Delta x}$. Στη δυναμική απόκριση περιγράφεται η πρόοδος της μεταβλητής y σε σχέση με το χρόνο σε περίπτωση που υπάρχει μεταβολή της μεταβλητής x. [1B]

2.5.1. Ελεγκτής P

Ο P ελεγκτής έχει σαν χαρακτηριστικό την γραμμική σχέση ανάμεσα στις μεταβλητές x ή e και y, δηλαδή κάθε μεταβλητή εισόδου έχει μία σταθερή και προκαθορισμένη αντίστοιχη τιμή για τη μεταβλητή ελέγχου. Επίσης είναι εξαρτημένος από το φορτίο (load-dependent) αφού για να ξεκινήσει να λειτουργεί πρέπει να δημιουργηθεί μία συγκεκριμένη διαφορά e, δηλαδή να μεταβληθεί η είσοδος κατά μία συγκεκριμένη τιμή. Αυτή του η εξάρτηση από την είσοδο τον καθιστά ακατάλληλο να χρησιμοποιείται μόνος του, στη περίπτωση αυτής της εργασίας, αφού θα περιμένει να δημιουργηθεί διαφορά για να πάρει το σύστημα σε σταθερή κατάσταση.

Στη σταθερή κατάσταση φαίνεται η σχέση ανάμεσα στις μεταβλητές εξόδου και ελέγχου (εικόνα 3.5). Το Yh αντιπροσωπεύει το εύρος τιμών της μεταβλητής εξόδου ενώ το Xp αντιπροσωπεύει το εύρος στο οποίο ο ελεγκτής θα έχει μεταβλητή συμπεριφορά, δηλαδή στο οποίο θεωρείται ανεκτό το σφάλμα. Η τιμή του Xp καθορίζει το πόσο καλή θα είναι η συμπεριφορά του ελεγκτή αφού όσο πιο μικρό είναι, τόσο πιο γρήγορα θα έρχεται το σύστημα σε σταθερή κατάσταση. Πρέπει όμως να δίνεται μεγάλη προσοχή στη τιμή που θα δοθεί στο Xp αφού όσο πιο μικρό είναι, τόσο πιο μεγάλη είναι η πιθανότητα το σύστημα να ταλαντώνεται. Ακόμα, πολύ εύκολα μπορούμε να αποδείξουμε ότι: $K_P = \frac{\Delta y}{\Delta x} = \frac{Y_h}{X_p} = \frac{100\%}{X_p} = \frac{1}{X_p} \Leftrightarrow K_P = \frac{1}{X_p}$

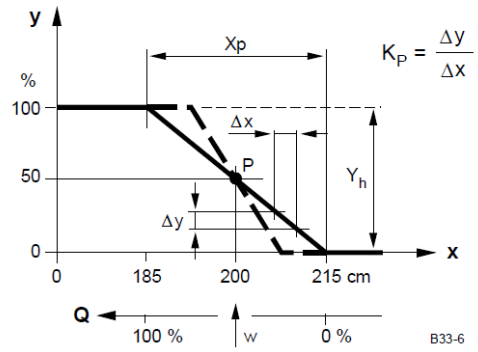


Fig. 3-7 Steady-state characteristic and P-band of the P-controller as per Fig. 3-5

Δx Controlled variable change
Δy Manipulated variable change
X_p Proportional band
Y_h Positioning range
K_P Transfer coefficient
Q Load
P Offset

Εικόνα 2.4 – Χαρακτηριστική σταθερής κατάστασης ελεγκτή P

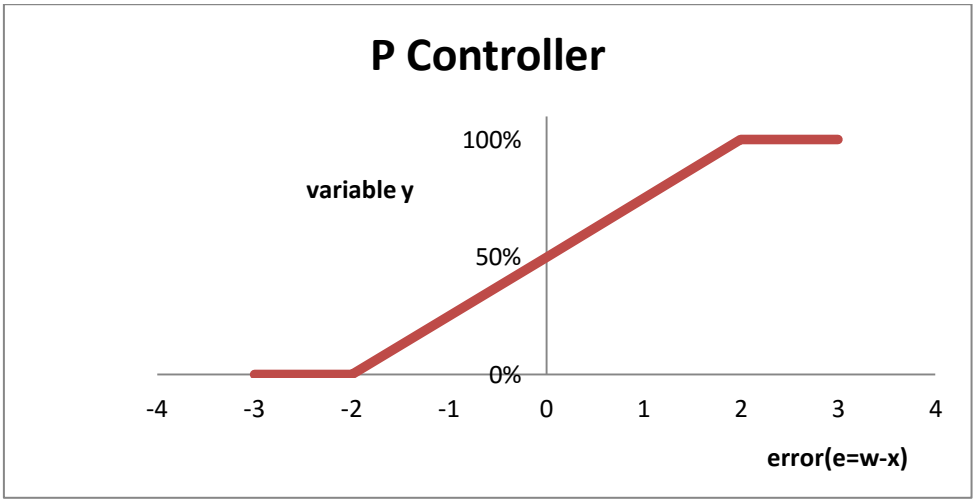
Η δυναμική απόκριση του ελεγκτή P, μπορεί να εξεταστεί με τη βηματική απόκριση. Για μία μεταβολή της μεταβλητής x, η έξοδος του ελεγκτή θα εξαρτηθεί τόσο από το μέγεθος της μεταβλητής όσο και από την τιμή του Kp και άρα αντίστοιχα από το Xp. Για παράδειγμα, αν

δημιουργηθεί μία μεταβολή Δx και το Xp=100%, τότε η έξοδος θα έχει μεταβολή Δy=Δx. Στη περίπτωση όμως που ήταν Xp=50%=0.5, τότε η έξοδος θα έχει μεταβολή $\Delta_y = \frac{\Delta_x}{0.5} = 2\Delta_x$. Από τη χαρακτηριστική της δυναμικής απόκρησης, μπορούμε να εξάγουμε τα συμπεράσματα ότι ο ελεγκτής P είναι εξαρτημένος από το φορτίο λόγω της γραμμικής του ιδιότητας και ότι είναι ένας γρήγορος ελεγκτής.

Ένα άλλο χαρακτηριστικό του συγκεκριμένου ελεγκτή είναι ότι μπορεί να μετατοπιστεί το σημείο αναφοράς, δηλαδή το setpoint να μην βρίσκεται στο μέσο της περιοχής που θα παρουσιάζει τη γραμμική συμπεριφορά. Στην εικόνα 2.5 για παράδειγμα που βρίσκεται στο μέσο, το offset είναι ίσο με 50% ενώ σε άλλες περιπτώσεις θα μπορούσε να είναι ακόμα και 0% ή 100%. [1B]

Στη συγκεκριμένη εργασία, θα χρησιμοποιηθεί ένας ελεγκτής P λόγω του χαρακτηριστικού του ότι είναι γρήγορος. Θεωρήθηκε το setpoint w ότι είναι η θερμοκρασία που επιθυμεί ο χρήστης να έχει τη συγκεκριμένη χρονική στιγμή, το x η θερμοκρασία που μετριέται από τον αισθητήρα και το y η τιμή που θα ελέγχει το χρόνο λειτουργίας της θέρμανσης. Στη σταθερή κατάσταση όπου e=0, θεωρούμε ότι ο κύκλος λειτουργίας της θέρμανσης θα πρέπει να είναι $T_{on}=T_{off}=0{,}5T_{total}$ και άρα y=0,5 ενώ επιλέγηκε ότι η γραμμική περιοχή του ελεγκτή θα είναι ±2 βαθμούς κελσίου από το setpoint και άρα το offset είναι ίσο με 50%. Τελικά, θα φτιαχτεί ένας ελεγκτής που ανάλογα με το σφάλμα και άρα με τη θερμοκρασία δωματίου, θα ελέγχει το πόσο χρόνο θα πρέπει να είναι ανοιχτή ή κλειστή η θέρμανση.

Αφού λοιπόν το Xp ορίζεται ως το εύρος στο οποίο ο ελεγκτής θα έχει γραμμική συμπεριφορά και ορίστηκε ότι η γραμμική περιοχή του ελεγκτή θα είναι ±2 βαθμούς κελσίου από το setpoint, τότε θα ισχύει ότι $X_p = 4 \Rightarrow K_p = 0.25$. Η χαρακτηριστική λοιπόν που θα πρέπει να έχει ο ελεγκτής φαίνεται στην εικόνα 2.6.



Εικόνα 2.5 - Χαρακτηριστική ελεγκτή P

Η συνάρτηση λοιπόν του ελεγκτή P που θα χρησιμοποιηθεί είναι:

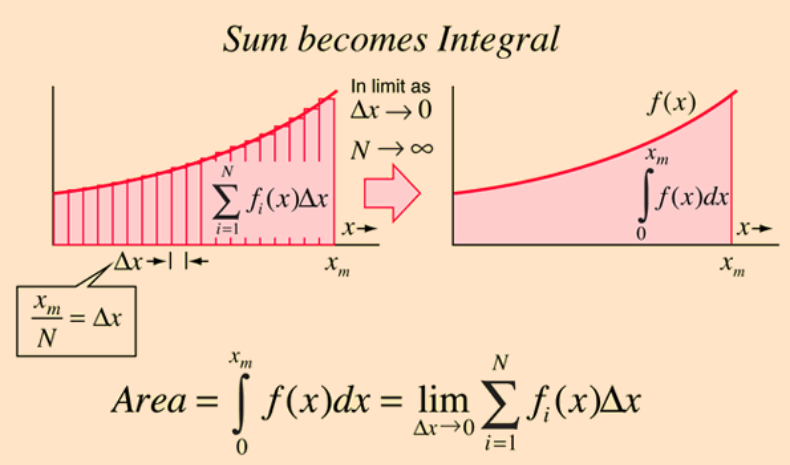
$$y = \begin{cases} 0, & e < -2 \\ K_p * e + 0.5, & -2 \le e \le 2 \\ 1, & e > 2 \end{cases}$$

Σχέση 2.1 – Εξίσωση ελεγκτή P

2.5.2. Ελεγκτής I

Ο ελεγκτής I (I controller ή integral-action controller), βασίζει τη λειτουργεία του στον ορισμό της ολοκλήρωσης, δηλαδή στον ορισμό της πρόσθεσης και του συνδυασμού. Η έξοδος y λοιπόν του συγκεκριμένου ελεγκτή, διαμορφώνεται ανάλογα από το άθροισμα των συνεχόμενων μεταβλητών εισόδου x στην περίοδο του χρόνου.

Ορισμός Ολοκληρώματος: Το ολοκλήρωμα, δίνει αριθμούς σε συναρτήσεις με τέτοιο τρόπο ώστε να μπορεί να περιγράψει μετατόπιση, εμβαδόν, όγκο και άλλα. Στη περίπτωση του επιπέδου χώρου με δοσμένα όρια [α,β] και συνάρτηση f(x), μπορεί να βρεθεί το εμβαδόν κάτω από την συνάρτηση και ανάμεσα στα όρια.[11]



Εικόνα 2.6 – Ορισμός Ολοκληρώματος

Όπως φαίνεται στην εικόνα 2.7, το ολοκλήρωμα μπορεί να γραφεί σαν ένα άθροισμα συναρτήσεων. Ουσιαστικά το εμβαδόν κάτω από τη συνάρτηση, χωρίζεται σε άλλα πιο μικρά εμβαδά τα οποία τελικά προστίθενται. Τα διαστήματα που χρησιμοποιούνται για τον υπολογισμό των επιμέρους εμβαδών, απαιτείται να είναι όσο πιο μικρά γίνεται ώστε να μπορεί να υπολογιστεί με μεγαλύτερη ακρίβεια το τελικό εμβαδόν.

Ο ελεγκτής I [12] είναι ανάλογος τόσο στο μέγεθος του σφάλματος όσο και στη διάρκεια του σφάλματος. Η ολοκλήρωση που κάνει τελικά, είναι το άθροισμα των στιγμιαίων σφαλμάτων e κατά τη διάρκεια του χρόνου και άρα τελικά δίνει σαν αποτέλεσμα το συσσωρευμένο αντιστάθμισμα που θα έπρεπε να είχε διορθωθεί από πιο πριν. Το συσσωρευμένο σφάλμα στο τέλος πολλαπλασιάζεται με τη σταθερά ολοκλήρωσης ελέγχου (integral control action constant) K_i και προστίθεται στην έξοδο του ελεγκτή.

$$y_n = K_i * I_n = K_i \int_0^t e(\tau)d\tau \quad \mu\varepsilon\ I_n = I_{n-1} + e * \Delta t\ \kappa\alpha\iota\ I_0 = 0, \qquad n = 1,2,3,\ldots$$

Σχέση 2.2 – Εξίσωση ελεγκτή I

Από τη σχέση 2.3 παρατηρούμε ότι λόγω του συσσωρευμένου σφάλματος από το παρελθόν, μπορεί να ξεπεραστεί η τιμή του setpoint. Για να αποφευχθεί αυτή η συμπεριφορά (integral windup), προστίθεται στον ελεγκτή μία ανώτατη τιμή (κατώφλι), πάνω από την οποία το ολοκλήρωμα σταματά να προσθέτει περεταίρω σφάλμα. Ακόμα, ο ελεγκτής I, επιταχύνει τη διαδικασία προς την επίτευξη του setpoint, αφού όσο πιο μεγάλο είναι το σφάλμα τόσο πιο μεγάλη θα είναι η απόκρισή του. Αφού λοιπόν για κάθε τιμή του σφάλματος και για κατάλληλο χρονικό διάστημα, ο ελεγκτής I μπορεί να φέρει το σύστημα σε σταθερή κατάσταση, είναι ανεξάρτητος από το φορτίο (load-independent).

Στη σταθερή κατάσταση, όπου το σφάλμα είναι μηδενικό, δεν υπάρχει καμία εξάρτηση ανάμεσα στο σφάλμα και στο σήμα εξόδου y και συνεπώς η έξοδος είναι ίση με 0. Στη δυναμική απόκριση η κάθε μεταβολή της εισόδου x και συνεπώς του σφάλματος e, προκαλούν μεταβολή του σήματος εξόδου με εξάρτηση ως προς το χρόνο. [1B]

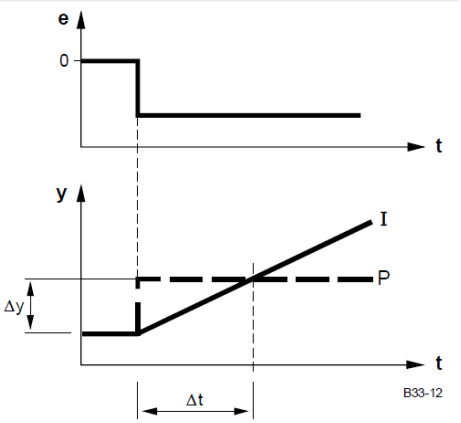


Fig. 3-14 Dynamic characteristic of the I-controller

e Input (control difference)
I Output of the integral-action controller
P Output of a proportional-action controller

Εικόνα 2.7 - Χαρακτηριστική δυναμικής απόκρισης I και P ελεγκτή

Στη συγκεκριμένη εργασία χρησιμοποιήθηκε ο ελεγκτής I για βελτιστοποίηση.

2.5.3. Ελεγκτής D

Ο ελεγκτής D (derivative-action element ή D-element), χρησιμοποιεί τον ορισμό της παραγώγου για να υπολογίσει την κλίση του σφάλματος e και να πολλαπλασιάσει το αποτέλεσμα με τη σταθερά παραγώγου ελέγχου (derivative control action constant) K_D. Το σήμα εξόδου y του ελεγκτή υπολογίζεται μόνο στη περίπτωση που υπάρχει μεταβολή της εισόδου και άρα του σφάλματος και όσο πιο απότομη είναι αυτή η μεταβολή, τόσο πιο μεγάλο θα είναι το σήμα εξόδου.

$$y = K_D \frac{de}{dt}$$

Σχέση 2.3 – Εξίσωση ελεγκτή D

Στη σχέση 2.4 φαίνεται ο ορισμός του ελεγκτής D και κατανοείται ότι ο συγκεκριμένος ελεγκτής δεν μετράει τη τιμή του σφάλματος αλλά το ρυθμό που αλλάζει το σφάλμα. Το συγκεκριμένο σήμα εξόδου ιδανικά θα ήταν μία συνάρτηση δέλτα απείρου ύψους αλλά επειδή στην πράξη αυτά τα σήματα δεν μπορούν να υλοποιηθούν, το σήμα είναι παρόμοιο με αυτό που προκύπτει από ένα ηλεκτρικό κύκλωμα RC.

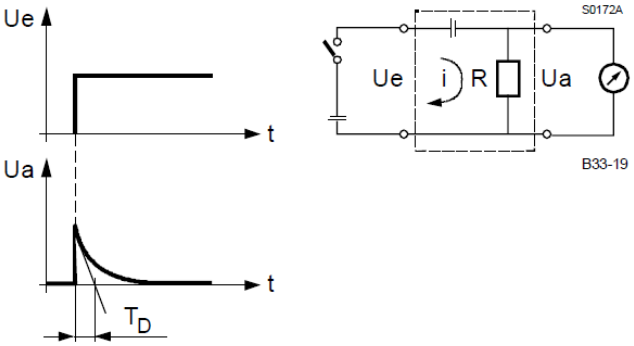


Fig. 3-19 RC circuit and associated step response as an example of a derivate action control element
T_v = Derivative action time

Εικόνα 2.8 - Κύκλωμα RC και αποτέλεσμα εξόδου ελεγκτή D

Επεξήγηση εικόνας 2.9: Κατά τα γνωστά οι πυκνωτές αποκόπτουν το συνεχές ρεύμα (DC current) και αφήνουν να περάσει το διακοπτόμενο ρεύμα (AC current) και άρα στη περίπτωση που σαν είσοδος δοθεί συνεχές ρεύμα (ή στη περίπτωσή μας e=σταθερό), το αποτέλεσμα θα είναι ίσο με 0. Όταν όμως δημιουργηθεί αλλαγή στη τιμή του ρεύματος (δηλαδή όπως στη βηματική απόκριση του σφάλματος), ο πυκνωτής θα αφήσει το ρεύμα να περάσει και θα δημιουργηθεί στιγμιαία τάση (ή έξοδος ελεγκτή) στην αντίσταση με πολύ μεγάλη τιμή λόγω της πολύ απότομης μεταβολής του ρεύματος (ή σφάλματος). Στη συνέχεια, ο πυκνωτής θα αρχίσει να εκφορτίζεται με αποτέλεσμα η τάση (ή έξοδος ελεγκτή) να αρχίζει να τείνει στο 0 όπου και τελικά θα καταλήξει.

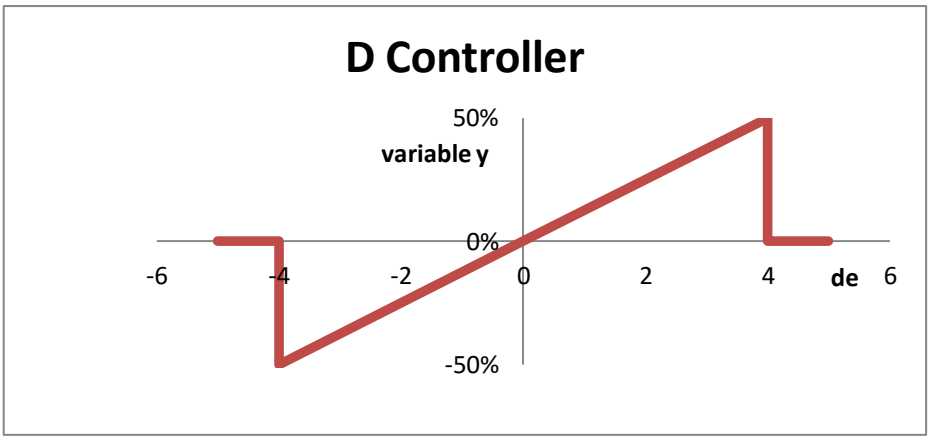
Από τη μορφή του σήματος εξόδου γίνεται κατανοητό ότι ο συγκεκριμένος ελεγκτής δίνει ένα προβάδισμα στην αρχή (head start) για να κάνει την αρχική απόκριση πιο γρήγορη και πιο μεγάλη. [1B]

Στη συγκεκριμένη εργασία, ο ελεγκτής D θα χρησιμοποιηθεί για βελτιστοποίηση. Η περίοδος όπου θα χρησιμοποιείται ο ελεγκτής θα είναι στο ±2 γύρω από το setpoint και άρα υπάρχουν 2 ακραίες περιπτώσεις που μπορούν να δημιουργηθούν.

- Το σφάλμα τείνει από -2 προς το 2 και άρα από εκεί που το setpoint ήταν μικρότερο από τη θερμοκρασία δωματίου, αρχίζει να γίνεται μεγαλύτερο. Για να αντισταθμιστεί αυτή η συμπεριφορά πρέπει να αρχίσει να λειτουργεί η θέρμανση. Υπολογίζεται ότι $\frac{de}{dt} > 0$ με de=2-(-2)=4.

- Το σφάλμα τείνει από 2 προς το -2 και άρα από εκεί που το setpoint ήταν μεγαλύτερο από τη θερμοκρασία δωματίου, αρχίζει να γίνεται μικρότερο. Για να αντισταθμιστεί αυτή η συμπεριφορά πρέπει η θέρμανση να κλείσει τελείως. Υπολογίζεται ότι $\frac{de}{dt} < 0$ με de=-2-2=-4.

Με βάση τα πιο πάνω ο ελεγκτής D για τη συγκεκριμένη εργασία θα παρουσιάζει μία χαρακτηριστική της πιο κάτω μορφής:



Εικόνα 2.9 - Χαρακτηριστική ελεγκτή D

2.5.4. Ελεγκτές PI και PD

Ο ελεγκτής PI είναι ο συνδυασμός των ελεγκτών P και I και το σήμα εξόδου υπολογίζεται ως το άθροισμα των σημάτων εξόδου των δύο ελεγκτών. Τα χαρακτηριστικά του συγκεκριμένου ελεγκτή είναι ότι είναι γρήγορος αφού χρησιμοποιείται ο ελεγκτής P και με τη βοήθεια του ελεγκτή I γίνεται πιο απότομη η σύγκληση στο setpoint όταν υπάρχουν μεγάλα σφάλματα.

$$y = y_P + y_I = (K_p * e + 0.5) + (K_i * I_n)\ \gamma\iota\alpha\ -2 \le e \le 2$$

$$\text{όπου } I_n = I_{n-1} + e * \Delta t \text{ και } I_0 = 0, \qquad n = 1,2,3,\dots$$

Σχέση 2.4 – Εξίσωση ελεγκτή PI

Αντίστοιχα, ο ελεγκτής PD είναι ο συνδυασμός των ελεγκτών P και D και το σήμα εξόδου υπολογίζεται ως το άθροισμα των σημάτων εξόδου των δύο ελεγκτών. Ο συγκεκριμένος ελεγκτής είναι γρήγορος καθώς έχει και μία απότομη ώθηση στην αρχή. [1B]

$$y = y_P + y_D = (K_p * e + 0.5) + K_D \frac{de}{dt}\ \ \gamma\iota\alpha\ -2 \le e \le 2$$

Σχέση 2.5 – Εξίσωση ελεγκτή PD

2.5.5. Ελεγκτής PID

Ο ελεγκτής PID είναι ο συνδυασμός των ελεγκτών P,I και D. Θεωρητικά η έξοδος του ελεγκτή είναι το άθροισμα των επιμέρους ελεγκτών αλλά πρακτικά η έξοδος του ελεγκτή P, επηρεάζει τους άλλους 2 ελεγκτές.

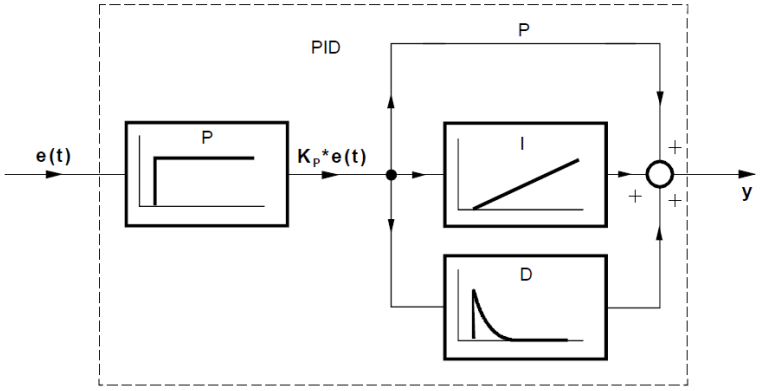
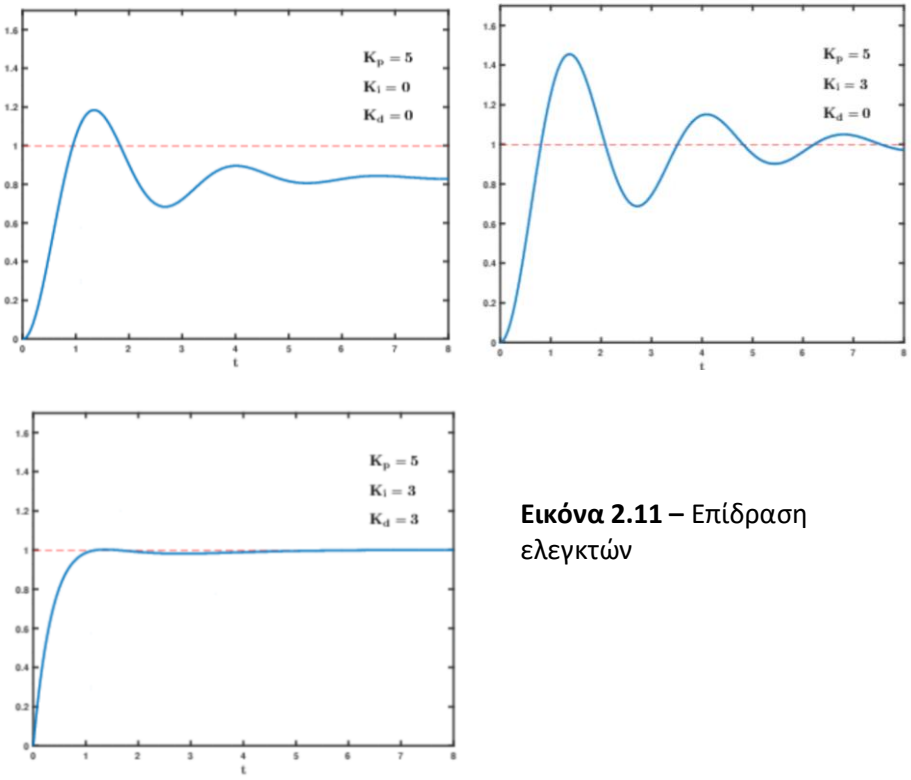


Fig. 3-26 Representation of a PID controller in accordance with actual practice

Εικόνα 2.10 - Χαρακτηριστική ελεγκτή PID

Για τον ορισμό του PID χρησιμοποιούνται οι σταθερές του κάθε ελεγκτή ώστε να επιτευχθεί η σωστή τιμή για την τιμή που θα έχει ο ελεγκτής P, για το πόσο απότομη θα είναι η κλίση στον ελεγκτή I και για το πόσο θα είναι το εμβαδόν και συνεπώς πόσο απότομη θα είναι η εκφόρτηση για τον ελεγκτή D. Ο ελεγκτής P, θα δώσει την μεγαλύτερη επίδραση στο τελικό αποτέλεσμα αλλά ο τελικός συνδυασμός είναι ικανός να ελέγξει πολύπλοκα και δύσκολα συστήματα με πολύ καλή απόδοση και αυτός είναι ο λόγος που χρησιμοποιήθηκε στην παρούσα εργασία.



Εικόνα 2.11 – Επίδραση ελεγκτών

Στην εικόνα 2.12 φαίνεται η επίδραση της κάθε παραμέτρου στο τελικό αποτέλεσμα. Οπτικά φαίνεται ότι ο ελεγκτής P δίνει μία αρχική ώθηση προς τη τιμή του setpoint. Λόγω της μη ικανότητάς του να κρατήσει σε σταθερή κατάσταση το σύστημα χρησιμοποιείται και ο ελεγκτής I, ο οποίος βοηθά το σύστημα να ταλαντωθεί γύρω από το setpoint. Τέλος, χρησιμοποιείται ο D ελεγκτής ο οποίος μειώνει την ταλάντωση και καταφέρνει να φέρει το σύστημα στην επιθυμητή κατάσταση γρήγορα και με πολύ καλή απόδοση. Ουσιαστικά ο ελεγκτής P χρησιμοποιείται για την μεγάλη επίδραση και οι ελεγκτές I και D για να κάνουν κάποια διόρθωση στην έξοδο του P. [1B]

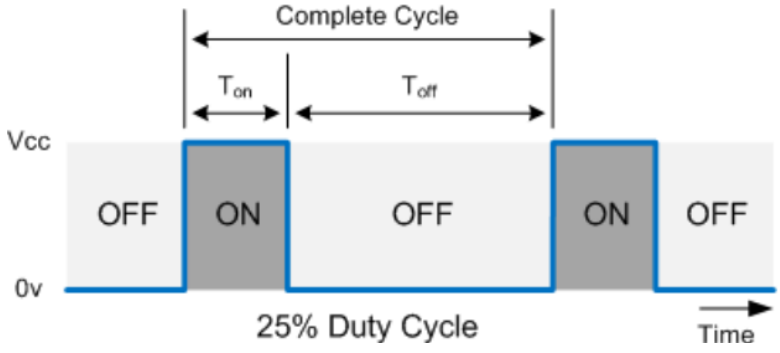
Στη συγκεκριμένη εργασία, ο ελεγκτής που θα χρησιμοποιηθεί θα είναι ο PID, ωστόσο, λόγω του ότι θα ελέγχει σύστημα θέρμανσης, οι τιμές που μπορεί να λαμβάνει σαν είσοδο το σύστημα θέρμανσης θα πρέπει να βρίσκονται στο διάστημα [0,1]. Για να υλοποιηθεί μία τέτοια λειτουργία, ο ελεγκτής θα ενεργοποιείται μόνο στο διάστημα όπου το σφάλμα είναι στο διάστημα [-2,2] και θα υπολογίζεται η έξοδος y ως το άθροισμα των τριών επιμέρους εξόδων των ελεγκτών. Στην περίπτωση όμως που το συγκεκριμένο άθροισμα λάβει τιμή μικρότερη του 0, θα παίρνει την τιμή 0 και στην περίπτωση που λάβει τιμή μεγαλύτερη του 1, θα παίρνει την τιμή 1. Η τιμή 0 θα αντιπροσωπεύει ότι η θέρμανση θα πρέπει να είναι κλειστή συνεχώς ενώ η τιμή 1 θα αντιπροσωπεύει ότι η θέρμανση θα πρέπει να είναι συνεχώς ανοιχτή. Η τιμή 0,5 υπονοεί ότι η θερμοκρασία δωματίου είναι ίση με την επιθυμητή και άρα για να παραμείνει σταθερή στη συγκεκριμένη τιμή θα πρέπει στη μισή περίοδο η θέρμανση να είναι ανοικτή και στην υπόλοιπη μισή περίοδο κλειστή.

Η συνολική περίοδος λειτουργίας της θέρμανσης και άρα και ο χρόνος που θα λαμβάνεται κάθε φορά η νέα τιμή για το y, T_{total}, τέθηκε ίση με 10 λεπτά στη συγκεκριμένη εργασία. Για κάθε νέα τιμή y που θα υπολογίζεται, θα πολλαπλασιάζεται με την τιμή T_{total} και θα προκύπτει ο χρόνος T_{on}. Ο χρόνος T_{off} ορίζεται ως το υπόλοιπο του χρόνου που απομένει για να συμπληρωθούν τα 10 λεπτά της περιόδου. Η έξοδος άρα από τον ελεγκτή θα είναι μία παλμοσειρά όπου κάθε παλμός θα έχει διάρκεια 10 λεπτά αλλά θα έχει διαφορετικό κύκλο(duty cycle).

$$T_{on} = y * T_{total} = y * 10min, \quad T_{off} = T_{total} - T_{on} = 10min - T_{on}$$

Σχέση 2.6 – Εξισώσεις κύκλου θέρμανσης

Το duty cycle ορίζεται ως το ποσοστό μίας περιόδου του σήματος, κατά την οποία το σήμα είναι σε λειτουργία, δηλαδή Ton προς τη συνολική διάρκεια της περιόδου. [13]



Εικόνα 2.12 - Ορισμός duty cycle

2.6. Θερμικά Συστήματα

Θερμικά συστήματα[14] είναι τα συστήματα τα οποία εμπλέκουν αποθήκευση και μεταφορά θερμότητας. Όπως ορίζει ο δεύτερος νόμος της θερμοδυναμικής, η θερμική ενέργεια μεταφέρεται πάντα προς την περιοχή με τη χαμηλότερη ενέργεια, δηλαδή από τις περιοχές υψηλής θερμοκρασίας προς τις περιοχές χαμηλής θερμοκρασίας. Η θερμότητα σε ένα σύστημα μπορεί να διαδοθεί με τους ακόλουθους τρείς τρόπους:

- Αγωγή (Conduction): Συμβαίνει όταν ένα αντικείμενο έχει διαφορά θερμοκρασίας όπως για παράδειγμα ο τοίχος σε κάποιο σπίτι που από τη μία πλευρά έχει θερμοκρασία δωματίου και από την άλλη έχει την εξωτερική θερμοκρασία.
- Συναγωγή ή μεταφορά μάζας (Convection) : Συμβαίνει όταν διαδίδεται η θερμότητα σε υγρή ή αέρια μορφή, όπως ο ανεμιστήρας που με αέρια μορφή απομακρύνει θερμότητα από κάποια περιοχή.
- Ακτινοβολία (Radiation): Συμβαίνει όταν ένα αντικείμενο έχει διαφορετικές θερμοκρασίες, όπως και στην αγωγή, αλλά στη συγκεκριμένη περίπτωση δεν απαιτείται κάποιο φυσικό μέσο για τη διάδοση της θερμότητας. Ένα τέτοιο παράδειγμα μπορεί να είναι η ακτινοβολία του ήλιου που ζεσταίνει τη Γη.

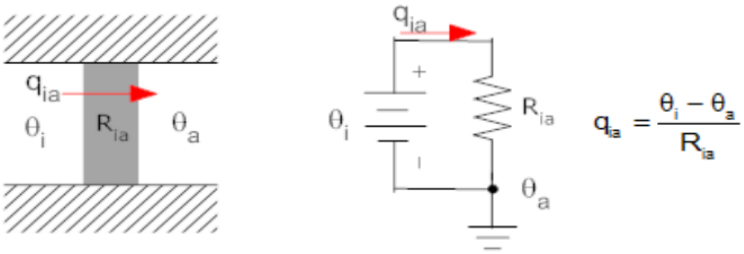
Ακόμα, ένα θερμικό σύστημα [15], αποτελείται από θερμικές αντιστάσεις και θερμικούς πυκνωτές, καθώς επίσης μπορούν να υπάρχουν τρείς πηγές θερμότητας: πηγές ισχύος, πηγές θερμοκρασίας και ροής υγρών.

Στη περίπτωση που υπάρχει ροή θερμότητας μέσω αγωγής σε ένα ομογενές σώμα, υπάρχουν πολλές παράμετροι που επηρεάζουν τη συγκεκριμένη ροή. Αρχικά, πρέπει να είναι γνωστές οι διαστάσεις του σώματος ώστε να υπολογιστεί το εμβαδόν της διατομής (A) και το μήκος (L) του υλικού, από όπου θα περάσει η ροή θερμότητας. Επίσης, πρέπει να είναι γνωστή η θερμική αγωγιμότητα (σ) του σώματος. Έχοντας γνωστές τις πιο πάνω παραμέτρους, μπορεί να υπολογιστεί η θερμική αντίσταση του σώματος με τη σχέση 2.7.

$$R = \frac{L}{A * \sigma} \ \ με\ μονάδες\ μέτρησης\ \frac{°C}{W}$$

Σχέση 2.7 – Εξίσωση θερμικής αντίστασης μέσω αγωγής

Τελικά, έχοντας γνωστή τη θερμική αντίσταση του σώματος και τις δύο θερμοκρασίες στα άκρα του σώματος μπορεί να υπολογιστεί η ροή θερμότητας. Στη περίπτωση που το σώμα είναι ένας τοίχος, οι δύο θερμοκρασίες που απαιτούνται είναι η θερμοκρασία δωματίου και η εξωτερική θερμοκρασία. Επίσης, ο τοίχος θα προσομοιαστεί με την θερμική αντίσταση και έτσι θα προκύψει το ηλεκτρικό ισοδύναμο που φαίνεται στην εικόνα 2.14. Η ροή ενέργειας q_{ia} φαίνεται με το κόκκινο βέλος και έχει φορά από τη θερμή περιοχή προς τη κρύα περιοχή. Ακόμα, στην εικόνα 2.14 φαίνεται και η σχέση για τον υπολογισμό της θερμικής ροής στη συγκεκριμένη περίπτωση.



Εικόνα 2.13 - Ηλεκτρικό ισοδύναμου τοίχου

Η ροή θερμότητας μέσω συναγωγής είναι ένας συνδυασμός της αγωγής μαζί με μεταφορά μάζας αφού όταν ένα υγρό συνορεύει με ένα στερεό σώμα, μπορεί να υπάρξει μεταφορά θερμότητας ανάμεσα στα δύο. Η συναγωγή χωρίζεται σε δύο κατηγορίες, στην εξαναγκασμένη συναγωγή όπου η ροή εξαναγκάζεται να δημιουργηθεί όπως στους ανεμιστήρες και στην ελεύθερη συναγωγή όπου η ροή υπάρχει αυθαίρετη στη φύση όπως για παράδειγμα η ροή υγρών λόγω διαφοράς πυκνό τας ανάμεσα στα δύο υγρά. Στη περίπτωση που υπάρχει ροή θερμότητας μέσω συναγωγής, η μοντελοποίηση είναι παρόμοια με την περίπτωση της αγωγής αλλά με τη διαφορά ότι η θερμική αντίσταση υπολογίζεται με διαφορετική σχέση. Στη συγκεκριμένη περίπτωση θα πρέπει να είναι γνωστή η σταθερά συναγωγής θερμικής μεταφοράς h (convection heat transfer coefficient) η οποία είναι διαφορετική για κάθε υγρό ή αέριο. Η σχέση από την οποία δίνεται η θερμική αντίσταση δίνεται στη σχέση 2.8.

$$R = \frac{1}{h * A}$$

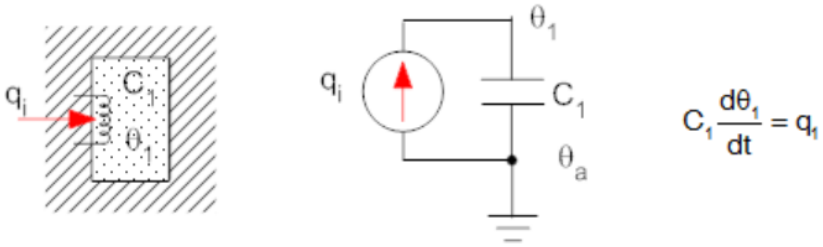
Σχέση 2.8 – Εξίσωση θερμικής αντίστασης μέσω συναγωγής

Η ροή θερμότητας μέσω ακτινοβολίας συμβαίνει όταν διάφορα αντικείμενα διαφορετικής θερμοκρασίας βρίσκονται σε οπτική επαφή. Η ενέργεια μεταφέρεται με ηλεκτρομαγνητικά κύματα και άρα δεν χρειάζεται κάποιο μέσο για να διαδοθεί. Ένα παράδειγμα είναι η ακτινοβολία του ηλίου στη Γη, η οποία θερμαίνει κάποιο σπίτι μέσω των παραθύρων. Για το συγκεκριμένο παράδειγμα, η ροή θερμότητας μπορεί να υπολογιστεί γνωρίζοντας την ενεργό επιφάνεια του παραθύρου A_w από όπου θα περάσει η ακτινοβολία, καθώς και την ηλιακή ακτινοβολία του ήλιου Φ_s με τη βοήθεια της σχέσης 2.9.

$$\frac{dQ}{dt} = A_w \Phi_s$$

Σχέση 2.9 – Εξίσωση ροής θερμότητας μέσω ακτινοβολίας

Επιπρόσθετα της αντίστασης θερμότητας, κάποια αντικείμενα μπορούν να έχουν και θερμική πυκνότητα η οποία περιγράφει τη ποσότητα της θερμικής ενέργειας που είναι αποθηκευμένη στο συγκεκριμένο αντικείμενο. Η θερμική ενέργεια συσσωρεύεται στο αντικείμενο μέσω κάποιων από τριών μεθόδων που περιεγράφηκαν πιο πάνω και όταν σταματήσουν τα φαινόμενα να επιδρούν, το αντικείμενο ξεκινά να χάνει αυτή την ενέργεια που είχε αποθηκεύσει από πιο πριν. Ένα αντίστοιχο ισοδύναμο ηλεκτρικό κύκλωμα φαίνεται στην εικόνα 2.15.



Εικόνα 2.14 - Ηλεκτρικό ισοδύναμου θερμικού πυκνωτή

Ακόμα, στα ισοδύναμα κυκλώματα υπάρχουν πηγές ισχύος ή πηγές θερμότητας που στην ουσία τροφοδοτούν το κύκλωμα. Οι συγκεκριμένες πηγές μπορούν να είναι είτε σταθερές είτε να μεταβάλλονται με τον χρόνο και αντιπροσωπεύονται από πηγές ρεύματος. Μία ιδανική πηγή θερμότητας παράγει ισχύ για το κύκλωμα ανεξάρτητα από τη θερμοκρασία και μπορεί για παράδειγμα να είναι ένα κλιματιστικό.

Οι πηγές θερμοκρασίας είναι πηγές ελέγχου της θερμοκρασίας ώστε να παραμείνει σε συγκεκριμένη τιμή η θερμοκρασία και ιδανικά σταθεροποιεί τη θερμοκρασία ανεξάρτητα από την ισχύ που χρειάζεται. Ένα τέτοιο παράδειγμα είναι το ψυγείο.

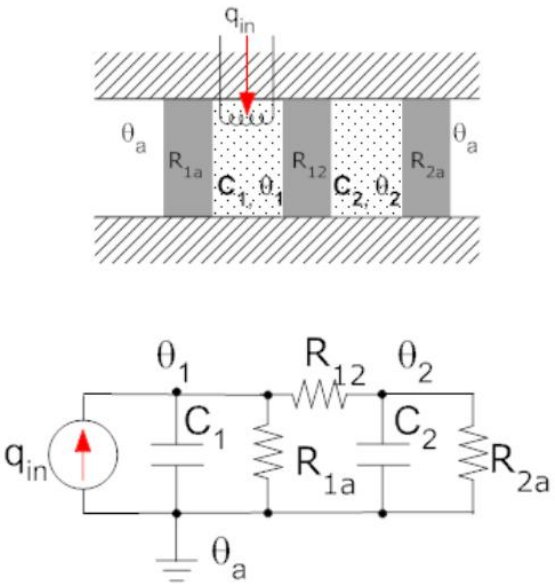
Οι πηγές ροής υγρών προσομοιώνουν όταν ρέει κάποιο υγρό συγκεκριμένης θερμοκρασίας με συγκεκριμένο ρυθμό και θερμοκρασία σε ένα σύστημα.

Από το θεώρημα διατήρησης της ενέργειας και το πρώτο νόμο της θερμοδυναμικής είναι γνωστό ότι η θερμότητα που έχει αποθηκευμένη ένα αντικείμενο ισούται με την θερμότητα που εισρέει στο αντικείμενο μείον την θερμότητα που απορρέει. [16]

Heat stored = Heat in - Heat out

Σχέση 2.10 – Εξίσωση διατήρησης θερμότητας

Ένα απλό παράδειγμα για την κατανόηση της εξίσωσης θερμότητας δίνεται πιο κάτω:



Εικόνα 2.15 - Παράδειγμα διατήρησης της θερμότητας

Στην εικόνα 2.16 στο πάνω σχήμα φαίνεται ένα ισοδύναμο που απεικονίζει δύο δωμάτια ενός σπιτιού, συμβολιζόμενα με 1 και 2. Στο ισοδύναμο έχουν προστεθεί τρείς θερμικές αντιστάσεις, μία για τον τοίχο ανάμεσα στα 2 δωμάτια (R_{12}), μία ανάμεσα στο δωμάτιο 1 και στον έξω κόσμο ($R_{1\alpha}$) και μία ανάμεσα στο δωμάτιο 2 και στον έξω κόσμο ($R_{2\alpha}$). Κάθε δωμάτιο έχει επίσης μία θερμική χωρητικότητα και μία θερμοκρασία τα οποία συμβολίζονται με C_1 και θ_1 αντίστοιχα για το δωμάτιο 1 και C_2 και θ_2αντίστοιχα για το δωμάτιο 2. Ακόμα, στο δωμάτιο 1 φαίνεται ότι υπάρχει κάποια πηγή θερμότητας που παράγει θερμότητα ίση με q_{in} ενώ στον έξω χώρο υπάρχει θερμοκρασία ίση με θ_α. Στο κάτω σχήμα φαίνονται όλα τα μεγέθη στο ισοδύναμο ηλεκτρικό κύκλωμα το οποίο προκύπτει λογικά. Για το δωμάτιο 1 η εξίσωση θερμότητας είναι η πιο κάτω:

$$Heat\ in = Heat\ out + Heat\ stored \Leftrightarrow q_{in} = \frac{\theta_1}{R_{1a}} + \frac{\theta_1 - \theta_2}{R_{12}} + C_1 \frac{d\theta_1}{dt}$$

Σχέση 2.10 – Εξίσωση δωματίου 1

Η εισερχόμενη ισχύς είναι μόνο από την πηγή θερμότητας, η θερμότητα που εκρέει από το δωμάτιο 1 οφείλεται στις 2 αντιστάσεις που εμπλέκονται στο δωμάτιο 1 και περιγράφονται με τους δύο πρώτους όρους στην εξίσωση 2.10. Ο τρίτος όρος της εξίσωσης 2.10 περιγράφει την θερμότητα που είναι αποθηκευμένη στο θερμικό πυκνωτή του συγκεκριμένου δωματίου. Αντίστοιχα, υπολογίζεται η σχέση 2.11 για το δωμάτιο 2.

$$Heat\ in = Heat\ out + Heat\ stored \Leftrightarrow \frac{\theta_1 - \theta_2}{R_{12}} = \frac{\theta_2}{R_{2a}} + C_2 \frac{d\theta_2}{dt}$$

Σχέση 2.11 – Εξίσωση δωματίου 2

Για το δωμάτιο 2, η εισερχόμενη θερμότητα οφείλεται λόγω της αντίστασης ανάμεσα στα δύο δωμάτια αφού η πηγή θερμότητας ήταν στο δωμάτιο 1. Όπως και στο δωμάτιο 1, εκρέει θερμότητα από την αντίσταση του τοίχου και αποθηκεύεται θερμότητα στον πυκνωτή του δωματίου.

2.7. Θερμικά Μοντέλα Σπιτιού

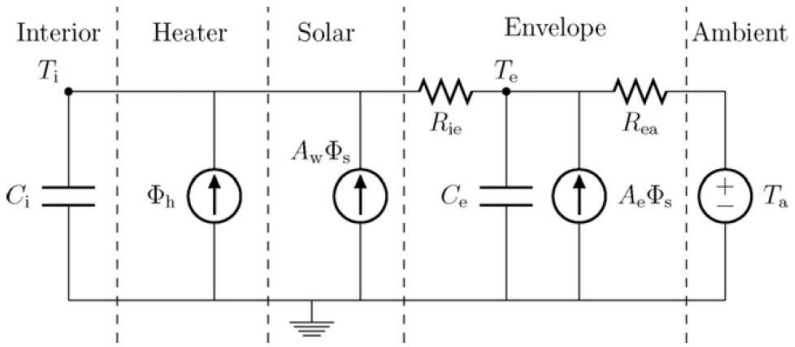
Σε περίπτωση που θέλουμε να μελετήσουμε τη συμπεριφορά της θερμότητας σε ένα σπίτι, μπορούμε να χρησιμοποιήσουμε ένα ισοδύναμο μοντέλου σπιτιού [2B] το οποίο θα προσομοιάζει το σπίτι που θέλουμε. Οι τεχνικές που μπορούν να χρησιμοποιηθούν είναι οι τρείς πιο κάτω:

- White-Box Modelling: Στη συγκεκριμένη μέθοδο χρησιμοποιείται το φυσικό μοντέλο του σπιτιού και η ροή θερμότητας προσομοιάζεται με διάφορες ατμοσφαιρικές συνθήκες. Λόγω της πολυπλοκότητας του συγκεκριμένου μοντέλου απαιτείται η χρήση έρευνας και συγκεκριμένο λογισμικό καθώς και δεδομένα για το συγκεκριμένο σπίτι.

- Black-Box Modelling: Σε αυτή τη μοντελοποίηση χρησιμοποιούνται δεδομένα από μετρήσεις και για αυτό το λόγο θεωρείται η πιο απλή. Το μειονέκτημα της συγκεκριμένης μεθόδου είναι τα μη ακριβή αποτελέσματα αφού δεν χρησιμοποιούνται πολλά δεδομένα για τη περιγραφή του σπιτιού.
- Gray-Box Modelling: Αυτή η μέθοδος είναι η ενδιάμεση των άλλων δύο μεθόδων. Χρησιμοποιεί ένα συνδυασμό των φυσικών δεδομένων του σπιτιού μαζί με στατιστικά. Ακόμα, κάνει χρήση στοχαστικών διαφορικών εξισώσεων και εξισώσεων μετρήσεων σε διακριτό χρόνο.

Στη συγκεκριμένη εργασία θα χρησιμοποιηθεί η μέθοδος του Gray-Box Modelling με τη διαφορά ότι για απλοποίηση δεν θα χρησιμοποιηθούν οι στοχαστικοί όροι. Μπορούν να χρησιμοποιηθούν διάφορα μοντέλα, από πολύ απλά μέχρι πολύ πολύπλοκα, ανάλογα με το τί επιθυμείτε να μοντελοποιηθεί και με το πόσο ακριβή αποτελέσματα επιθυμούμε. Κάθε μοντέλο μπορεί να περιγραφεί με ένα αριθμό διαφορικών εξισώσεων, ο αριθμός των οποίων εξαρτάται από την πολυπλοκότητα του μοντέλου. Το μοντέλο που θα επιλεγεί σε κάθε περίπτωση θα πρέπει να ανταποκρίνεται στην εκάστοτε πραγματικότητα και να είναι τόσο περίπλοκο όσο είναι οι πληροφορίες που ενσωματώνονται. Στη πράξη, θα πρέπει να χρησιμοποιηθούν διάφορα μοντέλα, ξεκινώντας από το πιο απλό και καταλήγοντας στο πιο περίπλοκο και να επιλεγεί αυτό που μετά από αυτό δεν υπάρχει περεταίρω βελτίωση στα αποτελέσματα.

Στην εικόνα 2.17 φαίνεται ένα ηλεκτρικό ισοδύναμο που μπορεί να περιγράψει ένα κτίριο το οποίο χρησιμοποιεί σύστημα θέρμανσης και ζεσταίνεται από την ακτινοβολία του ήλιου. Σαν envelope ορίζεται η περιοχή γύρω από το κτίριο που το απομονώνει από το περιβάλλον. Για να περιγραφεί το μοντέλο απαιτείται να γίνει χρήση κάποιων μεταβλητών κατάστασης, οι οποίες περιγράφουν την κατάσταση του συστήματος, ενός συνόλου εισόδων, οι οποίες θα επηρεάσουν το σύστημα, και ενός συνόλου διαφορικών εξισώσεων οι οποίες θα περιγράφουν τη δυναμική του συστήματος.



Εικόνα 2.16 - Μοντέλο κτιρίου

Το εσωτερικό (Interior) του κτηρίου περιγράφεται από τη θερμοκρασία T_i καθώς επίσης έχει μία θερμική χωριτικότητα C_i. Ακόμα, η θέρμανση (Heater) του χώρου γίνεται με ροή ενέργειας από το σύστημα θέρμανσης Φ_h καθώς επίσης το κτίριο θερμένεται και από την ακτινοβολία του ήλιου (Solar), η οποία έχει ροή ενέργειας Φ_s και διαπερνάει παράθυρο με εμβαδόν A_w. Ο περίγυρος του σπιτιού (Envelop) περιγράφεται από τη θερμοκρασία T_e και αποτελείται από δύο αντιστάσεις, μία

ανάμεσα στον περίγυρο και το εσωτερικό του σπιτιού R_{ie} και άλλη μία ανάμεσα στον περίγυρο και το περιβάλλον R_{ea}. Επίσης αποτελείται από μία θερμική χωριτικότητα C_e και από μία πηγή θερμότητας $A_e\Phi_s$ αφού και πάλι κάποιο μέρος της ακτινοβολίας του ήλιου εισρέει στο σπίτι από τον περίγυρο, ανάλογα τώρα με το εμβαδόν του περίγυρου. Τέλος, το περιβάλλον (Ambient) περιγράφεται σαν μία πηγή T_α αφού αυτή είναι η είσοδος του συστήματος. Αξιοσημείωτο να αναφερθεί είναι ότι στο συγκεκριμένο μοντέλο, η θερμική χωρητικότητα C_i περιγράφει τόσο τη χωρητικότητα του αέρα, όσο και τη χωρητικότητα όλων των αντικειμένων που βρίσκονται στο κτίριο. Είναι δυνατόν να υπάρξουν άλλα μοντέλα όπου η συγκεκριμένη χωρητικότητα συμβολίζει κάτι άλλο.

Οι μεταβλητές κατάστασης του κτιρίου είναι οι θερμοκρασίες T_i και T_e, αφού αυτές δίνουν την κατάσταση της θερμοκρασίας. Στο ισοδύναμο μοντέλο μπορούν να προστεθούν και άλλες μεταβλητές κατάστασης ώστε το μοντέλο να γίνει πιο κοντά στην πραγματικότητα και τα αποτελέσματα να είναι πιο ακριβές. Ένα παράδειγμα μίας μεταβλητής κατάστασης που θα μπορούσε να προστεθεί είναι να υπάρχει μία μεταβλητή κατάστασης για κάθε δωμάτιο που υπάρχει στο κτίριο και όχι μία μεταβλητή για όλο το κτίριο.

Οι είσοδοι στα διάφορα μοντέλα είναι οι μεταβλητές οι οποίες μετρήθηκαν και άρα εμπεριέχουν τα φυσικά δεδομένα. Στο μοντέλο της εικόνας 2.17 οι μεταβλητές εισόδου είναι η τιμή της εξωτερικής θερμοκρασίας T_α και η ροή ενέργειας από το σύστημα θέρμανσης Φ_h.

Η δυναμική του κτιρίου περιγράφεται από διαφορικές εξισώσεις πρώτης τάξεως και μπορούν να είναι είτε γραμμικές είτε μη γραμμικές. Αγνοώντας τη ροή ενέργειας από τον ήλιο, προκύπτουν οι δύο πιο κάτω διαφορικές εξισώσεις οι οποίες περιγράφουν το ισοδύναμο της εικόνας 2.17:

- $i = C_e \frac{dT_e}{dt}$ όπου το ρεύμα i ισούται με το ρεύμα που παράγει η πηγή τάσης T_α με την αντίσταση R_{ea}, μείον το ρεύμα που φεύγει προς την αντίσταση R_{ie}. Άρα ισχύει $i = \frac{T_a - T_e}{R_{ea}} - \frac{T_e - T_i}{R_{ie}}$ και συνεπώς η διαφορική εξίσωση είναι:

$$C_e \frac{dT_e}{dt} = \frac{1}{R_{ea}}(T_a - T_e) + \frac{1}{R_{ie}}(T_i - T_e)$$

Σχέση 2.12 – Διαφορική εξίσωση θερμοκρασίας περίγυρου

- $i = C_\iota \frac{dT_\iota}{dt}$ όπου το ρεύμα i ισούται με το ρεύμα που εισρέει από την αντίσταση R_{ie} συν το ρεύμα λόγω της θέρμανσης. . Άρα ισχύει $i = \frac{T_e - T_i}{R_{ie}} + \Phi_h$ και συνεπώς η διαφορική εξίσωση είναι:

$$C_\iota \frac{dT_\iota}{dt} = \frac{1}{R_{ie}}(T_e - T_i) + \Phi_h$$

Σχέση 2.13 – Διαφορική εξίσωση εσωτερικής θερμοκρασίας κτιρίου

Το πιο πάνω σύστημα διαφορικών εξισώσεων μπορεί επίσης να γραφεί σε μορφή πινάκων (Matrix Form) όπως φαίνεται στη σχέση 2.14:

$$\begin{bmatrix} \frac{dT_i}{dt} \\ \frac{dT_e}{dt} \end{bmatrix} = \begin{bmatrix} \frac{-1}{C_i R_{ie}} & \frac{1}{C_i R_{ie}} \\ \frac{1}{C_e R_{ie}} & \frac{-1}{C_e}\left(\frac{1}{R_{ie}} + \frac{1}{R_{ea}}\right) \end{bmatrix} \begin{bmatrix} T_i \\ T_e \end{bmatrix} + \begin{bmatrix} 0 & \frac{1}{C_i} \\ \frac{1}{C_e R_{e\alpha}} & 0 \end{bmatrix} \begin{bmatrix} T_a \\ \Phi_h \end{bmatrix}$$

Σχέση 2.14 – Διαφορικές εξισώσεις σε μορφή πινάκων

Η σχέση 2.14 μπορεί να γραφεί και σε μορφή διανυσμάτων όπως στη σχέση 2.15, όπου το διάνυσμα $\mathbf{T} = [T_i \quad T_e]^T$ είναι το διάνυσμα κατάστασης, το διάνυσμα $\mathbf{U} = [T_a \quad \Phi_h]^T$ είναι το διάνυσμα εισόδου, όπου το διάνυσμα **A** ορίζει πώς η παρούσα κατάσταση θα επηρεάσει τη δυναμική του κτιρίου και το διάνυσμα **B** ορίζει πώς οι είσοδοι εισάγονται στο σύστημα.

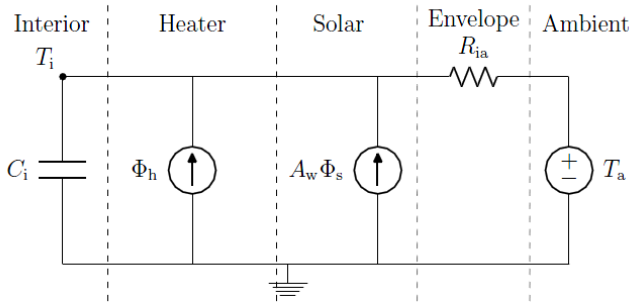
$$d\mathbf{T} = \mathbf{A}\mathbf{T}dt + \mathbf{B}\mathbf{U}dt$$

Σχέση 2.15 – Διαφορικές εξισώσεις σε διανυσματική μορφή

Το πιο πάνω σύστημα μπορεί εύκολα να λυθεί στο διακριτό χρόνο με αρχικές οριακές συνθήκες. Η απόδειξη καθώς και ο τελικός τύπος αναπτύσσονται πλήρως στη παράγραφο 3.1 του άρθρου Identifying suitable models for the heat dynamics of buildings [2B].

2.7.1. Μοντέλο Ti

Το μοντέλο Ti είναι το πιο απλό μοντέλο που μπορεί να χρησιμοποιηθεί και παρουσιάζεται στην εικόνα 2.19. Στη συγκεκριμένη εργασία ισχύει $\Phi_s = 0$ και άρα ουσιαστικά δίνεται σαν είσοδος η εξωτερική θερμοκρασία T_α και η ροή ενέργειας από το σύστημα θέρμανσης Φ_h και μετριέται η εσωτερική θερμοκρασία του κτιρίου T_i. Ο πυκνωτής C_i είναι η θερμική χωρητικότητα του κτιρίου και η αντίσταση R_{ia} είναι η θερμική αντίσταση των τοίχων του κτιρίου.



Εικόνα 2.17 - Μοντέλο Ti

Από την εικόνα 2.19, προκύπτει πολύ εύκολα η διαφορική εξίσωση που περιγράφει το συγκεκριμένο μοντέλο.

$$C_i \frac{dT_i}{dt} = \frac{1}{R_{ia}}(T_a - T_i) + \Phi_h$$

Σχέση 2.16 – Διαφορική εξίσωση μοντέλου Ti

2.7.2. Μοντέλο TiTh

Το μοντέλο παρουσιάζεται στην εικόνα 2.20 και η διαφορά του με το προηγούμενο μοντέλο είναι ότι προστίθεται η μεταβλητή κατάστασης Th, δηλαδή η θερμοκρασία του συστήματος θέρμανσης. Επίσης προστίθεται η θερμική χωρητικότητα C_h των σωμάτων θέρμανσης καθώς και μία θερμική αντίσταση R_{ih} που βρίσκεται ανάμεσα στο εσωτερικό του σπιτιού και του συστήματος θέρμανσης. Ακόμα, στη σχέση 2.17 φαίνεται το σύστημα διαφορικών εξισώσεων που περιγράφει το συγκεκριμένο μοντέλο.

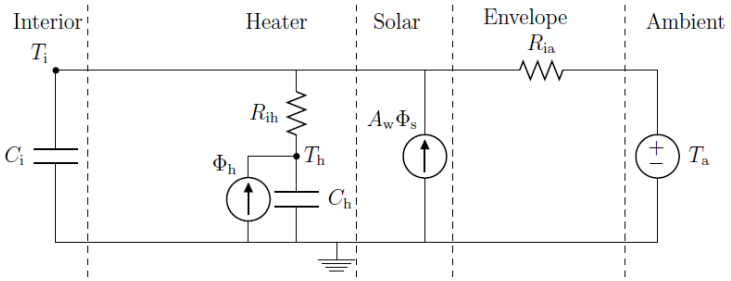


Figure B.4: RC-network diagram of *TiTh*.

Εικόνα 2.18 - Μοντέλο TiTh

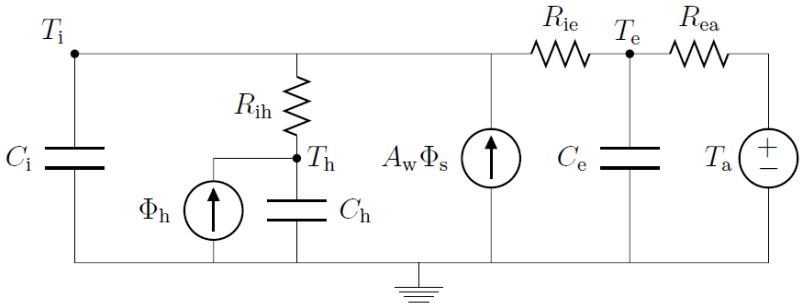
$$C_i \frac{dT_i}{dt} = \frac{1}{R_{ia}}(T_a - T_i) + \frac{1}{R_{ih}}(T_h - T_i)$$

$$C_h \frac{dT_i}{dt} = \frac{1}{R_{ih}}(T_i - T_h) + \Phi_h$$

Σχέση 2.17 – Σύστημα διαφορικών εξισώσεων μοντέλου TiTh

2.7.3. Μοντέλο TiTeTh

Η μεταβλητή κατάστασης που προστίθεται στο συγκεκριμένο μοντέλο είναι η Te, η οποία είναι η θερμοκρασία του περίγυρου του κτιρίου. Όπως φαίνεται και στην εικόνα 2.20, προστίθεται και η θερμική χωρητικότητα C_e αφού κάποια θερμότητα αποθηκεύεται και στον περίγυρο. Ακόμα η θερμική αντίσταση R_{ia} χωρίζεται πλεόν στα δύο. Η πρώτη, δηλαδή η R_{ie}, είναι η αντίσταση ανάμεσα στον περίγυρο και στο εσωτερικού του κτιρίου, ενώ η R_{ea} είναι η αντίσταση ανάμεσα στον περίγυρο και το εξωτερικό περιβάλλον. Το σύστημα των διαφορικών εξισώσεων για το συγκεκριμένο μοντέλο, φαίνεται στη σχέση 2.18.



Εικόνα 2.19 - Μοντέλο TiTeTh

$$C_i \frac{dT_i}{dt} = \frac{1}{R_{ie}}(T_e - T_i) + \frac{1}{R_{ih}}(T_h - T_i)$$

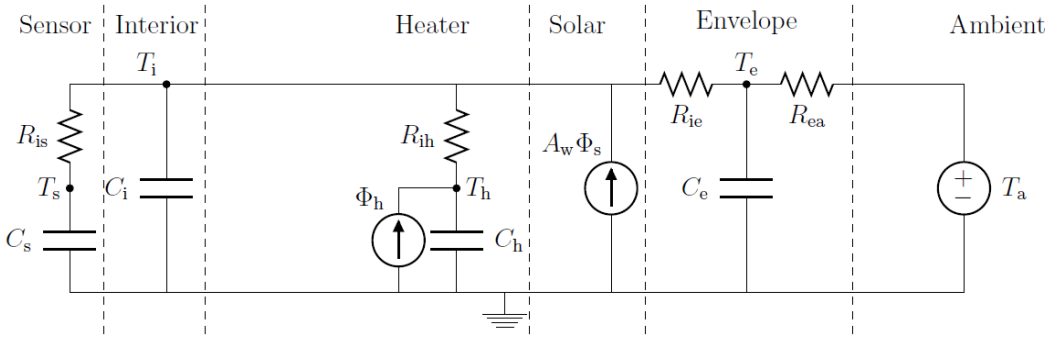
$$C_e \frac{dT_e}{dt} = \frac{1}{R_{ea}}(T_a - T_e) + \frac{1}{R_{ie}}(T_i - T_e)$$

$$C_h \frac{dT_i}{dt} = \frac{1}{R_{ih}}(T_i - T_h) + \Phi_h$$

Σχέση 2.18 – Σύστημα διαφορικών εξισώσεων μοντέλου TiTeTh

2.7.4. Μοντέλο TiTeThTs

Το συγκεκριμένο μοντέλο είναι το πιο περίπλοκο από αυτά που θα χρησιμοποιηθούν στη συγκεκριμένη εργασία, αφού προστίθεται και η μεταβλητή κατάστασης Ts για τη θερμοκρασία του αισθητήρα μέτρησης. Ανάλογα με πριν, προστέθηκε μία θερμική χωρητικότητα C_s και μία θερμική αντίσταση R_{is}. Το μοντέλο παρουσιάζεται στην εικόνα 2.21 και το σύστημα διαφορικών εξισώσεων στη σχέση 2.19.



Εικόνα 2.20 - Μοντέλο TiTeThTs

$$C_i \frac{dT_i}{dt} = \frac{1}{R_{ie}}(T_e - T_i) + \frac{1}{R_{ih}}(T_h - T_i) + \frac{1}{R_{is}}(T_s - T_i)$$

$$C_e \frac{dT_e}{dt} = \frac{1}{R_{ea}}(T_a - T_e) + \frac{1}{R_{ie}}(T_i - T_e)$$

$$C_h \frac{dT_i}{dt} = \frac{1}{R_{ih}}(T_i - T_h) + \Phi_h$$

$$C_s \frac{dT_s}{dt} = \frac{1}{R_{is}}(T_i - T_s)$$

Σχέση 2.19 – Σύστημα διαφορικών εξισώσεων μοντέλου TiTeThTs

2.8. Έξυπνα Προϊόντα

Οι καινοτομίες στις ασύρματες τεχνολογίες και στα συστήματα αίσθησης ξεκίνησαν να επιτρέπουν τη κατασκευή ψηφιακών αναπαραστάσεων σχεδόν οποιασδήποτε φυσικής οντότητας. Έτσι, κάνοντας χρήση της πληροφορικής και των τηλεπικοινωνιών κατέστη δυνατό να κατασκευαστούν νέα τεχνολογικά προϊόντα για να καλυφθούν όλες οι ανάγκες και οι απαιτήσεις του κάθε χρήστη-καταναλωτή.

Τα έξυπνα προϊόντα είναι προϊόντα που κατέχουν τα φυσικά επιτεύγματα του αρχικού προϊόντος και παράλληλα έχουν περιγραφές ψηφιακών προϊόντων. Κάποια χαρακτηριστικά αυτών των προϊόντων είναι:

- Σχεδιάζονται και κατασκευάζονται ώστε να ανταποκρίνονται στις ανάγκες του χρήστη-καταναλωτή.
- Έχουν τη δυνατότητα να προσαρμόζονται στις αντιδράσεις και στις απαιτήσεις του χρήστη-καταναλωτή.
- Έχουν τη δυνατότητα να συνδέονται, να επικοινωνούν και να αλληλεπιδρούν με άλλες συσκευές και προϊόντα.

Για την ανάπτυξη και κατασκευή έξυπνων προϊόντων, στην αγορά, απαιτείται έρευνα σε διάφορους τομείς όπως marketing, πληροφορική, οικονομικά, επικοινωνίες, καινοτομία. [17]

Οι έξυπνες συσκευές [18] είναι έξυπνα προϊόντα που είναι συνδεδεμένα με άλλες συσκευές και δίκτυα με ενσύρματο ή ασύρματο τρόπο και μπορούν να λειτουργήσουν αυτόνομα ή εώς ένα σημείο σε εξάρτηση με τις άλλες συσκευές. Κάποια παραδείγματα έξυπνων συσκευών που χρησιμοποιούνται καθημερινά είναι τα smartphones και τα tablets. Τα κυριότερα χαρακτηριστικά αυτών των συσκευών είναι:

- Χρησιμοποιούν ένα σύνολο από υλικό και λογισμικό το οποίο επιτρέπει την περάτωση διαφόρων έργων.
- Μπορεί να έχουν τη δυνατότητα φορητότητας και κινητικότητας
- Μπορούν να προσαρμοστούν ακριβώς στα χαρακτηριστικά του κάθε χρήστη

Στην ουσία, οι έξυπνες συσκευές είναι μέρος ενός έξυπνου συστήματος [19] το οποίο χρησιμοποιεί αισθήσεις και έλεγχο ώστε να περιγράψει και να αναλύσει κάποια κατάσταση. Τα έξυπνα συστήματα, μπορούν να λάβουν αποφάσεις ανάλογα με τα δεδομένα που έχουν στη διάθεσή τους είτε με κάποιο τρόπο πρόβλεψης είτε με κάποιο τρόπο που να προσαρμόζονται. Στις πιο πολλές περιπτώσεις, η εξυπνάδα του συστήματος βασίζεται σε αποφάσεις που πάρθηκαν από κάποιο κλειστό σύστημα ελέγχου, από κάποιο σύστημα ενεργειακής οικονομίας ή από τις δυνατότητες που μπορεί να προσφέρει το δίκτυο.

Τα έξυπνα συστήματα απαρτίζονται από αισθητήρες, συσκευές που ελέγχουν τη σηματοδότηση για τον έλεγχο, συσκευές που λαμβάνουν τις αποφάσεις και συσκευές οι οποίες θα αναλάβουν την πραγματοποίηση της απόφασης. Επίσης, χωρίζονται σε τρείς γενιές:

- Η πρώτη γενιά είναι συσκευές που αναγνωρίζουν αντικείμενα και πολυλειτουργικές συσκευές.
- Η δεύτερη γενιά είναι συσκευές για προηγμένα συστήματα διαχείρισης ενέργειας και για περιβαλλοντικά δίκτυα αίσθησης
- Η τρίτη γενιά είναι συσκευές που συνδυάζουν κάποια ευφυΐα και κάποιες γνωστικές λειτουργίες ώστε να παρέχουν μια διεπαφή ανάμεσα στο φυσικό και στον εικονικό κόσμο.

2.9. Γραφικό Περιβάλλον

Το γραφικό περιβάλλον χρήστη (Graphical User Interface – GUI)[20] είναι μία διεπαφή για την επικοινωνία του χρήστη με μία ηλεκτρονική συσκευή. Για να γίνει πιο εύκολη αυτή η επικοινωνία χρησιμοποιούνται εικονίδια, ενδείξεις, καθοδήγηση και άλλα ενώ επίσης υπάρχουν γραφικά περιβάλλοντα βασισμένα σε εντολές (command-line interfaces - CLIs), όπου ο χρήστης εισάγει εντολές για να φέρει εις πέρας την εργασία που θέλει να κάνει. Στη σύγχρονη εποχή, όπου χρησιμοποιούνται διάφορες ηλεκτρονικές συσκευές για διάφορους σκοπούς, έχουν αναπτυχθεί γραφικά περιβάλλοντα για συσκευές όπως τα MP3 players, οι κονσόλες παιχνιδιών, τα smarphones, τα tablets, οι ηλεκτρονικοί υπολογιστές, οι διάφορες συσκευές γραφείου και πολλά άλλα.

Η ανάπτυξη του γραφικού περιβάλλοντος χρήστη είναι ιδιαίτερα σημαντική και κατατάσσεται στην περιοχή έρευνας για αλληλεπίδραση ανθρώπου-μηχανής. Ο κύριος στόχος είναι να αναπτυχθεί ένα εύχρηστο περιβάλλον όπου ο χρήστης θα μπορεί με ευκολία να ολοκληρώσει τη διαδικασία που θέλει να κάνει. Επίσης, πρέπει να περιλαμβάνονται όλες οι πιθανές επιλογές και ενδείξεις που πιθανόν να θέλει να επιλέξει ή να δει ο χρήστης, χωρίς όμως υπάρχουν περιττές επιλογές. Περιττές επιλογές, μπορεί να αυξήσουν την περιπλοκότητα χρήσης του περιβάλλοντος και να προκαλέσουν σύγχυση στον χρήστη. Επίσης, κατά την ανάπτυξη του περιβάλλοντος πρέπει να δοθεί ιδιαίτερη προσοχή στο κοινό στο οποίο θα απευθύνεται η συγκεκριμένη συσκευή αφού μπορεί να υπάρξουν χρήστες που να μην έχουν μεγάλη εντριβή με ηλεκτρονικές συσκευές.

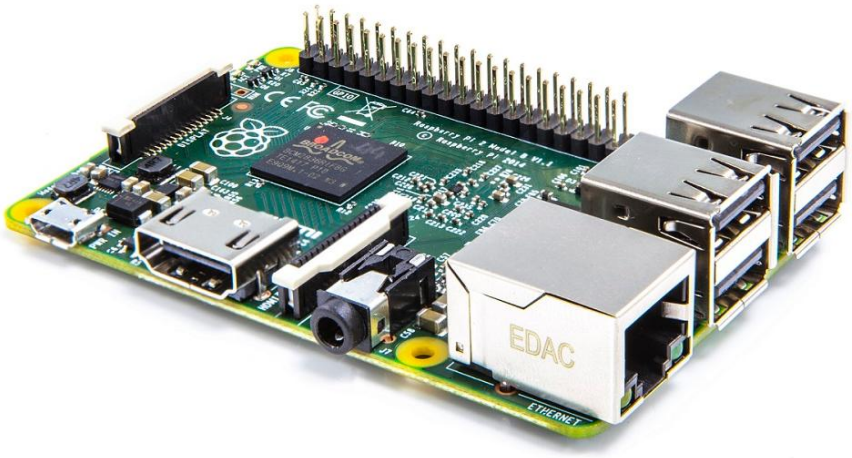
Στη συγκεκριμένη εργασία είναι απαραίτητο να αναπτυχθεί ένα γραφικό περιβάλλον ώστε ο χρήστης να μπορεί να εισάγει το εβδομαδιαίο πρόγραμμα θέρμανσης που επιθυμεί και για να μπορεί να παρακολουθεί κάποια στοιχεία και στατιστικά. Ακόμα, το γραφικό περιβάλλον θα αναπτυχθεί με τέτοιο τρόπο ώστε να μπορεί να είναι προσβάσιμο και από άλλες ηλεκτρονικές συσκευές όπως smartphones και tablets για να διευκολύνει περισσότερο τον χρήστη.

ΚΕΦΑΛΑΙΟ 3. ΣΧΕΔΙΑΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ

Στο συγκεκριμένο κεφάλαιο, θα παρουσιαστεί πιο αναλυτικά το υλικό που επιλέχθηκε καθώς και ο τρόπος που συνδέθηκαν μεταξύ τους όλα τα μέρη. Επίσης, παρουσιάζονται οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της διαδικτυακής εφαρμογής.

3.1. Raspberry Pi 2

Το Raspberry Pi 2, είναι ένα από τα πιο αποδοτικά μοντέλα καθώς έχει 900MHz quad-core CPU, 1GB SDRAM, 4 USB ports, full HDMI port, Ethernet port, micro SD card slot και 40 GPIO pins.[7]



Εικόνα 3.1 - Raspberry Pi 2

Με τη βοήθεια των GPIO pins, μπορούμε να συνδέσουμε στο Raspberry Pi διάφορα πρόσθετα όπως οθόνη αφής, κεραία Wi-Fi, κάμερα, αισθητήρες και άλλα. Τελικά έχουμε τη δυνατότητα να κατασκευάσουμε από ένα από web server μέχρι και ένα βιντεοπαιχνίδι ή ακόμα και ένα ρομπότ.

Στη συγκεκριμένη εργασία, χρησιμοποιούμε τα GPIO pins για να συνδέσουμε αισθητήρα θερμοκρασίας, ο οποίος θα μας παρέχει την δυνατότητα παρακολούθησης της θερμοκρασίας του σπιτιού. Ακόμα, θα χρησιμοποιηθεί σαν web server για τη φιλοξενία της web εφαρμογής.

3.1.1. Raspbian

Το λειτουργικό σύστημα που χρησιμοποιείται κατά κύριο λόγο στο Raspberry Pi είναι το Raspbian, το οποίο είναι βασισμένο στο λειτουργικό σύστημα Debian. Το Raspbian είναι λογισμικό ανοιχτού κώδικα και προσφέρει τις ίδιες δυνατότητες με τα υπόλοιπα UNIX λειτουργικά συστήματα.

3.2. Αισθητήρας Θερμοκρασίας: DS18B20

Τα βασικά προτερήματα του συγκεκριμένου αισθητήρα είναι ότι είναι αδιάβροχος, ότι έχει μεγάλος εύρος μέτρησης θερμοκρασίας (-55°C μέχρι +125°C) και έχει ακρίβεια ±0.5°C για θερμοκρασίες -10°C εώς +85°C. Ακόμα, χρησιμοποιεί τεχνολογία 1-wire.

Τα 3 καλώδια που χρησιμοποιεί είναι τάση, γείωση και δεδομένα.

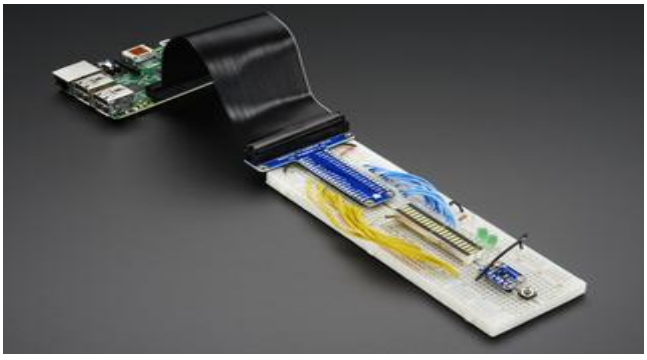
Εικόνα 3.2 - Αισθητήρας DS18B20

3.3. Υλοποίηση θερμομέτρου

Για να κατασκευαστεί το θερμόμετρο απαιτείται διαδικασία τόσο στο κομμάτι του υλικού, όσο και στο κομμάτι του λογισμικού. Πιο κάτω αναλύονται και οι δύο διαδικασίες.

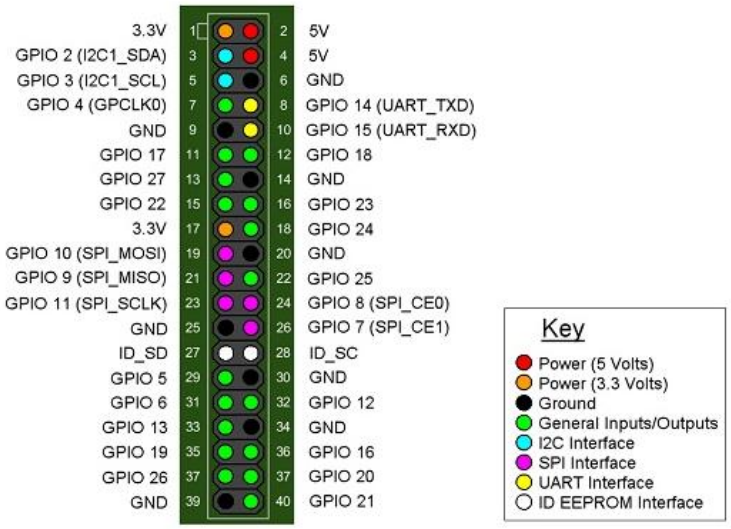
3.3.1. Υλικό

Για τη σύνδεση του αισθητήρα στο Raspberry Pi, είναι απαραίτητα τα GPIO pins. Επίσης είναι απαραίτητο ένα breadboard, πάνω στο οποίο θα γίνουν οι διάφορες συνδέσεις. Οι συνδέσεις πάνω στα GPIO, μπορούν να γίνουν απευθείας με καλώδια στα pins ή μπορεί να χρησιμοποιηθεί κάποιο πρόσθετο που θα επεκτείνει τα pins από πάνω στο Raspberry Pi, πάνω στο breadboard. Η μέθοδος που χρησιμοποιήθηκε στη προκειμένω περίπτωση ήταν η δεύτερη, χωρίς όμως να υπάρχει κάποια διαφορά από την πρώτη.



Εικόνα 3.3 - Επέκταση των GPIO pins στο breadboard

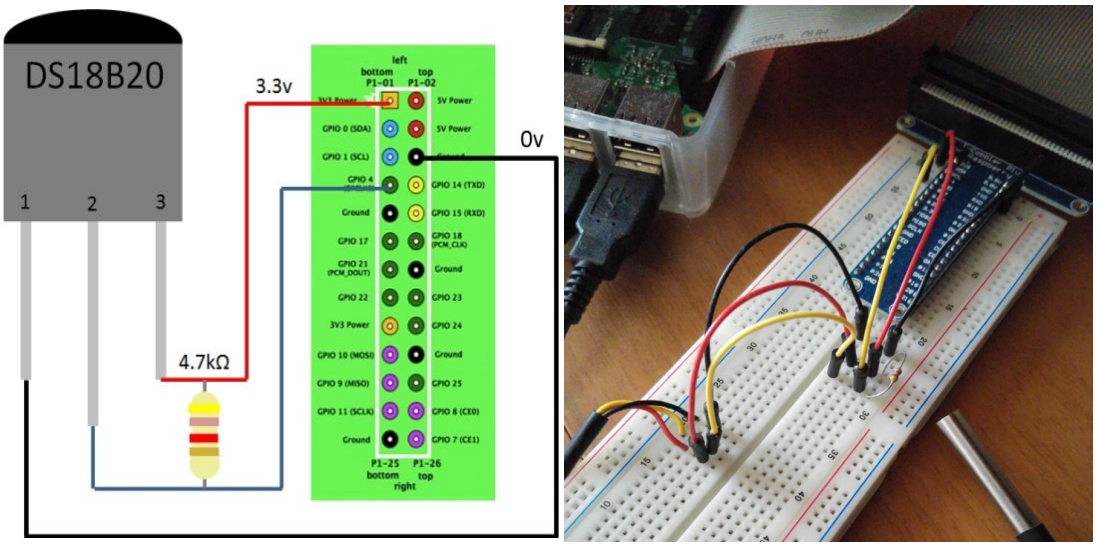
Απαραίτητο είναι επίσης να κατανοηθούν τα GPIO pins για να γίνει σωστά η συνδεσμολογία. Πιο κάτω φαίνονται στο σχήμα τα GPIO pins με τη χρήση τους.



Εικόνα 3.4 - GPIO pins Raspberry Pi 2

Λόγω του ότι ο αισθητήρας μπορεί να λειτουργήσει από 3-5.5V, επιλέγουμε να χρησιμοποιήσουμε το pin1 με τα 3.3V για την τροφοδοσία. Επιλέγεται επίσης το pin7 σαν η είσοδος για τα δεδομένα, αφού το συγκεκριμένο pin χειρίζεται γενικές εισόδους/εξόδους. Τέλος θα χρησιμοποιηθεί και το pin6 για τη γείωση.

Επίσης, ο αισθητήρας χρησιμοποιεί μία αντίσταση η οποία ονομάζεται pullup και η οποία πρέπει να παρεμβάλλεται μεταξύ της γραμμής δεδομένων και της τάσης. Σκοπός αυτής της αντίστασης είναι για να σταθεροποιεί τη μετάδοση των δεδομένων. Στη περίπτωση που επιθυμείτε η παράλληλη σύνδεση και άλλων 1-wire αισθητήρων, πρέπει να χρησιμοποιηθούν τα ίδια pins και μόνο μία αντίσταση [21]. Στη συγκεκριμένη εργασία, χρησιμοποιήθηκε μόνο ένας αισθητήρας. Τελικά το κύκλωμα που πρέπει να υλοποιηθεί είναι το πιο κάτω:



Εικόνα 3.5 - Κύκλωμα Θερμομέτρου (A) θεωρητικό, (B) πρακτικό

3.3.2. Λογισμικό

Αρχικά συνδεόμαστε με ssh στο μηχάνημα, στην διεύθυνση IP που του δόθηκε μέσω dhcp client. Τη συγκεκριμένη IP μπορούμε να τη δούμε από το router και στην προκειμένω περίπτωση είναι η 192.168.1.2. Το username και password που ζητούνται είναι pi και raspberry αντίστοιχα και μπορούν να αλλάξουν μέσω αργότερης ρύθμισης στο μηχάνημα.

Στη συνέχεια, πρέπει να ενημερώσουμε με τις τελευταίες αναβαθμίσεις το Raspberry Pi, το οποίο γίνεται εκτελώντας τις παρακάτω εντολές:

1. sudo apt-get update
2. sudo apt-get upgrade

Μετά, μπορεί να γίνει η εγκατάσταση του 1-wire αισθητήρα με τα εξής βήματα[22]: Ανοίγουμε το αρχείο /boot/config.txt και στο τέλος προσθέτουμε

dtoverlay = w1-gpio.

1. Κάνουμε επανεκκίνηση το μηχάνημα με την εντολή sudo reboot.
2. Όταν ξανανοίξει το μηχάνημα εκτελούμε τις εντολές
   - sudo modprobe w1-gpio
   - sudo modprobe w1-therm
   - cd /sys/bus/w1/devices
   - ls
3. Στην οθόνη θα εμφανιστεί ένας φάκελος με όνομα 28-X όπου στην περίπτωσή μας είναι X = 00000670834d. Τότε, εκτελούμε την εντολή cd ./28-00000670834d.
4. Τέλος, εκτελούμε την εντολή cat w1_slave και βλέπουμε το παρακάτω αποτέλεσμα:

```
pi@raspberrypi /sys/bus/w1/devices/28-00000670834d $ cat w1_slave
76 01 4b 46 7f ff 0a 10 79 : crc=79 YES
76 01 4b 46 7f ff 0a 10 79 t=23375
```

Εικόνα 3.6 - Λειτουργία αισθητήρα

Από αυτό το αποτέλεσμα καταλαβαίνουμε λόγω του YES ότι ο αισθητήρας λειτουργεί κανονικά και ότι τη συγκεκριμένη χρονική στιγμή είχε κάνει μέτρηση ίση με 23375/1000 °C = 23.375°C.

3.4. Ανάπτυξη web Εφαρμογής

Στην web εφαρμογή, ο χρήστης θα έχει την δυνατότητα να διαμορφώνει το πρόγραμμα που επιθυμεί για τη διαχείριση των θερμοκρασιών του σπιτιού και θα μπορεί να παρακολουθεί διάφορα στατιστικά τόσο για τη θέρμανση του σπιτιού όσο και για την πρόβλεψη του καιρού τις επόμενες μέρες και ώρες. Τα βασικά βήματα για την ανάπτυξη της εφαρμογής μαζί με τις τεχνικές και τις διαδικασίες που χρησιμοποιήθηκαν, αναλύονται στην παρούσα παράγραφο.

3.4.1. Web Server

Αρχικά πρέπει να φτιαχτεί ένας web server στον οποίο θα φιλοξενείται η εφαρμογή, όπου στη συγκεκριμένη περίπτωση, το μηχάνημα που θα φιλοξενεί την εφαρμογή θα είναι το ίδιο το Raspberry Pi. Εναλλακτικά, θα μπορούσε να χρησιμοποιηθεί κάποιο Virtual Machine(VM) ώστε να μην σπαταλιέται χώρος στο Raspberry Pi.

3.4.2. Apache Server

Στον web server όπου θα φιλοξενείται η εφαρμογή, θα εγκατασταθεί ένας Apache Server. Για την εγκατάσταση ακολουθούμε τα βήματα:

1. Εκτελούμε την εντολή sudo apt-get install apache2 –y
2. Εάν δεν είναι γνωστή ήδη η διεύθυνση IP του Raspberry Pi, τότε εκτελούμε την εντολή hostname –I για να την δούμε.
3. Στον browser δίνουμε σα διεύθυνση την IP του Raspberry Pi και παρατηρούμε ότι φορτώνεται μία σελίδα που εμφανίζει το μήνυμα It Works! όπως πιο κάτω και έτσι καταλαβαίνουμε ότι έχει γίνει σωστά η εγκατάσταση.

Εικόνα 3.7 - Λειτουργία apache server

4. Πηγαίνουμε στο φάκελο /var/www/html και διαγράφουμε το index.html ώστε να προσθέσουμε μετά τη δικιά μας αρχική σελίδα.

3.4.3. mySQL

Η εφαρμογή παρέχει την δυνατότητα εισαγωγής εβδομαδιαίου προγράμματος θέρμανσης του σπιτιού. Για κάθε μέρα ο χρήστης μπορεί να εισάγει διαφορετικό πρόγραμμα, με διαφορετικό αριθμό διαστημάτων και για αυτό το λόγο προτιμήθηκε η αποθήκευση των δεδομένων αυτών σε μια βάση δεδομένων SQL. Για την εγκατάσταση της συγκεκριμένης βάσης δεδομένων εκτελούμε την πιο κάτω διαδικασία.

1. Εκτελούμε την εντολή sudo apt-get install mysql-server. Κατά την εκτέλεση της εντολής ζητείται ένα password για τον χρήστη root, ο οποίος τέθηκε 1234.
2. Μετά εκτελούμε την εντολή sudo apt-get install php5-mysql για να εγκατασταθούν οι αναγκαίες βιβλιοθήκες της php για την mysql.
3. Για να ανοίξουμε το prompt της mysql με δικαιώματα χρήστη root, εκτελούμε την εντολή mysql –p –u root.
4. Τέλος, απαιτείται να δημιουργήσουμε ένα νέο χρήστη ο οποίος θα έχει περιορισμένα δικαιώματα σε σχέση με τον root ούτως ώστε να μην μπορεί να τροποποιήσει κάποιες ρυθμίσεις. Για την δημιουργία του χρήστη αυτού, ορίζουμε ως username και password τα project και 1111 αντίστοιχα και εκτελούμε στο prompt της mysql την εντολή CREATE USER ‘project’@’localhost’ IDENTIFIED BY ‘1111’;

3.4.4. phpMyAdmin

Για την ευκολότερη επεξεργασία της βάσης δεδομένων που δημιουργήσαμε, θα κάνουμε εγκατάσταση του phpMyAdmin, το οποίο μας παρέχει γραφικό περιβάλλον επεξεργασίας.

Αρχικά, πρέπει να γίνει η εγκατάσταση, η οποία γίνεται με την εντολή sudo apt-get install phpmyadmin. Κατά τη διάρκεια της εγκατάστασης ζητείται να επιλεγεί ο server που θα χρησιμοποιηθεί και επιλέγουμε Apache2. Έπειτα ζητείται το password (του root) του συγκεκριμένου server, που στην περίπτωσή μας είναι το 1234.

Στη συνέχεια πρέπει να εξασφαλίσουμε ότι ο phpmyadmin θα λειτουργεί σωστά μαζί με τον Apache Server και έτσι εκτελούμε τα δύο πιο κάτω βήματα:

1. Εκτελούμε sudo nano /etc/apache2/apache2.conf και προσθέτουμε στην τελευταία γραμμή του αρχείου Include /etc/phpmyadmin/apache.conf
2. Κάνουμε επανεκκίνηση του apache server εκτελώντας την εντολή sudo /etc/init.d/apache2 restart για να φορτώσει το νέο τροποποιημένο αρχείο.

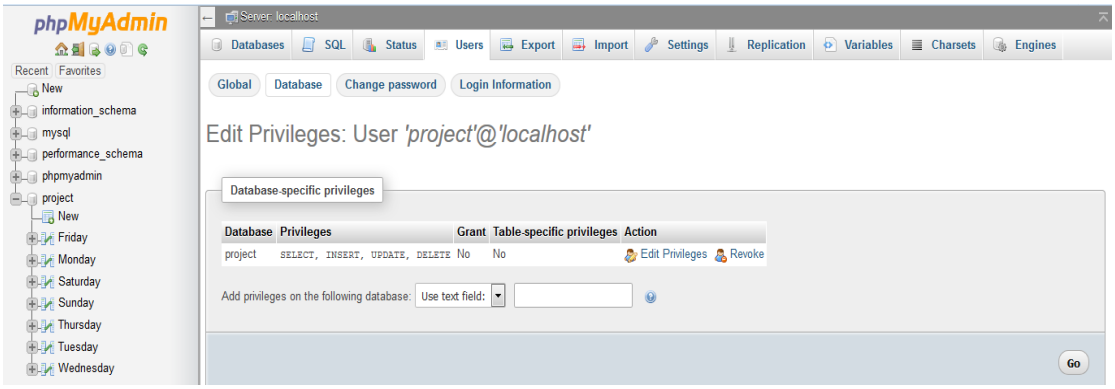
Έπειτα, αφού κάνουμε πρώτα login στο http://localhost/phpmyadmin/ (στη θέση του localhost πρέπει να μπει η διεύθυνση IP του web server αν το login γίνεται από απομακρυσμένη συσκευή) σαν root, δημιουργούμε την βάση δεδομένων από το μενού Databases. Κάτω από το πεδίο create database, της δίνουμε το όνομα project και πατάμε το κουμπί create. Η νέα βάση που δημιουργήθηκε εμφανίζεται πλέον στο αριστερό μενού, όπου επιλέγοντάς την μπορούμε να την τροποποιήσουμε κατάλληλα. Για την δημιουργία πινάκων, υπάρχει η περιοχή create table όπου μπορούμε να συμπληρώσουμε το όνομα του πίνακα και των αριθμό των στηλών που θα περιέχει. Στη δικιά μας περίπτωση, πρέπει να δημιουργηθούν 7 πίνακες -ένας πίνακας ανά ημέρα- που ο καθένας θα έχει 3 στήλες -αρχή και τέλος χρονικού διαστήματος και επιθυμητή θερμοκρασία.

	#	Name	Type	Collation	Attributes	Null	Default	Extra
☐	1	**From**	time			No	*None*	
☐	2	**To**	time			No	*None*	
☐	3	**Temperature**	int(11)			No	*None*	

Εικόνα 3.8 - Μορφή πίνακα για κάθε ημέρα της εβδομάδας

Για τα πεδία From και To δώσαμε σαν τύπο το time ώστε να μπορεί ο χρήστης να εισάγει δεδομένα σε μορφή χρόνου(δηλαδή στη μορφή HH:MM:SS) ενώ στο Temperature ορίσαμε να έχει τύπο integer ώστε να μπορεί να εισάγει ο χρήστης ένα ακέραιο αριθμό σαν την επιθυμητή θερμοκρασία.

Τέλος πρέπει να δοθούν κατάλληλα δικαιώματα στη βάση για τον χρήστη που δημιουργήθηκε πιο πριν. Επιλέγοντας την καρτέλα USERS μπορούμε να δούμε όλους τους χρήστες που έχουν πρόσβαση στη βάση δεδομένων που φτιάξαμε. Σε αυτή τη λίστα λοιπόν, βρίσκουμε τον χρήστη με το όνομα project και πατάμε στο edit privileges. Στη συνέχεια, πατάμε από το μενού την καρτέλα Database και δίνουμε στη βάση που δημιουργήσαμε πριν δικαιώματα SELECT, INSERT, UPDATE και DELETE και πατάμε GO.



Εικόνα 3.9 - Αλλαγή δικαιωμάτων

3.4.5. RRDTool

Το Round Robin Database Tool (RRDTool) [23] είναι μια βάση δεδομένων ανοιχτού κώδικα, στην οποία μπορούμε να αποθηκεύσουμε δεδομένα σε σχέση με τον χρόνο. Επίσης, οι βάσεις που δημιουργούνται έχουν την ιδιότητα να έχουν σταθερό μήκος και μπορούμε εύκολα να παράγουμε γραφικές παραστάσεις για συγκεκριμένα χρονικά διαστήματα για να δούμε τα δεδομένα. Τέλος, υποστηρίζεται εύκολη ενσωμάτωση του εργαλείου σε διάφορες δημοφιλής γλώσσες προγραμματισμού όπως shell scripts, python και ruby με αποτέλεσμα να είναι πολύ εύχρηστο.

Ένα βασικό χαρακτηριστικό του RRDTool είναι ο χρόνος καταγραφής της μέτρησης που θα εισαχθεί στη βάση. Κατά τη δημιουργία της βάσης ορίζουμε ένα time-slot και έτσι καταχωρείται μέτρηση από τη πηγή δεδομένων (DS) στη βάση ανά συγκεκριμένα διαστήματα. Ακόμα, για εξοικονόμηση χώρου, μπορεί να οριστεί ενοποίηση των δεδομένων μετά από συγκεκριμένο χρονικό διάστημα καθώς επίσης μπορεί να οριστεί και η συνάρτηση(CF) με την οποία θα γίνει η ενοποίηση (average, minimum, maximum και last). Τα δεδομένα που ενοποιούνται με τον ίδιο τρόπο αποθηκεύονται στα Round Robin Archives (RRA) και κάθε βάση μπορεί να έχει πέραν των ενός RRA. Με αυτό το τρόπο, το συγκεκριμένο εργαλείο προσφέρει εύκολη αποθήκευση δεδομένων και στατιστικών.

Ένα παράδειγμα ορισμού μιας βάσης με RRDTool αναλύεται πιο κάτω[24]:

> *rrdtool create temperature.rrd –start now –step 300 DS:temp:GAUGE:600:-50:50 RRA:AVERAGE:0.5:1:288 RRA:AVERAGE:0.5:12:168 RRA:AVERAGE:0.5:12:720*
>
> *Το όνομα της βάσης που θα δημιουργηθεί ορίζεται ως temperature.rrd και ξεκινά να λαμβάνει μετρήσεις απευθείας μετά την δημιουργία της. Ορίζονται ως time-slot τα 300sec, δηλαδή τα 5 λεπτά ενώ η πηγή δεδομένων temp ορίζεται ότι είναι GAUGE, ότι δηλαδή έχει κάποια αρχή και τέλος στις τιμές που μπορεί να λάβει. Επίσης, σε σχέση με την πηγή ορίζεται το heartbeat 600sec και άρα αν δεν ληφθεί κάποια μέτρηση σε αυτά τα 10λεπτά, εισάγεται τιμή UNKNOWN καθώς επίσης ορίζεται ότι η ελάχιστη τιμή που μπορεί να ληφθεί είναι το -50 και η μέγιστη το 50. Στη συνέχεια ορίζονται 3 διαφορετικά RRA. Στο πρώτο RRA αποθηκεύονται οι θερμοκρασίες για μία ημέρα αφού χρησιμοποιείται μία τιμή από αυτές που πάρθηκαν για να αποθηκευτεί και υπάρχουν 288 εγγραφές (1*300*288=86400sec=1440min=24hr).*

*Στο δεύτερο RRA αποθηκεύονται οι θερμοκρασίες για μια εβδομάδα αφού χρησιμοποιούνται 12 τιμές και υπάρχουν 128 εγγραφές (12*300*168=604800sec=10080min=168hr=7days). Αντίστοιχα για το τρίτο RRA υπολογίζεται 12*300*720=2592000sec=43200min=720hr=30days και άρα υπολογίζεται ο μέσος όρος θερμοκρασιών για ένα μήνα. Η τιμή 0.5 που εμφανίζεται και τα 3 RRA, ορίζει ποιό μέρος του ενοποιημένου τμήματος θα λάβει τιμή για τη περίπτωση που υπάρχει UNKNOWN.*

Η rrdtool create είναι μόνο μία από τις εντολές που υποστηρίζει το συγκεκριμένο εργαλείο. Άλλες εντολές που υπάρχουν είναι η update με την οποία μπορούμε να εισάγουμε δεδομένα στη βάση για τη συγκεκριμένη χρονική στιγμή, η fetch με την οποία μπορούμε να πάρουμε κάποια δεδομένα και η graph με την οποία μπορούμε να λάβουμε γραφική αναπαράσταση των δεδομένων μας.

Για την εγκατάσταση του rrdtool απαιτούνται να εκτελεστούν οι πιο κάτω[25]:

1. export BUILD_DIR =/tmp/rrdbuild
2. export INSTALL_DIR=/opt/rrdtool-1.4.5
3. mkdir -p $BUILD_DIR
4. cd $BUILD_DIR
5. sudo apt-get install libpango1.0-dev libxml2-dev
6. wget http://oss.oetiker.ch/rrdtool/pub/rrdtool-1.4.5.tar.gz
7. gunzip -c rrdtool-1.4.3.tar.gz | tar xf -
8. cd rrdtool-1.4.5
9. ./configure --prefix=$INSTALL_DIR && make && make install

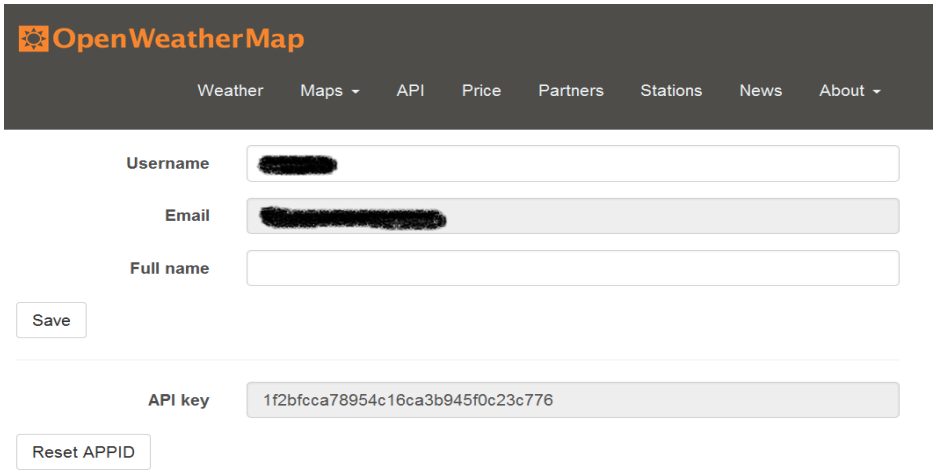
Το rrdtool χρησιμοποιήθηκε τόσο για την αποθήκευση των θερμοκρασιών του σπιτιού από τον αισθητήρα όσο και για την αποθήκευση των εξωτερικών θερμοκρασιών.

3.4.6. OpenWeatherMap API

Για την καταγραφή της εξωτερικής θερμοκρασίας δεν χρησιμοποιήθηκε αισθητήρας αλλά το OpenWeatherMap API (Application Programming Interface). Με τη βοήθεια του API, μπορούμε να κάνουμε διάφορα αιτήματα στο openweathermap.org και αυτό να μας παρέχει όλες τις πληροφορίες που χρειαζόμαστε. Εκτός από την συγκεκριμένη υπηρεσία, υπάρχουν και διάφορες άλλες υπηρεσίες οι οποίες παρέχουν διάφορα στατιστικά για τον καιρό και μπορούν να χρησιμοποιηθούν. Η συγκεκριμένη επιλέχθηκε γιατί προσφέρει δωρεάν τις υπηρεσίες της, έχει σταθμούς μέτρησης θερμοκρασιών σε πολλά σημεία της Αθήνας και προσφέρει επίσης πρόβλεψη των καιρικών συνθηκών για τις επόμενες 5 ημέρες.

Για να κάνουμε χρήση του συγκεκριμένου API πρέπει να ακολουθηθεί η παρακάτω διαδικασία[26][27]:

1. Κάνουμε επίσκεψη στην ιστοσελίδα http://openweathermap.org/ και πατάμε πάνω αριστερά το κουμπί Sign Up για να δημιουργήσουμε το λογαριασμό χρήστη. Αφού ολοκληρωθεί η διαδικασία, δίνεται το API key το οποίο θα χρησιμοποιείται κάθε φορά που θα γίνεται κλήση στην υπηρεσία.



Εικόνα 3.10 - API key

2. Από το μενού Maps -> Weather maps, μπορούμε να δούμε όλους τους σταθμούς ανά το παγκόσμιο και να βρούμε τον πιο κοντινό στη περιοχή, ο οποίο στη περίπτωσή μας είναι στο Βύρωνα. Έπειτα, κάνουμε επίσκεψη στην ιστοσελίδα http://bulk.openweathermap.org/sample/ και ανοίγουμε το αρχείο city.list.json.gz στο οποίο μπορούμε να βρούμε διάφορα χαρακτηριστικά του κάθε σταθμού. Το χαρακτηριστικό που μας ενδιαφέρει για να κάνουμε τις κλήσεις είναι η ταυτότητα (id) και για τον σταθμό στο Βύρωνα η ταυτότητα είναι 251948.
3. Έχοντας το ID του σταθμού που θα χρησιμοποιήσουμε μαζί με το API key, μπορούμε να κάνουμε διάφορες κλήσεις και να λάβουμε τα στοιχεία που επιθυμούμε. Οι κλήσεις που θα γίνονται είναι της μορφής: http://api.openweathermap.org/data/2.5/weather?id=251948&APPID=1f2bfcca78954c16ca3b945f0c23c776 και επιστρέφουν το αποτέλεσμα σε μορφή JSON από προεπιλογή.


```
{"coord":{"lon":23.75,"lat":37.97},"weather":
[{"id":500,"main":"Rain","description":"light rain","icon":"10d"}],"base":"cmc
stations","main":
{"temp":292.819,"pressure":1013.43,"humidity":67,"temp_min":292.819,"temp_max":292.81
9,"sea_level":1027.84,"grnd_level":1013.43},"wind":
{"speed":2.31,"deg":16.5022},"rain":{"3h":0.3275},"clouds":
{"all":76},"dt":1462530784,"sys":
{"message":0.0035,"country":"GR","sunrise":1462504959,"sunset":1462555262},"id":25194
8,"name":"Vyronas","cod":200}
```


Εικόνα 3.11 - Αποτέλεσμα κλήσης OpenWeatherMap API

4. Με παρόμοιο τρόπο του βήματος 3 γίνονται και οι κλήσεις για την πρόβλεψη του καιρού. Η συγκεκριμένη υπηρεσία παρέχει πρόβλεψη για τις επόμενες 5 ημέρες με ανανέωση πρόβλεψης κάθε 3 ώρες, καθώς και πρόβλεψη για τις επόμενες 16 ημέρες με ανανέωση πρόβλεψης κάθε 1 ημέρα.

3.4.7. Crontab

Το crontab είναι εργαλείο για χρονοδρομολόγηση διαδικασιών, όπου μπορούμε να θέσουμε κάθε πόσο χρόνο θέλουμε να τρέχουν στο υπόβαθρο κάποια scripts. Στη δικιά μας περίπτωση θέλουμε να τρέχουν κάθε 1 λεπτό τα scripts για την μέτρηση της εσωτερικής θερμοκρασίας από τον αισθητήρα και της κλήσης στο API για την εξωτερική θερμοκρασία. Ακόμα, επειδή θέλουμε να τρέχουν τα συγκεκριμένα scripts στον apache server που δημιουργήθηκε πιο πριν, πρέπει να τροποποιήσουμε το crontab του χρήστη www-data, ο οποίος είναι ο προκαθορισμένος χρήστης για τον apache server. Η κλήση άρα που θα γίνει είναι sudo crontab –u www-data –e και τότε θα ανοίξει το αρχείο όπου θα προσθέσουμε τις εντολές για να τρέξουν τα scripts που επιθυμούμε.

```
# ┌──────────── min (0 - 59)
# │ ┌────────── hour (0 - 23)
# │ │ ┌──────── day of month (1 - 31)
# │ │ │ ┌────── month (1 - 12)
# │ │ │ │ ┌──── day of week (0 - 6) (0 to 6 are
Sunday to Saturday, or use names; 7 is Sunday, the same as 0)
# │ │ │ │ │
# │ │ │ │ │
# * * * * *  command to execute
```

Εικόνα 3.12 - Μορφή εντολών στο crontab

Οι εντολές που προσθέτουμε στο crontab πρέπει να έχουν τη συγκεκριμένη μορφή που φαίνεται στην εικόνα 3.12 ώστε να ορίζεται σωστά ο χρόνος επανάληψης της κάθε εντολής. Δεδομένου ότι θέλουμε να τρέχουν τα scripts κάθε 1 λεπτό, οι εντολές θα είναι της μορφής */1 * * * * εντολή.

Για την ορθή λειτουργεία του crontab, υπάρχουν 3 σημεία τα οποία πρέπει να ληφθούν υπόψιν:

1. Αφού ολοκληρωθεί η κάθε εντολή, ο crontab daemon αποστέλλει στο χρήστη μέσω email το αποτέλεσμα της ολοκλήρωσης και ειδοποιήσεις. Για να ληφθούν αυτά τα email θα πρέπει να υπάρχει εγκατεστημένος ένας mail server ο οποίος θα έχει Mail Transfer Agent (MTA). Η συγκεκριμένη έκδοση των Raspbian δεν έχει προεγκατεστημένο τον mail server και άρα πρέπει να εγκατασταθεί με την εντολή sudo apt-get install postfix.
2. Το crontab που τροποποιήθηκε αφορά το χρήστη www-data και άρα όλες οι εντολές που θα εκτελεστούν θα εκτελεστούν από τον συγκεκριμένο χρήστη, δηλαδή θα είναι της μορφής sudo –u www-data command. Για να μπορέσει ωστόσο ο www-data να τις εκτελέσει θα πρέπει να έχει τα κατάλληλα δικαιώματα σε όλα τα επιμέρους αρχεία και φακέλους που εμπλέκονται. Για να δώσουμε τα δικαιώματα που απαιτούνται, πρέπει να τρέξουμε για κάθε αρχείο και φάκελο τις παρακάτω δύο εντολές:
    i. sudo chmod +x path
    ii. sudo chown www-data:www-data path

Η πρώτη εντολή δίνει δικαιώματα execution ενώ η δεύτερη δίνει δικαιώματα στον χρήστη www-data.

3. Αφού λειτουργήσει για κάποιο χρονικό διάστημα(π.χ. 5 λεπτά) ο crontab, πρέπει να επαληθευτεί η σωστή λειτουργία του από τα logs που αποθηκεύει. Για να δούμε τα logs, εκτελούμε την εντολή grep CRON /var/log/syslog. Εάν η λειτουργία είναι σωστή, στο αποτέλεσμα της πιο πάνω εντολής, βλέπουμε την κάθε εντολή να εκτελείται κάθε 1 λεπτό με την βοήθεια των timestamps που υπάρχουν.

3.4.8. HighCharts

Το HighCharts [28] είναι βιβλιοθήκες γραμμένες σε κώδικα JavaScript που αποσκοπούν στην παρουσίαση διαφόρων αποτελεσμάτων σε μορφή διαδραστικών γραφικών παραστάσεων σε μια ιστοσελίδα και παρέχεται δωρεάν για μη κερδοσκοπικούς σκοπούς. Λόγω του ότι είναι πολύ εύκολο στη χρήση, είναι ευέλικτο, παρέχει μια τεράστια γκάμα για παρουσίαση αποτελεσμάτων και είναι διαδραστικό, έχει γίνει ιδιαίτερα δημοφιλές και χρησιμοποιείται από μεγάλες παγκόσμιες εταιρίες όπως οι facebook, tweeter, yahoo, visa, nokia και πολλές άλλες.

Οι γραφικές παραστάσεις που μπορούν να φτιαχτούν, μπορεί να είναι απλές γραμμές (lines), μπάρες (bar), περιοχές (areas), πίτες(pie) και πολλές άλλες. Επίσης είναι δυνατός ο συνδυασμός διαφορετικών παραστάσεων στο ίδιο γράφημα καθώς επίσης υπάρχει η δυνατότητα πολλαπλών αξόνων με αποτέλεσμα να παρέχεται δυνατότητα προσαρμογής σε κάθε είδους ανάγκης. Στην ιστοσελίδα http://www.highcharts.com/demo, υπάρχουν όλες οι πιθανές επιλογές που μπορούν να γίνουν, μαζί με τον αντίστοιχο κώδικα για να είναι ακόμα πιο εύκολο στη χρήση. Ακόμα, για να απλοποιηθεί ακόμη περισσότερο η χρήση τους, ο κώδικας που παρέχεται μπορεί να επεξεργαστεί απευθείας στο jsfiddle που είναι μια διαδικτυακή υπηρεσία που επιτρέπει στο χρήστη να δει και να επεξεργαστεί απευθείας τον κώδικά του για την ιστοσελίδα που θέλει να παράξει.

Εικόνα 3.13 - Παράδειγμα δυνατότητας HighCharts

Επιπρόσθετα, το HighCharts παρέχει πολλές διαδραστικές επιλογές για τα γραφήματα που προσφέρει[29]. Μια εξ αυτών είναι η απευθείας εκτύπωση του γραφήματος ή η αποθήκευσή του σε μορφές PNG, JPEG, PDF και SVG. Επιπλέον, με την μετακίνηση του κέρσορα σε ένα σημείο εμφανίζεται η τιμή του συγκεκριμένου σημείου και το όνομα της σειράς στην οποία ανήκει. Επίσης παρέχονται δυνατότητες εμφάνισης ενός συγκεκριμένου

μέρος του γραφήματος, μεγέθυνση σε ένα συγκεκριμένο σημείο, ευφυΐα στη χρήση αξόνων χρόνου και πολλά άλλα.

Υπάρχει συμβατότητα στη χρήση του, στους browsers Internet Explorer, Firefox, Chrome, Safari, Opera, iOs(Safari) και Android Browser και μπορεί να τρέξει πάνω από jQuery ή Standalone Framework[30]. Για την χρήση του, δεν απαιτείται ούτε κάποια εγκατάσταση στον server ούτε κάποιο plugin τύπου Flash ή Java στην πλευρά του χρήστη.[31]

Για την εγκατάσταση του HighCharts εκτελούμε τις πιο κάτω εντολές:

1. wget http://code.highcharts.com/zips/Highcharts-4.2.1.zip
2. sudo mv ./Highcharts-4.2.1.zip ./path
3. sudo unzip ./path/Highcharts-4.3.1.zip

Στη συγκεκριμένη εργασία, το HighCharts, χρησιμοποιήθηκε για την εξαγωγή 3 γραφημάτων. Στο ένα εμφανίζεται η πρόβλεψη του καιρού για τις επόμενες 24 ώρες και στο άλλο εμφανίζονται στατιστικά διαφόρων θερμοκρασιών.

3.4.9. Bootstrap

Το Bootstrap είναι μια δωρεάν βιβλιοθήκη ανοιχτού κώδικα για την ανάπτυξη ανταποκρίσιμων ιστοσελίδων ή διαδικτυακών εφαρμογών. Είναι επίσης front end web framework, δηλαδή είναι διεπαφή για τον χρήστη και όχι κώδικας για τον server. Το Bootsrap είναι συμβατό με τις καινούριες εκδόσεις των φυλλομετρητών Internet Explorer, Firefox, Chrome, Safari και Opera. Ακόμα, υποστηρίζει ανταποκρίσιμο σχεδιασμό (responsive design), δηλαδή με βάση τα χαρακτηριστικά της συσκευής που χρησιμοποιείται, η διάταξη της ιστοσελίδας προσαρμόζεται δυναμικά με αποτέλεσμα να γίνεται δυνατή η ευαναγνωσία του περιεχομένου της ιστοσελίδας ή της εφαρμογής σε όλα τα μεγέθη οθόνης.

Το Bootstrap παρέχει στους προγραμματιστές μία μεγάλη γκάμα δυνατοτήτων. Μια από αυτές τις δυνατότητες είναι ο χωρισμός του πλάτους της οθόνης σε τμήματα διαφόρων μεγεθών. Κατά τον χωρισμό, υπάρχει επίσης επιλογή που ρυθμίζει το πλάτος της κάθε στήλης ανάλογα με τον τύπο της συσκευής. Ακόμα, χρησιμοποιεί κώδικα CSS για την μορφοποίηση όλων των βασικών στοιχείων HTML (πίνακες, φόρμες, κείμενα) ώστε αυτά να έχουν σύγχρονη εμφάνιση. Επιπρόσθετα στα βασικά στοιχεία HTML, έχουν προστεθεί κάποια στοιχεία περιβάλλοντος που διευκολύνουν τον προγραμματιστή και παράλληλα κάνουν την ιστοσελίδα ή την εφαρμογή πιο εντυπωσιακή. Κάποια από αυτά τα στοιχεία είναι η ομαδοποίηση κουμπιών, η πλοήγηση, η σελιδοποίηση, οι ετικέτες, τα εικονίδια, τα προειδοποιητικά μηνύματα και οι γραμμές προόδου. Είναι ακόμα δυνατόν να χρησιμοποιηθεί κάποιος κώδικας JavaScript σε μορφή jQuery plugin ώστε ο χρήστης να έχει μία περεταίρω διεπαφή με τα στοιχεία. Τέτοια στοιχεία είναι τα παράθυρα διαλόγου, οι επεξηγήσεις, οι ειδοποιήσεις και τα καρουσέλ.[32]

Αξιοσημείωτο είναι ότι στην ιστοσελίδα του Bootstrap, είναι διαθέσιμα όλα τα χαρακτηριστικά και οι δυνατότητες που προσφέρει καθώς υπάρχει επίσης και κώδικας σαν βοηθητικό υλικό. Επίσης, υπάρχουν πολλά παραδείγματα από διάφορες ιστοσελίδες που το

χρησιμοποιήσαν για να δοθούν σαν ιδέες κατά την ανάπτυξη της νέας ιστοσελίδας ή εφαρμογής. Το Bootstrap, φιλοξενείται, αναπτύσσεται και επιδιορθώνεται στο GitHub. [33]

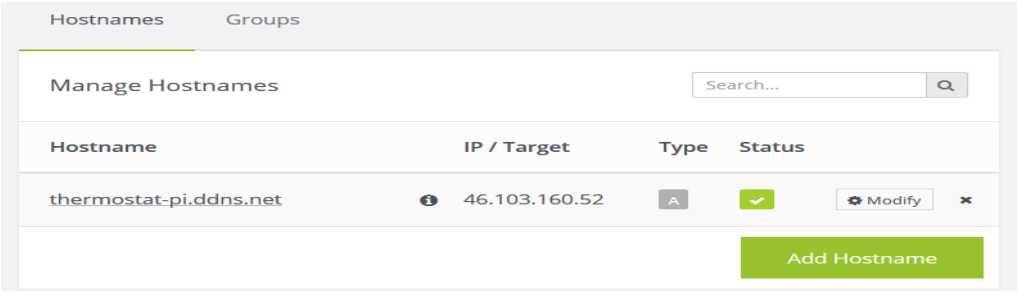
Στη συγκεκριμένη εργασία, χρησιμοποιήθηκε το Bootsrap, ούτως ώστε να αναπτυχθεί μία πιο σύγχρονη διαδικτυακή εφαρμογή και η οποία θα μπορεί να προσαρμόζεται σε όλων των ειδών οθόνες.

3.4.10. noIP

Το noIP είναι ένας πάροχος Dynamic DNS (Domain Name System) που εξασφαλίζει την απομακρυσμένη χρήση μιας συσκευής. Ο συγκεκριμένος πάροχος, μετράει 20 εκατομμύρια χρήστες και εγγυάται την συνεχόμενη λειτουργία της υπηρεσίας. Επιλέγοντας ένα hostname από την υπηρεσία, γίνεται αντιστοίχιση αυτού του hostname με την IP που έχουμε τη συγκεκριμένη στιγμή και κάθε φορά που αλλάζει η IP της συσκευής, η αντιστοίχιση ανανεώνεται. Η ανάγκη για χρήση μιας τέτοιας υπηρεσίας έγκειται στο γεγονός ότι οι IPs που χρησιμοποιούμε από τους παρόχους τηλεπικοινωνιών είναι δυναμικές και όχι στατικές.[34]

Ο συγκεκριμένος πάροχος, παρέχει δωρεάν τις υπηρεσίες του σε μέλη που θέλουν να χρησιμοποιήσουν έως 3 hostnames με την προϋπόθεση ότι θα έχουν περιορισμένες επιλογές από domains και ότι θα πρέπει να επιβεβαιώνουν τη χρήση της υπηρεσίας κάθε 30 ημέρες.[35]

Για να χρησιμοποιήσουμε την υπηρεσία, πρέπει πρώτα να δημιουργήσουμε λογαριασμό στην ιστοσελίδα http://www.noip.com/. Αφού εισέλθουμε στο λογαριασμό, επιλέγουμε create hostname και εισάγουμε το όνομα που επιθυμούμε. Στη παρούσα εργασία επιλέγηκε το thermostat-pi.ddns.net.



Εικόνα 3.14 - Δημιουργία hostname στον πάροχο noIP

Λόγω του ότι η public IP που μας δίνεται μπορεί να αλλάξει σε κάποια στιγμή, θα πρέπει να ανενώνεται συνεχώς η αντιστοίχιση του hostname με την IP. Για αυτό το λόγο πρέπει να χρησιμοποιηθεί το No-IP DUC το οποίο είναι υπεύθυνο για αυτή την αντιστοίχιση. Για τη χρήση του εκτελούμε τις πιο κάτω εντολές[36]:

1. wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
2. tar vzxf noip-duc-linux.tar.gz
3. cd noip-2.1.9-1
4. sudo make
5. sudo make install

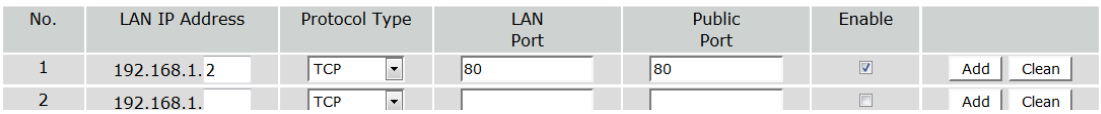
   Στο συγκεκριμένο βήμα θα ζητηθούν τα username και password του no-IP λογαριασμού που δημιουργήθηκε καθώς και ο χρόνος κατά τον οποίο θα τρέχει η υπηρεσία για να κάνει την ανανέωση. Σε αυτή την εργασία επιλέχθηκε ίσος με 30 λεπτά.
6. sudo /usr/local/bin/noip2
7. sudo noip2 -S

Για την ορθή λειτουργία της υπηρεσίας πρέπει επίσης να ρυθμιστεί κατάλληλα και ο δρομολογητής που χρησιμοποιούμε. Ανάλογα με το μοντέλο του δρομολογητή, η παρακάτω διαδικασία διαφέρει. Στη συγκεκριμένη περίπτωση, για να επεξεργαστούμε κάποιες ρυθμίσεις του δρομολογητή, κάνουμε είσοδο στην ιστοσελίδα 192.168.1.1.

Αρχικά, πρέπει να ενεργοποιήσουμε την υπηρεσία DDNS στο δρομολογητή και άρα επιλέγουμε από το μενού την καρτέλα DDNS. Συμπληρώνουμε τις πληροφορίες που απαιτούνται και στα πεδία Account και Password δίνουμε αυτά που χρησιμοποιήσαμε για την δημιουργία του λογαριασμού noIP.

Εικόνα 3.15 - Ενεργοποίηση υπηρεσίας noIP στον δρομολογητή

Στη συνέχεια πρέπει να γίνει Port Forwarding της πόρτας 80 που είναι η πόρτα hhtp. Με αυτό τον τρόπο κάθε αιτήματα που φτάνουν στον δρομολογητή στο port80, θα μεταφέρονται στο port80 του web server για προβολή του web-app. Επιλέγουμε λοιπόν από το μενού, NAT -> Virtual Server και συμπληρώνουμε τον πίνακα κατάλληλα.



Εικόνα 3.16 - Port Forwarding

Πληκτρολογώντας λοιπόν τη διεύθυνση thermostat-pi.ddns.net από οποιοδήποτε δίκτυο, μπορούμε να έχουμε πρόσβαση στο web application που δημιουργήθηκε για τον θερμοστάτη.

3.4.11. Γλώσσες Προγραμματισμού

Για τον προγραμματισμό των σελίδων της εφαρμογής χρησιμοποιήθηκαν οι γλώσσες HTML και PHP και η CSS για την μορφοποίηση. Η γλώσσα PHP δεν είναι προεγκατεστημένη στο Raspberry Pi αλλά μπορεί εύκολα να εγκατασταθεί με την εντολή sudo apt-get install php5 libapache2-mod-php5 –y.

Ακόμα, για όλα τα scripts έχει χρησιμοποιηθεί η γλώσσα Python που ήταν προεγκατεστημένη. Ωστόσο χρειάστηκε να εγκατασταθεί το πρόσθετο για το RRDTool με την εντολή sudo apt-get install python-rrdtool και για το πρόσθετο για την mySQL με την εντολή sudo apt-get install python-mysqldb.

Τέλος, έχει χρησιμοποιηθεί η γλώσσα JavaScript για τον χειρισμό των HighCharts, που επίσης ήταν προεγκατεστημένη.

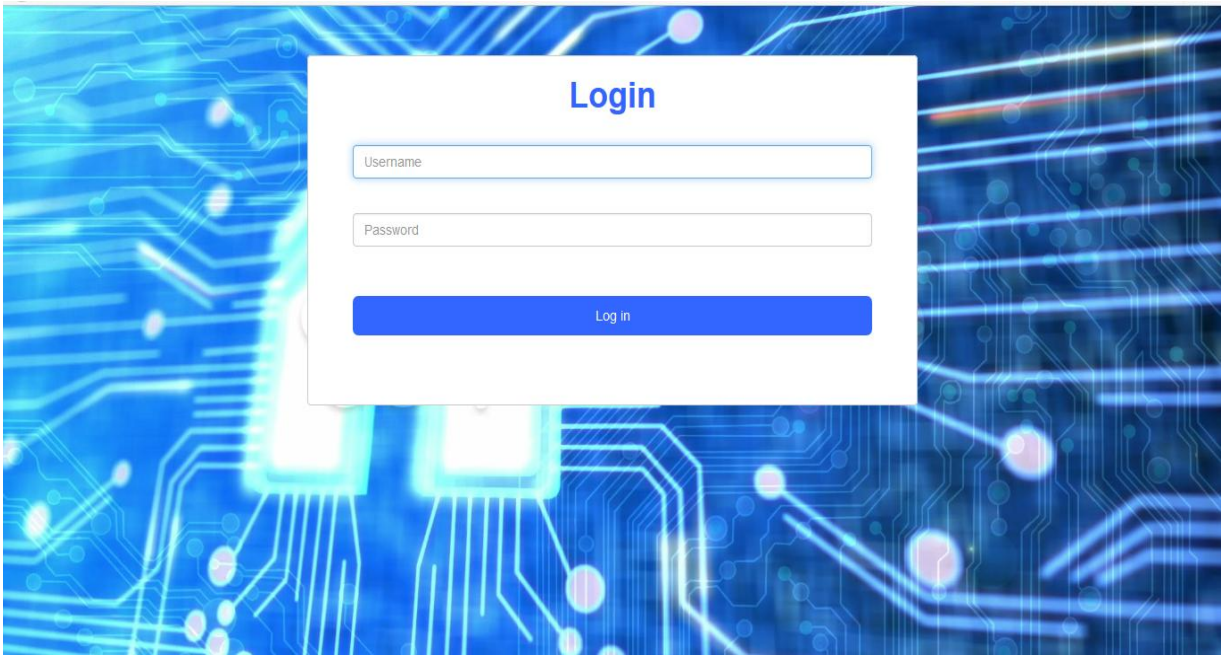
ΚΕΦΑΛΑΙΟ 4. WEB ΕΦΑΡΜΟΓΗ

Σε αυτό το κεφάλαιο, θα παρουσιαστεί η διαδικτυακή εφαρμογή που αναπτύχθηκε ώστε ο χρήστης να μπορεί να χειρίζεται και να παρακολουθεί το θερμοστάτη. Αρχικά, θα παρουσιαστεί η εφαρμογή καθώς επίσης και τα βασικά της χαρακτηριστικά. Στη συνέχεια, θα παρουσιαστεί ο τρόπος με τον οποίο ο χρήστης θα εισάγει τα δεδομένα για το εβδομαδιαίο πρόγραμμα θερμοκρασιών που επιθυμεί και τέλος θα παρουσιαστούν τα στατιστικά γραφήματα που μπορεί να παρακολουθεί ο χρήστης.

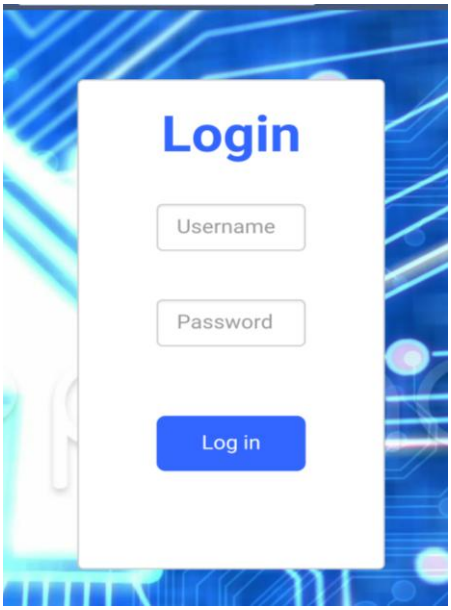
4.1. Παρουσίαση Εφαρμογής

Όπως έχει αναφερθεί στη παράγραφο 5.4.9, η εφαρμογή έχει αναπτυχθεί με τέτοιο τρόπο ώστε να είναι ανταποκρίσιμη και να μπορεί να χρησιμοποιείται από όλων των ειδών οθόνες. Για αυτό το λόγο, η εφαρμογή θα παρουσιαστεί τόσο σε κανονική οθόνη όσο και σε οθόνη κινητού τηλεφώνου.

Για την πρόσβαση στην εφαρμογή, πληκτρολογούμε τη διεύθυνση http://thermostat-pi.ddns.net όπως έχει αναφερθεί στη παράγραφο 3.4.10. Στη συνέχεια, εμφανίζεται η αρχική σελίδα όπου είναι απαραίτητη η εισαγωγή username και password για να μπορέσει ο χρήστης να λάβει πρόσβαση στην εφαρμογή. Με αυτό το τρόπο, διασφαλίζεται η ασφαλεία χειρισμού της εφαρμογής. Το username και password που δίνει πρόσβαση στην εφαρμογή είναι project και 1111 αντίστοιχα.

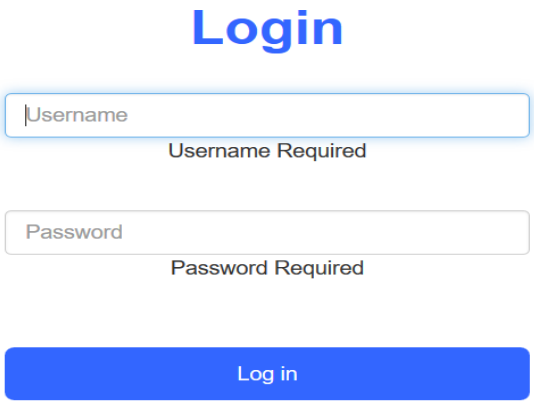


Εικόνα 4.1 - Αρχική σελίδα πρόσβασης στην εφαρμογή από υπολογιστή



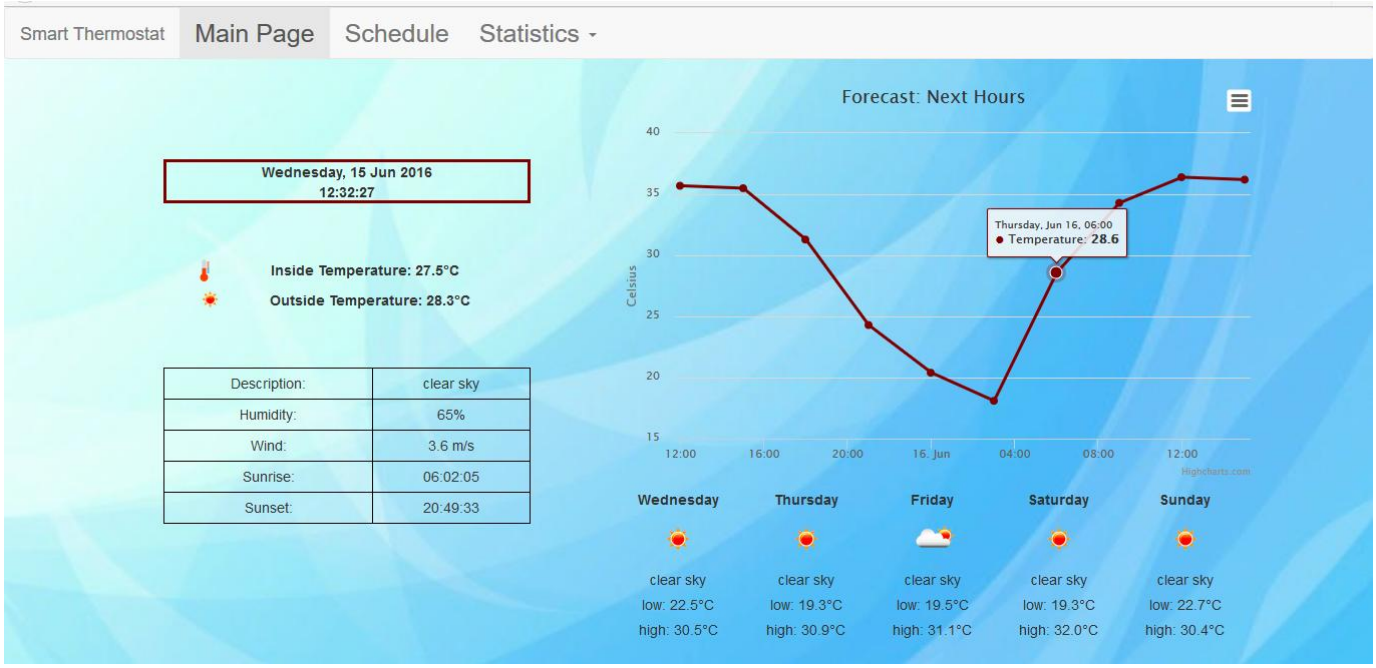
Εικόνα 4.2 - Αρχική σελίδα πρόσβασης στην εφαρμογή από κινητό τηλέφωνο

Σε περιπτώσεις όπου ο συνδυασμός δεν είναι σωστός ή που κάποιο από τα δύο στοιχεία παραλύπεται, εμφανίζονται ανάλογα μηνύματα σφάλματος.

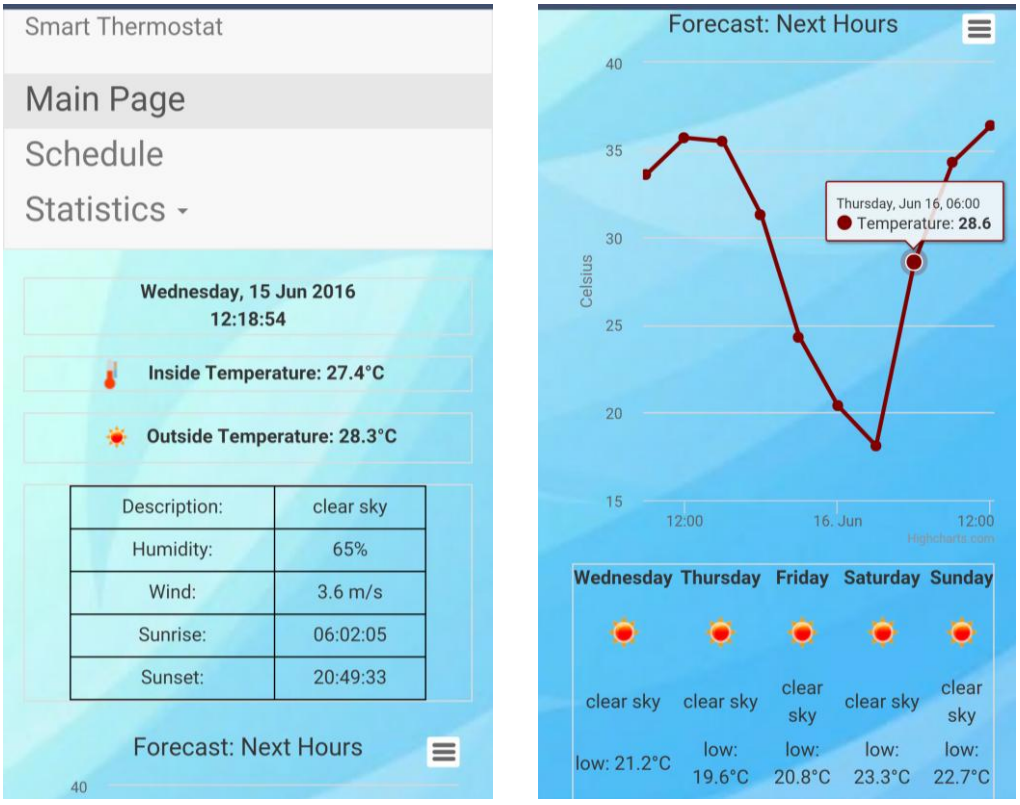


Εικόνα 4.3 - Μήνυμα Σφάλματος κατά την είσοδο στην εφαρμογή

Στη συνέχεια, αφού επιβεβαιωθεί η είσοδος του χρήστη, εμφανίζεται η κυρίως σελίδα της εφαρμογής όπου ο χρήστης μπορεί να διαβάσει διάφορες πληροφορίες. Οι πληροφορίες που εμφανίζονται, όπως φαίνεται και στην εικόνα 4.4, είναι ημερομηνία και ώρα, πληροφορίες για τις καιρικές συνθήκες που επικρατούν την συγκεκριμένη ημέρα στην συγκεκριμένη περιοχή και οι προβλέψεις για την εξωτερική θερμοκρασία τις επόμενες 24 ώρες σε μορφή γραφήματος και για τις επόμενες 5 ημέρες.



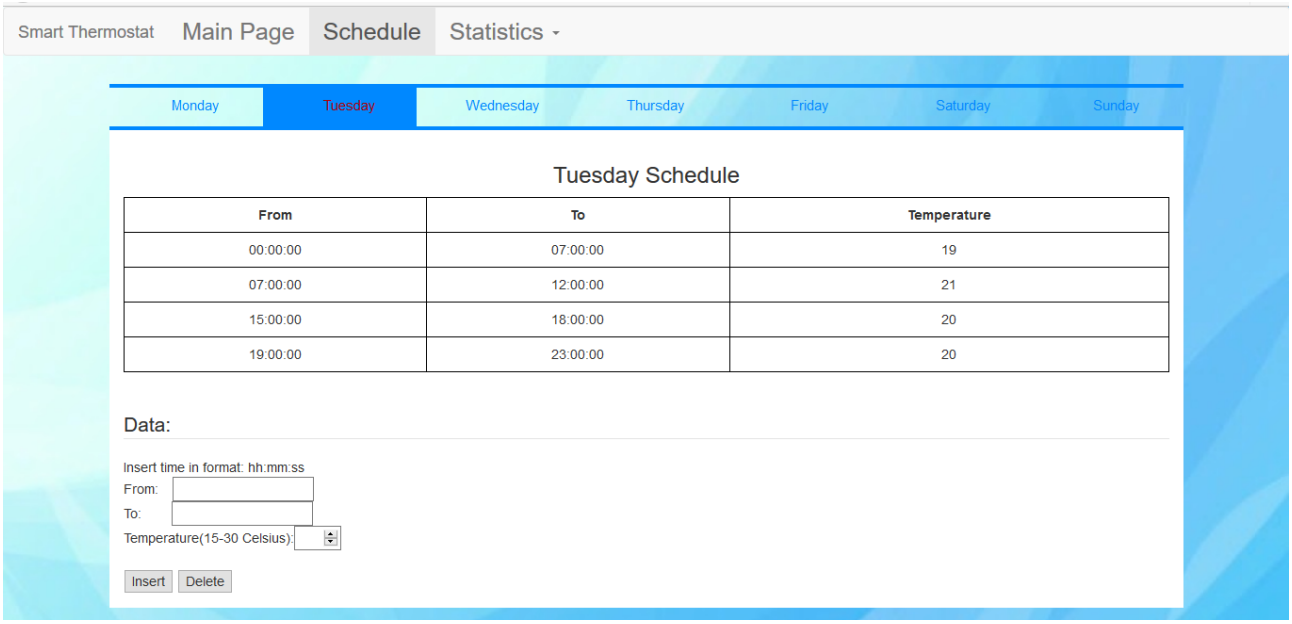
Εικόνα 4.4 - Κυρίως σελίδα στην εφαρμογή από υπολογιστή



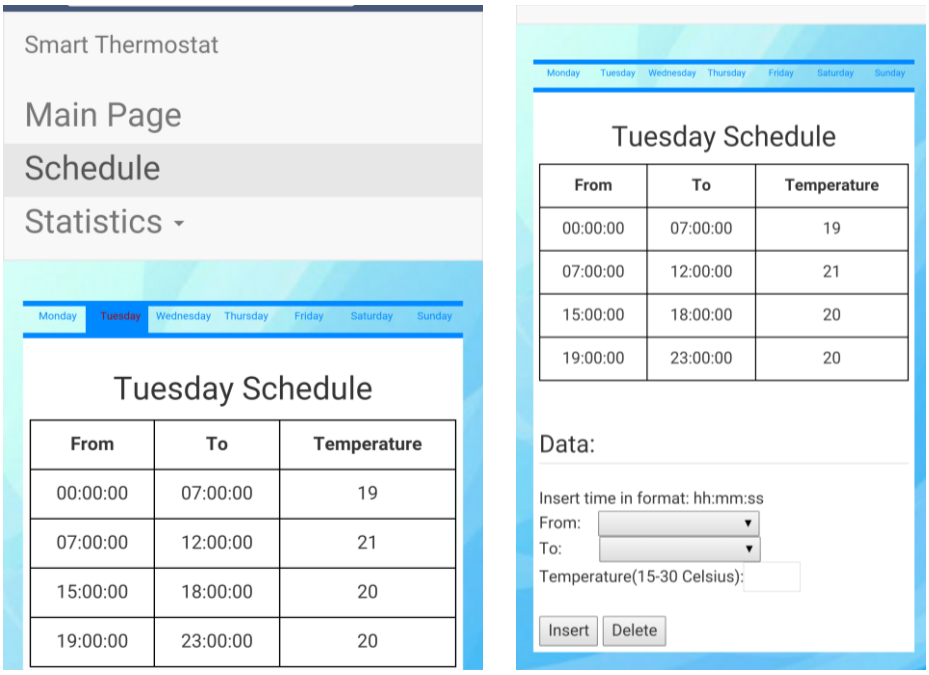
Εικόνα 4.5 - Κυρίως σελίδα στην εφαρμογή από κινητό τηλέφωνο (Η δεξιά εικόνα, εμφανίζεται στη συνέχεια της αριστερής εικόνας)

4.2. Χειρισμός Προγράμματος Θερμοκρασίας

Στην καρτέλλα “Schedule” εμφανίζεται ένα μενού όπου ο χρήστης μπορεί να εισάγει σε ποιά διαστήματα επιθυμεί κάποια συγκεκριμένη θερμοκρασία. Στο συγκεκριμένο μενού, υπάρχουν καρτέλλες για κάθε ημέρα της εβδομάδας και πατόντας πάνω στην καρτέλλα-ημέρα που επιθυμεί ο χρήστης, εμφανίζεται ένας πίνακας. Στον συγκεκριμένο πίνακα, εμφανίζονται με αύξουσα σειρά όλα τα διαστήματα με την επιθυμητή θερμοκρασία που υπάρχουν αποθηκευμένα για την συγκεκριμένη ημέρα. Σε περίπτωση που δεν υπάρχει καμία εγγραφή για την συγκεκριμένη ημέρα, εμφανίζεται το μήνυμα “No Schedule”.



Εικόνα 4.6 - Σελίδα προγράμματος στην εφαρμογή από υπολογιστή



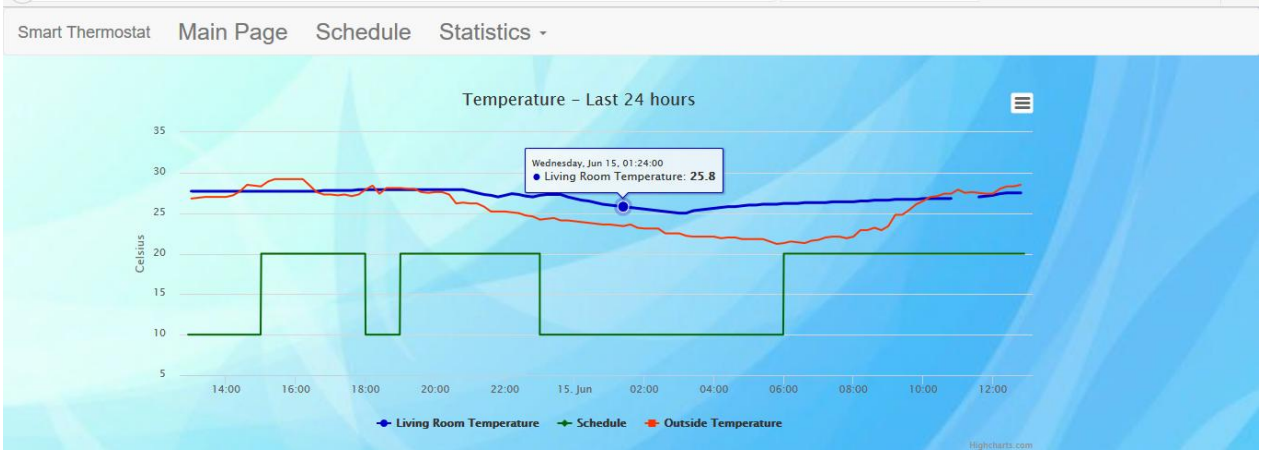
Εικόνα 4.7 - Σελίδα προγράμματος στην εφαρμογή από κινητό τηλέφωνο (Η δεξιά εικόνα, εμφανίζεται στη συνέχεια της αριστερής εικόνας)

Σε περίπτωση που επιθυμεί ο χρήστης να εισάγει μια νέα εγγραφή στο πρόγραμμα, θα πρέπει να συμπληρώσει τα απαιτούμενα στοιχεία στο χώρο κάτω από τον πίνακα και μετά να επιλέξει insert. Το νέο διάστημα που θα εισάγει ο χρήστης, θα πρέπει να είναι διαφορετικό από τα ήδη υπάρχοντα, δηλαδή να μην υπάρχει επικάλυψη. Σε περίπτωση που υπάρχει επικάλυψη, εμφανίζεται μήνυμα σφάλματος “Overlap in the time interval” και ο χρήστης πρέπει να ξαναεισάγει τα δεδομένα. Αν δεν υπάρχει κάποια επικάλυψη και τα δεδομένα δοθούν σωστά, τότε η εγγραφή εισάγεται κανονικά στον πίνακα προγράμματος.

Επίσης, δίνεται στο χρήστη η δυνατότητα να διαγράψει κάποια εγγραφή. Για τη διαδικασία της διαγραφής, ο χρήστης πρέπει να συμπληρώσει στο χώρο κάτω από τον πίνακα τα στοιχεία της εγγραφής που επιθυμεί να διαγράψει, ακριβώς όπως φαίνονται στον πίνακα και στη συνέχει να επιλέξει delete. Σε περίπτωση που εισάγει όλα τα στοιχεία σωστά, θα ανανεωθεί ο πίνακας προγράμματος και δεν θα εμφανίζεται πλέον η συγκεκριμένη εγγραφή. Αντίθετα, σε περίπτωση που τα στοιχεία που θα εισάγει δεν είναι ακριβώς όπως εμφανίζονται στον πίνακα, δεν θα ολοκληρωθεί η διαδικασία και απλά θα ξαναφορτωθεί ο υπάρχον πίνακας χωρίς καμία αλλαγή.

4.3. Παρακολούθηση Στατιστικών

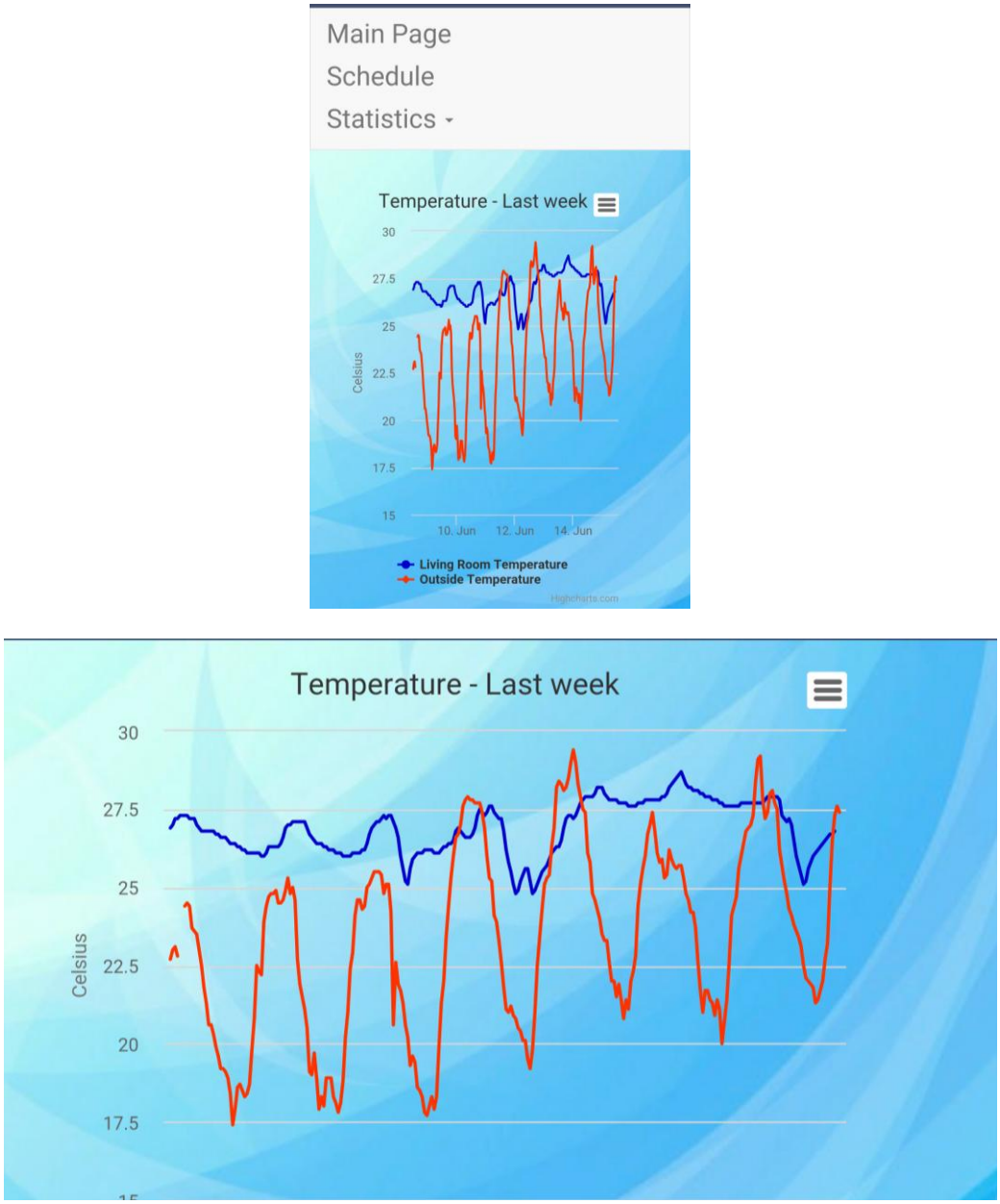
Επιλέγοντας από το κυρίως μενού την καρτέλλα “Statistics”, εμφανίζεται ένα υπομενού, όπου ανάλογα με την επιλογή “Day” ή “Week”, εμφανίζεται διαφορετικό γράφημα με στατιστικά. Επιλέγοντας το “Day”, εμφανίζεται ένα γράφημα που απεικονίζει την θερμοκρασία του δωματίου, την εξωτερική θερμοκρασία και την επιθυμητή θερμοκρασία για τις τελευταίες 24 ώρες. Η επιθυμητή θερμοκρασία είναι ουσιαστικά το πρόγραμμα που εμφανίζεται στην καρτέλλα “Schedule”, ενώ για τα διαστήματα που δεν είχε συμπληρωθεί κάποια συγκεκριμένη επιθυμητή θερμοκρασία, εμφανίζεται σαν προκαθορισμένη (default) επιθυμητή θερμοκρασία οι 10˚C. Από το συγκεκριμένο γράφημα, ο χρήστης μπορεί να επιβεβαιώσει την ορθή λειτουργία του θερμομέτρου και να κάνει συγκρίσεις για να συμπεράνει αν χρειάζεται να κάνει κάποιες αλλαγές στο εβδομαδιαίο πρόγραμμα.



Εικόνα 4.8 - Σελίδα στατιστικών ημέρας στην εφαρμογή από υπολογιστή

Σε περίπτωση που επιλεγεί το “Week”, εμφανίζεται ένα γράφημα που απεικονίζει τη θερμοκρασία δωματίου και την εξωτερική θερμοκρασία όπως καταγράφηκαν την τελευταία

μία εβδομάδα. Σκοπός του συγκεκριμένου γραφήματος είναι η παρατήρηση της επίδρασης της εξωτερικής θερμοκρασίας στην εσωτερική.



Εικόνα 4.9 - Σελίδα στατιστικών εβδομάδας στην εφαρμογή από κινητό τηλέφωνο όταν το τηλέφωνο είναι σε κάθετη θέση (πάνω εικόνα) και όταν είναι σε οριζόντια θέση (κατω εικόνα) για να υπάρχει μεγαλύτερη ανάλυση στο γράφημα

ΚΕΦΑΛΑΙΟ 5. ΠΕΙΡΑΜΑΤΑ ΘΕΡΜΟΣΤΑΤΩΝ ΣΕ ΜΟΝΤΕΛΑ

Στο συγκεκριμένο κεφάλαιο, θα παρουσιαστούν τα αποτελέσματα θερμοστατών σε θερμικά ισοδύναμα μοντέλα κτιρίων. Τα μοντέλα που θα αναπτυχθούν είναι τα 4 μοντέλα που παρουσιάστηκαν στην παράγραφο 2.7, για να επιλεγεί το πιο ακριβές και το πιο κοντινό στην πραγματικότητα. Αρχικά θα χρησιμοποιηθεί το πιο απλό μοντέλο και στη συνέχεια θα χρησιμοποιούνται όλο και πιο πολύπλοκα μοντέλα. Επίσης για κάθε μοντέλο θα προσομοιώνεται η συμπεριφορά της χρήσης ενός απλού θερμοστάτη on-off καθώς και ενός θερμοστάτη που κάνει χρήση ελεγκτή pid. Στο τέλος θα γίνει η τελική σύγκριση όλων των μοντέλων και θα επιλεγεί το πιο κατάλληλο.

5.1. Παραδοχές

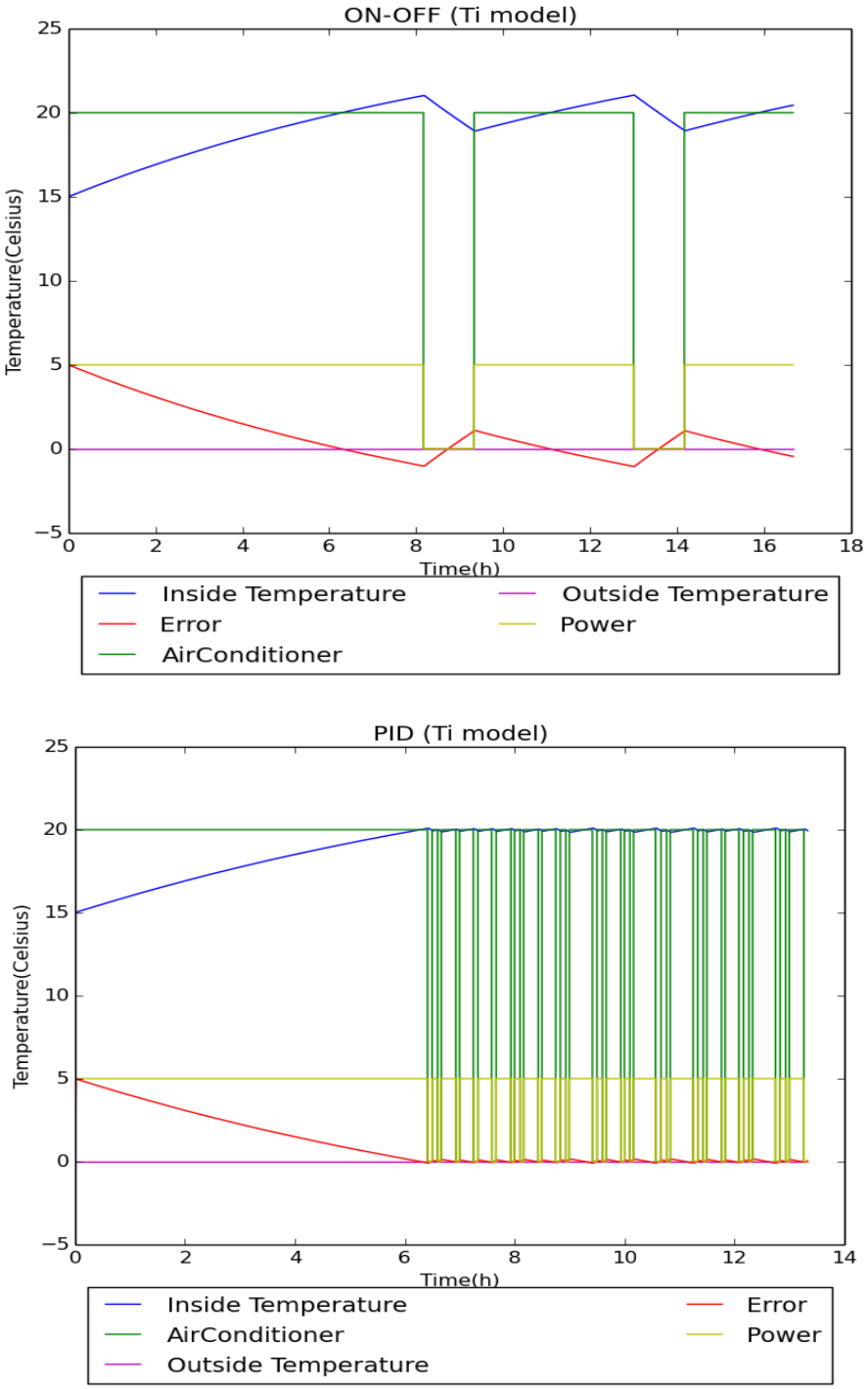
Για τις συγκεκριμένες μετρήσεις, για απλοποίηση, θεωρήθηκε ότι η εξωτερική θερμοκρασία είναι συνεχώς σταθερή και ίση με 0˚C και ότι η ακτινοβολία από τον ήλιο είναι ίση με 0 και άρα $\Phi_s = 0$. Ακόμα, θεωρήθηκε ότι η θερμοκρασία που επιθυμούμε να πετύχουμε (setpoint) είναι ίση με 20˚C και ότι το σύστημα θέρμανσης χρησιμοποιεί ισχύ ίση με 5kWatt. Επίσης, η αρχική θερμοκρασία του δωματίου (time=0) τέθηκε ίση με 15˚C και στις γραφικές παραστάσεις θα παρατηρηθεί η συμπεριφορά της μετά το πέρας συγκεκριμένου χρόνου. Τέλος, στα μοντέλα χρησιμοποιήθηκαν οι τιμές του άρθρου Identifying suitable models for the heat dynamics of buildings [2B] στον πίνακα 5.3, παράγραφος 5.3.2, και παρουσιάζονται και πιο κάτω, στην εικόνα 5.1.

Name	*Ti*	*TiTh*	*TiTeTh*	*TiTeThTs*	*TiTeThTsWithAe*
C_{i}	2.07	1.36	1.07	0.143	0.0928
C_{e}	-	-	2.92	3.24	3.32
C_{h}	-	0.309	0.00139	0.321	0.889
C_{s}	-	-	-	0.619	0.0549
R_{ia}	5.29	5.31	-	-	-
R_{ie}	-	-	0.863	0.909	0.897
R_{ea}	-	-	4.54	4.47	4.38
R_{ih}	-	0.639	93.4	0.383	0.146
R_{is}	-	-	-	0.115	1.89
A_{w}	7.89	6.22	5.64	6.03	5.75
A_{e}	-	-	-	-	3.87
τ_1	10.9	0.16	0.129	0.0102	0.0102
τ_2	-	8.9	0.668	0.105	0.105
τ_3	-	-	18.4	0.786	0.788
τ_4	-	-	-	19.6	19.3

Table 5.3: The estimated parameters. The heat capacities, C_{x}, are in [kWh/°C]. The thermal resistances, R_{x}, are in [°C/kW]. The areas, A_{x}, are in [m^2]. The time constants, τ_x, are in hours. Note that the physical interpretation most of the parameters is different for each model.

Εικόνα 5.1 - Σταθερές μοντέλου σπιτιού

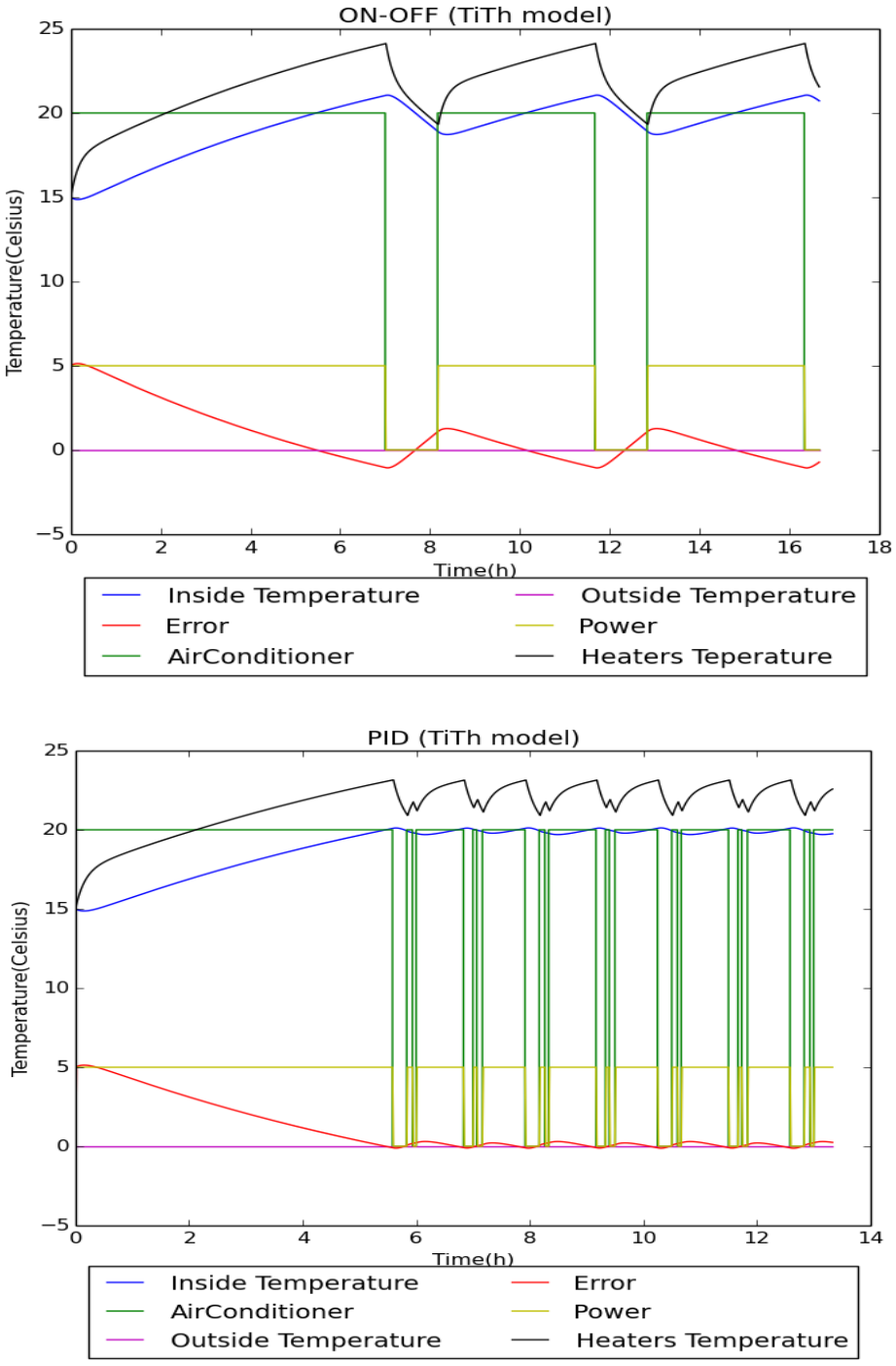
5.2. Θερμικό Μοντέλο Ti



Παράσταση 5.1 - Μοντέλο Ti (πάνω)Απλός θερμοστάτης (κάτω)Με χρήση PID ελεγκτή

Παρατηρούμε ότι και στους δύο θερμοστάτες, η θερμοκρασία δωματίου ανεβοκατεβαίνει απευθείας μετά το ανοιγοκλείσιμο του συστήματος θέρμανσης, κάτι το οποίο δεν συνάδει με την πραγματικότητα. Στη περίπτωση που χρησιμοποιείται ο pid ελεγκτής, η θερμοκρασία σταθεροποιείται καλύτερα στην επιθυμητή θερμοκρασία αφού πλέον δεν υπάρχει η υστέρηση του απλού θερμοστάτη. Ωστόσο, το μοντέλο παρουσιάζει ατέλειες.

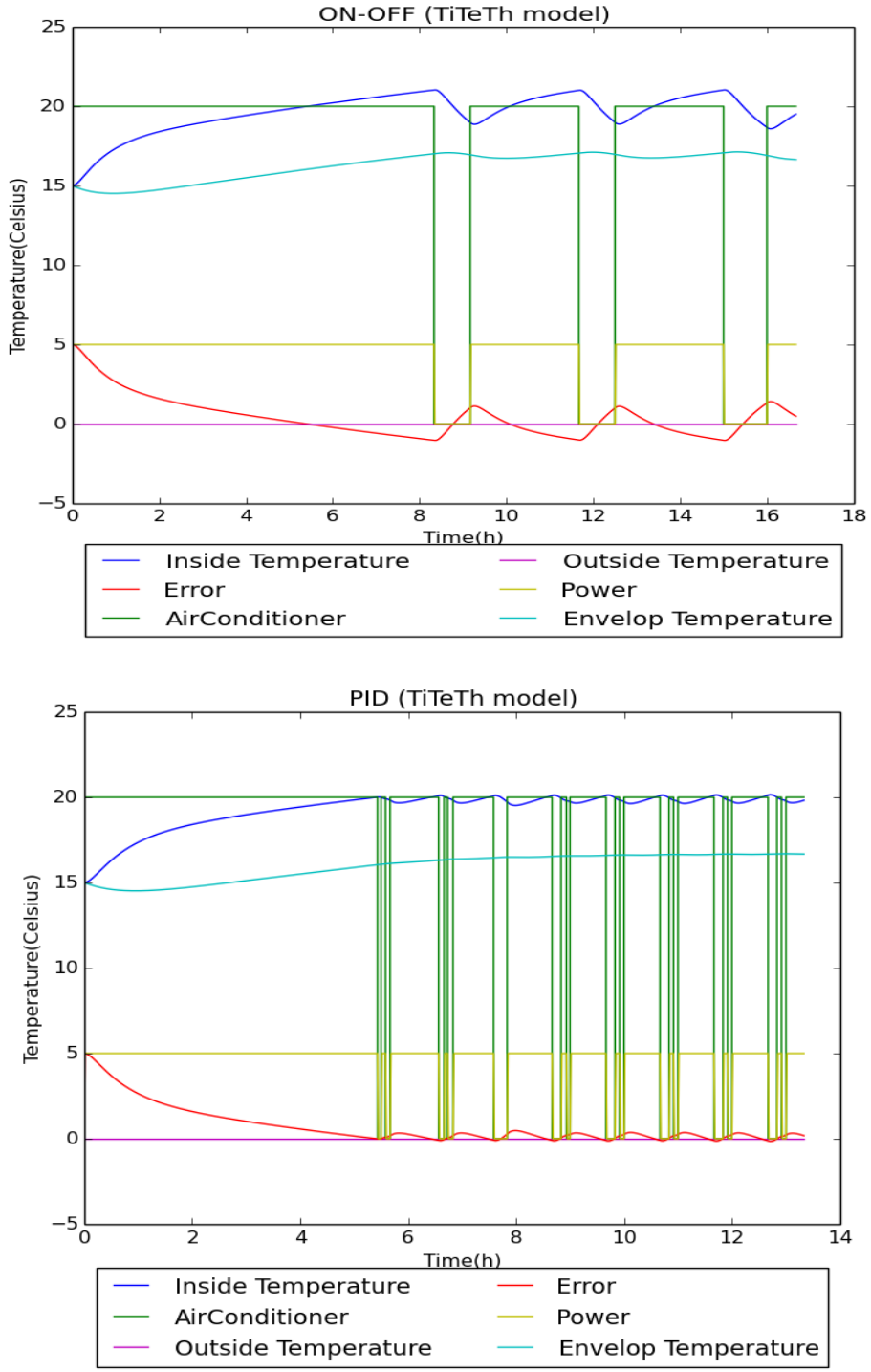
5.3. Θερμικό Μοντέλο TiTh



Παράσταση 5.2 - Μοντέλο TiTh (πάνω)Απλός θερμοστάτης (κάτω)Με χρήση PID ελεγκτή

Παρατηρούμε ότι το σύστημα θέρμανσης που έχει προστεθεί στο μοντέλο, παρουσιάζει υψηλότερη θερμοκρασία από 20˚C το όποιο είναι αναμενόμενο. Με αυτό το τρόπο, η θερμοκρασία του συστήματος θέρμανσης επηρεάζει τη θερμοκρασία δωματίου με αποτέλεσμα το δωμάτιο να ζεσταίνεται πιο γρήγορα σε σχέση με το προηγούμενο μοντέλο και η συμπεριφορά που παρουσιάζει να είναι πιο ομαλή. Επίσης, παρατηρούμε και πάλι καλύτερη σταθεροποίηση της θερμοκρασία λόγω του pid ελεγκτή.

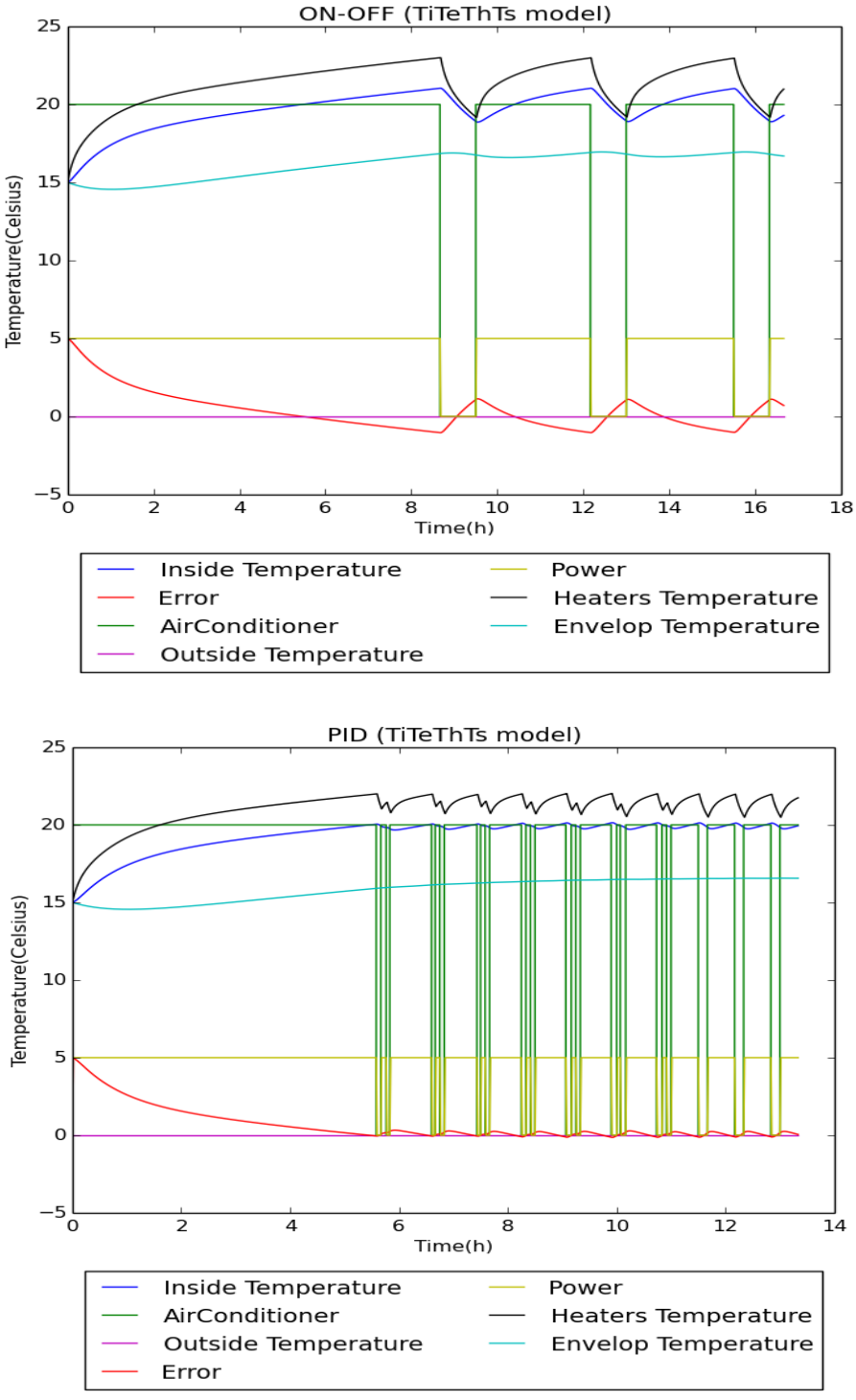
5.4. Θερμικό Μοντέλο TiTeTh



Παράσταση 5.3 - Μοντέλο TiTeTh (πάνω)Απλός θερμοστάτης (κάτω)Με χρήση PID ελεγκτή

Παρατηρούμε ότι ο περίγυρος που έχει προστεθεί στο μοντέλο, βοηθάει στη πιο γρήγορη αύξηση της θερμοκρασίας δωματίου. Με αυτό το τρόπο, πετυχαίνετε καλύτερη σταθεροποίηση της εσωτερικής θερμοκρασίας και γίνεται επίσης πιο ομαλή σε σχέση με το προηγούμενο μοντέλο. Ακόμα, παρατηρούμε και πάλι καλύτερη σταθεροποίηση της θερμοκρασία λόγω του pid ελεγκτή.

5.5. Θερμικό Μοντέλο TiTeThTs



Παράσταση 5.4 - Μοντέλο TiTeThTs (πάνω)Απλός θερμοστάτης (κάτω)Με χρήση PID ελεγκτή

Παρατηρούμε ότι η εσωτερική θερμοκρασία παρουσιάζει την πιο ομαλή συμπεριφορά σε σχέση με όλα τα προηγούμενα μοντέλα. Ακόμα, παρατηρούμε και πάλι καλύτερη σταθεροποίηση της θερμοκρασία λόγω του pid ελεγκτή.

5.6. Συμπεράσματα

Συγκρίνοντας τα πιο πάνω μοντέλα, παρατηρούμε από τη μορφή των παραστάσεων, ότι το πλησιέστερο στην πραγματικότητα είναι το τελευταίο, δηλαδή το μοντέλο TiTeThTs. Ακόμα, όπως φάνηκε στις προηγούμενες γραφικές υπολογηζόταν το σφάλμα, δηλαδή η απόκλιση της πραγματικής από την επιθυμητή θερμοκρασία, για κάθε χρονική στιγμή. Στον πίνακα 1 παρουσιάζονται τα μέγιστα και ελάχιστα σφάλματα για τον κάθε θερμοστάτη ώστε να μπορούν να συγκριθούν με περισσότερη ακρίβεια.

	on-off	**with pid**
Ti	(-1.05, 1.1)	(-0.1, 0.15)
TiTh	(-1.05, 1.3)	(-0.1, 0.3)
TiTeTh	(-1.05, 1.15)	(-0.1, 0.5)
TiTeThTs	(-1.05, 1.15)	(-0.05, 0.3)

Πίνακας 1 – Σφάλματα θερμοστατών

Από τον πίνακα 1, συμπεραίνουμε ότι το μοντέλο TiTeThTs προσφέρει πολύ μικρό εύρος σφαλμάτων. Το συγκεκριμένο μοντέλο είναι το πιο περίπλοκο αφού χρησιμοποιούνται διάφορες θερμοκρασίες (μεταβλητές κατάστασης στο μοντέλο) για τον υπολογισμό της εσωτερικής θερμοκρασίας και για αυτό το λόγο μπορούμε να λάβουμε καλύτερα και πιο ακριβή αποτελέσματα. Το συγκεκριμένο αποτέλεσμα ήταν αναμενόμενο, αφού η χρήση περισσοτέρων μεταβλητών κατάστασης σε κάποιο μοντέλο μπορεί να προσομοιώσει καλύτερα το κτίριο. Επίσης, στο συγκεκριμένο μοντέλο χρησιμοποιούνται οι περισσότερες θερμικές αντιστάσεις για να περιγράψουν τις απώλειες θερμότητας καθώς και οι περισσότεροι θερμικοί πυκνωτές για να περιγράψουν καλύτερα τη θερμική χωρητικότητα των διαφόρων αντικειμένων στο κτίριο.

Όσο αφορά τον τύπο του θερμοστάτη, φαίνεται καθαρά τόσο από τον πίνακα 1 όσο και από τις γραφικές παραστάσεις ότι η χρήση pid, προσφέρει καλύτερα αποτελέσματα σε σχέση με ένα απλό on-off θερμοστάτη. Τα αποτελέσματα που λαμβάνουμε με τη χρήση του pid είναι πολύ πιο ομαλοποιημένα αφού δεν υπάρχει υστέρηση στο θερμοστάτη, καθώς επίσης παρατηρείται ότι και η εσωτερική θερμοκρασία φτάνει στην επιθυμητή θερμοκρασία σε λιγότερο χρόνο. Η συγκεκριμένη συμπεριφορά ήταν αναμενόμενη σύμφωνα με τη θεωρία ελέγχου και τη θεωρία του ελεγκτή pid που αναλύθηκαν στις παραγράφους 2.4. και 2.5. αντίστοιχα.

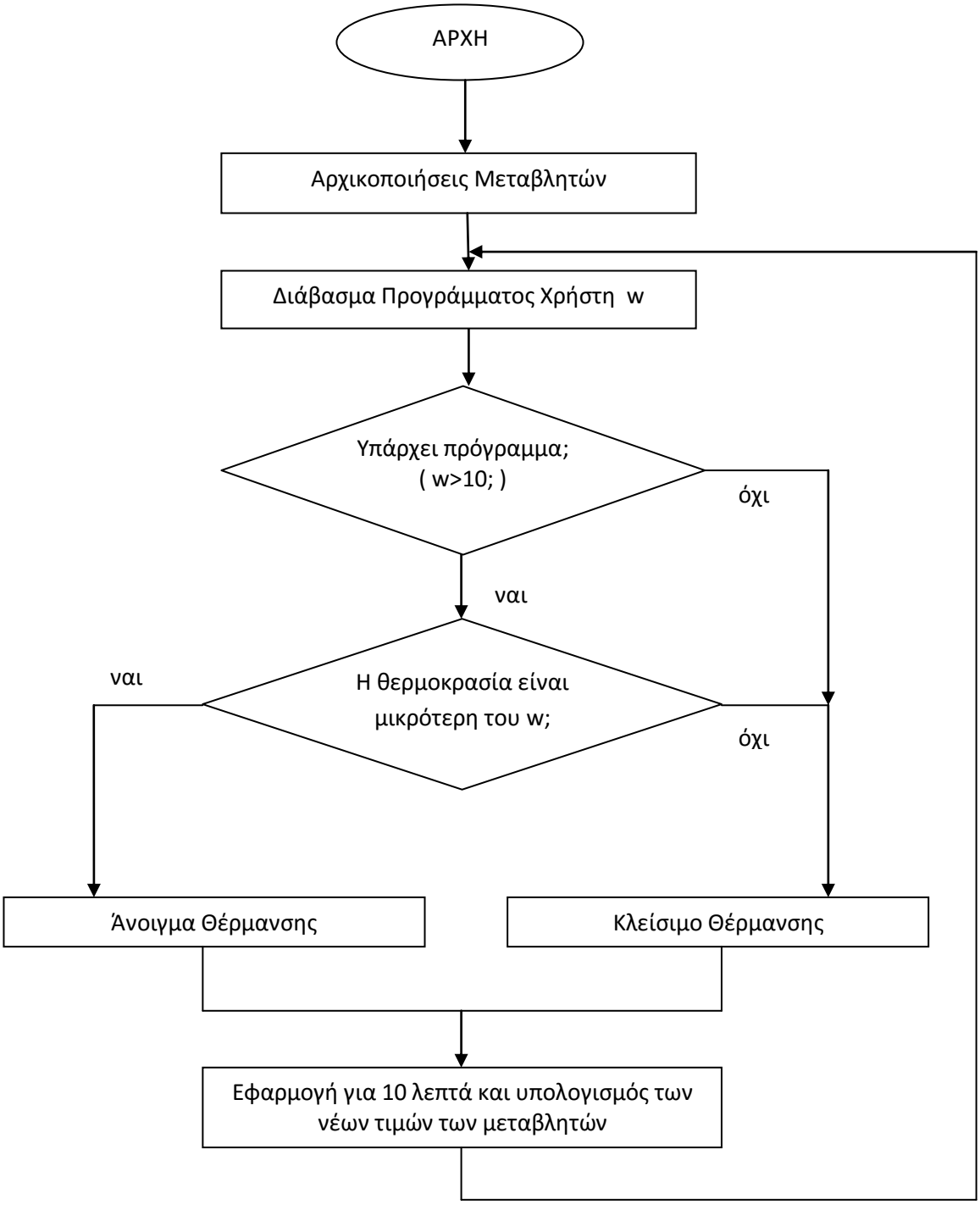
ΚΕΦΑΛΑΙΟ 6. ΕΞΥΠΝΟΣ ΘΕΡΜΟΣΤΑΤΗΣ

Στο συγκεκριμένο κεφάλαιο, θα παρουσιαστεί η διαφορά ανάμεσα σε ένα απλό και ένα έξυπνο θερμοστάτη σε πραγματικά δεδομένα. Αρχικά, θα παρουσιαστούν τα αποτελέσματα απλών θερμοστατών (τόσο για απλό on-off όσο και με χρήση pid) σε δεδομένα μίας εβδομάδας και θα επεξηγηθεί η ανάγκη για κατασκευή έξυπνων θερμοστατών. Στη συνέχεια, θα επεξηγηθεί ο αλγόριθμος που αναπτύχθηκε για τον έξυπνο θερμοστάτη καθώς επίσης θα δοθούν τα αποτελέσματα που προκύπτουν με εφαρμογή του έξυπνου αλγορίθμου στους απλούς θερμοστάτες για μια περίοδο τριών εβδομάδων. Τέλος, θα σχολιαστούν οι διαφορές ανάμεσα στους δύο ειδών θερμοστατών όπως προέκυψαν από τις προσομοιώσεις.

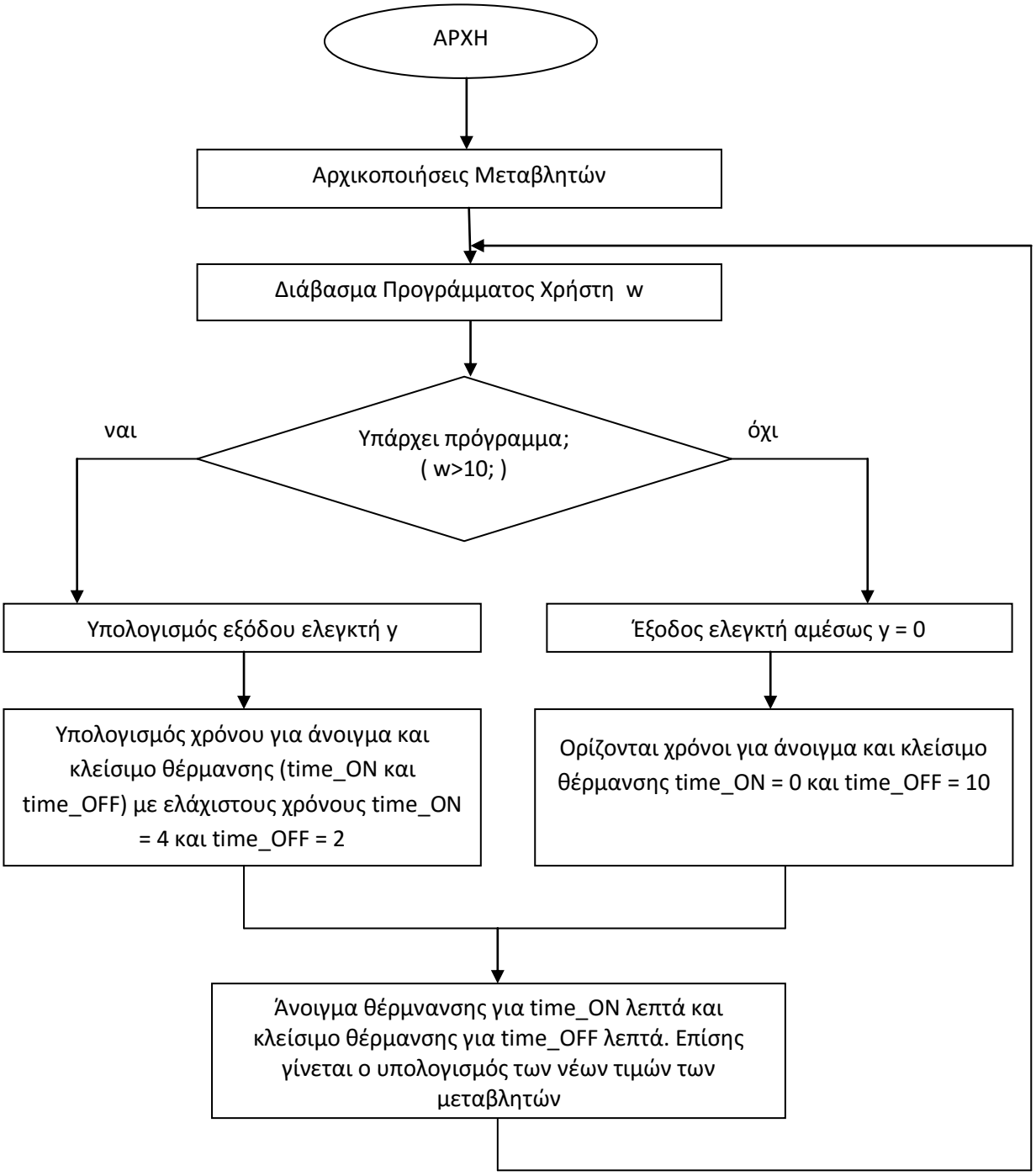
Όλα τα δεδομένα για την εξωτερική θερμοκρασία πάρθηκαν από ιστορικά αρχεία μετρήσεων του Χειμώνα στην Αθήνα ενώ η ισχύς της θέρμανσης τέθηκε στα 7kWatt. Ακόμα, για τους θερμοστάτες χρησιμοποιήθηκε το μοντέλο TiTeThTs για τους λόγους που επεξηγήθηκαν στο κεφάλαιο 5.

6.1. Παρουσίαση απλού θερμοστάτη

Ένας απλός θερμοστάτης έχει σαν στόχο τη σταθεροποίηση της θερμοκρασίας στην επιθυμητή τιμή, ανάλογα με το πρόγραμμα χρήστη. Για να ολοκληρώσει την συγκεκριμένη λειτουργία, ο θερμοστάτης απλά διαβάζει συνεχώς το χρόνο και βλέπει αν υπάρχει κάποια επιθυμητή θερμοκρασία για την συγκεκριμένη χρονική στιγμή. Όταν αντιληφθεί ότι τη συγκεκριμένη χρονική στιγμή υπάρχει πρόγραμμα, δίνει σήμα για να ξεκινήσει η θέρμανση του δωματίου και τότε φροντίζει να τη διατηρεί στη συγκεκριμένη τιμή. Όπως φάνηκε στο κεφάλαιο 5, ένας θερμοστάτης on-off σταθεροποιεί την επιθυμητή τιμή με σφάλμα περίπου ±1 ενώ με τη χρήση pid το αποτέλεσμα είναι σχεδόν συνεχώς σταθερό στην επιθυμητή τιμή καθώς το σφάλμα είναι σχεδόν αμελητέο. Πιο κάτω παρουσιάζονται τα διαγράμματα ροής δεδομένων που χρησιμοποιήθηκαν για τους δύο θερμοστάτες.

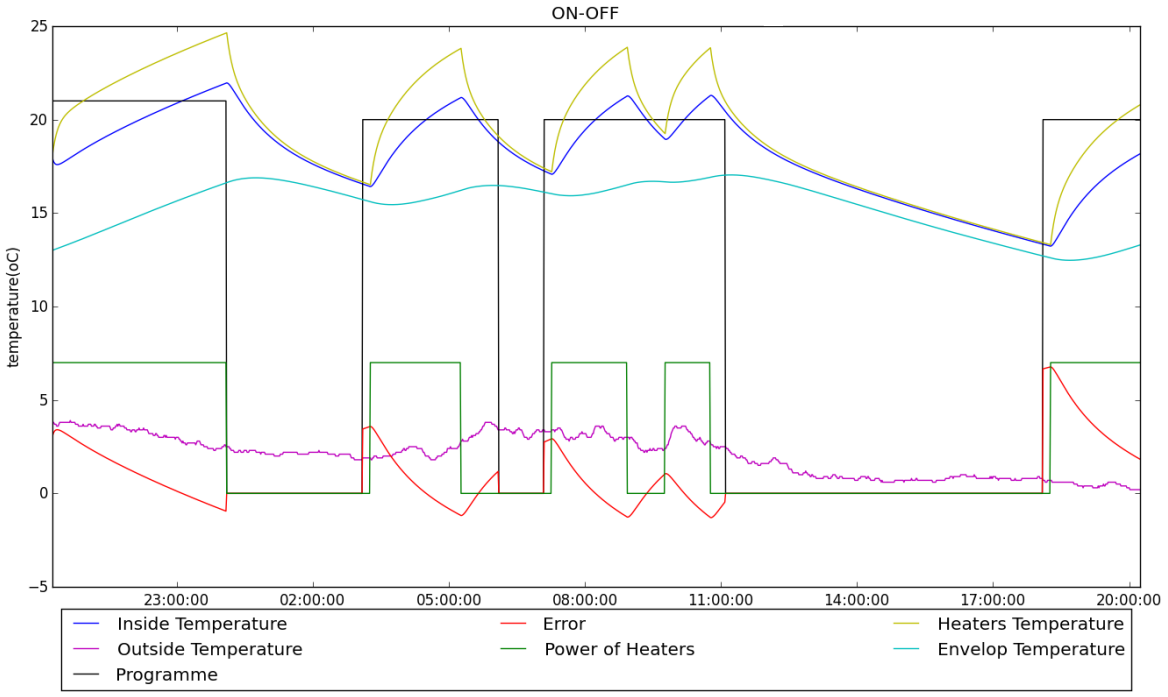


Διάγραμμα 6.1 - Διάγραμμα ροής δεδομένων για απλό θερμοστάτη on-off

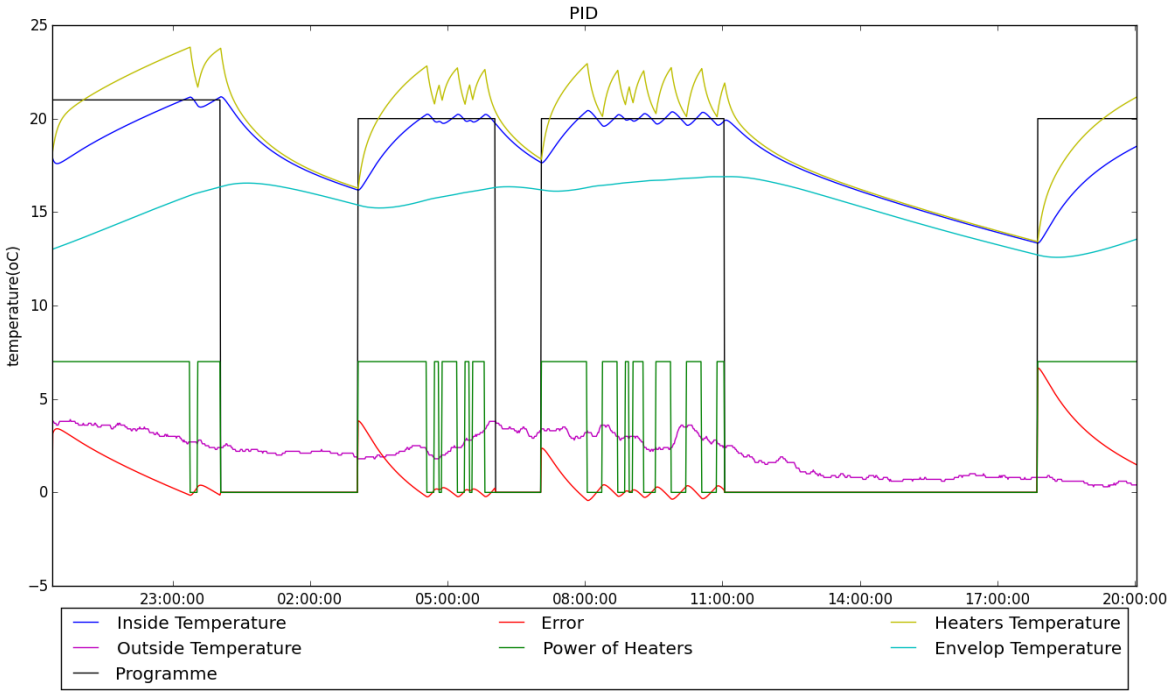


Διάγραμμα 6.2 - Διάγραμμα ροής δεδομένων για απλό θερμοστάτη με pid ελεγκτή

Με βάση τα διαγράμματα ροής δεδομένων εκτελούμε την προσομοίωση των δύο θερμοστατών σε πραγματικά δεδομένα για κάποιο μικρό χρονικό διάστημα (24 ώρες) ώστε να μπορούμε να επιβεβαιώσουμε την ορθή λειτουργία των θερμοστατών.



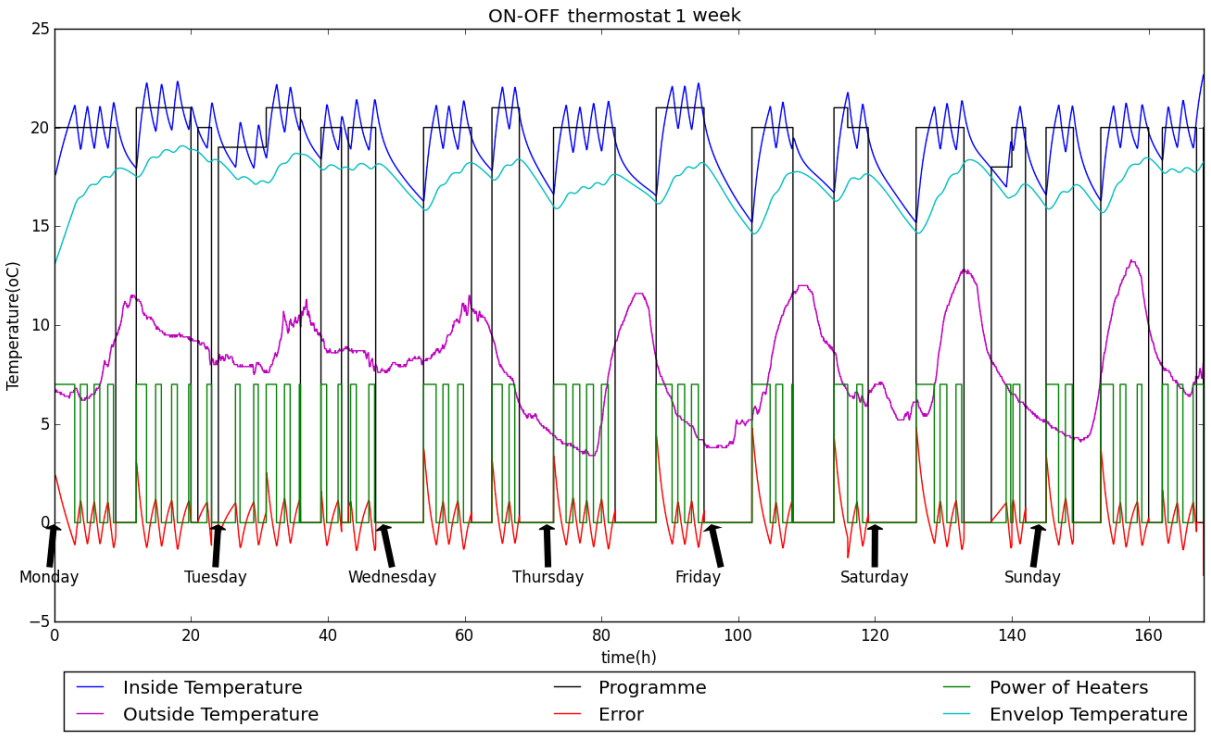
Παράσταση 6.1 - Θερμοστάτης on-off για το πρόγραμμα από Τρίτη από 20:00:00 εώς Τετάρτη 20:00:00



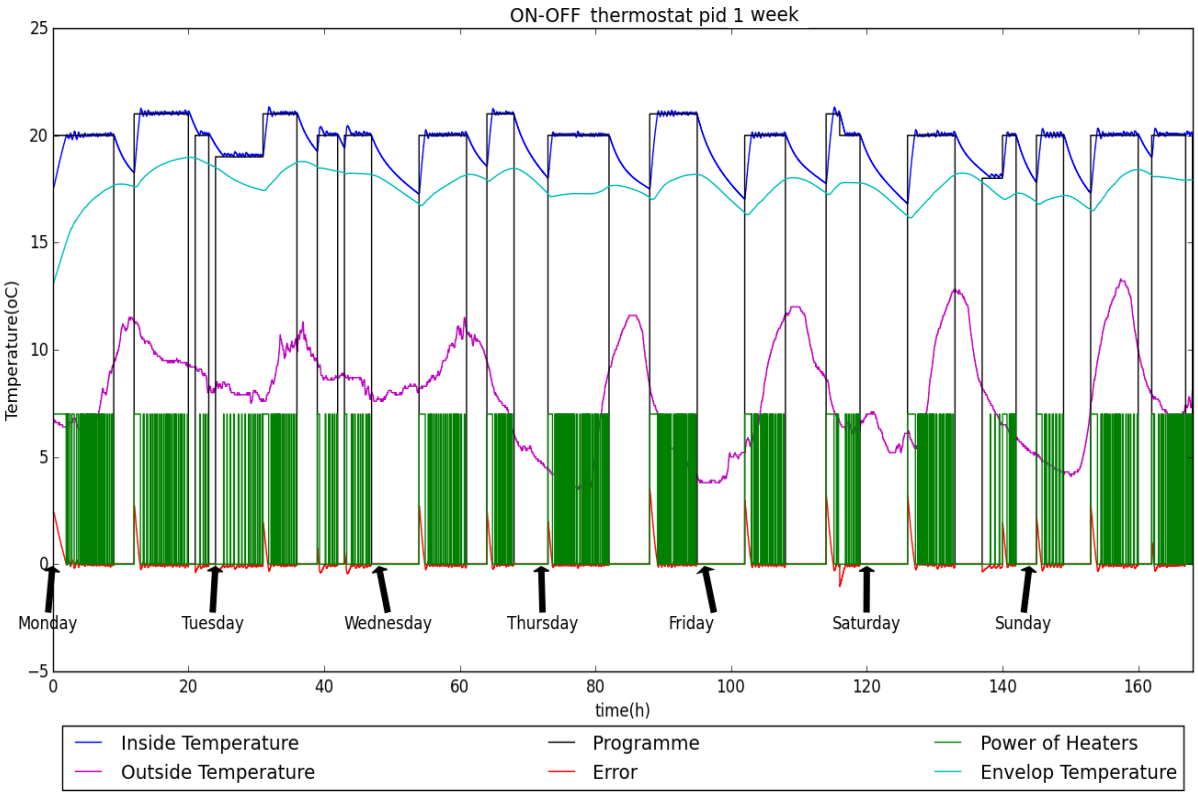
Παράσταση 6.2 - Θερμοστάτης με pid για το πρόγραμμα από Τρίτη από 20:00:00 εώς Τετάρτη 20:00:00

Παρατηρούμε για το θερμοστάτη on-off ότι κυμαίνονται οι τιμές της εσωτερικής θερμοκρασίας γύρω από την επιθυμητή τιμή με απόκλιση ±1 που είναι το σφάλμα, όπως και αναμέναμε. Επίσης, για το θερμοστάτη με pid παρατηρούμε ότι η θερμοκρασία είναι σχεδόν σταθεροποιημένη στην επιθυμητή. Ακόμα, φαίνεται στα αποτελέσματα ο χρόνος που απαιτείται ώστε να φτάσει η θερμοκρασία του δωματίου στην επιθυμητή. Η θέρμανση για την περίπτωση του θερμοστάτη on-off καθυστερεί 10 λεπτά να λειτουργήσει αφού τότε αντιλαμβάνεται ο θερμοστάτης ότι υπάρχει πρόγραμμα σε λειτουργία ενώ για την περίπτωση του θερμοστάτη με pid, παρατηρείται ότι η θέρμανση ξεκινά να δουλεύει μόλις ξεκινά να υπάρχει το πρόγραμμα.

Αφού επιβεβαιώθηκε λοιπόν η ορθή λειτουργία των θερμοστατών, προσομοιώνενται πιο κάτω η λειτουργία του κάθε θερμοστάτη στο πρόγραμμα που είναι για μία εβδομάδα. Ουσιαστικά, τα συγκεκριμένα αποτελέσματα θα είναι τα ίδια για κάθε εβδομάδα αφού οι θερμοστάτες αρχίζουν τη λειτουργία τους περίπου την ίδια ώρα που ξεκινάει το διάστημα στο πρόγραμμα. Οι μόνες διαφορές ανάμεσα σε διαφορετικές εβδομάδες που μπορεί να παρατηρηθούν είναι ο χρόνος που χρειάζεται για να πετύχει ο θερμοστάτης την επιθυμητή θερμοκρασία αφού η συγκεκριμένη λειτουργία εξαρτάται από την τιμή που έχει η εξωτερική θερμοκρασία κάθε φορά.



Παράσταση 6.3 - Θερμοστάτης on-off για το πρόγραμμα μιας εβδομάδας

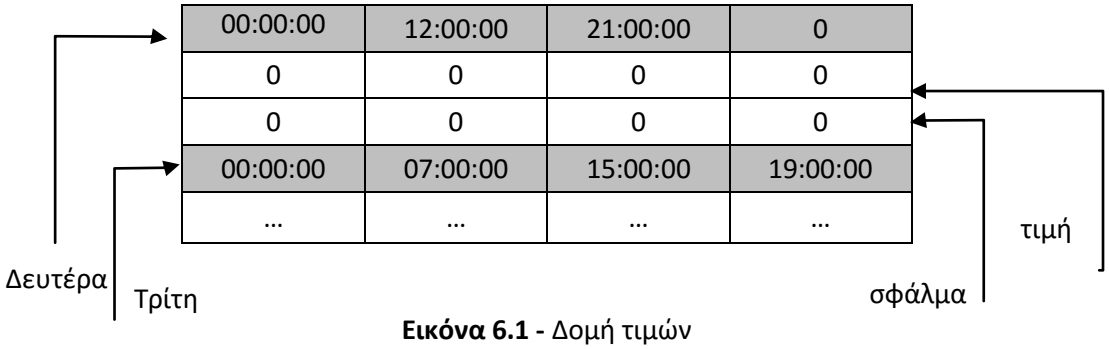


Παράσταση 6.4 - Θερμοστάτης με pid για το πρόγραμμα μιας εβδομάδας

Από τα αποτελέσματα των συγκεκριμένων προσομοιώσεων μπορούμε να καταλάβουμε εύκολα ότι απαιτείται αξιόλογος χρόνος για να πετύχουν οι θερμοστάτες την επιθυμητή θερμοκρασία. Επίσης, ο συγκεκριμένος χρόνος σε περιπτώσεις που η εξωτερική θερμοκρασία είναι πολύ χαμηλή, θα αυξάνεται ακόμη περισσότερο. Για να μειωθεί αυτός ο χρόνος και να πετυχαίνεται ακριβώς η επιθυμητή θερμοκρασία την ακριβώς συγκεκριμένη στιγμή που αναφέρεται στο πρόγραμμα, απαιτείται να γίνει χρήση ενός έξυπνου θερμοστάτη. Ο σκοπός του έξυπνου θερμοστάτη είναι να υπολογίζει κάθε φορά πόσος χρόνος χρειάστηκε για να φτάσει στη επιθυμητή τιμή η θερμοκρασία και να ξεκινά την λειτουργία της θέρμανσης τόσο χρόνο πιο πριν.

6.2. Ανάπτυξη έξυπνου αλγορίθμου

Στόχος του έξυπνου αλγορίθμου είναι να ξεκινάει τη λειτουργία τη θέρμανσης κάποιο χρόνο πριν από την αρχή του προγράμματος. Για να μπορεί να εξασφαλιστεί μια τέτοια λειτουργία, θα πρέπει να υπάρχει μια δομή όπου θα αποθηκεύονται κάποιες τιμές κλειδιά, με τη βοήθεια των οποίων ο έξυπνος αλγόριθμος θα υπολογίζει πότε ακριβώς πρέπει να ανοίγει τη θέρμανση. Η δομή που επιλέγηκε είναι πίνακας και έχει τη μορφή που φαίνεται στην εικόνα 6.1.



Εικόνα 6.1 - Δομή τιμών

Η συγκεκριμένη δομή έχει συνολικά 21 γραμμές, 3 γραμμές για κάθε μέρα. Στην 1η γραμμή της κάθε ημέρα εμφανίζονται οι ώρες όπου ξεκινά το κάθε πρόγραμμα για τη συγκεκριμένη ημέρα. Στην αμέσως επόμενη γραμμή εμφανίζεται η τιμή σε λεπτά που λέει πόσο πιό πριν πρέπει να ξεκινήσει τη λειτουργία της θέρμανσης. Αν, για παράδειγμα, υπάρχει η τιμή 70 κάτω από την ώρα προγράμματος 12:00:00, τότε υπονοείται ότι η θέρμανση πρέπει να ξεκινήσει η ώρα 10:50:00. Ωστόσο, ο αλγόριθμος έχει φτιαχτεί με τέτοιο τρόπο ώστε όταν υπάρχουν τιμές μεγαλύτερες των 2 ωρών να αμελούνται και η θέρμανση να ξεκινά 2 ώρες πιο πριν, για να μην σπαταλάται ενέργεια. Στην τρίτη γραμμή της κάθε ημέρας εμφανίζονται τιμές σφαλμάτων. Αν, δηλαδή, η θέρμανση ξεκινήσει με βάση τη τιμή της 2η γραμμής και πετύχει πιο γρήγορα ή πιο αργά την επιθυμητή θερμοκρασία θα προσθέτει κάποιο σφάλμα αρνητικό ή θετικό αντίστοιχα. Τέτοια σφάλματα προκύπτουν όταν η εξωτερική θερμοκρασία μεταβάλλεται σε σχέση με την προηγούμενη φορά, δηλαδή σε σχέση με την προηγούμενη εβδομάδα την ίδια ώρα.

Όσον αφορά τον αριθμό των στηλών της δομής, εξαρτατάται από το πρόγραμμα χρήστη, αφού η ημέρα με τα πιο πολλά διαστήματα καθορίζει το πόσες θα είναι οι στήλες. Οι ημέρες που έχουν λιγότερα διαστήματα, παραγεμίζουν τις κενές θέσεις που έχουν με μηδενικά όπως φαίνεται και στην εικόνα 6.1. για τη Δευτέρα. Ακόμα, θεωρήθηκε ότι το πρόγραμμα χρήστη δεν είναι κατ'ανάγκη σταθερό και ότι μπορεί ανα πάσα στιγμή ο χρήστης να το μεταβάλλει και για αυτό, η συγκεκριμένη δομή είναι ανταποκρίσιμη και μεταβάλλεται ανάλογα με τις ενέργειες του χρήστη. Τελικά, η δομή λαμβάνει μορφή πίνακα με 21 γραμμές και μεταβλητό αριθμό στηλών.

Οι μεταβολές στη δομή γίνονται σε τέσσερα στάδια:

- Στην αρχή κάθε ημέρας ελέγχεται εαν οι διαστάσεις του πίνακα και οι τιμές που ξεκινά το πρόγραμμα για κάθε ημέρα, ταιριάζουν με αυτές που ήταν την προηγούμενη ημέρα. Ουσιαστικά, κάθε μέρα ανανεώνει τη δομή με όλες τις αλλαγές που μπορεί να έκανε ο χρήστης το πρόγραμμα, ανεξάρτητα σε ποιά μέρα τις έκανε. Διάστημα προγράμματος το οποίο πρώτη φορά εισάγεται στη δομή, λαμβάνει στα αντίστοιχα πεδία στις επόμενες 2 γραμμές τις τιμές 0 για αρχικοποίηση.
- Όταν η θέρμανση λειτουργεί επειδή υπάρχει πρόγραμμα (και όχι πιο νωρίς λόγω της πρόβλεψης) ξεκινά ένας μετρητής χρόνου που υπολογίζει πόσος χρόνος πέρασε από τη στιγμή που ξεκίνησε η θέρμανση μέχρι τη στιγμή που η εσωτερική θερμοκρασία έφτασε την επιθυμητή. Ιδανικά, αυτή η τιμή θα έπρεπε να είναι 0 αφού όταν ξεκινά το πρόγραμμα θέλουμε να έχουμε την επιθυμητή θερμοκρασία,

και έτσι αυτός ο χρόνος που υπολογίστηκε θεωρείται ότι είναι στο σφάλμα. Άρα σε αυτό το βήμα, η νέα τιμή της γραμμής 2 γίνεται ίση με αυτή που είχε πριν μαζί με το σφάλμα που είχε πριν και το νέο σφάλμα γίνεται ίσο με το χρόνο που μετρήθηκε. Αν για παράδειγμα ο χρόνος που μετρήθηκε είναι 10 λεπτά και αφορά το διάστημα προγράμματος που ξεκινά η ώρα 12:00:00 τη Δευτέρα, η δομή αλλάζει με τον τρόπο που φαίνεται στην εικόνα 6.2.

00:00:00	12:00:00	21:00:00	0
50	40	20	0
-20	20	10	0

00:00:00	12:00:00	21:00:00	0
50	60	20	0
-20	10	10	0

Εικόνα 6.2 - Αλλαγή Δομής. Πάνω το πρίν και κάτω το μετά

- Πριν ξεκινήσει να λειτουργεί η θέρμανση λόγω της πρόβλεψης (δηλαδή πριν λειτουργήσει λόγω προγράμματος) προστίθενται οι δύο τιμές που αφορούν το διάστημα για να είναι έτοιμη η συνολική τιμή του χρόνου και το σφάλμα τίθεται ίσο με μηδέν. Πάλι αν αναφερθούμε στο παράδειγμα όπου το διάστημα προγράμματος ξεκινά η ώρα 12:00:00 τη Δευτέρα, η δομή αλλάζει με τον τρόπο που φαίνεται στην εικόνα 6.3.

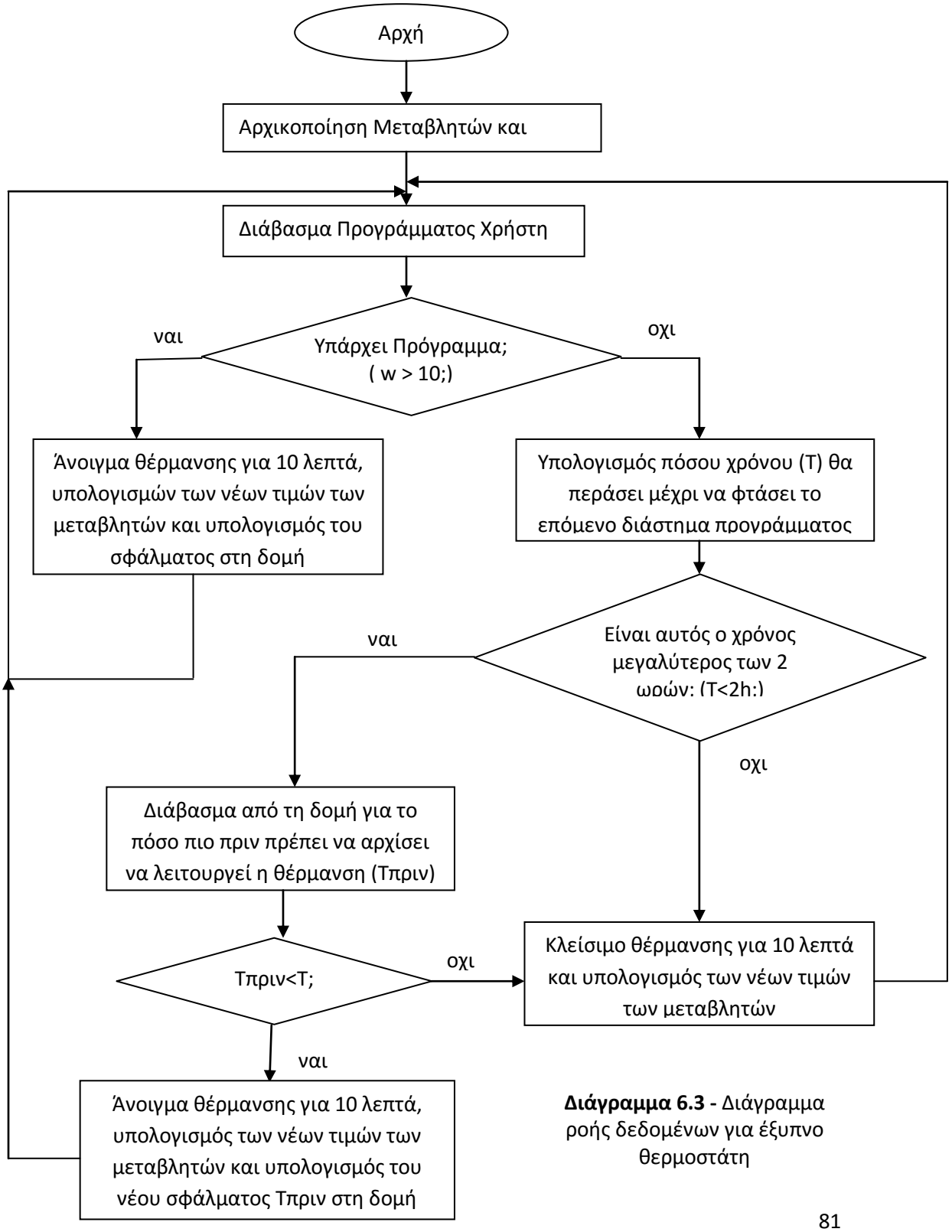
00:00:00	12:00:00	21:00:00	0
50	60	20	0
-20	10	10	0

00:00:00	12:00:00	21:00:00	0
50	70	20	0
-20	0	10	0

Εικόνα 6.3 - Αλλαγή Δομής. Πάνω το πρίν και κάτω το μετά

- Κατά τη διάρκεια που λειτουργεί η θέρμανση λόγω της πρόβλεψης (δηλαδή πριν λειτουργήσει λόγω προγράμματος), γίνεται μια νέα εκτήμηση του υπάρχοντος σφάλματος. Η συγκεκριμένη διόρθωση αρχικά έχει την τιμή 0 και τελικά λαμβάνει τη τιμή που δίνει το χρόνο σε λεπτά από τη στιγμή που επιτεύχθηκε η επιθυμητή θερμοκρασία, μέχρι τη στιγμή που θα πρέπει να ξεκινήσει το πρόγραμμα. Αυτή η τιμή αφαιρείται στο τέλος από το σφάλμα αφού η θέρμανση ξεκίνησε να λειτουργεί αρκετά πιο πριν από ότι χρειάστηκε και πέτυχε πιο νωρίς από όταν θέλαμε την επιθυμητή θερμοκρασία

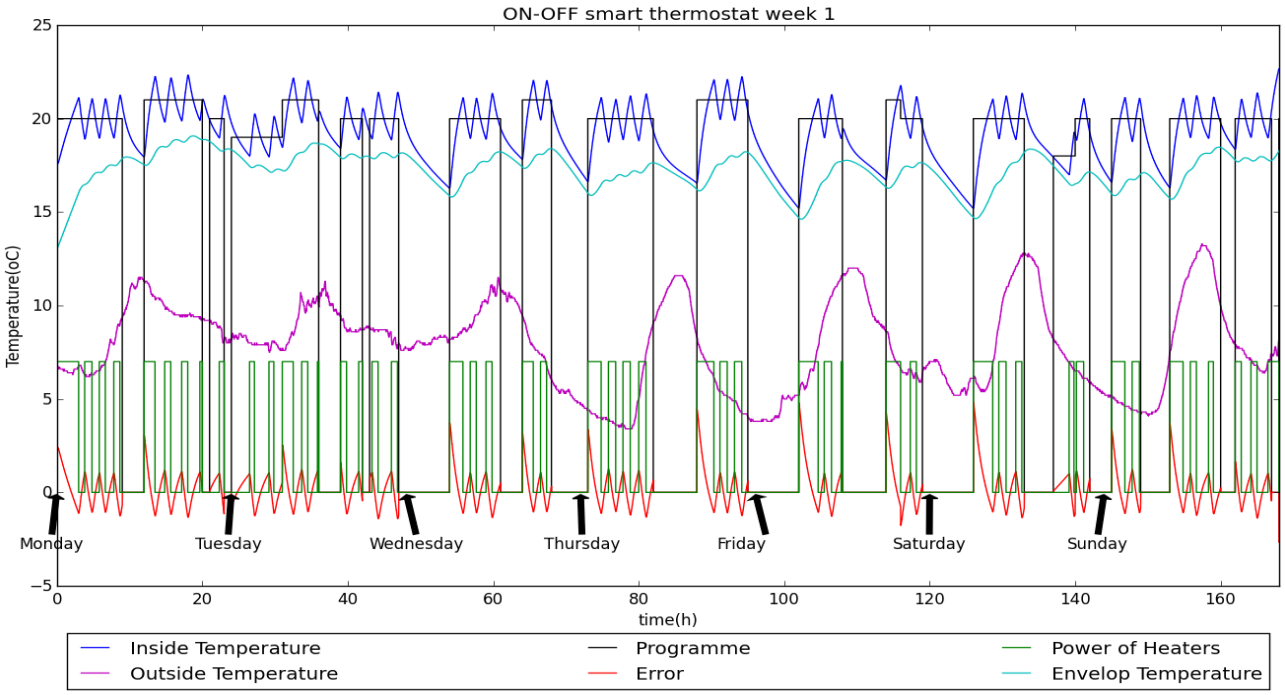
Στη συνέχεια ακολουθεί το διάγραμμα ροής δεδομένων όπου επεξηγείται αναλυτικά η λειτουργία του έξυπνου αλγορίθμου. Το διάγραμμα ροής δεδομένων είναι το ίδιο τόσο για το θερμοστάτη on-off όσο και για αυτόν που χρησιμοποιεί pid. Η μόνη διαφοροποίηση ανάμεσα στους δύο αλγορίθμους είναι ο χρόνος κατά τον οποίο θα είναι ανοικτή ή κλειστή η θέρμανση, που αυτό φάνηκε στα διαγράμματα 1 και 2. Στο διάγραμμα 3 λοιπόν, όταν αναφέρεται άνοιγμα θέρμανσης για 10 λεπτά, υπονοείται ότι στα συγκεκριμένα 10 λεπτά θα ακολουθηθεί η πολιτική ανοίγματος που ακολουθεί ο κάθε θερμοστάτης.



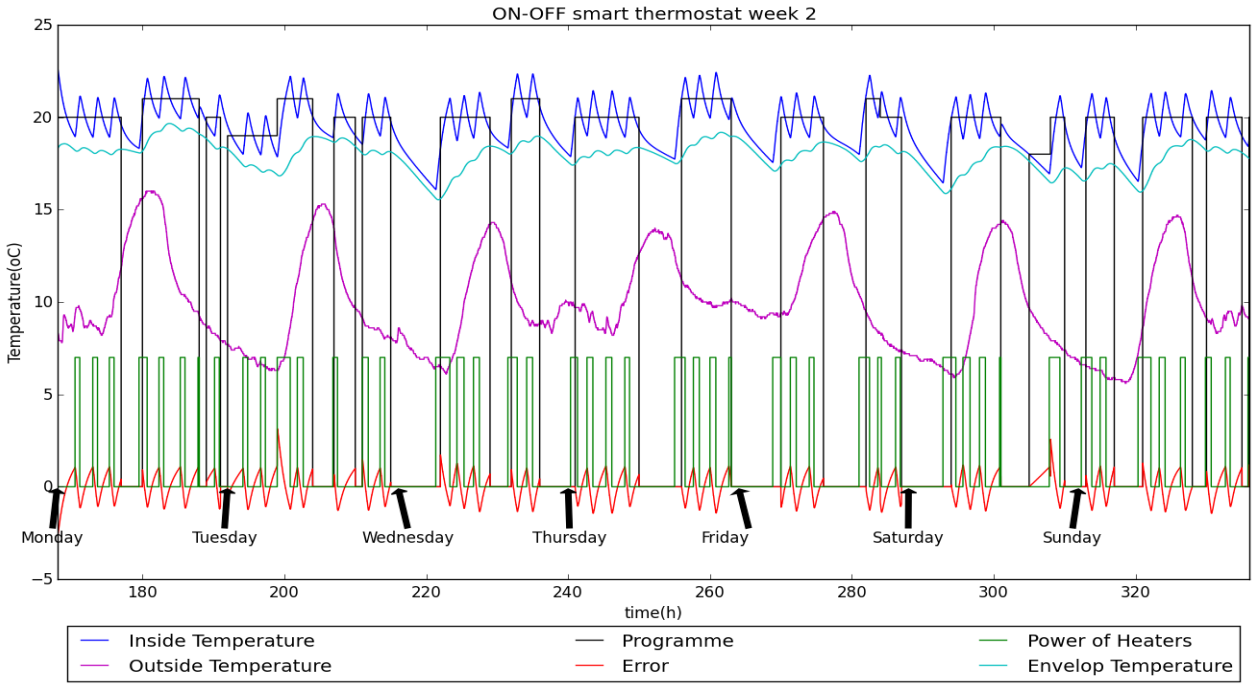
Διάγραμμα 6.3 - Διάγραμμα ροής δεδομένων για έξυπνο θερμοστάτη

6.3. Παρουσίαση έξυπνου θερμοστάτη

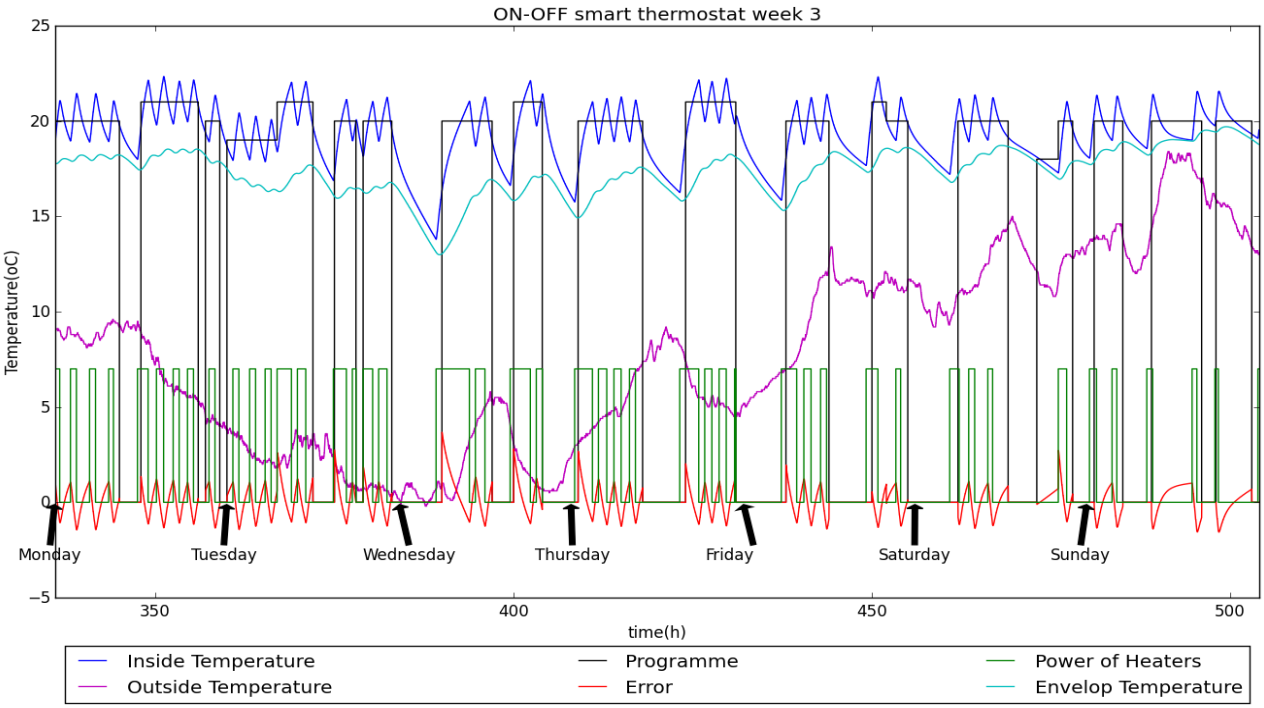
Αφού αναπτύχθηκε ο έξυπνος αλγόριθμος, μπορούν να γίνουν προσομοιώσεις για τους δύο θερμοστάτες, αυτή τη φορά χρησιμοποιώντας τον συγκεκριμένο αλγόριθμο. Οι προσομοιώσεις έγιναν για 3 συνεχόμενες εβδομάδες ώστε να μπορούν να παρατηρηθούν οι διαφορές προσαρμογές που κάνει ο αλγόριθμος. Πιο κάτω παρουσιάζονται τα αποτελέσματα της προσομοίωσης για τον θερμοστάτη on-off.



Παράσταση 6.5 - Θερμοστάτης on-off για το πρόγραμμα της πρώτης εβδομάδας

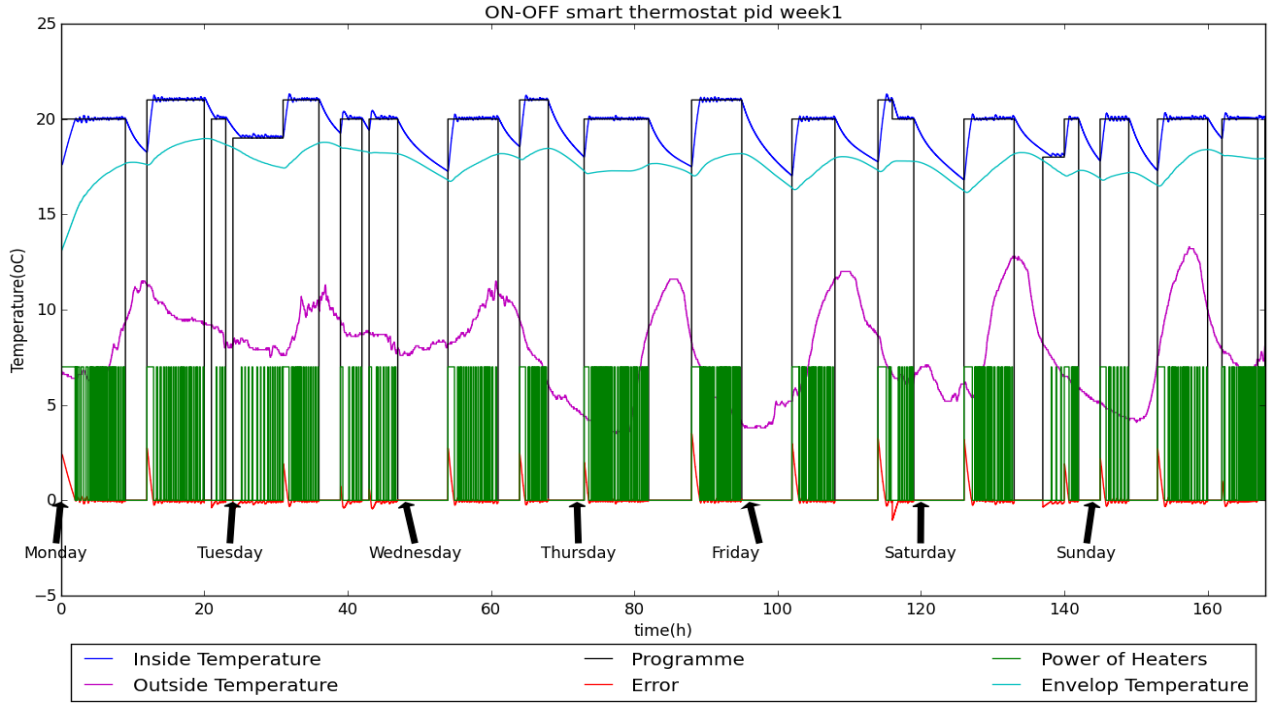


Παράσταση 5.6 - Θερμοστάτης on-off για το πρόγραμμα της δεύτερης εβδομάδας

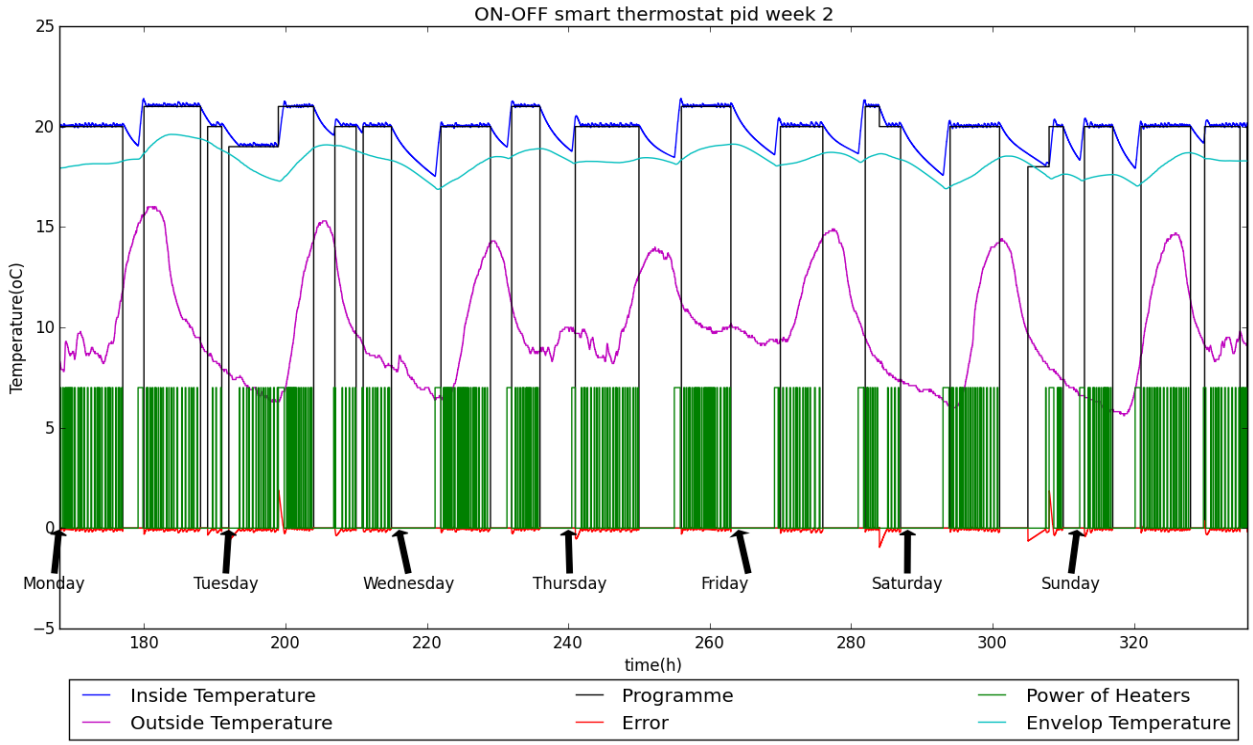


Παράσταση 6.7 - Θερμοστάτης on-off για το πρόγραμμα της τρίτης εβδομάδας

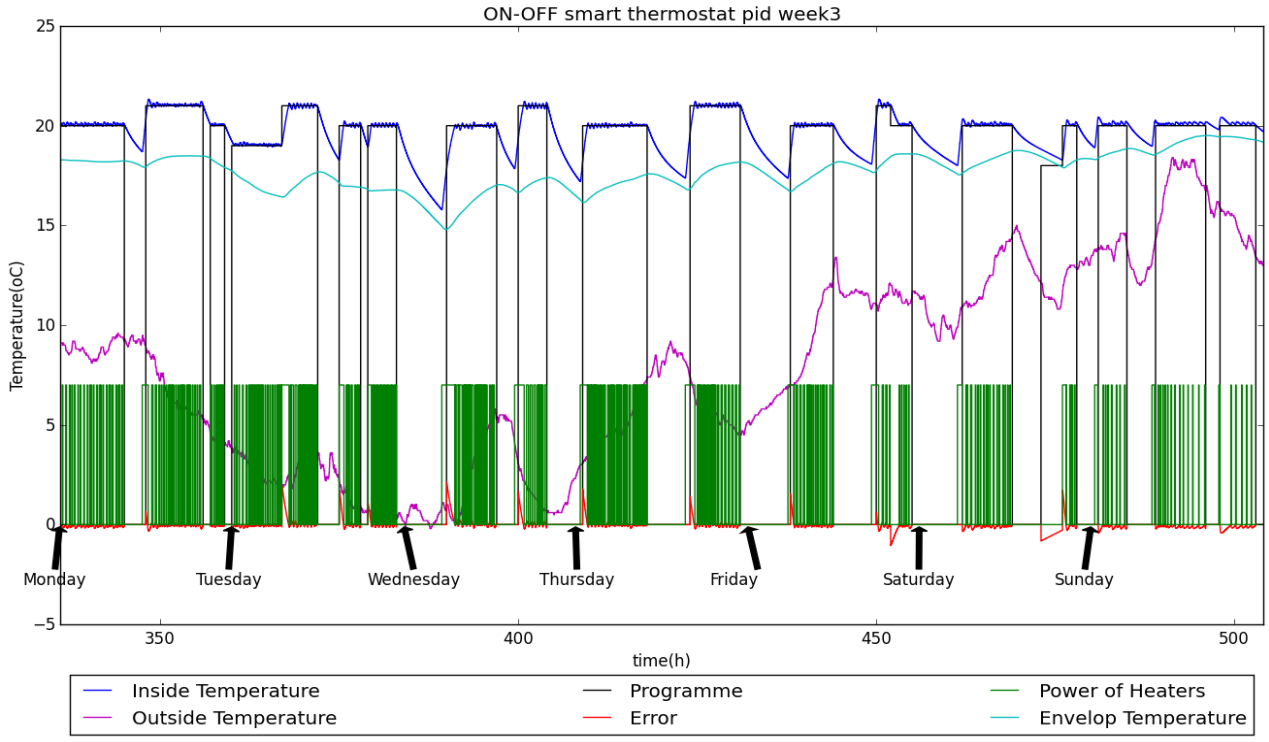
Στη συνέχεια ακολουθεί η προσομοίωση για το θερμοστάτη που χρησιμοποιεί τον ελεγκτή pid.



Παράσταση 6.8 - Θερμοστάτης με pid για το πρόγραμμα της πρώτης εβδομάδας



Παράσταση 6.9 - Θερμοστάτης με pid για το πρόγραμμα της δεύτερης εβδομάδας

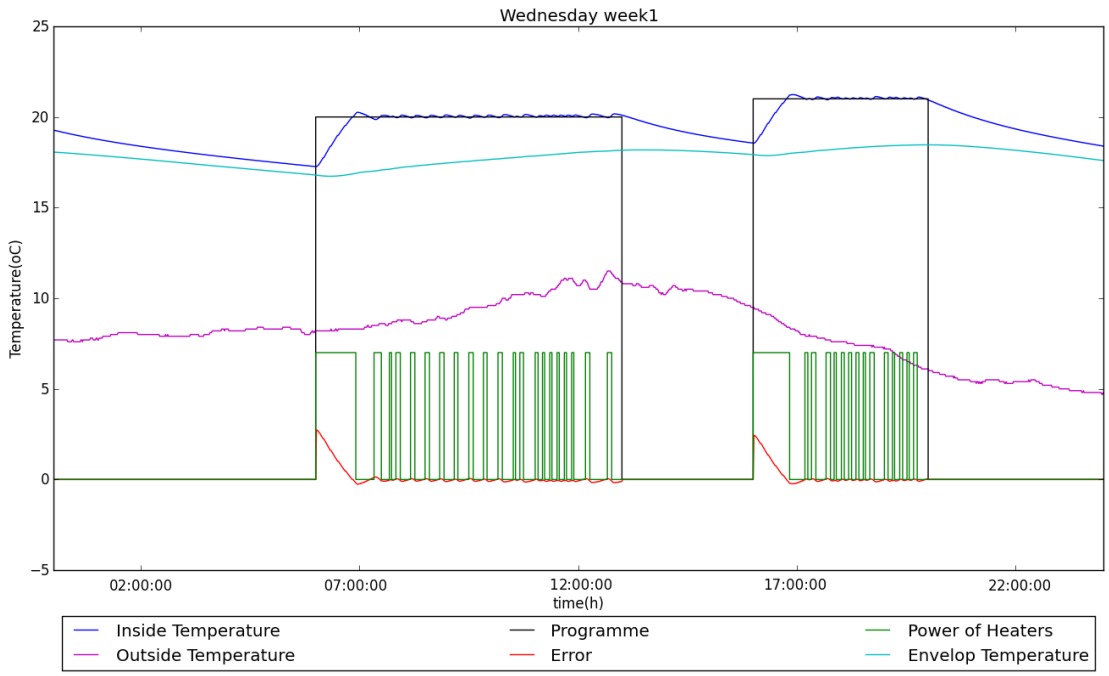


Παράσταση 6.10 - Θερμοστάτης με pid για το πρόγραμμα της τρίτης εβδομάδας

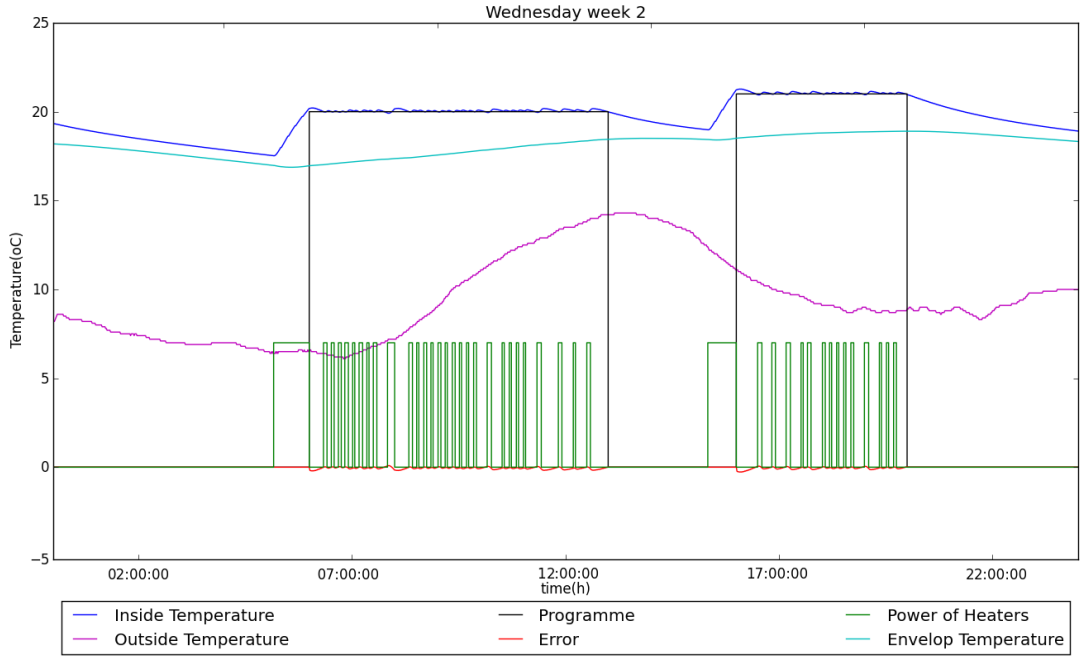
Και για τις δύο περιπτώσεις των θερμοστατών, παρατηρούμε ότι την πρώτη εβδομάδα επειδή η δομή είναι αρχικοποιημένη στο 0, η θέρμανση δεν ξεκινά να λειτουργεί πιο πριν από το διάστημα που καθορίζεται από το πρόγραμμα χρήστη, με αποτέλεσμα να παρατηρείται η ίδια συμπεριφορά με το όταν δεν υπήρχε ο έξυπνος αλγόριθμος. Ωστόσο, τη δεύτερη εβδομάδα παρατηρούμε ότι η θέρμανση ξεκινά να λειτουργεί από νωρίς αφού αποθηκεύτηκαν στη δομή στοιχεία από την πρώτη εβδομάδα. Σε κάποια διαστήματα παρατηρείται ότι δεν ξεκίνησε αρκετά πιο νωρίς η θέρμανση ενώ σε κάποια άλλα παρατηρείται ότι η θέρμανση δεν έπρεπε να ξεκινήσει να λειτουργεί τόσο πολύ πιο πρίν. Στα αποτελέσματα της τρίτης εβδομάδας παρατηρούμε κάποιες αλλαγές σε σχέση με τη δεύτερη κυρίως τις πρώτες ημέρες της εβδομάδας γιατί η εξωτερική θερμοκρασία είναι πολύ πιο χαμηλή σε σχέση με τη προηγούμενη εβδομάδα. Η διαφορά αυτή της εξωτερικής θερμοκρασία, δυσχερένει τη θέρμανση του δωματίου με αποτέλεσμα να μην πετυχαίνεται η ιδανική συμπεριφορά. Αντίθετα, σε περιπτώσεις όπου η εξωτερική θερμοκρασία είναι σχεδόν η ίδια ανάμεσα στη δεύτερη και στην τρίτη εβδομάδα, παρατηρούμε ότι ο θερμοστάτης λειτουργεί ιδανικά.

Στη συνέχεια παρουσιάζονται κάποιες περιπτώσεις πιο αναλυτικά για να μπορεί να ερμηνευτεί καλύτερα η συγκεκριμένη συμπεριφορά. Οι περιπτώσεις θα παρουσιαστούν μόνο για το θερμοστάτη που χρησιμοποιεί τον ελεγκτή pid αλλά αντίστοιχη συμπεριφορά και συμπεράσματα ισχύουν και για το θερμοστάτη on-off.

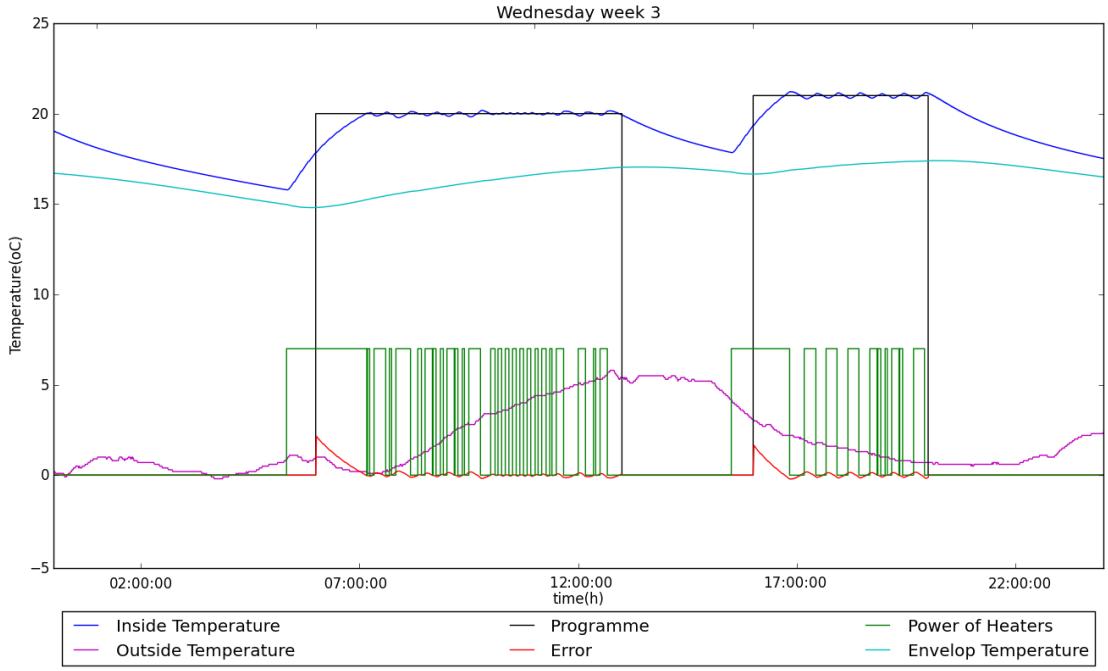
Περίπτωση 1:



Παράσταση 6.11 - Λειτουργία θερμοστάτη την Τετάρτη της πρώτης εβδομάδας



Παράσταση 6.12 - Λειτουργία θερμοστάτη την Τετάρτη της δεύτερης εβδομάδας



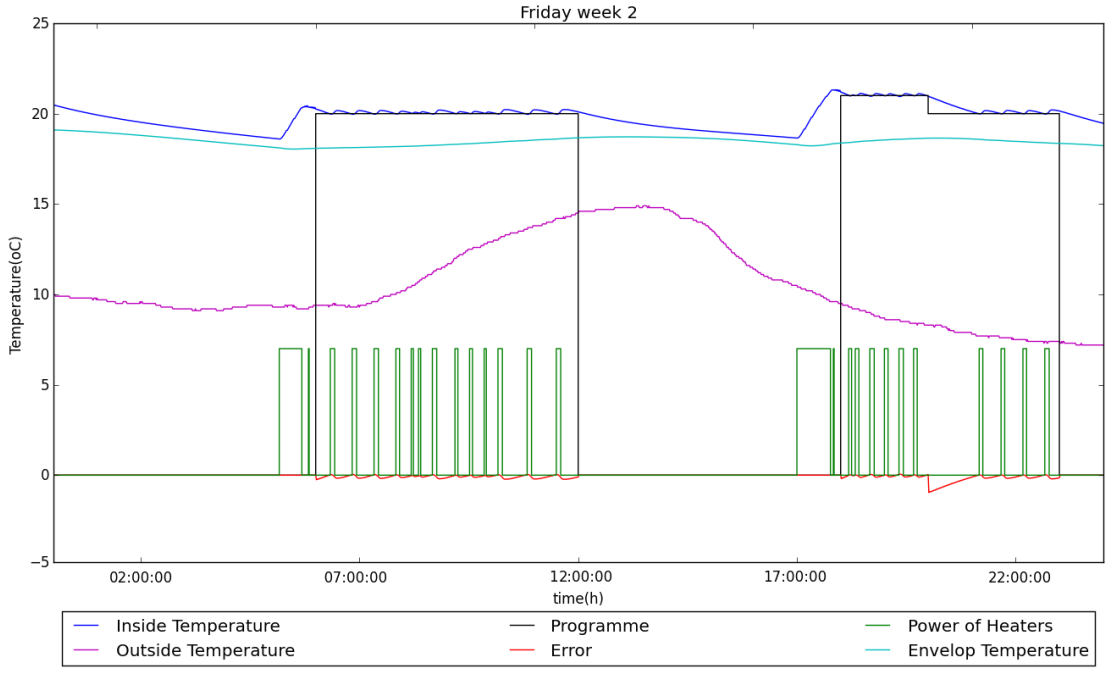
Παράσταση 6.13 - Λειτουργία θερμοστάτη την Τετάρτη της τρίτης εβδομάδας

Παρατηρούμε ότι τη δεύτερη εβδομάδα ο θερμοστάτης φροντίζει να ξεκινήσει τη λειτουργία του κάποιο χρόνο πιο πριν με αποτέλεσμα να πετυχαίνει την επιθυμητή θερμοκρασία, την επιθυμητή χρονική στιγμή. Οι τιμές της θερμοκρασίας περιβάλλοντος (εξωτερική θερμοκρασία) ανάμεσα στις δύο εβδομάδες έχουν μια πολύ μικρή διαφορά, αφού η δεύτερη εβδομάδα ειναι ελαφρώς πιο ζεστή, και για αυτό το λόγο παρατηρούμε ότι τη δεύτερη εβδομάδα η επιθυμητή θερμοκρασία πετυχαίνεται ελαφρώς πιο νωρίς. Ωστόσο, το αποτέλεσμα για τη δεύτερη εβδομάδα μπορεί να θεωρηθεί ιδανικό.

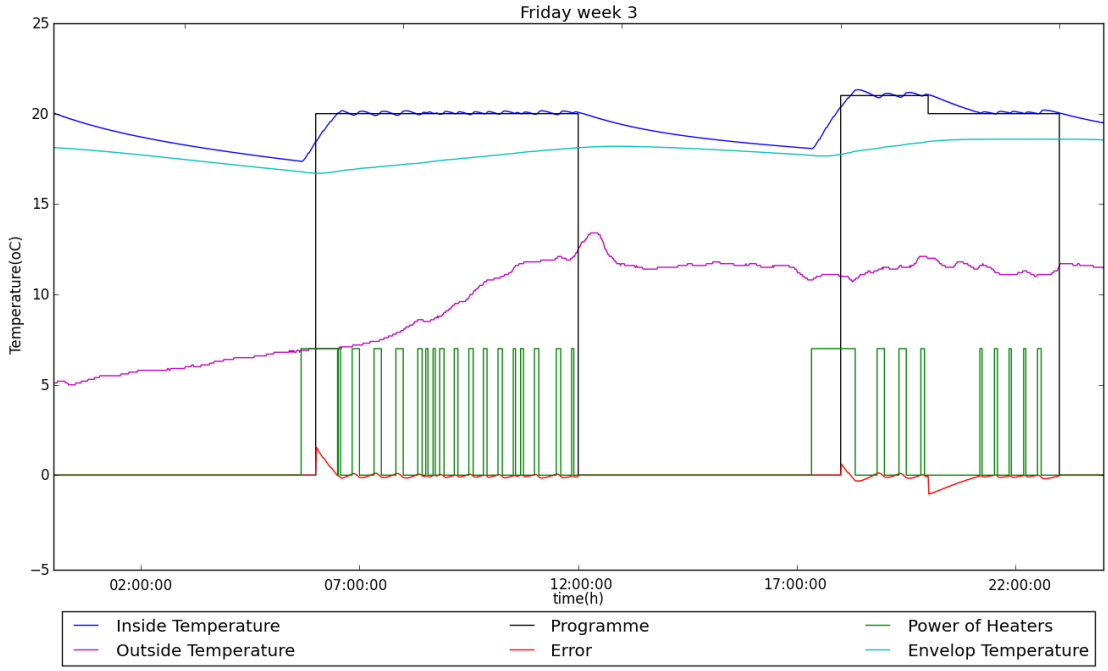
Με βάση το αποτελέσμα που παρατηρήθηκε πιο πριν, αναμενόταν ότι η τρίτη εβδομάδα θα είχε παρόμοια συμπεριφορά με τη δεύτερη αλλά όπως φαίνεται στο διάγραμμα 6.13, κάτι τέτοιο δεν ισχύει. Η συγκεκριμένη απόκλιση οφείλεται στη εξωτερική θερμοκρασία, αφού την τρίτη εβδομάδα είναι 5-8^{o}C πιο χαμηλή σε σχέση με τη δεύτερη και έτσι η χαμηλή εξωτερική θερμοκρασία καθυστερεί τη λειτουργία θέρμανσης του δωματίου.

Αξιοσημείωτο είναι να παρατηρηθεί ότι η απόκληση της πρώτης εβδομάδας και της τρίτης είναι σχεδόν η ίδια, με τη διαφορά ότι η τρίτη εβδομάδα είναι πολύ πιο ψυχρή από τη πρώτη (5-7^{o}C). Στη περίπτωση, λοιπόν, που χρησιμοποιόταν ένας απλός θερμοστάτης και όχι ένας έξυπνος, τα αποτελέσματα που θα λαμβάνονταν για την τρίτη εβδομάδα θα είχαν πολύ μεγάλο σφάλμα και θα άρα δεν θα ανταποκρίνονταν σχεδόν καθόλου στις επιθυμίες του χρήστη.

Περίπτωση 2:



Παράσταση 6.14 - Λειτουργία θερμοστάτη την Παρασκευή της δεύτερης εβδομάδας

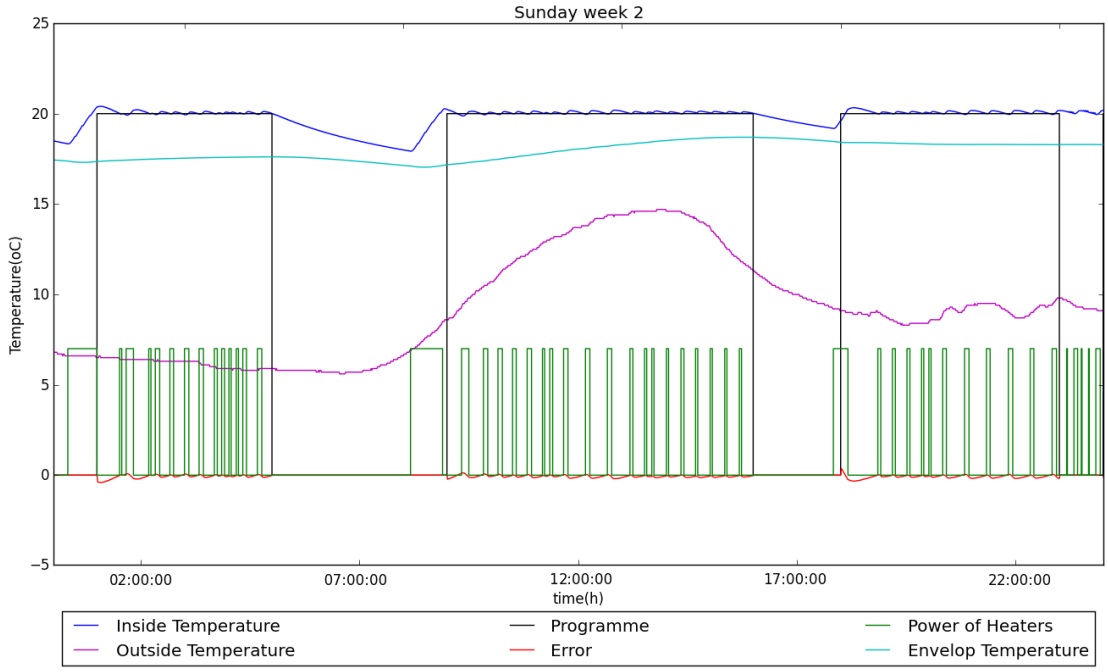


Παράσταση 6.15 - Λειτουργία θερμοστάτη την Παρασκευή της τρίτης εβδομάδας

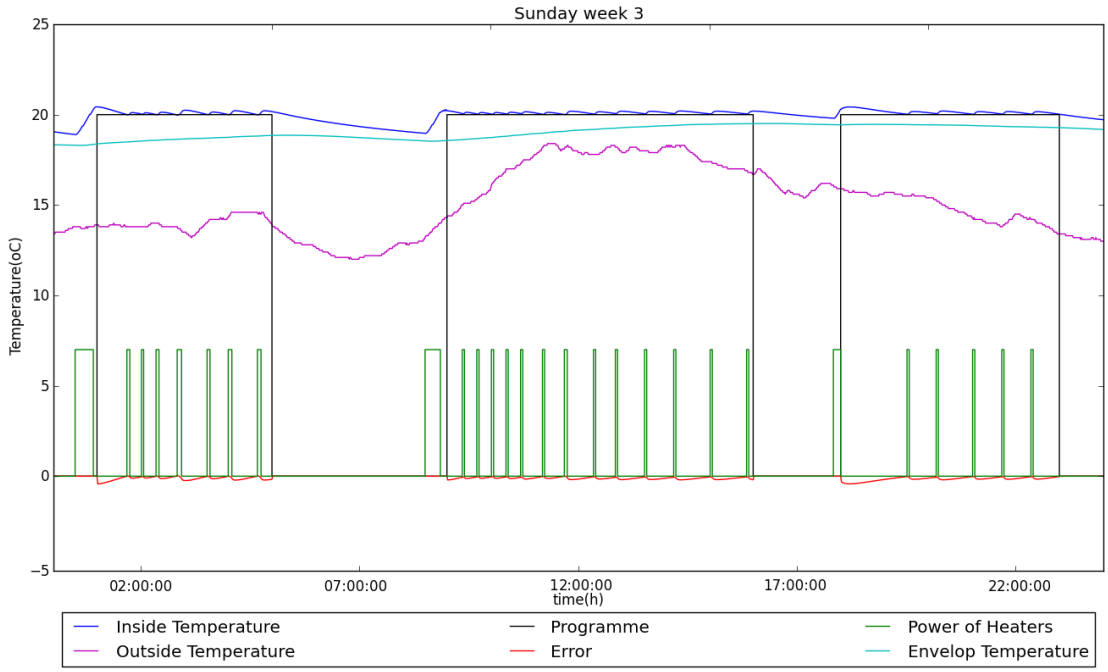
Στη συγκεκριμένη περίπτωση, παρατηρούμε ότι η πρόβλεψη για τη δεύτερη εβδομάδα είχε κάποιο σφάλμα, αφού η επιθυμητή θερμοκρασία επιτεύχθηκε πριν από τη χρονική στιγμή που όρισε ο χρήστης. Για να διορθωθεί το συγκεκριμένο σφάλμα για την τρίτη εβδομάδα, μειώθηκε στη δομή η τιμή που υπήρχε έτσι ώστε να καθυστερήσει περισσότερο να ξεκινήσει η θέρμανση τη τρίτη εβδομάδα. Άρα, αναμένουμε ότι τη τρίτη εβδομάδα θα έχουμε κάποια σχεδόν ιδανική συμπεριφορά.

Τα αποτελέσματα που παρατηρούμε όμως για την τρίτη εβδομάδα χωρίζεται σε δυο υποπεριπτώσεις, μία για κάθε διάστημα. Το πρώτο διάστημα ακολουθεί τη συμπεριφορά της περίπτωσης 1, αφού τη τρίτη εβδομάδα η εξωτερική θερμοκρασία ήταν πιο χαμηλή και έτσι δημιούργησε καθυστέρηση στη θέρμανση του δωματίου. Αντίθετα, παρατηρούμε ότι στο δεύτερο διάστημα η εξωτερική θερμοκρασία τη τρίτη εβδομάδα είναι ελαφώς πιο υψηλή από αυτή της δεύτερης εβδομάδας, ενώ τη χρονική στιγμή που ξεκινά η θέρμανση τη τρίτη εβδομάδα έχουν σχεδόν την ίδια εξωτερική θερμοκρασία. Αυτή η συμπεριφορά της εξωτερική θερμοκρασίας έδωσε σαν αποτέλεσμα μια σχεδόν ιδανική συμπεριφορά στην εσωτερική θερμοκρασία, αφού η απόκλιση που δημιουργήθηκε είναι πολύ μικρή.

Περίπτωση 3:



Παράσταση 6.16 - Λειτουργία θερμοστάτη την Κυριακή της δεύτερης εβδομάδας



Παράσταση 6.17 - Λειτουργία θερμοστάτη την Κυριακή της τρίτης εβδομάδας

Στη συγκεκριμένη περίπτωση παρατηρούμε ότι τη δεύτερη εβδομάδα η επιθυμητή θερμοκρασία επιτεύχθηκε πιο νωρίς από όταν ήθελε ο χρήστης, όπως και στην περίπτωση 2. Τη τρίτη εβδομάδα όμως παρόλου που η θέρμανση ξεκίνησε τη λειτουργεία της πιο αργά, η επιθυμητή τιμή και πάλι επιτεύχθηκε πιο νωρίς σε σχέση με τον ορισμό του προγράμματος χρήστη.

Η συγκεκριμένη συμπεριφορά, οφείλεται στο γεγονός ότι την τρίτη εβδομάδα, η εξωτερική θερμοκρασία κυμαίνεται σε πιο υψηλά επίπεδα σε σχέση με τη δεύτερη εβδομάδα, με αποτέλεσμα να βοηθάει στη λειτουργία θέρμανσης του δωματίου. Ακόμα, λόγω των υψηλών θερμοκρασιών που παρατηρήθηκαν την τρίτη εβδομάδα η θέρμανση λειτουργεί για πιο μικρά χρονικά διαστήματα, που είναι λογικό.

6.4. Συγκρίσεις - Συμπεράσματα

Έχοντας παρουσιάσει τα αποτελέσματα που προσφέρει ένας απλός και ένας έξυπνος θερμοστάτης, εύκολα γίνεται εμφανές η καλύτερη λειτουργία που προσφέρει ο έξυπνος. Ουσιαστικά, τα αποτελέσματα της πρώτης εβδομάδας με χρήση έξυπνου θερμοστάτη είναι τα ίδια με αυτά του απλού, όμως μετά το πέρας της πρώτης εβδομάδας ο έξυπνος προσαρμόζει καλύτερα της επιθυμητή θερμοκρασία στο χρονικά διαστήματα που ορίζει ο χρήστης.

Ακόμα, γίνεται φανερό πόσο μεγάλη σημασία έχει στη λειτουργία των θερμοστατών η διακύμανση της εξωτερικής θερμοκρασίας. Αυτές οι διακυμάνσεις μπορούν να προκαλέσουν αποκλίσεις στη λειτουργία του έξυπνου θερμοστάτη και έτσι να μη λάβουμε τελικά ιδανικά αποτελέσματα για κάποιες ημέρες, παρόλο που με βάση τις προηγούμενες εβδομάδες είχαμε ιδανική λειτουργία. Ωστόσο, αυτές οι αποκλίσεις θα ήταν ακόμα πιο μεγάλες αν χρησιμοποιόταν ένας απλός θερμοστάτης ενώ στη χειρότερη περίπτωση θα έχει την ίδια απόκλιση με τον απλό θερμοστάτη.

Η χρήση λοιπόν του έξυπνου θερμοστάτη για τη θέρμανση κάποιου δωματίου ή κάποιου κτιρίου, προσφέρει την εγγύηση ότι θα υπάρξουν καλύτερα αποτελέσματα από κάποιον άλλο απλό θερμοστάτη. Η χρήση τους, βεβαιώνει το χρήστη ότι θα λάβει την θερμοκρασία που επιθυμεί, όταν την επιθυμεί. Στις περιπτώσεις όπου υπάρχει η απόκλιση και δεν λαμβάνεται η επιθυμητή θερμοκρασία, η επίπτωση δεν έχει τόσο μεγάλη σημασία στις πιο πολλές περιπτώσεις. Αυτό οφείλεται στο ότι, όταν αυτή η απόκλιση είναι αρνητική (πετυχαίνεται η επιθυμητή θερμοκρασία πριν το επιθυμητό διάστημα), ο χρήστης πάλι θα λάβει τη θερμοκρασία που όρισε στο διάστημα που όρισε, απλά θα έχει κάποια μικρή επίπτωση σε περιττή κατανάλωση ενέργειας. Στη περίπτωση που η απόκλιση είναι θετική και άρα ο χώρος δεν έλαβε την σωστή θερμοκρασία όταν έπρεπε, τις πιο πολλές φορές η θερμοκρασία του δωματίου δεν βρίσκεται πολύ πιο χαμηλά από την επιθυμητή. Έτσι μπορεί ο χρήστης να μην έλαβε τους 20^{o}C που ζητούσε, όμως δεν θα νιώσει τη διαφορά αν λάβει 19^{o}C.

Ο έξυπνος θερμοστάτης λοιπόν ανταποκρίνεται στις απαιτήσεις του χρήστη και πετυχαίνει το σκοπό του, να θερμαίνει δηλαδή το δωμάτιο-κτίριο σε επιθυμητές τιμές. Ακόμα, στις περιπτώσεις που δεν καταφέρνει να πετύχει την επιθυμητή θερμοκρασία, εξασφαλίζει ότι η

θερμοκρασία του δωματίου-κτιρίου θα κυμαίνεται σε πολύ κοντινές θερμοκρασίες από την επιθυμητή και άρα σίγουρα δεν υπάρχει περίπτωση ο χρήστης να νιώσει δυσφορία από το κρύο.

Αξιοσημείωτο επίσης είναι το ότι ο θερμοστάτης που χρησιμοποιεί τον ελεγκτή pid πετυχένει πολύ καλύτερα αποτελέσματα από τον απλό on-off όπως έχει σχολιαστεί και στο κεφάλαιο 5. Λόγω της φύσης του θερμοστάτη on-off να έχει μια υστέρηση $\pm 1^{o}C$ από την επιθυμητή θερμοκρασία, τα αποτελέσματα που λαμβάνουμε για τον έξυπνο θερμοστάτη on-off έχουν αισθητή διαφορά με αυτά του έξυπνου θερμοστάτη που χρησιμοποιεί ελεγκτή pid.

Στη περίπτωση λοιπόν που θα έπρεπε να αποφασιστεί ποιός θερμοστάτης θα κατασκευαστεί για να χρησιμοποιηθεί σε πραγματικό χρόνο (real-time), η κατάλληλη επιλογή θα ήταν ο έξυπνος θερμοστάτης που κάνει χρήση ελεγκτή pid.

ΚΕΦΑΛΑΙΟ 7. ΣΥΝΟΨΗ

Στο συγκεκριμένο κεφάλαιο παρουσιάζονται τα τελικά συμπεράσματα για την ανάπτυξη του έξυπνου θερμοστάτη. Επίσης, αναλύονται και κάποιες επεκτάσεις που θα μπορούσαν να γίνουν στην παρούσα εργασία.

7.1. Συμπεράσματα

Στόχος της συγκεκριμένης εργασία ήταν υλοποίηση ενός έξυπνου θερμοστάτη, ο οποίος θα προσέφερε χαρακτηριστικά παρόμοια με αυτά των εμπορικών έξυπνων θερμοστατών. Κατά την υλοποίηση κατανοήθηκε ότι δεν ήταν δυνατή η εξ ολοκλήρου υλοποίηση σε υλικό του θερμοστάτη, λόγων των υψηλών θερμοκρασιών που παρουσιάζονται στην Αθήνα το καλοκαίρι. Για αυτό το λόγο, κατασκευάστηκε πρακτικά μόνο το κομμάτι του θερμομέτρου για παρατήρηση της θερμοκρασίας του δωματίου, καθώς επίσης αναπτύχθηκε και μία διαδικτυακή εφαρμογή για την παρουσίαση και τον έλεγχο του θερμοστάτη.

Αφού ολοκληρώθηκε η κατασκευή του θερμομέτρου με τη βοήθεια του Raspberry Pi 2 και η ανάπτυξη της εφαρμογής από όπου δίνεται το εβδομαδιαίο πρόγραμμα, ξεκίνησαν να μελετούνται κάποια θερμικά ισοδύναμα μοντέλα κτιρίων. Τα συγκεκριμένα μοντέλα ήταν αναγκαία για τις προσομοιώσεις αφού δεν θα κατασκευαζόταν ολοκληρωτικά με υλικό ο έξυπνος θερμοστάτης. Για κάθε μοντέλο, αναπτύχθηκε ένας απλός θερμοστάτης και ένας θερμοστάτης που χρησιμοποιεί pid ελεγκτή. Στη συνέχεια, αφού πάρθηκαν τα αποτελέσματα για κάθε μοντέλο και για τους δύο θερμοστάτες, καταλήξαμε στο συμπέρασμα ότι το μοντέλο TiTeThTs προσφέρει τα πιο κοντινά στη πραγματικότητα αποτελέσματα και ότι ο θερμοστάτης που χρησιμοποιεί ελεγκτή pid, προσφέρει καλύτερα και πιο αποδοτικά αποτελέσματα σε σχέση με αυτό που δεν χρησιμοποιεί pid.

Το επόμενο βήμα ήταν η ανάπτυξη του έξυπνου αλγορίθμου, όπου έγινε με τη χρήση του μοντέλου TiTeThTs, αφού πάλι τα αποτελέσματα τα λάβαμε για προσομοιώσεις. Η χρήση του έξυπνου αλγορίθμου τελικά επέτρεψε τη βελτίωση των αποτελεσμάτων που είχαμε λάβει στο πρηγούμενο βήμα, αφού ο έξυπνος αλγόριθμος που αναπτύχθηκε επέτρεψε το άνοιγμα της θέρμανσης πιο νωρίς από όταν το όριζε το πρόγραμμα χρήστη.

Το τελικό συμπέρασμα της παρούσας εργασίας είναι ότι ένας έξυπνος θερμοστάτης μπορεί να υλοποιηθεί και να είναι αρκετά αποτελεσματικός, βέλτιστος και να προσαρμόζεται στις επιθυμίες του χρήστη. Τέλος, από όλους τους θερμοστάτες που έχουν εξεταστεί, παρατηρήθηκε όπως και αναμέναμε, ότι αυτός που δίνει τα καλύτερα αποτελέσματα είναι ο έξυπνος θερμοστάτης με χρήση ελεγκτή pid.

7.2. Προτεινόμενες Επεκτάσεις

Μετά το πέρας της εργασίας, παρατηρήθηκαν κάποια σημεία τα οποία θα μπορούσαν να βελτιστοποιηθούν και κάποια άλλα τα οποία θα μπορούσαν να προστεθούν. Στη παρούσα παράγραφο λοιπόν περιγράφονται οι συγκεκριμένες επεκτάσεις που προτείνονται.

- Κατασκευή του έξυπνου θερμοστάτη με υλικό για να μπορούν να ληφθούν σε πραγματικό χρόνο και δεδομένα, αποτελέσματα.
- Κατασκευή θερμομέτρου και για την εξωτερική θερμοκρασία. Στη παρούσα εργασία, η εξωτερική θερμοκρασία λαβάνεται από κάποιο σταθμό στην Αθήνα μέσω της υπηρεσίας OpenWeatherMaps και για αυτό το λόγο μπορεί να παρουσιάζει κάποια απόκλιση από την θερμοκρασία που υπάρχει στη συγκεκριμένη περιοχή του σπιτιού.
- Ανάλυση για τη λειτουργία του κληματιστικού, ώστε για τη θέρμανση του δωματίου να χρησιμοποιείται το κλιματιστικό.
- Επέκταση του έξυπνου αλγορίθμου ώστε να λαμβάνει υπόψιν και την εξωτερική θερμοκρασία. Από τα αποτελέσματα που λήφθησαν για τον έξυπνο θερμοστάτη, παρατηρήθηκε ότι η εξωτερική θερμοκρασία μπορεί να αλλοιώσει τη λειτουργία του θερμοστάτη όταν παρουσιάζει μεγάλες διακυμάνσεις.
- Ενσωμάτωση πρόβλεψης της εξωτερικής θερμοκρασίας στον έξυπνο αλγόριθμο. Στον παρόν έξυπνο αλγόριθμο, οι αλλαγές στο χρόνο του πόσο πιο νωρίς πρέπει να ξεκινήσει η θέρμανση γίνονται αντιληπτές μετά από μία εβδομάδα. Εάν ενσωματωθεί η πρόβλεψη της εξωτερικής θερμοκρασίας, θα υπάρχει η δυνατότητα να προσαρμόζεται αυτός ο χρόνος για την ίδια εβδομάδα.
- Η ανάπτυξη παραστάσεων στη διαδικτυακή εφαρμογή όπου θα φαίνεται η ορθή λειτουργία του θερμοστάτη με βάση τις θερμοκρασίες και έτσι θα διευκολύνει το χρήστη στο να κατανοήσει αν επιθυμεί να κάνει κάποιες αλλαγές.
- Εμφάνιση στη διαδικτυακή εφαρμογή του συνολικού χρόνου λειτουργίας της θέρμανσης, ώστε να μπορεί να υπολογιστεί η συνολική ενέργεια και το κόστος που καταναλώνει ο θερμοστάτης.
- Προσθήκη ρύθμησης ώστε ο χρήστης να μπορεί να αλλάξει τη θερμοκρασία που επιθυμεί, να ανοίξει ή να κλείσει τη θέρμανση όταν αυτός επιθυμεί και ανεξάρτητα από το πρόγραμμα. Επίσης να υπάρχει υποστήρηξη για “All day” και “Away” για ενεργοποίηση ή απενεργοποίηση αντίστοιχα τη θέρμανσης για όλη την ημέρα ανεξάρτητα από το πρόγραμμα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Ηλεκτρονική Βιβλιογραφία

[1] Wikipedia. Smart Meter. [online]. https://en.wikipedia.org/wiki/Smart_meter

[2] Google NEST thermostats. [online]. https://nest.com/

[3] Martyn Warwick (2016). You wait months for a smart thermostats report and then several arrive together, just like London buses. [online] http://www.telecomtv.com/articles/iot/you-wait-months-for-a-smart-thermostats-report-and-then-several-arrive-together-just-like-london-buses-13636/

[4] Wikipedia. Internet of things. [online]. https://en.wikipedia.org/wiki/Internet_of_Things

[5] Wikipedia. Smart traffic light. [online]. https://en.wikipedia.org/wiki/Smart_traffic_light

[6] Wikipedia. Arduino. [online]. https://en.wikipedia.org/wiki/Arduino

[7] Wikipedia. Raspberry Pi. [online]. https://en.wikipedia.org/wiki/Raspberry_Pi

[8] Wikipedia. 1 wire. [online]. https://en.wikipedia.org/wiki/1-Wire

[9] Wikipedia. Thermostat. [online]. https://en.wikipedia.org/wiki/Thermostat

[10] Wikipedia. Controller. [online]. https://en.wikipedia.org/wiki/Controller_%28control_theory%29

[11] Wikipedia. Integral. [online]. https://en.wikipedia.org/wiki/Integral

[12] Wikipedia. PID controller. [online]. https://en.wikipedia.org/wiki/PID_controller

[13] Wikipedia. Duty cycle. [online]. https://en.wikipedia.org/wiki/Duty_cycle

[14] Erik Cheever (2015). Thermal Systems Background. [online]. http://lpsa.swarthmore.edu/Systems/Thermal/SysThermalIntro.html

[15] Erik Cheever (2015). Systems Elements. [online]. http://lpsa.swarthmore.edu/Systems/Thermal/SysThermalElem.html

[16] Erik Cheever (2015). Mathematical Models of Thermal Systems. [online]. http://lpsa.swarthmore.edu/Systems/Thermal/SysThermalModel.html

[17] Wikipedia. Smart products. [online]. https://en.wikipedia.org/wiki/Smart_products

[18] Wikipedia. Smart device. [online]. https://en.wikipedia.org/wiki/Smart_device

[19] Wikipedia. Smart system. [online]. https://en.wikipedia.org/wiki/Smart_system

[20] Wikipedia. Graphical user interface. [online]. https://en.wikipedia.org/wiki/Graphical_user_interface

[21] Simon Monk. Adafruit's Raspberry Pi Lesson 11.DS18B20 Temperature Sensing. Hardware. [online]. https://learn.adafruit.com/adafruits-raspberry-pi-lesson-11-ds18b20-temperature-sensing/hardware

[22] Simon Monk. Adafruit's Raspberry Pi Lesson 11.DS18B20 Temperature Sensing. DS18B20. [online]. https://learn.adafruit.com/adafruits-raspberry-pi-lesson-11-ds18b20-temperature-sensing/ds18b20

[23] Tobias Oetiker (2015). rrdtool. [online]. http://oss.oetiker.ch/rrdtool/doc/rrdtool.en.html

[24] Tobias Oetiker (2015). rrdcreate. [online]. http://oss.oetiker.ch/rrdtool/doc/rrdcreate.en.html

[25] Tobias Oetiker (2015). rrdbuild. [online]. https://oss.oetiker.ch/rrdtool/doc/rrdbuild.en.html

[26] OpenWeatherMap, Inc. How to start. [online]. http://openweathermap.org/appid

[27] OpenWeatherMap, Inc. Current weather data. [online]. http://openweathermap.org/current#one

[28] Highcharts. [online]. http://www.highcharts.com/

[29] Highcharts. Highcharts, Highstock and Highmaps documentation. [online] http://www.highcharts.com/docs/

[30] Highcharts. Compatibility. [online]. http://www.highcharts.com/documentation/compatibility

[31] Highcharts. Highcharts. [online]. http://www.highcharts.com/products/highcharts

[32] Wikipedia. Bootstrap. [online]. https://el.wikipedia.org/wiki/Bootstrap

[33] Bootstrap. [online]. http://getbootstrap.com/

[34] No-IP. Why Us? [online]. http://www.noip.com/why-us

[35] No-IP. Dynamic DNS. [online]. http://www.noip.com/remote-access

[36] No-IP. How to install the No-IP DUC on Raspberry Pi. [online]. http://www.noip.com/support/knowledgebase/install-ip-duc-onto-raspberry-pi/

Βιβλιογραφική

[1B] Siemens Switzerland Ltd, 2013. Control technology Brochure (2007). Answers for infrastructure

[2B] Bacher P., Madsen H., (2010), Procedure for identifying models for the heat dynamics of buildings, publication of Technical Univeristy of Denmark

Bacher P., Madsen H., 2011. Identifying suitable models for the heat dynamics of buildings. Energy and Buildings 43 (2011), 1511–1522

Madsen H. , Hoist J. 1994. Estimation of continuous-time models for the heat dynamics of a building. Energy and Buildings 22 (1995), 67-79

Παράρτημα Α – Πηγές Εικόνων

Εικόνα 1.1 : http://osarena.net/internet-things-diadiktyo-ton-pragmaton-ti-einai-ayto

Εικόνα 2.1: https://www.maximintegrated.com/en/app-notes/index.mvp/id/126

Εικόνα 2.2: Siemens Switzerland Ltd, 2013. Control technology Brochure (2007). Answers for infrastructure

Εικόνα 2.3: Siemens Switzerland Ltd, 2013. Control technology Brochure (2007). Answers for infrastructure

Εικόνα 2.4: Siemens Switzerland Ltd, 2013. Control technology Brochure (2007). Answers for infrastructure

Εικόνα 2.5: Siemens Switzerland Ltd, 2013. Control technology Brochure (2007). Answers for infrastructure

Εικόνα 2.7: http://hyperphysics.phy-astr.gsu.edu/hbase/integ.html

Εικόνα 2.8: Siemens Switzerland Ltd, 2013. Control technology Brochure (2007). Answers for infrastructure

Εικόνα 2.9: Siemens Switzerland Ltd, 2013. Control technology Brochure (2007). Answers for infrastructure

Εικόνα 2.11: Siemens Switzerland Ltd, 2013. Control technology Brochure (2007). Answers for infrastructure

Εικόνα 2.12: https://en.wikipedia.org/wiki/PID_controller

Εικόνα 2.13: https://onion.freshdesk.com/support/solutions/articles/6000081222-how-do-i-use-the-servo-expansion-

Εικόνα 2.14: http://lpsa.swarthmore.edu/Systems/Thermal/SysThermalElem.html

Εικόνα 2.15: http://lpsa.swarthmore.edu/Systems/Thermal/SysThermalElem.html

Εικόνα 2.16: http://lpsa.swarthmore.edu/Systems/Thermal/SysThermalModel.html

Εικόνα 2.17: Bacher P., Madsen H., 2011. Identifying suitable models for the heat dynamics of buildings. Energy and Buildings 43 (2011), 1511–1522

Εικόνα 2.18: Bacher P., Madsen H., (2010), Procedure for identifying models for the heat dynamics of buildings, publication of Technical Univeristy of Denmark

Εικόνα 2.19: Bacher P., Madsen H., (2010), Procedure for identifying models for the heat dynamics of buildings, publication of Technical Univeristy of Denmark

Εικόνα 2.20: Bacher P., Madsen H., (2010), Procedure for identifying models for the heat dynamics of buildings, publication of Technical Univeristy of Denmark

Εικόνα 2.21: Bacher P., Madsen H., (2010), Procedure for identifying models for the heat dynamics of buildings, publication of Technical Univeristy of Denmark

Εικόνα 3.1: http://grobotronics.com/raspberry-pi-2-model-b-armv7-with-1g-ram.html

Εικόνα 3.2: http://grobotronics.com/temperature-sensor-waterproof-ds18b20.html?sl=el

Εικόνα 3.3: http://grobotronics.com/assembled-pi-t-cobbler-plus-gpio-breakout-for-raspberry-pi-b.html

Εικόνα 3.4: http://www.rs-online.com/designspark/electronics/eng/blog/introducing-the-raspberry-pi-b-plus

Εικόνα 3.5. A: http://projects-raspberry.com/monitor-your-home-temperature-using-your-raspberry-pi/

Εικόνα 3.7: https://www.howtoforge.com/tutorial/install-apache-with-php-and-mysql-lamp-on-debian-jessie/

Εικόνα 3.12: https://en.wikipedia.org/wiki/Cron

Εικόνα 3.13: http://www.highcharts.com/demo/combo

Εικόνα 5.1: Bacher P., Madsen H., (2010), Procedure for identifying models for the heat dynamics of buildings, publication of Technical Univeristy of Denmark

Παράρτημα Β – Αρχεία Κώδικα και Πηγές

Για τις βάσεις δεδομένων χρησιμοποιήθηκαν τα:

1. database.py για τη δημιουργία της βάσης για τις θερμοκρασίες του δωματίου
   Tobias Oetiker (2015). rrdcreate. [online].
   http://oss.oetiker.ch/rrdtool/doc/rrdcreate.en.html
2. temperature.rrd το αρχείο που δημιουργείται για τη βάση των θερμοκρασιών δωματίου
3. thermometer.py για ανάγνωση θερμοκρασίας από αισθητήρα και εισαγωγή στη βάση δεδομένων
   Simon Monk. Adafruit's Raspberry Pi Lesson 11.DS18B20 Temperature Sensing. Hardware. [online]. https://learn.adafruit.com/adafruits-raspberry-pi-lesson-11-ds18b20-temperature-sensing/software
   bubbl(2014). Temperature Log|How to. [online]
   https://blog.bartbania.com/raspberry_pi/temperature-log-howto/
4. outside_database.py για τη δημιουργία της βάσης για τις εξωτερικές θερμοκρασίες
   Tobias Oetiker (2015). rrdcreate. [online].
   http://oss.oetiker.ch/rrdtool/doc/rrdcreate.en.html
5. out_temperature.rrd το αρχείο που δημιουργείται για τη βάση των εξωτερικών θερμοκρασιών
6. out_temp.py για την εισαγωγή της εξωτερικής θερμοκρασίας στη βάση δεδομένων
   Tobias Oetiker (2015). rrdupdate. [online].
   http://oss.oetiker.ch/rrdtool/doc/rrdupdate.en.html

Για την ανάπτυξη της διαδικτυακής εφαρμογής χρησιμοποιήθηκαν τα:

7. index.php για την σελίδα εισόδου (login)
   Ashbo. Clean Modal Login Form. [online].
   http://bootsnipp.com/snippets/featured/clean-modal-login-form
8. index.css για τη μορφοποίηση της σελίδας εισόδου (login)
   Ashbo. Clean Modal Login Form. [online].
   http://bootsnipp.com/snippets/featured/clean-modal-login-form
9. main.php για την κυρίως σελίδα
10. main.css για τη μορφοποίηση της κυρίως σελίδας
11. schedule.php για τη σελίδα εισαγωγής προγράμματος
    Chris Coyier(2011). Functional CSS Tabs Revisited. [online].
    https://css-tricks.com/functional-css-tabs-revisited/
12. schedule.css για τη μορφοποίηση της σελίδας εισαγωγής προγράμματος
    Chris Coyier(2011). Functional CSS Tabs Revisited. [online].
    https://css-tricks.com/functional-css-tabs-revisited/

13. show_schedule.php βοηθητική συνάρτηση της σελίδας εισαγωγής προγράμματος
14. check_insert.php βοηθητική συνάρτηση της σελίδας εισαγωγής προγράμματος
15. graph24h.php για τη σελίδα στατιστικών ημέρας
    Highcharts. Your First Graph. [online]. http://www.highcharts.com/docs/getting-started/your-first-chart
    Highcharts. Data from a database. [online]. http://www.highcharts.com/docs/working-with-data/data-from-a-database
16. graph24h.css για τη μορφοποίηση της σελίδας στατιστικών ημέρας
17. statistic24h.py για την επιλογή των εσωτερικών θερμοκρασιών από τη βάση για δεδομένα μίας ημέρας
18. out_statistic24.py για την επιλογή των εξωτερικών θερμοκρασιών από τη βάση για δεδομένα μίας ημέρας
19. graph_week.php για τη σελίδα στατιστικών εβδομάδας
20. graph_week.css για τη μορφοποίηση της σελίδας στατιστικών εβδομάδας
21. statistic_week.py για την επιλογή των εσωτερικών θερμοκρασιών από τη βάση για δεδομένα μίας εβδομάδας
22. out_statistic_week.py για την επιλογή των εξωτερικών θερμοκρασιών από τη βάση για δεδομένα μίας εβδομάδας

Για τις προσομοιώσεις των θερμοστατών στα μοντέλα χρησιμοποιήθηκαν τα:

23. Ti_on-off_delay.py για το μοντέλο Ti και τον θερμοστάτη on-off
24. Ti_pid.py για το μοντέλο Ti και τον θερμοστάτη pid
    ivmech. ivPID. [online]. https://github.com/ivmech/ivPID
25. TiTh_on-off_delay.py για το μοντέλο TiTh και τον θερμοστάτη on-off
26. TiTh_pid.py για το μοντέλο TiTh και τον θερμοστάτη pid
27. TiTeTh_on-off_delay.py για το μοντέλο TiTeTh και τον θερμοστάτη on-off
28. TiTeTh_pid.py για το μοντέλο TiTeTh και τον θερμοστάτη pid
29. TiTeThTs_on-off_delay.py για το μοντέλο TiTeThTs και τον θερμοστάτη on-off
30. TiTeThTs_pid.py για το μοντέλο TiTeThTs και τον θερμοστάτη pid

Για τον έξυπνο θερμοστάτη χρησιμοποιήθηκαν τα:

31. smart_table.py για τη δημιουργία της έξυπνης δομής
32. smart_compare.py για τη σύγκρηση του προγράμματος και της έξυπνης δομής
33. smart_update.py για ανανέωση της έξυπνης δομής
34. smart_thermostat.py για την υλοποίηση του on-off έξυπνου θερμοστάτη
35. smart.py είναι το βασικό script για τον on-off έξυπνο θερμοστάτη
36. smart_thermostat_pid_ext.py για την υλοποίηση του pid έξυπνου θερμοστάτη
37. smart_pid_ext.py είναι το βασικό script για τον pid έξυπνο θερμοστάτη

Παράρτημα Γ – Κώδικας

1. database.py

```
#!/usr/bin/python

import rrdtool
ret = rrdtool.create("temperature.rrd", "--start", '0',
"--step", "60",
"DS:a:GAUGE:300:-10:50",
"RRA:AVERAGE:0.5:1:72",
"RRA:AVERAGE:0.5:1:288",
"RRA:AVERAGE:0.5:12:168",
"RRA:AVERAGE:0.5:12:720",
"RRA:AVERAGE:0.5:36:720",
"RRA:AVERAGE:0.5:72:720",
"RRA:AVERAGE:0.5:288:365")
```

3. thermometer.py

```
#!/usr/bin/python
import os
import glob
import time
import rrdtool

os.system('modprobe w1-
gpio') #execute the command(a string) in a subshell
#the command adds kernel module w1-gpio
os.system('modprobe w1-
therm')#return value=exit status of the process

base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0] #path to 1-
wire devices
device_file = device_folder + '/w1_slave'#output file
databaseFile = '/home/pi/Desktop/temperature.rrd'
a = '/sys/bus/w1/devices/28-00000670834d/w1_slave'

def read_temp_raw():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close
    return lines

def read_temp():
    lines = read_temp_raw()
    while lines[0].strip()[-
3:] != 'YES': #check if yes is the last 3 letters of line
        time.sleep(0.2)
        lines = read_temp_raw()
    equals_pos = lines[1].find('t=') #read the temperature
    if equals_pos != -1: #if there is no error
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        return temp_c

```

```
def read_all():
    update = 'N:'
    temp = read_temp()
    update += '%f:' % temp
    update = update[:-1]
    rrdtool.update(databaseFile, update)

read_all()
```

4.outside_database.py

```
#!/usr/bin/python

import rrdtool
ret = rrdtool.create("out_temperature.rrd", "--start", '0',
"--step", "60",
"DS:b:GAUGE:300:-10:50",
"RRA:AVERAGE:0.5:1:72",
"RRA:AVERAGE:0.5:1:288",
"RRA:AVERAGE:0.5:12:168",
"RRA:AVERAGE:0.5:12:720",
"RRA:AVERAGE:0.5:36:720",
"RRA:AVERAGE:0.5:72:720",
"RRA:AVERAGE:0.5:288:365")
```

6.out_temp.py

```
#!/usr/bin/python
import sys
import rrdtool

celcius = sys.argv[1]
temp = float(celcius)
databaseFile = '/home/pi/Desktop/out_temperature.rrd'

update = 'N:'
update += '%f:' % temp
update = update[:-1]
rrdtool.update(databaseFile, update)
```

7.index.php

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1">
        <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/boo
tstrap/3.3.6/css/bootstrap.min.css">
        <link rel="stylesheet" type="text/css" href="index.css">
        <title>Raspberry Project</title>
    </head>
```

```
10.
11.     <body>
12.      <?php
13.     $first_time = 1;
14.     $user = $pass = "";
15.      // Get Values
16.      if ($_SERVER["REQUEST_METHOD"] == "POST") {
17.          if (emptyempty($_POST["user"])){
18.                 $userErr = "Username Required";
19.             $passErr = "Password Required";
20.         }else {       // if is ok with user, check pass
21.                 $user = test_input($_POST["user"]);
22.                 if (emptyempty($_POST["pass"])){
23.                     $passErr = "Password Required";
24.                     }else { //ok with pass too
25.                         $pass = test_input($_POST["pass"]);
26.                 $first_time = 0;
27.                 }
28.         }
29.     }
30.     function test_input($data) {
31.             $data = trim($data);
32.            $data = stripslashes($data);
33.            $data = htmlspecialchars($data);
34.            return $data;
35.            }
36.
37.    $servername = "localhost";
38.    $database = "project";
39.
40.    if (!$first_time){  //check only after submit button is pressed and
   user and pass not null
41.         // Create connection
42.         $conn = mysqli_connect ($servername, $user, $pass, $database);

43.         // Check connection
44.            if (mysqli_connect_errno($conn)) {
45.             echo "<script>alert('Wrong username or password')</script>"
   ;
46.            }else{//its ok so load main page
47.             mysqli_close($conn);
48.             header("Location:project/main.php");
49.         }
50.     }
51.   ?>
52.
53.
54.   <div id="background-carousel">
55.     <div id="myCarousel" class="carousel slide" data-ride="carousel">
56.             <div class="carousel-inner">
57.             <div class="item active" style="background-
   image:url(https://i.ytimg.com/vi/wNYgi2scaGU/maxresdefault.jpg)">
          </div>
58.         </div>
59.         </div>
60. </div>
61.
62.
63. <section id="login">
```

```
64.     <div class="container">
65.         <div class="row">
66.                 <div class="col-xs-12">
67.                 <div class="form-wrap">
68.                     <form class="text-center" id="login-
    form" method="POST" action="<?php echo htmlspecialchars($_SERVER["PHP_S
    ELF"]); ?>" role="form" autocomplete="off">
69.                         <h1>Login</h1>
70.                         <div class="form-group">
71.                             <label for="user" class="sr-
    only"></label>
72.                             <input type="text" name="user" class="form-
    control" id="user" placeholder="Username">
73.                             <span class="error"> <?php echo $userErr;?>
    </span></p>
74.                         </div>
75.                         <div class="form-group">
76.                                                         <label for="pass" c
    lass=="sr-only"></label>
77.                                                         <input type="passwo
    rd" class="form-
    control" id="pass" name="pass" placeholder="Password">
78.                             <span class="error"> <?php echo $passErr;?>
    </span></p>
79.                                                     </div>
80.                         <input type="submit" id="btn-
    login" class="btn btn-custom btn-lg btn-block" value="Log in">
81.                     </form>
82.                 </div>
83.             </div></div></div>
84.         </section>
85.     </body>
86.
87. </html>
```

8. index.css

```
1.  #background-carousel{
2.      position:fixed;
3.      width:100%;
4.      height:100%;
5.      z-index:-1;
6.  }
7.  .carousel,
8.  .carousel-inner {
9.      width:100%;
10.     height:100%;
11.     z-index:0;
12.     overflow:hidden;
13. }
14. .item {
15.     width:100%;
16.     height:100%;
17.     background-position:center center;
18.     background-size:cover;
19.     z-index:0;
20. }
```

```
21.
22. #login {
23.     padding-top: 50px;
24. }
25. #login .form-wrap {
26.     width: 60%;
27.     margin: 0 auto;
28.     border: 1px solid #ccc;
29.     border-radius: 4px;
30.     background-color: #ffffff;
31.     padding-left: 50px;
32.     padding-right: 50px;
33.     padding-bottom: 50px;
34.
35. }
36. #login h1 {
37.     color: #3366ff;
38.     font-size: 36px;
39.     text-align: center;
40.     font-weight: bold;
41.     padding-bottom: 20px;
42. }
43. #login .btn.btn-custom {
44.     margin-top: 50px;
45.     font-size: 14px;
46.     margin-bottom: 20px;
47. }
48. .form-control {
49.     color: #212121;
50. }
51. .btn-custom {
52.     color: #fff;
53.     background-color: #3366FF;
54. }
55. .btn-custom:hover,
56. .btn-custom:focus {
57.     color: #fff;
58. }
```

9.main.php

```
1.  <!DOCTYPE html>
2.  <html>
3.      <head>
4.          <meta charset="UTF-8">
5.          <title>Raspberry Project</title>
6.          <link rel="stylesheet" type="text/css" href="main.css">
7.          <meta name="viewport" content="width=device-width, initial-
    scale=1">
8.                  <link rel="stylesheet" href="http://maxcdn.bootstrapcdn
    .com/bootstrap/3.3.6/css/bootstrap.min.css">
9.
10.     <?php
11.         //insert details for temperature
12.         $jsonurl = "http://api.openweathermap.org/data/2.5/weather?id=2
    64371&APPID=1f2bfcca78954c16ca3b945f0c23c776";
13.             $json = file_get_contents($jsonurl);
```

```
14.             $weather = json_decode($json);
15.             $kelvin = $weather->main->temp;
16.             $celcius = $kelvin - 273.15;
17.
18.         //send the temperature to the python file to be added to the rr
    d database
19.         $send = shell_exec("python /home/pi/Desktop/out_temp.py $celciu
    s");
20.         $image = $weather->weather[0]->icon;
21.         $condition = $weather->weather[0]->description;
22.         $humidity = $weather->main->humidity;
23.         $wind = $weather->wind->speed;
24.         $sunrise = $weather->sys->sunrise;
25.         $sunrise = date("H:i:s",$sunrise);
26.         $sunset = $weather->sys->sunset;
27.         $sunset = date("H:i:s",$sunset);
28.         $today = date("l, d M Y");
29.         $time = date("H:i:s");
30.
31.         //forecast for next hours
32.         $jsonurl = "http://api.openweathermap.org/data/2.5/forecast?id=
    264371&APPID=1f2bfcca78954c16ca3b945f0c23c776";
33.         $json = file_get_contents($jsonurl);
34.         $fweather = json_decode($json);
35.         for ($i = 0; $i < 10; $i++){
36.             $ftemp1 = ($fweather->list[$i]->main->temp)-273.15;
37.             $ftime = ($fweather->list[$i]->dt)*1000;
38.             $ftemp[] = "[$ftime,$ftemp1]";
39.         }
40.
41.         //forecast for next 5 days
42.         $jsonurl = "http://api.openweathermap.org/data/2.5/forecast/dai
    ly?id=264371&APPID=1f2bfcca78954c16ca3b945f0c23c776";
43.         $json = file_get_contents($jsonurl);
44.         $fweather = json_decode($json);
45.         for ($i = 0; $i < 5; $i++){
46.         $temp_low[] = ($fweather->list[$i]->temp->min)-273.15;
47.           $temp_high[] = ($fweather->list[$i]->temp->max)-273.15;
48.           $descr[] = $fweather->list[$i]->weather[0]->description;
49.            $fimage[] = $fweather->list[$i]->weather[0]->icon;
50.            $fdate[] = $fweather->list[$i]->dt;
51.         }
52.           //get the inside temperature now
53.         $output = popen('python ./statistic24h.py', 'r');
54.         $list = fgets($output);
55.         $start = fgets($output);
56.         fclose($output);
57.         $data2 = explode(', ', $list); //making the list an array to be
     processed
58.         foreach ($data2 as $value) {
59.           if (trim($value) != "NAN" ) {
60.             $temp_in[]=$value; //the temperature now  = last
61.                 }else{
62.                         $temp_in='null';
63.                 }
64.         }$temp_now = $temp_in[119];//the last element is the 119th
65.         ?>
66.     <script type="text/javascript" src="jquery-
    1.12.0.min.js"></script>
```

```
67.         <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.
    0/jquery.min.js"></script>
68.     <script type="text/javascript">
69.         $(function() {
70.                 var chart = new Highcharts.Chart({
71.                         chart: {
72.                                 renderTo: 'container2',
73.                                 type: 'line',
74.                 backgroundColor: null
75.                         },
76.                         title: {
77.                                 text: 'Forecast: Next Hours'
78.                         },
79.                         xAxis: {
80.                                 type: 'datetime',
81.                         },
82.                         yAxis: {
83.                                 title: {
84.                                         text: 'Celsius'
85.                                 }
86.                         },
87.                         series: [{
88.                                 showInLegend: false,
89.                                 name: 'Temperature',
90.                                    color: '#800000',
91.                              data: [<?php echo join($ftemp, ',') ?>]
92.                         }]
93.                     });
94.         });
95.     </script>
96.
97.         </head>
98.     <body>
99.         <script src="../js/highcharts.js"></script>
100.            <script src="../js/modules/exporting.js"></script>
101.
102.                <nav class="navbar navbar-default" id="menu">
103.                        <div class="container-fluid">
104.                <div class="navbar-header">
105.                    <a class="navbar-brand">Smart Thermostat</a>
106.                </div>
107.                <ul class="nav navbar-nav">
108.                    <li class="active"><a href="main.php">Main Page</a>
    </li>
109.                            <li><a href="schedule.php">Schedule</a><li>

110.                                <li><a href="graph24h.php">Statistics</
    a></li>
111.                        </ul>
112.               </div>
113.            </nav>
114.
115.            <div class="container">
116.                <div class="row">
117.                            <div class="col-md-5">
118.                    <div class="table-responsive">
119.                        <table class="date">
120.                                    <tr><td><strong> <?php echo $to
    day ?></strong></td> </tr>
```

```
                                        <tr><td><strong><?php echo $tim
    e ?></strong></td> </tr>
                                        <tr></tr>
                                    </table>
                    </div>

                    <div class="table-responsive">
                        <table class="forec">
                                        <tr><td><img src="https://maxcd
    n.icons8.com/Color/PNG/24/Weather/thermometer-
    24.png" width="24"></td>
                                        <td><strong>Inside Temperature:
     <?php echo $temp_now ?>°C </strong></td><tr>
                                    </table>
                    </div>
                    <div class="table-responsive">
                        <table class="forec2">
                                        <tr><td><img src="http://openwe
    athermap.org/img/w/<?php echo $image?>.png"></td>
                                        <td><strong>Outside Temperature
    : <?php echo $celcius ?>°C </strong></td></tr>
                                    </table>
                    </div>
                    <div class="table-responsive">
                    <table class="info">
                            <tr><td> Description: </td>
                            <td> <?php echo "$condition"?></td></tr>
                            <tr><td> Humidity: </td>
                            <td> <?php echo "$humidity%"?> </td></tr>
                            <tr><td> Wind:  </td>
                            <td> <?php echo "$wind m/s" ?> </td><tr>
                            <tr><td> Sunrise: </td>
                            <td> <?php echo "$sunrise" ?> </td></tr>
                            <tr><td> Sunset: </td>
                            <td> <?php echo "$sunset" ?> </td></tr>
                     </table></div>
                </div>

                <div class="col-md-7">
                    <div id="container2"></div>
                    <div id="forecast">
                    <div class="table-responsive">
                        <table class="none">
                        <?php echo "<tr>";
                        for ($j = 0; $j<5; $j++){
                            echo "<td><b>" . date('l',$fdate[$j]). "</b
    ></td>";
                        }
                        echo "</tr><tr>";
                        for ($j = 0; $j<5; $j++){
                            echo "<td><img src='http://openweathermap.o
    rg/img/w/$fimage[$j].png'></td>";
                                        }
                            echo "</tr><tr>";
                        for ($j = 0; $j<5; $j++){
                            echo "<td>" . $descr[$j]. "</td>";
                             }
                       echo "</tr><tr>";
                        for ($j = 0; $j<5; $j++){
```

```
                        echo "<td>low: " . $temp_low[$j]. "°C</td>";

                        }
                    echo "</tr><tr>";
                    for ($j = 0; $j<5; $j++){
                      echo "<td>high: " . $temp_high[$j]. "°C</td>";
                    }
                    echo "</tr>";
                     ?>
                     </table>
                </div>
                </div>
            </div>
            </div></div>
        </body>

    </html>
```

10.main.css

```
body {
    background:url('https://projectmgtcoach.com/wp-content/uploads/2015/03/light-blue-backgrounds-for-websites.jpg');
   width:100%;
   height:100%;
   background-position:center center;
   background-size:cover;
   z-index:0;
}

#forecast {
    width:100%;
    float:left;
    padding:10px;
}
#menu{
    width:100%;
    font-size: 25px;
}
table.none{
    width:100%
}
table.none td{
    text-align: center;
    padding-right:5px;
}
table.forec{
    margin: auto;
    width:70%;
    margin-top:50px;
}
table.forec2{
    margin:auto;
    width:70%;
}
table.forec,table.forec2 td {
    text-align: center;
```

```
    padding-right:5px;
}
table.info {
    width:80%;
    margin:auto;
    border: 1px solid black;
    margin-top:50px;
}
table.info td{
    text-align: center;
    border: 1px solid black;
    padding-top:5px;
    padding-bottom:5px;
}
table.date {
    margin:auto;
    width:80%;
    border: 3px;
    border-style:solid;
    border-color:#800000;
    margin-top:80px;
}
table.date td{
    text-align: center;
}
```

11.schedule.php

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Raspberry Project</title>
<link rel="stylesheet" type="text/css" href="schedule.css">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.2/jquery.min.js"></script>
<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>

</head>
<body>
    <?php
        // get the values from the form
        $from=$to=$temp=$days="";
        $first_time=1;

        if ($_SERVER["REQUEST_METHOD"] == "POST") {
            if (empty($_POST["from"])){
                    $dataErr = "Please fill all the data";
            }else {
                  $from = test_input($_POST["from"]);
                  if (empty($_POST["to"])){
                       $dataErr = "Please fill all the data";
                  }else {
```

```
27.                         $to = test_input($_POST["to"]);
28.                         if (empty($_POST["temp"])){
29.                             $dataErr= "Please fill all the data";
30.                         } else{
31.                             $temp = test_input($_POST["temp"]);
32.                             $first_time = 0;
33.                         }
34.                     }
35.              }
36.         }
37.
38.         function test_input($data) {
39.                     $data = trim($data);
40.                     $data = stripslashes($data);
41.                     $data = htmlspecialchars($data);
42.                     return $data;
43.                 }
44.
45.         if (!$first_time){
46.             $vari=$_POST["day"];//day of the week
47.             $conn = mysqli_connect("localhost", "project", "1111", "pro
    ject");
48.             if (isset($_POST["insert"])) { //if insert is pressed
49.                 include("check_insert.php"); $validation = validate("$v
    ari","$from","$to");//check for overlaps
50.                 if ($validation){//no overlaps
51.                 $insert = "INSERT INTO `$vari`(`From`, `To`, `Temperatu
    re`) VALUES ('$from', '$to', '$temp')";
52.                 if (!mysqli_query($conn, $insert)){ //error
53.                    echo "<script>alert('Recond can not be inserted')</s
    cript>";
54.                 }
55.             }else{//overlap error
56.                 echo "<script>alert('Overlap in time interval')</script
    >";
57.             }
58.         }else{ //delete is pressed
59.             $delete = "DELETE FROM `$vari` WHERE `From`='$from' AND `To
    `='$to' AND `Temperature`='$temp'";
60.             if (!mysqli_query($conn,$delete)){//error
61.                 echo "<script>alert('Record can not be deleted')</scrip
    t>";
62.              }
63.        }
64.        mysqli_close($conn);
65.      }
66.     ?>
67.         <nav class="navbar navbar-default" id="menu">
68.             <div class="container-fluid">
69.                 <div class="navbar-header">
70.                      <a class="navbar-brand">Smart Thermostat</a>
71.                  </div>
72.                 <ul class="nav navbar-nav">
73.                 <li><a href="main.php">Main Page</a></li>
74.                 <li class="active"><a href="schedule.php">Schedule</a><
    /li>
75.                 <li class="dropdown">
76.                 <a class="dropdown-toggle" data-
    toggle="dropdown" >Statistics
```

```
77.                 <span class="caret"></span></a>
78.                 <ul class="dropdown-menu">
79.                 <li><a href="graph24h.php">Day</a></li>
80.                 <li><a href="graph_week.php">Week</a></li>
81.                 </ul>
82.                 </li>
83.                 </ul>
84.             </div>
85.          </nav>
86.
87.
88.         <div id="section">
89.         <div class="container">
90.                 <div class="row">
91.                 <div class="col-md-12 col-xs-12">
92.            <!-- make the tab menue for every day -->
93.            <!--
    php code so we stay at the same day tab after an insert/delete -->
94.             <div class="radio">
95.             <input type="radio" name="tabs" id="monday"
96.             <?php if ($_SERVER["REQUEST_METHOD"] == "POST"){$vari=$_POS
   T["day"];
97.                 if($vari=="Monday"){echo "checked"; }} ?> >
98.             <label for="monday">
99.                 <span> Monday </span>
100.                </label>
101.
102.                <input type="radio" name="tabs" id="tuesday"
103.                <?php if ($_SERVER["REQUEST_METHOD"] == "POST"){$vari=$
   _POST["day"];
104.                    if($vari=="Tuesday"){echo "checked"; }} ?> >
                    <label for="tuesday">
105.                    <span> Tuesday </span>
106.                    </label>
107.                <input type="radio" name="tabs" id="wednesday"
108.                <?php if ($_SERVER["REQUEST_METHOD"] == "POST"){$vari=$
   _POST["day"];
109.                    if($vari=="Wednesday"){echo "checked"; }} ?> >
110.                    <label for="wednesday">
111.                    <span> Wednesday </span>
112.                    </label>
113.                <input type="radio" name="tabs" id="thursday"
114.                <?php if ($_SERVER["REQUEST_METHOD"] == "POST"){$vari=$
   _POST["day"];
115.                    if($vari=="Thursday"){echo "checked"; }} ?> >
116.                    <label for="thursday">
117.                    <span> Thursday </span>
118.                    </label>
119.                <input type="radio" name="tabs" id="friday"
120.                <?php if ($_SERVER["REQUEST_METHOD"] == "POST"){$vari=$
   _POST["day"];
121.                    if($vari=="Friday"){echo "checked"; }} ?> >
122.                    <label for="friday">
123.                    <span> Friday </span>
124.                    </label>
125.                <input type="radio" name="tabs" id="saturday"
126.                <?php if ($_SERVER["REQUEST_METHOD"] == "POST"){$vari=$
   _POST["day"];
127.                    if($vari=="Saturday"){echo "checked"; }} ?> >
```

```
128.                     <label for="saturday">
129.                     <span> Saturday </span>
130.                     </label>
131.                 <input type="radio" name="tabs" id="sunday"
132.                 <?php if ($_SERVER["REQUEST_METHOD"] == "POST"){$vari=$
    _POST["day"];
133.                     if($vari=="Sunday"){echo "checked"; }} ?> >
134.                     <label for="sunday">
135.                     <span> Sunday </span>
136.                     </label>
137.
138.                 <!--
     show the data in the table for the appropriate day -->
139.                 <div id="mon-content" class="tab-content">
140.                     <h3>Monday Schedule</h3>
141.                     <?php
142.                     include("show_schedule.php"); show("Monday"); ?>
143.                     <form action="<?php echo htmlspecialchars($_SERVER[
    "PHP_SELF"]); ?>" method="POST" >
144.                         <br>
145.                          <fieldset>
146.                          <legend> Data: </legend>
147.                          <div class="width">
148.                          Insert time in format: hh:mm:ss<br>
149.                          From:<input type="time" name="from" id="as1"><b
    r>
150.                          To:<input type="time" name="to" id="as2"><br>
151.                          Temperature(15-
    30 Celsius):<input type="number" name="temp" min="15" max="30" id="as3"
    >
152.                          </br></br>
153.                          <input type="hidden" value="Monday" name="day"/
    >
154.                          <input type="submit" value="Insert" name="inser
    t"/>
155.                          <input type="submit" value="Delete" name="delet
    e"/>
156.                          </div>
157.                     </fieldset>
158.                 </div>
159.                 <div id="tue-content" class="tab-content">
160.                         <h3>Tuesday Schedule</h3>
161.                         <?php show("Tuesday"); ?>
162.                     <form action="<?php echo htmlspecialchars($_SERVER[
    "PHP_SELF"]); ?>" method="POST" class="form" >
163.                          <br>
164.                          <fieldset>
165.                          <legend> Data: </legend>
166.                          <div class="width">
167.                          Insert time in format: hh:mm:ss<br>
168.                          From:<input type="time" name="from" class="inpu
    t" id="as1"><br>
169.                          To:<input type="time" name="to" class="input" i
    d="as2"><br>
170.                          Temperature(15-
    30 Celsius):<input type="number" name="temp" min="15" max="30" id="as3"
    >
171.                          <br><br>
```

```
                        <input type="hidden" value="Tuesday" name="day"
/>
                         <input type="submit" value="Insert" name="inse
rt"/>
                        <input type="submit" value="Delete" name="delet
e"/>
                        </div>
                        </fieldset>
                      </form>
                        </div>
                <div id="wed-content" class="tab-content">
                        <h3>Wednesday Schedule</h3>
                        <?php show("Wednesday");?>
                    <form action="<?php echo htmlspecialchars($_SERVER[
"PHP_SELF"]); ?>" method="POST" >
                        <br>
                        <fieldset>
                        <legend> Data: </legend>
                        <div class="width">
                        Insert time in format: hh:mm:ss<br>
                        From: <input type="time" name="from" id="as1"><
br>
                        To: <input type="time" name="to" id="as2"><br>

                        Temperature(15-
30 Celsius): <input type="number" name="temp" min="15" max="30" id="as3
">
                        <br><br>
                        <input type="hidden" value="Wednesday" name="da
y"/>
                        <input type="submit" value="Insert" name="inser
t"/>
                        <input type="submit" value="Delete" name="delet
e"/>
                        </div>
                        </fieldset>
                    </form>
                </div>
                <div id="thu-content" class="tab-content">
                    <h3>Thursday Schedule</h3>
                    <?php show("Thursday"); ?>
                    <form action="<?php echo htmlspecialchars($_SERVER[
"PHP_SELF"]); ?>" method="POST" >
                        <br>
                        <fieldset>
                        <legend> Data: </legend>
                        <div class="width">
                        Insert time in format: hh:mm:ss<br>
                        From: <input type="time" name="from" id="as1"><b
r>
                        To: <input type="time" name="to" id="as2"><br>
                        Temperature(15-
30 Celsius): <input type="number" name="temp" min="15" max="30" id="as3
">
                        <br><br>
                        <input type="hidden" value="Thursday" name="day"
/>
                        <input type="submit" value="Insert" name="insert
"/>
```

```
                    <input type="submit" value="Delete" name="delete
"/>
                    </div>
                    </fieldset>
                </form>
            </div>
            <div id="fri-content" class="tab-content">
                <h3>Friday Schedule</h3>
                <?php show("Friday"); ?>
                <form action="<?php echo htmlspecialchars($_SERVER[
"PHP_SELF"]); ?>" method="POST" >
                    <br>
                    <fieldset>
                    <legend> Data: </legend>
                    <div class="width">
                    Insert time in format: hh:mm:ss<br>
                    From: <input type="time" name="from" id="as1"><b
r>
                    To: <input type="time" name="to" id="as2"><br>
                    Temperature(15-
30 Celsius): <input type="number" name="temp" min="15" max="30" id="as3
">
                    <br><br>
                    <input type="hidden" value="Friday" name="day"/>

                    <input type="submit" value="Insert" name="insert
"/>
                    <input type="submit" value="Delete" name="delete
"/>
                    </div>
                    </fieldset>
                </form>
            </div>
            <div id="sat-content" class="tab-content">
                <h3>Saturday Schedule</h3>
                <?php show("Saturday"); ?>
                <form action="<?php echo htmlspecialchars($_SERVER[
"PHP_SELF"]); ?>" method="POST" >
                    <br>
                    <fieldset>
                    <legend> Data: </legend>
                    <div class="width">
                    Insert time in format: hh:mm:ss<br>
                    From: <input type="time" name="from" id="as1"><br
>
                    To: <input type="time" name="to" id="as2"><br>
                    Temperature(15-
30 Celsius): <input type="number" name="temp" min="15" max="30" id="as3
">
                    <br><br>
                    <input type="hidden" value="Saturday" name="day"/
>
                    <input type="submit" value="Insert" name="insert"
/>
                    <input type="submit" value="Delete" name="delete"
/>
                    </div>
                    </fieldset>
                </form>
```

```
                </div>
                <div id="sun-content" class="tab-content">
                   <h3>Sunday Schedule</h3>
                   <?php show("Sunday"); ?>
                   <form action="<?php echo htmlspecialchars($_SERVER[
   "PHP_SELF"]); ?>" method="POST" >
                      <br>
                      <fieldset>
                      <legend> Data: </legend>
                      <div class="width">
                      Insert time in format: hh:mm:ss<br>
                      From: <input type="time" name="from" id="as1"><b
   r>
                      To: <input type="time" name="to" id="as2"><br>
                      Temperature(15-
   30 Celsius): <input type="number" name="temp" min="15" max="30" id="as3
   ">
                      <br><br>
                      <input type="hidden" value="Sunday" name="day"/>

                      <input type="submit" value="Insert" name="insert
   "/>
                      <input type="submit" value="Delete" name="delete
   "/>
                      </div>
                      </fieldset>
                   </form>
                </div>
             </div>
          </div></div></div>

          </div>

   </body>
   </html>
```

12.schedule.css

```
body {
    background:url('https://projectmgtcoach.com/wp-
   content/uploads/2015/03/light-blue-backgrounds-for-websites.jpg');
    width:100%;
    height:100%;
    background-position:center center;
    background-size:cover;
    z-index:0;
}
#menu{
    width:100%;
    font-size: 25px;
}
fieldset {
    width: 100%;
}
#as1{
    position:relative !important;
    left:15px !important;
```

```
}
#as2{
    position:relative !important;
    left:32px !important;
}
#as3{
    width:50px !important;
}
.width {
    position: center;
}
table {
    width:100%;
    margin-top:10px;
}
table th,td {
    border: 1px solid black !important;
    border-collapse: collapse;
    text-align: center;
}
.radio{  /* for all the area */
    max-width: 100%;
    float: none;
    list-style: none;
    padding: 0;
}
.radio:after{
    content: ' ';
    display:table;
    clear: both;
}

.radio input[type=radio] { /* dont display radio buttons */
    display: none;
}
.radio label {   /* format of each button */
    display: block;
    float: left;
    width: 14.285%;
    color: #0088FF;
    font-size: 15px;
    text-align: center;
    line-height:3;
    cursor:pointer;
    box-shadow: inset 0 4px #0088FF;
    border-bottom: 4px solid #0088FF;
    transition: all 0.5s;
}
.radio label:hover{ /* format when on cursor */
    color:#b30000;
    background-color: #0088FF;
}
.tab-content{  /* format of the context inside area */
    display: none;
    width: 100%;
    float: left;
    text-align:center;
    padding:15px;
    background-color: #FFFFFF;
```

```
    margin-bottom:10px;
}
.radio [id^="tab"]:checked + label{   /* display the chlicked one */
    background: #FFFFFF;
}
#monday:checked ~ #mon-content,
#tuesday:checked ~ #tue-content,
#wednesday:checked ~ #wed-content,
#thursday:checked ~ #thu-content,
#friday:checked ~ #fri-content,
#saturday:checked ~ #sat-content,
#sunday:checked ~ #sun-content {
    display: block;
}
fieldset{
    text-align: justify;
}

@media (max-width:978px){
.radio{  /* for all the area */
        max-width: 100%;
        float: none;
        list-style: none;
        padding: 0;
    }
    .radio:after{
        content:'';
        display:table;
        clear: both;
    }

    .radio label {   /* format of each button */
        display: block;
        float: left;
        width: 14.285%;
        color: #0088FF;
        font-size: 8px;
        line-height:3;
        text-indent:-15px;
        clear:left
        box-shadow: inset 0 4px #0088FF;
        border-bottom: 4px solid #0088FF;
        transition: all 0.5s;
    }
    .tab-content{  /* format of the context inside area */
        display: none;
        width: 100%;
        float: left;
        text-align:center;
        padding:5px;
        background-color: #FFFFFF;
    }
    }
```

13.show_schedule.php

```
<?php
```

```
  //print the current data of the table
  function show($day){
$conn = mysqli_connect("localhost", "project", "1111", "project");
    $data = "SELECT * FROM `$day` ORDER BY `From`";
$result = mysqli_query($conn,$data);
    if (mysqli_num_rows($result) > 0) {
        echo "<table id=\"table\">"; //show results into table"
            echo "<tr><th>" . From. "</th><th>" . To. "</th><th> " . Te
mperature. "</th></tr>";
            while ($row = mysqli_fetch_array($result)){
                echo "<tr><td>" . $row["From"]. "</td><td>" . $row["To"
]. "</td><td> " . $row["Temperature"]. "</td></tr>";
            }
            echo "</table>";
    }else{
    echo "No schedule";
    }
    mysqli_close($conn);
return;
 }

>
```

14.check_insert.php

```
<?php
   //first check is the data is valid (no overlaps)
   function validate($day, $from, $to){
        $status = 0;//default=overlap
    $conn = mysqli_connect("localhost", "project", "1111", "project");

        $data = "SELECT * FROM `$day` ORDER BY `From`";
        $result = mysqli_query($conn, $data);
        if (!$result){ //error
                echo "<script>alert('Data cannot be selected')</script>
";
        }else{ //we got ok the data so we check
                if (mysqli_num_rows($result) > 0) { //if we get a non-
empty table(data)
                        while ($row = mysqli_fetch_array($result)){ //c
heck in every line for no overlaps
                //no onerlap ONLY new_to<from or new_from>to
                if (($from >= $row["To"]) || ($to <= $row["From"])){
                    $status = 1;
                }else{ //overlap found so exit
                    $status = 0;
                    break;
                }
                        }
                }else{$status = 1;}//ok
        }
    mysqli_close($conn);
    return $status;
   }
?>
```

15.graph24h.php

```
1.  <!DOCTYPE html>
2.  <html>
3.  <head>
4.  <meta charset="UTF-8">
5.  <title>Raspberry Project</title>
6.  <link rel="stylesheet" type="text/css" href="graph24h.css">
7.  <meta name="viewport" content="width=device-width, initial-scale=1">
8.  <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3
    .3.6/css/bootstrap.min.css">
9.
10. <?php
11.     //get the data from the rrd database (temperature)
12.     $output = popen('python ./statistic24h.py', 'r');
13.     $list = fgets($output);
14.     $start = fgets($output);
15.     fclose($output);
16.     $data = explode(', ', $list); //making the list an array to be proc
    essed
17.     $start+=720; //resolution = 720
18.     //get the specified field we need (last 24h time-
    templerature) from the database
19.     foreach ($data as $value) {
20.     //for earch value in the table data do
21.         if (trim($value) != "None" ) {
22.             $mydata[]=$value;
23.         }else{
24.             $mydata[]='null';
25.         }
26.     }
27.     $final = implode(" , ", $mydata); //implode a comma between all val
    ues
28.
29.
30.     //get the schedule from the sql server
31.     $today = date("l");
32.     $time = date("H:i:s"); //get schedule from today midnight to time(s
    econd part)
33.     $yesterday = date("l",strtotime("-1 days"));
34.     $yest_time = date("H:i:s",strtotime("-
    24 hours")); //get schedule from yest_time to midnight(first part)
35.     //first part - starting the plot series
36.     $conn = mysqli_connect("localhost", "project", "1111", "project");

37.     $data1 = "SELECT * FROM `$yesterday` WHERE `To`>'$yest_time' ORDER
    BY `From`";
38.     $result = mysqli_query($conn,$data1);
39.     if (mysqli_num_rows($result) > 0) {
40.         $numrows = 0;
41.         while ($row = mysqli_fetch_array($result)){
42.             $numrows = $numrows + 1; //check if we are at the first row
      // +2hours for Greek timezone and -1day(schedule from previus day)
43.             $temporary1 = (strtotime($row[0])+10800-86400)*1000;
44.             //the first time the graph must start from $yest_time
```

```
45.             if ($numrows == 1){
46.                 $i = (strtotime($yest_time)+10800-86400)*1000;
47.                 while ($i < $temporary1){ //padding until first schedul
    e time
48.                     $schedule1[] = "[$i,10]"; //default temperature is
    10
49.                     $i = $i + 60000 ;
50.                                 }
51.             }else{ //all other times must check if the points are conti
    nuous between rows
52.                 $i = $temporary1;
53.                 $l = $temporary2;
54.                 while ($l < $temporary1){ // while old end now start
55.                     $schedule1[] = "[$l,10]"; //default temperature is
    10
56.                     $l = $l + 60000 ;
57.                 }
58.             }
59.             $temporary2 = (strtotime($row[1])+10800-
    86400)*1000; // *1000 - highchart works with milisecs
60.             //must made series to plot (multiple points)
61.             while ($i<=$temporary2) {
62.                 $schedule1[] = "[$i,$row[2]]";
63.                 $i = $i + 60000; //60000msec = 1min
64.             }
65.                 }
66.         }else{ //no schedule - padding with default
67.             $i = (strtotime($yest_time)+10800-86400)*1000;
68.         $temporary2 = (strtotime($yest_time)+10800-
    72000)*1000; //random yesterday time
69.         while ($i < $temporary2){
70.             $schedule1[] = "[$i,10]";
71.             $i = $i + 60000;
72.         }
73.         }
74.         mysqli_close($conn);
75.     //second part - graph until now
76.     $conn = mysqli_connect("localhost", "project", "1111", "project");

77.         $data1 = "SELECT * FROM `$today` WHERE `From`<'$time' ORDER BY
    `From`";
78.         $result = mysqli_query($conn,$data1);
79.         if (mysqli_num_rows($result) > 0) {
80.         $totrows = mysqli_num_rows($result);
81.         $numrows = 0;
82.                 while ($row = mysqli_fetch_array($result)){
83.                         $numrows = $numrows + 1; $k = 1;
84.             // +2hours for Greek timezone
85.                         $temporary3 = (strtotime($row[0])+10800)*1000;

86.                         if ($numrows == 1){ //first time check if it is
     continuous with previous series
87.                 while ($temporary2 < $temporary3){
88.                     $schedule1[] = "[$temporary2,10]"; //default temper
    ature is 10
89.                     $temporary2 = $temporary2 + 60000 ;
90.                 }
91.             }else{
92.                  while ($temporary4 < $temporary3){
```

```
93.                     $schedule1[] = "[$temporary4,10]"; //default tempe
   rature is 10
94.                     $temporary4 = $temporary4 + 60000 ;
95.                                 }
96.             }
97.             $temporary4 = (strtotime($row[1])+10800)*1000; // *1000 -
    highchart works with milisecs
98.                     //must make series to plot (multiple points)
99.             if ($numrows == $totrows){ //the last time the graph must e
   nd at $time if on schedule
100.                    $i = (strtotime($time)+10800)*1000;
101.                    if ($i < $temporary4){
102.                        $temporary4 = (strtotime($time)+7200)*1000;
103.                    }else {
104.                        $k = 0;
105.                    }
106.                            }
107.            $i = $temporary3;
108.            while ($i<=$temporary4) {
109.                  $schedule1[] = "[$i,$row[2]]";
110.                  $i = $i + 60000; //60000msec = 1min
111.                           }
112.            if ($k == 0){
113.                  $i = (strtotime($time)+10800)*1000;
114.                  while ($temporary4 < $i){
115.                      $schedule1[] = "[$temporary4,10]";
116.                      $temporary4 = $temporary4 + 60000; //60000msec =
    1min
117.                  }
118.              }
119.            }
120.            }else{  //no schedule - padding with default
121.                    $i = (strtotime($time)+10800)*1000;
122.                    while ($i > $temporary2){
123.                            $schedule1[] = "[$temporary2,10]";
124.                            $temporary2 = $temporary2 + 60000;
125.                    }
126.
127.            }
128.            mysqli_close($conn);
129.
130.
131.        //get the outside temperature from rrd daabase
132.            $output = popen('python ./out_statistic24.py', 'r');
133.            $list = fgets($output);
134.            $start2 = fgets($output);
135.            fclose($output);
136.            $data2 = explode(', ', $list); //making the list an array t
   o be processed
137.            $start2+=720; //resolution = 720
138.            //get the specified field we need (last 24h time-
   templerature) from the database
139.            foreach ($data2 as $value) {
140.            //for earch value in the table data do
141.                    if (trim($value) != "None" ) {
142.                            $mydata2[]=$value;
143.                    }else{
144.                            $mydata2[]='null';
145.                    }
```

```
146.            }
147.            $final2 = implode(" , ", $mydata2); //implode a comma betwe
    en all values
148.
149.
150.    ?>
151.            <script type="text/javascript" src="https://ajax.googleapis
    .com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
152.    <script src="https://code.highcharts.com/highcharts.js"></script>
153.    <script src="https://code.highcharts.com/modules/exporting.js"></sc
    ript>
154.
155.    <script type="text/javascript">
156.        $(document).ready(function() {
157.            var chart = new Highcharts.Chart({
158.                chart: {
159.                    renderTo: 'container2',
160.                    zoomType: 'x',
161.                    backgroundColor: null
162.                },
163.                title: {
164.                    text: 'Temperature - Last 24 hours'
165.                },
166.                xAxis: {
167.                    type: 'datetime',
168.                    maxZoom: 12* 600 * 1000
169.                },
170.                yAxis: {
171.                    title: {
172.                        text: 'Celsius'
173.                    }
174.                },
175.                series: [{
176.                    name: 'Living Room Temperature',
177.                    color: '#0000cc',
178.                                    data: [<?php echo $final; ?>],
179.                                    pointStart: (<?php echo $start;?>+1
    0800 )*1000,
180.                                    pointInterval: 720*1000  //interval
     between time = 12min
181.                                },{
182.                    name: 'Schedule',
183.                    color: '#006600',
184.                    data: [<?php echo join($schedule1, ',') ?>]
185.                },{
186.                    name: 'Outside Temperature',
187.                    color: '#ff3300',
188.                    data: [<?php echo $final2; ?>],
189.                    pointStart: (<?php echo $start2; ?>+10800)*1000,
190.                    pointInterval:720*1000
191.                }]
192.            });
193.        });
194.    </script>
195.    </head>
196.    <body>
197.
198.            <nav class="navbar navbar-default" id="menu">
199.                    <div class="container-fluid">
```

```
                            <div class="navbar-header">
                                    <a class="navbar-
    brand">Smart Thermostat</a>
                            </div>
                            <ul class="nav navbar-nav">
                        <li><a href="main.php">Main Page</a><li>
                        <li><a href="schedule.php">Schedule</a><li>
                                    <li class="active"><a href="graph24
    h.php">Statistics</a></li>
                            </ul>
                        </div>
                    </nav>

            <div class="container">
                            <div class="row">
                                <div class="col-xs-12">
                                <div id="container2"></div>
                    </div>
                </div>
            </div>
    </body>
    </html>
```

16.graph24h.css

```
#container2 {
    width:90%;
        float:left;
        padding:10px;
}
body {
        background:url('https://projectmgtcoach.com/wp-
    content/uploads/2015/03/light-blue-backgrounds-for-websites.jpg');
        width:100%;
        height:100%;
        background-position:center center;
        background-size:cover;
        z-index:0;
}
#menu{
        width:100%;
        font-size: 25px;
}
```

17.statistic24h.py

```
#!/usr/bin/python

import rrdtool
ret = rrdtool.fetch('/home/pi/Desktop/temperature.rrd','AVERAGE', '--
    start', 'now-24h')
time = rrdtool.first
max = ret[2]
list = []
for temp in max:
```

```
    n = temp[0]
    list.append(n) #list with all the temperatures
print str(list)[1:-5] #the last one is a None
time = ret[0] #get the start
timestamp = time[0]
print timestamp
```

18.out_statistic24.py

```
#!/usr/bin/python

import rrdtool
ret = rrdtool.fetch('/home/pi/Desktop/out_temperature.rrd','AVERAGE', '--start', 'now-24h')
time = rrdtool.first
max = ret[2]
list = []
for temp in max:
    n = temp[0]
    list.append(n) #list with all the temperatures
print str(list)[1:-5] #the last one is a None
time = ret[0] #get the start
timestamp = time[0]
print timestamp
```

19.graph_week.php

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Raspberry Project</title>
<link rel="stylesheet" type="text/css" href="graph_week.css">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.2/jquery.min.js"></script>
<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>

<?php
    //get the data from the rrd database (temperature)
    $output = popen('python ./statistic_week.py', 'r');
    $list = fgets($output);
    $start = fgets($output);
    fclose($output);
    $data = explode(', ', $list); //making the list an array to be processed
    $start+=720; //resolution = 720
    //get the specified field we need (last 24h time-templerature) from the database
    foreach ($data as $value) {
    //for earch value in the table data do
        if (trim($value) != "None" ) {
```

```
24.             $value = number_format($value,1,'.',',');//show only one de
    cimal
25.             $mydata[]=$value;
26.         }else{
27.             $mydata[]='null';
28.         }
29.     }
30.     $final = implode(" , ", $mydata); //implode a comma between all val
    ues
31.
32.
33.     //get the outside temperature from rrd daabase
34.         $output = popen('python ./out_statistic_week.py', 'r');
35.         $list = fgets($output);
36.         $start2 = fgets($output);
37.         fclose($output);
38.         $data2 = explode(', ', $list); //making the list an array to be
     processed
39.         $start2+=720; //resolution = 720
40.         //get the specified field we need (last 24h time-
    templerature) from the database
41.         foreach ($data2 as $value) {
42.         //for earch value in the table data do
43.             if (trim($value) != "None" ) {
44.             $value = number_format($value,1,'.',',');//show only one de
    cimal
45.                         $mydata2[]=$value;
46.                 }else{
47.                         $mydata2[]='null';
48.                 }
49.         }
50.         $final2 = implode(" , ", $mydata2); //implode a comma between a
    ll values
51.
52.
53. ?>
54.         <script type="text/javascript" src="https://ajax.googleapis.com
    /ajax/libs/jquery/1.12.0/jquery.min.js"></script>
55. <script src="https://code.highcharts.com/highcharts.js"></script>
56. <script src="https://code.highcharts.com/modules/exporting.js"></script
    >
57.
58. <script type="text/javascript">
59.     $(document).ready(function() {
60.         var chart = new Highcharts.Chart({
61.             chart: {
62.                 renderTo: 'container2',
63.                 zoomType: 'x',
64.                 backgroundColor: null
65.             },
66.             title: {
67.                 text: 'Temperature - Last week'
68.             },
69.             xAxis: {
70.                 type: 'datetime',
71.                 maxZoom: 12* 600 * 1000
72.             },
73.             yAxis: {
74.                 title: {
```

```
                    text: 'Celsius'
                }
            },
            series: [{
                name: 'Living Room Temperature',
                color: '#0000cc',
                data: [<?php echo $final; ?>],
                pointStart: (<?php echo $start;?>+10800 )*1000,
                pointInterval: 720*1000*3  //interval between time = 12 min
                },{
                name: 'Outside Temperature',
                color: '#ff3300',
                data: [<?php echo $final2; ?>],
                pointStart: (<?php echo $start2; ?>+10800)*1000,
                pointInterval:720*1000*3
            }]
        });
    });
</script>
</head>
<body>

        <nav class="navbar navbar-default" id="menu">
                    <div class="container-fluid">
                        <div class="navbar-header">
                      <a class="navbar-brand">Smart Thermostat</a>
                     </div>
                    <ul class="nav navbar-nav">
                    <li><a href="main.php">Main Page</a><li>
                    <li><a href="schedule.php">Schedule</a><li>
                    <li class="dropdown">
                    <a class="dropdown-toggle" data-toggle="dropdown" >Statistics
                    <span class="caret"></span></a>
                    <ul class="dropdown-menu">
                    <li><a href="graph24h.php">Day</a></li>
                  <li class="active"><a href="graph_week.php">Week</a></li>
                     </ul>
                      </li>
                    </ul>
                 </div>
    </nav>
             <div class="container">
                            <div class="row">
                                <div class="col-xs-12">
                                <div id="container2"></div>
                    </div>
                </div>
            </div>
   </body>
   </html>
```

20.graph_week.css

```
1. #container2 {
```

```
    width:90%;
        float:left;
        padding:10px;
}
body {
        background:url('https://projectmgtcoach.com/wp-content/uploads/2015/03/light-blue-backgrounds-for-websites.jpg');
        width:100%;
        height:100%;
        background-position:center center;
        background-size:cover;
        z-index:0;
}
#menu{
        width:100%;
        font-size: 25px;
}
```

21.statistic_week.py

```
#!/usr/bin/python

import rrdtool
ret = rrdtool.fetch('/home/pi/Desktop/temperature.rrd','AVERAGE', '--start', 'now-7d')
time = rrdtool.first
max = ret[2]
list = []
for temp in max:
    n = temp[0]
    list.append(n) #list with all the temperatures
print str(list)[1:-5] #the last one is a None
time = ret[0] #get the start
timestamp = time[0]
print timestamp
```

22.out_statistic_week.py

```
#!/usr/bin/python

import rrdtool
ret = rrdtool.fetch('/home/pi/Desktop/out_temperature.rrd','AVERAGE', '--start', 'now-7d')
time = rrdtool.first
max = ret[2]
list = []
for temp in max:
    n = temp[0]
    list.append(n) #list with all the temperatures
print str(list)[1:-5] #the last one is a None
time = ret[0] #get the start
timestamp = time[0]
print timestamp
```

23.Ti_on-off_delay.py

```
#!/usr/bin/python

import math
import matplotlib.pyplot as plt
import numpy as np
import time
from scipy.integrate import quad

Ci = 2.07 *60.0 #kWmin/oC
Ria = 5.29 #oC/kW
A = -1/(Ria*Ci)
B1 = 1/(Ria*Ci)
B2 = 1/Ci

Ta = [0.0]
Twant =  20.0 #celsius
Ti = [15]
AC = [0] #at start ac was off
error = []
error.append( Twant - Ti[0])
Fh = []
Fh.append( 5 - (5*Ta[0]/Twant)) #kW

S = (A*A)/2
F = 1 + A + S

k1 = lambda r: (1 + (A*r) + (S*r*r))*B1
k2 = lambda r: (1 + (A*r) + (S*r*r))*B2

C1 = quad(k1,0,1)
C2 = quad(k2,0,1)

t = [0]
for i in range (1,1001):
    t.append(i) #t=1,2,3,..,1000min!!!
    if (i % 10) == 1 or i==1:
        if error[i-1] > 1: #Twant>Ti+1 then open ac
            AC.append(Twant)
            Fh.append( 5 - (5*Ta[0]/Twant))

        elif error[i-1] < 1 and error[i-1] > -1:
            old = AC[i-1]
            AC.append(old)
            old = Fh[i-1]
            Fh.append(old)

        else: #close ac
            AC.append(0)
            Fh.append(0)

    Ti.append((F*Ti[i-1]) + (C1[0]*Ta[i-1]) + (C2[0]*Fh[i-1]))
    error.append( Twant - Ti[i]) #new error
    if i!=1 and ((i % 10) != 1):
```

```
        old = AC[i-1]
        AC.append(old)
        old = Fh[i-1]
        Fh.append(old)
    old = Ta[i-1] #to be able to plot outside temp
    Ta.append(old)

time=[]
for x in t:
    time.append(x/60.0)

#plot
lc = plt.plot(time,Ti,'b-',label = "Inside Temperature")
la = plt.plot(time,error,'r-', label= "Error")
lb = plt.step(time,AC,'g-', label="AirConditioner")
ld = plt.plot(time,Ta,'m-', label="Outside Temperature")
lf = plt.step(time,Fh,'y-',label="Power")
ll = plt.title('ON-OFF (Ti model)')
lx = plt.xlabel('Time(h)')
ly = plt.ylabel('Temperature(Celsius)')
plt.legend(loc=3, bbox_to_anchor = (0, -
  0.3, 1., 1.102),mode="expand",ncol=2)
plt.savefig('./Ti_ON-OFF_delay.png',bbox_inches='tight')
plt.show()
```

24.Ti_pid.py

```
#!/usr/bin/python

import math
import matplotlib.pyplot as plt
import numpy as np
import time
import datetime
from scipy.integrate import quad

#simulation of pid controller
def update(kp,ki,kd,period,windup):
    #constants of model
    Ta = [0.0]
    Twant =  20.0 #celsius
    Ti = [15]
    Fh = []
    Fh.append(5 - (5*Ta[0]/Twant)) #kW
    Ci = 2.07*60
    Ria = 5.29
    AC = [0]
    A = -1.0/(Ria*Ci)
    B1 = 1.0/(Ria*Ci)
    B2 = 1/Ci
    w = Twant
    x = Ti[0]
    t = [0]
    error = []
    error.append(0)

    S = (A*A)/2
```

```
    F = 1 + A + S

    k1 = lambda r: (1 + (A*r) + (S*r*r))*B1
    k2 = lambda r: (1 + (A*r) + (S*r*r))*B2
    C1 = quad(k1,0,1)
    C2 = quad(k2,0,1)

    last_error = 0
    Iy = 0
    period_min = period / 60
    #calculate the y ratio for this 10min period
    for i in range(1,81):
        x = Ti [(i*10) -9 -1]
        e = w - x # error
        de = e - last_error
        dt = 600 #sec
        if dt >=period :
            if e>-2 and e<2:
                Py = (kp * e) + 0.5
                dt = dt / 60 #dt must be in minutes
                Iy += e*dt
                if Iy < -windup:
                    Iy = -windup
                elif Iy > windup:
                    Iy = windup
                Iy = Iy * ki
                if dt > 0 :
                    Dy = (de / dt) * kd
                y = Py + Iy + Dy
                if y > 1:
                    y = 1
                elif y < 0:
                    y = 0
            elif e>=2:
                y = 1
            else:
                y = 0
            last_error = e

        #sleep for 10min until next evaluation
        timeON = y*period_min #for period_min - timeON = timeO$
            timeON = round(timeON)
        if timeON<=3:
            timeON = 0
        elif timeON>=7:
            timeON = 10
        timeON = int(timeON)
        timeOFF = int(period_min - timeON)

            if timeON!=0:
            for l in range(1,timeON+1):
                k = (i*10) -9 +l -1
                Fh.append(5 - (5*Ta[k-1]/Twant))
                        AC.append(Twant)
                old = Ta[k-1]
                                Ta.append(old)
                tempera = (F*Ti[k-1]) + (C1[0]*Ta[k-1]) + (C2[0]*Fh[k-
1])
                        Ti.append( tempera)
```

```
                t.append(k)
                error.append(Twant-tempera)

        if timeOFF!=0:
                for l in range(1,timeOFF+1):
                k = (i*10) - 9 + l -1 + timeON
                Fh.append(0)
                        AC.append(0)
                old = Ta[k-1]
                Ta.append(old)
                tempera = (F*Ti[k-1]) + (C1[0]*Ta[k-1]) + (C2[0]*Fh[k-
   1])
                            Ti.append( tempera)
                    t.append(k)
                    error.append(Twant - tempera)

        time = []
        Tout =[]
        for k in t:
            time.append(k/60.0)
        lc = plt.plot(time,Ti,'b-',label = "Inside Temperature")
        lb = plt.step(time,AC,'g-', label="AirConditioner")
        la = plt.plot(time,Ta,'m-',label="Outside Temperature")
        ld = plt.plot(time,error,'r-',label="Error")
        lf = plt.step(time,Fh,'y-',label="Power")
        ll = plt.title('PID (Ti model)')
        lx = plt.xlabel('Time(h)')
        ly = plt.ylabel('Temperature(Celsius)')
        plt.legend(loc=3, bbox_to_anchor = (0, -
   0.3, 1., 1.102),mode="expand",ncol=2)
        plt.savefig('./Ti_pid.png',bbox_inches='tight')
        plt.show()
        return y

    #set parameters
    kp = 0.25 #Xp = 4 bathmoi
    period = 600 # 10 minutes
    #Tu = 4minutes
    ki = 0.2#(2 * kp) / 4 #0.125
    period_minu = period / 60 #in minutes
    kd = 0.05#period_minu / 8.0 # 0.25
    windup = 20.0
    list = []

    y = update(kp,ki,kd,period,windup)
```

25. TiTh_on-off_delay.py

```
#!/usr/bin/python

import math
import matplotlib.pyplot as plt
import numpy as np
import time
from scipy.integrate import quad

Ci = 1.36 *60.0 #kWmin/oC
```

```
Ria = 5.31 #oC/kW
Ch = 0.309 *60.0
Rih = 0.639
A1 = -1.0/(Rih*Ch)
A2 = 1.0/(Rih*Ch)
A3 = 1.0/(Rih*Ci)
A4 = (-Rih-Ria)/(Ria*Rih*Ci)
A = np.array([[A1, A2], [A3, A4]])
B = np.array([[0.0, 1/Ch], [(1/(Ria*Ci)), 0.0]])

Ta = [0.0]
Twant =  20.0 #celsius
Ti = [15]
Th = [15] #temperature of the heaters
AC = [0] #at start ac was off
error = []
error.append( Twant - Ti[0])
Fh = []
Fh.append( 5 - (5*Ta[0]/Twant) ) #kW

I = np.array([[1, 0], [0, 1]])
S = (np.dot(A,A))/2
F = I + A + S

k1 = lambda r: (I[0,1] + np.dot(A[0,1],r) + np.dot(S[0,1],r**2))*B[1,0]

k2 = lambda r: (I[0,0] + np.dot(A[0,0],r) + np.dot(S[0,0],r**2))*B[0,1]

k3 = lambda r: (I[1,1] + np.dot(A[1,1],r) + np.dot(S[1,1],r**2))*B[1,0]

k4 = lambda r: (I[1,0] + np.dot(A[1,0],r) + np.dot(S[1,0],r**2))*B[0,1]

C1 = quad(k1,0,1)
C2 = quad(k2,0,1)
C3 = quad(k3,0,1)
C4  = quad(k4,0,1) #integration from 0 to 1min (exp(A*r)*B ) dr
C = np.array([[C1[0], C2[0]],[C3[0], C4[0]]])

t = [0]
for i in range (1,1001):
    t.append(i) #t=1,2,3,..,1000min!!!
    if i==1 or (i % 10) == 1:
        if error[i-1] > 1: #Twant>Ti then open ac
            Fh.append( 5 - (5*Ta[i-1]/Twant)) #kW
            AC.append(Twant)

        elif error[i-1] < 1 and error[i-1] > -1:
            old = AC[i-1]
            AC.append(old)
            old = Fh[i-1]
            Fh.append(old)
        else: #close ac
            Fh.append(0)
            AC.append(0)

    Ti.append((F[1,0]*Th[i-1]) + (F[1,1]*Ti[i-1]) + (C[1,0]*Ta[i-
1]) + (C[1,1]*Fh[i-1]) ) #solution to the d.e
    Th.append((F[0,0]*Th[i-1]) + (F[0,1]*Ti[i-1]) + (C[0,0]*Ta[i-
1]) + (C[0,1]*Fh[i-1]) )
```

```
    error.append( Twant - Ti[i]) #new error
    if i!=1 and ((i % 10) != 1):
        old = AC[i-1]
        AC.append(old)
        old = Fh[i-1]
        Fh.append(old)
    old = Ta[i-1] #to be able to plot outside temp
    Ta.append(old)

time=[]
for x in t:
    time.append(x/60.0)

#plot
lc = plt.plot(time,Ti,'b-',label = "Inside Temperature")
la = plt.plot(time,error,'r-', label= "Error")
lb = plt.step(time,AC,'g-', label="AirConditioner")
ld = plt.plot(time,Ta,'m-', label="Outside Temperature")
lf = plt.plot(time,Fh,'y-',label="Power")
lg = plt.plot(time,Th,'k-', label = "Heaters Teperature")
ll = plt.title('ON-OFF (TiTh model)')
lx = plt.xlabel('Time(h)')
ly = plt.ylabel('Temperature(Celsius)')
plt.legend(loc=3, bbox_to_anchor = (0, -
0.3, 1., 1.102),mode="expand",ncol=2)
plt.savefig('./TiTh_ON-OFF_delay.png',bbox_inches='tight')
plt.show()
```

26. TiTh_pid.py

```
#!/usr/bin/python

import math
import matplotlib.pyplot as plt
import numpy as np
import time
import datetime
from scipy.integrate import quad

#simulation of pid controller
def update(kp,ki,kd,period,windup):
    #constants of model
    Ta = [0.0]
    Twant =  20.0 #celsius
    Ti = [15]
    Th = [15]
    Fh = []
    Fh.append( 5 - (5*Ta[0]/Twant)) #kW
        Ci = 1.36*60
        Ria = 5.31
    Ch = 0.309 *60.0
    Rih = 0.639
    A1 = -1.0/(Rih*Ch)
    A2 = 1.0/(Rih*Ch)
    A3 = 1.0/(Rih*Ci)
    A4 = (-Rih-Ria)/(Ria*Rih*Ci)
    A = np.array([[A1, A2], [A3, A4]])
```

```
    B = np.array([[0.0, 1/Ch], [(1/(Ria*Ci)), 0.0]])

    AC = [0]
    w = Twant
    x = Ti[0]
    t = [0]
    error = []
    error.append(0)

    I = np.array([[1, 0], [0, 1]])
    S = (np.dot(A,A))/2
    F = I + A + S

    k1 = lambda r: (I[0,1] + np.dot(A[0,1],r) + np.dot(S[0,1],r**2))*B[
1,0]
    k2 = lambda r: (I[0,0] + np.dot(A[0,0],r) + np.dot(S[0,0],r**2))*B[
0,1]
    k3 = lambda r: (I[1,1] + np.dot(A[1,1],r) + np.dot(S[1,1],r**2))*B[
1,0]
    k4 = lambda r: (I[1,0] + np.dot(A[1,0],r) + np.dot(S[1,0],r**2))*B[
0,1]
    C1 = quad(k1,0,1)
    C2 = quad(k2,0,1)
    C3 = quad(k3,0,1)
    C4  = quad(k4,0,1) #integration from 0 to 1min (exp(A*r)*B ) dr
    C = np.array([[C1[0], C2[0]],[C3[0], C4[0]]])

    last_error = 0
    Iy = 0
    period_min = period / 60
    #calculate the y ratio for this 10min period
    for i in range(1,81):
        x = Ti [(i*10) -9 -1]
        e = w - x # error
        de = e - last_error
        dt = 600 #sec
        if dt >=period :
            if e>-2 and e<2:
                Py = (kp * e) + 0.5
                dt = dt / 60 #dt must be in minutes
                Iy += e*dt
                if Iy < -windup:
                    Iy = -windup
                elif Iy > windup:
                    Iy = windup
                Iy = Iy * ki
                if dt > 0 :
                    Dy = (de / dt) * kd
                y = Py + Iy + Dy
                if y > 1:
                    y = 1
                elif y < 0:
                    y = 0
            elif e>=2:
                y = 1
            else:
                y = 0
            last_error = e

```

```
83.         #sleep for 10min until next evaluation
84.         timeON = y*period_min #for period_min - timeON = timeO$
85.             timeON = round(timeON)
86.         if timeON<=3:
87.             timeON = 0
88.         elif timeON>=7:
89.             timeON = 10
90.         timeON = int(timeON)
91.         timeOFF = int(period_min - timeON)
92.
93.             if timeON!=0:
94.             for l in range(1,timeON+1):
95.                 k = (i*10) - 9 + l - 1
96.                         Fh.append( 5 - (5*Ta[k -1]/Twant))
97.                 AC.append(Twant)
98.                 old = Ta[k  -1]
99.                 Ta.append(old)
100.                    tempera = (F[1,0]*Th[k-1]) + (F[1,1]*Ti[k-
    1]) + (C[1,0]*Ta[k-1]) + (C[1,1]*Fh[k-1])
101.                            Ti.append( tempera)
102.                        Th.append((F[0,0]*Th[k-1]) + (F[0,1]*Ti[k-
    1]) + (C[0,0]*Ta[k-1]) + (C[0,1]*Fh[k-1]) )
103.                    t.append(k)
104.                    error.append(Twant-tempera)
105.            if timeOFF!=0:
106.                    for l in range(1,timeOFF+1):
107.                    k = (i*10) - 9 + l -1 + timeON
108.                    Fh.append(0)
109.                            AC.append(0)
110.                    old = Ta[k-1]
111.                    Ta.append(old)
112.                    tempera = (F[1,0]*Th[k-1]) + (F[1,1]*Ti[k-
    1]) + (C[1,0]*Ta[k-1]) + (C[1,1]*Fh[k-1])
113.                            Ti.append( tempera)
114.                    Th.append((F[0,0]*Th[k-1]) + (F[0,1]*Ti[k-
    1]) + (C[0,0]*Ta[k-1]) + (C[0,1]*Fh[k-1]) )
115.                    t.append(k)
116.                    error.append(Twant - tempera)
117.        time = []
118.        for k in t:
119.            time.append(k/60.0)
120.
121.        #plot
122.        lc = plt.plot(time,Ti,'b-',label = "Inside Temperature")
123.        lb = plt.step(time,AC,'g-', label="AirConditioner")
124.        la = plt.plot(time,Ta,'m-',label="Outside Temperature")
125.        ld = plt.plot(time,error,'r-',label="Error")
126.        lf = plt.plot(time,Fh,'y-',label="Power")
127.        lg = plt.plot(time,Th,'k-',label="Heaters Temperature")
128.        ll = plt.title('PID (TiTh model)')
129.        lx = plt.xlabel('Time(h)')
130.        ly = plt.ylabel('Temperature(Celsius)')
131.        plt.legend(loc=3, bbox_to_anchor = (0, -
    0.3, 1., 1.102),mode="expand",ncol=2)
132.        plt.savefig('./TiTh_pid.png',bbox_inches='tight')
133.        plt.show()
134.        return y
135.
136.    #set parameters
```

```
137.    kp = 0.25 #Xp = 4 bathmoi
138.    period = 600 # 10 minutes
139.    #Tu = 4minutes
140.    ki = 0.2#(2 * kp) / 4 #0.125
141.    period_minu = period / 60 #in minutes
142.    kd = 0.05#period_minu / 8.0 # 0.25
143.    windup = 20.0
144.    list = []
145.
146.    y = update(kp,ki,kd,period,windup)
```

27. TiTeTh_on-off_delay.py

```
1.  #!/usr/bin/python
2.
3.  import math
4.  import matplotlib.pyplot as plt
5.  import numpy as np
6.  import time
7.  from scipy.integrate import quad
8.
9.  Ci = 1.07 *60.0 #kWmin/oC
10. Ch = 0.00139 *60.0
11. Ce = 2.92 * 60.0
12. Rih = 93.4
13. Rie = 0.863
14. Rea = 4.54
15. A1 = (-Rie-Rea)/(Rie*Rea*Ce)
16. A2 = 0.0
17. A3 = 1.0/(Rie*Ce)
18. A4 = 0.0
19. A5 = -1.0/(Rih*Ch)
20. A6 = 1.0/(Rih*Ch)
21. A7 = 1.0/(Rie*Ci)
22. A8 = 1.0/(Rih*Ci)
23. A9 = (-Rih-Rie)/(Rie*Rih*Ci)
24. A = np.array([[A1, A2, A3], [A4, A5, A6], [A7, A8, A9]])
25. B = np.array([[1.0/(Rea*Ce), 0.0], [0.0, 1.0/Ch], [0.0, 0.0]])
26.
27. Ta = [0.0]
28. Twant =  20.0 #celsius
29. Ti = [15]
30. Th = [15] #temperature of the heaters
31. Te = [15] #temperature of the envelop
32. AC = [0] #at start ac was off
33. error = []
34. error.append( Twant - Ti[0])
35. Fh = []
36. Fh.append( 5 - (5*Ta[0]/Twant) ) #kW
37.
38. I = np.array([[1, 0, 0], [0, 1, 0], [0, 0, 1]])
39. S = (np.dot(A,A))/2
40. F = I + A + S
41.
42. k1 = lambda r: (I[0,0] + np.dot(A[0,0],r) + np.dot(S[0,0],r**2))*B[0,0]
```

```
k2 = lambda r: (I[0,1] + np.dot(A[0,1],r) + np.dot(S[0,1],r**2))*B[1,1]

k3 = lambda r: (I[1,0] + np.dot(A[1,0],r) + np.dot(S[1,0],r**2))*B[0,0]

k4 = lambda r: (I[1,1] + np.dot(A[1,1],r) + np.dot(S[1,1],r**2))*B[1,1]

k5 = lambda r: (I[2,0] + np.dot(A[2,0],r) + np.dot(S[2,0],r**2))*B[0,0]

k6 = lambda r: (I[2,1] + np.dot(A[2,1],r) + np.dot(S[2,1],r**2))*B[1,1]

C1 = quad(k1,0,1)
C2 = quad(k2,0,1)
C3 = quad(k3,0,1)
C4 = quad(k4,0,1) #integration from 0 to 1min (exp(A*r)*B ) dr
C5 = quad(k5,0,1)
C6 = quad(k6,0,1)
C = np.array([[C1[0], C2[0]],[C3[0], C4[0]], [C5[0], C6[0]]])

t = [0]
for i in range (1,1001):
    t.append(i) #t=1,2,3,..,1000min!!!
    if i==1 or (i % 10)==1:
        if error[i-1] > 1: #Twant>Ti then open ac
            Fh.append( 5 - (5*Ta[i-1]/Twant)) #kW
            AC.append(Twant)

        elif error[i-1] < 1 and error[i-1] > -1:
            old = AC[i-1]
            AC.append(old)
            old = Fh[i-1]
            Fh.append(old)

        else: #close ac
            Fh.append(0)
            AC.append(0)

    Te.append((F[0,0]*Te[i-1]) + (F[0,1]*Th[i-1]) + (F[0,2]*Ti[i-
1]) + (C[0,0]*Ta[i-1]) + (C[0,1]*Fh[i-1])) #solution to the d.e
    Th.append((F[1,0]*Te[i-1]) + (F[1,1]*Th[i-1]) + (F[1,2]*Ti[i-
1]) + (C[1,0]*Ta[i-1]) + (C[1,1]*Fh[i-1]))
    Ti.append((F[2,0]*Te[i-1]) + (F[2,1]*Th[i-1]) + (F[2,2]*Ti[i-
1]) + (C[2,0]*Ta[i-1]) + (C[2,1]*Fh[i-1]))
    error.append( Twant - Ti[i]) #new error
    if i!=1 and ((i % 10) != 1):
        old = AC[i-1]
        AC.append(old)
        old = Fh[i-1]
        Fh.append(old)
    old = Ta[i-1] #to be able to plot outside temp
    Ta.append(old)

time=[]
for x in t:
    time.append(x/60.0)

#plot
lc = plt.plot(time,Ti,'b-',label = "Inside Temperature")
la = plt.plot(time,error,'r-', label= "Error")
lb = plt.step(time,AC,'g-', label="AirConditioner")
```

```
ld = plt.plot(time,Ta,'m-', label="Outside Temperature")
lf = plt.plot(time,Fh,'y-',label="Power")
#lg = plt.plot(time,Th,'k-', label = "Heaters Temperature")
ls = plt.plot(time,Te,'c-', label = "Envelop Temperature")
ll = plt.title('ON-OFF (TiTeTh model)')
lx = plt.xlabel('Time(h)')
    ly = plt.ylabel('Temperature(Celsius)')
    plt.legend(loc=3, bbox_to_anchor = (0, -
  0.3, 1., 1.102),mode="expand",ncol=2)
    plt.savefig('./TiTeTh_ON-OFF_delay.png',bbox_inches='tight')
    plt.show()
```

28. TiTeTh_pid.py

```
#!/usr/bin/python

import math
import matplotlib.pyplot as plt
import numpy as np
import time
import datetime
from scipy.integrate import quad

#simulation of pid controller
def update(kp,ki,kd,period,windup):
    #constants of model
    Ta = [0.0]
    Twant =  20.0 #celsius
    Ti = [15]
    Th = [15]
    Te = [15]
    Fh = []
    Fh.append( 5 - (5*Ta[0]/Twant)) #kW
        Ci = 1.07*60
    Ch = 0.00139 *60.0
    Ce = 2.92 * 60.0
    Rih = 93.4
    Rie = 0.863
    Rea = 4.54
    A1 = (-Rie-Rea)/(Rie*Rea*Ce)
    A2 = 0.0
    A3 = 1.0/(Rie*Ce)
    A4 = 0.0
    A5 = -1.0/(Rih*Ch)
    A6 = 1.0/(Rih*Ch)
    A7 = 1.0/(Rie*Ci)
    A8 = 1.0/(Rih*Ci)
    A9 = (-Rih-Rie)/(Rie*Rih*Ci)
    A = np.array([[A1, A2, A3], [A4, A5, A6], [A7, A8, A9]])
    B = np.array([[1.0/(Rea*Ce), 0.0], [0.0, 1.0/Ch], [0.0, 0.0]])

    AC = [0]
    w = Twant
    x = Ti[0]
    t = [0]
    error = []
    error.append(0)
```

```
44.
45.     I = np.array([[1, 0, 0], [0, 1, 0], [0, 0, 1]])
46.     S = (np.dot(A,A))/2
47.     F = I + A + S
48.
49.     k1 = lambda r: (I[0,0] + np.dot(A[0,0],r) + np.dot(S[0,0],r**2))*B[
    0,0]
50.     k2 = lambda r: (I[0,1] + np.dot(A[0,1],r) + np.dot(S[0,1],r**2))*B[
    1,1]
51.     k3 = lambda r: (I[1,0] + np.dot(A[1,0],r) + np.dot(S[1,0],r**2))*B[
    0,0]
52.     k4 = lambda r: (I[1,1] + np.dot(A[1,1],r) + np.dot(S[1,1],r**2))*B[
    1,1]
53.     k5 = lambda r: (I[2,0] + np.dot(A[2,0],r) + np.dot(S[2,0],r**2))*B[
    0,0]
54.     k6 = lambda r: (I[2,1] + np.dot(A[2,1],r) + np.dot(S[2,1],r**2))*B[
    1,1]
55.     C1 = quad(k1,0,1)
56.     C2 = quad(k2,0,1)
57.     C3 = quad(k3,0,1)
58.     C4 = quad(k4,0,1) #integration from 0 to 1min (exp(A*r)*B ) dr
59.     C5 = quad(k5,0,1)
60.     C6 = quad(k6,0,1)
61.     C = np.array([[C1[0], C2[0]],[C3[0], C4[0]], [C5[0], C6[0]]])
62.
63.     last_error = 0
64.     Iy = 0
65.     period_min = period / 60
66.     #calculate the y ratio for this 10min period
67.     for i in range(1,81):
68.         x = Ti [(i*10) -9 -1]
69.         e = w - x # error
70.         de = e - last_error
71.         dt = 600 #sec
72.         if dt >=period :
73.             if e>-2 and e<2:
74.                 Py = (kp * e) + 0.5
75.                 dt = dt / 60 #dt must be in minutes
76.                 Iy += e*dt
77.                 if Iy < -windup:
78.                     Iy = -windup
79.                 elif Iy > windup:
80.                     Iy = windup
81.                 Iy = Iy * ki
82.                 if dt > 0 :
83.                     Dy = (de / dt) * kd
84.                 y = Py + Iy + Dy
85.                 if y > 1:
86.                     y = 1
87.                 elif y < 0:
88.                     y = 0
89.             elif e>=2:
90.                 y = 1
91.             else:
92.                 y = 0
93.             last_error = e
94.
95.         #sleep for 10min until next evaluation
96.         timeON = y*period_min #for period_min - timeON = timeO$
```

```
            timeON = round(timeON)
        if timeON<=3:
            timeON = 0
            elif timeON>=7:
                timeON = 10
            timeON = int(timeON)
            timeOFF = int(period_min - timeON)

                if timeON!=0:
                for l in range(1,timeON+1):
                    k = (i*10) - 9 + l - 1
                            Fh.append( 5 - (5*Ta[k -1]/Twant))
                    AC.append(Twant)
                    old = Ta[k  -1]
                    Ta.append(old)
                    Te.append((F[0,0]*Te[k-1]) + (F[0,1]*Th[k-
   1]) + (F[0,2]*Ti[k-1]) + (C[0,0]*Ta[k-1]) + (C[0,1]*Fh[k-1]))
                    Th.append((F[1,0]*Te[k-1]) + (F[1,1]*Th[k-
   1]) + (F[1,2]*Ti[k-1]) + (C[1,0]*Ta[k-1]) + (C[1,1]*Fh[k-1]) )
                    tempera = ((F[2,0]*Te[k-1]) + (F[2,1]*Th[k-
   1]) + (F[2,2]*Ti[k-1]) +(C[2,0]*Ta[k-1]) + (C[2,1]*Fh[k-1]))
                            Ti.append( tempera)
                    t.append(k)
                    error.append(Twant-tempera)

            if timeOFF!=0:
                    for l in range(1,timeOFF+1):
                    k = (i*10) - 9 + l -1 + timeON
                    Fh.append(0)
                            AC.append(0)
                    old = Ta[k-1]
                    Ta.append(old)
                    Te.append((F[0,0]*Te[k-1]) + (F[0,1]*Th[k-
   1]) + (F[0,2]*Ti[k-1]) + (C[0,0]*Ta[k-1]) + (C[0,1]*Fh[k-1]))
                                        Th.append((F[1,0]*Te[k-
   1]) + (F[1,1]*Th[k-1]) + (F[1,2]*Ti[k-1]) + (C[1,0]*Ta[k-
   1]) + (C[1,1]*Fh[k-1]) )
                    tempera = ((F[2,0]*Te[k-1]) + (F[2,1]*Th[k-
   1]) + (F[2,2]*Ti[k-1]) +(C[2,0]*Ta[k-1]) + (C[2,1]*Fh[k-1]))
                                        Ti.append( tempera)
                                        t.append(k)
                    error.append(Twant - tempera)
        time = []
        for k in t:
            time.append(k/60.0)

        #plot
        lc = plt.plot(time,Ti,'b-',label = "Inside Temperature")
        lb = plt.step(time,AC,'g-', label="AirConditioner")
        la = plt.plot(time,Ta,'m-',label="Outside Temperature")
        ld = plt.plot(time,error,'r-',label="Error")
        lf = plt.plot(time,Fh,'y-',label="Power")
        #le = plt.plot(time,Th,'k-', label = "Heaters Temperature")
        lg = plt.plot(time,Te,'c-',label="Envelop Temperature")
        ll = plt.title('PID (TiTeTh model)')
        lx = plt.xlabel('Time(h)')
        ly = plt.ylabel('Temperature(Celsius)')
        plt.legend(loc=3, bbox_to_anchor = (0, -
   0.3, 1., 1.102),mode="expand",ncol=2)
```

```
        plt.savefig('./TiTeTh_pid.png',bbox_inches='tight')
        plt.show()
        return y

    #set parameters
    kp = 0.25 #Xp = 4 bathmoi
    period = 600 # 10 minutes
    #Tu = 4minutes
    ki = 0.2#(2 * kp) / 4 #0.125
    period_minu = period / 60 #in minutes
    kd = 0.05#period_minu / 8.0 # 0.25
    windup = 20.0
    list = []

    y = update(kp,ki,kd,period,windup)
```

29. TiTeThTs_on-off_delay.py

```
#!/usr/bin/python

import math
import matplotlib.pyplot as plt
import numpy as np
import time
from scipy.integrate import quad

Ci = 0.143 *60.0 #kWmin/oC
Ch = 0.321 *60.0
Ce = 3.24 * 60.0
Cs = 0.619 * 60.0
Rih = 0.383
Rie = 0.909
Rea = 4.47
Ris = 0.115
A1 = (-Rie-Rea)/(Rie*Rea*Ce)
A2 = 0.0
A3 = 0.0
A4 = 1.0/(Rie*Ce)
A5 = 0.0
A6 = -1.0/(Rih*Ch)
A7 = 0.0
A8 = 1.0/(Rih*Ch)
A9 = 0.0
A10 = 0.0
A11 = -1.0/(Ris*Cs)
A12 = 1.0/(Ris*Cs)
A13 = 1.0/(Rie*Ci)
A14 = 1.0/(Rih*Ci)
A15 = 1.0/(Ris*Ci)
A16 = (-(Rih*Ris)-(Rie*Ris) - (Rih*Rie))/(Rie*Rih*Ris*Ci)
A = np.array([[A1, A2, A3, A4], [A5, A6, A7, A8], [A9, A10, A11, A12],
[A13, A14, A15, A16]])
B = np.array([[1.0/(Rea*Ce), 0.0], [0.0, 1.0/Ch], [0.0, 0.0], [0.0, 0.0
]])

Ta = [0.0]
Twant =  20.0 #celsius
```

```
Ti = [15]
Th = [15] #temperature of the heaters
Te = [15] #temperature of the envelop
Ts = [15] #temperature of the sensor
AC = [0] #at start ac was off
error = []
error.append( Twant - Ti[0])
Fh = []
Fh.append( 5 - (5*Ta[0]/Twant) ) #kW

I = np.array([[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]])

S = (np.dot(A,A))/2
F = I + A + S

k1 = lambda r: (I[0,0] + np.dot(A[0,0],r) + np.dot(S[0,0],r**2))*B[0,0]

k2 = lambda r: (I[0,1] + np.dot(A[0,1],r) + np.dot(S[0,1],r**2))*B[1,1]

k3 = lambda r: (I[1,0] + np.dot(A[1,0],r) + np.dot(S[1,0],r**2))*B[0,0]

k4 = lambda r: (I[1,1] + np.dot(A[1,1],r) + np.dot(S[1,1],r**2))*B[1,1]

k5 = lambda r: (I[2,0] + np.dot(A[2,0],r) + np.dot(S[2,0],r**2))*B[0,0]

k6 = lambda r: (I[2,1] + np.dot(A[2,1],r) + np.dot(S[2,1],r**2))*B[1,1]

k7 = lambda r: (I[3,0] + np.dot(A[3,0],r) + np.dot(S[3,0],r**2))*B[0,0]

k8 = lambda r: (I[3,1] + np.dot(A[3,1],r) + np.dot(S[3,1],r**2))*B[1,1]

C1 = quad(k1,0,1)
C2 = quad(k2,0,1)
C3 = quad(k3,0,1)
C4 = quad(k4,0,1) #integration from 0 to 1min (exp(A*r)*B ) dr
C5 = quad(k5,0,1)
C6 = quad(k6,0,1)
C7 = quad(k7,0,1)
C8 = quad(k8,0,1)
C = np.array([[C1[0], C2[0]],[C3[0], C4[0]], [C5[0], C6[0]], [C7[0], C8
[0]]])

t = [0]
for i in range (1,1001):
    t.append(i) #t=1,2,3,..,1000min!!!
    if (i % 10) == 1 or i==1:
        if error[i-1] > 1: #Twant>Ti+1 then open ac
            AC.append(Twant)
            Fh.append( 5 - (5*Ta[0]/Twant))

        elif error[i-1] < 1 and error[i-1] > -1:
            old = AC[i-1]
            AC.append(old)
            old = Fh[i-1]
            Fh.append(old)

        else: #close ac
            AC.append(0)
            Fh.append(0)
```

```
87.
88.     #solutions to the differential equations
89.     Te.append((F[0,0]*Te[i-1]) + (F[0,1]*Th[i-1]) + (F[0,2]*Ts[i-
   1]) + (F[0,3]*Ti[i-1]) + (C[0,0]*Ta[i-1]) + (C[0,1]*Fh[i-1]))
90.     Th.append((F[1,0]*Te[i-1]) + (F[1,1]*Th[i-1]) + (F[1,2]*Ts[i-
   1]) + (F[1,3]*Ti[i-1]) + (C[1,0]*Ta[i-1]) + (C[1,1]*Fh[i-1]))
91.     Ts.append((F[2,0]*Te[i-1]) + (F[2,1]*Th[i-1]) + (F[2,2]*Ts[i-
   1]) + (F[2,3]*Ti[i-1]) + (C[2,0]*Ta[i-1]) + (C[2,1]*Fh[i-1]))
92.     Ti.append((F[3,0]*Te[i-1]) + (F[3,1]*Th[i-1]) + (F[3,2]*Ts[i-
   1]) + (F[3,3]*Ti[i-1]) + (C[3,0]*Ta[i-1]) + (C[3,1]*Fh[i-1]))
93.     error.append( Twant - Ti[i]) #new error
94.     if i!=1 and ((i % 10) != 1):
95.         old = AC[i-1]
96.         AC.append(old)
97.         old = Fh[i-1]
98.         Fh.append(old)
99.     old = Ta[i-1] #to be able to plot outside temp
100.        Ta.append(old)
101.
102.    time=[]
103.    for x in t:
104.        time.append(x/60.0)
105.
106.    #plot
107.    lc = plt.plot(time,Ti,'b-',label = "Inside Temperature")
108.    la = plt.plot(time,error,'r-', label= "Error")
109.    lb = plt.step(time,AC,'g-', label="AirConditioner")
110.    ld = plt.plot(time,Ta,'m-', label="Outside Temperature")
111.    lf = plt.plot(time,Fh,'y-',label="Power")
112.    lg = plt.plot(time,Th,'k-', label = "Heaters Temperature")
113.    ls = plt.plot(time,Te,'c-', label = "Envelop Temperature")
114.    #lm = plt.plot(time,Ts, 'k-', label ="Sensor Temperature")
115.    ll = plt.title('ON-OFF (TiTeThTs model)')
116.    lx = plt.xlabel('Time(h)')
117.    ly = plt.ylabel('Temperature(Celsius)')
118.    plt.legend(loc=3, bbox_to_anchor = (0, -
   0.4, 1., 1.102),mode="expand",ncol=2)
119.    plt.savefig('./TiTeThTs_ON-OFF_delay.png',bbox_inches='tight')
120.    plt.show()
```

30. TiTeThTs_pid.py

```
1.  #!/usr/bin/python
2.
3.  import math
4.  import matplotlib.pyplot as plt
5.  import numpy as np
6.  import time
7.  import datetime
8.  from scipy.integrate import quad
9.
10. #simulation of pid controller
11. def update(kp,ki,kd,period,windup):
12.     #constants of model
13.     Ta = [0.0]
14.     Twant =  20.0 #celsius
15.     Ti = [15]
```

```
    Th = [15]
    Te = [15]
    Ts = [15]
    Fh = []
    Fh.append( 5 - (5*Ta[0]/Twant)) #kW
    Ci = 0.143 *60.0 #kWmin/oC
    Ch = 0.321 *60.0
    Ce = 3.24 * 60.0
    Cs = 0.619 * 60.0
    Rih = 0.383
    Rie = 0.909
    Rea = 4.47
    Ris = 0.115
    A1 = (-Rie-Rea)/(Rie*Rea*Ce)
    A2 = 0.0
    A3 = 0.0
    A4 = 1.0/(Rie*Ce)
    A5 = 0.0
    A6 = -1.0/(Rih*Ch)
    A7 = 0.0
    A8 = 1.0/(Rih*Ch)
    A9 = 0.0
    A10 = 0.0
    A11 = -1.0/(Ris*Cs)
    A12 = 1.0/(Ris*Cs)
    A13 = 1.0/(Rie*Ci)
    A14 = 1.0/(Rih*Ci)
    A15 = 1.0/(Ris*Ci)
    A16 = (-(Rih*Ris)-(Rie*Ris) - (Rih*Rie))/(Rie*Rih*Ris*Ci)
    A = np.array([[A1, A2, A3, A4], [A5, A6, A7, A8], [A9, A10, A11, A1
2], [A13, A14, A15, A16]])
    B = np.array([[1.0/(Rea*Ce), 0.0], [0.0, 1.0/Ch], [0.0, 0.0], [0.0,
 0.0]])

    AC = [0]
    w = Twant
    x = Ti[0]
    t = [0]
    error = []
    error.append(0)

    I = np.array([[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1
]])
    S = (np.dot(A,A))/2
    F = I + A + S

    k1 = lambda r: (I[0,0] + np.dot(A[0,0],r) + np.dot(S[0,0],r**2))*B[
0,0]
    k2 = lambda r: (I[0,1] + np.dot(A[0,1],r) + np.dot(S[0,1],r**2))*B[
1,1]
    k3 = lambda r: (I[1,0] + np.dot(A[1,0],r) + np.dot(S[1,0],r**2))*B[
0,0]
    k4 = lambda r: (I[1,1] + np.dot(A[1,1],r) + np.dot(S[1,1],r**2))*B[
1,1]
    k5 = lambda r: (I[2,0] + np.dot(A[2,0],r) + np.dot(S[2,0],r**2))*B[
0,0]
    k6 = lambda r: (I[2,1] + np.dot(A[2,1],r) + np.dot(S[2,1],r**2))*B[
1,1]
```

```
    k7 = lambda r: (I[3,0] + np.dot(A[3,0],r) + np.dot(S[3,0],r**2))*B[
0,0]
        k8 = lambda r: (I[3,1] + np.dot(A[3,1],r) + np.dot(S[3,1],r**2)
)*B[1,1]
    C1 = quad(k1,0,1)
    C2 = quad(k2,0,1)
    C3 = quad(k3,0,1)
    C4 = quad(k4,0,1) #integration from 0 to 1min (exp(A*r)*B ) dr
    C5 = quad(k5,0,1)
    C6 = quad(k6,0,1)
    C7 = quad(k7,0,1)
    C8 = quad(k8,0,1)
    C = np.array([[C1[0], C2[0]],[C3[0], C4[0]], [C5[0], C6[0]], [C7[0]
, C8[0]]])

    last_error = 0
    Iy = 0
    period_min = period / 60
    #calculate the y ratio for this 10min period
    for i in range(1,81):
        x = Ti [(i*10) -9 -1]
        e = w - x # error
        de = e - last_error
        dt = 600 #sec
        if dt >=period :
            if e>-2 and e<2:
                Py = (kp * e) + 0.5
                dt = dt / 60 #dt must be in minutes
                Iy += e*dt
                if Iy < -windup:
                    Iy = -windup
                elif Iy > windup:
                    Iy = windup
                Iy = Iy * ki
                if dt > 0 :
                    Dy = (de / dt) * kd
                y = Py + Iy + Dy
                if y > 1:
                        y = 1
                    elif y < 0:
                        y = 0
                elif e>=2:
                    y = 1
                else:
                    y = 0
                last_error = e

            #sleep for 10min until next evaluation
            timeON = y*period_min #for period_min - timeON = timeO$
                timeON = round(timeON)
            if timeON<=3:
                timeON = 0
            elif timeON>=7:
                timeON = 10
            timeON = int(timeON)
            timeOFF = int(period_min - timeON)

                if timeON!=0:
                for l in range(1,timeON+1):
```

```
                    k = (i*10) - 9 + l - 1
                            Fh.append( 5 - (5*Ta[k -1]/Twant))
                    AC.append(Twant)
                    old = Ta[k -1]
                    Ta.append(old)
                    Te.append((F[0,0]*Te[k-1]) + (F[0,1]*Th[k-
   1]) + (F[0,2]*Ts[k-1]) + (F[0,3]*Ti[k-1]) + (C[0,0]*Ta[k-
   1]) + (C[0,1]*Fh[k-1]))
                        Th.append((F[1,0]*Te[k-1]) + (F[1,1]*Th[k-
   1]) + (F[1,2]*Ts[k-1]) + (F[1,3]*Ti[k-1]) + (C[1,0]*Ta[k-
   1]) + (C[1,1]*Fh[k-1]))
                        Ts.append((F[2,0]*Te[k-1]) + (F[2,1]*Th[k-
   1]) + (F[2,2]*Ts[k-1]) + (F[2,3]*Ti[k-1]) + (C[2,0]*Ta[k-
   1]) + (C[2,1]*Fh[k-1]))
                    tempera = ((F[3,0]*Te[k-1]) + (F[3,1]*Th[k-
   1]) + (F[3,2]*Ts[k-1]) + (F[3,3]*Ti[k-1]) + (C[3,0]*Ta[k-
   1]) + (C[3,1]*Fh[k-1]))
                            Ti.append( tempera)
                    t.append(k)
                    error.append(Twant-tempera)

            if timeOFF!=0:
                    for l in range(1,timeOFF+1):
                    k = (i*10) - 9 + l -1 + timeON
                    Fh.append(0)
                            AC.append(0)
                    old = Ta[k-1]
                    Ta.append(old)
                    Te.append((F[0,0]*Te[k-1]) + (F[0,1]*Th[k-
   1]) + (F[0,2]*Ts[k-1]) + (F[0,3]*Ti[k-1]) + (C[0,0]*Ta[k-
   1]) + (C[0,1]*Fh[k-1]))
                                        Th.append((F[1,0]*Te[k-
   1]) + (F[1,1]*Th[k-1]) + (F[1,2]*Ts[k-1]) + (F[1,3]*Ti[k-
   1]) + (C[1,0]*Ta[k-1]) + (C[1,1]*Fh[k-1]))
                                        Ts.append((F[2,0]*Te[k-
   1]) + (F[2,1]*Th[k-1]) + (F[2,2]*Ts[k-1]) + (F[2,3]*Ti[k-
   1]) + (C[2,0]*Ta[k-1]) + (C[2,1]*Fh[k-1]))
                                        tempera = ((F[3,0]*Te[k-
   1]) + (F[3,1]*Th[k-1]) + (F[3,2]*Ts[k-1]) + (F[3,3]*Ti[k-
   1]) + (C[3,0]*Ta[k-1]) + (C[3,1]*Fh[k-1]))
                                        Ti.append( tempera)
                                        t.append(k)
                    error.append(Twant - tempera)
        time = []
        for k in t:
            time.append(k/60.0)

        #plot
        lc = plt.plot(time,Ti,'b-',label = "Inside Temperature")
        lb = plt.step(time,AC,'g-', label="AirConditioner")
        la = plt.plot(time,Ta,'m-',label="Outside Temperature")
        ld = plt.plot(time,error,'r-',label="Error")
        lf = plt.plot(time,Fh,'y-',label="Power")
        le = plt.plot(time,Th,'k-', label = "Heaters Temperature")
        lg = plt.plot(time,Te,'c-',label="Envelop Temperature")
        #lm = plt.plot(time,Ts,'k-',label="Sensor Temperature")
        ll = plt.title('PID (TiTeThTs model)')
        lx = plt.xlabel('Time(h)')
        ly = plt.ylabel('Temperature(Celsius)')
```

```
164.        plt.legend(loc=3, bbox_to_anchor = (0, -
    0.4, 1., 1.102),mode="expand",ncol=2)
165.        plt.savefig('./TiTeThTs_pid.png',bbox_inches='tight')
166.        plt.show()
167.        return y
168.
169.    #set parameters
170.    kp = 0.25 #Xp = 4 bathmoi
171.    period = 600 # 10 minutes
172.    #Tu = 4minutes
173.    ki = 0.2#(2 * kp) / 4 #0.125
174.    period_minu = period / 60 #in minutes
175.    kd = 0.05#period_minu / 8.0 # 0.25
176.    windup = 20.0
177.    list = []
178.
179.    y = update(kp,ki,kd,period,windup)
```

31. smart_table.py

```
1.  #!/usr/bin/python
2.
3.  import MySQLdb
4.  import numpy as np
5.
6.  def start_point():
7.      #fetch the timetable of today
8.      connection = MySQLdb.connect(host = "localhost", user = "project",
    passwd="1111", db="project")
9.      cur = connection.cursor()
10.     days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Sa
    turday', 'Sunday']
11.     #find max number of rows
12.     num = 0
13.     for day in days:
14.         num2 = 0
15.         cur.execute("SELECT * FROM %s" %day)
16.         for row in cur.fetchall():
17.                     num2 += 1
18.         if num2>num:
19.             num = num2 #num contains max rows
20.     #make data and smart at the right(max) size
21.     data = []
22.     smart = []
23.     for i in range(num):
24.         data.append(str(0))
25.         smart.append(0)
26.     data = np.asarray(data)
27.     smart = np.asarray(smart)
28.
29.     for day in days:
30.         cur.execute("SELECT * FROM %s" %day)
31.         num_rows = 0
32.         row_0 =[]
33.         for row in cur.fetchall():
34.             num_rows += 1
35.         #if num_rows < num: #calculate extra zeros for padding
```

```
36.         difference = num - num_rows
37.
38.         cur.execute("SELECT * FROM %s" %day)
39.         for row in cur.fetchall():
40.             help = str(row[0])[-8:-
    7] #add 0 if needed in time to be in correct format
41.             if help!= "1" and help !="2":
42.                 row_0.append(str(0) + str(row[0]))
43.             else:
44.                 row_0.append(str(row[0]))
45.         for d in range(difference): #make the padding if needed
46.             row_0.append(str(0))
47.
48.         smart_1 = np.asarray(row_0)
49.         smart = np.vstack((smart,smart_1,data,data)) #combine times and
     data
50.     smart = np.delete(smart,0, axis=0)
51.     connection.close()
52.     return smart
```

32. smart_compare.py

```
1.  #!/usr/bin/python
2.
3.  import MySQLdb
4.  import numpy as np
5.  import time
6.
7.  def compare(smart,numday,day):
8.          #values from smart table
9.          values = []
10.         columns = smart.shape[1]
11.         for c in range(columns):
12.                 values.append(smart[numday][c])
13.
14.     #time.sleep(30)
15.         #values from schedule
16.         connection = MySQLdb.connect(host = "localhost", user = "projec
    t", passwd="1111", db="project")
17.         cur = connection.cursor()
18.         days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
     'Saturday', 'Sunday']
19.         #find max number of rows
20.         num = 0
21.         for dayss in days:
22.                 num2 = 0
23.                 cur.execute("SELECT * FROM %s" %dayss)
24.                 for row in cur.fetchall():
25.                         num2 += 1
26.                 if num2>num:
27.                         num = num2 #num contains max rows
28.         #make data and smart at the right(max) size
29.     data = []
30.         smart = []
31.         for i in range(num):
32.                 data.append(0)
33.                 smart.append(0)
```

```
34.         data = np.asarray(data)
35.         smart = np.asarray(smart)
36.
37.         cur.execute("SELECT * FROM %s" %day)
38.         num_rows = 0
39.         row_0 =[]
40.         for row in cur.fetchall():
41.                 num_rows += 1
42.         #if num_rows < num: #calculate extra zeros for padding
43.         difference = num - num_rows
44.
45.     cur.execute("SELECT * FROM %s" %day)
46.         for row in cur.fetchall():
47.                 help = str(row[0])[-8:-
    7] #add 0 if needed in time to be in correct format
48.                 if help!= "1" and help !="2":
49.                         row_0.append(str(0) + str(row[0]))
50.                 else:
51.                         row_0.append(str(row[0]))
52.         for d in range(difference): #make the padding if needed
53.                 row_0.append(str(0))
54.
55.         values.sort()
56.         row_0.sort()
57.         if values==row_0:
58.                 same = 1 #same
59.         else:
60.                 same = 0
61.         return same
```

33. smart_update.py

```
1.  #!/usr/bin/python
2.
3.  import MySQLdb
4.  import numpy as np
5.
6.  def update(smart_old,numday,day):
7.
8.          #values from programme
9.          connection = MySQLdb.connect(host = "localhost", user = "projec
    t", passwd="1111", db="project")
10.         cur = connection.cursor()
11.         days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
     'Saturday', 'Sunday']
12.
13.         #find max number of rows
14.         num = 0
15.         for dayss in days:
16.                 num2 = 0
17.                 cur.execute("SELECT * FROM %s" %dayss)
18.                 for row in cur.fetchall():
19.                         num2 += 1
20.                 if num2>num:
21.                         num = num2 #num contains max rows
22.         #make data and smart at the right(max) size
23.         data = []
```

```
        smart = []
        for i in range(num):
                data.append(0)
        smart.append(0)
        data = np.asarray(data)
        smart = np.asarray(smart)

    numday = 0
    for day in days:
           #values from smart table
            values = []
            columns = smart_old.shape[1]
            for c in range(columns):
                    values.append(smart_old[numday][c])
        numday += 3

        #values from schedule
            cur.execute("SELECT * FROM %s" %day)
            num_rows = 0
            row_0 =[]
            for row in cur.fetchall():
                    num_rows += 1
            #if num_rows < num: #calculate extra zeros for padding
            difference = num - num_rows

            cur.execute("SELECT * FROM %s" %day)
            for row in cur.fetchall():
                    help = str(row[0])[-8:-7] #add 0 if needed in t$
                    if help!= "1" and help !="2":
                row_0.append(str(0) + str(row[0]))
                    else:
                            row_0.append(str(row[0]))
            for d in range(difference): #make the padding if needed
                    row_0.append(str(0))

        #the new row of smart with times is row_0
        smart_1 = np.asarray(row_0)
        #values and row_0 have the data to compare
        data_new = []
        errors = []
        for i in range(len(row_0)):
            if row_0[i] != str(0): #it might has a value from smart_old

                if row_0[i] in values: #the element has a value from be
    fore
                    for s in range(len(values)): #find the pointer
                        if row_0[i] == values[s]:
                            data_new.append(smart_old[numday-2][s])
                            errors.append(smart_old[numday-1][s])
                else: #new element
                    data_new.append(str(0))
                    errors.append(str(0))
            else:
                data_new.append(str(0))
                errors.append(str(0))

        data_1 = np.asarray(data_new)
        errors_1 = np.asarray(errors)
        smart = np.vstack((smart,smart_1,data_1,errors_1))
```

```
81.         smart = np.delete(smart,0, axis=0)
82.     connection.close()
83.
84.     return(smart)
```

34. smart_thermostat.py

```
1.  #!/usr/bin/python
2.
3.  import MySQLdb
4.  import math
5.  import numpy as np
6.  import time
7.  from scipy.integrate import quad
8.  import csv
9.
10. #get the set point value (w) = program
11. #we get it from start and it is next_temperature
12.
13. def outsidetemp(idloop,tenmin,m):
14.     if idloop < 16:
15.         toadd = 151216 + idloop
16.     else:
17.         toadd = 160101-16+idloop
18.     path = '/home/pi/Desktop/Measurements/trend_data_1_20'
19.     path = path+str(toadd)+".csv"
20.     with open(path, 'rb') as csvfile:
21.         reader = csv.reader(csvfile, delimiter=';')
22.         reader = list(reader)
23.         res = reader[12+(tenmin*40)+(m*4)][9][-4:]
24.     return res
25.
26. def thermostat(Ta1,Ti1,Th1,Te1,Ts1,w,prev,before,minus,tenmin,idloop):

27.     #values from previus 10min season
28.     #Ta = [Ta1]
29.     out = outsidetemp(idloop,tenmin,0)
30.     out = float(out)
31.     Ta = [out]
32.     Ti = [Ti1]
33.     Th = [Th1]
34.     Te = [Te1]
35.     Ts = [Ts1]
36.     error = []
37.     Fh = []
38.     if w>10 and prev ==0:
39.         error.append( w - Ti[0])
40.         Twant = [w]
41.     elif w<10 and prev ==0:
42.         error.append(0.0)
43.         Twant = [0]
44.     elif w>10 and prev ==1:
45.         error.append(0.0)
46.         Twant =[0]
47.     elif w<10 and prev==1:
48.         error.append(0.0)
49.                 Twant =[0]
50.
51.     if error[0] > 1:
52.                 Fh.append(7)
```

```
53.         elif error[0] < 1 and error[0] > -1:
54.                 old = before
55.                 Fh.append(old)
56.         else: #close ac
57.                 Fh.append(0)
58.
59.
60.     #constants
61.     Ci = 0.143 *60.0 #kWmin/oC
62.     Ch = 0.321 *60.0
63.     Ce = 3.24 * 60.0
64.     Cs = 0.619 * 60.0
65.     Rih = 0.383
66.     Rie = 0.909
67.     Rea = 4.47
68.     Ris = 0.115
69.     A1 = (-Rie-Rea)/(Rie*Rea*Ce)
70.     A2 = 0.0
71.     A3 = 0.0
72.     A4 = 1.0/(Rie*Ce)
73.     A5 = 0.0
74.     A6 = -1.0/(Rih*Ch)
75.     A7 = 0.0
76.     A8 = 1.0/(Rih*Ch)
77.     A9 = 0.0
78.     A10 = 0.0
79.     A11 = -1.0/(Ris*Cs)
80.     A12 = 1.0/(Ris*Cs)
81.     A13 = 1.0/(Rie*Ci)
82.     A14 = 1.0/(Rih*Ci)
83.     A15 = 1.0/(Ris*Ci)
84.     A16 = (-(Rih*Ris)-(Rie*Ris) - (Rih*Rie))/(Rie*Rih*Ris*Ci)
85.     A = np.array([[A1, A2, A3, A4], [A5, A6, A7, A8], [A9, A10, A11, A1
    2], [A13, A14, A15, A16]])
86.     B = np.array([[1.0/(Rea*Ce), 0.0], [0.0, 1.0/Ch], [0.0, 0.0], [0.0,
     0.0]])
87.
88.     I = np.array([[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1
    ]])
89.     S = (np.dot(A,A))/2
90.     F = I + A + S
91.
92.     k1 = lambda r: (I[0,0] + np.dot(A[0,0],r) + np.dot(S[0,0],r**2))*B[
    0,0]
93.     k2 = lambda r: (I[0,1] + np.dot(A[0,1],r) + np.dot(S[0,1],r**2))*B[
    1,1]
94.     k3 = lambda r: (I[1,0] + np.dot(A[1,0],r) + np.dot(S[1,0],r**2))*B[
    0,0]
95.     k4 = lambda r: (I[1,1] + np.dot(A[1,1],r) + np.dot(S[1,1],r**2))*B[
    1,1]
96.     k5 = lambda r: (I[2,0] + np.dot(A[2,0],r) + np.dot(S[2,0],r**2))*B[
    0,0]
97.     k6 = lambda r: (I[2,1] + np.dot(A[2,1],r) + np.dot(S[2,1],r**2))*B[
    1,1]
98.     k7 = lambda r: (I[3,0] + np.dot(A[3,0],r) + np.dot(S[3,0],r**2))*B[
    0,0]
99.     k8 = lambda r: (I[3,1] + np.dot(A[3,1],r) + np.dot(S[3,1],r**2))*B[
    1,1]
100.        C1 = quad(k1,0,1)
```

```
        C2 = quad(k2,0,1)
        C3 = quad(k3,0,1)
        C4 = quad(k4,0,1) #integration from 0 to 1min (exp(A*r)*B ) dr

        C5 = quad(k5,0,1)
        C6 = quad(k6,0,1)
        C7 = quad(k7,0,1)
        C8 = quad(k8,0,1)
        C = np.array([[C1[0], C2[0]],[C3[0], C4[0]], [C5[0], C6[0]], [C
    7[0], C8[0]]])


        #run for 10 minutes
        for i in range (1,10): #i = 1,2,3,..9
            if i==1:
                if w>10:
                    if error[i-
    1] > 1 or prev==1: #Twant>Ti+1 then open ac
                        Fh.append(7)
                    elif error[i-1] < 1 and error[i-1] > -1:
                        old = Fh[i-1]
                        Fh.append(old)
                    else: #close ac
                        Fh.append(0)
                else:
                    Fh.append(0)
            #solutions to the differential equations
            Te.append((F[0,0]*Te[i-1]) + (F[0,1]*Th[i-
    1]) + (F[0,2]*Ts[i-1]) + (F[0,3]*Ti[i-1]) + (C[0,0]*Ta[i-
    1]) + (C[0,1]*Fh[i-1]))
            Th.append((F[1,0]*Te[i-1]) + (F[1,1]*Th[i-
    1]) + (F[1,2]*Ts[i-1]) + (F[1,3]*Ti[i-1]) + (C[1,0]*Ta[i-
    1]) + (C[1,1]*Fh[i-1]))
            Ts.append((F[2,0]*Te[i-1]) + (F[2,1]*Th[i-
    1]) + (F[2,2]*Ts[i-1]) + (F[2,3]*Ti[i-1]) + (C[2,0]*Ta[i-
    1]) + (C[2,1]*Fh[i-1]))
            Ti.append((F[3,0]*Te[i-1]) + (F[3,1]*Th[i-
    1]) + (F[3,2]*Ts[i-1]) + (F[3,3]*Ti[i-1]) + (C[3,0]*Ta[i-
    1]) + (C[3,1]*Fh[i-1]))
            if w>10 and prev==0:
                error.append( w - Ti[i]) #new error
                Twant.append(w)
            elif w<10 and prev==0:
                error.append(0.0)
                Twant.append(0.0)
            elif w>10 and prev==1:
                error.append(0.0)
                Twant.append(0.0)
            else:
                error.append(0.0)
                Twant.append(0.0)
            if i!=1:
                old = Fh[i-1]
                Fh.append(old)
                #Ta.append(0.0)
            out = outsidetemp(idloop,tenmin,i)
                out = float(out)
                Ta.append(out)

```

```
149.        if prev ==1:
150.            for m in range(len(Ti)):
151.                if Ti[m]>w:
152.                    minus = 2
153.
154.        tenmin += 1
155.        return(Ta,Ti,Th,Te,Ts,error,Fh,Twant,minus,tenmin)
```

35. smart.py

```
1.  #!/usr/bin/python
2.
3.  from smart_table import start_point
4.  from smart_compare import compare
5.  from smart_update import update
6.  from smart_thermostat import thermostat
7.  import matplotlib
8.  matplotlib.use('Agg')
9.  import matplotlib.pyplot as plt
10. import numpy as np
11. import MySQLdb
12. import time
13. from datetime import datetime
14.
15. #start values for thermostat
16. Ta = [12.0]
17. Ti = [18.0]
18. Th = [18.0]
19. Te = [13.0]
20. Ts = [18.0]
21. error = [0.0]
22. Fh = [0.0]
23. Twant = [1.0]
24. prev = 0
25. firsthere=0
26. idloop = 0
27.
28. smart = start_point() #the first smart table
29. days = ['Monday' , 'Tuesday' , 'Wednesday', 'Thursday', 'Friday', 'Satu
    rday', 'Sunday']
30. onlyatfirst =0
31.
32. for week in range(0,3): #for 3 weeks week=0,1,2
33.     numday = 0 #num 0 is monday
34.     for day in days: #for each day of the week
35.         same = compare(smart,numday,day) #check for the day program and
     compare to smart
36.         if same == 0: #there are not same (someone made a change at the
     schedule)
37.             smart = update(smart,numday,day) #call to update the smart
    table
38.         numday += 3 #for the next iteration (needed for function compar
    e)
39.         idloop += 1
40.         tenmin = 0
41.         #now is sure that the smart table is ok so flow chart starts
```

```
        for i in range (0,144): #144 periods of 10min in one day(midnight is next day)
            # STEP 1 check if now there is a program going on
            #the time now = (i*10)/60
            hours = i / 6
                        minutes = (i % 6)*10
                        if hours >=0 and hours <=9:
                            if minutes ==0:
                                    timen = '0%s:00:00' %hours
                                else:
                                    timen = '0%s:%s:00' % (hours, minutes)
                        else:
                                if minutes==0:
                                        timen = '%s:00:00' %hours
                                else:
                                        timen = '%s:%s:00' %(hours, minutes)
            schedule = 0
            print idloop,timen
            connection = MySQLdb.connect(host = "localhost", user = "project", passwd="1111", db="project")
            cur = connection.cursor()
            cur.execute("SELECT * FROM %s" %day)
            for row in cur.fetchall():
                        help = str(row[0])[-8:-7]
                        if help != "1" and help != "2": #insert 0 at the start of time -> 24hour format
                                row_0 = str(0) + str(row[0])
                        else:
                                row_0 = str(row[0])
                        help = str(row[1])[-8:-7]
                        if help != "1" and help != "2": #insert 0 at the start of time -> 24hour format
                                row_1 = str(0) + str(row[1])
                        else:
                                row_1 = str(row[1])
                #check if there is a schedule now
                    if row_0 <= timen and timen < row_1:
                                schedule = 1
                    next_temperature = row[2]
            connection.close()

            if schedule == 1: #now there is schedule so START HEATER
                #print "TYPE A, time=%s" %timen
                #call thermostat
                Ta1 = []
                Ti1 = []
                Th1 = []
                Te1 = []
                Ts1 = []
                error1 =[]
                Fh1 = []
                Twant1 = []
                w = next_temperature
                prev = 0
                minus = 0
```

```
93.                 Ta1, Ti1, Th1, Te1, Ts1, error1, Fh1, Twant1,minus,tenm
    in = thermostat(Ta[-1],Ti[-1],Th[-1],Te[-1],Ts[-1],w,prev,Fh[-
    1],minus,tenmin,idloop)
94.                 Ta = Ta + Ta1
95.                 Ti = Ti + Ti1
96.                 Th = Th + Th1
97.                 Te = Te + Te1
98.                 Ts = Ts + Ts1
99.                 error = error + error1
100.                    Fh = Fh + Fh1
101.                    Twant = Twant + Twant1
102.                    if float(Twant[-11]) != float(Twant[-
    1]) and float(Twant[-11]) <= 10: #first time we run this schedule
103.                        timestart = datetime.strptime(timen, '%H:%M:%S'
    ) #time that the schedule started
104.                        Tempstart = Ti[-11]
105.                        onlyatfirst = 1
106.                        if Ti[-11] > Twant[-
    1]+1: #error must be negative
107.                            onlyatfirst =2
108.                    if Ti[-1]>=Twant[-1]-
    1 and onlyatfirst==1: #we can calculate smart time
109.                        onlyatfirst=0
110.                        forlast=1
111.                        timestop = datetime.strptime(timen, '%H:%M:%S')

112.                        smartdif = str(timestop - timestart)
113.                                            help = smartdif[-8:-7]
114.                                            if help != "1" and help !=
    "2": #insert 0 at the start of time -> 24hour format
115.                                                smartdif = str(0) + str
    (smartdif)
116.                                            else:
117.                                                    smartdif = str(smar
    tdif)
118.                        #calculate how much minutes smartdif is
119.                        timeadd = 0
120.                        if int(smartdif[:-6]) != 0:
121.                            timeadd += int(smartdif[:-6]) * 60
122.                        if int(smartdif[-5:-3]) !=0:
123.                            timeadd += int(smartdif[-5:-3])
124.                        #update error on smart table
125.                                            test = numday-1
126.                                            test1 = numday-2
127.                        test2 = numday -3
128.                                            columns = smart.shape[1]
129.                        for s in range(columns):
130.                                if day=='Sunday' and smart[18][s] == st
    r(timestart)[-8:]:
131.                                help = int(smart[19][s]) + int(smart[20
    ][s])
132.                                smart[19][s] = str(help)
133.                                smart[20][s] = str(timeadd)
134.                                elif smart[test2][s] ==str(timestart)[-
    8:]:
135.                                help = int(smart[test1][s]) + int(smart
    [test][s])
136.                                smart[test1][s] = str(help)
137.                                smart[test][s] = str(timeadd)
```

```
138.
139.                 # STEP 2 find when the next schedule will be
140.                 else: #schedule ==0 so no programme now
141.                     T = '04:00:00'
142.                     timenn = datetime.strptime(timen, '%H:%M:%S')
143.                     connection = MySQLdb.connect(host = "localhost", us
    er = "project", passwd="1111", db="project")
144.                                 cur = connection.cursor()
145.                                 cur.execute("SELECT * FROM %s" %day)
146.                                 for row in cur.fetchall():
147.                                         help = str(row[0])[-8:-7]
148.                                         if help != "1" and help != "2":
     #insert 0 at the start of time -> 24hour format
149.                                         row_0 = str(0) + str(row[0])
150.                                         else:
151.                                                 row_0 = str(row[0])
152.                         row_00 = datetime.strptime(row_0, '%H:%M:%S')
153.                         if row_00 > timenn:
154.                             dif = str(row_00 - timenn)
155.                             help = dif[-8:-7]
156.                                             if help != "1" and help !=
    "2": #insert 0 at the start of time -> 24hour format
157.                                                 dif = str(0) + str(
    dif)
158.                                             else:
159.                                                 dif = str(dif)
160.                             if dif <= T:
161.                                 T = dif
162.                                 next_schedule = str(row_00)
163.                                 next_schedule = next_schedule[-8:]
164.                                 next_temperature = row[2]
165.                                 flag =0
166.                     connection.close()
167.                     #check if the next schedule comes in the next day
168.                     if str(timen) >= '22:00:00' and T > '02:00:00':
169.                         if day == 'Monday':
170.                             next_day = 'Tuesday'
171.                         if day == 'Tuesday':
172.                             next_day = 'Wednesday'
173.                         if day == 'Wednesday':
174.                             next_day = 'Thursday'
175.                         if day == 'Thursday':
176.                             next_day = 'Friday'
177.                         if day == 'Friday':
178.                             next_day = 'Saturday'
179.                         if day == 'Saturday':
180.                             next_day = 'Sunday'
181.                         if day == 'Sunday':
182.                             next_day = 'Monday'
183.                         connection = MySQLdb.connect(host = "localhost"
    , user = "project", passwd="1111", db="project")
184.                         cur = connection.cursor()
185.                         cur.execute("SELECT * FROM %s" %next_day)
186.                         for row in cur.fetchall():
187.                              help = str(row[0])[-8:-7]
188.                              if help != "1" and help != "2": #insert 0
    at the start of time -> 24hour format
189.                                     row_0 = str(0) + str(row[0])
190.                             else:
```

```
                                    row_0 = str(row[0])
                                    if row_0 <= '02:00:00':
       row_00 = datetime.strptime(row_0, '%H:%M:%S')
                                dif = str( row_00 - timenn)
                                dif = dif[-7:]
                                dif = str(0) + str(dif)
                                if dif <= T:
                                    T = dif
                                    next_schedule = str(row_00)
                                    next_schedule = next_schedule[-
   8:]
                                    next_temperature = row[2]
                                    flag =1
                                    connection.close()

                        # STEP 3 check if T <= 2
                        if T > '02:00:00': #no heaters
                            #print "TYPE B, time=%s" %timen
                            next_temperature = 0
                            #call thermostat
                            Ta1 = []
                            Ti1 = []
                            Th1 = []
                            Te1 = []
                            Ts1 = []
                            error1 =[]
                            Fh1 = []
                            Twant1 = []
                            w = next_temperature
                            prev = 0
                            minus = 0
                            Ta1, Ti1, Th1, Te1, Ts1, error1, Fh1, Twant1,mi
   nus,tenmin = thermostat(Ta[-1],Ti[-1],Th[-1],Te[-1],Ts[-1],w,prev,Fh[-
   1],minus,tenmin,idloop)
                            Ta = Ta + Ta1
                            Ti = Ti + Ti1
                            Th = Th + Th1
                            Te = Te + Te1
                            Ts = Ts + Ts1
                            error = error + error1
                            Fh = Fh + Fh1
                            Twant = Twant + Twant1

                        else: #T <= '02:00:00' so check if heaters must go
   on
                            # STEP 4 find how much time before it has to st
   art
                            if day=='Sunday':
                                numday= 18
                            test = numday-1
                            test1 = numday-2
                            test2 = numday -3
                            columns = smart.shape[1]
                            for s in range(columns):
                                #s = int(s)-int(1)
                                if day=='Sunday' and smart[18][s] == next_s
   chedule and flag==0:
                                    if smart[20][s][:1]=="-":
```

```
                                    Tpro = int(smart[19][s]) -
    int(smart[20][s][1:])
                                else:
                                    Tpro = int(smart[19][s]) + int(smar
    t[20][s])
                                smart[19][s] = str(Tpro)
                                smart[20][s] = str(0)
                                #print "sunday"
                            elif smart[test2][s] == next_schedule and f
    lag==0:
                                if smart[test][s][:1]=="-":
                                    Tpro = int(smart[test1][s]) -
    int(smart[test][s][1:])
                                else:
                                    Tpro = int(smart[test1][s]) + int(s
    mart[test][s])
                                    smart[test1][s] = str(Tpro)
                                    smart[test][s] = str(0)
                            elif day=='Sunday' and flag==1 and smart[0]
    [s] == next_schedule:
                                if smart[2][s][:1]=="-":
                                    Tpro = int(smart[1][s]) -
    int(smart[2][s][1:])
                                else:
                                    Tpro = int(smart[1][s]) + int(smart
    [2][s])
                                    smart[1][s] = str(Tpro)
                                    smart[2][s] = str(0)
                            elif smart[numday][s] == next_schedule and
    flag==1: #other days with flag==1
                                test = numday +1
                                test1 = numday +2
                                if smart[test1][s][:1]=="-":
                                    Tpro = int(smart[test][s]) -
    int(smart[test1][s][1:])
                                else:
                                    Tpro = int(smart[test][s]) + int(sm
    art[test1][s])
                                    smart[test][s] = str(Tpro)
                                    smart[test1][s] = str(0)

        # STEP 5 if the time measure is less than time to next schedule
    do nothing
                        Tpro_hrs = Tpro//int(60)
                        Tpro_mins = Tpro%int(60)
                        Tpro = "%s:%s:00" % (str(Tpro_hrs), str(Tpro_mi
    ns))
                        Tpro = datetime.strptime(Tpro, '%H:%M:%S') #mak
    e minutes in time format
                        Tpro = str(Tpro)[-8:]
                        if Tpro < T or Tpro==0: #no need to open heater
    s yet
                            next_temperature = 0
                            #call thermostat
                            Ta1 = []
                            Ti1 = []
                            Th1 = []
                            Te1 = []
                            Ts1 = []
```

```
                            error1 =[]
                            Fh1 = []
                            Twant1 = []
                            w = next_temperature
                            prev = 0
                            minus = 0
                            Ta1, Ti1, Th1, Te1, Ts1, error1, Fh1, Twant
   1,minus,tenmin = thermostat(Ta[-1],Ti[-1],Th[-1],Te[-1],Ts[-
   1],w,prev,Fh[-1],minus,tenmin,idloop)
                            Ta = Ta + Ta1
                            Ti = Ti + Ti1
                            Th = Th + Th1
                            Te = Te + Te1
                            Ts = Ts + Ts1
                            error = error + error1
                            Fh = Fh + Fh1
                            Twant = Twant + Twant1
                        else: # Tpro >= T so heaters must go on
                            # STEP 6
                            #print "TYPE D, time=%s" %timen
                            Ta1 = []
                            Ti1 = []
                            Th1 = []
                            Te1 = []
                            Ts1 = []
                            error1 =[]
                            Fh1 = []
                            Twant1 = []
                            w = next_temperature
                            prev = 1
                            minus = 1
                            Ta1, Ti1, Th1, Te1, Ts1, error1, Fh1, Twant
   1,minus,tenmin = thermostat(Ta[-1],Ti[-1],Th[-1],Te[-1], Ts[-
   1],w,prev,Fh[-1],minus,tenmin,idloop)
                            Ta = Ta + Ta1
                            Ti = Ti + Ti1
                            Th = Th + Th1
                            Te = Te + Te1
                            Ts = Ts + Ts1
                            error = error + error1
                            Fh = Fh + Fh1
                            Twant = Twant + Twant1
                            if minus == 2 and firsthere==0: #we have ne
   gative error
                                minus = 3
                                timenow = timen
                                firsthere=1
                                #print timenow
                            next_sch = datetime.strptime(next_schedule,
    '%H:%M:%S')
                            timen = datetime.strptime(timen, '%H:%M:%S'
   )
                            smartdif = str(next_sch - timen)[-8:]
                            help = smartdif[-8:-7]
                             if help != "1" and help != "2": #insert 0
   at the start of time -> 24hour form$
                                smartdif = str(0) + str(smartdif)[-
   7:]
                            else:
```

```
                                    smartdif = str(smartdif)
                                if firsthere==1 and smartdif == "00:10:00":
    #last time in this section
                                    minus = 0
                                    firsthere=0
                                    #calculate time to sub
                                    timenow = datetime.strptime(timenow, '%
   H:%M:%S')
                                    smartdif = str(next_sch-timenow)
                                    help = smartdif[-8:-7]
                                       if help != "1" and help != "2": #ins
   ert 0 at the start of time -> 24hour format
                                           smartdif = str(0) + str(smartdif)
   [-7:]
                                       else:
                                           smartdif = str(smartdif)
                                    #calculate how much minutes smartdif is

                                    timesub = 0
                                    if int(smartdif[:-6]) != 0:
                                        timesub += int(smartdif[:-
   6]) * 60
                                    if int(smartdif[-5:-3]) !=0:
                                        timesub += int(smartdif[-5:-3])
                                    print timesub
                                    #update error on table
                                    test = numday-1
                                    test2 = numday -3
                                    columns = smart.shape[1]
                                    for s in range(columns):
                                        if day=='Sunday' and smart[18][s] =
   = next_schedule and flag==0:
                                            help = int(smart[20][s]) -
    timesub
                                            smart[20][s] = str(help) #str(h
   elp)
                                        elif day=='Sunday' and smart[0][s]
   == next_schedule and flag==1:
                                           help = int(smart[2][s]) -
    timesub
                                           smart[2][s] = str(help)
                                    elif smart[test2][s] ==next_schedule an
   d flag==0:
                                            help = int(smart[test][s]) -
    timesub
                                            smart[test][s] = str(help) #str
   (help)
                                    elif smart[numday][s] == next_sched
   ule and flag==1: #other days with flag==1
                                           test1 = numday +2
                                           help = int(smart[test1][s]) -

    timesub
                                           smart[test1][s] = str(help)

   #PLOT
   times= []
   timess = len(Ti)
   for s in range(timess):
       times.append(s/60.0)
```

```
379.
380.    #axes = plt.gca()
381.    #axes.set_xlim([0,170])
382.
383.    #fig,ax = plt.figure
384.    fig, ax = plt.subplots()
385.    ax.set_xlim(0,168)
386.    ax.set_xlabel('time(h)')
387.    ax.set_ylabel('Temperature(oC)')
388.
389.    ax.annotate('Monday',xy=(0,0), xytext=(-5,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
390.    ax.annotate('Tuesday',xy=(24,0), xytext=(19,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
391.    ax.annotate('Wednesday',xy=(48,0), xytext=(43,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
392.    ax.annotate('Thursday',xy=(72,0), xytext=(67,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
393.    ax.annotate('Friday',xy=(96,0), xytext=(91,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
394.    ax.annotate('Saturday',xy=(120,0), xytext=(115,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
395.    ax.annotate('Sunday',xy=(144,0), xytext=(139,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
396.
397.    lc = plt.plot(times,Ti,'b-',label = "Inside Temperature")
398.    la = plt.plot(times,Ta,'m-',label="Outside Temperature")
399.    ln = plt.step(times,Twant,'k-',label="Programme")
400.    ld = plt.plot(times,error,'r-',label="Error")
401.    lf = plt.step(times,Fh,'g-',label="Power of Heaters")
402.    #le = plt.plot(times,Th,'y-', label = "Heaters Temperature")
403.    lg = ax.plot(times,Te,'c-',label="Envelop Temperature")
404.    #lm = plt.plot(time,Ts, 'k-', label ="Sensor Temperature")
405.    ll = plt.title('ON-OFF smart thermostat week 1')
406.    plt.legend(loc=3, bbox_to_anchor = (0.,-
    0.2, 1., 1.102), mode="expand",ncol=3)
407.    figure = plt.gcf()
408.    figure.set_size_inches(16,8)
409.    plt.savefig('./smart_ON-OFF.png',bbox_inches='tight')
410.    plt.show()
411.
412.    fig, ax = plt.subplots()
413.    ax.set_xlim(168,336)
414.    ax.set_xlabel('time(h)')
415.    ax.set_ylabel('Temperature(oC)')
416.
417.    ax.annotate('Monday',xy=(168,0), xytext=(163,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
418.    ax.annotate('Tuesday',xy=(192,0), xytext=(187,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
419.    ax.annotate('Wednesday',xy=(216,0), xytext=(211,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
420.    ax.annotate('Thursday',xy=(240,0), xytext=(235,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
421.    ax.annotate('Friday',xy=(264,0), xytext=(259,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
422.    ax.annotate('Saturday',xy=(288,0), xytext=(283,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
```

```
423.    ax.annotate('Sunday',xy=(312,0), xytext=(307,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
424.
425.
426.    lc = plt.plot(times,Ti,'b-',label = "Inside Temperature")
427.    la = plt.plot(times,Ta,'m-',label="Outside Temperature")
428.    ln = plt.step(times,Twant,'k-',label="Programme")
429.    ld = plt.plot(times,error,'r-',label="Error")
430.    lf = plt.step(times,Fh,'g-',label="Power of Heaters")
431.    #le = plt.plot(times,Th,'y-', label = "Heaters Temperature")
432.    lg = ax.plot(times,Te,'c-',label="Envelop Temperature")
433.    #lm = plt.plot(time,Ts, 'k-', label ="Sensor Temperature")
434.    ll = plt.title('ON-OFF smart thermostat week 2')
435.    plt.legend(loc=3, bbox_to_anchor = (0.,-
    0.2, 1., 1.102), mode="expand",ncol=3)
436.    figure = plt.gcf()
437.    figure.set_size_inches(16,8)
438.    plt.savefig('./smart_ON-OFF2.png',bbox_inches='tight')
439.    plt.show()
440.
441.    fig, ax = plt.subplots()
442.    ax.set_xlim(336,504)
443.    ax.set_xlabel('time(h)')
444.    ax.set_ylabel('Temperature(oC)')
445.
446.    ax.annotate('Monday',xy=(336,0), xytext=(331,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
447.    ax.annotate('Tuesday',xy=(360,0), xytext=(355,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
448.    ax.annotate('Wednesday',xy=(384,0), xytext=(379,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
449.    ax.annotate('Thursday',xy=(408,0), xytext=(403,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
450.    ax.annotate('Friday',xy=(432,0), xytext=(427,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
451.    ax.annotate('Saturday',xy=(456,0), xytext=(451,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
452.    ax.annotate('Sunday',xy=(480,0), xytext=(475,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
453.
454.    lc = plt.plot(times,Ti,'b-',label = "Inside Temperature")
455.    la = plt.plot(times,Ta,'m-',label="Outside Temperature")
456.    ln = plt.step(times,Twant,'k-',label="Programme")
457.    ld = plt.plot(times,error,'r-',label="Error")
458.    lf = plt.step(times,Fh,'g-',label="Power of Heaters")
459.    #le = plt.plot(times,Th,'y-', label = "Heaters Temperature")
460.    lg = ax.plot(times,Te,'c-',label="Envelop Temperature")
461.    #lm = plt.plot(time,Ts, 'k-', label ="Sensor Temperature")
462.    ll = plt.title('ON-OFF smart thermostat week 3')
463.    plt.legend(loc=3, bbox_to_anchor = (0.,-
    0.2, 1., 1.102), mode="expand",ncol=3)
464.    figure = plt.gcf()
465.    figure.set_size_inches(16,8)
466.    plt.savefig('./smart_ON-OFF3.png',bbox_inches='tight')
467.    plt.show()
```

36. smart_thermostat_pid_ext.py

```
1.  #!/usr/bin/python
2.
3.  import MySQLdb
4.  import math
5.  import numpy as np
6.  import time
7.  from scipy.integrate import quad
8.  import csv
9.
10. #get the set point value (w) = program
11. #we get it from start and it is next_temperature
12.
13. def outsidetemp(idloop,tenmin,m):
14.     if idloop < 16:
15.         toadd = 151216 + idloop
16.     else:
17.         toadd = 160101-16+idloop
18.     path = '/home/pi/Desktop/Measurements/trend_data_1_20'
19.     path = path+str(toadd)+".csv"
20.     with open(path, 'rb') as csvfile:
21.         reader = csv.reader(csvfile, delimiter=';')
22.         reader = list(reader)
23.         res = reader[12+(tenmin*40)+(m*4)][9][-4:]
24.     return res
25.
26. def thermostat_pid_ext(Ta1,Ti1,Th1,Te1,Ts1,w,prev,before,minus,tenmin,i
    dloop):
27.     #values from previus 10min season
28.     out = outsidetemp(idloop,tenmin,0)
29.     out = float(out)
30.     Ta = [out]
31.     #Ta = [Ta1]
32.     Ti = [Ti1]
33.     Th = [Th1]
34.     Te = [Te1]
35.     Ts = [Ts1]
36.     error = []
37.     Fh = []
38.     if w>10 and prev ==0:
39.         error.append( w - Ti[0])
40.         Twant = [w]
41.     elif w<10 and prev ==0:
42.         error.append(0.0)
43.         Twant = [0]
44.     elif w>10 and prev ==1:
45.         error.append(0.0)
46.         Twant =[0]
47.     elif w<10 and prev==1:
48.         error.append(0.0)
49.                 Twant =[0]
50.
51.     if error[0] > 0 or prev==1:
52.         Fh.append(7)
53.         else: #close ac
54.                 Fh.append(0)
55.
56.     #parameters of pid
```

```
57.     kp = 0.25
58.     period = 600
59.     ki = 0.2
60.     period_minu = period / 60
61.     kd = 0.05
62.     windup = 20.0
63.
64.     #constants
65.     Ci = 0.143 *60.0 #kWmin/oC
66.     Ch = 0.321 *60.0
67.     Ce = 3.24 * 60.0
68.     Cs = 0.619 * 60.0
69.     Rih = 0.383
70.     Rie = 0.909
71.     Rea = 4.47
72.     Ris = 0.115
73.     A1 = (-Rie-Rea)/(Rie*Rea*Ce)
74.     A2 = 0.0
75.     A3 = 0.0
76.     A4 = 1.0/(Rie*Ce)
77.     A5 = 0.0
78.     A6 = -1.0/(Rih*Ch)
79.     A7 = 0.0
80.     A8 = 1.0/(Rih*Ch)
81.     A9 = 0.0
82.     A10 = 0.0
83.     A11 = -1.0/(Ris*Cs)
84.     A12 = 1.0/(Ris*Cs)
85.     A13 = 1.0/(Rie*Ci)
86.     A14 = 1.0/(Rih*Ci)
87.     A15 = 1.0/(Ris*Ci)
88.     A16 = (-(Rih*Ris)-(Rie*Ris) - (Rih*Rie))/(Rie*Rih*Ris*Ci)
89.     A = np.array([[A1, A2, A3, A4], [A5, A6, A7, A8], [A9, A10, A11, A1
    2], [A13, A14, A15, A16]])
90.     B = np.array([[1.0/(Rea*Ce), 0.0], [0.0, 1.0/Ch], [0.0, 0.0], [0.0,
     0.0]])
91.
92.     I = np.array([[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1
    ]])
93.     S = (np.dot(A,A))/2
94.     F = I + A + S
95.
96.     k1 = lambda r: (I[0,0] + np.dot(A[0,0],r) + np.dot(S[0,0],r**2))*B[
    0,0]
97.     k2 = lambda r: (I[0,1] + np.dot(A[0,1],r) + np.dot(S[0,1],r**2))*B[
    1,1]
98.     k3 = lambda r: (I[1,0] + np.dot(A[1,0],r) + np.dot(S[1,0],r**2))*B[
    0,0]
99.     k4 = lambda r: (I[1,1] + np.dot(A[1,1],r) + np.dot(S[1,1],r**2))*B[
    1,1]
100.        k5 = lambda r: (I[2,0] + np.dot(A[2,0],r) + np.dot(S[2,0],r**2)
    )*B[0,0]
101.        k6 = lambda r: (I[2,1] + np.dot(A[2,1],r) + np.dot(S[2,1],r**2)
    )*B[1,1]
102.        k7 = lambda r: (I[3,0] + np.dot(A[3,0],r) + np.dot(S[3,0],r**2)
    )*B[0,0]
103.        k8 = lambda r: (I[3,1] + np.dot(A[3,1],r) + np.dot(S[3,1],r**2)
    )*B[1,1]
104.        C1 = quad(k1,0,1)
```

```
105.        C2 = quad(k2,0,1)
106.        C3 = quad(k3,0,1)
107.        C4 = quad(k4,0,1) #integration from 0 to 1min (exp(A*r)*B ) dr

108.        C5 = quad(k5,0,1)
109.        C6 = quad(k6,0,1)
110.        C7 = quad(k7,0,1)
111.        C8 = quad(k8,0,1)
112.        C = np.array([[C1[0], C2[0]],[C3[0], C4[0]], [C5[0], C6[0]], [C
    7[0], C8[0]]])
113.
114.        last_error = 0
115.        Iy = 0
116.        period_min = period / 60
117.
118.        #calculate the y ratio for this 10min period
119.        if w>10 or prev==1:
120.            x = Ti[0]
121.            e = w - x # error
122.            de = e - last_error
123.            dt = 600 #sec
124.            if dt >=period :
125.                if e>-2 and e<2:
126.                    Py = (kp * e) + 0.5
127.                    dt = dt / 60 #dt must be in minutes
128.                    Iy += e*dt
129.                    if Iy < -windup:
130.                        Iy = -windup
131.                    elif Iy > windup:
132.                        Iy = windup
133.                    Iy = Iy * ki
134.                    if dt > 0 :
135.                        Dy = (de / dt) * kd
136.                    y = Py + Iy + Dy
137.                    if y > 1:
138.                        y = 1
139.                    elif y < 0:
140.                        y = 0
141.                elif e>=2:
142.                    y = 1
143.                else:
144.                    y = 0
145.                last_error = e
146.        else: #if w=10 then AC must be off
147.            y = 0
148.
149.        #calculate on-off time
150.        timeON = y*period_min #for period_min - timeON = timeO$
151.            timeON = round(timeON)
152.        if timeON<=3:
153.            timeON = 0
154.        elif timeON>=7:
155.            timeON = 10
156.        timeON = int(timeON)
157.        timeOFF = int(period_min - timeON)
158.
159.        if timeON!=0:
160.            for l in range(1,timeON): # l = 1,2,3,..,timeON-1
161.                k = l
```

```
                Fh.append(7)
                Te.append((F[0,0]*Te[k-1]) + (F[0,1]*Th[k-
   1]) + (F[0,2]*Ts[k-1]) + (F[0,3]*Ti[k-1]) + (C[0,0]*Ta[k-
   1]) + (C[0,1]*Fh[k-1]))
                Th.append((F[1,0]*Te[k-1]) + (F[1,1]*Th[k-
   1]) + (F[1,2]*Ts[k-1]) + (F[1,3]*Ti[k-1]) + (C[1,0]*Ta[k-
   1]) + (C[1,1]*Fh[k-1]))
                Ts.append((F[2,0]*Te[k-1]) + (F[2,1]*Th[k-
   1]) + (F[2,2]*Ts[k-1]) + (F[2,3]*Ti[k-1]) + (C[2,0]*Ta[k-
   1]) + (C[2,1]*Fh[k-1]))
                tempera = ((F[3,0]*Te[k-1]) + (F[3,1]*Th[k-
   1]) + (F[3,2]*Ts[k-1]) + (F[3,3]*Ti[k-1]) + (C[3,0]*Ta[k-
   1]) + (C[3,1]*Fh[k-1]))
                        Ti.append( tempera)
                if w>10 and prev==0:
                                error.append( w - tempera) #new error
                                Twant.append(w)
                        elif w<10 and prev==0:
                                error.append(0.0)
                                Twant.append(0.0)
                        elif w>10 and prev==1:
                                error.append(0.0)
                                Twant.append(0.0)
                        else:
                                error.append(0.0)
                                Twant.append(0.0)

                out = outsidetemp(idloop,tenmin,k)
                    out = float(out)
                    Ta.append(out)

                #Ta.append(0.0)

        if timeOFF!=0:
            if timeON==0:
                timeOFF = timeOFF -1
            for l in range(1,timeOFF+1): # l = 1,2,3,..,timeOFF
                k = l  + timeON - 1  # k = timeON + 0,1,2,...timeOFF-
   1
                Fh.append(0.0)
                Te.append((F[0,0]*Te[k-1]) + (F[0,1]*Th[k-
   1]) + (F[0,2]*Ts[k-1]) + (F[0,3]*Ti[k-1]) + (C[0,0]*Ta[k-
   1]) + (C[0,1]*Fh[k-1]))
                            Th.append((F[1,0]*Te[k-1]) + (F[1,1]*Th[k-
   1]) + (F[1,2]*Ts[k-1]) + (F[1,3]*Ti[k-1]) + (C[1,0]*Ta[k-
   1]) + (C[1,1]*Fh[k-1]))
                            Ts.append((F[2,0]*Te[k-1]) + (F[2,1]*Th[k-
   1]) + (F[2,2]*Ts[k-1]) + (F[2,3]*Ti[k-1]) + (C[2,0]*Ta[k-
   1]) + (C[2,1]*Fh[k-1]))
                            tempera = ((F[3,0]*Te[k-1]) + (F[3,1]*Th[k-
   1]) + (F[3,2]*Ts[k-1]) + (F[3,3]*Ti[k-1]) + (C[3,0]*Ta[k-
   1]) + (C[3,1]*Fh[k-1]))
                Ti.append( tempera)
                if w>10 and prev==0:
                    error.append( w - tempera) #new error
                    Twant.append(w)
                elif w<10 and prev==0:
                    error.append(0.0)
                    Twant.append(0.0)
```

```
                elif w>10 and prev==1:
                    error.append(0.0)
                    Twant.append(0.0)
                else:
                    error.append(0.0)
                    Twant.append(0.0)
                #Ta.append(0.0)
                out = outsidetemp(idloop,tenmin,k)
                            out = float(out)
                            Ta.append(out)

        if prev ==1:
            for m in range(len(Ti)):
                if Ti[m]>w:
                    minus = 2
        tenmin += 1
        return(Ta,Ti,Th,Te,Ts,error,Fh,Twant,minus,tenmin)
```

37. smart_pid_ext.py

```
#!/usr/bin/python

from smart_table import start_point
from smart_compare import compare
from smart_update import update
from smart_thermostat_pid_ext import thermostat_pid_ext
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import numpy as np
import MySQLdb
import time
from datetime import datetime

#start values for thermostat
Ta = [12.0]
Ti = [18.0]
Th = [18.0]
Te = [13.0]
Ts = [18.0]
error = [0.0]
Fh = [0.0]
Twant = [1.0]
prev = 0
firsthere=0
idloop = 0

smart = start_point() #the first smart table
days = ['Monday' , 'Tuesday' , 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
onlyatfirst =0

for week in range(0,3): #for 3 weeks week=0,1,2
    numday = 0 #num 0 is monday
    for day in days: #for each day of the week
        same = compare(smart,numday,day) #check for the day program and compare to smart
```

```
        if same == 0: #there are not same (someone made a change at the
 schedule)
            smart = update(smart,numday,day) #call to update the smart
table
        numday += 3 #for the next iteration (needed for function compar
e)
        idloop += 1
        tenmin = 0
        #now is sure that the smart table is ok so flow chart starts
        for i in range (0,144): #144 periods of 10min in one day(midnig
ht is next day)
            # STEP 1 check if now there is a program going on
            #the time now = (i*10)/60
            hours = i / 6
                        minutes = (i % 6)*10
                        if hours >=0 and hours <=9:
                            if minutes ==0:
                                    timen = '0%s:00:00' %hours
                                else:
                                    timen = '0%s:%s:00' % (hours, minut
es)
                        else:
                                if minutes==0:
                                        timen = '%s:00:00' %hours
                                else:
                                        timen = '%s:%s:00' %(hours, min
utes)
            print idloop, timen
            schedule = 0
            connection = MySQLdb.connect(host = "localhost", user = "pr
oject", passwd="1111", db="project")
                cur = connection.cursor()
            cur.execute("SELECT * FROM %s" %day)
            for row in cur.fetchall():
                        help = str(row[0])[-8:-7]
                        if help != "1" and help != "2": #insert 0 at th
e start of time -> 24hour format
                                row_0 = str(0) + str(row[0])
                        else:
                                row_0 = str(row[0])
                        help = str(row[1])[-8:-7]
                        if help != "1" and help != "2": #insert 0 at th
e start of time -> 24hour format
                                row_1 = str(0) + str(row[1])
                        else:
                                row_1 = str(row[1])
                #check if there is a schedule now
                    if row_0 <= timen and timen < row_1:
                                schedule = 1
                        next_temperature = row[2]
            connection.close()

            if schedule == 1: #now there is schedule so START HEATER
                #print "TYPE A, time=%s" %timen
                #call thermostat
                Ta1 = []
                Ti1 = []
                Th1 = []
                Te1 = []
```

```
                Ts1 = []
                error1 =[]
                Fh1 = []
                Twant1 = []
                w = next_temperature
                prev = 0
                minus = 0
                Ta1, Ti1, Th1, Te1, Ts1, error1, Fh1, Twant1,minus,tenmin = thermostat_pid_ext(Ta[-1],Ti[-1],Th[-1],Te[-1],Ts[-1],w,prev,Fh[-1],minus,tenmin,idloop)
                Ta = Ta + Ta1
                Ti = Ti + Ti1
                Th = Th + Th1
                Te = Te + Te1
                Ts = Ts + Ts1
                error = error + error1
                    Fh = Fh + Fh1
                    Twant = Twant + Twant1
                    if float(Twant[-11]) != float(Twant[-1]) and float(Twant[-11]) <= 10: #first time we run this schedule
                        timestart = datetime.strptime(timen, '%H:%M:%S') #time that the schedule started
                        Tempstart = Ti[-11]
                        onlyatfirst = 1
                        if Ti[-11] > Twant[-1]+1: #error must be negative
                            onlyatfirst =2
                    if Ti[-1]>=Twant[-1] and onlyatfirst==1: #we can calculate smart time
                        onlyatfirst=0
                        forlast=1
                        timestop = datetime.strptime(timen, '%H:%M:%S')

                        smartdif = str(timestop - timestart)
                                                help = smartdif[-8:-7]
                                                if help != "1" and help != "2": #insert 0 at the start of time -> 24hour format
                                                    smartdif = str(0) + str(smartdif)
                                                else:
                                                        smartdif = str(smartdif)
                        #calculate how much minutes smartdif is
                        timeadd = 0
                        if int(smartdif[:-6]) != 0:
                            timeadd += int(smartdif[:-6]) * 60
                        if int(smartdif[-5:-3]) !=0:
                            timeadd += int(smartdif[-5:-3])
                        #update error on smart table
                        test = numday-1
                        test1 = numday-2
                        test2 = numday -3
                        columns = smart.shape[1]
                        for s in range(columns):
                            if day=='Sunday' and smart[18][s] == str(timestart)[-8:]:
                                help = int(smart[19][s]) + int(smart[20][s])
                                smart[19][s] = str(help)
```

```
                                smart[20][s] = str(timeadd)
                            elif smart[test2][s] ==str(timestart)[-
8:]:
                                help = int(smart[test1][s]) + int(smart
[test][s])
                                smart[test1][s] = str(help)
                                smart[test][s] = str(timeadd)
                # STEP 2 find when the next schedule will be
                else: #schedule ==0 so no programme now
                    T = '04:00:00'
                    timenn = datetime.strptime(timen, '%H:%M:%S')
                    connection = MySQLdb.connect(host = "localhost", us
er = "project", passwd="1111", db="project")
                    cur = connection.cursor()
                    cur.execute("SELECT * FROM %s" %day)
                    for row in cur.fetchall():
                        help = str(row[0])[-8:-7]
                           if help != "1" and help != "2": #insert 0 at
 the start of time -> 24hour format
                                row_0 = str(0) + str(row[0])
                           else:
                                row_0 = str(row[0])
                        row_00 = datetime.strptime(row_0, '%H:%M:%S')
                        if row_00 > timenn:
                            dif = str(row_00 - timenn)
                            help = dif[-8:-7]
                            if help != "1" and help != "2": #insert 0 a
t the start of time -> 24hour format
                                  dif = str(0) + str(dif)
                            else:
                                  dif = str(dif)
                            if dif <= T:
                                T = dif
                                next_schedule = str(row_00)
                                next_schedule = next_schedule[-8:]
                                next_temperature = row[2]
                                flag =0
                    connection.close()
                    #check if the next schedule comes in the next day
                    if str(timen) >= '22:00:00' and T > '02:00:00':
                        if day == 'Monday':
                            next_day = 'Tuesday'
                        if day == 'Tuesday':
                            next_day = 'Wednesday'
                        if day == 'Wednesday':
                            next_day = 'Thursday'
                        if day == 'Thursday':
                            next_day = 'Friday'
                        if day == 'Friday':
                           next_day = 'Saturday'
                        if day == 'Saturday':
                           next_day = 'Sunday'
                        if day == 'Sunday':
                           next_day = 'Monday'
                        connection = MySQLdb.connect(host = "localhost"
, user = "project", passwd="1111", db="project")
                        cur = connection.cursor()
                        cur.execute("SELECT * FROM %s" %next_day)
                        for row in cur.fetchall():
```

```
                            help = str(row[0])[-8:-7]
                            if help != "1" and help != "2": #insert 0 at
    the start of time -> 24hour format
                                row_0 = str(0) + str(row[0])
                            else:
                                row_0 = str(row[0])
                            if row_0 <= '02:00:00':
                                row_00 = datetime.strptime(row_0, '%H:%M:
   %S')
                                dif = str( row_00 - timenn)
                                dif = dif[-7:]
                                dif = str(0) + str(dif)
                            if dif <= T:
                                T = dif
                                next_schedule = str(row_00)
                                next_schedule = next_schedule[-8:]
                                next_temperature = row[2]
                                flag =1
                                connection.close()

                    # STEP 3 check if T <= 2
                    if T > '02:00:00': #no heaters
                        next_temperature = 0
                        #call thermostat
                        Ta1 = []
                        Ti1 = []
                        Th1 = []
                        Te1 = []
                        Ts1 = []
                        error1 =[]
                        Fh1 = []
                        Twant1 = []
                        w = next_temperature
                        prev = 0
                        minus = 0
                        Ta1, Ti1, Th1, Te1, Ts1, error1, Fh1, Twant1,mi
   nus,tenmin = thermostat_pid_ext(Ta[-1],Ti[-1],Th[-1],Te[-1],Ts[-
   1],w,prev,Fh[-1],minus,tenmin,idloop)
                        Ta = Ta + Ta1
                        Ti = Ti + Ti1
                        Th = Th + Th1
                        Te = Te + Te1
                        Ts = Ts + Ts1
                        error = error + error1
                        Fh = Fh + Fh1
                        Twant = Twant + Twant1

                    else: #T <= '02:00:00' so check if heaters must go
   on
                        # STEP 4 find how much time before it has to st
   art
                        if day=='Sunday':
                            numday= 18
                        test = numday-1
                        test1 = numday-2
                        test2 = numday -3
                        columns = smart.shape[1]
                        for s in range(columns):
                            #s = int(s)-int(1)
```

```
                            if day=='Sunday' and smart[18][s] == next_schedule and flag==0:
                                if smart[20][s][:1]=="-":
                                    Tpro = int(smart[19][s]) - int(smart[20][s][1:])
                                else:
                                    Tpro = int(smart[19][s]) + int(smart[20][s])
                                    smart[19][s] = str(Tpro)
                                    smart[20][s] = str(0)
                            elif smart[test2][s] == next_schedule and flag==0:
                                if smart[test][s][:1]=="-":
                                    Tpro = int(smart[test1][s]) - int(smart[test][s][1:])
                                else:
                                    Tpro = int(smart[test1][s]) + int(smart[test][s])
                                    smart[test1][s] = str(Tpro)
                                    smart[test][s] = str(0)

                            elif day=='Sunday' and flag==1 and smart[0][s] == next_schedule:
                                if smart[2][s][:1]=="-":
                                    Tpro = int(smart[1][s]) - int(smart[2][s][1:])
                                else:
                                    Tpro = int(smart[1][s]) + int(smart[2][s])
                                    smart[1][s] = str(Tpro)
                                    smart[2][s] = str(0)

                            elif smart[numday][s] == next_schedule and flag==1: #other days with flag==1
                                test = numday +1
                                test1 = numday +2
                                if smart[test1][s][:1]=="-":
                                    Tpro = int(smart[test][s]) - int(smart[test1][s][1:])
                                else:
                                    Tpro = int(smart[test][s]) + int(smart[test1][s])
                                    smart[test][s] = str(Tpro)
                                    smart[test1][s] = str(0)

                        # STEP 5 if the time measure is less than time to next schedule do nothing
                        Tpro_hrs = Tpro//int(60)
                        Tpro_mins = Tpro%int(60)
                        Tpro = "%s:%s:00" % (str(Tpro_hrs), str(Tpro_mins))
                        Tpro = datetime.strptime(Tpro, '%H:%M:%S') #make minutes in time format
                        Tpro = str(Tpro)[-8:]
                        if Tpro < T or Tpro==0: #no need to open heaters yet
                            next_temperature = 0
                            #call thermostat
                            Ta1 = []
```

```
                                Ti1 = []
                                Th1 = []
                                Te1 = []
                                Ts1 = []
                                error1 =[]
                                Fh1 = []
                                Twant1 = []
                                w = next_temperature
                                prev = 0
                                minus = 0
                                Ta1, Ti1, Th1, Te1, Ts1, error1, Fh1, Twant
   1,minus,tenmin = thermostat_pid_ext(Ta[-1],Ti[-1],Th[-1],Te[-1],Ts[-
   1],w,prev,Fh[-1],minus,tenmin,idloop)
                                Ta = Ta + Ta1
                                Ti = Ti + Ti1
                                Th = Th + Th1
                                Te = Te + Te1
                                Ts = Ts + Ts1
                                error = error + error1
                                Fh = Fh + Fh1
                                Twant = Twant + Twant1

                            else: # Tpro >= T so heaters must go on
                                # STEP 6
                                Ta1 = []
                                Ti1 = []
                                Th1 = []
                                Te1 = []
                                Ts1 = []
                                error1 =[]
                                Fh1 = []
                                Twant1 = []
                                w = next_temperature
                                prev = 1
                                minus =1
                                Ta1, Ti1, Th1, Te1, Ts1, error1, Fh1, Twant
   1,minus,tenmin = thermostat_pid_ext(Ta[-1],Ti[-1],Th[-1],Te[-1], Ts[-
   1],w,prev,Fh[-1],minus,tenmin,idloop)
                                Ta = Ta + Ta1
                                Ti = Ti + Ti1
                                Th = Th + Th1
                                Te = Te + Te1
                                Ts = Ts + Ts1
                                error = error + error1
                                Fh = Fh + Fh1
                                Twant = Twant + Twant1
                                if minus == 2 and firsthere==0: #we have ne
   gative error
                                    minus = 3
                                    timenow = timen
                                    firsthere=1
                                next_sch = datetime.strptime(next_schedule,
    '%H:%M:%S')
                                timen = datetime.strptime(timen, '%H:%M:%S'
   )
                                smartdif = str(next_sch - timen)[-8:]
                                help = smartdif[-8:-7]
                                if help != "1" and help != "2": #insert 0 a
   t the start of time -> 24hour form$
```

```
                                    smartdif = str(0) + str(smartdif)[-
    7:]
                                else:
                                    smartdif = str(smartdif)
                                if firsthere==1 and smartdif == "00:10:00":
     #last time in this section
                                    minus = 0
                                    firsthere=0
                                    #calculate time to sub
                                    timenow = datetime.strptime(timenow, '%
    H:%M:%S')
                                    smartdif = str(next_sch-timenow)
                                    help = smartdif[-8:-7]
                                    if help != "1" and help != "2": #insert
     0 at the start of time -> 24hour format
                                       smartdif = str(0) + str(smartdif)[-
    7:]
                                    else:
                                       smartdif = str(smartdif)
                                    #calculate how much minutes smartdif is

                                    timesub = 0
                                    if int(smartdif[:-6]) != 0:
                                        timesub += int(smartdif[:-
    6]) * 60
                                    if int(smartdif[-5:-3]) !=0:
                                        timesub += int(smartdif[-5:-3])

                                    #update error on table
                                    test = numday-1
                                    test2 = numday -3
                                    columns = smart.shape[1]
                                    for s in range(columns):
                                         if day=='Sunday' and smart[18][s]
    == next_schedule and flag==0:

                                            help = int(smart[20][s]) -
     timesub
                                            smart[20][s] = str(help)
                                        elif day=='Sunday' and smart[0][s]
    == next_schedule and flag==1:
                                            help = int(smart[2][s]) -
     timesub
                                            smart[2][s] = str(help)

    elif smart[test2][s] ==next_schedule and flag==0:
                                            help = int(smart[test][s]) -
     timesub
                                            smart[test][s] = str(help)
                                        elif smart[numday][s] == next_sched
    ule and flag==1: #other days with flag==1
                                            test1 = numday +2
                                            help = int(smart[test1][s]) -

     timesub
                                            smart[test1][s] = str(help)

       #PLOT
       times= []
       timess = len(Ti)
```

```
376.    for s in range(timess):
377.        times.append(s/60.0)
378.
379.    #axes = plt.gca()
380.    #axes.set_xlim([0,170])
381.
382.    #fig,ax = plt.figure
383.    fig, ax = plt.subplots()
384.    ax.set_xlim(0,168)
385.    ax.set_xlabel('time(h)')
386.    ax.set_ylabel('Temperature(oC)')
387.
388.    ax.annotate('Monday',xy=(0,0), xytext=(-5,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
389.    ax.annotate('Tuesday',xy=(24,0), xytext=(19,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
390.    ax.annotate('Wednesday',xy=(48,0), xytext=(43,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
391.    ax.annotate('Thursday',xy=(72,0), xytext=(67,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
392.    ax.annotate('Friday',xy=(96,0), xytext=(91,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
393.    ax.annotate('Saturday',xy=(120,0), xytext=(115,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
394.    ax.annotate('Sunday',xy=(144,0), xytext=(139,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
395.    lc = plt.plot(times,Ti,'b-',label = "Inside Temperature")
396.    la = plt.plot(times,Ta,'m-',label="Outside Temperature")
397.    ln = plt.step(times,Twant,'k-',label="Programme")
398.    ld = plt.plot(times,error,'r-',label="Error")
399.    lf = plt.step(times,Fh,'g-',label="Power of Heaters")
400.    #le = plt.plot(times,Th,'y-', label = "Heaters Temperature")
401.    lg = ax.plot(times,Te,'c-',label="Envelop Temperature")
402.    #lm = plt.plot(time,Ts, 'k-', label ="Sensor Temperature")
403.    ll = plt.title('ON-OFF smart thermostat pid week1')
404.    plt.legend(loc=3, bbox_to_anchor = (0.,-
    0.2, 1., 1.102), mode="expand",ncol=3)
405.    figure = plt.gcf()
406.    figure.set_size_inches(16,8)
407.    plt.savefig('./smart_ON-OFF_pid_ext.png',bbox_inches='tight')
408.    plt.show()
409.
410.    fig, ax = plt.subplots()
411.    ax.set_xlim(168,336)
412.    ax.set_xlabel('time(h)')
413.    ax.set_ylabel('Temperature(oC)')
414.
415.    ax.annotate('Monday',xy=(168,0), xytext=(163,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
416.    ax.annotate('Tuesday',xy=(192,0), xytext=(187,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
417.    ax.annotate('Wednesday',xy=(216,0), xytext=(211,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
418.    ax.annotate('Thursday',xy=(240,0), xytext=(235,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
419.    ax.annotate('Friday',xy=(264,0), xytext=(259,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
420.    ax.annotate('Saturday',xy=(288,0), xytext=(283,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
```

```
421.    ax.annotate('Sunday',xy=(312,0), xytext=(307,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
422.
423.    lc = plt.plot(times,Ti,'b-',label = "Inside Temperature")
424.    la = plt.plot(times,Ta,'m-',label="Outside Temperature")
425.    ln = plt.step(times,Twant,'k-',label="Programme")
426.    ld = plt.plot(times,error,'r-',label="Error")
427.    lf = plt.step(times,Fh,'g-',label="Power of Heaters")
428.    #le = plt.plot(times,Th,'y-', label = "Heaters Temperature")
429.    lg = ax.plot(times,Te,'c-',label="Envelop Temperature")
430.    #lm = plt.plot(time,Ts, 'k-', label ="Sensor Temperature")
431.    ll = plt.title('ON-OFF smart thermostat pid week 2')
432.    plt.legend(loc=3, bbox_to_anchor = (0.,-
    0.2, 1., 1.102), mode="expand",ncol=3)
433.    figure = plt.gcf()
434.    figure.set_size_inches(16,8)
435.    plt.savefig('./smart_ON-OFF2_pid2.png',bbox_inches='tight')
436.    plt.show()
437.
438.    fig, ax = plt.subplots()
439.    ax.set_xlim(336,504)
440.    ax.set_xlabel('time(h)')
441.    ax.set_ylabel('Temperature(oC)')
442.
443.    ax.annotate('Monday',xy=(336,0), xytext=(331,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
444.    ax.annotate('Tuesday',xy=(360,0), xytext=(355,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
445.    ax.annotate('Wednesday',xy=(384,0), xytext=(379,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
446.    ax.annotate('Thursday',xy=(408,0), xytext=(403,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
447.    ax.annotate('Friday',xy=(432,0), xytext=(427,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
448.    ax.annotate('Saturday',xy=(456,0), xytext=(451,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
449.    ax.annotate('Sunday',xy=(480,0), xytext=(475,-
    3), arrowprops=dict(facecolor='black',shrink=0.05))
450.
451.    lc = plt.plot(times,Ti,'b-',label = "Inside Temperature")
452.    la = plt.plot(times,Ta,'m-',label="Outside Temperature")
453.    ln = plt.step(times,Twant,'k-',label="Programme")
454.    ld = plt.plot(times,error,'r-',label="Error")
455.    lf = plt.step(times,Fh,'g-',label="Power of Heaters")
456.    #le = plt.plot(times,Th,'y-', label = "Heaters Temperature")
457.    lg = ax.plot(times,Te,'c-',label="Envelop Temperature")
458.    #lm = plt.plot(time,Ts, 'k-', label ="Sensor Temperature")
459.    ll = plt.title('ON-OFF smart thermostat pid week3')
460.    plt.legend(loc=3, bbox_to_anchor = (0.,-
    0.2, 1., 1.102), mode="expand",ncol=3)
461.    figure = plt.gcf()
462.    figure.set_size_inches(16,8)
463.    plt.savefig('./smart_ON-OFF3_pid3.png',bbox_inches='tight')
464.    plt.show()
```