Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή
Στα Συστήματα Ασύρματης Επικοινωνίας

Αμφίδρομη Μεταφορά Δεδομένων Από και Προς Κατανεμημένους Αισθητήρες και Υλοποίηση Συστήματος Ελέγχου των Συσκευών Θέρμανσης Ψύξης με Σκοπό την Εξοικονόμηση Ηλεκτρικής Ενέργειας

Δημήτριος Πατρινός

Επιβλέπων Καθηγητής
Συμεών Νικολάου

Μάϊος 2020

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών
Συστήματα Ασύρματης Επικοινωνίας

Μεταπτυχιακή Διατριβή

Αμφίδρομη Μεταφορά Δεδομένων Από και Προς Κατανεμημένους Αισθητήρες και Υλοποίηση Συστήματος Ελέγχου των Συσκευών Θέρμανσης Ψύξης με Σκοπό την Εξοικονόμηση Ηλεκτρικής Ενέργειας

Δημήτριος Πατρινός

Επιβλέπων Καθηγητής
Συμεών Νικολάου

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων για απόκτηση μεταπτυχιακού τίτλου σπουδών στα Συστήματα Ασύρματης Επικοινωνίας

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών του Ανοικτού Πανεπιστημίου Κύπρου

Μάϊος 2020

Περίληψη

Σκοπός της παρούσας μεταπτυχιακής διατριβής είναι η μελέτη οικοσυστημάτων IoT και η ανάλυση των δυνατοτήτων που προσφέρουν και συγκεκριμένα ο σχεδιασμός και η υλοποίηση συστήματος ελέγχου συσκευών θέρμανσης ψύξης με σκοπό την εξοικονόμηση ηλεκτρικής ενέργειας. Ο έλεγχος αυτός πραγματοποιείται με αμφίδρομη επικοινωνία μεταξύ συσκευής θέρμανσης/ψύξης και αναπτυσσόμενης συσκευής που θα περιέχει αισθητήρες μέτρησης, ασύρματη δυνατότητα διασύνδεσης, δέκτη και πομπό υπέρυθρων σημάτων. Σχεδιάστηκε και έχει την ικανότητα να λειτουργεί πλήρως αυτοματοποιημένα, χωρίς ανθρώπινη παρέμβαση, με στόχο την εξοικονόμηση ενέργειας, που μειώνεται σχεδόν στο 1/3 με την χρήση του αναπτυσσόμενου συστήματος. Παρόλο που δεν απαιτείται η ανθρώπινη παρέμβαση, προσφέρεται διεπαφή του χρήστη υλοποιημένη σε περιβάλλον Android για απομακρυσμένη παρακολούθηση, όπως επίσης και δυνατότητα αλλαγής των προκαθορισμένων παραμέτρων λειτουργίας της συσκευής.

Στην αρχή της διατριβής γίνετε αναφορά για τη IoT τεχνολογία, τα οφέλη που προσφέρει τόσο σε οικιακές εφαρμογές όσο και σε βιομηχανικές εφαρμογές. Αναλύεται η αρχιτεκτονική και ο τρόπος λειτουργίας μιας τέτοιας τεχνολογίας. Στη συνέχεια περιγράφεται το σύστημα ελέγχου που θα υλοποιηθεί. Ο χώρος της συγκεκριμένης εφαρμογής που θα παρακολουθεί και θα διαχειρίζεται η συσκευή, καθώς και οι τεχνολογίες με το διαθέσιμο υλικό που υπάρχει, επιλέγοντας το καταλληλότερο, για βέλτιστη δυνατότητα καταγραφής και ελέγχου του συστήματος. Περιγράφονται τα στάδια της υλοποίησης της ασύρματης συσκευής, τα προβλήματα που εμφανίστηκαν και οι λύσεις που υιοθετήθηκαν, ώστε να ολοκληρωθεί η τελική συσκευή. Αναλύονται οι αποφάσεις που ελήφθησαν ώστε να επιτύχουμε την καλύτερη αυτοματοποίηση θερμοκρασίας. Η καταγραφή δεδομένων κατανάλωσης ενέργειας και συνθηκών δωματίου ξεκίνησε από τις 21 Ιανουαρίου 2020 και συνεχίζεται έως και σήμερα. Τέλος παρουσιάζονται τα συμπεράσματα από την αξιολόγηση των δεδομένων καθώς και μελλοντικές προτάσεις για επεκτασιμότητα της IoT συσκευής. Σημειώνεται ότι η συσκευή συνεχίζει να λειτουργεί καθώς αποδεικνύεται ότι η ενεργειακή και κατ' επέκταση οικονομική εξοικονόμηση για την Υπηρεσία είναι πολύ μεγάλη.

Summary

The purpose of this M.Sc. dissertation is the study of IoT ecosystems and analyze the possibilities they offer. The goal is to design and implement a control system for cooling / heating devices in order to save electricity. This implementation is succeed through two-way communication between the cooling / heating device and a growing device that will contain measuring sensors, wireless interface and infrared signal receiver and transmitter module. After the successfully design and testing the system has the ability to operate fully automated without human intervention, with the aim of saving energy, which is reduced to almost 1/3 with the user of the growing system. Although no human intervention is required, the user interface is implemented in an Android environment offering remote monitoring, as well as the ability to change the default operating parameters of the device.

The structure of the dissertation include information about IoT technology, the benefits it offers in both home applications and industrial applications. Analysis of the architecture and operation of such technology. The description of the control system is presented, analyzing the location of the specific application that will be monitored and managed by the device, as well as the technologies with the available material that exists, choosing the most suitable, for optimal possibility of recording and control of the system. The stages of the implementation of the wireless device, the problems that occurred and the solutions that were adopted are described, in order to complete the final device. The decisions made to achieve the best temperature automation are analyzed. The recording of energy consumption data and room conditions started on January 21, 2020 and continues to this day. Finally, the conclusions from the data evaluation are presented as well as future proposals for the extensibility of the IoT device. The device continues to operate as it proves that the energy and therefore economical savings for the building were very high.

Ευχαριστίες

Για την εκπόνηση και ολοκλήρωση της παρούσας μεταπτυχιακής διατριβής, θα ήθελα να ευχαριστήσω την οικογένειά μου για την συνεχή τους υποστήριξη. Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Συμεών Νικολάου για την στήριξη που μου παρείχε, καθώς και την ενθάρρυνση για την υλοποίηση του στόχου μου.

Επίσης, θέλω να ευχαριστήσω τον προϊστάμενό μου κ. Ιωάννη Κλαψόπουλο, που μου επέτρεψε να εγκαταστήσω και να λειτουργώ την συσκευή που υλοποίησα κάνοντας χρήση του κλιματιστικού εξοπλισμού του κτιρίου, καταγράφοντας δεδομένα σε πραγματικές συνθήκες λειτουργίας.

Περιεχόμενα

Κεφάλαιο 1
Εισαγωγή

Η συνεχιζόμενη αύξηση των εφαρμογών του Διαδικτύου των Πραγμάτων (IoT) με υλοποιήσεις ολοένα και σε περισσότερους τομείς, έχει φτάσει και απασχολεί σημαντικά τον τομέα της ενέργειας. Μελέτες και παραδείγματα εφαρμογών αποδεικνύουν ότι με μικρή οικονομική επένδυση, της οποίας γίνεται απόσβεση σε σύντομο χρονικό διάστημα, επιτυγχάνεται ενεργειακή αλλά και οικονομική εξοικονόμηση, προσφέροντας ταυτόχρονα βελτίωση στις συνθήκες του περιβάλλοντος αλλά και στην κοινωνική άνεση.

Η παρούσα εργασία έχει στόχο την ενεργειακή εξοικονόμηση. Αφορά στην αυτοματοποίηση της λειτουργίας κλιματιστικών μονάδων. Στόχος είναι να αναλυθούν οι παράμετροι και τα προβλήματα που μπορεί να υπάρξουν σε μια αμφίδρομη επικοινωνία μεταξύ αναπτυσσόμενης ασύρματης συσκευής και κλιματιστικού, καθώς και η ύπαρξη συνεχούς ενημέρωσης του ανθρώπινου παράγοντα με τεχνολογικά μέσα. Η υλοποίηση δεν απευθύνεται σε εταιρίες και οργανισμούς διαχείρισης ενέργειας αλλά σε καταναλωτές που επιθυμούν να ελαχιστοποιήσουν περιττές καταναλώσεις ενέργειας από συσκευές που μπορούν να ελεγχτούν μέσω υπέρυθρων σημάτων επικοινωνίας.

1.1 Ανάγκες εφαρμογής ενεργειακού αυτοματισμού

Η ανάγκη για εξοικονόμηση ενέργειας σε κτίρια είναι παγκόσμια και δεν αφορά μόνο σε οικιακές εφαρμογές αλλά και σε κτίρια ή βιομηχανίες με μεγάλες ενεργειακές καταναλώσεις, όπου συναντάμε αρκετά παραδείγματα υιοθέτησης ενεργειακών αυτοματισμών.
Όλο και περισσότερα κτίρια προχωρούν στην υιοθέτηση και εγκατάσταση διαφόρων αυτοματισμών που έχουν ως στόχο την εξυπηρέτηση του ανθρώπου, με απώτερο σκοπό την εξοικονόμηση ενέργειας.

Παράδειγμα τέτοιας εφαρμογής είναι και ένα νέο κτίριο του Πανεπιστημίου της Χάγης που εγκαινιάστηκε το καλοκαίρι του 2019, και στο οποίο κατά την ξενάγηση που έγινε στα πλαίσια ενός επαγγελματικού ταξιδιού, παρουσιάστηκαν πολλές διαφορετικές χρήσεις συστημάτων αυτοματισμού. Όλα τα συστήματα φωτισμού και κλιματισμού (θερμοκρασίας) ήταν αυτοματοποιημένα. Για παράδειγμα τα φώτα άναβαν όταν κάποιος εισέρχονταν σε κάποιο δωμάτιο και αυτό μετά από αυτόματο έλεγχο του υπάρχοντος φωτισμού του χώρου από εξωτερικές πηγές φωτός. Εκτός του φωτισμού και κλιματισμού, υπήρχαν αυτοματισμοί για την σκίαση των χώρων, συλλογή και αξιοποίηση ηλιακής ενέργειας που καθιστούσε σχεδόν ενεργειακά αυτόνομο το κτίριο, αυτόματος εντοπισμός θέσης των συνδεδεμένων χρηστών στο δίκτυο του κτιρίου. Το τελευταίο ήταν αρκετά ενδιαφέρων αφού οι διδάσκοντες δεν είχαν σταθερό χώρο εργασίας (γραφείο καθηγητή), με αποτέλεσμα να χρειάζεται ο αυτόματος εντοπισμός τους μέσα στο πολυώροφο συγκρότημα, με επιλογή απενεργοποίησης από τον χρήστη, ακολουθώντας τον κανονισμό προστασίας προσωπικών δεδομένων.

Η αυτοματοποίηση ενός κτιρίου γίνεται από το Σύστημα Διαχείρισης Κτιρίου (BMS). Το BMS (Building Management Systems) αλλιώς γνωστό και ως BAS (Building Automation System) είναι ένα σύστημα ελέγχου που εποπτεύει και ελέγχει ηλεκτρολογικά αλλά και μηχανολογικά συστήματα του κτιρίου, όπως:

- Φωτισμός
- Ηλεκτρική ισχύ
- Θέρμανση, εξαερισμός και κλιματισμός
- Ασφάλεια και παρακολούθηση
- Πρόσβαση
- Σύστημα πυρόσβεσης
- Ανελκυστήρες
- Υδραυλικά

- Κλειστό κύκλωμα τηλεόρασης (CCTV)
- Παρακολούθηση συναγερμού

Το BMS αποτελείται από υλικό (Hardware) και από λογισμικό (Software), χρησιμοποιώντας ανοιχτά πρωτόκολλα επικοινωνίας όπως BACnet, LonWorks, Modbus.

- BACnet είναι ένα πρωτόκολλο επικοινωνίας δικτύων για Αυτοματισμούς και Ελέγχους Κτιρίων (BAC - Building Automation and Control) το οποίο επωφελείται τα τυπικά πρωτόκολλα ASHRAE, ANSI και ISO 16484-5.
- LonWorks είναι μια πλατφόρμα δικτύωσης που δημιουργήθηκε ειδικά για να καλύψει τις ανάγκες των εφαρμογών ελέγχου. Η πλατφόρμα είναι χτισμένη με ένα πρωτόκολλο που δημιουργήθηκε από την Echelon Corporation για δικτυακές συσκευές μέσα από twisted pair, γραμμές μεταφοράς ηλεκτρικής ενέργειας (powerlines), οπτικές ίνες, και RF. Χρησιμοποιείται για την αυτοματοποίηση διαφόρων λειτουργιών μέσα σε κτίρια, όπως τον φωτισμό και τον κλιματισμό.
- Modbus είναι ένα πρωτόκολλο σειριακής επικοινωνίας που παρουσιάστηκε από την Modicon το 1979 για χρήση με προγραμματιζόμενους λογικούς ελεγκτές (PLCs). Το Modbus είναι δημοφιλής σε βιομηχανικό περιβάλλον διότι είναι ανοιχτό και ελεύθερο. Αναπτύχθηκε για βιομηχανικές εφαρμογές και είναι σχετικά εύκολο να εφαρμοστεί και να υποστηριχτεί σε σύγκριση με άλλα πρότυπα, έχοντας περιορισμούς μόνο στο μέγεθος και τη μορφή των δεδομένων που πρέπει να διαβιβάζονται. Το Modbus χρησιμοποιεί το RS485 ως φυσικό στρώμα του.

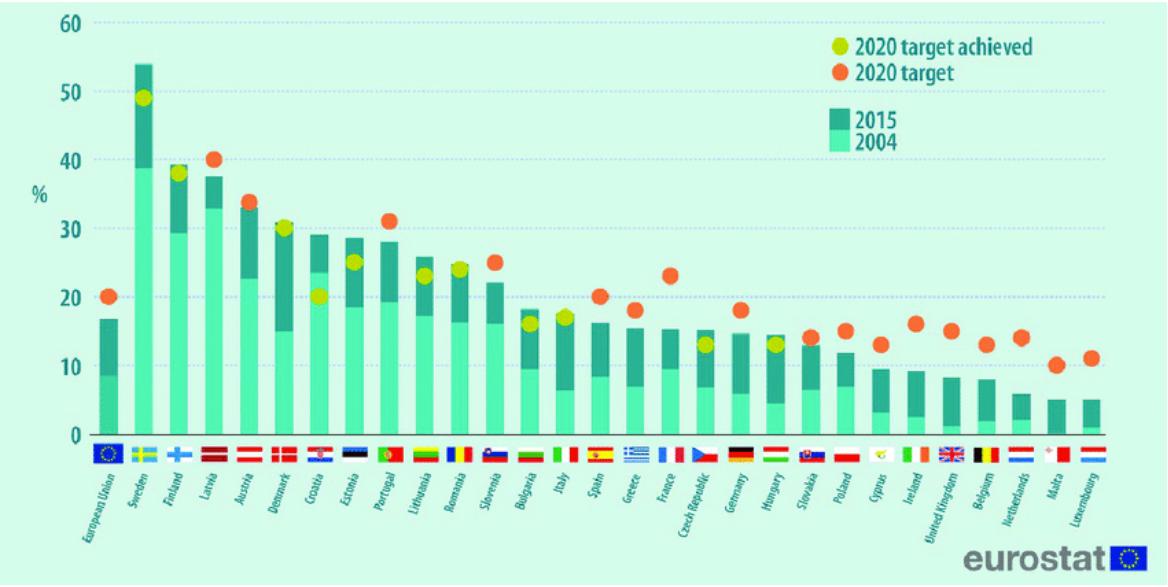
Τα BMS συνήθως παρουσιάζουν 40% εξοικονόμηση ενέργειας και φτάνει στο 70% αν συμπεριληφθεί και ο φωτισμός. Η μη σωστή ρύθμιση ενός συστήματος BMS πιστεύεται ότι ευθύνεται για το 20% της ενεργειακής χρήσης του κτιρίου, ή περίπου το 8% της συνολικής χρήσης ενέργειας των Ηνωμένων Πολιτειών [02].

Το μεγαλύτερο μέρος της ηλεκτρικής ενέργειας που παράγεται και καταναλώνεται προέρχεται από την χρήση συμβατικών καυσίμων (άνθρακα , λιγνίτη, πετρέλαιο) ρυπαίνοντας τον ατμοσφαιρικό αέρα με διοξείδιο του άνθρακα. Εφαρμόζοντας σε μεγάλα κτιριακά συγκροτήματα συστήματα ελέγχου, όπως το BMS, μπορούμε εκτός από την μείωση της κατανάλωσής ενέργειας να μειώσουμε ταυτόχρονα και τους ατμοσφαιρικούς ρύπους.

Τα οφέλη λοιπόν της εφαρμογής ενός τέτοιου συστήματος είναι

- Ενεργειακά
- Οικονομικά
- Περιβαλλοντικά
- Κοινωνικά

Ο τρόπος κατανάλωσης ενέργειας πρέπει να αλλάξει τόσο για τα παραπάνω οφέλη αλλά κυρίως γιατί οι επόμενες γενεές θα φτάσουν σε κομβικό σημείο στο μέλλον με την συνεχή μείωση των συμβατικών καυσίμων. Θα πρέπει να ερευνηθούν και να επενδυθούν μελέτες με αντικείμενο τους τρόπους παραγωγής εναλλακτικών πηγών ενέργειας και σωστότερης διαχείρισης. Το σημαντικότερο βήμα είναι η ενεργειακή πράσινη υλοποίηση σε δημόσια και ιδιωτικά μεγάλα κτίρια και βιομηχανίες, που έχουν τις μεγαλύτερες ενεργειακές καταναλώσεις. Οι κυβερνήσεις των κρατών θα πρέπει να ευαισθητοποιηθούν αλλά και να ευαισθητοποιήσουν τον κόσμο μέσα από ενημερώσεις, επιδοτήσεις προγραμμάτων κλπ. Αρκετά προγράμματα εφαρμόζονται συχνά κυρίως για ιδιωτικές κατοικίες π.χ. Πρόγραμμα «Εξοικονόμηση κατ΄ Οίκον ΙΙ» [1], που βασίζονταν στην ενεργειακή βελτιστοποίηση του σπιτιού μέσω νέων κουφωμάτων, θερμομόνωσης, , ηλιακών εγκαταστάσεων, κλπ. Η ευρωπαϊκή ένωση θέτει στόχους παραγωγής ενέργειας από εναλλακτικές πηγές. Στο γράφημα (εικόνα 1.1) παρουσιάζονται τα στοιχεία ποσοστού επίτευξης παραγωγής τέτοιας ενέργειας ανά χώρα με δεδομένα τα έτη 2004, 2015 και επίτευξη στόχου το 2020. Παρατηρείται ότι ορισμένα κράτη πετυχαίνουν τους στόχους του 2020, ενώ άλλα απέχουν αρκετά.



Εικόνα 1.1 - Στόχοι παραγωγής ενέργειας ανά χώρα

[1] https://exoikonomisi.ypen.gr/to-programma

1.2 Βασικοί στόχοι υλοποίησης

Στόχος του παραδοτέου συστήματος είναι να καλύπτει πλήρως τα οφέλη ενός IoT συστήματος. Αναλυτικότερα το σύστημα πρέπει να είναι πλήρως αυτοματοποιημένο χωρίς ανθρώπινη παρέμβαση, να μπορεί να επαναλειτουργεί σε διάφορα σενάρια σφαλμάτων, όπως σε απώλεια τροφοδοσίας κλπ. Να παρέχει φορητότητα, να είναι δηλαδή ευέλικτο, μικρό σε μέγεθος και να μπορεί να λειτουργήσει αυτόνομα σε οποιαδήποτε κλιματιστική μονάδα. Να διαθέτει απομακρυσμένη διαχείριση του συστήματος και ανανέωση κώδικα. Οικονομικό, να προσφέρεται δηλαδή με χαμηλό κόστος υλοποίησης. Να μπορεί να επεκταθεί ο έλεγχος και η εφαρμογή σε μεγαλύτερη κλίμακα κάλυψης και εκτός του κτιρίου και να μπορεί να υποστηρίξει περισσότερες διαφορετικές κλιματιστικές μονάδες ταυτόχρονα. Να διαθέτει αμφίδρομη και γρήγορη επικοινωνία με άμεση ανταπόκριση τόσο των εντολών προς το κλιματιστικό όσο και της ενημέρωσης του χρήστη για τα αποτέλεσματα μέσα από την εφαρμογή του κινητού. Τέλος να αποφέρει σημαντική μείωση ενεργειακής κατανάλωσης που αποτελεί και τον βασικότερο στόχο και που οδηγεί σε οικονομική και περιβαλλοντική βελτιστοποίηση.

Κεφάλαιο 2
Παρουσίαση IoT

2.1 Βιβλιογραφική επισκόπηση

Δεδομένου ότι ο όρος Διαδίκτυο των Πραγμάτων (IoT) επεκτείνεται ραγδαία υπάρχει μια τεράστια αύξηση του αριθμού των μητρικών ανάπτυξης που εισάγονται στην αγορά και κυμαίνονται σε μέγεθος ενός νομίσματος μέχρι και μιας πιστωτικής κάρτας.
Τα κριτήρια για να χαρακτηριστεί κάποια μητρική (board) υποψήφια και κατάλληλη για IoT εφαρμογές είναι:

- Η σχέση κόστους - αποτελεσματικότητας
- Η αποδοτικότητα της ισχύς που παρέχει
- Η εύκολη εγκατάσταση / ανάπτυξη

Στην διεθνή αρθρογραφία υπάρχει μεγάλος συναγωνισμός για το ποια θεωρούνται τα καταλληλότερα χαμηλού κόστους (low cost) boards, για ανάπτυξη στον τομέα της εξοικονόμησης ενέργειας μέσω IoT

Εφαρμογών. Στο συνέδριο μελλοντικών τεχνολογιών του San Francisco των Ηνωμένων Πολιτειών [16] παρουσιάστηκε μία τέτοια σύγκριση.

Οι εφαρμογές του Διαδικτύου των Πραγμάτων έχουν αναπτυχθεί σε πολλούς διαφορετικούς τομείς, ορισμένοι εκ των οποίων είναι:

- Ασύρματοι αισθητήρες και συστήματα παρακολούθησης δικτύων
- Συστήματα παρακολούθησης Υγείας
- Έξυπνα Σπίτια
- Συνδεδεμένα Αυτόνομα Οχήματα
- Αυτοματοποιημένη διαχείριση των ηλεκτρικών δικτύων (Smart Grids)
- Έξυπνες βιομηχανίες (Smart Industries)

Οι διαφορετικοί τύποι υλικού boards που συναντάμε είναι:

- Micro-Controller,
- System-on-Chips,
- Single Board Computers (SBC).

Με βασικά χαρακτηριστικά:

- Πολλαπλές εισόδους/ εξόδους για να χρησιμοποιηθούν με διαφορά πρωτοκολλά επικοινωνίας όπως τα GPIO, I2C, UART, SPI, κ.λπ.
- Ενσωματωμένο το ADC (Analog-to-Digital Converter), απαραίτητο για την επικοινωνία και την αλληλεπίδραση του board με τους αισθητήρες.
- Πολλαπλά πρωτόκολλα επικοινωνίας
- Δυνατότητες Δικτύωσης όπως BLE (Bluetooth Low Energy), Wi-Fi και Ethernet.
- Χαμηλή κατανάλωση ενέργειας
- Εύκολη εγκατάσταση και χαμηλή συντήρηση
- Υποστήριξη πολλαπλών OS
- OpenSource σχεδιασμός
- Δυνατότητα απομακρυσμένης διαχείρισης

2.1.1 Έρευνες διαχείρισης της ενεργειακής εξοικονόμησης με Arduino Boards

Οι διάφορες έρευνες και υλοποιήσεις εφαρμογών πάνω στην διαχείριση της ενεργειακής εξοικονόμησης και στους αυτοματισμούς μετρήσεων κατανάλωσης χρησιμοποιούν όλο και περισσότερο Arduino boards, γιατί κατέχει την μεγαλύτερη κοινότητα ανάπτυξης λύσεων σε

λογισμικό, βιβλιοθήκες, αισθητήρες και επεκτάσιμα τμήματα (Modules). Αυτή η μεγάλη αποδοχή ώθησε τις εταιρίες κατασκευής hardware να παράγουν προϊόντα συμβατά με Arduino board.

Ορισμένες έρευνες που αναπτυχτήκαν με τέτοια boards είναι:

- Smart Metering for Low Voltage Electrical Distribution System using Arduino Due

Η δημοσίευση [19] παρουσιάζει ένα οικονομικό, ευέλικτο και αξιόπιστο σύστημα που μετρά καταναλώσεις σε πραγματικό χρόνο και εκτελεί αυτοματοποιημένες εντολές.

- Demo Abstract: Design and Implementation of a Low-cost Arduino-based High-Frequency AC Waveform Meter Board for the Raspberry Pi

Η δημοσίευση [13] παρουσιάζει ένα μετρητή χαμηλού κόστους για AC κυματομορφές τάσης και ρεύματος σε ποσοστό μέχρι και 14 kHz, σχεδιασμένο για χρήση με μικροελεγκτή όπως το Arduino UNO αλλά και συμβατό με Raspberry Pi 3. Η συσκευή μπορεί να μετρά μονοφασική αλλά και τριφασική παροχή ρεύματος.

- Measuring of Electric Energy Consumption in Households by Means of Arduino Platform

Το άρθρο [07] παρουσιάζει τις δυνατότητες της μέτρησης κατανάλωσης της ηλεκτρικής ενέργειας στα νοικοκυριά. Οι τρέχουσες προσεγγίσεις και τεχνολογικές λύσεις αναλύονται και εξετάζονται προσεκτικά σε σχέση με τις θετικές και αρνητικές πτυχές τους.

- Advanced Energy Management of a Micro-grid Using Arduino and Multi-agent System

Ο στόχος της δημοσίευσης [15] ήταν η ανάπτυξη Arduino με βάση το multi-agent σύστημα (MAS) για την προηγμένη κατανεμημένη διαχείριση ενέργειας τόσο της αιολικής αλλά και της ηλιακής. Η υψηλή διείσδυση των ανανεώσιμων πηγών ενέργειας ανάπτυξε την ανάγκη για σωστό συντονισμό και ελεγχόμενες προσεγγίσεις. Αναπτύχθηκε μοντέλο προσομοίωσης χρησιμοποιώντας Java Agent Development Environment (JADE) σε λογισμικό Eclipse IDE για τη δυναμική διαχείριση της ενέργειας. Περιβαλλοντικές μετρήσεις ανιχνεύονται μέσω Arduino Mega και στην συνέχεια λαμβάνονται οι αποφάσεις. Επειδή η δημοσίευση αφορούσε προσομοίωση, οι προκύπτουσες ενέργειες αντανακλούσαν σε LED εξόδους με μελλοντική ανάπτυξη στο πραγματικό πεδίο εφαρμογής.

- Low cost Arduino / Android - based Energy - Efficient Home Automation System with Smart Task Scheduling

Το άρθρο [01] παρουσιάζει τη χρήση τεχνικών αυτοματισμών, ώστε να σχεδιαστεί και να αναπτυχθεί ένα σύστημα ενεργειακής απόδοσης υψηλής επεκτασιμότητας Smart Home, με χαρακτηριστικά διασφάλισης άνεσης και ασφάλειας των κατοίκων. Το περιγραφόμενο σύστημα αποτελείται από ένα δίκτυο με αισθητήρες και συσκευές, οι οποίες λαμβάνουν πληροφορίες για το περιβάλλον του σπιτιού. Ως κεντρικό ελεγκτή, χρησιμοποιείται ένας Arduino μικροελεγκτής, που επικοινωνεί με την Android εφαρμογή η οποία είναι η διεπαφή του χρήστη.

- Utilization of Solar Cells as Energy Sources for Heating and Fan (Exhouse) in White Copra Dryers with Arduino Uno as Temperature Control

Η μελέτη [23] αναπτύχθηκε χρησιμοποιώντας ως κεντρικό ελεγκτή δεδομένων ένα Arduino Uno. Στόχος ήταν να διατηρείται η θερμοκρασία σταθερή σε δωμάτιο στεγνωτήριο, ελέγχοντας την θερμοκρασία με έναν LM35 αισθητήρα θερμοκρασίας και ενεργοποιώντας ή απενεργοποιώντας την θέρμανση ώστε να κυμαίνεται μέσα στο διάστημα 60^{o}C - 80^{o}C. Παράλληλα αν αυξάνονταν πάνω από τους 80^{o}C ενεργοποιούνταν ειδικός ανεμιστήρας για εκτόνωση της θερμοκρασίας του δωματίου.

- Arduino and GSM Based Smart Energy Meter for Advanced Metering and Billing System
- Smart Energy Meter Using Arduino and GSM

Και οι δύο μελέτες [14],[03] παρουσιάζουν ένα έξυπνο μετρητή ενέργειας για αυτόματη μέτρηση με ενσωματωμένο χρεωστικό σύστημα. Υλοποιείται η ενσωμάτωση του Arduino board με GSM module για να χρησιμοποιεί την υπηρεσία σύντομων μηνυμάτων (SMS). Με την χρήση του GSM module υπήρχε το πλεονέκτημα της φορητότητας του συστήματος, να μπορεί δηλαδή να εγκατασταθεί οπουδήποτε. Παρέχοντας πληροφόρηση και αυτοματισμούς, όπως την ενημέρωση κατανάλωσης ενέργειας σε kWh, παραγωγή λογαριασμών σύμφωνα με την κατανάλωση, αυτόματες υπηρεσίες ασφαλείας όπως την ενεργοποίηση ή απενεργοποίηση της παροχής ρεύματος. Η υλοποίηση απευθύνονταν στις εταιρίες και οργανισμούς διαχείρισης ενέργειας παρέχοντας υπηρεσίες εξ αποστάσεως για τη μέτρηση και χρέωση με υψηλή πιστότητα.

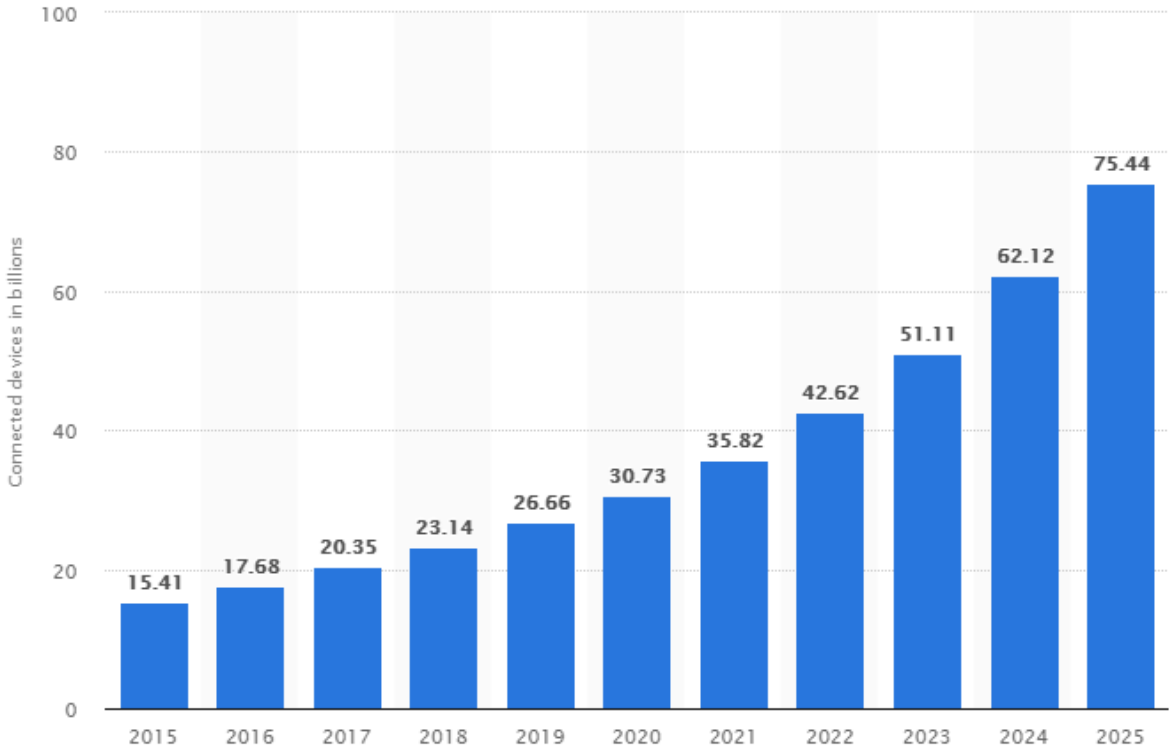
- YaY - An Open-Hardware Energy Measurement System for Feedback and Appliance Detection based on the Arduino Platform

Η μελέτη αυτή [10] παρουσιάζει ένα κατανεμημένο σύστημα μέτρησης που καταγράφει και παρακολουθεί ηλεκτρικές οικιακές συσκευές με την χρήση αισθητήρων, έξυπνων πριζών και τον μετρητή ενέργειας YaY, αναλύοντας και ενημερώνοντας τον χρήστη για το σύνολο της κατανάλωσης ρεύματος του νοικοκυριού.

2.2 Διαδίκτυο των Πραγμάτων (IoT)

Μια από τις πιο πολύ αναφερόμενες ορολογίες στον κόσμο της ανάπτυξης εφαρμογών είναι το IoT ή αλλιώς το Internet of Things. Από τις πιο απλές καταναλωτικές εφαρμογές που αφορούν και εφαρμόζονται σε έξυπνα σπίτια και wearables συσκευές στις πιο περίπλοκες λύσεις για βιομηχανική εφαρμογή, όπως ανυψωτικά μηχανήματα χωρίς οδηγό το IoT είναι παντού και σταδιακά θα αλλάζει τον τρόπο που ζουν οι καταναλωτές, την εργασία τους και την αλληλεπίδραση με τις Internet-enabled συσκευές τους.

Σύμφωνα με μια πρόσφατη έκθεση από τη Statista[2], θα υπάρχουν σχεδόν 31 δισεκατομμύρια συνδεδεμένες smartphones, wearables συσκευές, έξυπνα ρολόγια, αυτοκίνητα, και άλλες συσκευές μέχρι το τέλος του 2020. Επί του παρόντος, ο αριθμός αυτός είναι περίπου 27 δισεκατομμύρια σε όλο τον κόσμο (εικόνα 2.1). Αυτό αποδεικνύει πόσο εκθετικά αυξάνεται το IoT.



Εικόνα 2.1 - Εκθετική αύξηση των IoT συσκευών (Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025(in billions))

2.2.1 Τι είναι το Διαδίκτυο των Πραγμάτων (IoT)

Το Διαδίκτυο των πραγμάτων, ή IoT, είναι ουσιαστικά ένα οικοσύστημα από φυσικές συσκευές (appliances), οχήματα και άλλα πράγματα που έχουν τη δυνατότητα να συνδεθούν, να συλλέξουν ή και να ανταλλάξουν δεδομένα μέσω ενός ενσύρματου ή ασύρματου δικτύου, με ελάχιστη ή καμία παρέμβαση μεταξύ ανθρώπου με άνθρωπο (human-to-human) ή ανθρώπου με υπολογιστή (human-to-computer). Επιτρέποντας την ένταξη και την ανταλλαγή δεδομένων μεταξύ των φυσικών συσκευών και του υπολογιστή, αυτό το νέο κύμα της τεχνολογίας εστιάζει στο να κάνει την ανθρώπινη

[2] https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

ζωή πιο απλοποιημένη και άνετη κρατώντας την αποτελεσματικότητα και την παραγωγικότητα σε υψηλά επίπεδα.

Πιο συγκεκριμένα, αξιοποιώντας τεχνολογίες αιχμής όπως η Μηχανική Μάθηση, η Machine-to-Machine (M2M) Επικοινωνία και η Τεχνητή Νοημοσύνη (AI), οι στόχοι του IoT επεκτείνονται και αλλάζουν πέρα από την υπάρχουσα συνδεσιμότητα με το τυπικό Διαδίκτυο που υποστηρίζει φυσικές συσκευές (smartphones, ταμπλέτες, υπολογιστές γραφείου, και φορητούς υπολογιστές), σε ένα ευρύ φάσμα μη-internet-enabled φυσικές συσκευές και αντικείμενα καθημερινής χρήσης, όπως καφετιέρες, πλυντήρια, κλειδαριές στις πόρτες, κλπ, ώστε να μπορούμε με τις διαχειριστούμε απομακρυσμένα με τη βοήθεια μιας κινητής συσκευής ή tablet .

2.3 Οφέλη του Διαδικτύου των Πραγμάτων (IoT)

Σκοπός του Διαδικτύου των Πραγμάτων (IoT) είναι η αυτοματοποίηση της ανθρώπινης καθημερινότητας με στόχο να γίνει πιο αποτελεσματική και παραγωγική. Προκύπτουν επιπλέον οφέλη με την χρήση του, τόσο για τις επιχειρήσεις όσο και για τους καταναλωτές με κυριότερα τα παρακάτω.

1. Η πρόσβαση σε υψηλής ποιότητας δεδομένων

Με την ανάπτυξη των συσκευών IoT, οι επιχειρήσεις διαθέτουν μεγαλύτερη πρόσβαση σε δεδομένα που αφορούν είτε τους πελάτες τους, είτε τα παραγόμενα προϊόντα τους. Μπορούν να επωφεληθούν από αυτά σε πραγματικό χρόνο με λειτουργικές γνώσεις για την παρακολούθηση της συμπεριφοράς των καταναλωτών, προσφέροντας καλύτερη εμπειρία στον πελάτη και παίρνοντας έξυπνες επιχειρηματικές αποφάσεις.

2. Η βελτίωση της παρακολούθησης και διαχείρισης

Οποιοδήποτε και να είναι το αντικείμενο της επιχείρησης, το IoT κάνει την παρακολούθηση και τη διαχείριση ευκολότερη. Συναντάμε παραδείγματα που μπορούν να βελτιώσουν την παρακολούθηση αντικειμένων σε περιπτώσεις πχ. απογραφής, για παρακολούθηση συνθηκών οδικής κυκλοφορίας και καιρικών συνθηκών με κοινοποίηση δεδομένων στις αρμόδιες αρχές. Φτάνοντας σε σημείο πλέον το IoT να χρησιμοποιείται όχι μόνο για μετατροπή έξυπνων σπιτιών αλλά επίσης έξυπνων «οτιδήποτε». Χρησιμοποιείται από οργανισμούς και επιχειρήσεις ή μεμονωμένους χρήστες διευκολύνοντας τον τρόπο λειτουργίας και παρακολούθησης, για παράδειγμα έξυπνη αποθήκη, έξυπνο γραφείο, έξυπνο αυτοκίνητο, έξυπνη συσκευή.

3. Η αποτελεσματική χρήση των πόρων

Ανεξάρτητα από τον χώρο υλοποίησης (το σπίτι, την επιχείρηση, το γραφείο, την αποθήκη, το ξενοδοχείο ή το αυτοκίνητο), το IoT βοηθάει την αποτελεσματικότερη αξιοποίηση των περιουσιακών στοιχείων με τη βελτίωση της παραγωγικότητας. Αξιοποιώντας τη δύναμη της αλληλεπίδρασης της μηχανής-προς-μηχανή, συλλέγοντας δεδομένα σε πραγματικό χρόνο με τη βοήθεια των αισθητήρων, μπορεί να χρησιμοποιηθεί για την περαιτέρω βελτίωση συνθηκών και την ελαχιστοποίηση της ανθρώπινης παρέμβασης. Ένα χαρακτηριστικό παράδειγμα χρήσης είναι η αποτελεσματική εξοικονόμηση κατανάλωσης ηλεκτρικού ρεύματος.

4. Ο αυτοματισμός και ο έλεγχος

Δεδομένου ότι οι περισσότερες από τις συσκευές IoT μπορούν να επικοινωνούν μεταξύ τους μέσω μιας ασύρματης υποδομής, είναι σε θέση να λειτουργούν από μόνες τους με ελάχιστη ή χωρίς καμία χειροκίνητη παρέμβαση. Για παράδειγμα, οικιακές συσκευές, όπως air condition, πλυντήρια, φούρνοι, ψυγεία και φωτισμός μπορούν να τεθούν αυτόματα σε λειτουργία, να παρακολουθηθούν και να ελεγχθούν από απόσταση. Επίσης μπορεί να αυτοματοποιήσει διάφορους ηλεκτρονικούς σηματοδότες σε ένα περιβάλλον οδικής κυκλοφορίας, ώστε να γίνει καλύτερη διαχείριση της κυκλοφορίας ή να ενημερωθούν οι χρήστες του οδικού περιβάλλοντος.

5. Η άνεση και η ευκολία

Η διασύνδεση των συσκευών και η ομαδοποίηση των δεδομένων μας παρέχει πλήρη έλεγχο σε όλες τις συσκευές που είναι συνδεδεμένες μεταξύ τους μέσω του συστήματος IoT. Από τη στιγμή που μπορούμε να ελέγχουμε όλες τις συσκευές μας μόνο μέσα από μία κεντρική συσκευή, όπως για παράδειγμα το τηλέφωνό μας, μας προσφέρεται μεγαλύτερη ευκολία και άνεση.

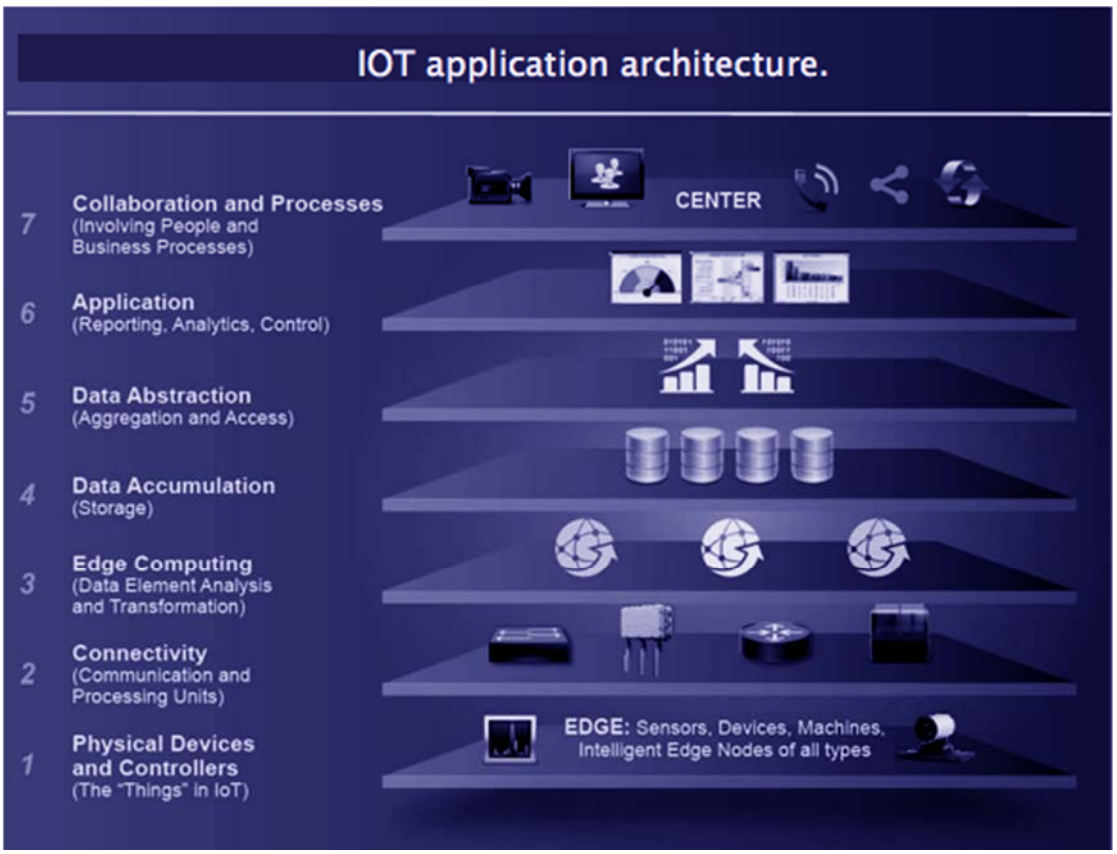
6. Η εξοικονόμηση χρόνου και χρήματος

Η έννοια του Διαδικτύου των πραγμάτων περιστρέφεται γύρω από την συλλογή δεδομένων και την αυτοματοποίηση εργασιών που απαιτούν λίγο ή καθόλου ανθρώπινη παρέμβαση, επιτρέποντας την εκτέλεση εργασιών γρηγορότερα και με βέλτιστη αξιοποίηση της ενέργειας. Έτσι ένα σύστημα IoT εξοικονομεί πολύτιμο χρόνο αλλά και χρήματα.

2.4 Αρχιτεκτονική (IoT)

Παρακάτω παρουσιάζονται τα επίπεδα ενός IoT συστήματος τα οποία και ακολουθήθηκαν και κατά την υλοποίηση της αναπτυσσόμενης συσκευής της διατριβής.

1. Το χαμηλότερο μέρος του IoT είναι οι αισθητήρες και οι ηλεκτρονικές συσκευές που είναι σε θέση να συνδεθούν και να κατανοήσουν γεγονότα.
2. Στο επόμενο στάδιο συναντάται η διασύνδεση των αισθητήρων και διαφόρων Modules. Εκτός από την διασύνδεση υπάρχει και η μετατροπή σε κατανοητή μορφή των δεδομένων που συλλέγονται, καθορίζοντας τις πληροφορίες που θα χρησιμοποιηθούν για την περεταίρω έξυπνη επιλογή που θα πραγματοποιηθεί σε ανώτερο επίπεδο.
3. Ακολουθεί η διασύνδεση με δίκτυο. Το στάδιο αυτό εξαρτάται από το πλαίσιο εφαρμογής του συστήματος IoT.
4. Το επίπεδο αυτό είναι ένα από τα σημαντικότερα στην αρχιτεκτονική του IoT και αφορά στην αποθήκευση των δεδομένων που συλλέχτηκαν από το δεύτερο στάδιο. Μπορεί να θεωρηθεί και ως στρώμα προστασίας αφού περιέχει όλα τα δεδομένα (ιστορικό) που αντιλήφθηκε το σύστημα.
5. Σε αυτό το επίπεδο χρησιμοποιούνται τα δεδομένα και οι πληροφορίες για την λήψη έξυπνης επιλογής ή για την εξαγωγή κάποιας έκθεσης προς το έκτο επίπεδο ή την εντολή πίσω στο δεύτερο επίπεδο προς κάποιον αισθητήρα ή controller, περνώντας από την αποθήκευση (επίπεδο 4, ιστορικό).
6. Το επόμενο επίπεδο, θεωρείται επίπεδο παρουσίασης και επίπεδο ελέγχου που επίσης στέλνει εντολές πίσω στο επίπεδο 2 (αισθητήρες).
7. Τελευταίο και ανώτερο επίπεδο της αρχιτεκτονικής είναι το στρώμα με το UI (User Interface). Ένα επίπεδο που εμπλέκεται ο χρήστης / καταναλωτής.



Εικόνα 2.2 – Αρχιτεκτονική ενός IoT συστήματος

2.5 Ο τρόπος λειτουργίας του οικοσυστήματος IOT

Με το Διαδίκτυο έχει αλλάξει ο τρόπος που εργαζόμαστε και επικοινωνούμε μεταξύ μας, χρησιμοποιώντας την σύνδεση μας με το World Wide Web (Internet). Το IoT έχει ως στόχο να εκμεταλλευτεί αυτή την σύνδεση πηγαίνοντας την σε ένα άλλο επίπεδο με τη προσθήκη και σύνδεση πολλαπλών συσκευών ταυτόχρονα, διευκολύνοντας έτσι αλληλεπιδράσεις όπως είναι άνθρωπος - μηχανή και μηχανή - μηχανή.

Το μεγαλύτερο πλεονέκτημα του οικοσυστήματος IoT είναι ότι δεν περιορίζεται σε ένα συγκεκριμένο τομέα, αλλά έχει εφαρμογές σε τομείς οικιακού αυτοματισμού, αυτοματισμού οχημάτων, εργοστασιακή αυτοματοποίηση της γραμμής παραγωγής, στην ιατρική, στο λιανικό εμπόριο, στην υγειονομική περίθαλψη και πολλά άλλα.

Τα θεμελιώδη στοιχεία ενός IoT συστήματος είναι τέσσερα:

- Αισθητήρες/Συσκευές
- Συνδεσιμότητα
- Αποθήκευση και επεξεργασία
- Περιβάλλον χρήστη

Εικόνα 2.3 - Τέσσερα θεμελιώδη στοιχεία του συστήματος IoT

2.5.1 Αισθητήρες / Συσκευές

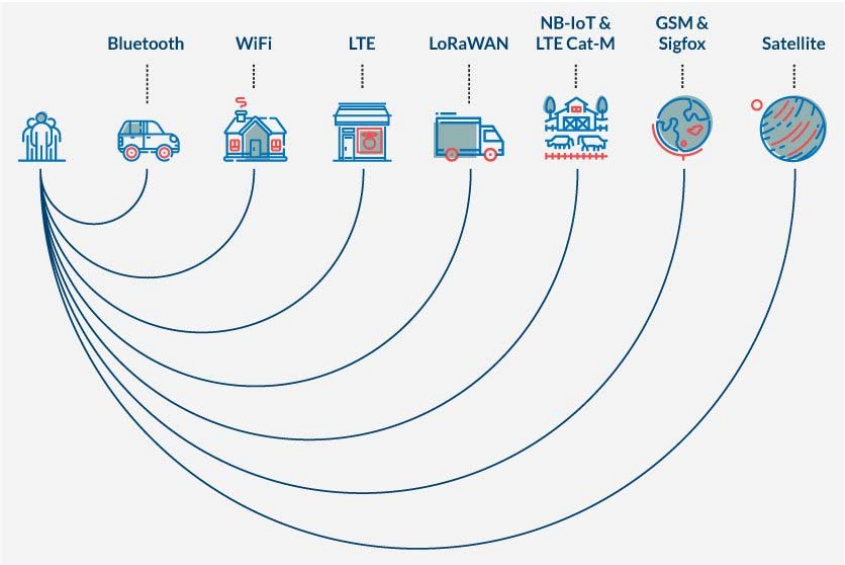
Σε πρώτο στάδιο οι αισθητήρες ή συσκευές συλλέγουν στοιχεία από τον περιβάλλοντα χώρο. Τα δεδομένα που συλλέγονται θα μπορούσε να είναι απλά όπως μια γεωγραφική θέση ή πιο σύνθετα όπως στοιχεία πρώτης ανάγκης για την υγεία ενός ασθενούς.

Για να καταγράψει πιο ευαίσθητα ή πολυπλοκότερα δεδομένα, μπορούν να συνδυαστούν πολλαπλοί αισθητήρες μαζί οι οποίοι θα είναι μέρος μιας συσκευής. Για παράδειγμα, το κινητό τηλέφωνο είναι μια συσκευή με αρκετούς ενσωματωμένους αισθητήρες, όπως GPS, κάμερα, επιταχυνσιόμετρο (accelerometer) κ.α. που χωρίς αυτούς δεν είναι σε θέση να «αισθανθεί» πράγματα. Είτε λοιπόν είναι ένας αυτόνομος αισθητήρας είτε μια συσκευή που έχει πολλαπλούς αισθητήρες, το πρώτο βήμα περιλαμβάνει τη τακτική συλλογή δεδομένων από τον περιβάλλοντα χώρο.

Εικόνα 2.4 - IoT Components – Sensors/Devices

2.5.2 Συνδεσιμότητα

Όταν συλλεχτούν τα δεδομένα, αποστέλλονται σε υποδομές cloud, γνωστές και ως πλατφόρμα Διαδικτύου των Πραγμάτων, με την χρήση διαφόρων τεχνολογιών δικτύωσης. Αυτές μπορεί να είναι ασύρματες ή ενσύρματες όπως το Bluetooth, Wi-Fi, κυψελοειδή δίκτυα, δορυφορικά δίκτυα, LoRaWAN, Ethernet, κλπ. Αν και κάθε μία από αυτές τις επιλογές συνδεσιμότητας αντιπροσωπεύει μια αντίστροφη σχέση μεταξύ της κατανάλωσης ηλεκτρικής ενέργειας, το εύρος σύνδεσης, και το εύρος ζώνης, είναι σημαντική η καλύτερη επιλογή σύνδεσης η οποία εξαρτάται αποκλειστικά από το επίπεδο πολυπλοκότητας και τις ειδικές απαιτήσεις που αναμένεται από το υλοποιούμενο σύστημα IoT.



Εικόνα 2.5 – Κυριότερες τεχνολογίες δικτύωσης

2.5.2.1 IoT Τεχνολογία & Πρωτόκολλα

Αρκετά πρωτόκολλα επικοινωνίας και τεχνολογίας χρησιμοποιούνται στο Διαδίκτυο των Πραγμάτων ώστε να καλύψουν τις ειδικές λειτουργικές απαιτήσεις ενός συστήματος IoT. Τα σημαντικότερα από αυτά είναι το Bluetooth, Zigbee, Radio Protocols (όπως το Z-Wave), Wi-Fi, Cellular, και LoRaWAN.

Bluetooth

Ένα σημαντικό μικρής εμβέλειας IoT πρωτόκολλο επικοινωνίας. Το Bluetooth, το οποίο έχει γίνει πολύ σημαντικό στην πληροφορική και σε πολλά καταναλωτικά προϊόντα, χρησιμοποιείται πολύ με wearable προϊόντα για να διασυνδέει IoT συστήματα σε πολλές περιπτώσεις μέσω ενός smartphone ή ενός Bluetooth / Network Hub. Το νέο Bluetooth χαμηλής ενέργειας (BLE) - ή Bluetooth Smart είναι ένα σημαντικό πρωτόκολλο για εφαρμογές IoT. Ενώ προσφέρει παρόμοια εμβέλεια με το Bluetooth έχει σχεδιαστεί για να καταναλώνει σημαντικά μειωμένη ενέργεια.

Zigbee

Το ZigBee είναι παρόμοιο με το Bluetooth και χρησιμοποιείται κυρίως σε βιομηχανικές εφαρμογές. Έχει κάποια σημαντικά πλεονεκτήματα σε πολύπλοκα συστήματα, όπως να παρέχει λειτουργία χαμηλής ισχύος, υψηλή ασφάλεια, να επωφελείται από τα ασύρματα δίκτυα ελέγχου και αισθητήρων σε εφαρμογές IoT. Η τελευταία έκδοση του ZigBee είναι η πρόσφατα ανακοινωθέν έκδοση 3.0, η οποία είναι ουσιαστικά η ενοποίηση των διάφορων ασύρματων προτύπων ZigBee σε ένα ενιαίο πρότυπο.

Z-Wave

Το Z-Wave είναι μια χαμηλής ισχύος RF επικοινωνιών τεχνολογία IoT που σχεδιάστηκε κατά κύριο λόγο για την οικιακή αυτοματοποίηση προϊόντων όπως οι ελεγκτές φωτισμού και αισθητήρες ανάμεσα σε πολλές άλλες συσκευές. Το Z-Wave χρησιμοποιεί ένα απλούστερο πρωτόκολλο συγκριτικά με άλλα, επιτρέποντας του την ταχύτερη και απλούστερη ανάπτυξη εφαρμογών. Μειονεκτεί στο ότι μόνο μια κατασκευάστρια εταιρία, η Sigma Designs, υλοποιεί τα συγκεκριμένα τσιπ.

Wi-Fi

Η συνδεσιμότητα Wi-Fi είναι ένα από τα πιο δημοφιλή πρωτόκολλα επικοινωνίας Διαδικτύου των Πραγμάτων, συχνά η πιο προφανής επιλογή για πολλούς προγραμματιστές, ιδίως αν ληφθεί υπόψη η αυξημένη διαθεσιμότητα των Wi-Fi στο εσωτερικό περιβάλλον του σπιτιού μέσα σε τοπικά δίκτυα. Υπάρχει μια μεγάλη υπάρχουσα υποδομή που υιοθετείται ταχύτατα από την τεχνολογία IoT, καθώς προσφέρει γρήγορη μεταφορά δεδομένων και την ικανότητα να χειριστεί μεγάλες ποσότητες δεδομένων.

Cellular

Κάθε υλοποίηση IoT που απαιτεί τη λειτουργία του με μεγαλύτερες αποστάσεις επωφελείται από τις δυνατότητες κυψελωτής επικοινωνίας GSM / 3G / 4G. Παρόλο που μπορεί ένα δίκτυο κινητής επικοινωνίας να διαχειριστεί μεγάλες ποσότητες δεδομένων, ανεβάζει το κόστος και την κατανάλωση

ενέργειας σε υψηλό επίπεδο. Είναι ιδανική όμως η χρήστη του για εφαρμογές με αισθητήρες βασισμένοι σε χαμηλό εύρος ζώνης που στέλνουν πολύ μικρές ποσότητες δεδομένων μέσω του Διαδικτύου.

LoRaWAN

Το LoRaWAN είναι ένα από τα δημοφιλέστερα πρωτόκολλα επικοινωνίας στην IoT τεχνολογία επικεντρώνοντας σε wide-area network (WAN) εφαρμογές. Το LoRaWAN σχεδιάστηκε για παροχή ευρυζωνικών δικτύων χαμηλής ισχύος με χαρακτηριστικά ειδικά για την υποστήριξη χαμηλού κόστους ασφαλή επικοινωνία στο Διαδίκτυο των Πραγμάτων και δυνατότητα σύνδεσης εκατομμυρίων συσκευών με ρυθμούς μετάδοσης δεδομένων να κυμαίνονται από 0,3 kbps έως 50 kbps. Συναντάται σε εφαρμογές που σχετίζονται με θέματα έξυπνης πόλης και βιομηχανικές εφαρμογές.

2.5.3 Αποθήκευση και επεξεργασία

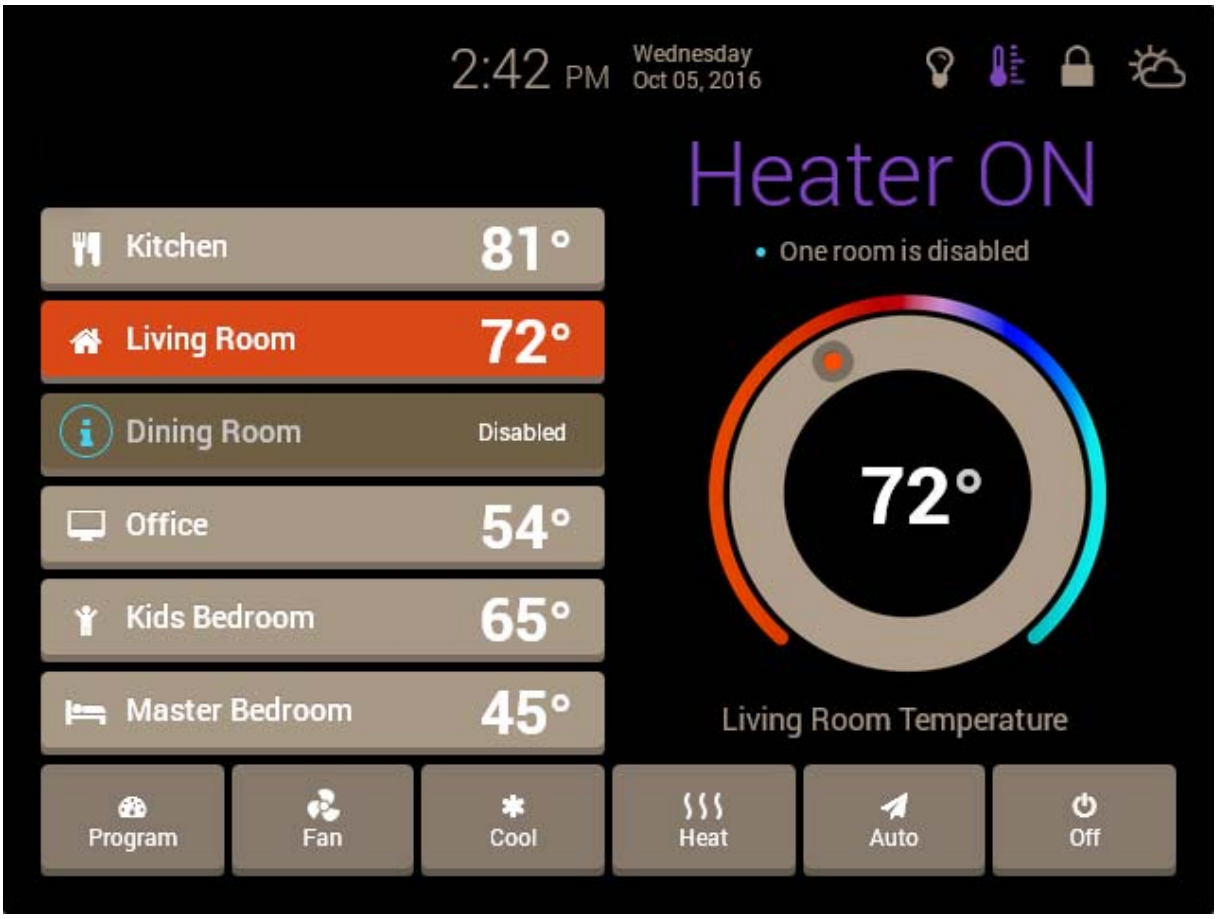
Μετά τα στάδια της συλλογής και της συνδεσιμότητας, τα δεδομένα έχουν αποθηκευτεί και χρησιμοποιούνται με την διαδικασία ανάλυσης τύπου Big Data Analytics Engine. Όπως φαίνεται και από το όνομα, είναι μια πολύπλοκη διαδικασία εξέτασης, που αφορά μεγάλες και ποικίλες ομάδες δεδομένων, με στόχο να αποκαλύψει πληροφορίες και να συνεισφέρει στην καλύτερη λήψη αποφάσεων.

Η ανάλυση αυτή μπορεί να είναι απλής μορφής, όπως ο έλεγχος κατά πόσον ή όχι η ένδειξη θερμοκρασίας σε ένα AC ή θερμάστρα είναι εντός ενός αποδεκτού εύρους, ή πιο σύνθετης μορφής όπως είναι ο εντοπισμός αντικείμενων με την χρήση video, για παράδειγμα ο εντοπισμός εισβολέων σε σπίτι με τη βοήθεια καμερών επιτήρησης. Τα επεξεργασμένα δεδομένα στη συνέχεια χρησιμοποιούνται για την εκτέλεση άμεσων αποφάσεων. Αποτέλεσμα αυτής της διαδικασίας είναι να υπάρχουν ευφυής ενέργειες που μετατρέπουν απλές συσκευές σε έξυπνες συσκευές.

2.5.4 Περιβάλλον χρήστη

Το τελευταίο βήμα περιλαμβάνει την ενημέρωση του τελικού χρήστη, διαθέτοντάς του τις πληροφορίες που συλλέχτηκαν με διάφορους τρόπους όπως:

- με την ενεργοποίηση ειδοποιήσεων σε τηλέφωνα ή κοινοποίηση μέσω κείμενων με μηνύματα ηλεκτρονικού ταχυδρομείου.
- με διεπαφή του χρήστη μέσω της οποίας μπορεί να ελέγξει ενεργά το σύστημά IoT (εικόνα 2.6).



Εικόνα 2.6 - Παράδειγμα από περιβάλλον χρήστη που ελέγχει κλιματιστικές συσκευές σε 6 διαφορετικούς χώρους του σπιτιού,

2.6 Εφαρμογές Ενέργειας

Στα πλεονεκτήματα από το Διαδίκτυο των Πραγμάτων συμπεριλαμβάνονται και οι περιοχές κατανάλωσης ενέργειας προσφέροντας μια μεγάλη ποικιλία λειτουργιών παρακολούθησης και ελέγχου της ενέργειας, με εφαρμογές που αφορούν στην εμπορική και οικιακή χρήση ενέργειας.

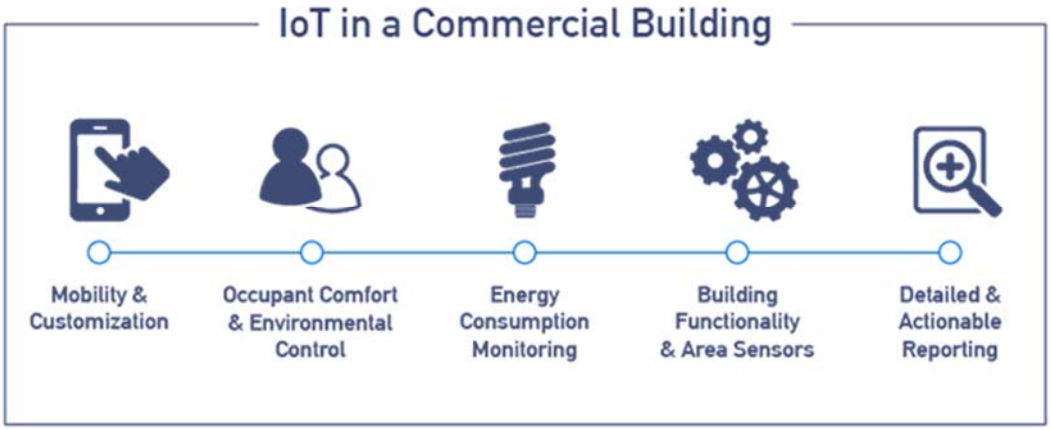
2.6.1 Αξιοπιστία - Reliability

Το σύστημα IoT μέσα από πρακτικές πληροφορίες και αναλυτικά στοιχεία συμβάλλει στη διασφάλιση της αξιοπιστίας του συστήματος. Εκτός από αποδοτική κατανάλωση, το IoT αποτρέπει την υπερφόρτωση του συστήματος. Το σύστημα προστατεύεται από σενάρια ζημιών, όπως κατεστραμμένος εξοπλισμός, downtime από απώλεια επικοινωνίας είτε με το δίκτυο είτε με αισθητήρες, έγκαιρη ανίχνευση απειλών και διατήρηση της σταθερότητας του συστήματος.

2.6.2 Εμπορική Ενέργεια - Commercial Energy

Οι μεγάλες ενεργειακές ανάγκες των επιχειρήσεων οδηγούν σε ενεργειακή σπατάλη, που μπορεί εύκολα να επηρεάσει τις επιχειρήσεις σε σημαντικό βαθμό. Αυτές οι επιχειρήσεις μπορεί να παραδίδουν ένα προϊόν με μικρότερα περιθώρια κέρδους ή να παλεύουν με το κόστος εξισορρόπησης της επιχείρησης. Μεγαλύτεροι επιχειρηματικοί οργανισμοί αναγκάζονται να παρακολουθούν ένα τεράστιο, πολύπλοκο οικοσύστημα της ενεργειακής τους κατανάλωσης που προσφέρει λίγες αποτελεσματικές λύσεις πάνω στην διαχείριση της χρήσης της ενέργειας.

Το IoT, διατηρώντας παράλληλα χαμηλό κόστος και υψηλό επίπεδο ακρίβειας απλοποιεί τη διαδικασία της παρακολούθησης και διαχείρισης της ενέργειας. Όλα τα σημεία κατανάλωσης ενός οργανισμού / επιχείρησης εξετάζονται προσεκτικά ενημερώνοντας και βελτιστοποιώντας τόσο την παραγωγή όσο και την διαχείριση της κατανάλωσης με εξοικονόμηση κόστους, με τον ίδιο τρόπο που επισημαίνει και ελέγχει λειτουργικά προβλήματα.



Εικόνα 2.7 - Internet of Things Energy Applications – Εμπορική Ενέργεια

2.6.3 Ενέργεια Κατοικίων - Residential Energy

Το IoT προσφέρει τρόπους για την ανάλυση και βελτιστοποίηση της χρήσης όχι μόνο σε ολόκληρο το σύστημα του σπιτιού, αλλά και σε ένα συγκεκριμένο επίπεδο μιας συσκευής. Αυτό μπορεί να κυμαίνεται από πολύ βασικές λειτουργίες, όπως η εξασθένιση του φωτισμού, ή η απενεργοποίησή του ή αλλαγή των ρυθμίσεων μιας συσκευής για να επιτύχει τη βελτιστοποίηση ενέργειας, έως πιο σύνθετες, όπως να εντοπίζει και να ενημερώνει για καταναλώσεις που είναι «προβληματικές» και ενδεχομένως να οφείλονται σε έλλειψη συντήρησης ηλεκτρικών συσκευών, ή σε κατεστραμμένες συσκευές ή ελαττωματικά εξαρτήματα.

Κεφάλαιο 3
Παρουσίαση – επιλογή τεχνολογιών

3.1 Περιγραφή των IR Κυμάτων

Η χρήση υπέρυθρων σημάτων είναι μια ευρέως χρησιμοποιουμένη ασύρματη τεχνολογία με πολλές εφαρμογές. Τα πιο γνωστά παραδείγματα στην καθημερινή ζωή είναι τα τηλεχειριστήρια για τηλεόραση, ήχο-συστήματα και κλιματιστικά καθώς και οι αισθητήρες κίνησης και τα θερμόμετρα με υπέρυθρες.

Κατά την εφαρμογή ενός τυπικού συστήματος υπέρυθρης επικοινωνίας απαιτείται ένας πομπός IR και ένας δέκτης IR. Ένας πομπός υπέρυθρων, ή πομπός IR, είναι το υλικό που αποστέλλει πληροφορίες από ένα τηλεχειριστήριο υπέρυθρων με άλλη συσκευή όπου υπάρχει ο δέκτης IR και είναι υπεύθυνος για την λήψη και αποκωδικοποίηση των σημάτων. Σε γενικές γραμμές, ο πομπός εξάγει ένα μοναδικό κώδικα που προσδιορίζει το υπέρυθρο σήμα που μεταδίδει. Αυτός ο κώδικας χρησιμοποιείται στη συνέχεια για να μετατραπούν τα σήματα από το τηλεχειριστήριο σε μια μορφή που μπορεί να γίνει

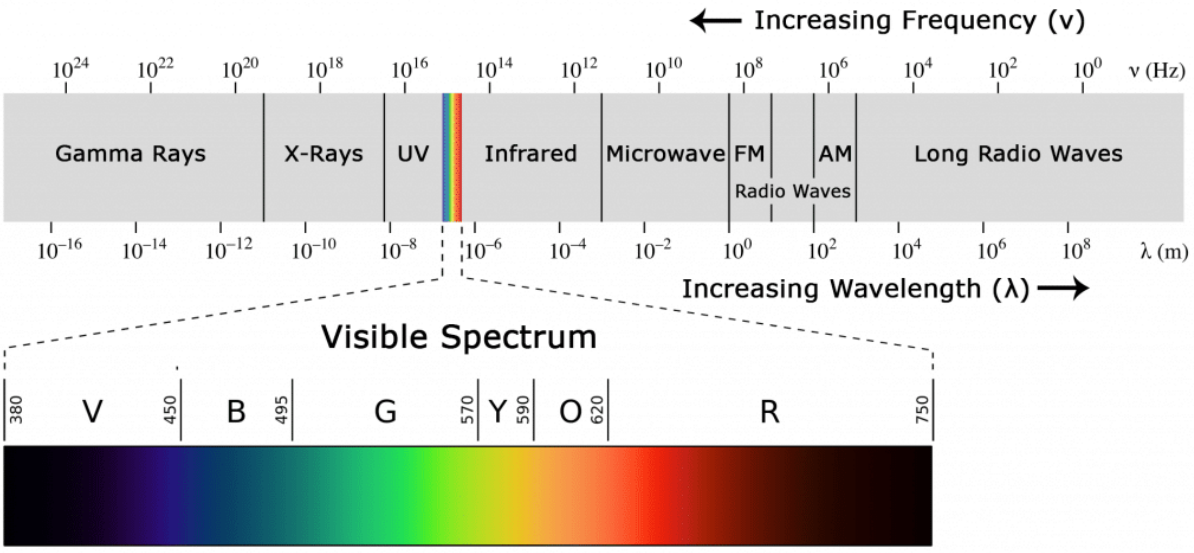
κατανοητή από την άλλη συσκευή. Επειδή το υπέρυθρο σήμα είναι στην ουσία φως, απαιτείται line-of-sight προβολή για την καλύτερη δυνατή λειτουργία, αλλά μπορεί και να αντανακλάται από στοιχεία του περιβάλλοντος χώρου, όπως το γυαλί και τους τοίχους.

Δεν υπάρχουν νομικοί περιορισμοί για τη χρήση IR δέσμης και μπορούν να μεταδίδουν οποιουδήποτε είδους δεδομένων.

Οι κακώς τοποθετημένοι δέκτες IR σε μια συσκευή μπορεί να οδηγήσουν στο φαινόμενο που ονομάζεται «tunnel vision», δηλαδή τη μείωση της περιοχής λειτουργίας ενός απομακρυσμένου ελέγχου, επειδή οι δέκτες είναι τοποθετημένοι αρκετά μέσα στο σασί μιας συσκευής.

3.2 Ανάλυση των διαφορετικών IR κυμάτων που υπάρχουν

Η υπέρυθρη ακτινοβολία είναι μια μορφή φωτός παρόμοιο με το φως που βλέπουμε γύρω μας. Η μόνη διαφορά μεταξύ του IR φως και του ορατού φως είναι η συχνότητα και το μήκος κύματος. Η υπέρυθρη ακτινοβολία βρίσκεται εκτός της περιοχής του ορατού φωτός, έτσι ώστε οι άνθρωποι δεν μπορούν να τη δουν. Επειδή το IR είναι ένα είδος φωτός, η επικοινωνία IR απαιτεί μια άμεση οπτική επαφή από τον δέκτη στον πομπό. Δεν μπορεί να μεταδοθεί μέσα από τοίχους ή άλλα υλικά, όπως γίνετε με άλλους τρόπους ασύρματης επικοινωνίας πχ. το Wi-Fi ή το Bluetooth.



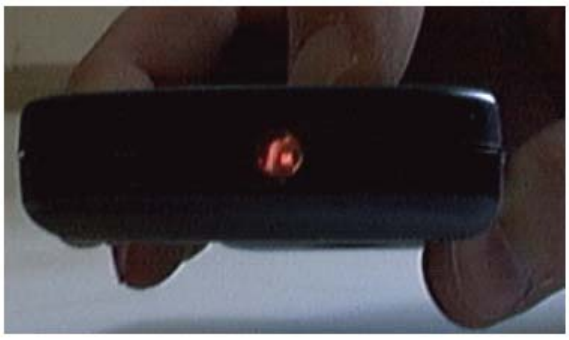
Εικόνα 3.1 - Φάσμα υπέρυθρης ακτινοβολίας

Σε ένα σύστημα επικοινωνίας με υπέρυθρα κύματα, ανάλογα την διαφορετική συχνότητα που χρησιμοποιείται έχουμε αντίστοιχα πλεονεκτήματα στην απόσταση που μπορούν να επικοινωνήσουν πομπός και δέκτης αλλά και στον ρυθμό μετάδοσης δεδομένων.

Ο πομπός μοιάζει με ένα τυπικό λαμπτήρα LED, εκτός του ότι παράγει φως στο φάσμα IR αντί του ορατού φάσματος. Αν παρατηρήσουμε στο μπροστινό μέρος ενός τηλεκοντρόλ, θα δούμε το λαμπτήρα LED που είναι ο πομπός IR.

Εικόνα 3.2 - Εικόνα τηλεχειριστηρίου κλιματιστικής μονάδας που χρησιμοποιήθηκε

Ανάλογα με την συχνότητα που εκπέμπει ο πομπός δημιουργούνται και τα αντίστοιχα χρωματικά αποτελέσματα (εικόνα 3.4). Επειδή το φως που εκπέμπεται κατά τη μετάδοση του σήματος δεν είναι ορατό, όπως προαναφέρθηκε, μπορούμε να το παρατηρήσουμε με την χρήση μιας κάμερας ενός κινητού τηλεφώνου καταγράφοντας τον πομπό την ώρα της μετάδοσης. Στην αναπτυσσόμενη συσκευή της διατριβής η μετάδοση του πομπού μας είναι χρωματικά στο μπλε χρώμα.



Εικόνα 3.3 - Στιγμιότυπο μετάδοση σήματος πορτοκαλί και μωβ από τηλεχειριστήριο τηλεόρασης και κλιματιστικού

Εικόνα 3.4 - Χρωματικές διαφορές ανάλογα την συχνότητα.

Ο δέκτης IR είναι μια φωτοδίοδος για αυτό και το μαύρο χρώμα του δέκτη. Η φωτοδίοδος είναι μια συσκευή ημιαγωγών που μετατρέπει το IR φως σε ένα ηλεκτρικό σήμα, το οποίο παράγεται όταν τα φωτόνια απορροφούνται από την φωτοδίοδο. Ένας IR δέκτης συνήθως μοιάζει με την παρακάτω εικόνα.

Εικόνα 3.5 - Δέκτης IR

Στα παραδοσιακά τηλεχειριστήρια όλων των ηλεκτρονικών συσκευών, οι πληροφορίες που μεταδίδονται αντιστοιχούν ανά πλήκτρο τηλεχειριστηρίου. Σε αντίθετη περίπτωση τα τηλεχειριστήρια των AC κωδικοποιούν και αποστέλλουν όλες τις παραμέτρους της συσκευής ως μια εντολή. Αναλυτικότερα κάθε πλήκτρο ενός τηλεχειριστηρίου από κλιματιστικό μεταδίδει σε μια

εκπομπή σήματος όλη την κατάσταση της συσκευής που θα πρέπει να έχει, συμπεριλαμβάνει δηλαδή την ενεργοποίηση ή απενεργοποίησή της, την θερμοκρασία που θα πρέπει να ρυθμιστεί, τον τρόπο λειτουργίας [mode cool, dry, heat, etc], την ένταση του ανεμιστήρα, την κατεύθυνση των περσίδων, κ.α.

Τα IR τηλεχειριστήρια χρησιμοποιούν ένα IR LED (τον πομπό) για να μεταδώσει το σήμα εκπομπής από τηλεχειριστήριο στον δέκτη που υπάρχει στην ηλεκτρική συσκευή. Αυτή η αποστολή επιτυγχάνεται με πολλές γρήγορες εναλλαγές του IR LED από κλειστό σε ανοιχτό. Το σήμα που αποστέλλετε είναι ένα PWM (Pulse-width modulation) σήμα που χρησιμοποιεί ορισμένες συχνότητες. Οι διαμορφώσεις των συχνοτήτων (frequency modulation) που χρησιμοποιούνται σε όλα σχεδόν τα τηλεχειριστήρια υπέρυθρων είναι 30kHz, 33kHz, 36khz, 38kHz, 40kHz και 56kHz. Εξαίρεση αποτελεί η χρήση των ακτίνων IR στα 455 kHz που χρησιμοποιείται αποκλειστικά από την Δανέζικη κατασκευαστική εταιρία υψηλής ακουστικής τεχνολογίας Bang & Olufsen. Τα εξαρτήματα για αυτήν τη ζώνη διαμόρφωσης είναι δύσκολο για κάποιον να τα προμηθευτεί, παρόλα αυτά η χρήση αυτής της διαμόρφωσης επιτρέπει περίπου 10 φορές μεγαλύτερη μετάδοση δεδομένων σε σχέση με των υπόλοιπων διαμορφώσεων IR.
Η διαμόρφωση σε ορισμένη συχνότητα είναι απλώς απαραίτητη για τη διάκριση του σήματος από άλλες κοντινές πηγές υπερύθρων, από το φως της ημέρας έως το φως λαμπτήρων φθορισμού.

Οι διαφορετικές συχνότητες έχουν και διαφορετικό ρυθμό μετάδοσης. Έτσι η διαμόρφωση των 33 kHz, που χρησιμοποιείται για παράδειγμα από το 1989 για μονοκατευθυντικές επικοινωνίες από επιστημονικές αριθμομηχανές HP προς τον θερμικό εκτυπωτή HP 82240B, έχει ρυθμό μετάδοσης που φτάνει στα 1170 Bd. Σε μια διαμόρφωση στα 38 kHz η μετάδοση των δεδομένων είναι 2400 Bd ή στις υπέρυθρες που αναφερόμαστε είναι 2400 pulses per second[3], στα 56 kHz η μεταφορά δεδομένων είναι στα 3600 Bd ενώ στα 455 kHz επιτυγχάνεται ρυθμός μετάδοσης που φτάνει στα 19200 Bd.
Στην αναπτυσσόμενη συσκευή της διατριβής η διαμόρφωση συχνότητας που χρησιμοποιείται είναι στα 38 kHz.

[3] https://en.wikipedia.org/wiki/Baud

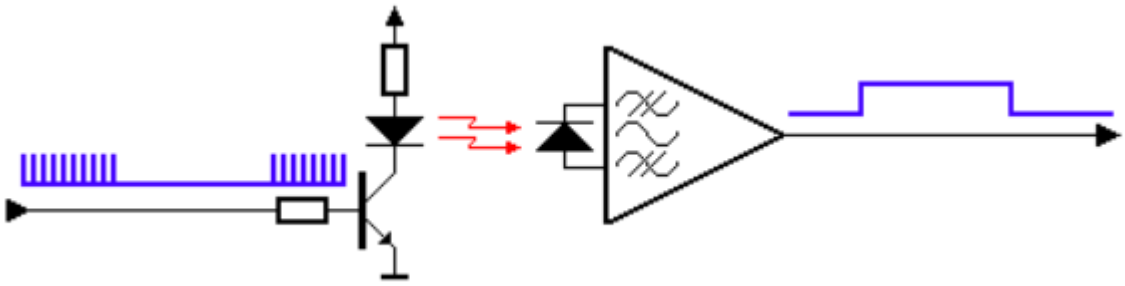
Συγκεντρωτικά οι διάφορες συχνότητες μαζί με παραδείγματα υιοθέτησης [4]

Band	Calibrated	Usage Examples
30 kHz	28800 Hz	"Lazer Tag"
33 kHz	32768 Hz	HP 82240B Printer, Canon RC-6
36 kHz		RC-5 (Philips)
38 kHz	38400 Hz	NEC Protocol, Nikon ML-L3
40 kHz		SIRC (Sony)
56 kHz	57600 Hz	Thomson / RCA
455 kHz	460800 Hz	Bang & Olufsen

Εικόνα 3.6 - Διάφορες συχνότητες για IR διαμορφώσεις με παραδείγματα

Με τη διαμόρφωση του σήματος σε μια φέρουσα συχνότητα κάνουμε την πηγή φωτός IR (τον πομπό) να αναβοσβήνει σε μια συγκεκριμένη συχνότητα και να ξεχωρίζει. Ο δέκτης IR θα συντονιστεί σε αυτή τη συχνότητα, οπότε και αγνοεί όλες τις άλλες πηγές.

Στην συνέχεια παρουσιάζεται (εικόνα 3.7) πως ένα διαμορφωμένο σήμα ενεργοποιεί το IR LED του πομπού στην αριστερή πλευρά. Αναπαράγετε το παλμικό σήμα (ανάβοντας και σβήνοντας το φως IR στην συγκεκριμένη συχνότητα) και ο δέκτης ανιχνεύει και λαμβάνει το σήμα (στη δεξιά πλευρά).

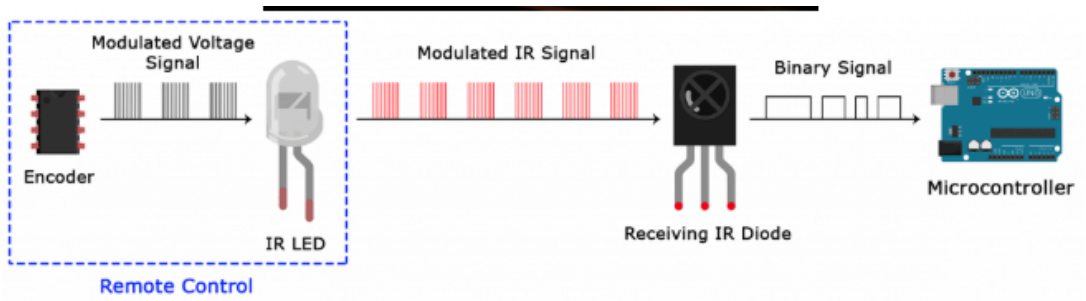


Εικόνα 3.7 - Αναπαράσταση παλμικού σήματος (πομπός - > σήμα -> δέκτης)

Η ίδια διαδικασία λίγο ποιο επεξηγηματική παρουσιάζεται στην παρακάτω εικόνα (εικόνα 3.8), όπου ένα τηλεχειριστήριο εκπέμπει ένα σήμα το οποίο λαμβάνει ο δέκτης που είναι εγκατεστημένος πάνω στην IoT συσκευή. Αυτή η διαδικασία υλοποιήθηκε και στα βήματα της μεθοδολογίας που ακολουθήθηκε στη διατριβή, ώστε να αντιγράψουμε το τηλεχειριστήριο. Αναλυτικά στην

[4] http://www.numericana.com/answer/ir.htm

διαμόρφωση σήματος IR, ένας κωδικοποιητής μετατρέπει ένα δυαδικό σήμα σε ένα διαμορφωμένο ηλεκτρικό σήμα. Αυτό το ηλεκτρικό σήμα στέλνεται στο IR LED (πομπός). Ο πομπός μετατρέπει το διαμορφωμένο ηλεκτρικό σήμα σε ένα διαμορφωμένο σήμα IR. Ο δέκτης IR στην συνέχεια αποδιαμορφώνει το σήμα IR και το μετατρέπει πίσω σε δυαδικό, όπου το παραλαμβάνει ο μικροεπεξεργαστής και ανάλογα αξιοποιεί την πληροφορία. Στην περίπτωση της υλοποίησης μας, διαφοροποιούμαστε στο τελευταίο στάδιο όπου ο δέκτης IR μετατρέπει σε δυαδικό το σήμα IR. Λόγω ιδιαιτερότητας του τηλεχειριστηρίου κρατήθηκε η μορφή του σήματος σε IR μορφοποίηση και χρησιμοποιήθηκε αυτούσια ως αναπαραγωγή.



Εικόνα 3.8 - Βήματα διαμόρφωσης από το τηλεχειριστήριο στο δεκτή

3.3 Ανάλυση του χώρου εφαρμογής

Για καλύτερη ανάλυση των μετρήσιμων δεδομένων έπρεπε να επιλεχτεί για έρευνα ένα μηχάνημα κλιματιστικού το οποίο θα λειτουργούσε όσο το δυνατόν περισσότερες ώρες και το οποίο θα προσέφερε περισσότερα δεδομένα μέτρησης σε σχέση με ένα κλιματιστικό οικιακής χρήσης που δεν θα μπορούσε να λειτουργεί όλο το 24ωρο.

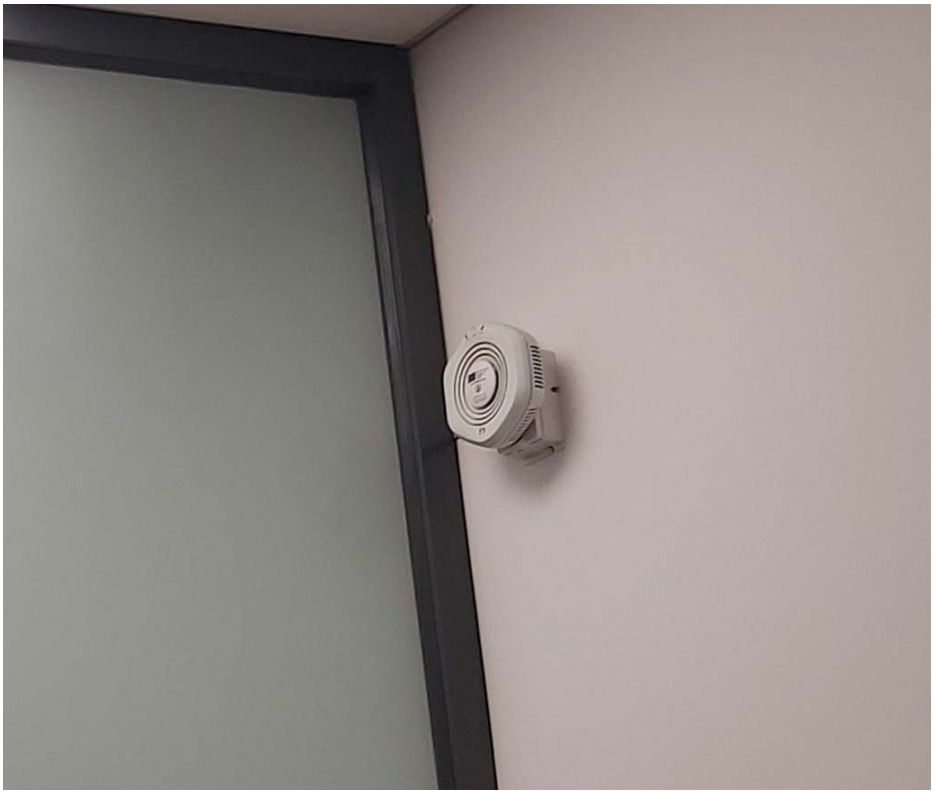
Στον χώρο εργασίας που επιλέχτηκε[5] υπάρχουν 4 δωμάτια, τα οποία περιέχουν δικτυακό εξοπλισμό, διακομιστές και κτιριακό UPS, και τα οποία διαθέτουν κλιματισμό που λειτουργεί όλο το 24ωρο. Ένα από αυτά τα δωμάτια επιλέχτηκε για τις δοκιμές με απώτερο σκοπό την ενεργειακή βελτίωση της κατανάλωσης του κλιματιστικού. Ο εξοπλισμός (ενεργά στοιχεία) που εμπεριέχεται στο χώρο εφαρμογής είναι 2 μεγάλες rack ντουλάπες με τον τερματισμό της δικτυακής καλωδίωσης του 3ου , 4ου και 5ου ορόφου του κτιρίου, οι κεντρικοί δρομολογητές και ο τερματισμός της οπτικής ίνας, καθώς και τα τηλεφωνικά κέντρα που εξυπηρετούν το τρέχον κτίριο μαζί με κοντινά κτίρια της περιοχής (πχ. την φοιτητική εστία).

[5] Κτίριο Κεντρικής Βιβλιοθήκης Πανεπιστημίου Θεσσαλίας

Η συνεχής ψύξη του χώρου είναι επιτακτική για το λόγο ότι μέσα σε μια ώρα χωρίς αυτή ο εξοπλισμός παράγει θερμότητα που ανεβαίνει πάνω από τους 26^{o}C τους χειμερινούς μήνες, ενώ τους καλοκαιρινούς μήνες μπορεί να φτάσει στους 30 ^{o}C. Ο χώρος είναι ένα μικρό γυάλινο δωμάτιο με δική του κλιματιστική μονάδα (εικόνα 3.9 ,3.11).

Εικόνα 3.9 – Εξωτερική άποψη του χώρου εφαρμογής

Η εμβέλεια του τοπικού Wi-Fi δικτύου είναι σε όλο το κτίριο διαθέσιμη με Access Points, άρα δεν υπάρχει περιοχή που δεν έχει κάλυψη.



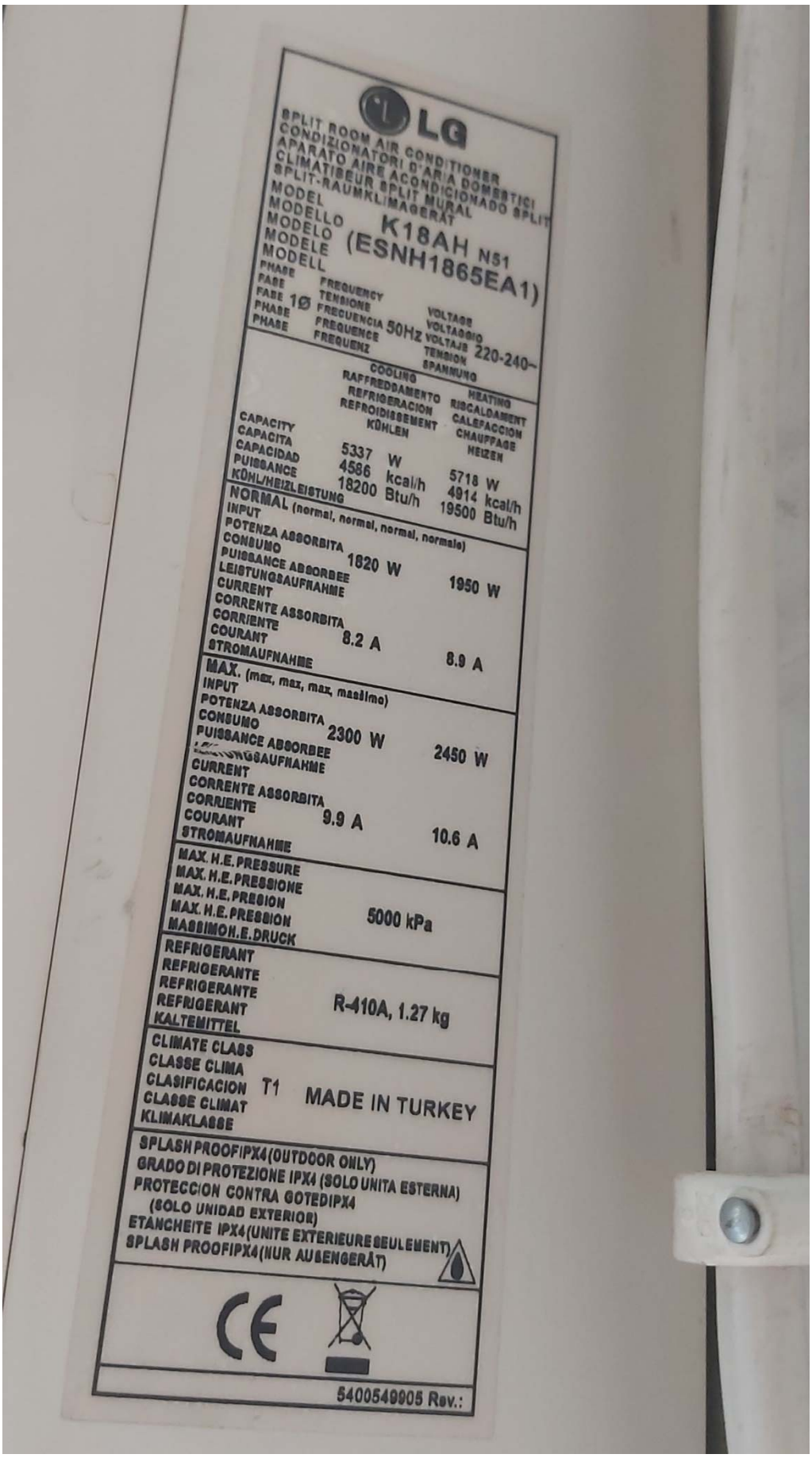
Εικόνα 3.10 - Access Points του χώρου εφαρμογής

Εσωτερικά, όπως προαναφέρθηκε, υπάρχει ο εξοπλισμός με το κλιματιστικό πίσω από της ντουλάπες Rack. Όπως διακρίνετε στην σχετική φωτογραφία (εικόνα 3.11) πάνω στις ντουλάπες είναι προσωρινά τοποθετημένοι δύο φορητοί υπολογιστές. Ο ένας καταγραφεί τα στοιχεία κατανάλωσης ενέργειας (καταγραφέας / energy data logger) που θα αναλυθούν στον εξοπλισμό σε μεταγενέστερη ενότητα, ενώ ο δεύτερος χρησιμοποιείται για την χρήση της κάμερας ώστε να υπάρχει οπτική επιβεβαίωση της σωστής αυτοματοποιημένης λειτουργίας με live streaming service, προσφέροντας εικόνα αλλά και video (εικόνα 3.12) με την χρήση του ελεύθερου λογισμικό Yawcam[6].
Τα ενεργειακά χαρακτηριστικά της κλιματιστικής μονάδας φαίνονται στην εικόνα 3.13.

Εικόνα 3.11 - Εσωτερική άποψη του χώρου εφαρμογής

Εικόνα 3.12 - Live streaming A/C status

[6] https://www.yawcam.com



Εικόνα 3.13 – Ενεργειακά χαρακτηριστικά μοντέλου κλιματιστικού του χώρου εφαρμογής

Ο χώρος μελέτης έχει ορισμένες επιπλέον ιδιαιτερότητες / προβλήματα:

- ο προσανατολισμός του δωματίου είναι ανατολικός , που σημαίνει ότι από την ανατολή του ηλίου και μέχρι της μεσημβρινές ώρες το δωμάτιο θερμαίνεται επιπλέον από τον ήλιο,.
- ο εξωτερικός τοίχος του δωματίου έχει ένα μεγάλο υαλοπίνακα (παράθυρο) που καλύπτει σχεδόν όλη την πλευρά του δωματίου. Αποτέλεσμα να προκαλεί επιπλέον αύξηση της θερμοκρασίας τις ηλιόλουστες ημέρες, ενώ σε αντίθετη περίοδο που έχει χαμηλότερες θερμοκρασίες να βοηθάει στην ψύξη του χώρου.

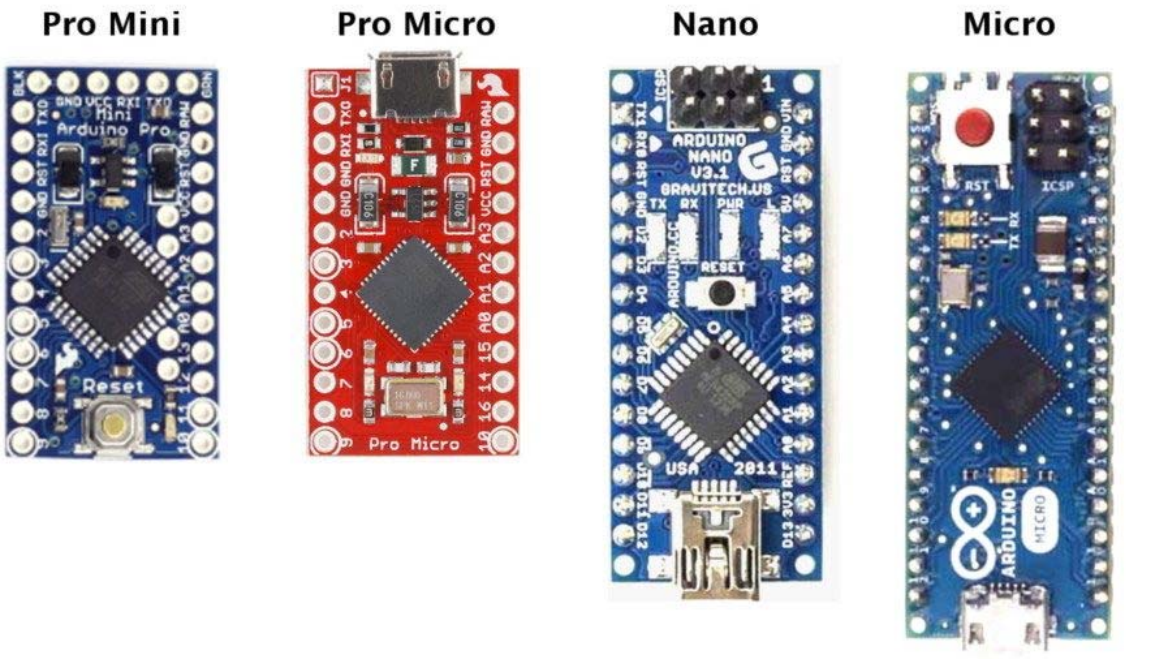
3.4 Καθορισμός και ανάλυση απαιτήσεων

Οι λειτουργικές απαιτήσεις του αμφίδρομου συστήματος που πρέπει να συμπεριλαμβάνονται στην υλοποίηση είναι:

- Τα δεδομένα των αισθητήρων που εξάγονται από την συσκευή πρέπει να αποθηκεύονται σε βάσεις δεδομένων με timestamp μέτρησης
- Τα δεδομένα που εισάγονται στην συσκευή (π.χ. εντολές αλλαγής θερμοκρασίας) πρέπει να αποθηκεύονται σε βάσεις δεδομένων με timestamp αποστολής
- Τα δεδομένα των μετρήσεων θα πρέπει να είναι διαθέσιμα και προσβάσιμα από το περιβάλλον της εφαρμογής για κινητά.
- Η αναπτυσσόμενη συσκευή θα πρέπει να συμπεριλαμβάνει και οπτική ειδοποίηση όταν στέλνει εντολή προς την κλιματιστική μονάδα.
- Η αναπτυσσόμενη συσκευή θα πρέπει να ηλεκτροδοτείται αυτόνομα.
- Η αναπτυσσόμενη συσκευή θα πρέπει να συνδέεται στην ασύρματη υποδομή του εκάστοτε χώρου εγκατάστασης.
- Η λειτουργία του IoT συστήματος θα πρέπει να είναι συνεχής και αδιάληπτη με προσθήκη διάφορων ελέγχων ασφαλείας.
- Το αναπτυσσόμενο περιβάλλον του χρήστη θα είναι αποκλειστικά για λειτουργικά συστήματα Android.

3.4.1 Επιλογή κατάλληλου εξοπλισμού

Υπάρχουν αρκετά μικρά board για ανάλογες IoT εφαρμογές, αρκετά από τα οποία προσφέρουν και δυνατότητες ασύρματης επικοινωνίας. Στην εικόνα που ακολουθεί παρουσιάζονται ενδεικτικά κάποια μικρά board.



Εικόνα 3.14 - Μικρά board για IoT υλοποιήσεις

Παρόλο που θα ήταν ενδιαφέρον σε μια αναπτυσσόμενη IoT εφαρμογή να γίνει χρήση τόσο μικρών boards, υπάρχουν αρκετοί λειτουργικοί περιορισμοί για την υιοθέτησή τους στη παρούσα διατριβή, μερικοί εκ των οποίων είναι:

- Τα προσφερόμενα volt περιορίζονται μόνο σε 3.3V και όχι στα 5V.
- Η τροφοδοσία γίνετε μόνο με ένα τρόπο (mini usb)
- Δεν δέχονται επιπλέων modules για επεκτάσεις (από τους σημαντικότερους περιορισμούς)
- Έχουν μειωμένο αριθμό ελεύθερων θέσεων που μπορούν να συνδεθούν αισθητήρες.
- Υπάρχει περιορισμός στη διαθέσιμη μνήμη
- Δεν υπάρχει η δυνατότητα επέκτασης λειτουργιών
- Είναι αρκετά ακριβά.

Η χρήση δηλαδή ενός εκ των παραπάνω μικρών boards που θα συνδεθεί με έναν αισθητήρα και με μια μπαταρία, παρόλο που θα είναι ένα φυσικό αποτέλεσμα μικρών διαστάσεων, θα μπορεί να συλλέξει περιορισμένα στοιχεία και για αυτό το λόγο συνήθως χρησιμοποιείται για να αποτελέσει μέρος ενός μεγαλύτερου συστήματος.

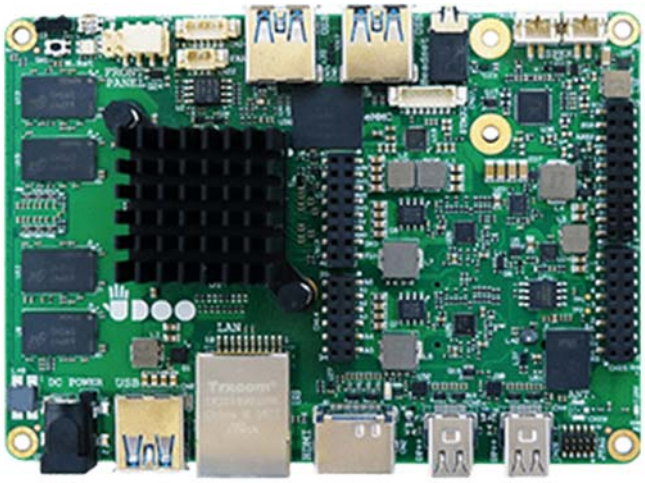
3.4.2 Σύγκριση διαθέσιμων προτάσεων hardware

Παρουσιάζοντας ορισμένα από τα κυριότερα boards με τα χαρακτηριστικά τους, καταλήγουμε στην τελική επιλογή ενός board για το αναπτυσσόμενο σύστημα.

3.4.2.1 UDOO x86

Χαρακτηριστικό αυτού του Board είναι ότι ενσωμάτωσε το Arduino Board πάνω στο αναπτυσσόμενο Board κερδίζοντας με αυτόν τον τρόπο πλήρης συμβατότητα με την μεγάλη αγορά, κοινότητα και αποδοχή του Arduino που έχει στον κόσμο των μικροεπεξεργαστών. Μπορεί να χρησιμοποιεί όλα τα ήδη σχεδιασμένα add-on module που έχουν αναπτυχθεί για Arduino Board. Επιπλέον για το λόγο ότι έχει και δυνατούς επεξεργαστές και χαρακτηριστικά μπορεί να λειτουργήσει και με γνωστά λειτουργικά όπως είναι τα Windows 7,8,10, όλες τις Linux διανομές τόσο οι 32bit όσο και οι 64bit εκδόσεις καθώς και android διανομές.

Έχει διαθέσιμες δύο διαφορετικές εκδόσεις[7], την ultra και την advance plus, ανάλογα τις απαιτήσεις.



Εικόνα 3.14 - UDOO x86

Χαρακτηριστικά των δύο διαφορετικών εκδόσεων		
	ULTRA	**ADVANCED PLUS**
processor	CPU INTEL PENTIUM N3710 UP TO 2.56 GHZ	CPU INTEL CELERON N3160 UP TO 2.24 GHZ
memory	8 GB DDR3L DUAL CHANNEL	4 GB DDR3L DUAL CHANNEL
graphics	INTEL HD GRAPHICS 405 UP TO 700 MHZ	INTEL HD GRAPHICS 400 UP TO 640 MHZ
storage	32GB EMMC STORAGE	32GB EMMC STORAGE

[7] https://www.udoo.org/udoo-x86/

Με κοινά χαρακτηριστικά των δύο εκδόσεων	
Processor	Intel® Braswell 64 bit 14 nm: Quad Core up to 2.56GHz
Cores	4
Memory	DDR3L soldered-down memory up to 8GB RAM dual channel
Graphics	Intel HD Graphics Gen 8 LP Up to 700 MHz Up to 16 execution units
Video Interfaces	1x HDMI 2x miniDP++ connectors
Mass Storage	eMMC disk up to 32 GB soldered on board Standard SATA connector M.2 Key B SSD slot Micro SD card slot
Networking	Gigabit Ethernet connector M.2 Key E slot for optional Wireless modules
USB	3 x USB 3.0 type-A sockets
Multimedia	HW Video decode: H.265/HEVC, H264, MPEG2, MVC, VC-1, WMV9, JPEG, VP8; HW Video encode: H.264, MVC, JPEG
Audio	Microphone + Headphone combo connector Speaker internal header S/PDIF output
Serial Ports	2x UART ports
Other Interfaces	IR interface LPC – 2 x I2C – GPIOs – Touch Screen Management signals on expansion connector RTC Battery + Connector Included
Digital I/O Pins	Up to 20 extended GPIOs, multiplexed with other interfaces
Operating System	Windows 10, 8.1, 7 Any Linux Distribution for X86 64bit platform Android X86

Dimensions	120mm x85mm (4.72″ x 3.35″)
Arduino χαρακτηριστικά του board	
Microcontroller	ATmega32U4
Other Interfaces	1x UART 1x i2C 1x SPI
Arduino Pinout	Arduino Leonardo-Compatible and compatible with most Arduino Shields 5V compliant.
Digital I/O Pins	Up to 23 x digital I/O (7 PWM)
Analog I/O Pins	12 x Analog Input

3.4.2.2 BeagleBoneBlack

Το συγκεκριμένο board είναι μικρότερο σε μέγεθος, σχεδιάστηκε από την εταιρία BeagleBone με ενσωματωμένο επεξεργαστή Sitara XAM3359AZCZ100 Cortex A8 ARM. Ο επεξεργαστής λειτουργεί στο 1 GHz και έχει επεξεργαστική ισχύ των 2000 MIPS με 512 MB DDR3L RAM λειτουργίας στα 800 MHz. Τέσσερα προγραμματιζόμενα LEDs είναι διαθέσιμα για σκοπούς δοκιμών. Διαθέτει επιπλέον έξοδο για οθόνη μέσω θύρας HDMI υποστηριζόμενης ανάλυσης μέχρι τα 1920 x 1080 @24 Hz. Το board έχει μέγιστη κατανάλωση 460 mA @ 5V. Διαθέτει 4 GB eMMC flash memory και επιπλέον προσφέρετε επέκταση μνήμης μέσω Micro SD card slot. Για την δικτύωση διαθέτει 10/100 Ethernet θύρα. Η παροχή σε όλες τις συνδέσεις I/O είναι 3,3V γεγονός που περιορίζει τη χρήση του

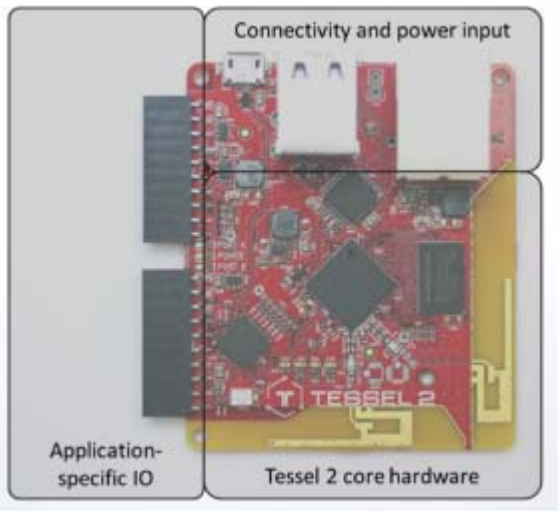
Εικόνα 3.15 - BeagleBoneBlack

Είναι συμβατό με λειτουργικά συστήματα Debian, Android, Ubuntu, και παρέχει cloud υποστήριξη με χρήστη του Cloud9 IDE (λογισμικού της εταιρίας).

3.4.2.3 Tessel 2

Το board αυτό έχει μικρό μέγεθος και έχει ενσωματωμένο 580 MHz Wi-Fi router SoC (αναπτυγμένο από την Mediatek), τρέχει OpenWRT, διαθέτει 64MB DDR2 RAM με 32 MB flash storage. Ο επεξεργαστής του είναι ένας 48 MHz ARM Cortex M0 Microcontroller (Atmel SAM D21). Διαθέτει μια 10/100 Ethernet port, 2 USB 2.0 ports, micro USB port για παροχή ενέργειας (power) και δύο module ports με 10 pins το κάθε ένα.

Υπάρχει ένας σημαντικός περιορισμός ανάπτυξης γιατί τα I/O modules του, δεν είναι συμβατά με board από άλλους κατασκευαστές και δεν μπορεί να δεχτεί και modules εκτός εταιρίας. Η λειτουργία του Tessel είναι με 3.3V και ο προγραμματισμός γίνετε με JavaScript μέσο Micro USB.



Εικόνα 3.16 - Tessel 2

Το Tessel board διαθέτει αρκετά έτοιμα modules τα οποία συνδέονται με το κεντρικό board. Ορισμένα από αυτά είναι τα παρακάτω και παρουσιάζονται αναλυτικά στο παράρτημα Ε.

- **Accelerometer**
- **Ambient Light+sound**
- **Climate**
- **GPS**
- **Infrared**
- **Relay**
- **RFID**

3.4.2.4 Intel® Edison Kit for Arduino

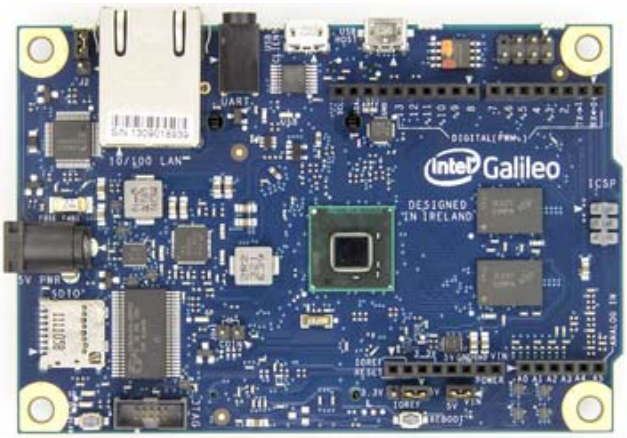
Σαν την πρώτη εταιρία που αναλύσαμε (UDOO x86), την ιδιά λογική υιοθέτησης Arduino board μέσα στο αναπτυσσόμενο board ακολουθεί και η intel με το Intel® Edison Kit για Arduino.

Εικόνα 3.17 - Intel® Edison Kit for Arduino

Το Intel® Edison Kit για Arduino παρέχει το Arduino 1.0 pinout με τυπική συνδεσιμότητα όπως ένα micro USB, μια θύρα USB OTG που μπορεί να εναλλάσσεται μεταξύ μιας δεύτερης σύνδεσης micro USB συσκευής, μια τυπική υποδοχή μεγέθους USB Type-A, μια SD θύρα, και ένα βύσμα συνεχούς ρεύματος. Όπως ένα Arduino Uno, το Intel® Edison Kit για Arduino καθιστά δυνατόν να έχουμε 20 ψηφιακές εισόδους / εξόδους, εκ των οποίων 6 μπορούν να χρησιμοποιηθούν ως αναλογικές είσοδοι. Το Intel® Edison έχει 4 εξόδους PWM που μπορεί να χρησιμοποιηθούν μέσω σύνδεσης με οποιαδήποτε από τις 6 ακίδες που υποστηρίζουν PWM στο Arduino Uno (ακίδες 3, 5, 6, 9, 10, ή 11).

Το Intel® Edison Kit για το Arduino έχει σχεδιαστεί για να είναι hardware και software συμβατό με τα Arduino shields (τα modules του Arduino) τα οποία έχουν σχεδιαστεί για χρήση με την τελευταία έκδοση το Arduino Uno R3. Οι ψηφιακές 0 ως 13 και αναλογικές 0 έως 5 είσοδοι, η power header, η ICSP header, και τα UART port pins (0 and 1), είναι όλα στις ίδιες θέσεις όπως στο Arduino Uno R3. Οι ψηφιακές και αναλογικές ακίδες μπορούν να ρυθμιστούν ώστε να λειτουργούν είτε σε 5V ή 3.3V, με έξοδο στα 24 mA με 3.3V και 32 mA με 5V.

Ιδίας ανάπτυξης από την Intel υπήρχε και το Intel Galileo το οποίο η εταιρία αποφάσισε να σταματήσει την διάθεση του. Ήταν πιο δυνατό από την έκδοση Edison αλλά λειτουργούσε μόνο με παροχή ρεύματος το οποίο αποτελεί μειονέκτημα για τις IoT εφαρμογές στις οποίες η εγκατάσταση απαιτείται να γίνει σε χώρους χωρίς ηλεκτροδότηση.



Εικόνα 3.18 - Intel Galileo

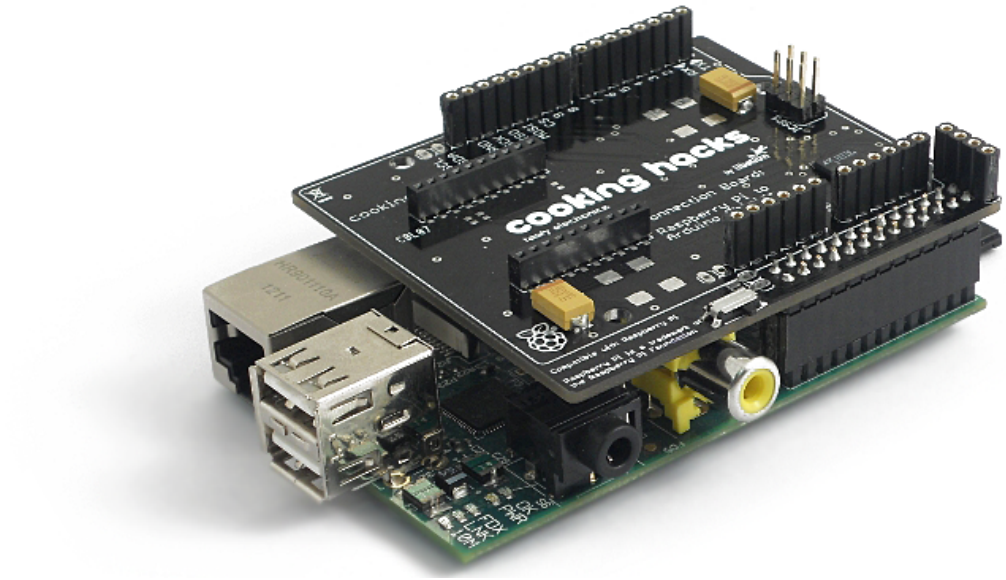
Έγινε μια προσπάθεια να ξανάκυκλοφορήσει με το Intel Galileo Gen 2, αλλά δεν κέρδισε εντυπώσεις αφού υπήρχαν προβλήματα στην απώλειά του αναπτυσσόμενου κώδικα, όπου χάνονταν μετά από απώλεια ενέργειας και αναγκάζονταν η συσκευή να ξανά φορτώνει τον κώδικα από εξωτερική κάρτα SD.

3.4.2.5 Raspberry Pi3

Τέλος παρουσιάζεται το Raspberry Pi3, ένα ιδιαίτερο board πολύ δυνατό και οικονομικό. Είναι το πιο ευέλικτο και κατάλληλο board για ανάπτυξη IoT. Επιλέγεται από μεγάλο μέρος των χρηστών λόγω του χαμηλού κόστους του, των πολλαπλών επιλογών συνδεσιμότητας που προσφέρει και το ευρύ φάσμα της υποστήριξης λειτουργικών συστημάτων (OS). Λόγω της δημοτικότητας του Raspberry Pi 3, πολλές βιβλιοθήκες με διάφορο υλικό είναι ήδη διαθέσιμα στο διαδίκτυο, καθιστώντας ευκολότερο για τους προγραμματιστές να ξεκινήσουν την ανάπτυξη συστημάτων.

Εικόνα 3.19 - Raspberry Pi3

Συγκρίνοντάς το με το Arduino, η χαρακτηριστική τους διαφορά είναι ότι το Arduino περιέχει μικροελεγκτή πάνω σε ένα board, ενώ Raspberry Pi είναι ένας μίνι υπολογιστής.
Με αναπτυσσόμενα connection bridge (με κόστος 130$) σαν αυτό που παρουσιάζεται στην παρακάτω εικόνα (εικόνα 3.20), δίνετε η δυνατότητα στο Raspberry να χρησιμοποιήσει οποιοδήποτε εξάρτημα υπάρχει για Arduino boards, αφού εξομοιώνει την βασική συνδεσμολογία του Arduino.



Εικόνα 3.20 - connection bridge Raspberry με Arduino εξαρτήματα

Η επιλογή του board για τη συσκευή υλοποίησης ήταν τέτοια ώστε να αυτό να είναι ένα διεθνώς αναγνωρισμένο board το οποίο θα μπορούσε να χρησιμοποιήσει όσο το δυνατότερο περισσότερους αισθητήρες με το μικρότερο επιτρεπτό μέγεθος ώστε να λειτουργούν τα modules, χαμηλού κόστους και με τις περισσότερες παροχές διασύνδεσης. Κάτι σαν το μισό σε μέγεθος λύσεων της Intel αλλά με συνδεσμολογία Wi-Fi.
Δεν επιλέχτηκε η λύση του Raspberry, γιατί παρόλο που ήταν καλύτερη σε παροχές, μας παρείχε περιττές συνδέσεις πχ. hdmi, 2 usb, ethernet κπλ, καταλήγοντας έτσι στην επιλογή του Arduino Uno Wi-Fi Rev 2. Το Arduino Uno WiFi Rev 2 διαθέτει επίσης υποστήριξη για over-the-air (OTA) σε επίπεδο προγραμματισμού για τη μεταφορά των Arduino sketches (κωδικά) και επιπλέον ασύρματη ασφάλεια που παρέχεται από ένα μικροτσίπ ECC608 CryptoAuthentication IC.

3.4.3 Εξοπλισμός για καταμέτρηση

Ο σημαντικότερος παράγοντας μιας εφαρμογής που στοχεύει στην εξοικονόμηση της ενέργειας είναι η συνεχής καταμέτρηση της ηλεκτρικής κατανάλωσης της μονάδας που ελέγχει η εφαρμογή, που στην περίπτωση μας είναι το κλιματιστικό. Μετά από έρευνα των διαθέσιμων τρόπων μέτρησης της κατανάλωσης ενέργειας και καταγραφής της, υιοθετήθηκε η χρήση του smart metering που γίνεται με την χρήση ενός καταγραφέα δεδομένων ενέργειας (energy data logger) απευθείας από τη μονάδα AC.

Παρουσιάζονται οι διάφοροι τρόποι που υπάρχουν για να εκτιμηθεί και να μετρηθεί η κατανάλωση του κλιματιστικού.

1. Αμπεροτσιμπίδα

   Μειονέκτημά της η προϋπόθεση να ξεχωρίσουν τα καλώδια τροφοδοσίας της συσκευής ώστε να μετρηθεί η κατανάλωση. Επίσης προσφέρει μόνο στιγμιαία μέτρηση, με ορισμένα πολύ ακριβότερα μοντέλα να καταγράφουν και κάποιο μικρό χρονικό διάστημα.

Εικόνα 3.21 – Αμπεροτσιμπίδα

2. Ένας αισθητήρας που αναπτύχθηκε για Arduino

   Η σειρά SCT013[8] είναι αισθητήρες μετασχηματιστές ρεύματος που μετρούν την ένταση του ρεύματος που διασχίζει έναν αγωγό. Έχει παρόμοια χρήση με μια αμπεροτσιμπίδα με ίδιο μειονέκτημα τη στιγμιαία μέτρηση.

Εικόνα 3.22 - Αισθητήρας τύπου αμπεροτσιμπίδας για εφαρμογές πάνω σε Arduino

3. Χρήση απλού Energy Meter

   Προσφέρει στοιχειώδη ανάλυση της κατανάλωσης ενέργειας, καταγράφει στοιχεία όπως μέτρηση τάσης (V), ρεύματος (A), ισχύος (W), ενέργειας (KWh), συντελεστή ισχύος (PF), συχνότητας (Hz). Το μειονέκτημά του είναι ότι δεν τα συνδέει με συγκεκριμένες ημερομηνίες, απλά οι καταγραφές είναι Day 1, Day 2, Day 3, κλπ. και δεν είναι εφικτή η εξαγωγή των δεδομένων για περεταίρω επεξεργασία.

Εικόνα 3.23 - Απλός Energy Meter

[8] https://www.poweruc.pl/blogs/news/non-invasive-sensor-yhdc-sct013-000-ct-used-with-arduino-sct-013

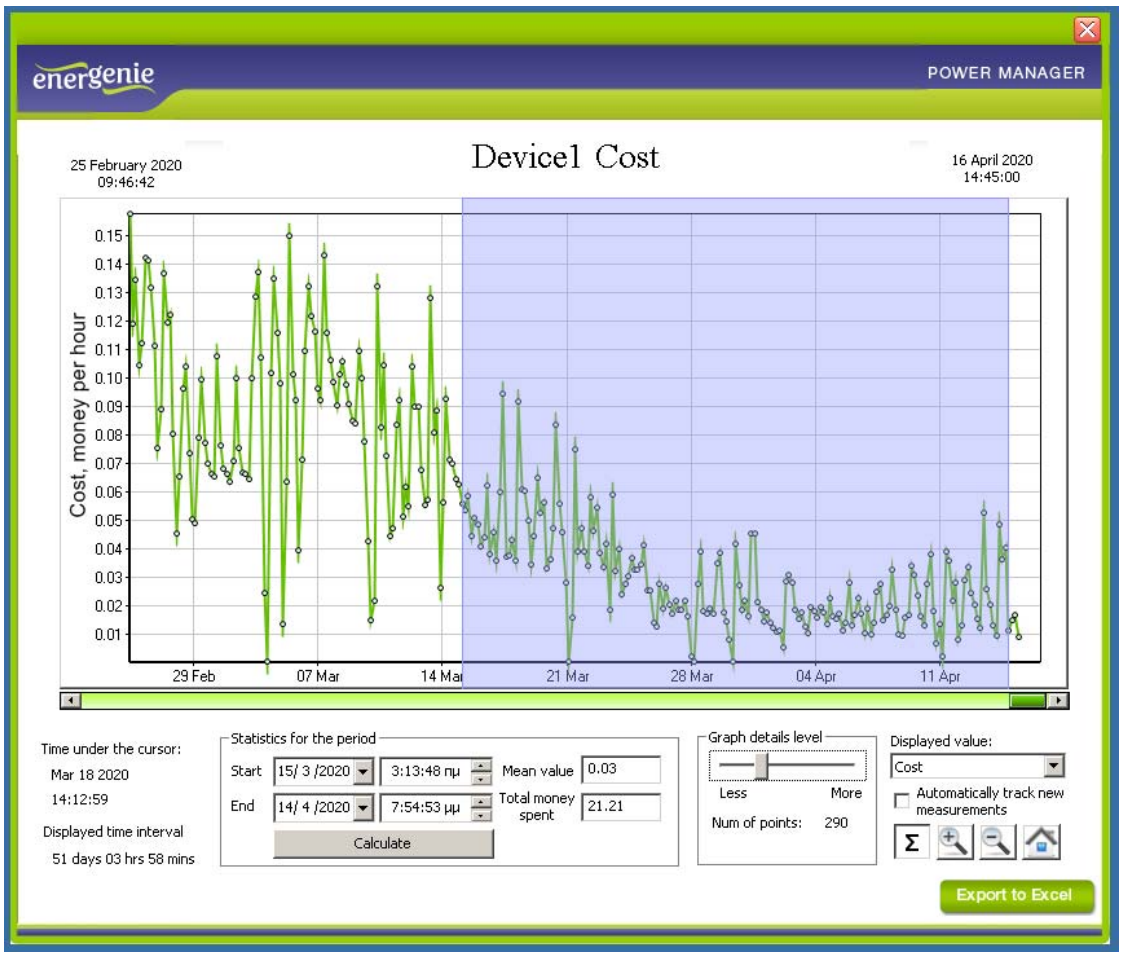
4. Χρήση Power Energy Meter ENERGENIE EGM-PWM[9]

Η πιο ακριβή λύση σε επίπεδο κόστους (το κόστος της ανέρχεται περίπου στα 40€) αλλά με τα χαρακτηριστικά που απαιτούνται για μια σωστή μελέτη καταγραφής ενέργειας. Προσφέρει συνδεσμολογία USB, Wi-Fi, LAN ανάλογα την έκδοση. Η επιλεγμένη έκδοση που αγοράστηκε διαθέτει μόνο δυνατότητα συνδεσμολογίας με USB καλώδιο και αυτό προϋποθέτει την συνεχή σύνδεση του με κάποιο υπολογιστή. Στην περίπτωση της έρευνας μας, χρησιμοποιήθηκε ένα μικρό φορητό laptop. Τα πλεονεκτήματα του είναι ότι προσφέρει στιγμιαία και συνεχείς καταγραφή και με ημερολογιακή πληροφόρηση. Το χαρακτηριστικό αυτό ήταν απαραίτητο στην καταμέτρηση γιατί προσέφερε πληρέστερα στοιχεία για την κατανάλωση του κλιματιστικού, ειδικά σε περιόδους επιλογής απενεργοποίησης της αναπτυσσόμενης συσκευής. Επίσης ήταν δυνατή η εξαγωγή συμπερασμάτων για το πως οι διαφορετικές εντολές είχαν αντίκτυπο στην κατανάλωση ενέργειας του κλιματιστικού.

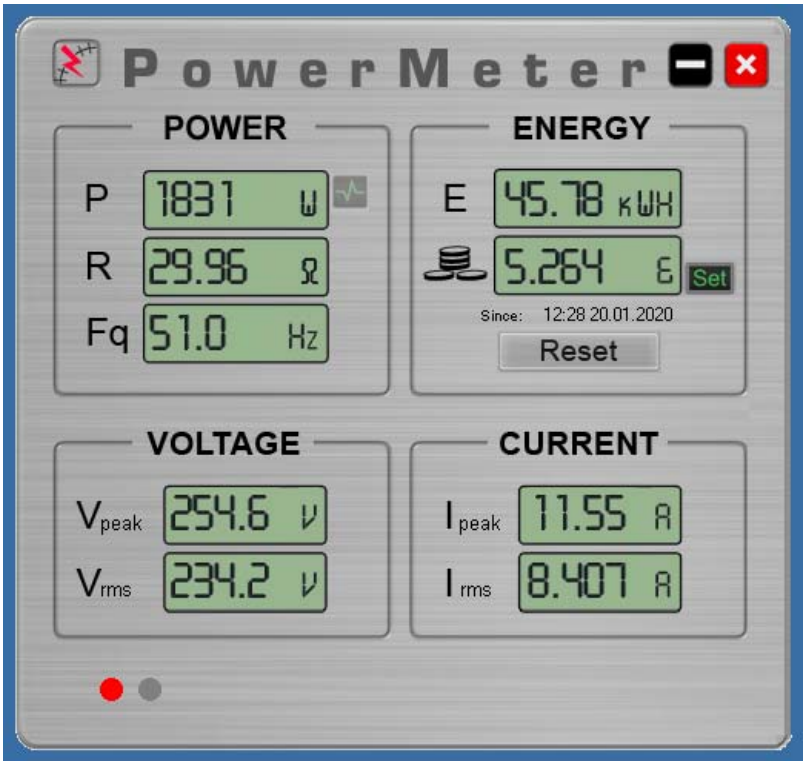
Εικόνα 3.24 - Επιλεγμένος Smart Power Meter

Οι τρέχουσες στιγμιαίες τιμές κατανάλωσης (εικόνα 3.26) παρουσιάζονται μέσα από το λογισμικό του smart power meter, καθώς και η κατανάλωση από 20-1-2020 με κόστος που υπολογίζεται με οριζόμενη τιμή κιλοβατώρας (kWH) τα 0,115 Ευρώ, που είναι περίπου η τιμή που προσφέρει η ΔΕΗ. Ενδεικτικά, με αυτοματοποιημένη την διαδικασία και σταθερή λειτουργία της συσκευής, το διάστημα από 15-3-2020 μέχρι 14-4-2020 η κατανάλωση του κλιματιστικού ήταν 21.21 ευρώ, όπως παρουσιάζεται στο γράφημα της εικόνας 2.25.

[9] https://energenie.com/item.aspx?id=6853



Εικόνα 3.25 - Κόστος κατανάλωσης από 15-3-2020 μέχρι 14-4-2020.



Εικόνα 3.26 -Στιγμιαίες καταμετρήσεις κατανάλωσης από το power meter

Επίσης με τη χρήση του Power Energy Meter ENERGENIE EGM-PWM καταγράφονται οι τιμές για τα εξής:

- Frequency Of Current σε Hz
- Active Impedance σε Ohm
- Root Mean Squared Current σε A (amperes)
- Root Mean Squared Voltage σε V (volts)
- Real Power σε Watt

Ο υπολογισμός της πραγματικής ενέργειας (Real Power (Watts)) που καταναλώθηκε με χρήση ή μη της αναπτυσσόμενης συσκευής γίνεται από τον παρακάτω τύπο:

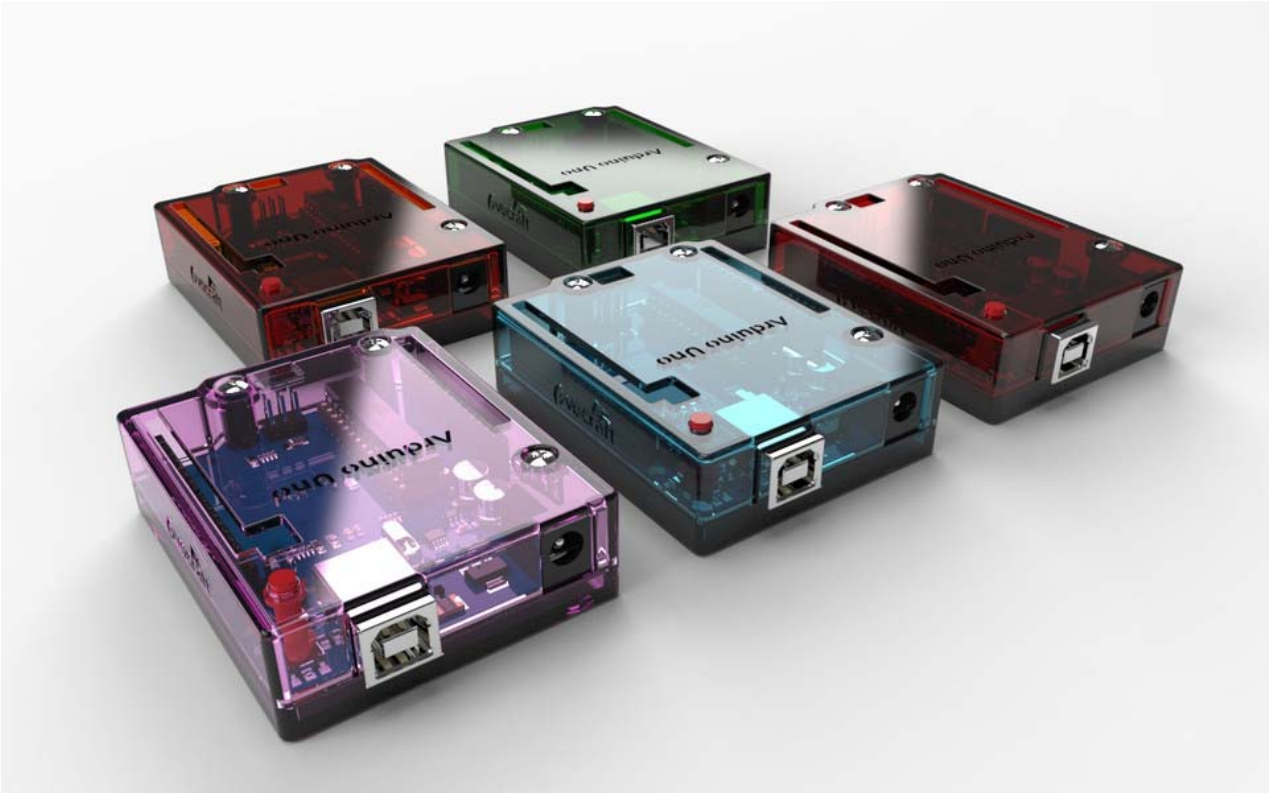
Amps (A) x Volts (V) x Power Factor = Watts (W)

και εξάγεται αυτόματα από το λογισμικό της συσκευής με τιμή του power factor[10] ~0.91.

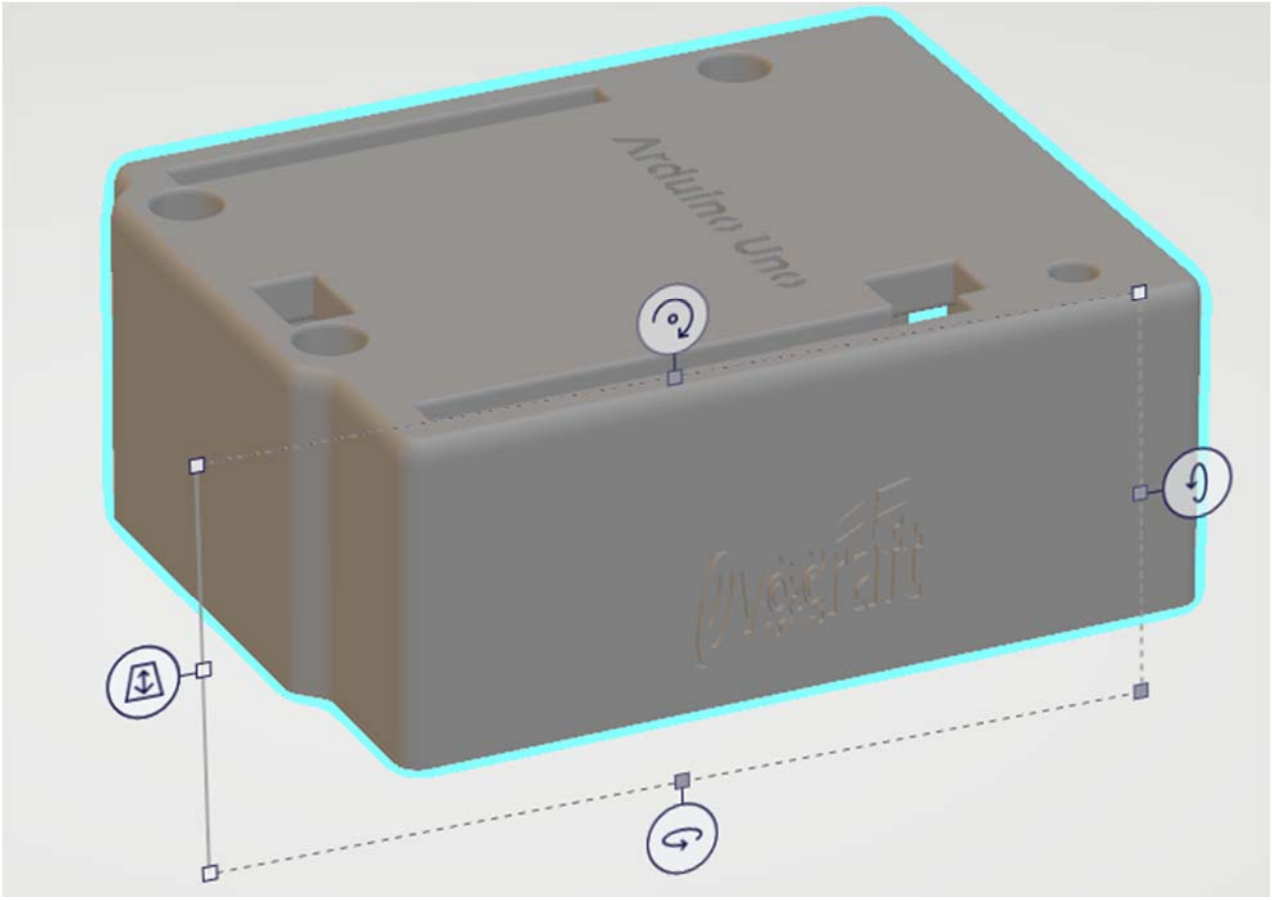
3.4.4 Σχεδίαση για packaging

Είχε επιλεχτεί για αισθητικούς λόγους, παρόλο που θα είχαμε αύξηση του τελικού κόστους της παραγόμενης συσκευής, να κατασκευαστεί ένα περίβλημα με χρήση 3D Printers. Δεν κατέστη όμως εφικτό να τυπωθεί το περίβλημα λόγω καθολικής απαγόρευσης λειτουργίας όλων των εμπορικών καταστημάτων εξαιτίας του COVID-19. Συμπεριλαμβάνονται τα σχέδια της εκτύπωσης (εικόνες 3.28 έως 3.34). Τα αρχικά σχέδια προήλθαν από την GRABCAD κοινότητα[11] που σχεδιάζει διάφορες λύσεις για 3D εκτυπωτές και παρέχονται ελεύθερα προς χρήση. Το επιλεγμένο σχέδιο μετά από τροποποίηση στο ύψος της βάσης (διπλασιασμός), μπορούσε να φιλοξενήσει την τελική συσκευή μας. Στα σχέδια υπάρχουν προσχεδιασμένα τα ανοίγματα που χρειαζόμασταν για την τροφοδοσία και την USB θύρα. Επιπλέον υπήρξε σχέδιο για υλοποίησή κουμπιού που θα χρησιμοποιούνταν για επανέναρξη (reset). Τα αρχεία είναι σε μορφή .STL που χρησιμοποιούνται με λογισμικά επεξεργασίας 3D σχεδίων.

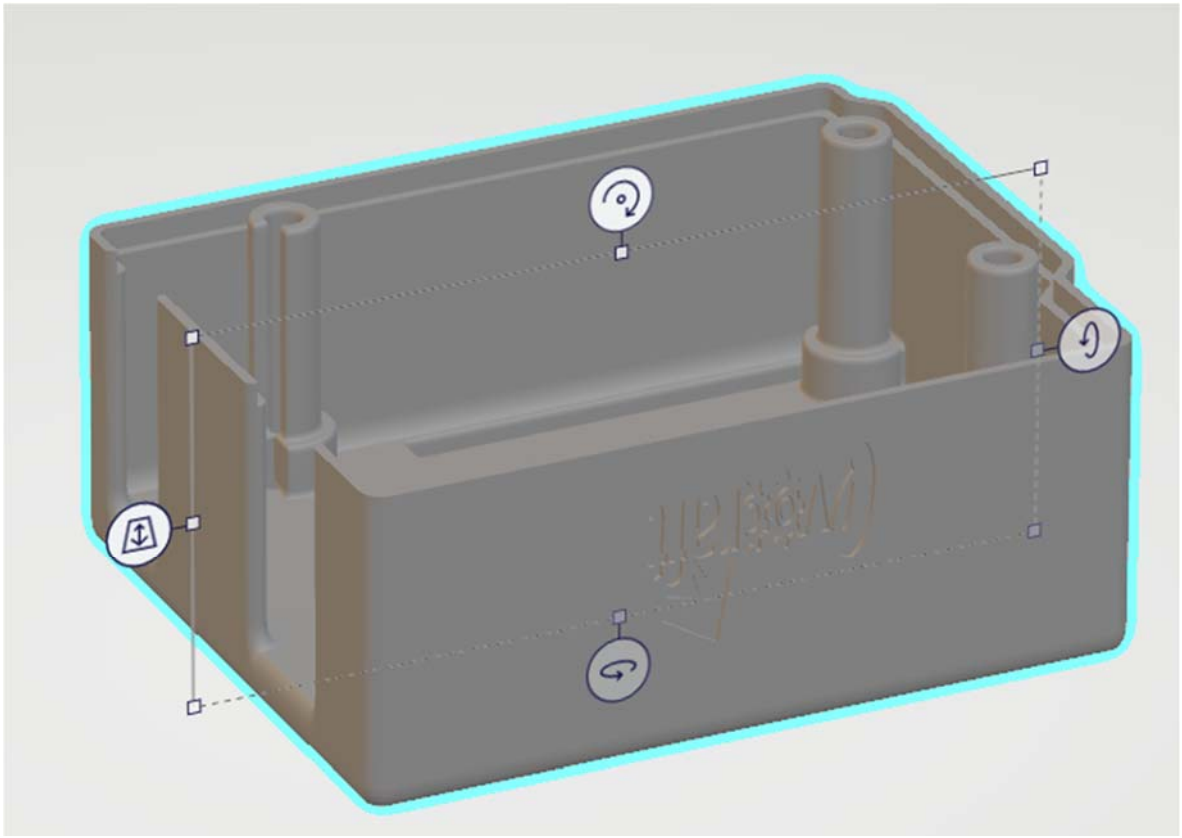
[10] https://www.electronics-tutorials.ws/accircuits/power-triangle.html
[11] https://grabcad.com/library/arduino-uno-case-3d-printable-1



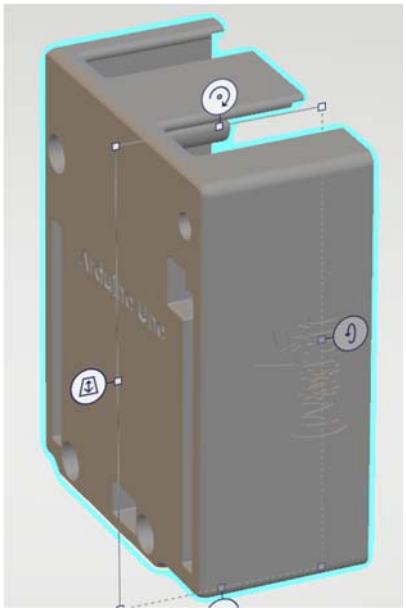
Εικόνα 3.27 - Αρχικά σχέδια 3D



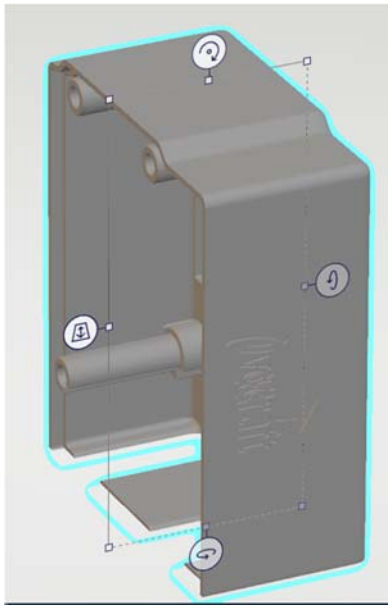
Εικόνα 3.28 - το κάτω μέρος της συσκευής – Α όψη – 3D σχέδιο



Εικόνα 3.29 - το κάτω μέρος της συσκευής – Β όψη– 3D σχέδιο

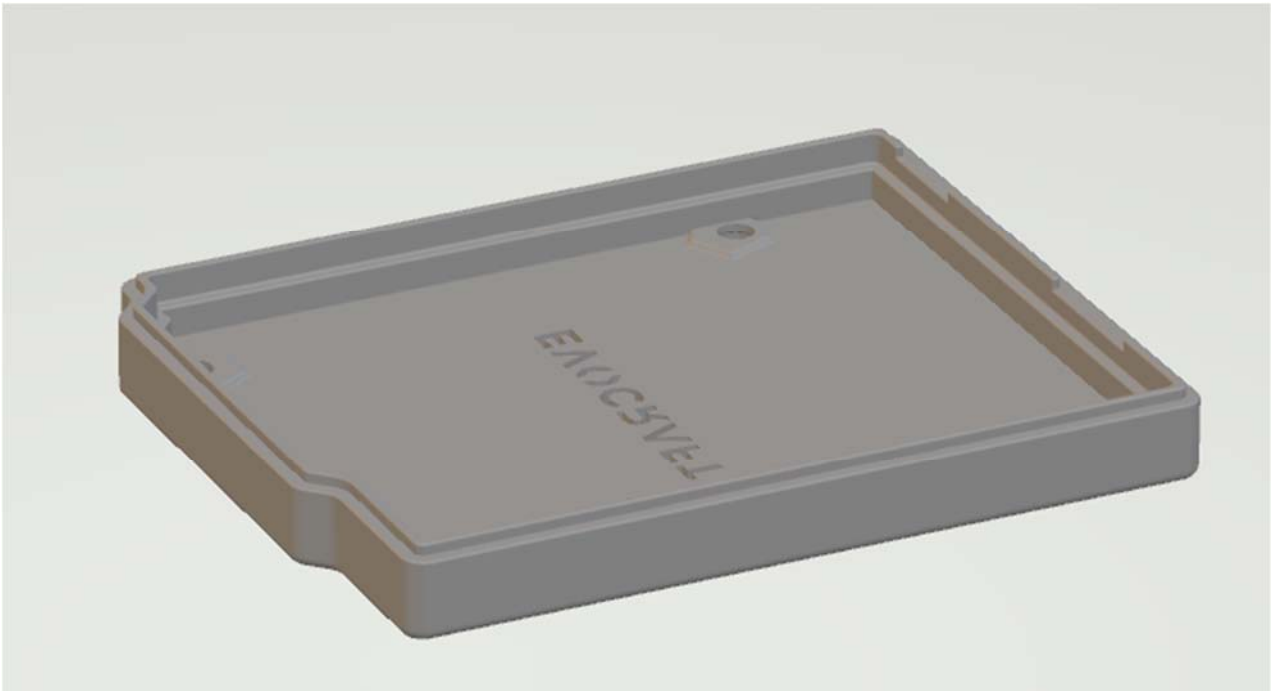


Εικόνα 3.30 - το κάτω μέρος της συσκευής – Γ όψη - 3D σχέδιο

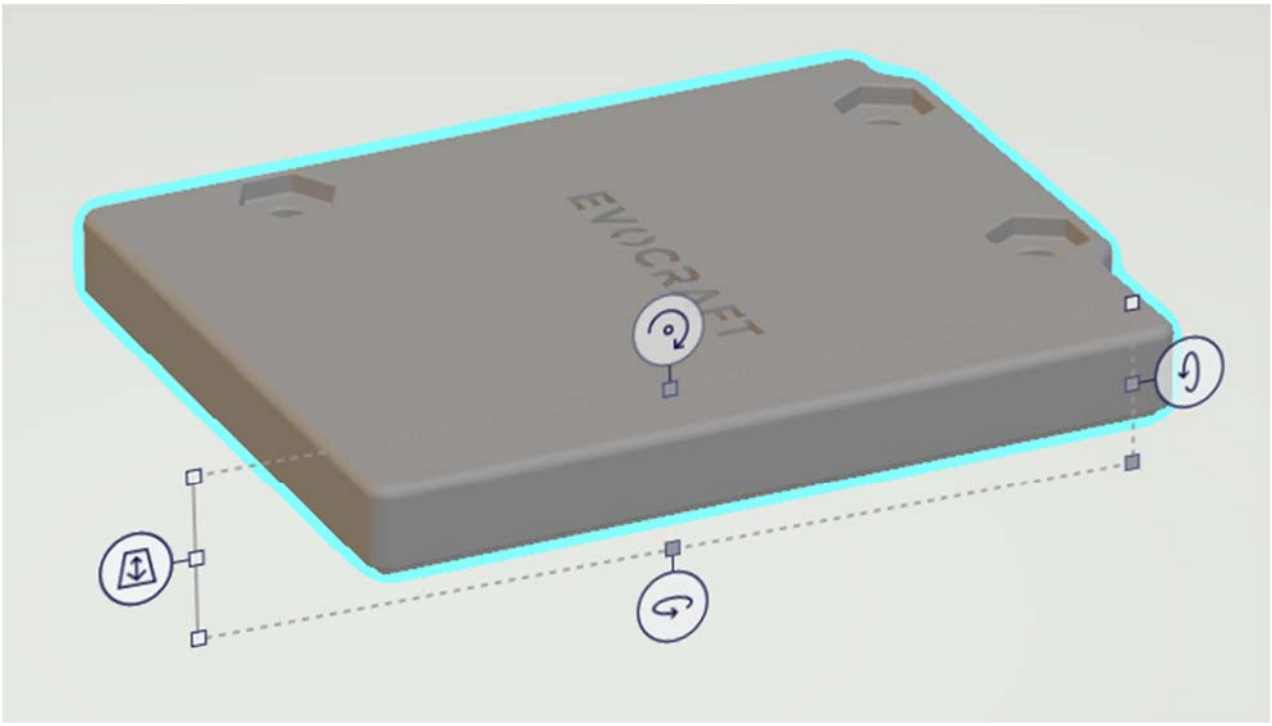


Εικόνα 3.31 - το κάτω μέρος της συσκευής – Δ όψη- 3D σχέδιο

Εικόνα 3.32 - για πάτημα του reset κουμπιού



Εικόνα 3.33 - το άνω μέρος της συσκευής – Α όψη- 3D σχέδιο



Εικόνα 3.34 - το άνω μέρος της συσκευής – Β όψη - 3D σχέδιο

3.5 Εργαλεία - Λογισμικά

3.5.1 Android Studio IDE

Για την ανάπτυξη της εφαρμογής που θα παρακολουθούσε το IoT σύστημα μέσα από το περιβάλλον κινητής συσκευής (smartphone), επιλέχτηκε το λογισμικό Android Studio IDE. Είναι ένα ολοκληρωμένο πρόγραμμα ανάπτυξης (IDE - integrated development environment) το οποίο αποτελείται από τουλάχιστον ένα πρόγραμμα επεξεργασίας πηγαίου κώδικα, εργαλεία αυτοματισμού κατασκευής και ένα πρόγραμμα εντοπισμού σφαλμάτων. Στοχεύει στην ανάπτυξη εφαρμογών για το λειτουργικό σύστημα Android της Google. Οι εφαρμογές που είναι αναπτυγμένες στο Android Studio μπορούν να εξαχθούν σε μορφή APK αρχείων και να υποβληθούν στο Google Play Store. Είναι αρκετά εύκολο περιβάλλον για προγραμματισμό γιατί υπάρχει ενσωματωμένο πρόγραμμα επεξεργασίας κώδικά που βοηθά τον προγραμματιστή στην γραφή του κώδικά του, στην ανάλυση, εντοπισμό των σφαλμάτων κ.α. Το λογισμικό ανακοινώθηκε στο συνέδριο της Google I/O τον Μάιο του 2013, ενώ η πρώτη σταθερή έκδοση κυκλοφόρησε τον Δεκέμβριο του 2014. Το Android Studio είναι διαθέσιμο για πλατφόρμες λειτουργικών συστημάτων Mac, Windows και Linux. Αντικατέστησε το Eclipse Android Development Tools (ADT) ως το πρωταρχικό IDE για ανάπτυξη εφαρμογών Android.

Τα κύρια χαρακτηριστικά[12] που προσφέρει είναι :

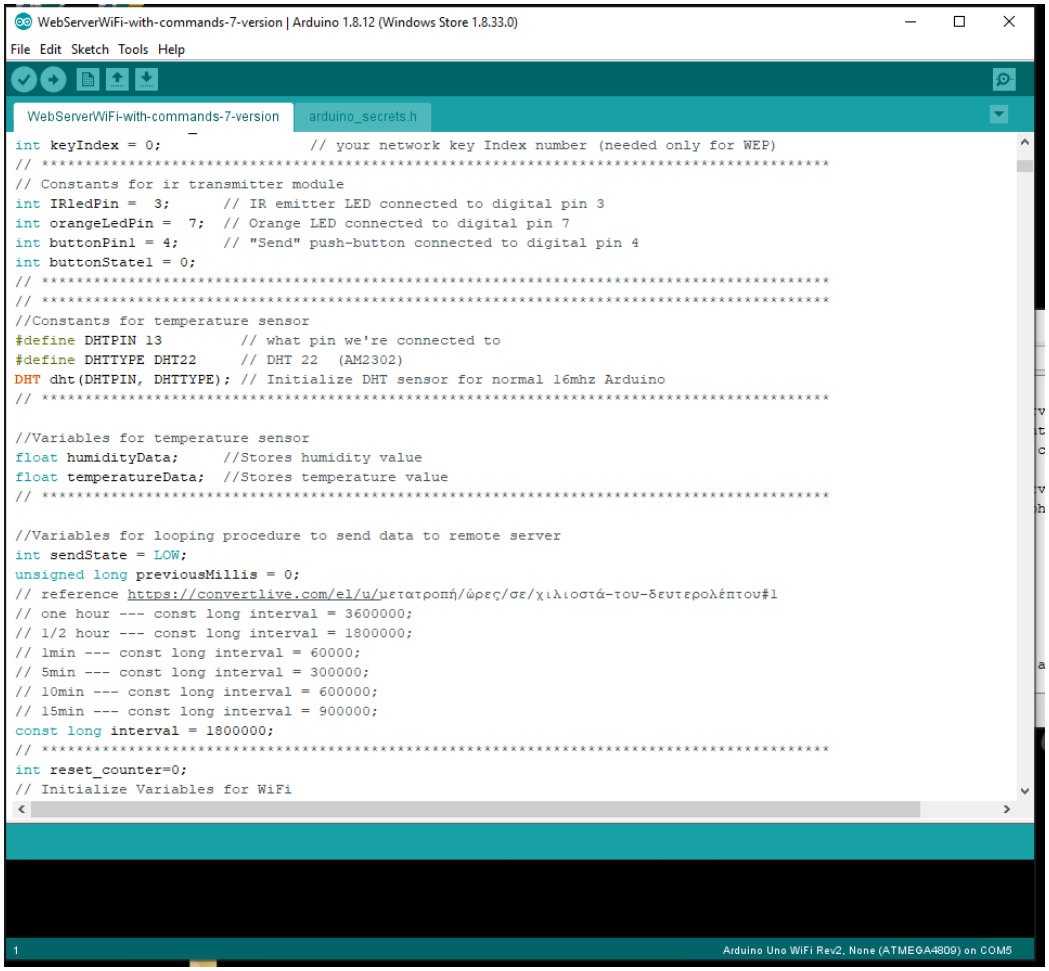
- Υποστήριξη οικοδόμησης με βάση το Gradle.
- Αντανακλαστικά ειδικά για Android και γρήγορες επιδιορθώσεις.
- Εργαλεία Lint για να επιτύχουν την απόδοση, τη χρηστικότητα, τη συμβατότητα έκδοσης και άλλα προβλήματα.
- ProGuard ενσωμάτωσης και δυνατότητες app-signing (δηλαδή η χρήση κάποιου λογαριασμού π.χ. sign in with Facebook account).
- Οδηγός πρότυπων για τη δημιουργία κοινών σχεδίων και εξαρτημάτων Android.
- Ένας πλούσιος επεξεργαστής διάταξης που επιτρέπει τη μεταφορά και απόθεση συστατικών UI, καθώς και την προεπισκόπηση σε πολλαπλές διαμορφωμένες οθόνες.
- Android Virtual Device (εξομοιωτή). Επιλέγοντας την έκδοση του λογισμικού Android καθώς και τα χαρακτηριστικά του κινητού και το μέγεθος της οθόνης βλέπουμε πως συμπεριφέρεται η εφαρμογή μας με συγκεκριμένα μοντέλα κινητού τηλεφώνου. Με αυτόν τον τρόπο μπορεί ο προγραμματιστής να στοχεύσει κατηγορίες τελικών χρηστών.

[12] https://en.wikipedia.org/wiki/Android_Studio

3.5.2 Arduino IDE

Το περιβάλλον Arduino IDE είναι ένα λογισμικό το οποίο υποστηρίζει τόσο τους ομώνυμους μικροελεγκτές (board) της εταιρίας αλλά και μικροελεγκτές (board) από άλλες εταιρίες, αρκεί να κατασκευάζονται με τις προτεινόμενες προδιαγραφές της ομώνυμης εταιρίας. Στην περίπτωση μας, η επιλογή έγινε από την ομώνυμη εταιρία board για καλύτερη συμβατότητα. Είναι ένα ολοκληρωμένο πρόγραμμα ανάπτυξης (IDE) το οποίο αποτελείται από τουλάχιστον ένα πρόγραμμα επεξεργασίας πηγαίου κώδικα, εργαλεία αυτοματισμού κατασκευής και ένα πρόγραμμα εντοπισμού σφαλμάτων.
Το περιβάλλον ανάπτυξης αποτελείται από:

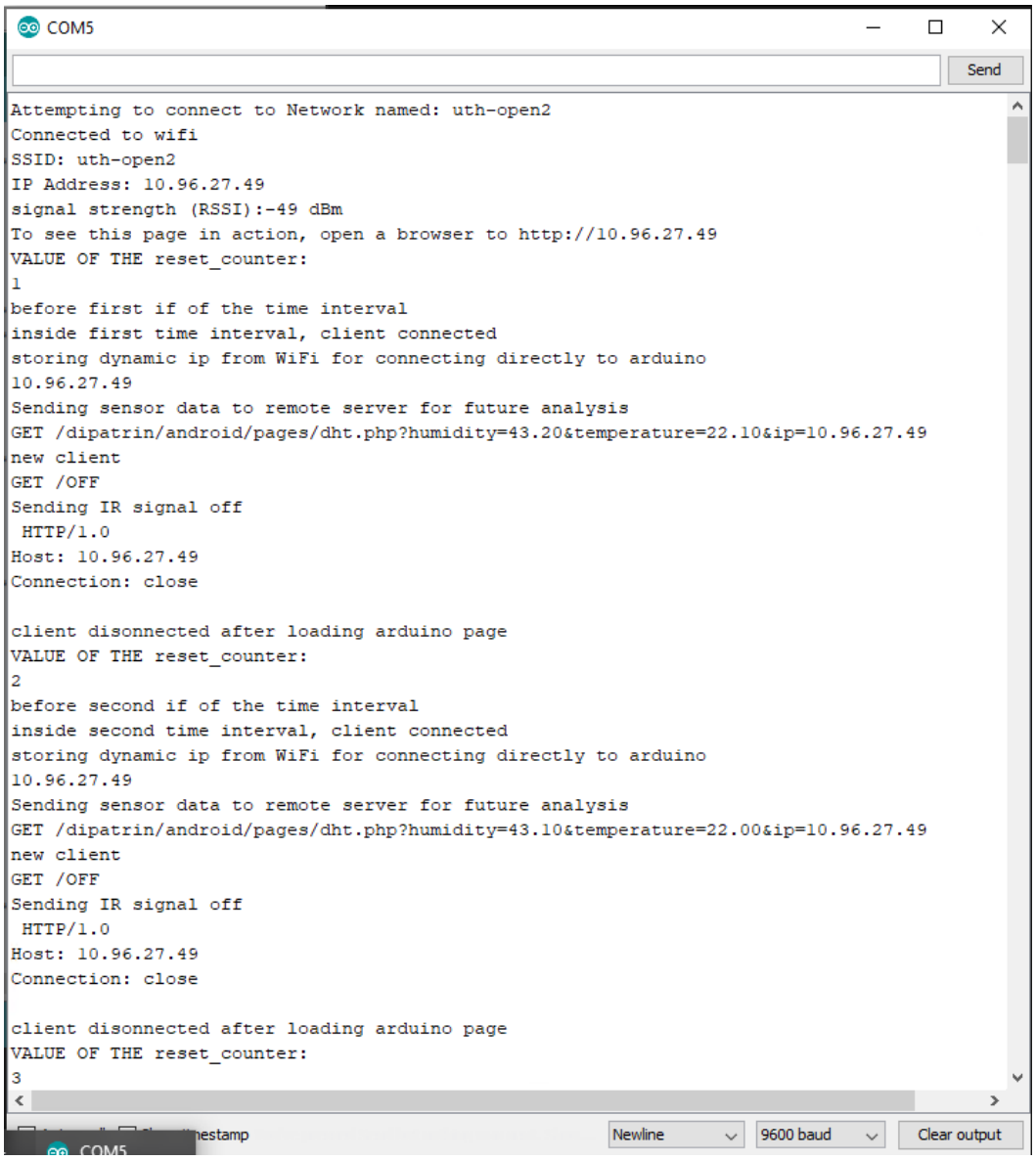
- Ένα πλαίσιο επεξεργασίας κειμένου για την συγγραφή του κώδικα
- Μια γραμμή εργαλείων που ελέγχει και στέλνει (upload) τον κώδικα στην συνδεδεμένη συσκευή
- Ένα μενού με επιλογές για διαχείριση των βιβλιοθηκών του λογισμικού και επιλογή του υλικού που υπάρχει συνδεδεμένο με το λογισμικό ανάπτυξης
- Ένα πλαίσιο μηνυμάτων, όπου εμφανίζονται τυχόν λάθη της μεταγλώττισης



Εικόνα 3.35 – Περιβάλλον του Arduino Studio

Το πρόγραμμα που δημιουργείται, ονομάζεται σκίτσο (sketch) και έχει κατάληξη .ino. Κάθε βιβλιοθήκη που εγκαθίσταται περιέχει έτοιμα πρότυπα κώδικα που μπορεί κάποιος να ξεκινήσει να αναπτύσσει πάνω σε αυτά. Οι βιβλιοθήκες είναι σημαντικές γιατί χωρίς την σωστή βιβλιοθήκη δεν μπορεί να λειτουργήσουν οι αισθητήρες που τοποθετούνται πάνω στο board.

Επιπλέον υπάρχει κονσόλα εξαγωγής αποτελεσμάτων του προγράμματος που χρησιμεύει στο να βλέπουμε αποτελέσματα από τους αισθητήρες και τα modules (εικόνα 3.36). Παράδειγμα όταν το module συνδεθεί στο ασύρματο δίκτυο μπορούμε να ζητήσουμε να πας γνωστοποιήσει την ip διεύθυνση που πήρε από το Access Point. Η κονσόλα αυτή χρησιμοποιείται για την παρατήρηση όλων των διαδικασιών ξεχωριστά όπως, μέτρηση θερμοκρασίας, επικοινωνίας με βάση δεδομένων, λήψη απομακρυσμένης εντολής είτε αυτοματοποιημένης είτε από χρήστη.



Εικόνα 3.36 – Βοηθητικό παράθυρο που λαμβάνουμε πληροφορίες από το Arduino studio

Κεφάλαιο 4

Μεθοδολογία και υλοποίηση

4.1 Υπάρχουσα κατάσταση

Τα τελευταία χρόνια με την ανάπτυξη όλο και περισσότερων έξυπνων συσκευών παρατηρείται ότι από τις πρώτες συσκευές που εντάχτηκαν σε αυτήν την κατηγορία ήταν οι κλιματιστικές μονάδες ψύξης / θέρμανσης. Αυτή η ένταξη όμως στην κατηγορία έξυπνων συσκευών δεν ολοκληρώθηκε πλήρως αφού τις περιορίζει προσφέροντας μόνο απομακρυσμένη διαχείριση από τον χρήστη και όχι κάποιο είδους αυτοματοποιημένη λειτουργία.

4.2 Σχεδιαστικοί παράγοντες

Με τη θέσπιση και την εφαρμογή ορισμένων προκαθορισμένων κανόνων, ολόκληρο το σύστημα IoT μπορεί να προσαρμοστεί με αυτόματες ρυθμίσεις που να μην είναι υποχρεωτική η ανθρώπινη

παρέμβαση. Ο αλγόριθμος προγραμματισμού που ακολουθείται είναι ένας RCPSP Resource-Constrained Project Scheduling Problem στον οποίο προγραμματίζονται μια σειρά από δραστηριότητες. Κάθε δραστηριότητα έχει μια προκαθορισμένη διάρκεια και δεν μπορεί να διακοπεί. Υπάρχουν μια σειρά προτεραιότητας με σχέσεις μεταξύ των ζευγών των δραστηριοτήτων που δηλώνουν πότε η δεύτερη δραστηριότητα πρέπει να ξεκινήσει μετά το πέρας της πρώτης. Ο αλγόριθμος στοχεύει στην ελαχιστοποίηση της συνολικής διάρκειας του έργου.

Απαιτείται ιδιαίτερη προσοχή κατά τον σχεδιασμό και την ανάπτυξη της εφαρμογής IoT σε ότι αφορά το περιβάλλον του χρήστη και τις δυνατότητες που θα του προσφέρονται. Παρόλο που ο χρήστης μπορεί να είναι σε θέση να εκτελέσει μια ενέργεια αυτή μπορεί να αποτύχει ή να είναι λανθασμένη επιλογή και να επηρεάσει όλο το σύστημα προξενώντας κάποια σοβαρή ζημιά σε εξοπλισμό. Συνεπώς θα πρέπει να υλοποιηθούν δικλίδες ασφαλείας ώστε να διορθώνονται τυχόν εσφαλμένες επιλογές του χρήστη.
Τα βασικά στάδια που σχεδιάστηκαν και ακολουθήθηκαν στο αναπτυσσόμενο σύστημα IoT είναι τα εξής:

Α. Προεργασία

Πριν τα βασικά στάδια εκτέλεσης του συστήματος, πραγματοποιήθηκε μια προεργασία που αφορούσε:

1. Αντιγραφή από το τηλεχειριστήριο και ομαδοποίηση όλων των IR εντολών που μας ενδιαφέρουν να αναπαράγει η συσκευή μας
2. Εγκατάσταση της αναπτυσσόμενης συσκευής στο χώρο του κλιματιστικού στην σωστή θέση μακριά από την ροή του αέρα. Η συσκευή μπορεί να λειτουργεί είτε με την βοήθεια ενός power bank είτε με σύνδεση σε κάποια παροχή ρεύματος

Β. Βασικά στάδια εκτέλεσης της αναπτυσσόμενης συσκευής

1. Κάθε 30 λεπτά το σύστημα διαβάζει και στέλνει στον web server την μέτρηση της θερμοκρασίας και υγρασίας στο χώρο.
2. Ο web server, αποθηκεύει τα δεδομένα θερμοκρασίας και εκτελεί σύγκριση με τον στόχο του δωματίου (που αρχικά τέθηκε στους 22^{o}C). Ανάλογα με το αποτέλεσμα της σύγκρισης στέλνει πίσω στην συσκευή την ανάλογη εντολή. Αποθηκεύει την αποστολή στην βάση δεδομένων ώστε να υπάρχει ιστορικό κινήσεων/εντολών.
3. Στην συσκευή με τον αισθητήρα, εκτελείται μια ακόμα web server υπηρεσία η οποία είναι σε συνεχή αναμονή για λήψη εντολής που του στέλνει ο διακομιστής που εκτέλεσε τις συγκρίσεις

των δυο θερμοκρασιών (δωματίου και στόχου). Όταν λάβει κάποια εντολή την εκτελεί, δείχνοντας το και πάνω στην συσκευή (εναλλαγή φωτεινής σήμανσης όταν αποστέλλετε σήμα IR) για οπτική επιβεβαίωση.

4. Επαναλαμβάνεται η ίδια διαδικασία πάλι από το βήμα 1. Μετά την τέταρτη επανάληψη το σύστημα με το Arduino πραγματοποιεί αυτόματη επανεκκίνηση ώστε να διατηρείται πάντα στην βέλτιστη λειτουργική κατάσταση διαχείρισης μνήμης.

Η εφαρμογή Android επικοινωνεί για την αποστολή εντολών ή την προβολή στατιστικών δεδομένων με τον ξεχωριστό διακομιστή και όχι απευθείας με την IoT συσκευή για λογούς μεγαλύτερης ασφάλειας και για να αποθηκεύονται τα ιστορικά των εντολών.

4.3 Επιλογή κατάλληλου threshold θερμοκρασίας

Σύμφωνα με το Αμερικάνικο τμήμα ενέργειας U.S. Department of Energy's (DOE) [13] μπορεί να επιτευχθεί οικονομική εξοικονόμηση ρυθμίζοντας σωστά τους θερμοστάτες των κτιρίων και υιοθετώντας αυτοματοποιημένους θερμοστάτες. Ένα κοινό λάθος που γίνεται είναι ότι κλείνουμε την μονάδα θέρμανσης ή ψύξης θεωρώντας ότι κάνουμε εξοικονόμηση ενέργειας ενώ στην πραγματικότητα όταν επανεργοποιείται μια μονάδα αυτή δουλεύει πολύ περισσότερο από το κανονικό για να επιτύχει τον στόχο της. Αποτέλεσμα αυτού του λάθους είναι η καταπόνηση των μονάδων καθώς λειτουργούν συνέχεια σε υψηλότερους ρυθμούς κατανάλωσης και άρα την μη εξοικονόμηση ενέργειας και χρημάτων.
Παράδειγμα σεναρίου κλιματισμού θέρμανσης σε γραφείο εργασίας κατά την περίοδο του χειμώνα. Κλείνοντας τον κλιματισμό και επιστέφοντας την επόμενη ημέρα στην εργασία μας θα χρειάζεται σχεδόν όλη την ημέρα να δουλεύει η μονάδα για να προσπαθεί να επιτύχει έναν στόχο στους 22-23 βαθμούς από τους 10-15 που θα βρούμε τον εργασιακό χώρο το πρωί. Αν είχε επιλεχτεί να διατηρείται η θερμοκρασία με κάποιον αυτοματοποιημένο θερμοστάτη στους 19-20 βαθμούς η λειτουργία της μονάδας κλιματισμού θα περιορίζονταν σε λιγότερες ώρες και θα λειτουργούσε πιο σωστά χωρίς να καταπονείται η μονάδα με την συνεχόμενη λειτουργία της.

Όσο μικρότερη είναι η διαφορά μεταξύ των εσωτερικών και εξωτερικών θερμοκρασιών, τόσο χαμηλότερη είναι και η κατανάλωση. Οι ιδανικές ρυθμίσεις είναι να κρατάμε την εσωτερική

[13] https://www.energy.gov/energysaver/thermostats

θερμοκρασία με λίγους βαθμούς διαφορά από την εξωτερική. Αυτό μας προσφέρει το πλεονέκτημα ότι ο εσωτερικός χώρος θα χάσει την ενέργεια στο περιβάλλον με πιο αργούς ρυθμούς.

Με την χρήση ενός αυτοματοποιημένου θερμοστάτη έχοντας την θερμοκρασία ρυθμισμένη στην «ιδανική θερμοκρασία» μπορούμε χωρίς να καταναλώσουμε πολύ ενέργεια και κατ' επέκταση χρήματα να μεταβούμε σε μια πχ. πιο ζεστή θερμοκρασία όταν επιστρέφουμε στο γραφείο ή στο σπίτι. Καθώς και σε μια πιο χαμηλή θερμοκρασία πχ. σε κάποιο server room.

Η ιδανική θερμοκρασία όπως ορίζετε στο American Heritage Dictionary [14] προσδιορίζει την θερμοκρασία δωματίου περίπου 20-22 ° C, ενώ το Oxford English Dictionary περίπου 20° C.

Το Αμερικάνικο τμήμα ενέργειας U.S. Department of Energy's αναφέρει ότι κάνοντας σωστή χρήση του θερμοστάτη μπορεί να επιφέρει 5-15% χαμηλότεροι οικονομικοί λογαριασμοί ενέργειας.

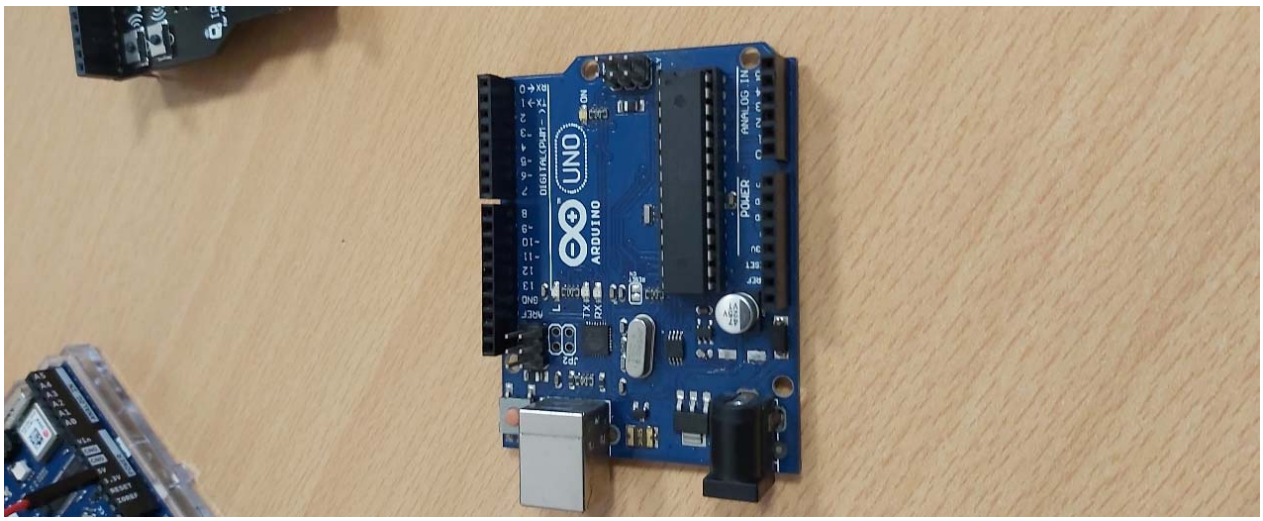
4.4 Συνδεσμολογία και επίλυση

Αρχικά σε πρώτη φάση υλοποίησης χρησιμοποιήθηκε εξοπλισμός αποτελούμενος από ένα Arduino UNO και ένα Ethernet shield για την δυνατότητα δικτύωσης. Στην συνέχεια μετά από την ολοκλήρωση των βασικών εντολών και δικτυακής επικοινωνίας επιλέχτηκε το καταλληλότερο board (αναλύθηκε στο τμήμα 3.4.2 Σύγκριση διαθέσιμων προτάσεων hardware) που θα μπορούσε να διαχειριστεί το τελικό μέγεθος του κώδικα, με στόχο την ελαχιστοποίηση του φυσικού μεγέθους και την οικονομικότερη επιλογή υλικού.

4.4.1 Κλωνοποίηση αρχικού IR σήματος

Το αρχικό βήμα της εργασίας ήταν να αποκωδικοποιηθεί το σήμα από το τηλεχειριστήριο του κλιματιστικού, ώστε στην συνέχεια να μπορεί να χρησιμοποιηθεί στην περεταίρω υλοποίηση. Χρησιμοποιήθηκε για αρχικός εξοπλισμός ένα Arduino Uno και αισθητήρες IR Receiver VS1838.

[14] https://ahdictionary.com/word/search.html?q=room+temperature



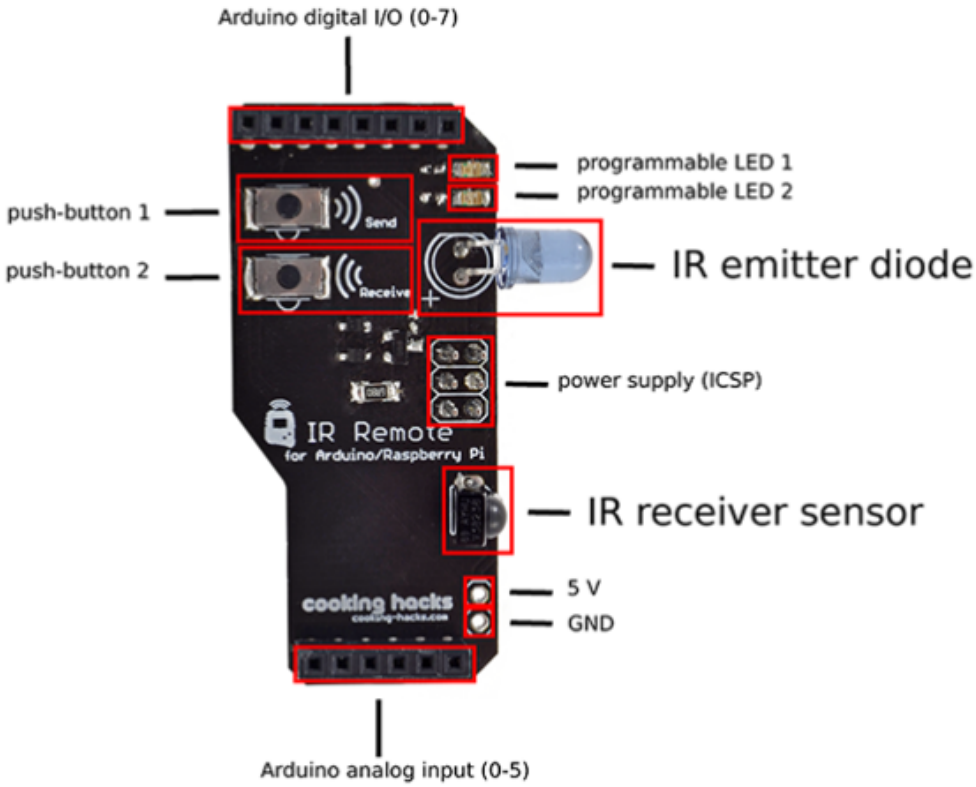
Εικόνα 4.1 - Arduino Uno board

Εικόνα 4.2 - IR Receiver VS1838

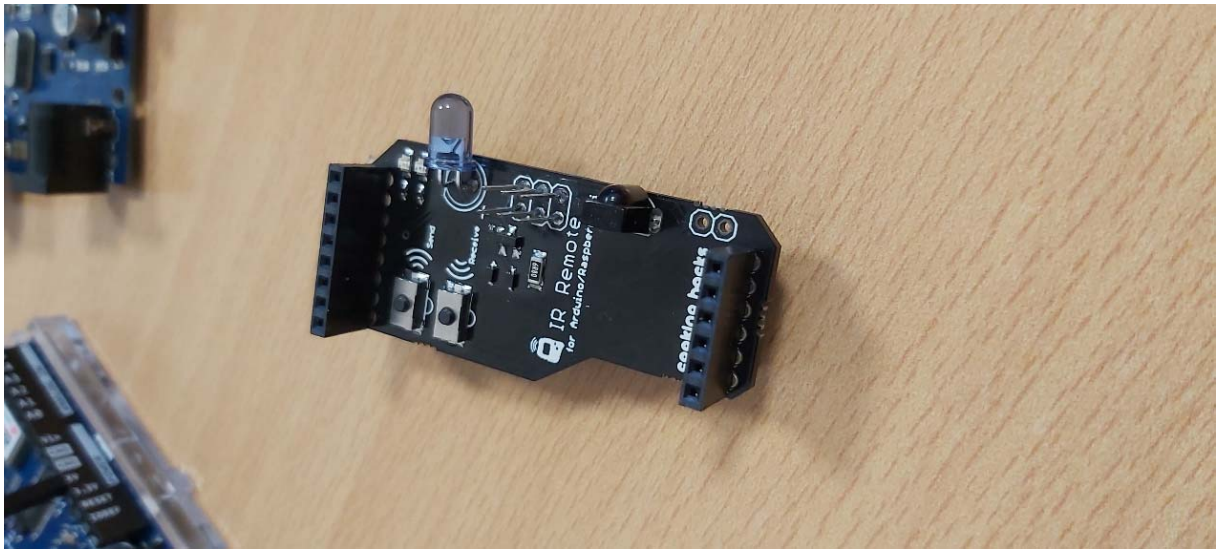
Διαπιστώθηκε ότι η εκπομπή του σήματος του συγκεκριμένου τηλεχειριστήριου προς την συγκεκριμένη μονάδα κλιματισμού ανήκει σε κατηγορία IR σήματος η οποία είναι κρυπτογραφημένη και δεν μπορούσε ένας απλός IR Receiver να το καταγράψει. Για να αντιμετωπιστεί και να επιλυθεί το πρόβλημα του κρυπτογραφημένου σήματος, εντοπίστηκε και αγοράστηκε ένα Module το οποίο είναι «υβριδικό» δηλαδή χρησιμοποιείται και από Arduino compatible συσκευές και από Raspberry συσκευές, έχοντας σαν βασική λειτουργία το «πλεονέκτημα» στην περίπτωσή μας, ότι μπορεί να αντιγράψει ένα IR σήμα χωρίς να το αποκωδικοποιήσει, ακόμα και αν είναι κρυπτογραφημένο. Το module αυτό αναπτύχθηκε από την εταιρία Cooking Hacks[15] η οποία τον Δεκέμβριο του 2019 απορροφήθηκε από την εταιρία Libelium μετά από 9 χρόνια στην ανάπτυξη IoT συσκευών. Το συγκεκριμένο IR Remote Module μας επιτρέπει να καταγράψουμε οποιοδήποτε υπέρυθρο σήμα στην μορφή που αποστέλλεται από ένα τηλεχειριστήριο με υπέρυθρες και να το στείλουμε ξανά μέσω ενός

[15] https://www.cooking-hacks.com

IR πομπού. Το συγκεκριμένο Module έχει ενσωματωμένο και τον πομπό και τον δεκτή των υπέρυθρων (εικόνα 4.3). Επίσης διαθέτει φωτεινές ενδείξεις οι οποίες χρησιμοποιούνται στον κώδικα κάθε φορά που στέλνουμε ένα σήμα ώστε να υπάρχει και οπτική επιβεβαίωση πάνω στην συσκευή.



Εικόνα 4.3 - Τμήματα του IR module από την εταιρία Cooking Hacks



Εικόνα 4.4 - Το module από την εταιρία Cooking Hacks

Συνδέοντας το μικροελεγκτή Arduino Uno μαζί με το module και χρησιμοποιώντας το σχετικό κώδικα που προσφέρει η κατασκευάστρια εταιρία του module (Παράρτημα Α.1 Κώδικας για εξαγωγή IR σήματος) στο περιβάλλον προγραμματισμού που προσφέρει το ίδιο το Arduino, καταγράφηκαν στην έξοδο του serial monitor οι λαμβανόμενες ξεχωριστές IR εντολές που έστελνε το τηλεχειριστηρίου ανά πλήκτρο.

Η καταγραφή των ξεχωριστών εντολών έγινε σε σκοτεινό δωμάτιο ώστε να ελαχιστοποιήσουμε τυχόν σφάλματα αντιγραφής του σήματος IR από εξωτερικές παρεμβολές φωτός. Για το λόγο ότι η εφαρμογή του αναπτυσσόμενου συστήματος δεν θα χρησιμοποιούνταν σε λειτουργία ζεστού αέρα καταγράφηκαν μόνο οι εντολές του τηλεχειριστηρίου σε λειτουργία ψύξης μαζί με τα κουμπιά λειτουργίας (on/off), όπως φαίνεται στον παρακάτω πίνακα. Οι εντολές για θερμοκρασία από 18-23^{o}C περιέχουν εκτός των άλλων ρυθμίσεων και την ενεργοποίηση της μονάδας. Από θερμοκρασίες 24 ^{o}C και πάνω λόγο περιορισμού του τηλεχειριστηρίου δεν συμπεριλαμβάνονταν η ενεργοποίηση της μονάδας. Η ενεργοποίηση της μονάδας είναι πολύ χρήσιμη σε ότι αφορά τον απομακρυσμένο έλεγχο ενός κλιματιστικού γιατί δεν μπορείς να γνωρίζεις την φυσική κατάσταση λειτουργίας της μονάδας.
Για παράδειγμα αν είχε κλείσει η μονάδα λόγω διακοπής ρεύματος και στέλναμε την εντολή α/α 9 δεν θα ανταποκρίνονταν η μονάδα γιατί δε θα εμπεριέχονταν η εντολή ενεργοποίησης της μονάδας. Θα έπρεπε να προηγείται της εντολής 9 η εντολή 1 (on AC unit).

Πίνακας Εντολών Τηλεχειριστηρίου

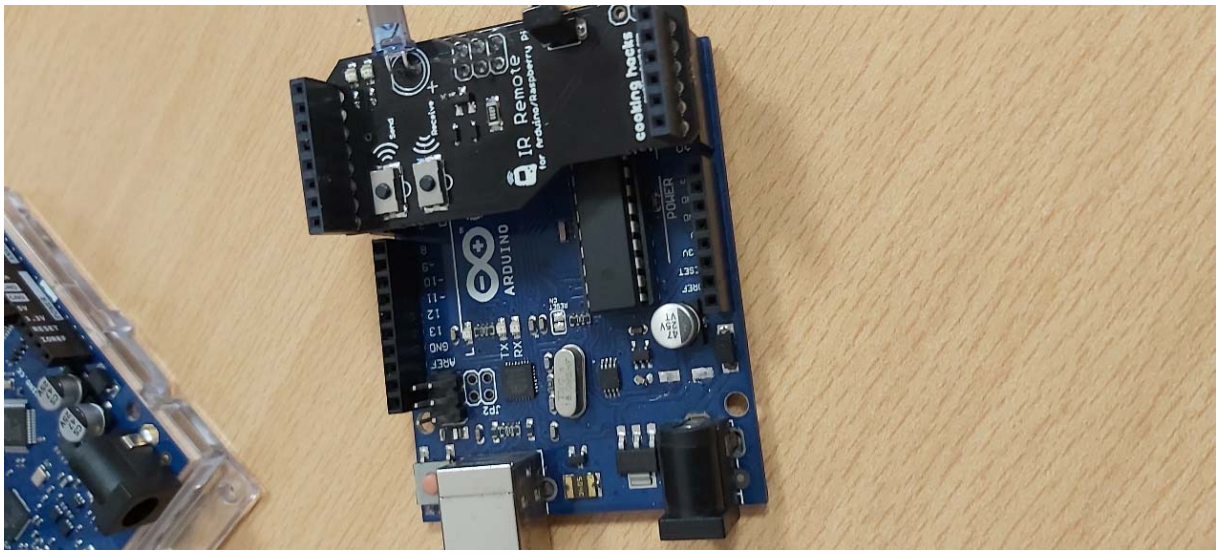
α/α	Όνομα εντολής	Λεπτομερή περιεχόμενα εντολής
1	On AC unit	Ενεργοποίηση μονάδας με θερμοκρασία 22 βαθμούς, αυτόματη ένταση αέρα, λειτουργία ψύξης
2	Off AC unit	Απενεργοποίηση μονάδας
3	18 θερμοκρασία	Ενεργοποίηση μονάδας με θερμοκρασία 18 βαθμούς, αυτόματη ένταση αέρα, λειτουργία ψύξης
4	19 θερμοκρασία	Ενεργοποίηση μονάδας με θερμοκρασία 19 βαθμούς, αυτόματη ένταση αέρα, λειτουργία ψύξης
5	20 θερμοκρασία	Ενεργοποίηση μονάδας με θερμοκρασία 20 βαθμούς, αυτόματη ένταση αέρα, λειτουργία ψύξης
6	21 θερμοκρασία	Ενεργοποίηση μονάδας με θερμοκρασία 21 βαθμούς, αυτόματη ένταση αέρα, λειτουργία ψύξης
7	22 θερμοκρασία	Ενεργοποίηση μονάδας με θερμοκρασία 22 βαθμούς, αυτόματη ένταση αέρα, λειτουργία ψύξης

8	23 θερμοκρασία	Ενεργοποίηση μονάδας με θερμοκρασία 23 βαθμούς, αυτόματη ένταση αέρα, λειτουργία ψύξης
9	24 θερμοκρασία	Αλλαγή θερμοκρασίας 24 βαθμούς, αυτόματη ένταση αέρα, λειτουργία ψύξης, λόγω περιορισμού τηλεχειριστηρίου συμπεριλαμβάνεται η ενεργοποίηση της μονάδας με θερμοκρασίες 24 και πάνω
10	25 θερμοκρασία	Αλλαγή θερμοκρασίας 25 βαθμούς, αυτόματη ένταση αέρα, λειτουργία ψύξης, λόγω περιορισμού τηλεχειριστηρίου συμπεριλαμβάνεται η ενεργοποίηση της μονάδας με θερμοκρασίες 24 και πάνω
11	26 θερμοκρασία	Αλλαγή θερμοκρασίας 26 βαθμούς, αυτόματη ένταση αέρα, λειτουργία ψύξης, λόγω περιορισμού τηλεχειριστηρίου συμπεριλαμβάνεται η ενεργοποίηση της μονάδας με θερμοκρασίες 24 και πάνω
12	27 θερμοκρασία	Αλλαγή θερμοκρασίας 27 βαθμούς, αυτόματη ένταση αέρα, λειτουργία ψύξης, λόγω περιορισμού τηλεχειριστηρίου συμπεριλαμβάνεται η ενεργοποίηση της μονάδας με θερμοκρασίες 24 και πάνω
13	28 θερμοκρασία	Αλλαγή θερμοκρασίας 28 βαθμούς, αυτόματη ένταση αέρα, λειτουργία ψύξης, λόγω περιορισμού τηλεχειριστηρίου συμπεριλαμβάνεται η ενεργοποίηση της μονάδας με θερμοκρασίες 24 και πάνω
14	29 θερμοκρασία	Αλλαγή θερμοκρασίας 29 βαθμούς, αυτόματη ένταση αέρα, λειτουργία ψύξης, λόγω περιορισμού τηλεχειριστηρίου συμπεριλαμβάνεται η ενεργοποίηση της μονάδας με θερμοκρασίες 24 και πάνω
15	30 θερμοκρασία	Αλλαγή θερμοκρασίας 30 βαθμούς, αυτόματη ένταση αέρα, λειτουργία ψύξης, λόγω περιορισμού τηλεχειριστηρίου συμπεριλαμβάνεται η ενεργοποίηση της μονάδας με θερμοκρασίες 24 και πάνω

Παρακάτω παρουσιάζεται ένα δείγμα της εντολής που στάλθηκε από το τηλεχειριστήριο και διαβάστηκε από τον δέκτη, που στην ουσία είναι χρονικές περίοδοι λειτουργίας του led που στέλνει το υπέρυθρο φως καθώς και οι χρονικές παύσεις μέχρι την επόμενη δέσμη υπέρυθρου φωτός.

Received:
pulseIR(8980);
delayMicroseconds(4160);
pulseIR(600);
delayMicroseconds(1500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
……

Έπειτα από σχετική έρευνα και δοκιμές διαπιστώθηκε ότι αρκετά air condition για να λειτουργούν απαιτούν τα IR σήματα που λαμβάνουν οι τελικές μονάδες να τα λαμβάνουν περισσότερες από μια φορές. Η περίπτωση που μελετάτε απαιτεί την λήψη της εντολής (IR σήματος) δυο (2) φορές και για αυτό ο κώδικας που αρχικά διαβάσαμε από το τηλεχειριστήριο για κάθε σήμα αποστέλλετε δύο φορές.



Εικόνα 4.5 - Η πλακέτα του uno μαζί με το IR module

4.4.2 Διασύνδεση συσκευής με δίκτυο

Στην συνέχεια αφού καταγράφηκαν οι εντολές των πλήκτρων του τηλεχειριστηρίου που επιλέχτηκαν υλοποιήθηκε μια τοπική υπηρεσία web server μαζί με ένα ethernet module (εικόνα 4.6), για να αποκτήσει δυνατότητα δικτύωσης το Arduino Board και να πραγματοποιηθούν δοκιμές (εικόνα 4.7) απομακρυσμένης αποστολής καταγεγραμμένων εντολών, ώστε να διαπιστωθεί αν λειτουργούν σωστά.

Εικόνα 4.6 – Το ethernet module για δοκιμές δικτύωσης και αποστολής καταγεγραμμένων εντολών IR

Control Links
Click RESET for Soft Reset
Click ON to power on AC
Click OFF to power off AC
Click 18 for temperature 18
Click 19 for temperature 19
Click 20 for temperature 20
Click 21 for temperature 21
Click 22 for temperature 22
Click 23 for temperature 23
Click 24 for temperature 24
Click 25 for temperature 25
Click 26 for temperature 26
Click 27 for temperature 27
Click 28 for temperature 28
Click 29 for temperature 29
Click 30 for temperature 30

Εικόνα 4.7 - Δοκιμή web server που καλεί τις συγκεκριμένες συναρτήσεις που υπάρχουν οι εντολές IR

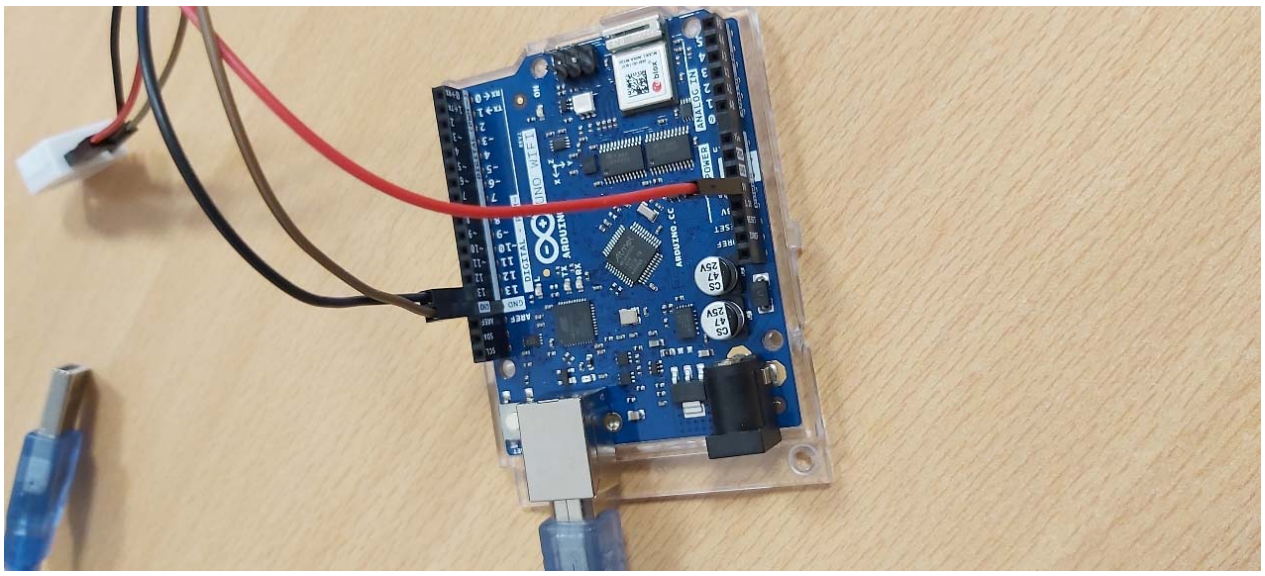
4.4.3 Αναβάθμιση κεντρικής πλακέτας

Στην συνέχεια της υλοποίησης (δεύτερη φάση) ερευνήθηκε και υιοθετήθηκε η βέλτιστη λύση που θα μπορούσαμε να έχουμε σε μικρότερο φυσικό μέγεθος, διατηρώντας την χρήση του απαραίτητου Module που ήταν υπεύθυνο για την αποστολή του IR σήματος.
Συγκεκριμένα, για την μείωση του μεγέθους της αρχικής δοκιμαστικής συσκευής, αποφασίστηκε η αντικατάσταση του κεντρικού board από Arduino Uno σε Arduino Uno Wi-Fi Rev2. Το νέο αυτό board προσφέρει επιπλέον ασύρματη συνδεσμολογία, χωρίς την απαίτηση για προμήθεια και χρήση κάποιου πρόσθετου module [08] υπεύθυνο για την δικτύωση, όπως το μεγάλο σε μέγεθος ethernet module που χρησιμοποιήθηκε στις δοκιμές και το οποίο πια καταργείται.
Ένα επιπλέον πλεονέκτημα του Arduino Uno Wi-Fi Rev2 είναι ότι αποτελεί οικονομικότερη λύση από την χρήση ενός Arduino Uno με ασύρματη προσθήκη κάποιου επιπλέον Module Wi-Fi του οποίου το κόστος είναι σχεδόν διπλάσιο το ίδιο το Arduino Uno Wi-Fi Rev2 που έχει ενσωματωμένο Wi-Fi.

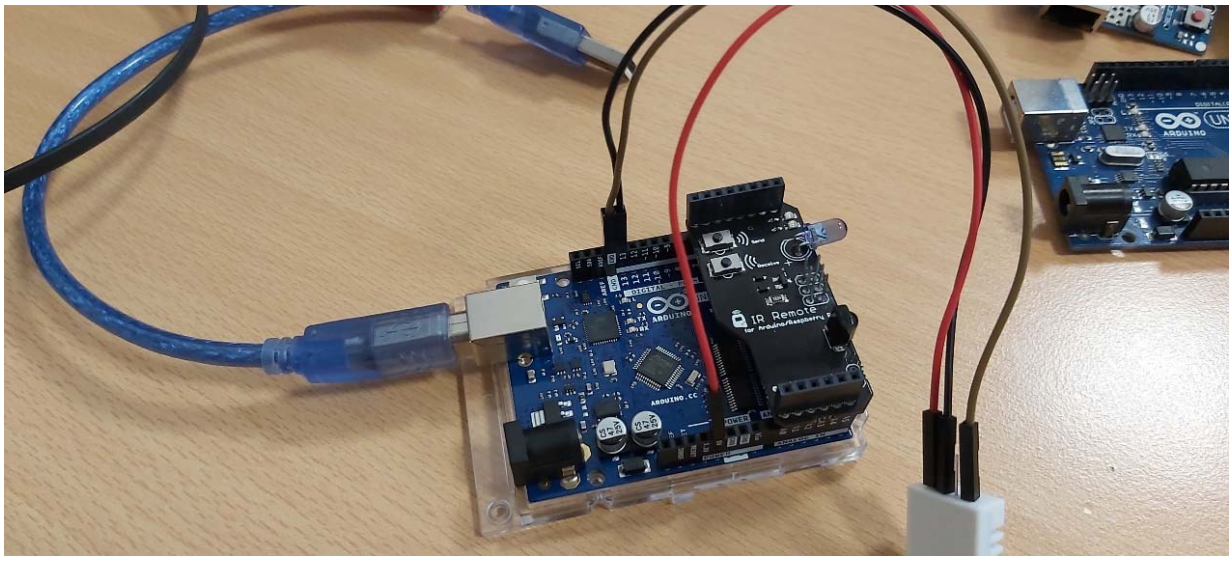
4.4.4 Εγκατάσταση αισθητήρα θερμοκρασίας και δοκιμές

Για τις ανάγκες της μέτρησης της θερμοκρασίας στο χώρο επιλέχτηκε ο αισθητήρας DHT22, ο οποίος είναι από τους πιο αξιόπιστους σε σχέση με παρόμοια πιο οικονομικά μοντέλα αισθητήρων που μπορεί να «χάνουν» μετρήσεις. Ο συγκεκριμένος είναι ικανός να μας δίνει τις μετρήσεις του χώρου με την μορφή και την συχνότητα που επιθυμούμε. Τα πιο οικονομικά μοντέλα μερικές φορές «κολλάνε» και δεν ανταποκρίνονται σε εντολές όπως λήψη μέτρησης ανά 5 λεπτά . Ο συγκεκριμένος αισθητήρας είναι εργοστασιακά προ-ρυθμισμένος για άμεση χρήση. Επίσης έχει την δυνατότητα λειτουργίας με παροχή ρεύματος εκτός από 3.3V και 5V.
Επειδή χρειαζόμασταν μόνο ένα αισθητήρα, αυτόν της θερμοκρασίας, για την υλοποίηση της συσκευής μας, και το module που χρησιμοποιήθηκε για τις υπέρυθρες ήταν έτσι διαμορφωμένο ώστε να κουμπώνει στο υπάρχον board, δεν ήταν απαραίτητο να χρησιμοποιηθεί ένα Breadboard που ενδεχομένως να συναντάται σε παρόμοιες υλοποιήσεις. Για την συνδεσμολογία, μόνο 3 καλώδια male to female χρειάστηκαν για την σύνδεση του αισθητήρα, τα οποία ήταν τύπου Solderless για να μην χρειάζεται η χρήση κολλητηριού.

Στο αναβαθμισμένο κεντρικό board τοποθετήθηκε ο αισθητήρας (εικόνα 4.8) για την θερμοκρασία DHT22 καθώς και το IR module (εικόνα 4.9) που είναι υπεύθυνο για το αποστολή των IR σημάτων.



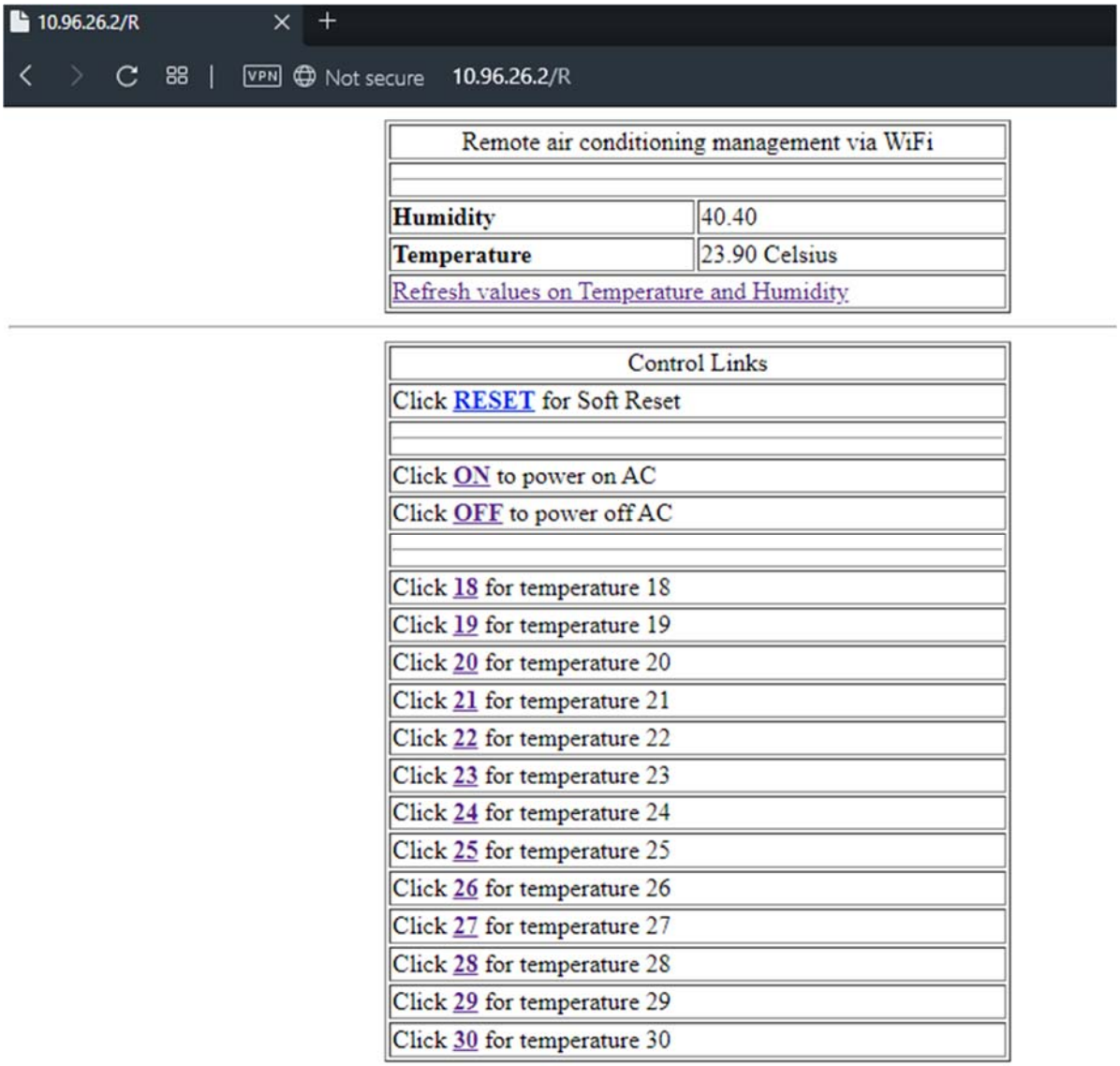
Εικόνα 4.8 - Arduino uno Rev 2 με temperature sensor και ενσωματωμένο module για Wi-Fi



Εικόνα 4.9 - Arduino uno Rev 2 μαζί με το module για το IR transmission και τον temperature sensor

Η δυνατότητα αποθηκευτικής χωρητικότητας του board είναι αρκετή ακόμα και για μια πλήρη μεταφορά όλων των εντολών του τηλεχειριστηρίου δηλαδή τόσο της ψύξης (όπως στην περίπτωσή μας) αλλά και της θέρμανσης. Αν απαιτούνταν επιπλέων χωρητικότητα υπήρχε η δυνατότητα προσθήκης οικονομικού module (περίπου στα 5 ευρώ) εξωτερικής κάρτας microSD το οποίο δεν ήταν στην παρούσα απαραίτητο.

Στο επόμενο στάδιο προγραμματισμού και αφού έχουμε καταγράψει τις εντολές στο προηγούμενο στάδιο, τροποποιούμε τον κώδικα ώστε να λειτουργεί ο web server με Wi-Fi και όχι με ethernet σύνδεση. Επίσης προσθέτουμε τις απαραίτητες βιβλιοθήκες για χρήση του αισθητήρα θερμοκρασίας. Ο προσφερόμενος πλέον web server που λειτουργεί (εικόνα 4.10) στην μονάδα Arduino uno Wi-Fi Rev 2, προσφέρει και στιγμιαία μέτρηση θερμοκρασίας μαζί με εντολές IR.



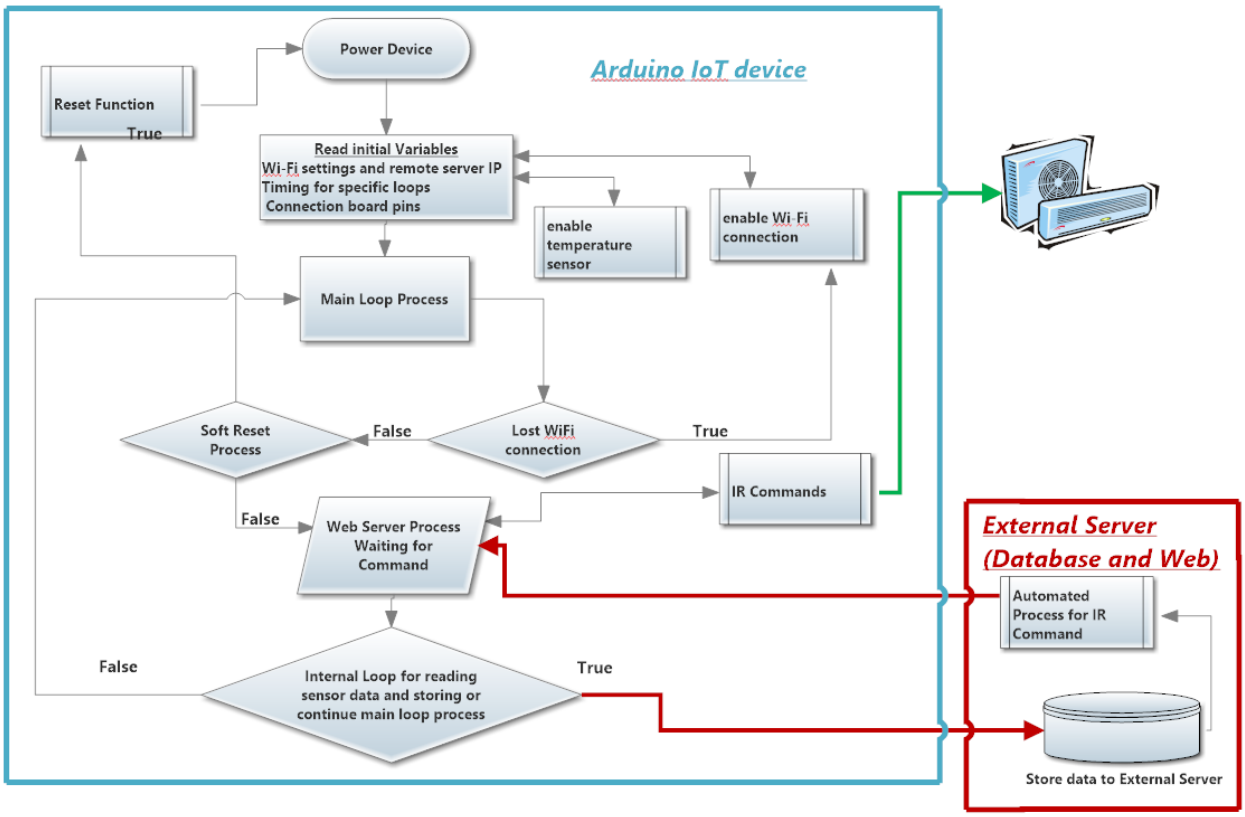
Εικόνα 4.10 - Η νέα κατάσταση του web-server

4.4.5 Έλεγχος από απόσταση και αντιμετώπιση προβλημάτων

Οι πιο πολλές εφαρμογές IoT που υλοποιούνται χρησιμοποιούν τα board του Arduino της επώνυμης ή παρεμφερής εταιρείας που ακολουθεί την συμβατότητα και την δια-λειτουργικότητα της επώνυμης κατασκευής. Αυτές με την βοήθεια αισθητήρων που τοποθετούνται πάνω τους, παρακολουθούν το

περιβάλλον. Μετά από σχετική παραμετροποίηση μπορούν να καταγράψουν, να ακολουθήσουν συγκεκριμένες αποφάσεις και να εκτελέσουν προγραμματισμένες ρουτίνες.

Όταν γίνετε προσπάθεια αυτοματοποίησης και εξάλειψης ανθρώπινης παρέμβασης πρέπει να καταγραφούν όλες οι περιπτώσεις και τα σενάρια που μπορεί να αντιμετωπίσει το υπό ανάπτυξη αυτοματοποιημένο σύστημα (εικόνα 4.11).



Εικόνα 4.11 - Process Flow Diagram

Μετά από δοκιμές και παρακολούθηση συμπεριφοράς πάνω στην υλοποίηση της IoT συσκευής μας παρατηρήθηκαν κάποια προβλήματα στα οποία έπρεπε να δοθεί ιδιαίτερη σημασία και να αντιμετωπιστούν. Αυτά παραθέτονται παρακάτω:

Πρόβλημα 1

Παρατηρήθηκε ότι μετά από μερικές μέρες (2-3) συνεχόμενης λειτουργίας η συσκευή έχανε την Wi-Fi σύνδεση ή απλά σταματούσε να λειτουργεί.

Λύση

Μετά από διάφορες δοκιμές, προστέθηκε μια συνάρτηση για soft-reset η οποία εκτελεί επανεκκίνηση της συσκευής ανά συγκεκριμένο αριθμό κύκλων λειτουργίας. Το όριο του κάθε κύκλου ορίστηκε ίδιο με την περιοδική λειτουργία που εκτελούνταν οι καταμετρήσεις της θερμοκρασίας, για να μην χαθεί

κάποια προγραμματισμένη καταγραφή δεδομένων. Ο αριθμός κύκλων ορίστηκε στους 5, άρα κάθε 4 μετρήσεις θερμοκρασίας πραγματοποιείται ένα reset.

Πρόβλημα 2

Ορισμένα access points μετά από αρκετό διάστημα παραμονής κάποιας συσκευής συνδεδεμένης σε αυτά, τις αποσυνέδεαν.

Λύση

Επιδιορθώθηκε με την ανανέωση της σύνδεσης (reconnect) που πραγματοποιείται μετά από τακτικό έλεγχο εσωτερικά της Arduino συσκευής.

Πρόβλημα 3

Έπρεπε να οριστεί η χρονική περίοδος του αυτοματοποιημένου ελέγχου της θερμοκρασίας και η αυτόματη επιλογή εντολής της νέας κατάστασης που θα αποκτούσε το κλιματιστικό.

Λύση

Δεν μπορούσε να είναι πολύ μεγάλο το χρονικό διάστημα, ούτε πολύ μικρό. Στην περίπτωση που θα επιλέγαμε μεγάλο χρονικό διάστημα υπήρχε ο κίνδυνος να ανέβει πολύ η θερμοκρασία και στην επόμενη εντολή θα δυσκολεύαμε την μονάδα να ανταπεξέλθει στη μείωση της θερμοκρασίας. Στην περίπτωση που θα επιλέγαμε μικρό χρονικό διάστημα, θα είχαμε περιττές καταγεγραμμένες μετρήσεις και κατ' επέκταση περιττές εγγραφές στην βάση δεδομένων, όπως επίσης περιττές αυτόματες εντολές προς το κλιματιστικό για διατήρηση της ίδιας θερμοκρασία. Η επιλογή να πραγματοποίει μετρήσεις κάθε 30 λεπτά της ώρας ήταν ικανοποιητική, κρατούσε την θερμοκρασία σε καλά επίπεδα χωρίς να έχουμε μεγάλες διαφορές θερμοκρασίας όταν πραγματοποιούνταν ξανά μέτρηση. Το μοναδικό μειονέκτημα, αν μπορεί να θεωρηθεί μειονέκτημα, είναι ότι σε συνδυασμό με την πρώτη λύση της αυτόματης επανεκκίνησης οι τελικές εγγραφές στην βάση είναι ανά τέσσερις, με μια ώρα κενό και ξανά τέσσερις. Δηλαδή αν καταγράφουμε την πρώτη μέτρηση στις 10:00, η επόμενη είναι 10:30, 11:00, 11:30, χάνεται η πέμπτη καταμέτρηση των 12:00 λόγω της επανεκκίνησης και καταγράφουμε ξανά στις 12:30. Αυτό βέβαια δεν παρουσίασε κάποια αισθητή διαφορά στην λειτουργία και θεωρήθηκε μη σημαντικό.

Πρόβλημα 4

Εκτέλεση και επανάληψη επιπρόσθετων εργασιών ταυτόχρονα με την κεντρική ροή.

Λύση

Για το λόγο ότι ο προγραμματισμός στο Arduino εκτελείται σειριακά και επαναλαμβάνεται ότι βρίσκεται στην Loop συνάρτηση, ορισμένες φορές χρειάζεται να εκτελέσουμε κάποια άλλη διαδικασία παράλληλα σε συγκεκριμένο χρόνο χωρίς να σταματήσουμε την ροή του προγράμματος.

Η επαναλαμβανόμενη διαδικασία αναπτύχθηκε βασισμένη στην εντολή Millis [05], η οποία με χρονικό ορισμό του προγραμματιστή εκτελεί παράλληλα μια διαδικασία χωρίς να διακόπτει την κεντρική ροή. Η μονάδα χρονικού διαστήματος ορίζετε σε milliseconds.

Πρόβλημα 5

Επιλογή κατάλληλης θερμοκρασίας που θα αποσταλεί αυτόματα μετά την αντίστοιχη μέτρηση.

Λύση

Η καταγεγραμμένη μέτρηση θερμοκρασίας στρογγυλοποιείται και συγκρίνεται με την κεντρική επιθυμητή τιμή που ορίζει ο χρήστης ότι θέλει να έχει ο χώρος. Στην περίπτωσή μας, η προεπιλεγμένη τιμή θερμοκρασίας είναι αυτή των 22^{o}C. Στην συνέχεια στέλνετε η ανάλογη εντολή θερμοκρασίας που είναι το αποτέλεσμα της αφαίρεσης της θερμοκρασίας καταμέτρησης μείον τρεις βαθμούς (– 3).

Παράδειγμα για καλύτερη κατανόηση μιας μέτρησης και ενός κύκλου ελέγχου.

Ορισμένη τιμή από τον χρήστη - 22 βαθμούς

Καταμέτρηση θερμοκρασίας 25,9βαθμοί (στρογγυλοποιείται στους 26βαθμούς)

Έλεγχος αν το 26<=22

- Αν *ναι* κλείνει το κλιματιστικό γιατί πετύχαμε μικρότερη ή ίδια τιμή με την επιθυμητή
- αν *όχι* αφαιρεί 3 βαθμούς και στέλνει την ανάλογη εντολή, στο παράδειγμα στέλνει το 23

Πρόβλημα 6

Πρόβλημα επικοινωνίας με βάση δεδομένων για αποθήκευση μετρήσεων

Λύση

Σε μια διασύνδεση δυο μηχανών που είναι σε διαφορετικές τοποθεσίες, πρέπει να συμπεριλάβουμε το σενάριο της απώλειας επικοινωνίας και να προσπαθήσουμε να επιδιορθώσουμε τη σύνδεση σε κάποιο βαθμό. Υιοθετήθηκε η λύση της επανεκκίνηση ασύρματης σύνδεσης με το access point και προσπάθεια επανυποβολής των μετρήσεων.

Πρόβλημα 7

Εύρεση σωστής θέσης για την τοποθέτηση της συσκευής.

Λύση

Παρόλο που από την περιγραφή μπορεί να θεωρηθεί κάτι πολύ απλό, θέλει ιδιαίτερη προσοχή η συσκευή να τοποθετείται μακριά από το air flow του κλιματιστικού. Στην περίπτωση που είναι κοντά στην ροή του αέρα, τα δεδομένα από τις μετρήσεις θα είναι εντελώς λανθασμένα. Λόγω κοντινής ύπαρξης των 2 ντουλαπών στο χώρο του δωματίου με το κλιματιστικό, επιλέχτηκε να τοποθετηθεί η συσκευή πάνω στις ντουλάπες. Η ροή του αέρα ρυθμίστηκε από τις γρίλιες χειροκίνητα να είναι προς τα δεξιά του κλιματιστικού και η συσκευή με τον αισθητήρα τοποθετήθηκε αριστερά του κλιματιστικού. Σχετική δοκιμή που έγινε με την ροή του αέρα να είναι πάνω στον αισθητήρα ήταν αποτυχημένη γιατί έστελνε συνεχώς εντολές απενεργοποίησης της μονάδα χωρίς να έχει φτάσει η επιθυμητή θερμοκρασία στο χώρο.

Πρόβλημα 8

Έλεγχος της εφαρμογής για την κατάσταση της IoT συσκευής

Λύση

Σε περίπτωση που δεν λειτουργεί η IoT συσκευή πρέπει να προλαμβάνουμε τον χρήστη από την περιττή αποστολή εντολών. Ο έλεγχος πραγματοποιείται με Ping της διεύθυνση που έχει πάρει η IoT συσκευή. Αν απαντήσει τότε μπορεί να δεχτεί εντολές αφού είναι Online. Αν δεν απάντησει, πιθανά να υπάρχει κάποιο πρόβλημα στην τροφοδοσία της IoT συσκευής (α) ή στην ασύρματη σύνδεση της με το access point (β) . Στην περίπτωση της τροφοδοσία (α) αναμένουμε και όταν αποκατασταθεί θα συνεχίσει να εκτελεί την αυτοματοποιημένη διαδικασία. Στην περίπτωση της προβληματικής ασύρματης σύνδεσης (β) θα αποκατασταθεί μόνη της με την λύση που αναλύθηκε σε προηγούμενο πρόβλημα, πραγματοποιώντας ανανέωση της σύνδεσης (reconnect).

Ιδιαιτερότητα του συστήματος και όχι πρόβλημα είναι ότι δεν δόθηκε η δυνατότητα της απενεργοποίησης της αυτοματοποίησης από την χρήστη. Τι σημαίνει αυτό με παράδειγμα σεναρίου.

- Ο χώρος του δωματίου έχει 20 βαθμούς.
- Ο επιθυμητός στόχος είναι 22 βαθμοί, άρα το κλιματιστικό πρέπει να είναι κλειστό.
- Ο χρήστης βλέπει ότι η μονάδα του κλιματιστικού είναι απενεργοποιημένη, επιλέγει όμως και στέλνει από το κινητό του την θερμοκρασία 18 βαθμούς.
- Η μονάδα ανάβει και λειτουργεί στους 18 βαθμούς.

- Όταν φτάσει η χρονική περίοδο καταμέτρησης / ελέγχου θα ξανά κλείσει το κλιματιστικό αυτόματα αναιρώντας τις επιθυμίες του χρήστη.

Το σύστημα ακολουθεί πιστά και προσαρμόζει την αυτοματοποιημένη διαδικασία μόνο με την θερμοκρασία του επιθυμητού στόχου.

4.4.6 Εξωτερικοί Διακομιστές

Στην υλοποίηση του συστήματος IoT, σημαντικό κομβικό σημείο αποτελεί η συνεχής αποθήκευση δεδομένων και πληροφοριών, είτε αυτές προέρχονται από την αυτοματοποιημένη διαδικασία, είτε από την αναπτυσσόμενη συσκευή, είτε από τον χρήστη, είτε από άλλες βοηθητικές εξωτερικές μονάδες μέτρησης τύπου smart power meter.
Όλα τα παραπάνω δεδομένα και πληροφορίες λαμβάνονται και αποθηκεύονται στους διακομιστές, Αναπτυχθήκαν δύο διαφορετικοί διακομιστές σε ένα μηχάνημα με λειτουργικό σύστημα Windows 10 64bit και χρησιμοποιήθηκε το ελεύθερο πακέτο λογισμικού και υπηρεσιών XAMMP, αναλυτικότερα:

- ένας web server (Apache web server)
- ένας database server (MySQL)

Στο μηχάνημα αυτό έγινε ανάθεση δημόσιας ip διεύθυνσης για να υπάρχει δυνατότητα πρόσβασης και από εξωτερικό δίκτυο. Επίσης και οι δυο διακομιστές δέχονται συνδέσεις από την IoT συσκευή (Arduino) και από την εφαρμογή που αναπτύχθηκε για τα κινητά.

4.4.6.1 Διακομιστής Βάσης Δεδομένων (Database Server)

Στον Database server έχουμε την καταγραφή πληροφοριών που αφορούν είτε μετρήσεις από την IoT συσκευή είτε εντολές που στέλνονται χειροκίνητες από τον χρήστη ή αυτοματοποιημένες από τον web server. Τα δεδομένα από τις εξωτερικές βοηθητικές συσκευές δεν καταγράφονται αυτόματα στους αντίστοιχους πίνακες αλλά εισάγονται ανά τακτά χρονικά διαστήματα ώστε να εμπλουτίζεται το σύστημα με νέα δεδομένα. Αποσπάσματα των πινάκων με τα πεδία που χρησιμοποιούνται στην βάση δεδομένων παρουσιάζονται στην συνέχεια.

Αναλυτικότερα υπάρχουν πέντε πίνακες που χρησιμοποιούνται για διαφορετική καταγραφή πληροφοριών:

1. Ο Πίνακας ac_temp_log

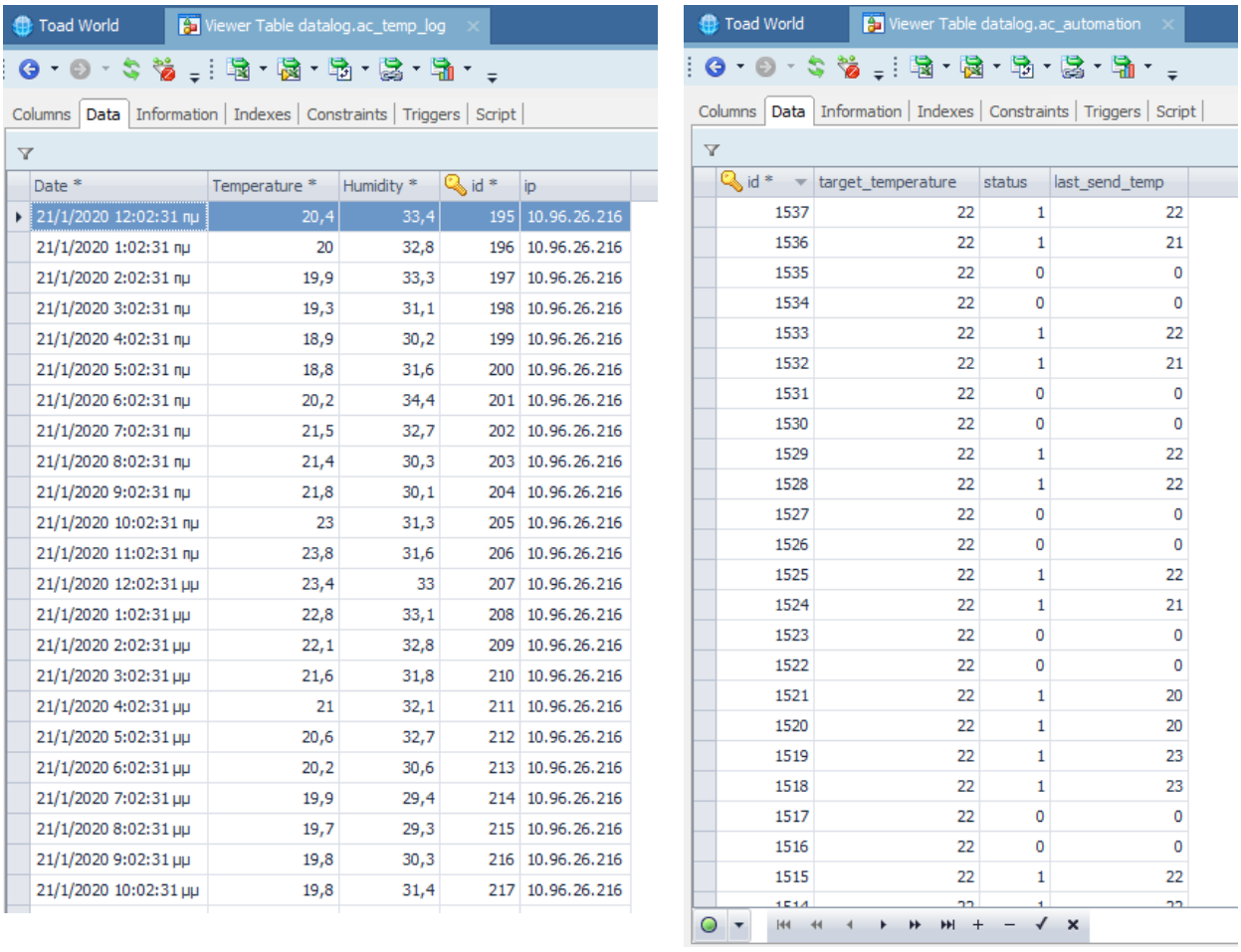
   Σε αυτόν καταγράφονται όλες οι μετρήσεις που προέρχονται από την αναπτυσσόμενη συσκευή του Arduino προς την βάση δεδομένων, δηλαδή:

- Η ημερομηνία (timestamp)
- Η θερμοκρασία του δωματίου
- Η υγρασία του δωματίου
- Ένα id για κλειδί ως primary key
- Η διεύθυνση ip του Arduino

2. Ο Πίνακας ac_automation

Καταγράφονται μέσα σε αυτόν όλες οι εντολές που στέλνονται από τον διακομιστή (web server) προς την αναπτυσσόμενη συσκευή του Arduino, δηλαδή:

- Ένα id για κλειδί ως primary key
- Η θερμοκρασία που υπάρχει σας στόχος επίτευξης
- Το status που ανταποκρίνεται στην κατάσταση του κλιματιστικού
- Η τελευταία εντολή θερμοκρασίας που στάλθηκε. (η τιμή 0 ανταποκρίνεται στην κατάσταση της απενεργοποιημένης μονάδας κλιματισμού)



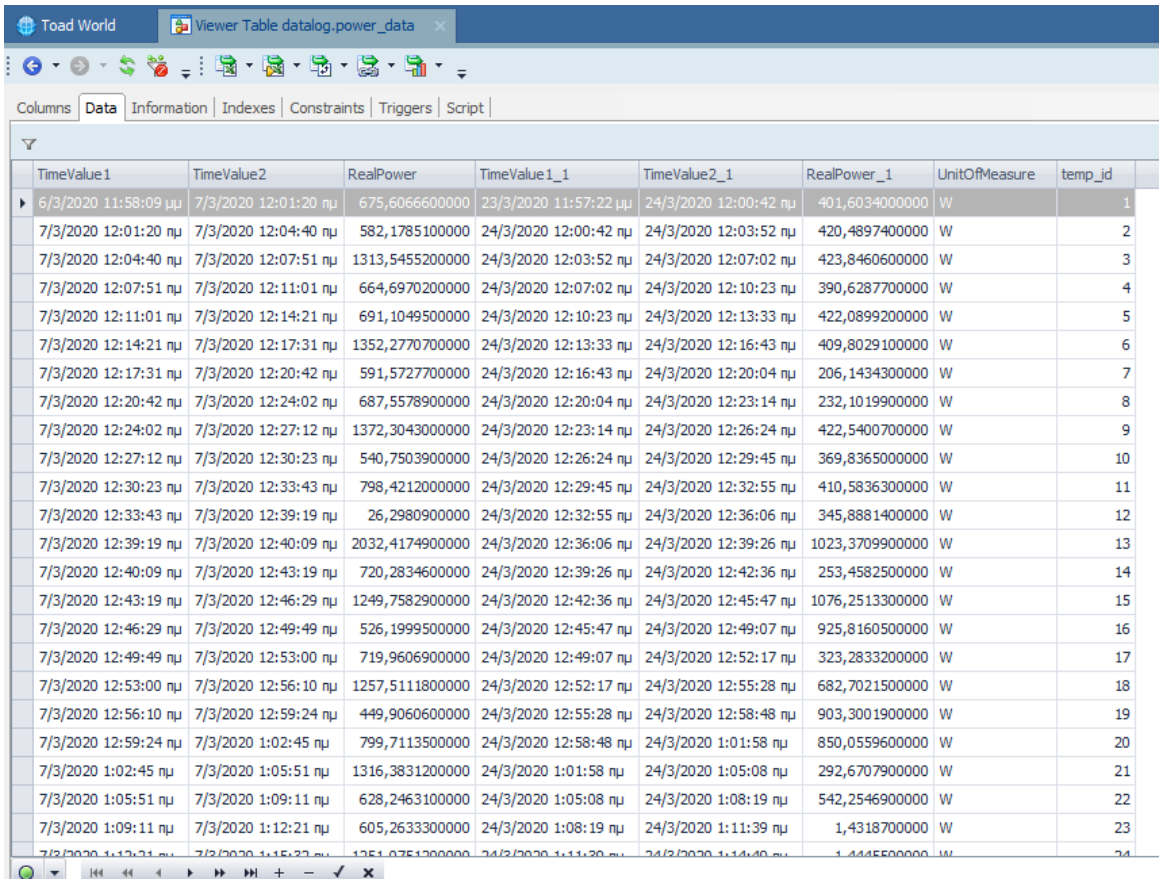
Εικόνα 4.12 – Πίνακας ac_temp_log και ac_automation

3. Ο Πίνακας power_data

Μέσα σε αυτόν εισάγονται οι συνεχείς μετρήσεις που αφορούν στην κατανάλωση ενέργειας του κλιματιστικού, δεδομένα δηλαδή που μας παρείχε η βοηθητική συσκευή καταμέτρησης ενέργειας (energy power meter). Στον συγκεκριμένο πίνακα έχουν εισαχθεί δυο διαφορετικά σετ της ιδίας ώρας αλλά διαφορετικής ημερομηνίας. Στο πρώτο σετ (7-3-2020 έως 8-3-2020) έγινε εσκεμμένη επιλογή να απενεργοποιηθεί η Arduino συσκευή και να λειτουργεί συνεχώς το κλιματιστικό για 2 ημέρες (48ωρες) στους 22 βαθμούς. Στο δεύτερο σετ (24-3-2020 έως 25-3-2020) οι τιμές είναι με ενεργοποιημένη την Arduino συσκευή και χρήση του αυτοματισμού.

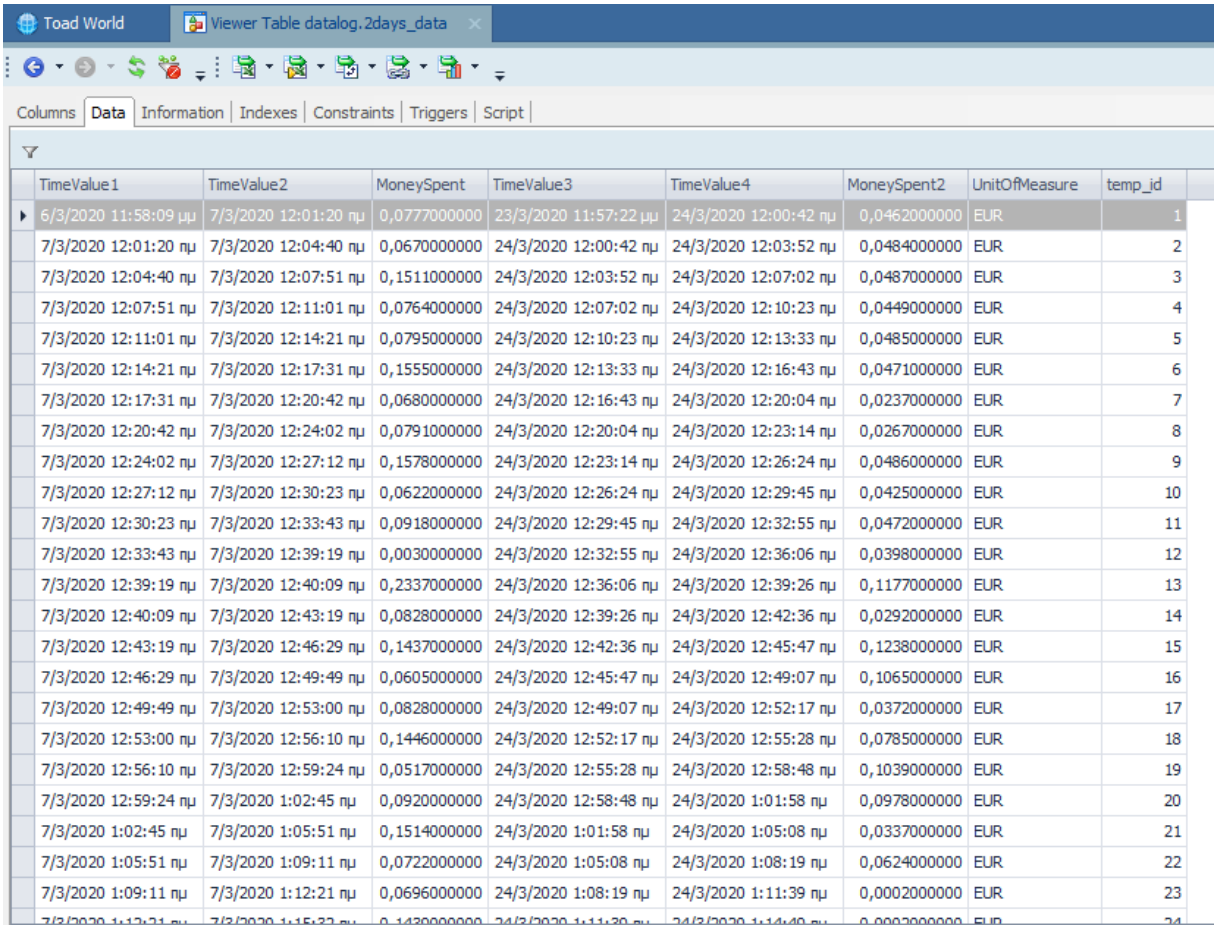
4. Ο Πίνακας 2days_data

Μέσα σε αυτόν εισάγονται οι συνεχείς μετρήσεις που αφορούν στο πραγματικό κόστος κατανάλωσης του κλιματιστικού, δεδομένα δηλαδή που μας παρείχε η βοηθητική συσκευή καταμέτρησης ενέργειας (energy power meter). Στον συγκεκριμένο πίνακα έχουν εισαχθεί δυο διαφορετικά σετ της ιδίας ώρας αλλά διαφορετικής ημερομηνίας, κατά αντιστοιχία με τον πίνακα power data όπως περιγράφεται παραπάνω.



TimeValue1	TimeValue2	RealPower	TimeValue1_1	TimeValue2_1	RealPower_1	UnitOfMeasure	temp_id
6/3/2020 11:58:09 μμ	7/3/2020 12:01:20 πμ	675,6066600000	23/3/2020 11:57:22 μμ	24/3/2020 12:00:42 πμ	401,6034000000	W	1
7/3/2020 12:01:20 πμ	7/3/2020 12:04:40 πμ	582,1785100000	24/3/2020 12:00:42 πμ	24/3/2020 12:03:52 πμ	420,4897400000	W	2
7/3/2020 12:04:40 πμ	7/3/2020 12:07:51 πμ	1313,5455200000	24/3/2020 12:03:52 πμ	24/3/2020 12:07:02 πμ	423,8460600000	W	3
7/3/2020 12:07:51 πμ	7/3/2020 12:11:01 πμ	664,6970200000	24/3/2020 12:07:02 πμ	24/3/2020 12:10:23 πμ	390,6287700000	W	4
7/3/2020 12:11:01 πμ	7/3/2020 12:14:21 πμ	691,1049500000	24/3/2020 12:10:23 πμ	24/3/2020 12:13:33 πμ	422,0899200000	W	5
7/3/2020 12:14:21 πμ	7/3/2020 12:17:31 πμ	1352,2770700000	24/3/2020 12:13:33 πμ	24/3/2020 12:16:43 πμ	409,8029100000	W	6
7/3/2020 12:17:31 πμ	7/3/2020 12:20:42 πμ	591,5727700000	24/3/2020 12:16:43 πμ	24/3/2020 12:20:04 πμ	206,1434300000	W	7
7/3/2020 12:20:42 πμ	7/3/2020 12:24:02 πμ	687,5578900000	24/3/2020 12:20:04 πμ	24/3/2020 12:23:14 πμ	232,1019900000	W	8
7/3/2020 12:24:02 πμ	7/3/2020 12:27:12 πμ	1372,3043000000	24/3/2020 12:23:14 πμ	24/3/2020 12:26:24 πμ	422,5400700000	W	9
7/3/2020 12:27:12 πμ	7/3/2020 12:30:23 πμ	540,7503900000	24/3/2020 12:26:24 πμ	24/3/2020 12:29:45 πμ	369,8365000000	W	10
7/3/2020 12:30:23 πμ	7/3/2020 12:33:43 πμ	798,4212000000	24/3/2020 12:29:45 πμ	24/3/2020 12:32:55 πμ	410,5836300000	W	11
7/3/2020 12:33:43 πμ	7/3/2020 12:39:19 πμ	26,2980900000	24/3/2020 12:32:55 πμ	24/3/2020 12:36:06 πμ	345,8881400000	W	12
7/3/2020 12:39:19 πμ	7/3/2020 12:40:09 πμ	2032,4174900000	24/3/2020 12:36:06 πμ	24/3/2020 12:39:26 πμ	1023,3709900000	W	13
7/3/2020 12:40:09 πμ	7/3/2020 12:43:19 πμ	720,2834600000	24/3/2020 12:39:26 πμ	24/3/2020 12:42:36 πμ	253,4582500000	W	14
7/3/2020 12:43:19 πμ	7/3/2020 12:46:29 πμ	1249,7582900000	24/3/2020 12:42:36 πμ	24/3/2020 12:45:47 πμ	1076,2513300000	W	15
7/3/2020 12:46:29 πμ	7/3/2020 12:49:49 πμ	526,1999500000	24/3/2020 12:45:47 πμ	24/3/2020 12:49:07 πμ	925,8160500000	W	16
7/3/2020 12:49:49 πμ	7/3/2020 12:53:00 πμ	719,9606900000	24/3/2020 12:49:07 πμ	24/3/2020 12:52:17 πμ	323,2833200000	W	17
7/3/2020 12:53:00 πμ	7/3/2020 12:56:10 πμ	1257,5111800000	24/3/2020 12:52:17 πμ	24/3/2020 12:55:28 πμ	682,7021500000	W	18
7/3/2020 12:56:10 πμ	7/3/2020 12:59:24 πμ	449,9060600000	24/3/2020 12:55:28 πμ	24/3/2020 12:58:48 πμ	903,3001900000	W	19
7/3/2020 12:59:24 πμ	7/3/2020 1:02:45 πμ	799,7113500000	24/3/2020 12:58:48 πμ	24/3/2020 1:01:58 πμ	850,0559600000	W	20
7/3/2020 1:02:45 πμ	7/3/2020 1:05:51 πμ	1316,3831200000	24/3/2020 1:01:58 πμ	24/3/2020 1:05:08 πμ	292,6707900000	W	21
7/3/2020 1:05:51 πμ	7/3/2020 1:09:11 πμ	628,2463100000	24/3/2020 1:05:08 πμ	24/3/2020 1:08:19 πμ	542,2546900000	W	22
7/3/2020 1:09:11 πμ	7/3/2020 1:12:21 πμ	605,2633300000	24/3/2020 1:08:19 πμ	24/3/2020 1:11:39 πμ	1,4318700000	W	23

Εικόνα 4.13 - Πίνακας power_data

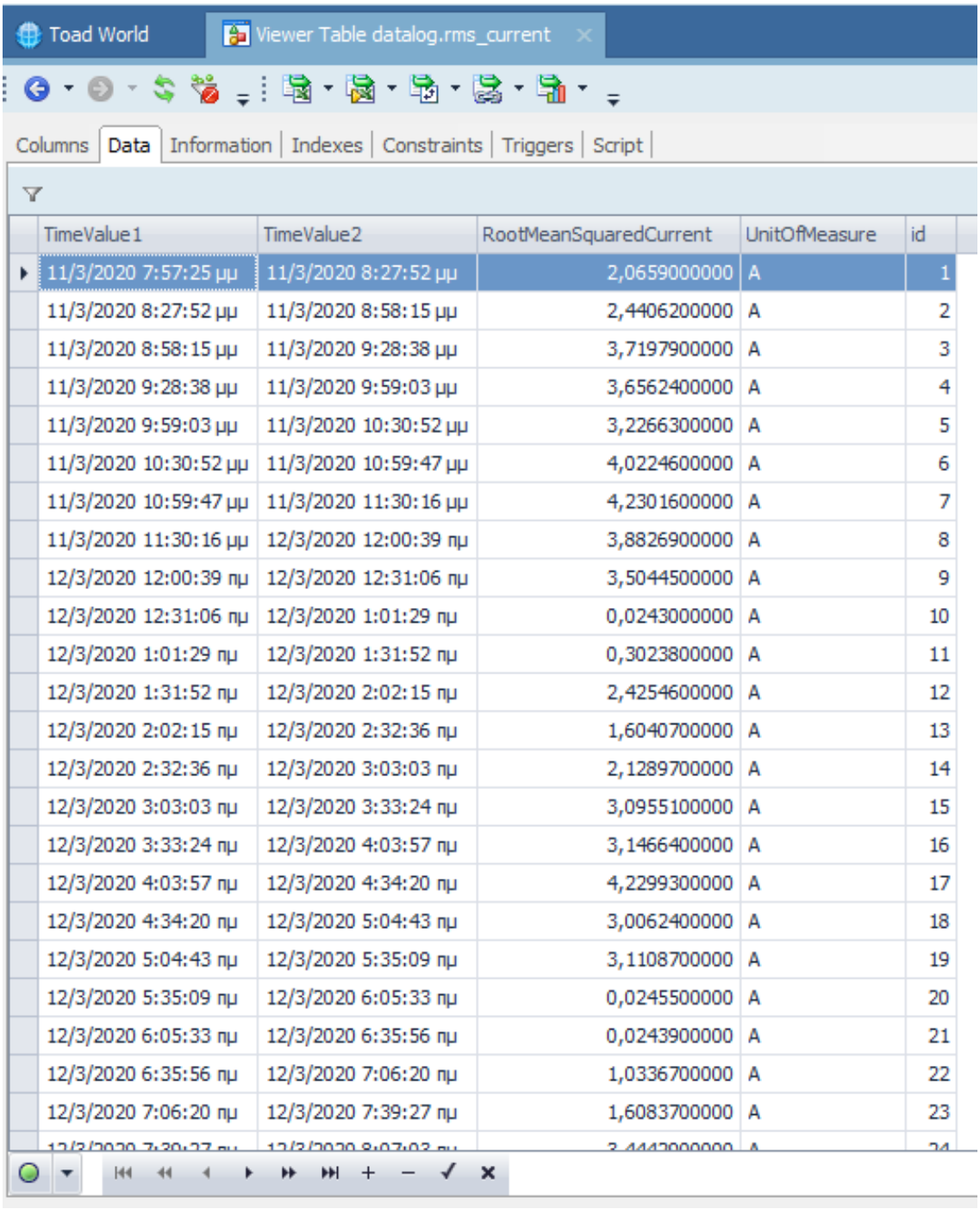


TimeValue1	TimeValue2	MoneySpent	TimeValue3	TimeValue4	MoneySpent2	UnitOfMeasure	temp_id
6/3/2020 11:58:09 μμ	7/3/2020 12:01:20 πμ	0,0777000000	23/3/2020 11:57:22 μμ	24/3/2020 12:00:42 πμ	0,0462000000	EUR	1
7/3/2020 12:01:20 πμ	7/3/2020 12:04:40 πμ	0,0670000000	24/3/2020 12:00:42 πμ	24/3/2020 12:03:52 πμ	0,0484000000	EUR	2
7/3/2020 12:04:40 πμ	7/3/2020 12:07:51 πμ	0,1511000000	24/3/2020 12:03:52 πμ	24/3/2020 12:07:02 πμ	0,0487000000	EUR	3
7/3/2020 12:07:51 πμ	7/3/2020 12:11:01 πμ	0,0764000000	24/3/2020 12:07:02 πμ	24/3/2020 12:10:23 πμ	0,0449000000	EUR	4
7/3/2020 12:11:01 πμ	7/3/2020 12:14:21 πμ	0,0795000000	24/3/2020 12:10:23 πμ	24/3/2020 12:13:33 πμ	0,0485000000	EUR	5
7/3/2020 12:14:21 πμ	7/3/2020 12:17:31 πμ	0,1555000000	24/3/2020 12:13:33 πμ	24/3/2020 12:16:43 πμ	0,0471000000	EUR	6
7/3/2020 12:17:31 πμ	7/3/2020 12:20:42 πμ	0,0680000000	24/3/2020 12:16:43 πμ	24/3/2020 12:20:04 πμ	0,0237000000	EUR	7
7/3/2020 12:20:42 πμ	7/3/2020 12:24:02 πμ	0,0791000000	24/3/2020 12:20:04 πμ	24/3/2020 12:23:14 πμ	0,0267000000	EUR	8
7/3/2020 12:24:02 πμ	7/3/2020 12:27:12 πμ	0,1578000000	24/3/2020 12:23:14 πμ	24/3/2020 12:26:24 πμ	0,0486000000	EUR	9
7/3/2020 12:27:12 πμ	7/3/2020 12:30:23 πμ	0,0622000000	24/3/2020 12:26:24 πμ	24/3/2020 12:29:45 πμ	0,0425000000	EUR	10
7/3/2020 12:30:23 πμ	7/3/2020 12:33:43 πμ	0,0918000000	24/3/2020 12:29:45 πμ	24/3/2020 12:32:55 πμ	0,0472000000	EUR	11
7/3/2020 12:33:43 πμ	7/3/2020 12:39:19 πμ	0,0030000000	24/3/2020 12:32:55 πμ	24/3/2020 12:36:06 πμ	0,0398000000	EUR	12
7/3/2020 12:39:19 πμ	7/3/2020 12:40:09 πμ	0,2337000000	24/3/2020 12:36:06 πμ	24/3/2020 12:39:26 πμ	0,1177000000	EUR	13
7/3/2020 12:40:09 πμ	7/3/2020 12:43:19 πμ	0,0828000000	24/3/2020 12:39:26 πμ	24/3/2020 12:42:36 πμ	0,0292000000	EUR	14
7/3/2020 12:43:19 πμ	7/3/2020 12:46:29 πμ	0,1437000000	24/3/2020 12:42:36 πμ	24/3/2020 12:45:47 πμ	0,1238000000	EUR	15
7/3/2020 12:46:29 πμ	7/3/2020 12:49:49 πμ	0,0605000000	24/3/2020 12:45:47 πμ	24/3/2020 12:49:07 πμ	0,1065000000	EUR	16
7/3/2020 12:49:49 πμ	7/3/2020 12:53:00 πμ	0,0828000000	24/3/2020 12:49:07 πμ	24/3/2020 12:52:17 πμ	0,0372000000	EUR	17
7/3/2020 12:53:00 πμ	7/3/2020 12:56:10 πμ	0,1446000000	24/3/2020 12:52:17 πμ	24/3/2020 12:55:28 πμ	0,0785000000	EUR	18
7/3/2020 12:56:10 πμ	7/3/2020 12:59:24 πμ	0,0517000000	24/3/2020 12:55:28 πμ	24/3/2020 12:58:48 πμ	0,1039000000	EUR	19
7/3/2020 12:59:24 πμ	7/3/2020 1:02:45 πμ	0,0920000000	24/3/2020 12:58:48 πμ	24/3/2020 1:01:58 πμ	0,0978000000	EUR	20
7/3/2020 1:02:45 πμ	7/3/2020 1:05:51 πμ	0,1514000000	24/3/2020 1:01:58 πμ	24/3/2020 1:05:08 πμ	0,0337000000	EUR	21
7/3/2020 1:05:51 πμ	7/3/2020 1:09:11 πμ	0,0722000000	24/3/2020 1:05:08 πμ	24/3/2020 1:08:19 πμ	0,0624000000	EUR	22
7/3/2020 1:09:11 πμ	7/3/2020 1:12:21 πμ	0,0696000000	24/3/2020 1:08:19 πμ	24/3/2020 1:11:39 πμ	0,0002000000	EUR	23

Εικόνα 4.14 - Πίνακας 2days_data

5. Ο Πίνακας rms_current

Εισάγονται μέσα σε αυτόν οι συνέχεις μετρήσεις που αφορούν το Root Mean Squared Current (Irms, A) κατανάλωση του κλιματιστικού. Στον συγκεκριμένο πίνακα έχουν εισαχθεί δεδομένα από 24/3/2020 έως 1/4/2020.



TimeValue1	TimeValue2	RootMeanSquaredCurrent	UnitOfMeasure	id
11/3/2020 7:57:25 μμ	11/3/2020 8:27:52 μμ	2,0659000000	A	1
11/3/2020 8:27:52 μμ	11/3/2020 8:58:15 μμ	2,4406200000	A	2
11/3/2020 8:58:15 μμ	11/3/2020 9:28:38 μμ	3,7197900000	A	3
11/3/2020 9:28:38 μμ	11/3/2020 9:59:03 μμ	3,6562400000	A	4
11/3/2020 9:59:03 μμ	11/3/2020 10:30:52 μμ	3,2266300000	A	5
11/3/2020 10:30:52 μμ	11/3/2020 10:59:47 μμ	4,0224600000	A	6
11/3/2020 10:59:47 μμ	11/3/2020 11:30:16 μμ	4,2301600000	A	7
11/3/2020 11:30:16 μμ	12/3/2020 12:00:39 πμ	3,8826900000	A	8
12/3/2020 12:00:39 πμ	12/3/2020 12:31:06 πμ	3,5044500000	A	9
12/3/2020 12:31:06 πμ	12/3/2020 1:01:29 πμ	0,0243000000	A	10
12/3/2020 1:01:29 πμ	12/3/2020 1:31:52 πμ	0,3023800000	A	11
12/3/2020 1:31:52 πμ	12/3/2020 2:02:15 πμ	2,4254600000	A	12
12/3/2020 2:02:15 πμ	12/3/2020 2:32:36 πμ	1,6040700000	A	13
12/3/2020 2:32:36 πμ	12/3/2020 3:03:03 πμ	2,1289700000	A	14
12/3/2020 3:03:03 πμ	12/3/2020 3:33:24 πμ	3,0955100000	A	15
12/3/2020 3:33:24 πμ	12/3/2020 4:03:57 πμ	3,1466400000	A	16
12/3/2020 4:03:57 πμ	12/3/2020 4:34:20 πμ	4,2299300000	A	17
12/3/2020 4:34:20 πμ	12/3/2020 5:04:43 πμ	3,0062400000	A	18
12/3/2020 5:04:43 πμ	12/3/2020 5:35:09 πμ	3,1108700000	A	19
12/3/2020 5:35:09 πμ	12/3/2020 6:05:33 πμ	0,0245500000	A	20
12/3/2020 6:05:33 πμ	12/3/2020 6:35:56 πμ	0,0243900000	A	21
12/3/2020 6:35:56 πμ	12/3/2020 7:06:20 πμ	1,0336700000	A	22
12/3/2020 7:06:20 πμ	12/3/2020 7:39:27 πμ	1,6083700000	A	23

Εικόνα 4.15 - Πίνακας rms_current

4.4.6.2 Διακομιστής ιστού (Web Server)

Στον web server υπάρχει όλη η αυτοματοποίηση του συστήματος που αναπτύχθηκε καλύπτοντας τις παρακάτω λειτουργίες:

- Αποθήκευση στην βάση δεδομένων του στόχου επιθυμητής θερμοκρασίας του δωματίου, με εισαγωγή πληροφορίας από την εφαρμογή για κινητά (mobile app)
- Λήψη και αποθήκευση περιοδικής μέτρησης θερμοκρασίας / υγρασίας με εισαγωγή πληροφορίας από την αναπτυσσόμενη συσκευή Arduino.
- Υπολογισμός αυτόματης θερμοκρασίας σε σχέση με τον στόχο επιθυμητής θερμοκρασίας του δωματίου
- Αποστολή αυτόματης θερμοκρασίας στην IoT συσκευή
- Αποθήκευση στην βάση δεδομένων της απεσταλμένης θερμοκρασία.
- Έλεγχος λειτουργίας της IoT συσκευής με την χρήση ping στην διεύθυνση IP που έχει πάρει από το Wi-Fi access point.
- Αποστολή καταστάσεων λειτουργίας προς το Mobile app τόσο της IoT συσκευής όσο και του κλιματιστικού με σχετικές εικόνες
- Δημιουργία δυναμικών γραφικών παραστάσεων από τα καταγεγραμμένα δεδομένα
- Δημιουργία στατικών γραφικών παραστάσεων από τα καταγεγραμμένα δεδομένα

4.4.7 Σχεδιασμός User interface

Η mobile εφαρμογή αναπτύχθηκε με τη χρήση της ελεύθερης πλατφόρμα ανάπτυξης Android Studio IDE. Έγινε επιλογή του θέματος εμφάνισης και συγκεκριμένα να υπάρχει κρυμμένο το μενού με τις επιλογές και να εμφανίζεται με επιλογή από τον χρήστη. Με αυτό τον τρόπο εκμεταλλευόμαστε περισσότερο χώρο στην οθόνη του κινητού. Ο προσανατολισμός των διάφορων οθονών αναλύθηκε και είναι κλειδωμένος στον κατάλληλο προσανατολισμό (κάθετο ή οριζόντιο) ώστε να προσφέρει το βέλτιστο φιλικό περιβάλλον στον τελικό χρήστη.
Το μενού αποτελείται από δυο κατηγορίες επιλογών οι πληροφοριακές και οι διαχειριστικές.
Στις πληροφοριακές έχουμε:

- A/C Status Device
  Μια ενημερωτική καρτέλα που παρουσιάζει την κατάσταση των 2 συσκευών ελέγχου. Του κλιματιστικού και της IoT συσκευής.

- Γραφήματα

  Παρουσιάζονται διάφορα γραφήματα/στατιστικά για το κόστος, την κατανάλωση, την θερμοκρασία και /με την υγρασία, ένα δυναμικό γράφημα για την θερμοκρασία με επιλογή του ημερολογιακού ορισμού της περιόδου από τον χρήστη. RMS Current με συνολικά δεδομένα από 24/3/2020 έως 1/4/2020, και Δεδομένα Θερμοκρασίας από 11 Μαρτίου 2020 έως σήμερα. Τα γραφήματα έγιναν με την βοήθεια της Google Visualization[16].

- Ιστορικό

  Παρουσιάζεται το ιστορικό των καταμετρήσεων καθώς και το ιστορικό των απεσταλμένων εντολών. Μια χρήσιμη ενημερωτική καρτέλα που αποδεικνύει την σωστή λειτουργία της IoT συσκευής. Στο κάτω τμήμα συμπεριλαμβάνεται η ζωντανή αναμετάδοση (Live streaming) του κλιματιστικού για οπτική επιβεβαίωση της κατάστασής του.

- About

  Παρουσιάζονται στατιστικά και ποσοστά ενεργειακής εξοικονόμησης με την συσκευή IoT και χωρίς.

Οι διαχειριστικές επιλογές είναι δυο:

- Απομακρυσμένες εντολές

  Μια δυνατότητα παράκαμψης των αυτόματων εντολών από ίδιο τον χρήστη. Ισχύουν μέχρι να επανέλθει ο χρονικός αυτοματισμός του συστήματος.

- Ορισμός στόχου

  Ο χρήστης ορίζει τον στόχο επιθυμητής θερμοκρασίας του δωματίου, η οποία χρησιμοποιείται έπειτα από την αυτοματοποιημένη διαδικασία. Ο στόχος αυτός ρυθμίζεται με τρόπο φιλικό προς τον χρήστη του smartphone, κάνοντας εξομοίωση πλήκτρου τύπου κυκλικού ροοστάτη (dimmer).

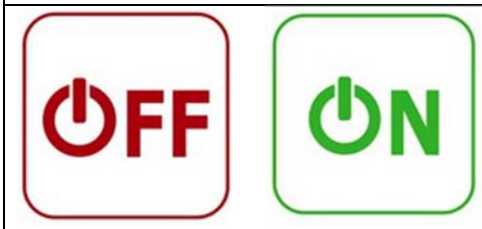
  Στην καρτέλα με τον ορισμό στόχου η υλοποίηση του κυκλικού ροοστάτη παραμετροποιήθηκε για χρήση στις ανάγκες και στο range που απαιτούνταν για το Mobile app. Η αρχική έκδοση αναπτύχθηκε και διατίθεται από τον Anthony Terrien[17] με ελεύθερη διάθεση χρήσης [The MIT License (MIT)] για ενσωματώσεις του κώδικα, μορφοποιήσεις κλπ.

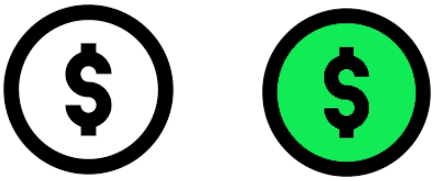
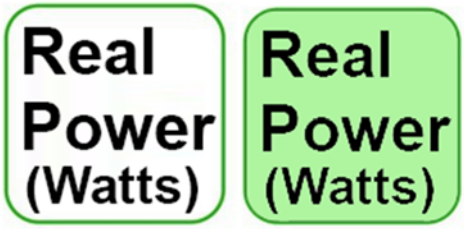
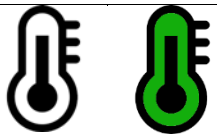
[16] https://developers.google.com/chart/interactive/docs/reference
[17] http://anthonyterrien.com/demo/knob/

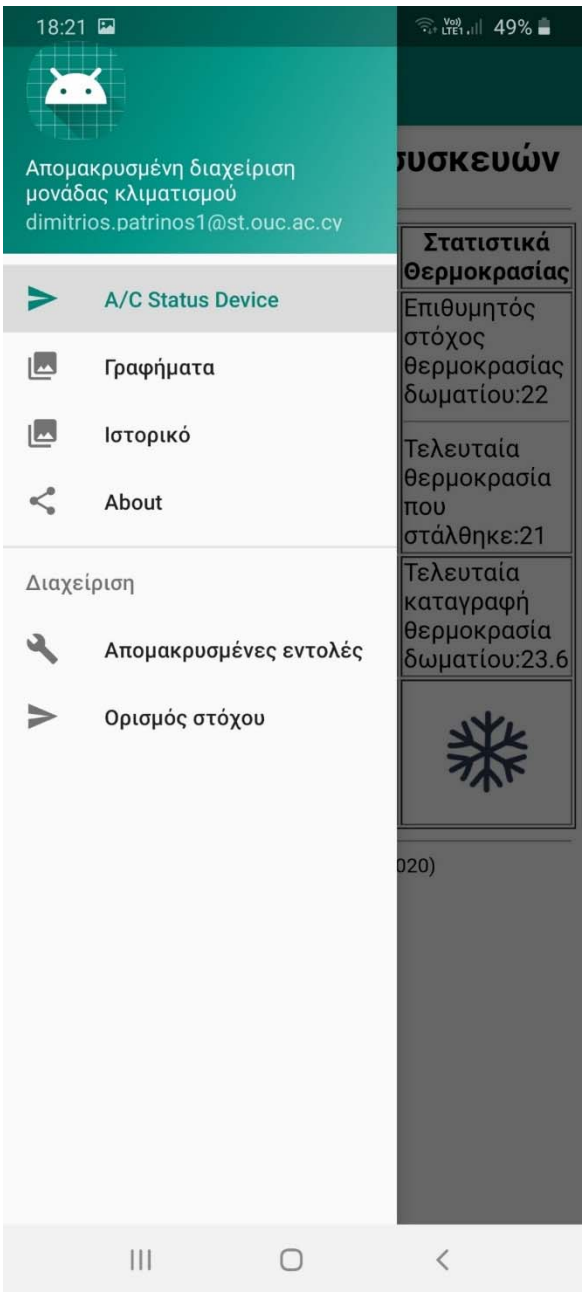
4.4.7.1 Δημιουργία εικονιδίων και στιγμιότυπα οθόνης εφαρμογής κινητού

Δημιουργήθηκαν μια σειρά από εικόνες / buttons για χρήση από της εφαρμογή του κινητού. Παρουσιάζονται παρακάτω σε συνοπτικό πίνακα οι διαφορετικές καταστάσεις που μεταβαίνει η εφαρμογή μετά από ελέγχους, καθώς και τα στιγμιότυπα οθόνης από την εφαρμογή του κινητού (εικόνες 4.16 έως 4.26).

Πίνακας εικονιδίων για χρήση στο UI του χρήστη

Εικονίδια	Επεξήγηση
	Κατάσταση κλιματιστικής μονάδας και κουμπί για εναλλαγή λειτουργίας
	Κατάσταση IoT συσκευής Ενημερωτικό εικονίδιο, για μελλοντική προσθήκη και εναλλαγή ελέγχου του Mode της κλιματιστικής μονάδας.
  	Οι διαθέσιμες θερμοκρασίες που υποστηρίζει το συγκεκριμένο μοντέλο κλιματιστικού είναι από 18°C έως 29 °C. Η πρώτη εικόνα (θερμοκρασία σε άσπρο φόντο) επιλέγεται από τον χρήστη ως επιθυμητή θερμοκρασία (και μπορεί να κυμαίνεται από 18°C έως 29 °C) (ενεργό πλήκτρο). Η δεύτερη εικόνα (θερμοκρασία σε πράσινο φόντο) παρουσιάζει την επιλεγμένη τρέχουσα θερμοκρασίας που εκτελείται στο κλιματιστικό (ενεργό πλήκτρο). Η τρίτη εικόνα (θερμοκρασία σε άσπρο φόντο με σχόλιο offline R/C) παρουσιάζεται όταν υπάρχει πρόβλημα με την IoT συσκευή και αποτελεί πλήκτρο ανενεργό (δίνει μόνο πληροφορία)

	Γράφημα για θερμοκρασία / υγρασία. Όταν είναι πράσινο (κατάσταση στα δεξιά) είναι ενεργό.
	Γράφημα με μετρήσεις πραγματικού κόστους με και χωρίς IoT συσκευή. Όταν είναι πράσινο (κατάσταση στα δεξιά) είναι ενεργό.
	Γράφημα με μετρήσεις κατανάλωσης ενέργειας με και χωρίς IoT συσκευή. Όταν είναι πράσινο (κατάσταση στα δεξιά) είναι ενεργό.
	Γράφημα θερμοκρασίας με δυναμική ημερολογιακή επιλογή από τον χρήστη. Όταν είναι πράσινο (κατάσταση στα δεξιά) είναι ενεργό.
	Επιπλέον διαχωρισμός των γραφημάτων κόστους και ενέργειας σε 12 ώρα τμήματα ώστε να είναι ακόμα πιο κατανοητά.
	Γράφημα δεδομένων θερμοκρασίας από 11 Μαρτίου 2020 έως σήμερα, περίοδος που εκτελείται η τελική έκδοση λογισμικού. Όταν είναι πράσινο (κατάσταση στα δεξιά) είναι ενεργό.
	RMS Current με συνολικά δεδομένα 24/3/2020 έως 1/4/2020 Όταν είναι πράσινο (κατάσταση στα δεξιά) είναι ενεργό.

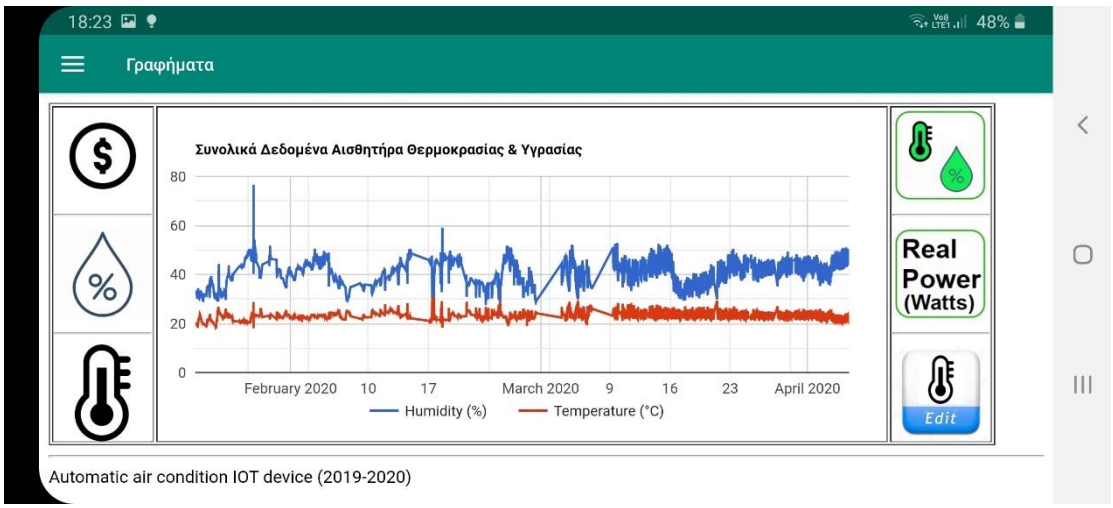


Εικόνα 4.16 - μενού από mobile app

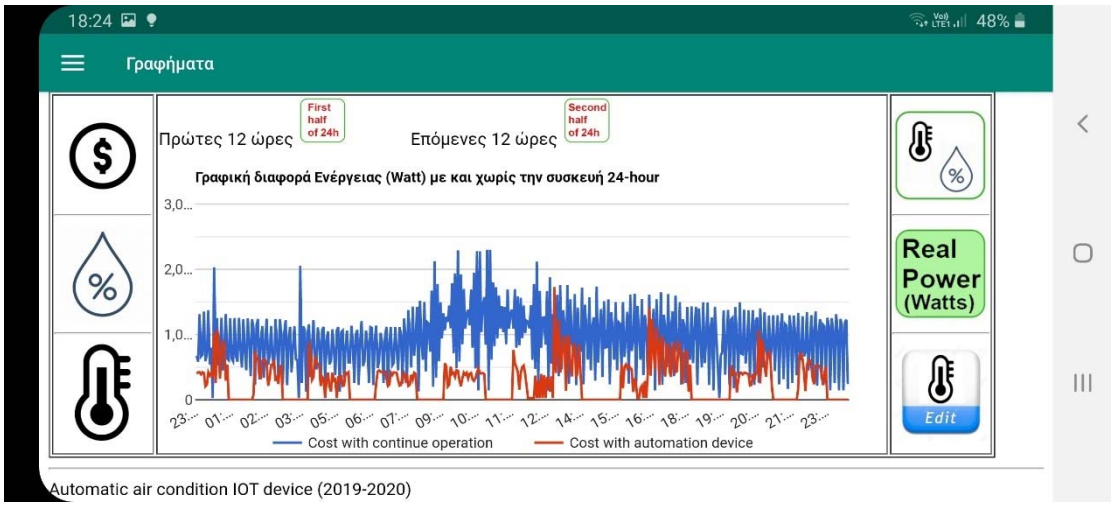
Εικόνα 4.17 – Αρχική οθόνη από UI mobile app

Εικόνα 4.17 – Οθόνη παράκαμψης εντολών UI mobile app

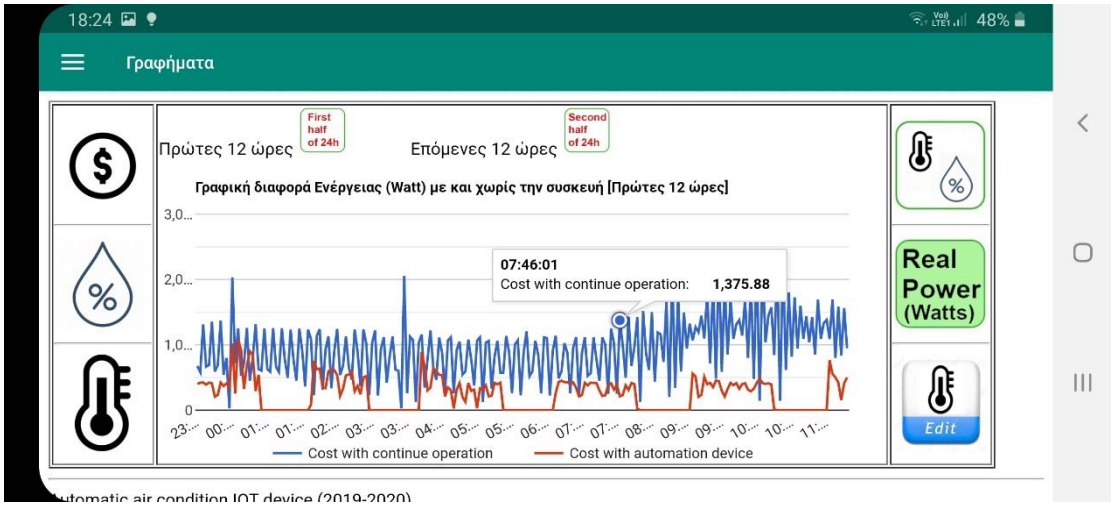
Εικόνα 4.18 – Οθόνη ορισμού στόχου θερμοκρασίας (temperature threshold) UI mobile app



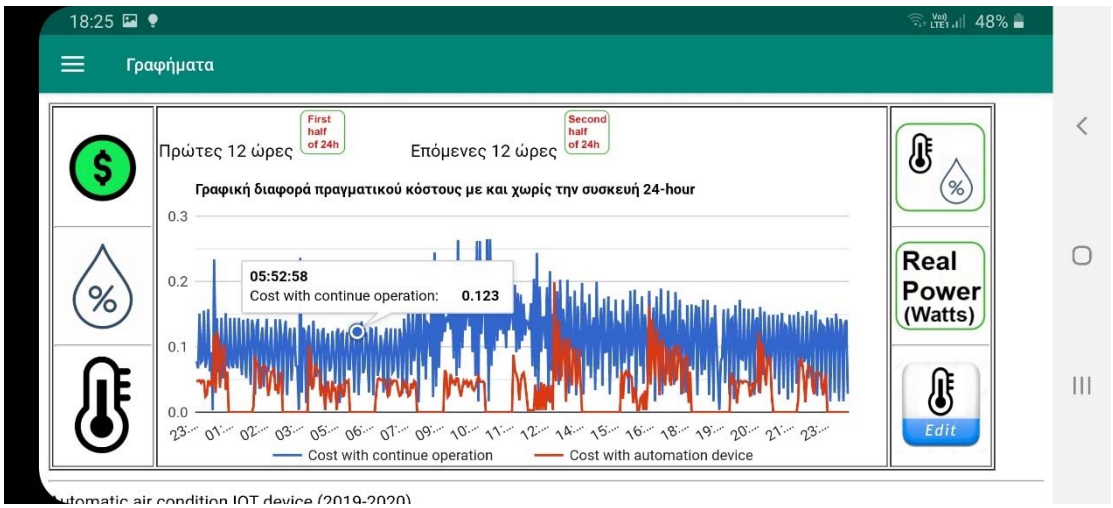
Εικόνα 4.19 – Γραφήματα από μετρήσεις (α)



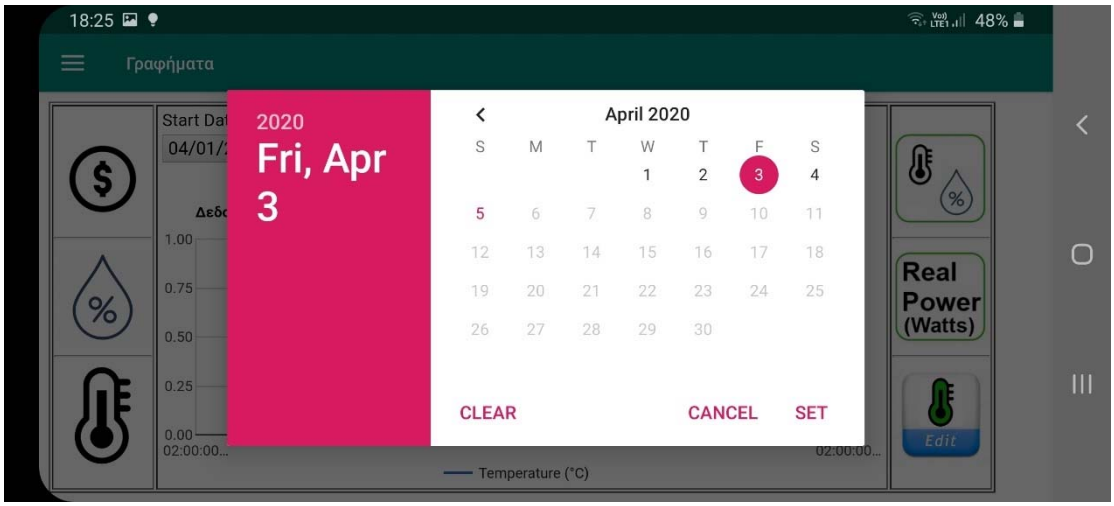
Εικόνα 4.20 - Γραφήματα από μετρήσεις (β)



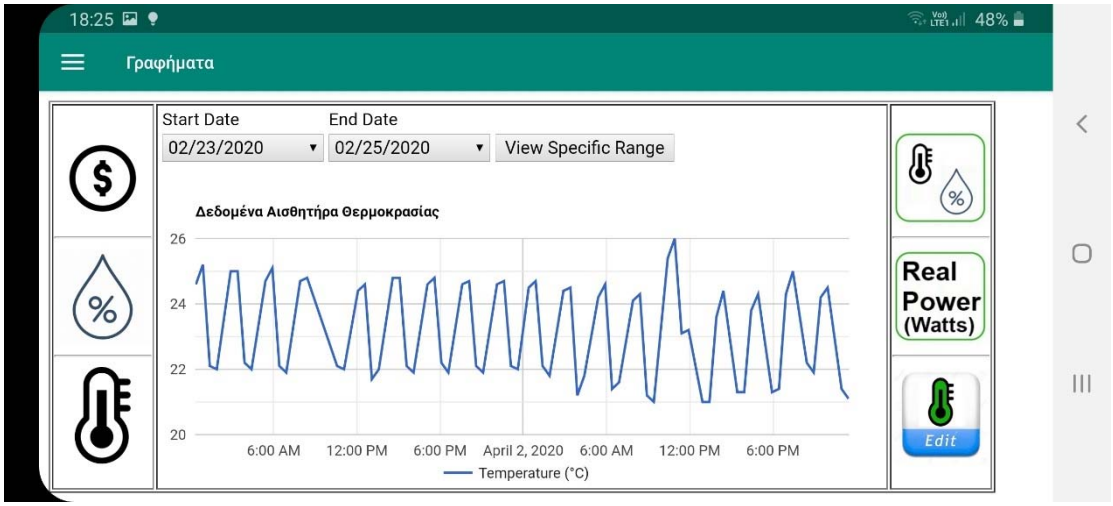
Εικόνα 4.21 - Γραφήματα από μετρήσεις (γ)



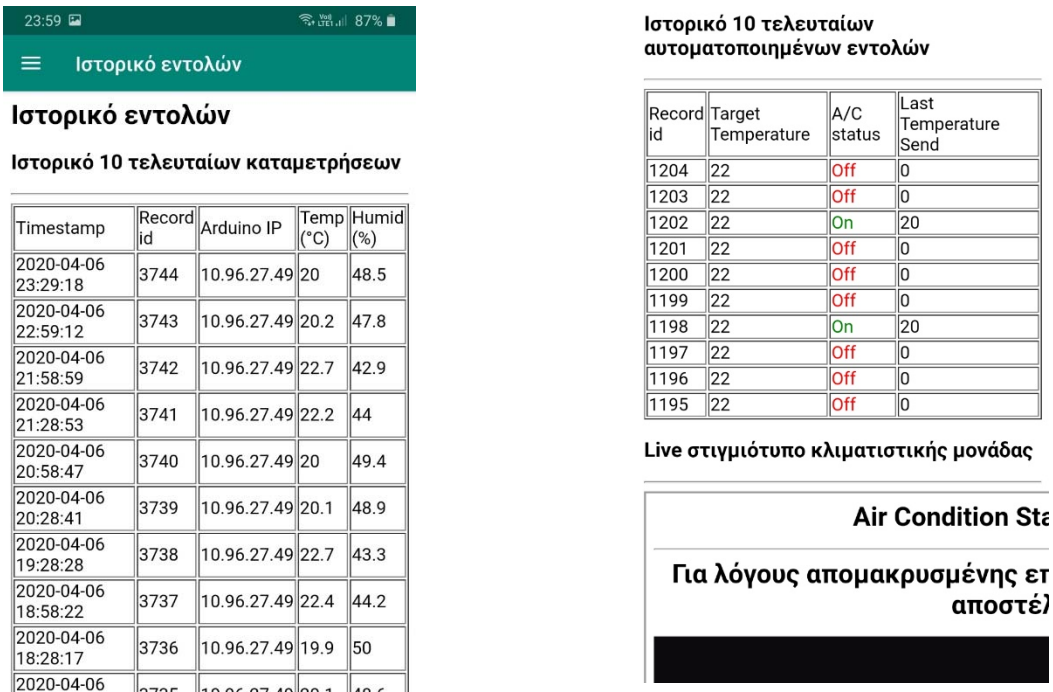
Εικόνα 4.22 - Γραφήματα από μετρήσεις (δ)



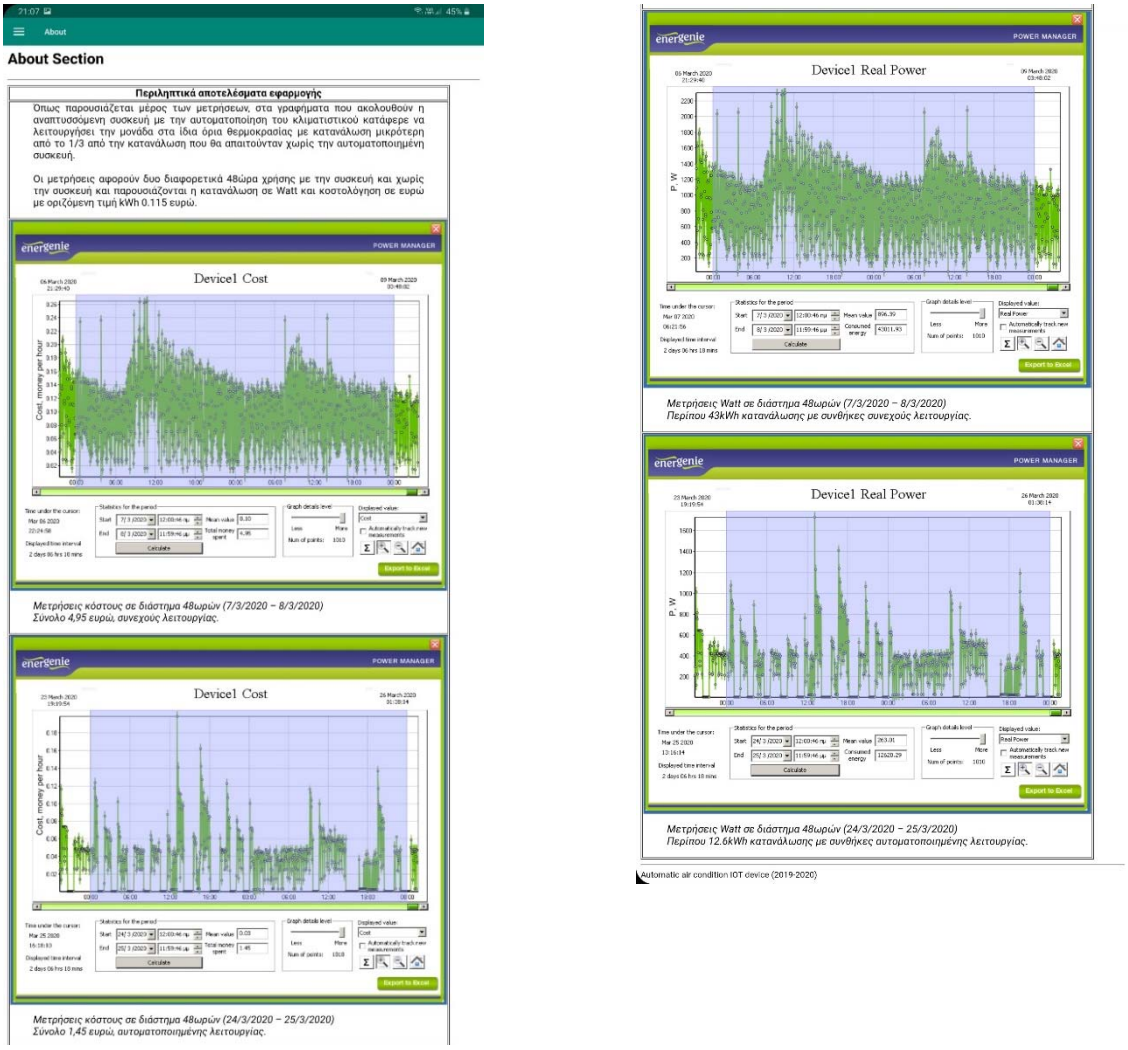
Εικόνα 4.23 - Γραφήματα από μετρήσεις (ε)



Εικόνα 4.24 - Γραφήματα από μετρήσεις (ζ)



Εικόνα 4.25 – Στιγμιότυπο οθόνης ιστορικού κινήσεων και εντολών UI mobile app



Εικόνα 4.26 – Στιγμιότυπο οθόνης about section UI mobile app

4.5 Arduino server

4.5.1 Υλοποίηση και περιγραφή Arduino

Η υλοποίηση κώδικα που αφορά τον προγραμματισμό του Arduino ήταν μια σταδιακή και συνεχές διαδικασία που με δοκιμές και βελτιώσεις οριστικοποιήθηκε σε μια σταθερή τελική έκδοση. Η παραδοτέα έκδοση του λογισμικού είναι παραμετροποιημένη ώστε κάθε συσκευή με αισθητήρα θερμοκρασίας να κατέχει σταθερή ip διεύθυνση και να μπορούν να συνυπάρχουν με άλλες ίδιες συσκευές στο κτίριο αλλά και να εντοπίζονται στο χώρο του κτιρίου. Το τελικό λογισμικό αποτελείται από ένα web διακομιστή, ο οποίος αναμένει μέσω διαδικτύου εντολές του κλιματιστικού. Αυτός ο διακομιστής προσφέρει πρόσβαση και μέσω απλού browser, που χρησιμοποιήθηκε για δοκιμές μέχρι να υλοποιηθεί η εφαρμογή για τα κινητά.
Παράλληλα εκτελείται ένας συνεχής βρόχος ανά 30 λεπτά όπου καταγραφεί μετρήσεις θερμοκρασίας οι οποίες αποστέλνονται για αποθήκευση σε εξωτερικό κεντρικό διακομιστή.
Ο αυτοματοποιημένος έλεγχος πραγματοποιείται στον εξωτερικό κεντρικό διακομιστή και ενημερώνει την συσκευή για την εκτέλεση της εντολής. Επιπλέον έχουν αναπτυχθεί ρουτίνες ασφαλείας ώστε να διατηρηθεί η αξιοπιστία που πρέπει να κατέχει ένα IoT σύστημα με κυριότερες:

- Την αυτοματοποίηση της επανασύνδεση του Module με το Wi-Fi access point αν χαθεί η σύνδεση
- Την πραγματοποίηση αυτοματοποιημένης επανεκκίνησης της συσκευής για καλύτερη διαχείριση της μνήμης

4.5.2 Κόστος κατασκευής

Αφού παρουσιάστηκαν τα διαφορετικά υλικά που είναι διαθέσιμα και οι τελικές επιλογές που έγιναν για την υλοποίηση, ενδεικτικά παρουσιάζεται το κόστος των υλικών/εξαρτημάτων μιας τέτοιας IoT συσκευής.

Περιγραφή είδους	Αρ. Τεμαχίων	Κόστος σε €
Αισθητήρας Θερμοκρασίας / Υγρασίας DHT22	1	6,72
M/F Solderless Flexible Breadboard Jumper Cable Wires For Arduino	Pack – 65 (χρησιμοποιήθηκαν 3)	4,00
Arduino UNO Wi-Fi R2	1	39,90
HVAC IR Remote module for Arduino	1	58,03
Έξοδα Αποστολής		3,00
Smart Energy Meter	1	40
ΣΥΝΟΛΟ		**(IoT - 111,65) 151,65 €**

4.6 Σχεδιάγραμμα (schematics) συσκευής

4.6.1 Χαρακτηριστικά του Arduino uno Rev 2

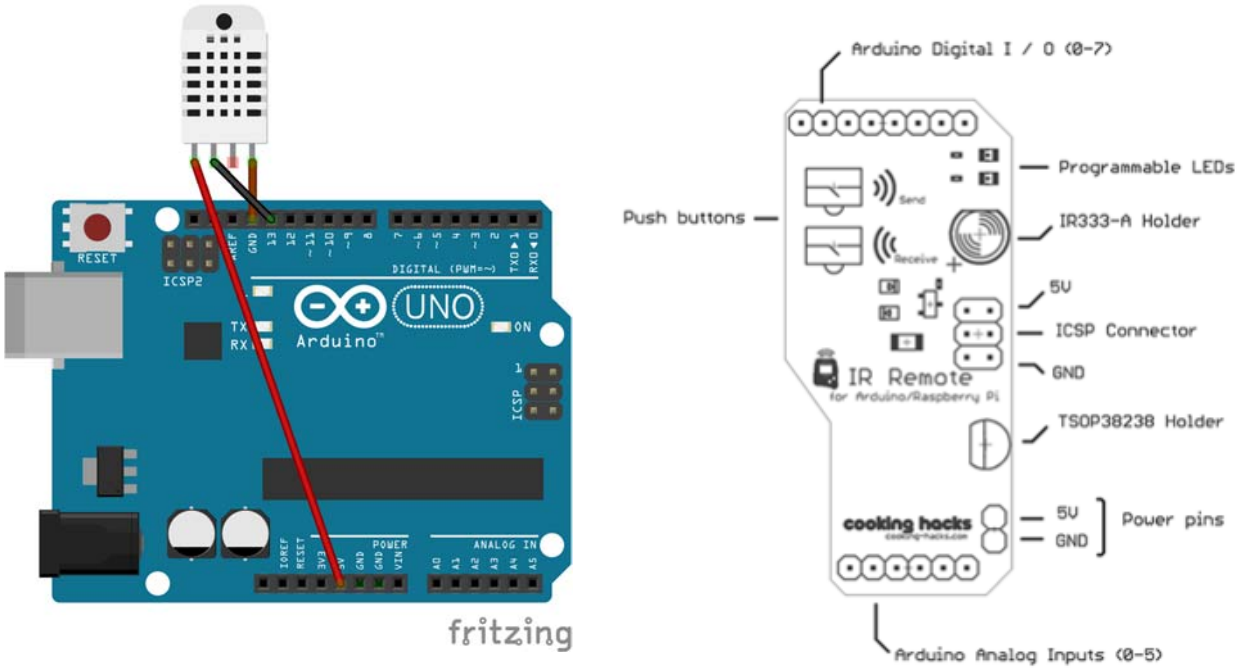
Το Arduino Uno Rev 2 είναι ένα board που ενσωματώνει τον καινούργιο μικροεπεξεργαστή 8-bit από την Microchip. Έχει ενσωματωμένο IMU (Inertial Measurement Unit) [18] το οποίο ορίζεται ως αισθητήρας 9-άξονων και μετρά προσανατολισμό, ταχύτητα, βαρυτικές δυνάμεις σε συνδυασμό με το επιταχυνσιόμετρο, γυροσκόπιο, Magnetometer. Ακόμα χρησιμοποιεί τον ασφαλή ECC608 crypto chip accelerator. Η μονάδα Wi-Fi είναι ένα αυτόνομο SoC με ενσωματωμένη στοίβα πρωτοκόλλου TCP / IP που μπορούν να παρέχουν πρόσβαση σε ένα δίκτυο Wi-Fi, ή να ενεργεί και ως σημείο πρόσβασης.
Το Arduino Uno WiFi Rev 2 έχει:

- 14 ψηφιακές εισόδους/εξόδους,
- 5 μπορούν να χρησιμοποιηθούν σας ως PWM έξοδοι
- 6 αναλογικές εισόδους
- μια σύνδεση USB
- μια υποδοχή ρεύματος
- μια κεφαλίδα ICSP
- και ένα κουμπί επαναφοράς.

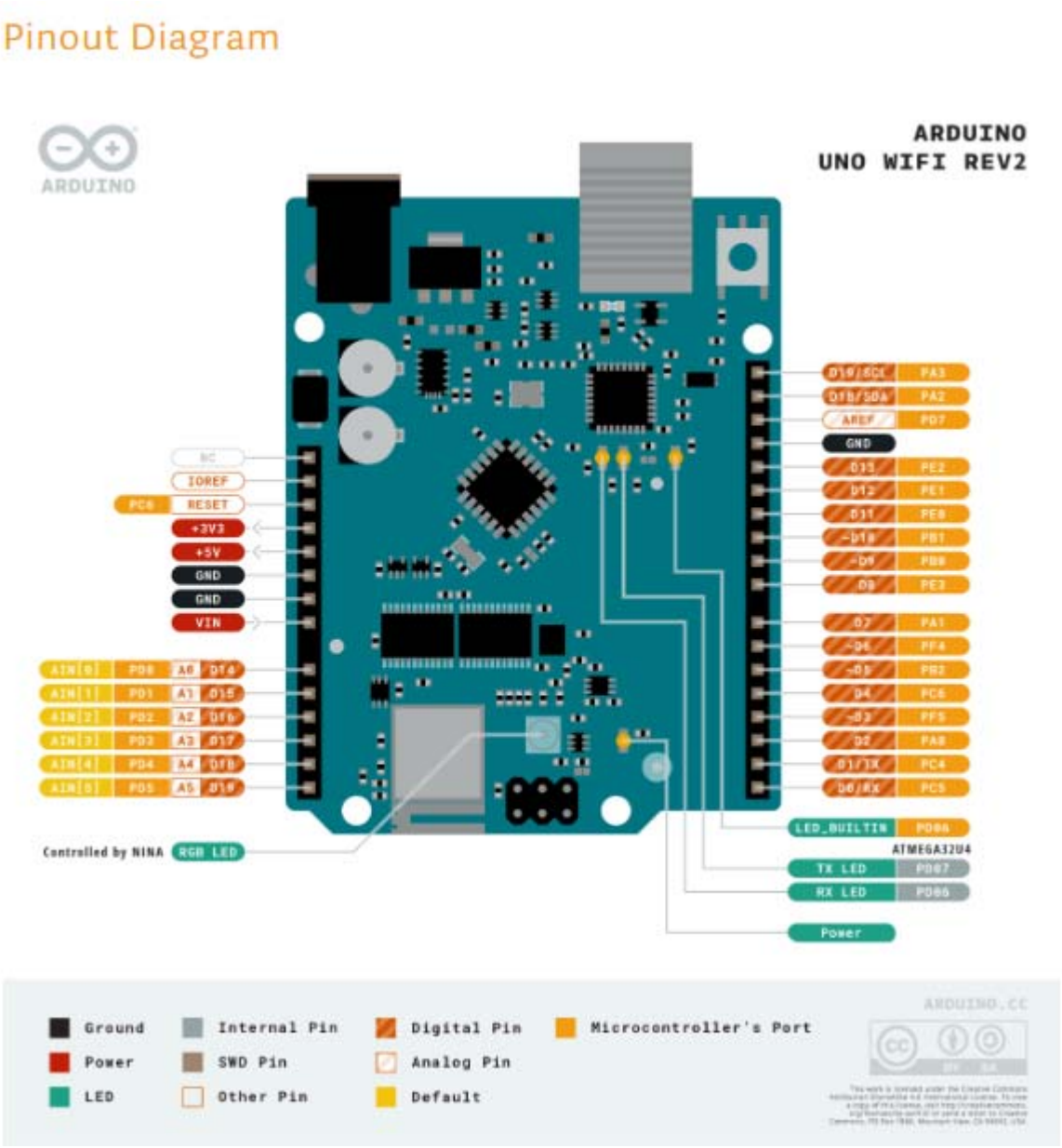
[18] https://www.seeedstudio.com/blog/2020/01/17/what-is-imu-sensor-overview-with-arduino-usage-guide/

Αναλυτικά τα τεχνικά χαρακτηριστικά του Arduino Uno Rev 2 είναι

Microcontroller	ATMEGA4809
Operating Voltage	5V
Input Voltage (recommended)	7 - 12V
Input Voltage (limit)	6 - 12V
Digital I/O Pins	14 — 5 Provide PWM Output
PWM Digital I/O Pins	5
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	48 KB (ATMEGA4809)
SRAM	6,144 B (ATMEGA4809)
EEPROM	256 Bytes (ATMEGA4809)
Clock Speed	16 MHz
LED_BUILTIN	25
Length	68.6 mm
Width	53.4 mm
Weight	25 g



Εικόνα 4.27 - σχεδιάγραμμα συνδεσμολογίας



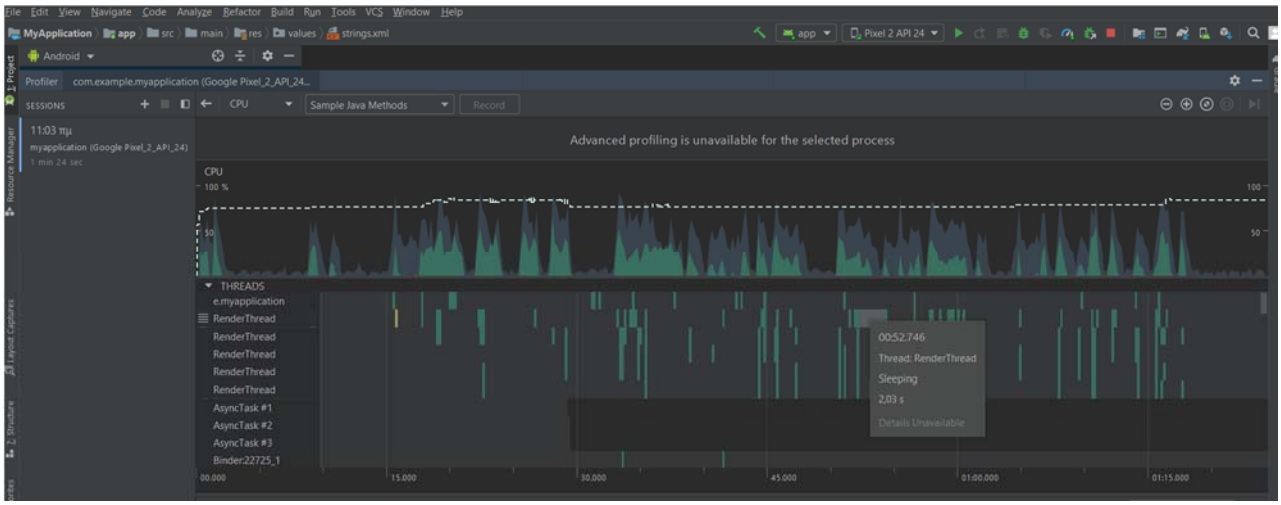
Εικόνα 4.28 - Διάγραμμα κεντρικής πλακέτας

4.7 Testing και benchmark εφαρμογής Android

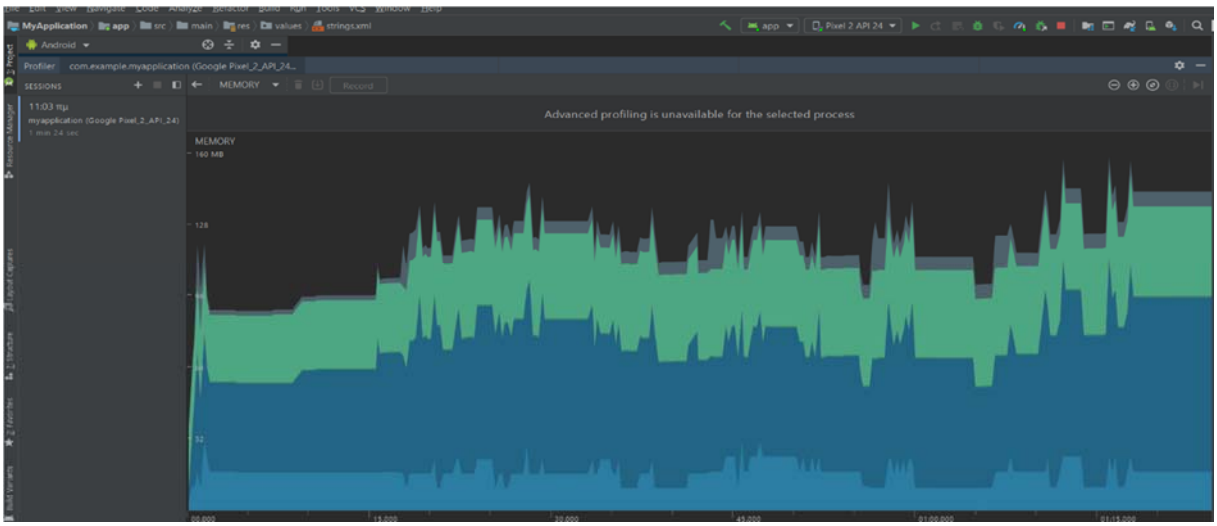
Οι δοκιμές και το testing διήρκησαν σε όλη την ανάπτυξη του λογισμικού και επιλύθηκαν όλα τα προβλήματα που παρουσιάστηκαν. Η τελική εφαρμογή του κινητού, με κατάληξη .apk είναι πολύ μικρή σε μέγεθος (~2,5 MB). Δεν επιβαρύνει καθόλου την συσκευή και η μόνη απαίτηση που υπάρχει είναι η συσκευή να έχει πρόσβαση στο διαδίκτυο. Το ότι δεν επιβαρύνει την συσκευή επιτυγχάνεται με το να έχουμε μεταφέρει όλη την επεξεργασία και την αυτοματοποίηση στον εξωτερικό διακομιστή, κρατώντας την mobile app μόνο για ενημέρωση και μικρή παράκαμψη του αυτοματισμού. Ακόμα και

όταν ζητείται η δημιουργία γραφημάτων που έχουν πολλά δεδομένα να αναπαραστήσουν, η διαδικασία εκτελείται στον διακομιστή.

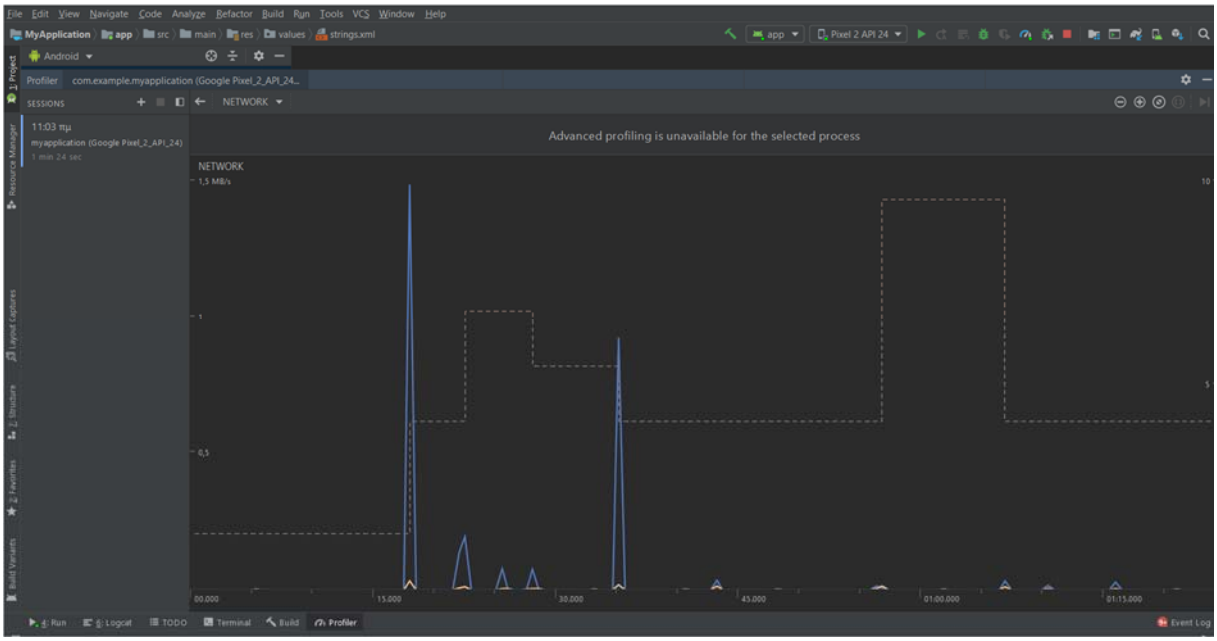
Η εφαρμογή ανάπτυξης λογισμικού Android studio IDE διαθέτει εξομοιώσεις με συσκευές διαφορετικών χαρακτηριστικών και πραγματοποιεί benchmark πάνω στην χρήση της εφαρμογής σε σχέση με τη χρήση του επεξεργαστική του τηλεφώνου (εικόνα 4.29), την χρήση του δικτύου (εικόνα 4.31), και της μνήμης που καταλαμβάνει (εικόνα 4.30). Σύμφωνα με τα αποτελέσματα των εξομοιώσεων που πραγματοποιήθηκαν φορτώνει σε λιγότερο από 2 δευτερόλεπτα, η χρήση του δικτύου είναι απειροελάχιστη αφού στέλνει μόνο καλέσματα συναρτήσεων, ενώ η χρήση του επεξεργαστή παραμένει σε μηδενικά επίπεδα αφού ούτε αυτός χρησιμοποιείται για κάποιον αυτόματο υπολογισμό.



Εικόνα 4.29 - Χρηση CPU



Εικόνα 4.30 - Χρήση μνήμης



Εικόνα 4.31 - Χρήση Δικτύου

Κεφάλαιο 5

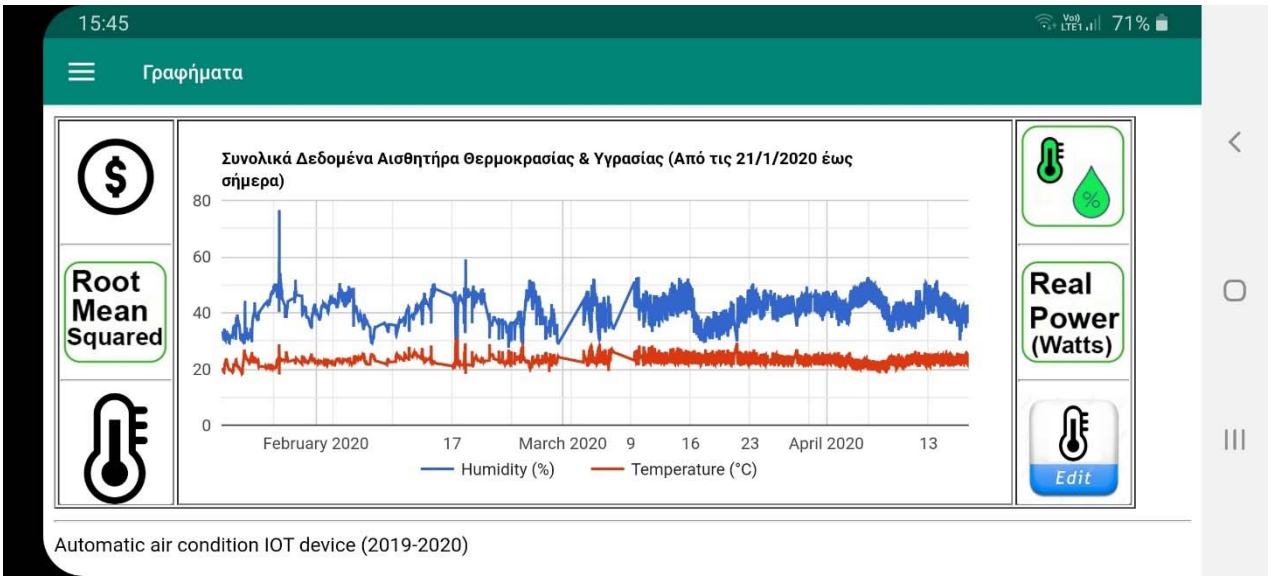
Αξιολόγηση και συμπεράσματα

5.1 Συλλογή και ανάλυση καταγεγραμμένων δεδομένων

5.1.1 Δεδομένα από αναπτυσσόμενη συσκευή

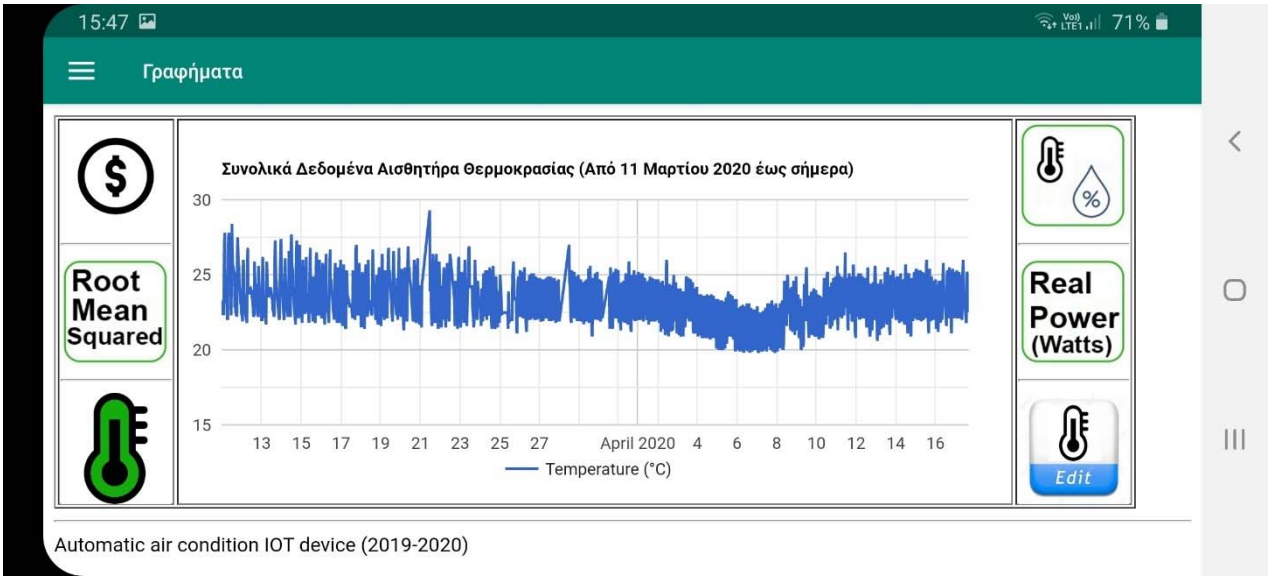
Τα δεδομένα που καταγράφηκαν από την IoT συσκευή δεν είναι συνεχή. Η καταγραφή τους ξεκίνησε τέλος Ιανουαρίου 2020. Με την ανάπτυξη του κώδικα, τις επιδιορθώσεις των σφαλμάτων και τις προσπάθειες για σταθεροποίηση του συστήματος ώστε να γίνει αξιόπιστο, υπήρχαν αρκετές χρονικές διακοπές καταγραφής ενώ σε ορισμένες περιπτώσεις πολύ πυκνές καταγραφές (ανά λεπτό ή 10 λεπτό). Από τις 11 Μαρτίου 2020 πραγματοποιείται συνεχής και σταθερή καταγραφή και αυτοματοποίηση ανά 30 λεπτά. Ακολουθούν τα στιγμιότυπα των γραφημάτων και παρουσιάζεται η σταθερότητα των μετρήσεων καθώς και η κατανάλωση ενέργειας με και χωρίς IoT συσκευή.

Στην εικόνα 5.1 παρουσιάζονται οι τιμές θερμοκρασίας και υγρασίας του χώρου. Παρατηρείται ότι οι μετρήσεις από τέλος Ιανουαρίου σταθεροποιούνται τελικά στις αρχές Μαρτίου με πυκνότερες και σταθερές μετρήσεις στα χρονικά διαστήματα.



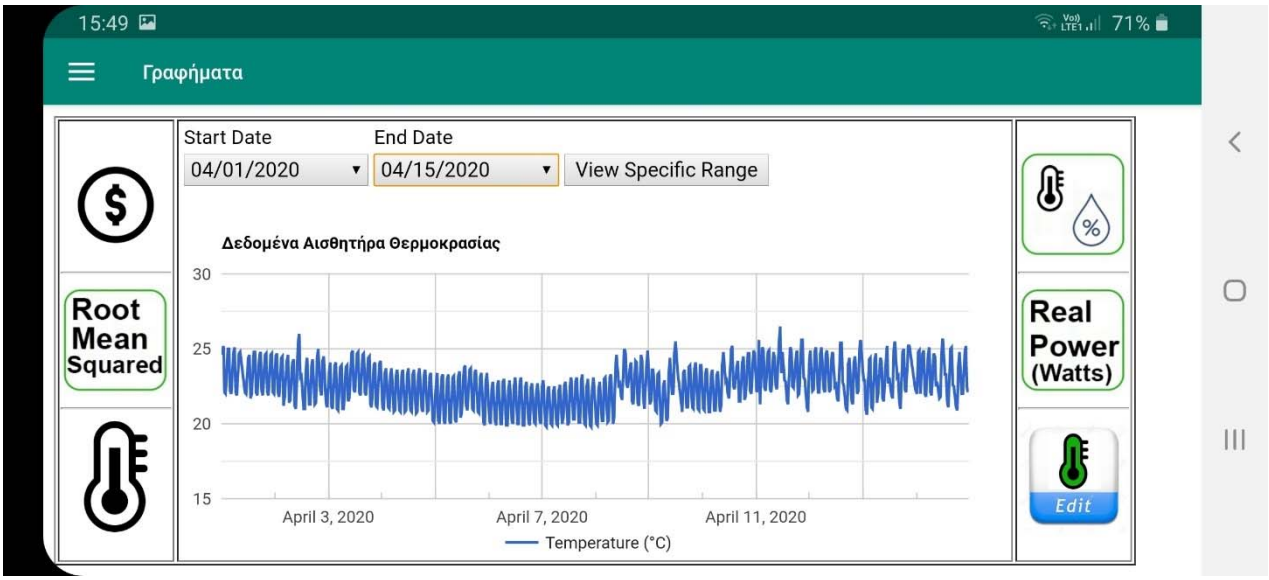
Εικόνα 5.1 - Συνολικά Δεδομένα Αισθητήρα Θερμοκρασίας & Υγρασίας (Από τις 21/1/2020 έως σήμερα)

Στην εικόνα 5.2 παρουσιάζονται μόνο οι τιμές τις θερμοκρασίες από 11 Μαρτίου έως σήμερα. Τα δεδομένα είναι αρκετά πυκνά και παρατηρείται η σταθερότητα στην διατήρηση της θερμοκρασίας στους 22 βαθμούς που λειτουργεί το αναπτυσσόμενο σύστημα.



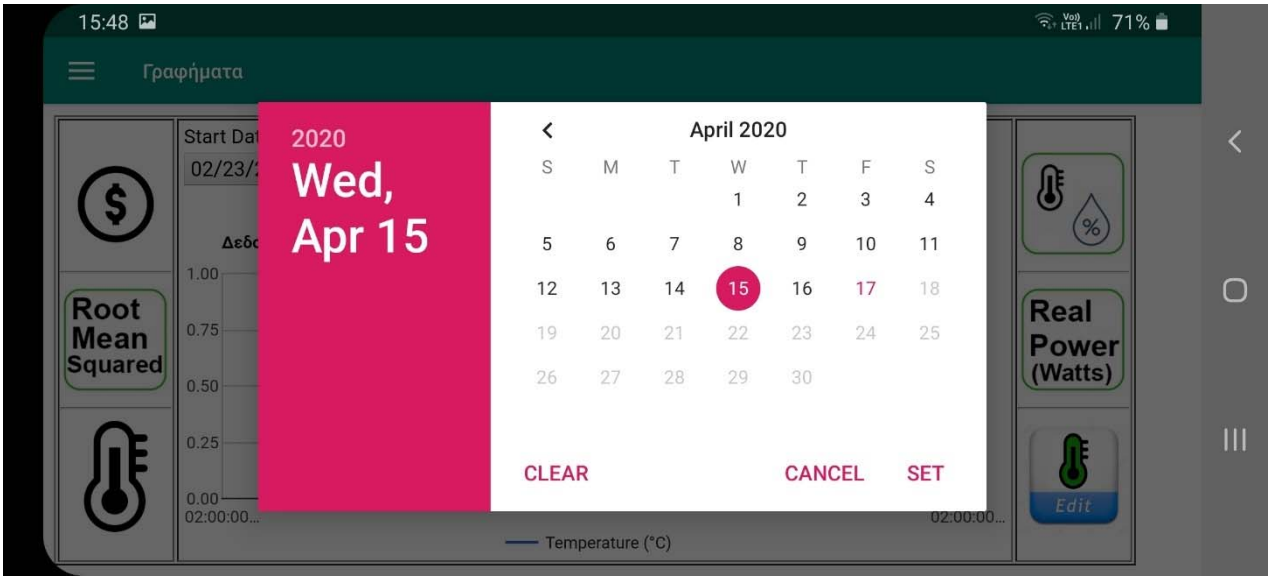
Εικόνα 5.2 - Συνολικά Δεδομένα Αισθητήρα Θερμοκρασίας (Από 11 Μαρτίου 2020 έως σήμερα)

Στην εικόνα 5.3 παρουσιάζονται οι τιμές τις θερμοκρασίας που ζητά από το σύστημα ο χρήστης. Το επιλεγμένο διάστημα είναι το πρώτο 15ημερο του Απριλίου. Το γράφημα είναι παρόμοιο με αυτό της εικόνας 5.2 αλλά πιο κατανοητό αφού παρουσιάζεται περισσότερη λεπτομέρεια και επικεντρωμένη πληροφόρηση.



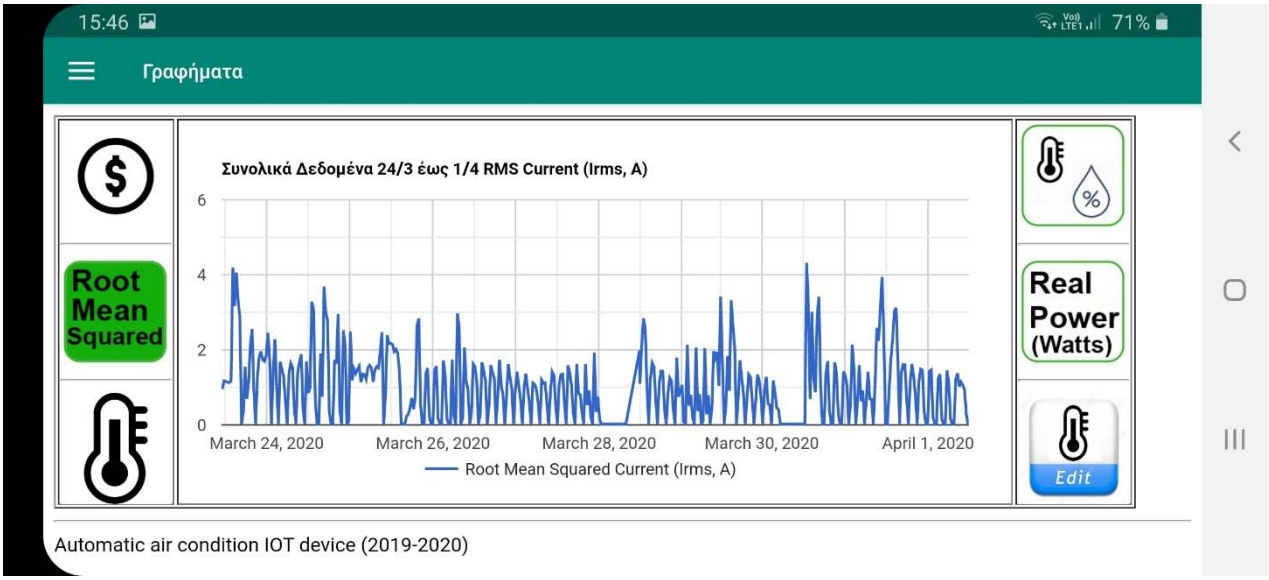
Εικόνα 5.3 - Δεδομένα Αισθητήρα Θερμοκρασίας (ορισμένο χρονικό διάστημα από τον χρήστη)

Στην εικόνα 5.4 ο χρήστης επιλέγει το χρονικό διάστημα για το οποίο θέλει να δει το σχετικό γράφημα. Δεν είναι δυνατή η επιλογή ημερομηνίας εκτός πραγματικού διαστήματος καταμέτρησης.



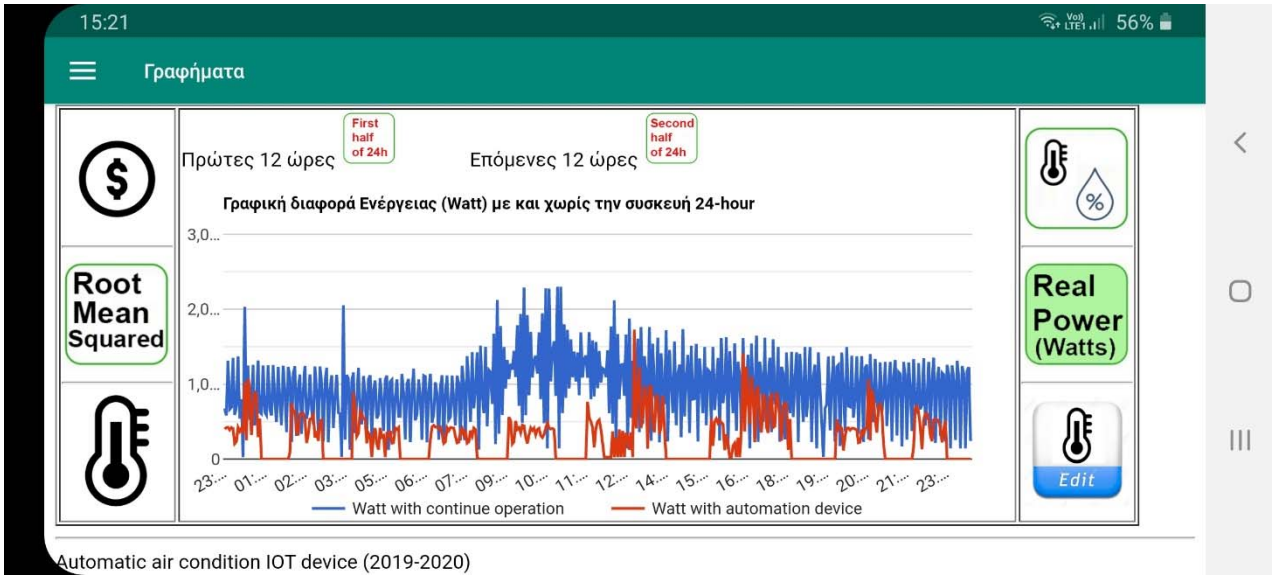
Εικόνα 5.4 – Επιλογή από χρήστη χρονικής διάρκειας δεδομένων για Αισθητήρα Θερμοκρασίας

Στην εικόνα 5.5 παρουσιάζονται τα Αμπέρ που καταναλώνονται σε διάστημα από 24/3/2020 έως 1/4/2020. Επιλέχτηκε μικρό χρονικό διάστημα για πιο ευανάγνωστη πληροφόρηση του χρήστη.



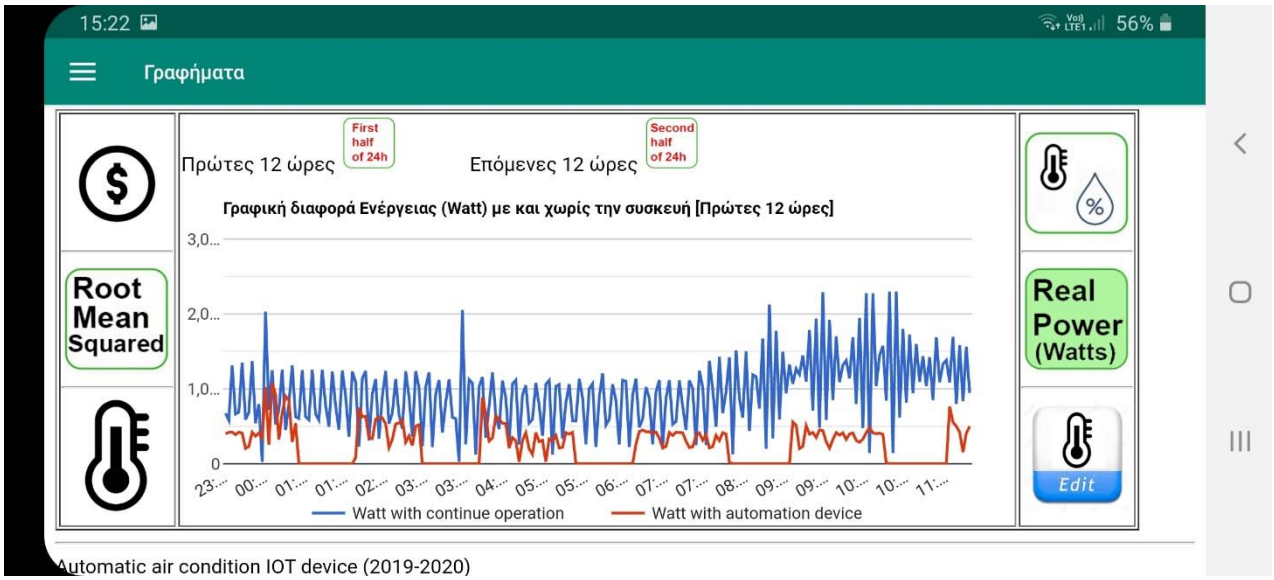
Εικόνα 5.5 -Συνολικά Δεδομένα από 24/3/2020 έως 1/4/2020 RMS Current (Irms, A)

Στην εικόνα 5.6 παρουσιάζεται η σύγκριση με και χωρίς την αναπτυσσόμενη συσκευή σε κατανάλωση ενέργειας Watt για διάστημα 24 ωρών.



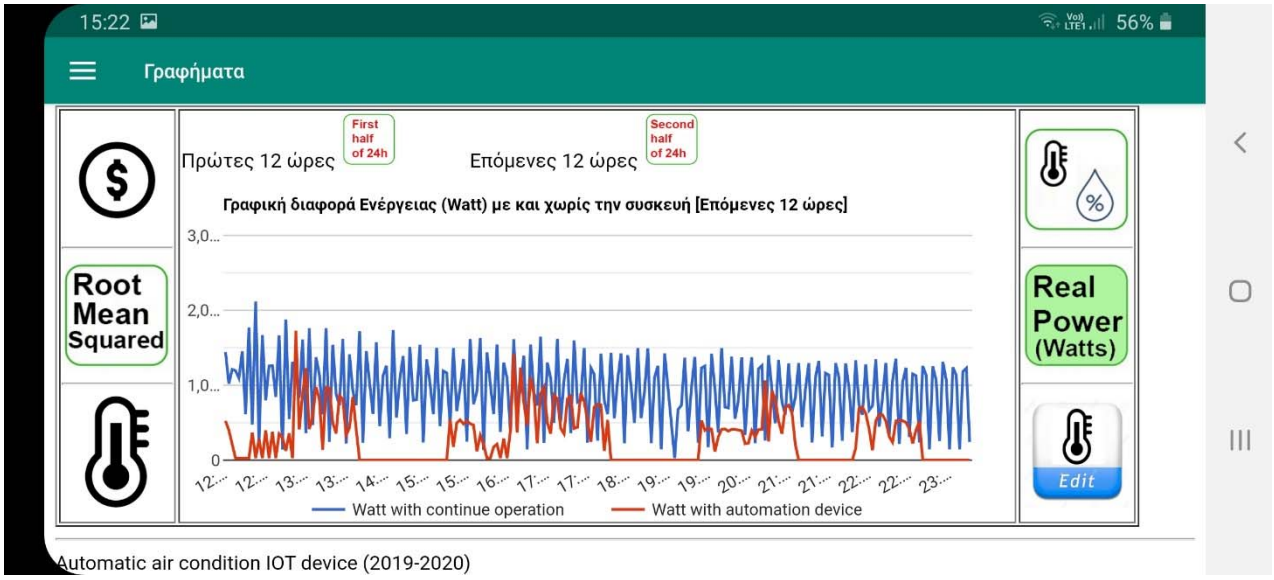
Εικόνα 5.6 - Γραφική διαφορά Ενέργειας (Watt) με και χωρίς την συσκευή σε ένα 24ωρο (24-hour)

Στην εικόνα 5.7 παρουσιάζεται η σύγκριση με και χωρίς την αναπτυσσόμενη συσκευή σε κατανάλωση ενέργειας Watt για τις πρώτες 12 ώρες, ώστε να μπορεί ο χρήστης να δει πιο συγκεκριμένες ώρες την κατανάλωση.



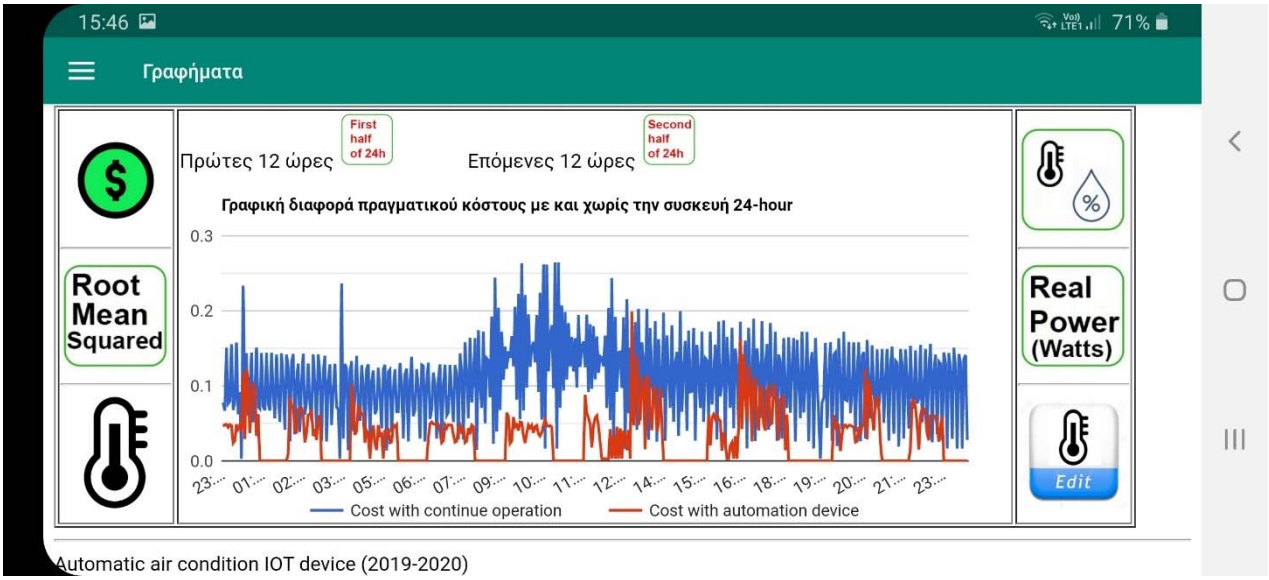
Εικόνα 5.7 -Γραφική διαφορά Ενέργειας (Watt) με και χωρίς την συσκευή τις πρώτες 12 ώρες του 24ωρου

Στην εικόνα 5.8 παρουσιάζεται η σύγκριση με και χωρίς την αναπτυσσόμενη συσκευή σε κατανάλωση ενέργειας Watt για τις επόμενες 12 ώρες.



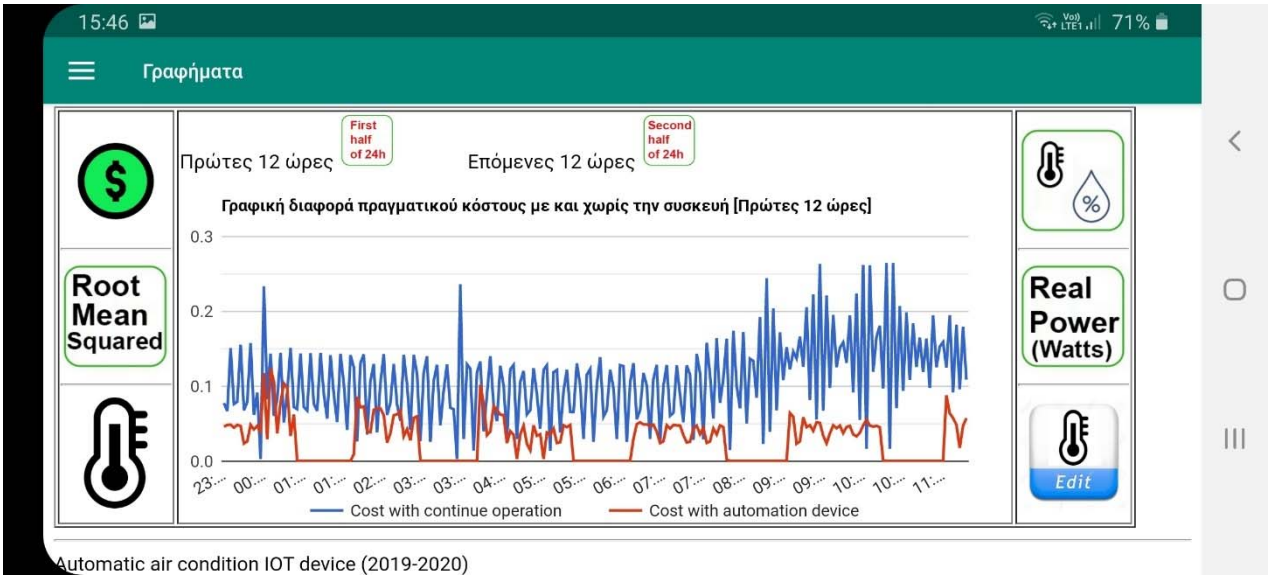
Εικόνα 5.8 - Γραφική διαφορά Ενέργειας (Watt) με και χωρίς την συσκευή το επόμενο 12ωρο

Στην εικόνα 5.9 παρουσιάζεται η σύγκριση με και χωρίς την αναπτυσσόμενη συσκευή πραγματικής κοστολόγησης ανά μέτρηση, για διάστημα 24 ωρών.

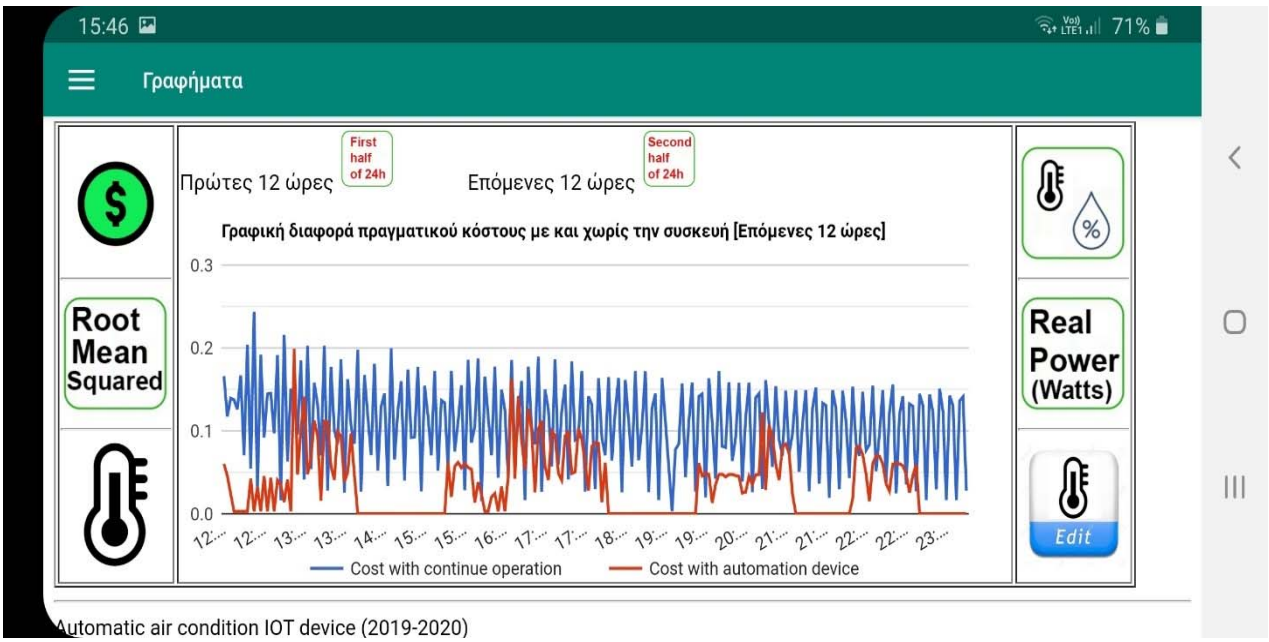


Εικόνα 5.9 - Γραφική διαφορά πραγματικού κόστους με και χωρίς την συσκευή 24-hour

Στην εικόνα 5.10 και 5,11 παρουσιάζεται η σύγκριση με και χωρίς την αναπτυσσόμενη συσκευή πραγματικής κοστολόγησης ανά μέτρηση, για διαστήματα 12 ωρών και όχι 24ωρών ώστε ο χρήστη να μπορεί να επιλέξει και να δει πιο αναλυτικά την ώρα για την οποία ενδιαφέρεται.



Εικόνα 5.10 - Γραφική διαφορά πραγματικού κόστους με και χωρίς την συσκευή [Πρώτες 12 ώρες]



Εικόνα 5.11 - Γραφική διαφορά πραγματικού κόστους με και χωρίς την συσκευή [Επόμενες 12 ώρες]

5.1.2 Δεδομένα από βοηθητικό εξοπλισμό

Τα δεδομένα που καταγράφηκαν είναι συνεχή με αποτέλεσμα να είναι πολύ μεγάλος ο όγκος της εξαγόμενης πληροφορίας και να μπορούν παρουσιαστούν όλα στην παρούσα εργασία. Επιλεχθήκαν ορισμένα παραδείγματα μετρήσεων που καταγράφηκαν από το smart energy meter και παρουσιάζονται μέσα από γραφικές παραστάσεις για συγκεκριμένα ημερολογιακά διαστήματα σε δύο διαφορετικές λειτουργικές φάσεις μέτρησης με και χωρίς IoT συσκευή. Λόγω της σχεδόν συνεχούς λειτουργίας της συσκευής IoT δεν υπάρχουν πολλά συνεχόμενα χρονικά διαστήματα με εξαγωγή δεδομένων χωρίς χρήση της IoT συσκευής.

Για να γίνει μια σύγκριση κατανάλωσης με και χωρίς IoT συσκευή, επιλέχθηκαν δύο ίσα χρονικά διαστήματα καταγραφής.

Το πρώτο χρονικό διάστημα (7-8 Μαρτίου) επιλέχτηκε να αποσυνδεθεί η IoT συσκευή ώστε να καταγραφεί η κατανάλωση χωρίς αυτοματισμούς. Το συγκεκριμένο διάστημα χρησιμοποιείται στα γραφήματα για παρουσίαση κόστους (εικόνες 5.12, 5.13, 5.14) και κατανάλωσης (εικόνα 5.18) με συνολικές και μέσες τιμές ανά 24ωρο και 48ωρο.

Το δεύτερο χρονικό διάστημα (24-25 Μαρτίου) καταγράφηκε η κατανάλωση με χρήση της IoT συσκευής δηλαδή με αυτοματισμούς. Το συγκεκριμένο διάστημα χρησιμοποιείται στα γραφήματα για παρουσίαση κόστους (εικόνες 5.15, 5.16, 5.17) και κατανάλωσης (εικόνα 5.19) με συνολικές και μέσες τιμές ανά 24ωρο και 48ωρο.

Τα ίδια χρονικά διαστήματα 48 ωρών χρησιμοποιούνται επίσης για την σύγκριση των υπόλοιπων μετρήσεων (εικόνες 5.20 έως 5.27). Στα γραφήματα με τις συχνότητες του ρεύματος δεν παρατηρούμε διαφορές των δύο γραφημάτων και αυτό γιατί το συγκεκριμένο στατιστικό χρησιμοποιείται για παρακολούθηση σε περιπτώσεις που το ρεύμα δεν είναι σταθερό στο κτίριο. Παρόλα αυτά παρουσιάζονται τα δυο γραφήματα ως απόδειξη της σταθερότητας του ρεύματος.

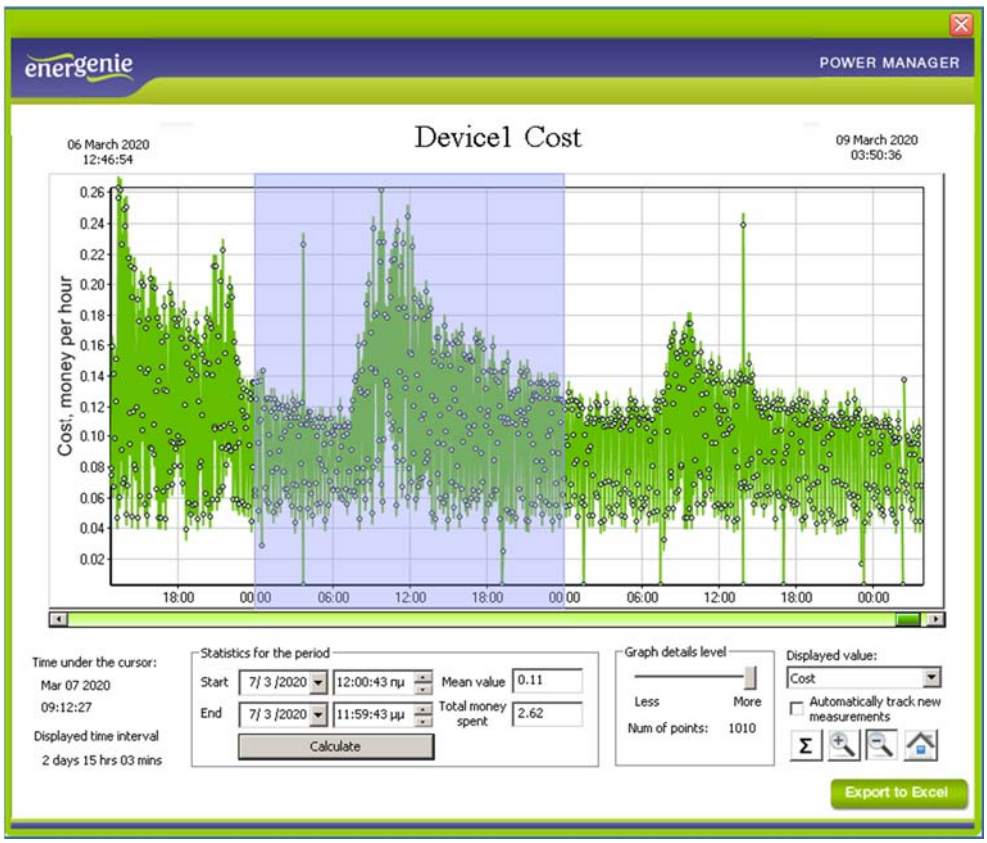
Ομοίως με τα γραφήματα των RMS Voltage όπου και πάλι δεν παρατηρείται διαφορά μεταξύ των δυο γραφημάτων για τον ίδιο με τον παραπάνω λόγο.

Συγκεντρωτικά οι τιμές φαίνονται στους παρακάτω πίνακες

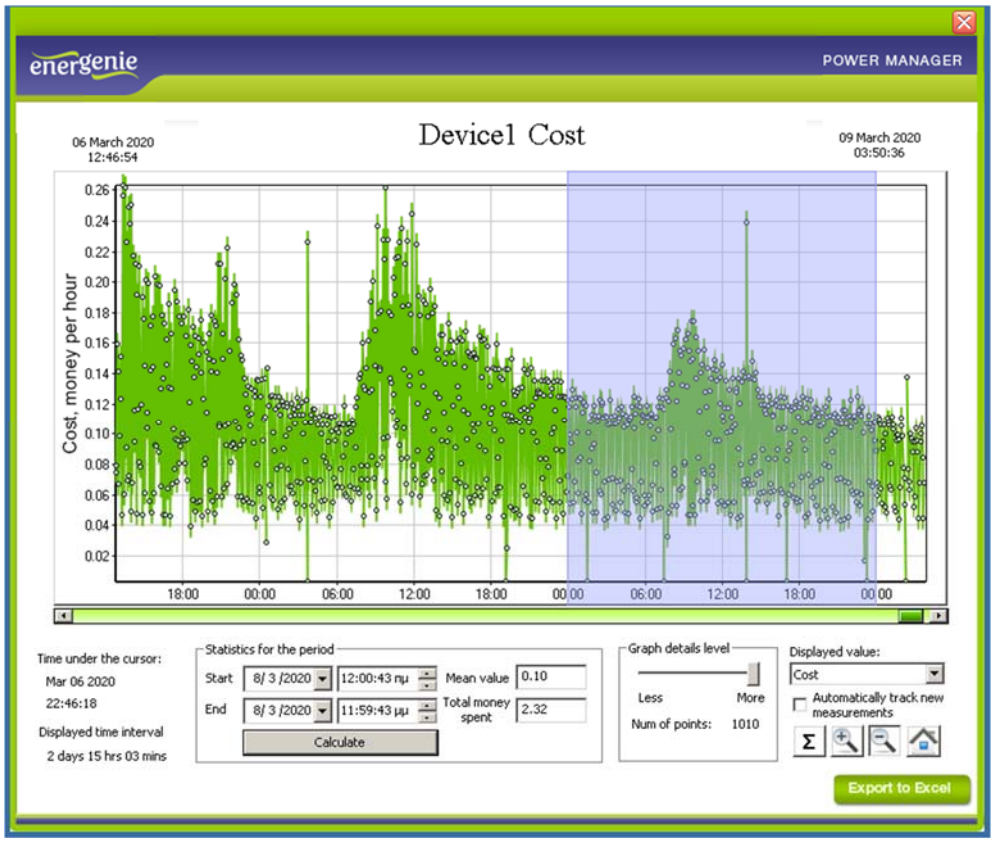
Μετρήσεις τιμών κόστους χωρίς την IoT συσκευή	Μέση τιμή	Συνολική τιμή
Κόστος 24h (7-3-2020) χωρίς την IoT συσκευή	0,11€	2,62€
Κόστος 24h (8-3-2020) χωρίς την IoT συσκευή	0,10€	2,32€
Κόστος 48h (7-3-2020 και 8-3-2020) χωρίς την IoT συσκευή	0,10€	4,95€

Μετρήσεις τιμών κόστους με την IoT συσκευή	Μέση τιμή	Συνολική τιμή
Κόστος 24h (24-3-2020) με την IoT συσκευή	0,03€	0,74€
Κόστος 24h (25-3-2020) με την IoT συσκευή	0,03€	0,71€
Κόστος 48h (24-3-2020 και 25-3-2020) με την IoT συσκευή	0,03€	1,45€

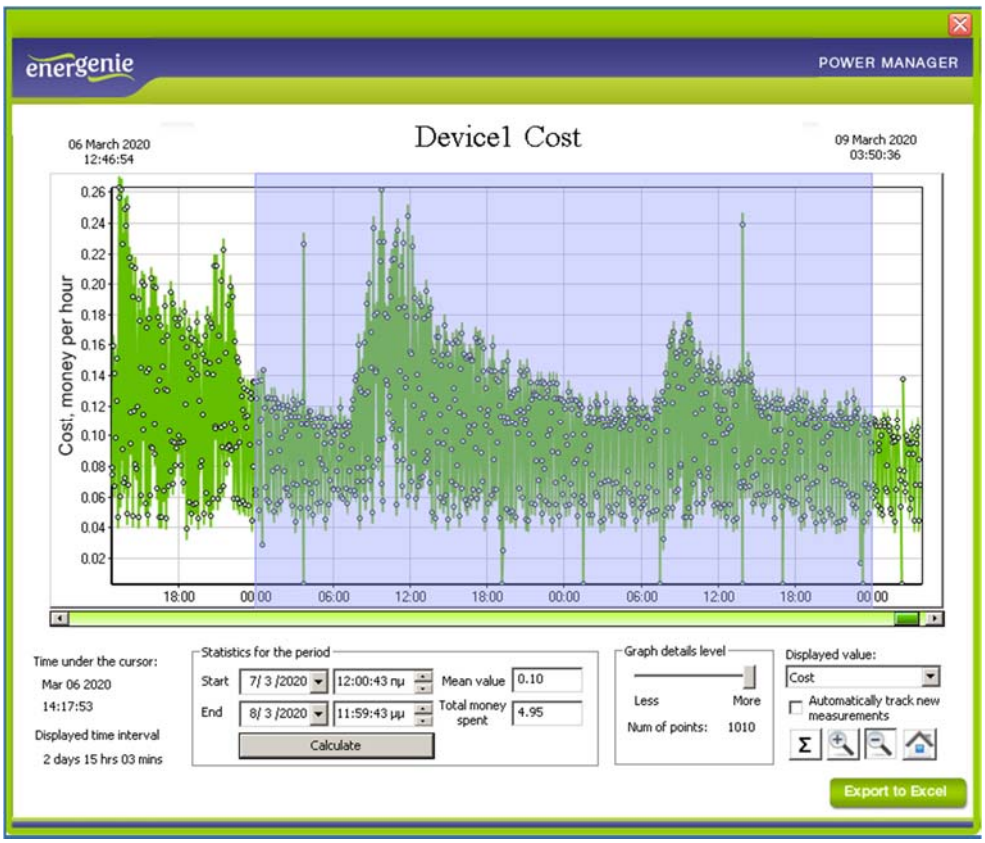
Μετρήσεις τιμών ενέργειας	Μέση τιμή	Συνολική τιμή
48h (7-3-2020 και 8-3-2020) χωρίς την IoT συσκευή	896,39	43011,93
48h (24-3-2020 και 258-3-2020) την IoT συσκευή	263,01	12620,29



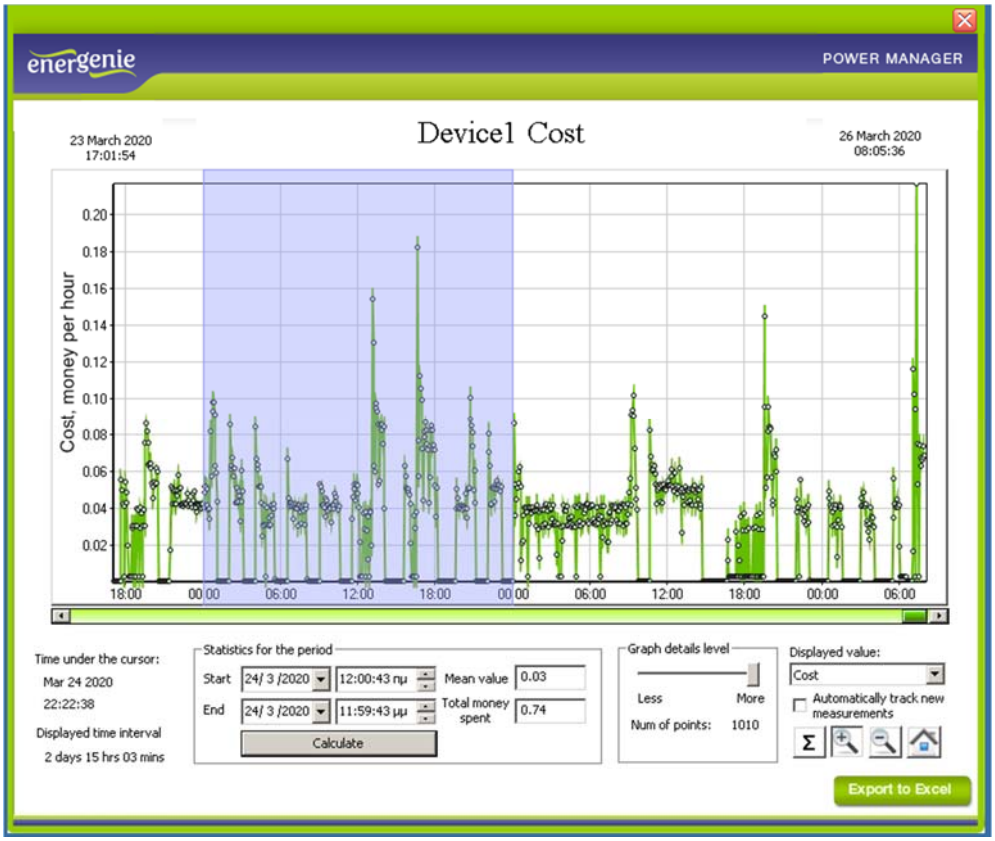
Εικόνα 5.12 – Κόστος με συνεχή λειτουργία 24h (7-3-2020) (χωρίς την IoT συσκευή)



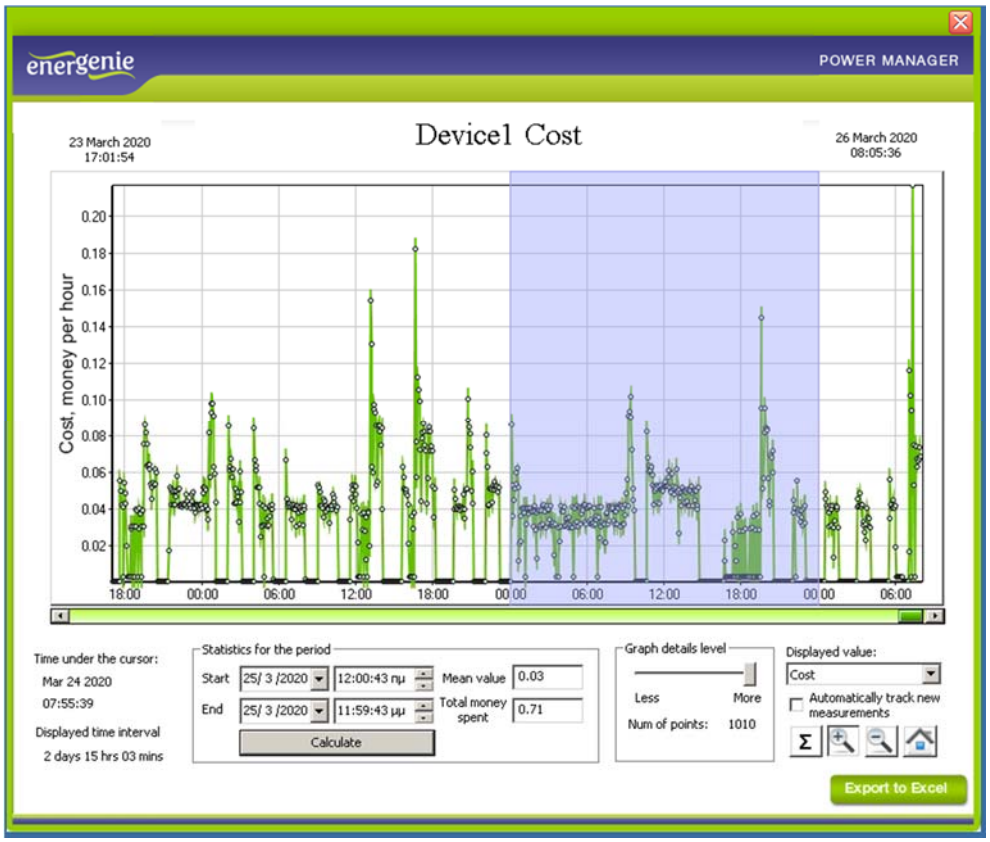
Εικόνα 5.13 - Κόστος με συνεχή λειτουργία 24h (8-3-2020) (χωρίς την IoT συσκευή)



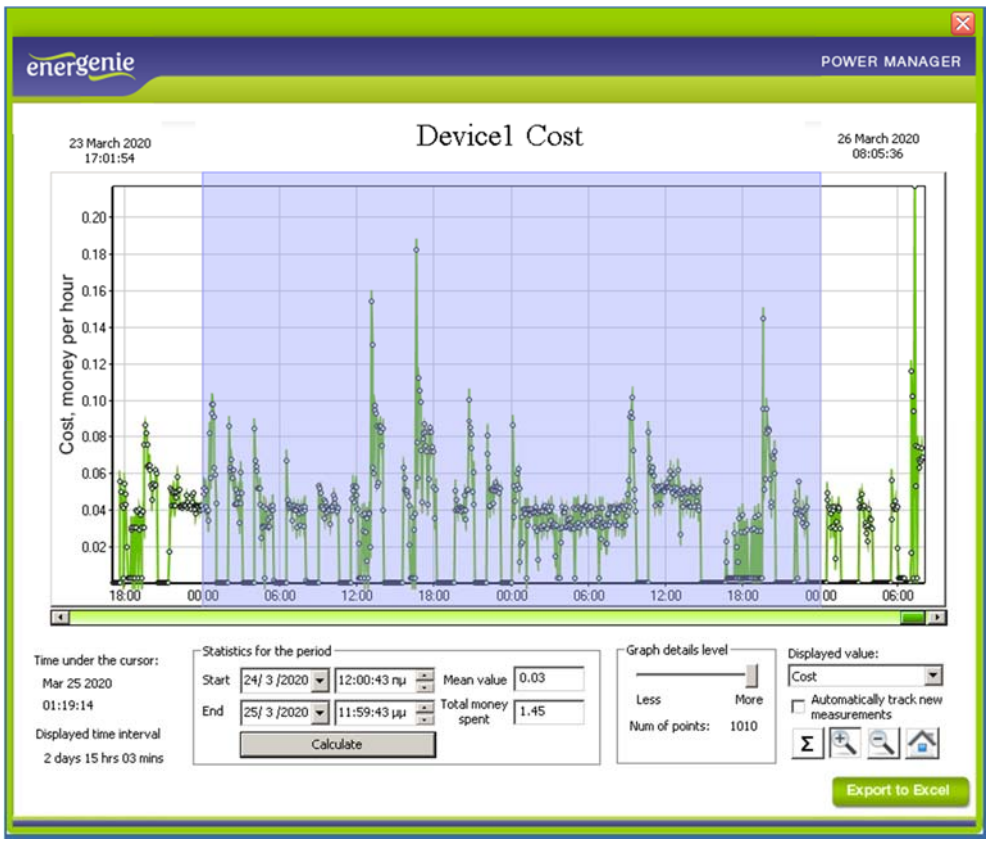
Εικόνα 5.14 - Κόστος με συνεχή λειτουργία 48h (7-3-2020 και 8-3-2020) (χωρίς την IoT συσκευή)



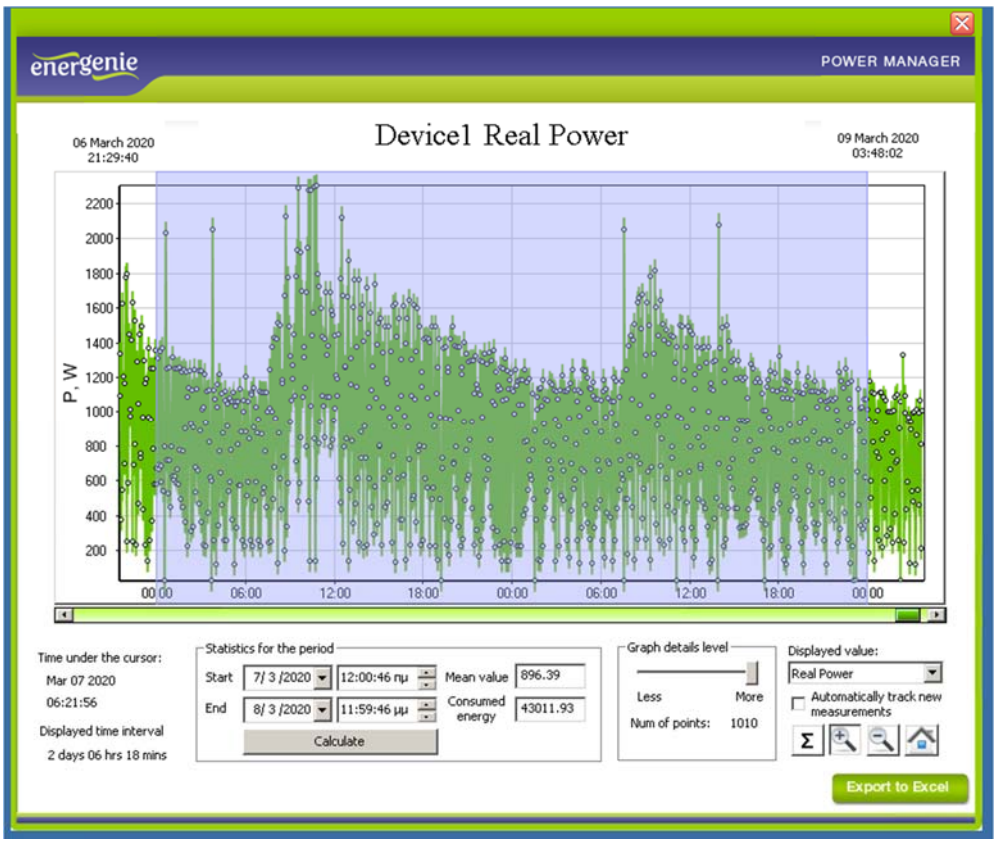
Εικόνα 5.15 - Κόστος με αυτοματοποιημένη λειτουργία 24h (24-3-2020)



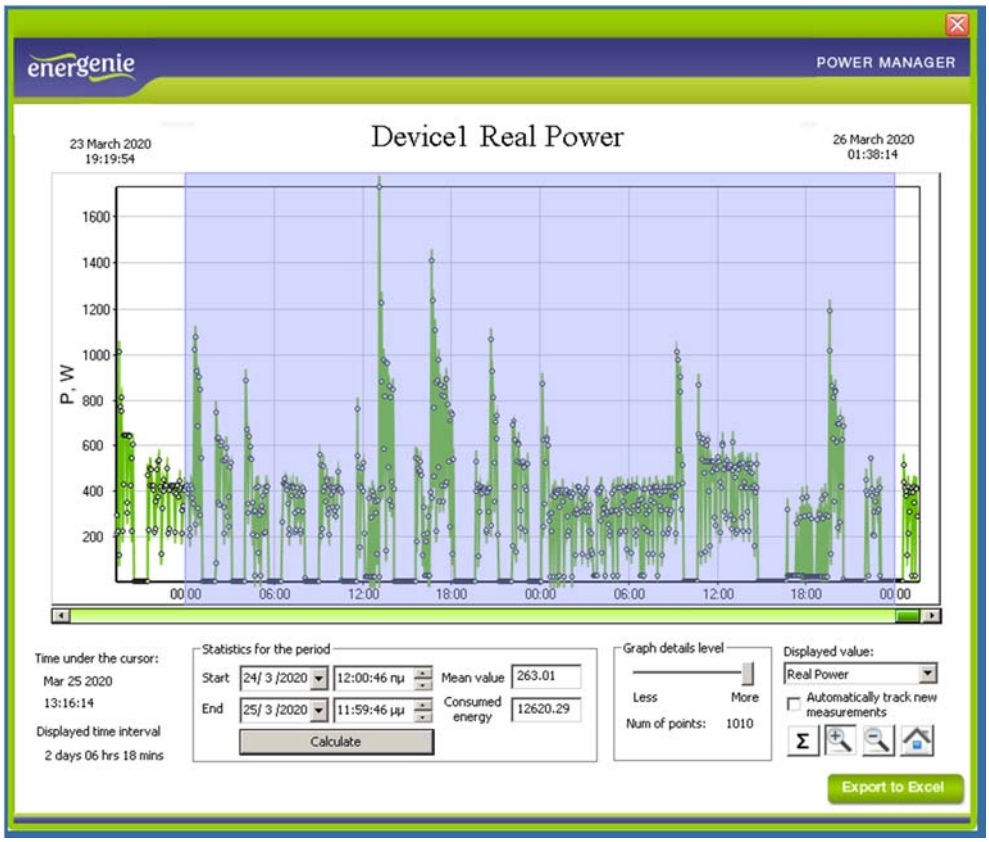
Εικόνα 5.16 - Κόστος με αυτοματοποιημένη λειτουργία 24h (25-3-2020)



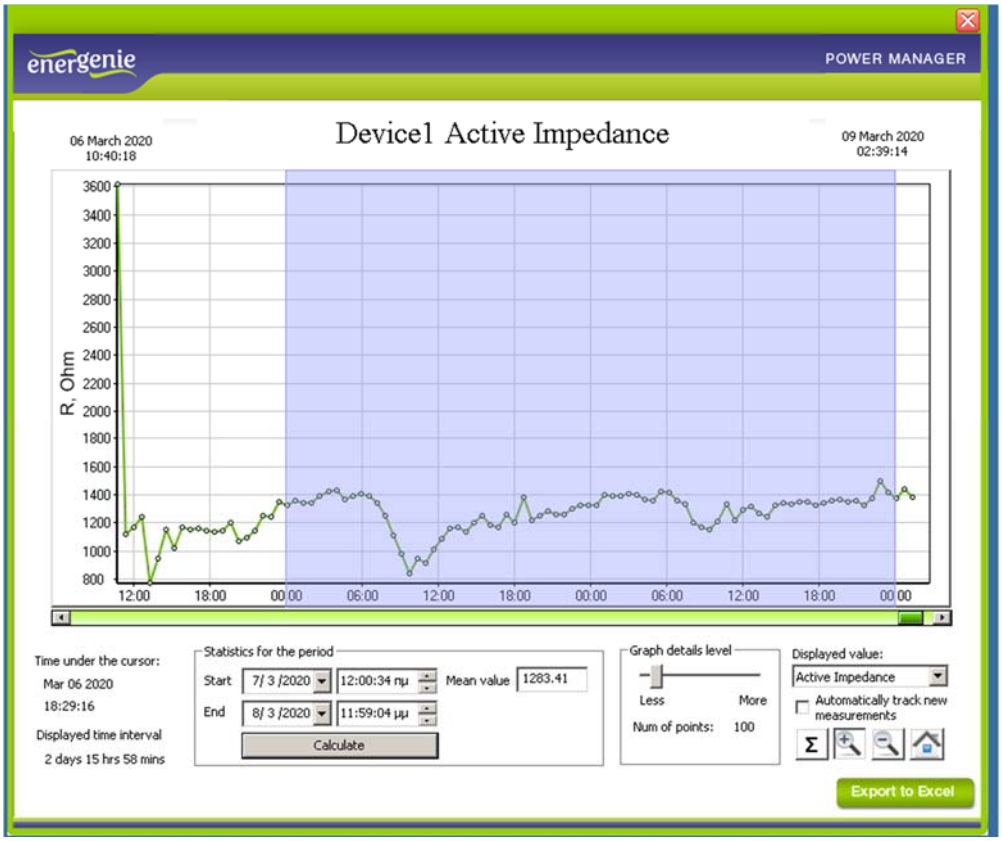
Εικόνα 5.17 - Κόστος με αυτοματοποιημένη λειτουργία 48h (24-3-2020 και 25-3-2020)



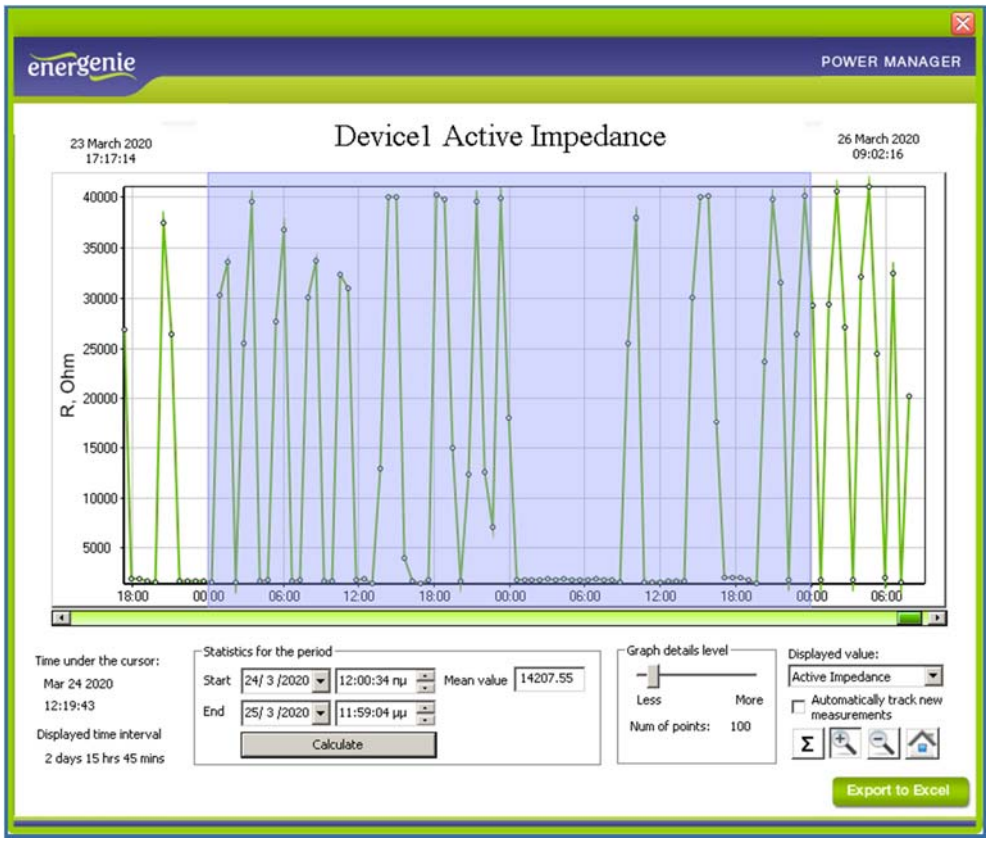
Εικόνα 5.18 – Real Power με συνεχές λειτουργία 48h (7-3-2020 και 8-3-2020) (χωρίς την IoT συσκευή)



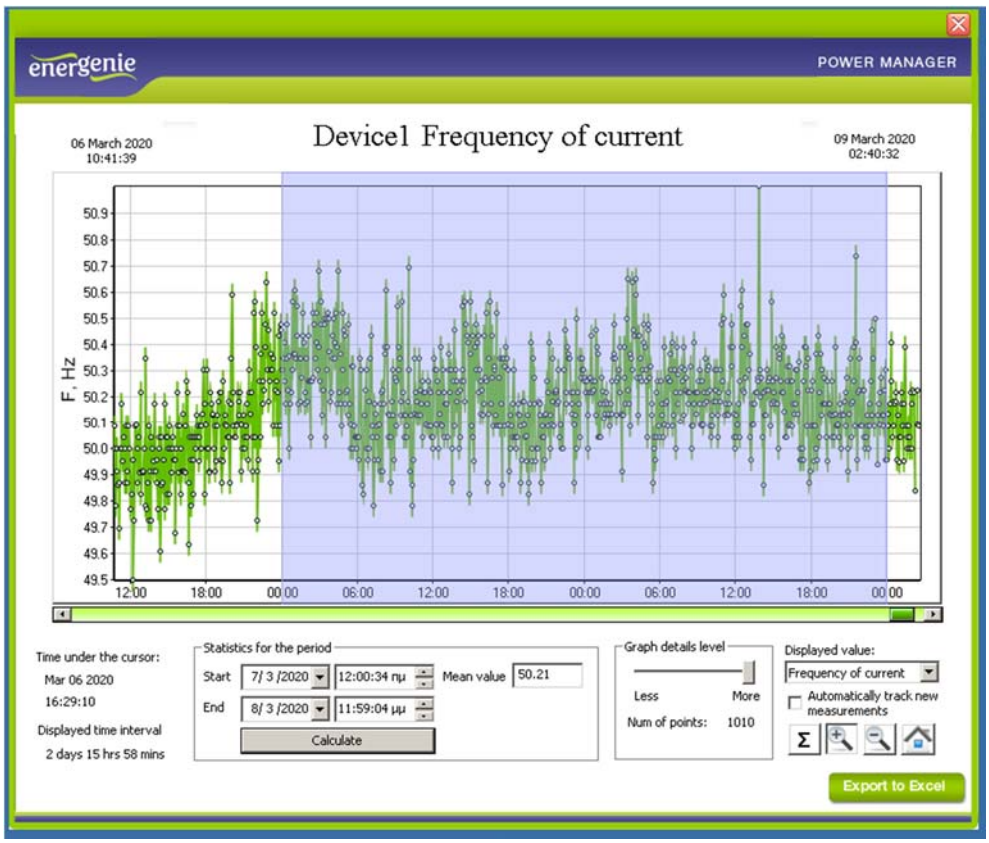
Εικόνα 5.19 – Real Power με αυτοματοποιημένη λειτουργία 48h (24-3-2020 και 25-3-2020)



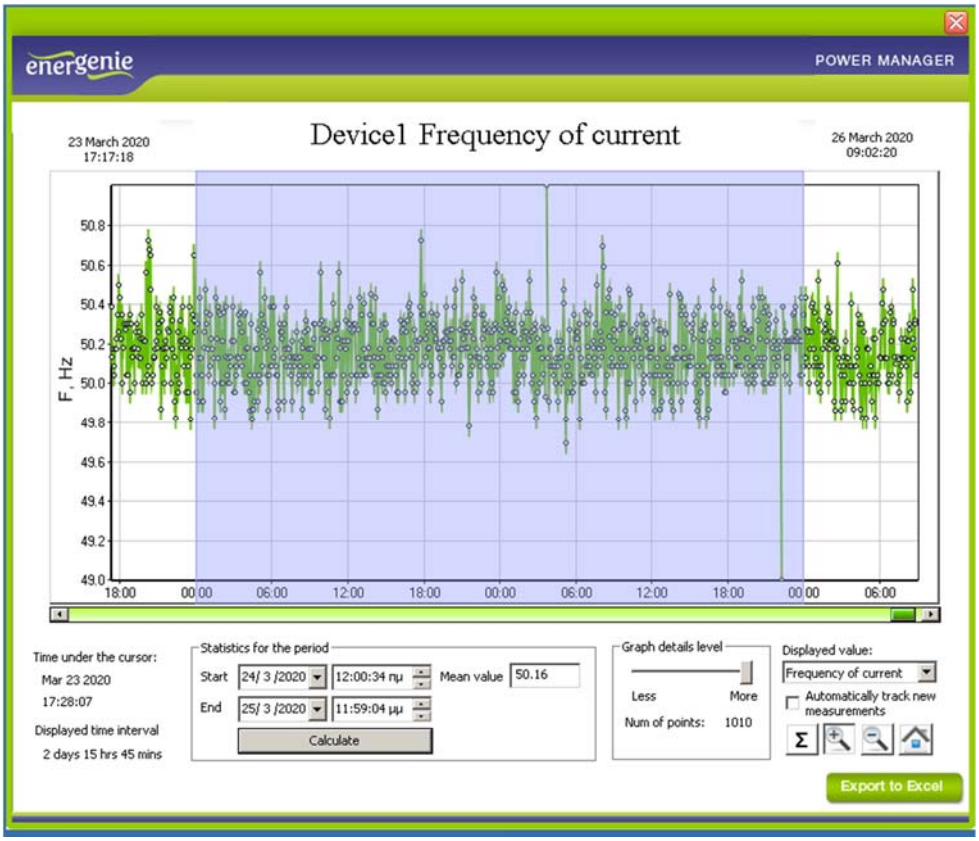
Εικόνα 5.20 – Ενεργή αντίσταση με συνεχές λειτουργία 48h (7-3-2020 και 8-3-2020) (χωρίς την IoT συσκευή)



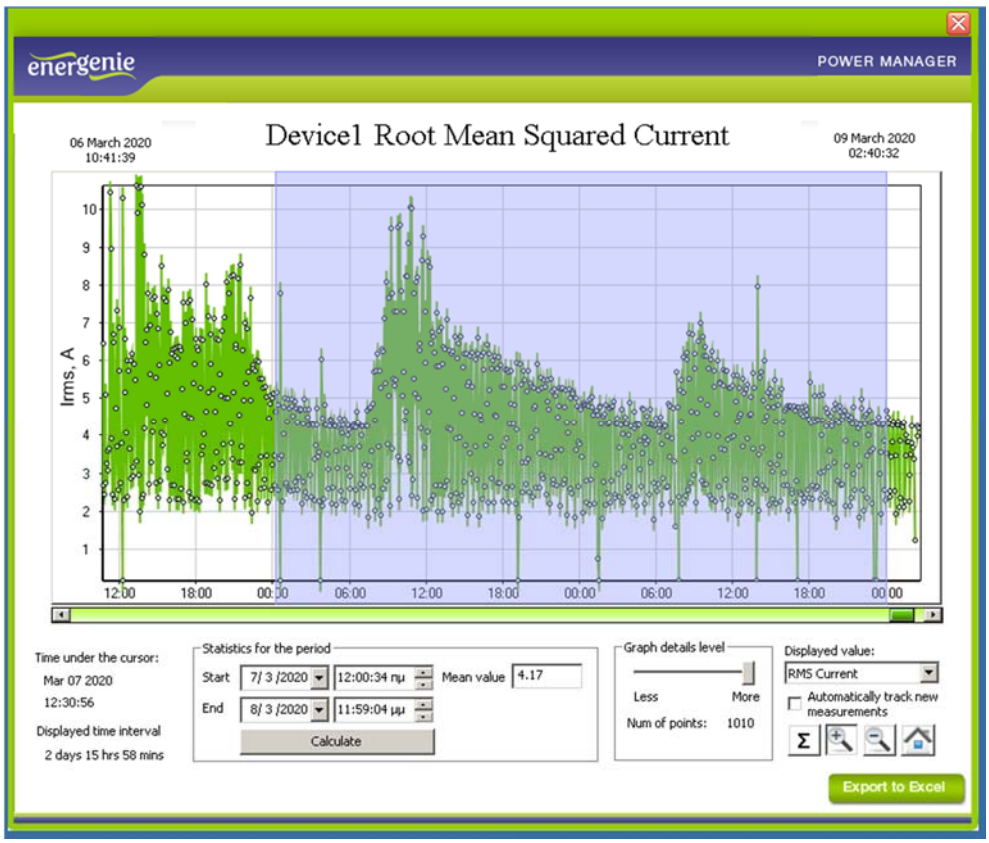
Εικόνα 5.21 - Ενεργή αντίσταση με αυτοματοποιημένη λειτουργία 48h (24-3-2020 και 25-3-2020)



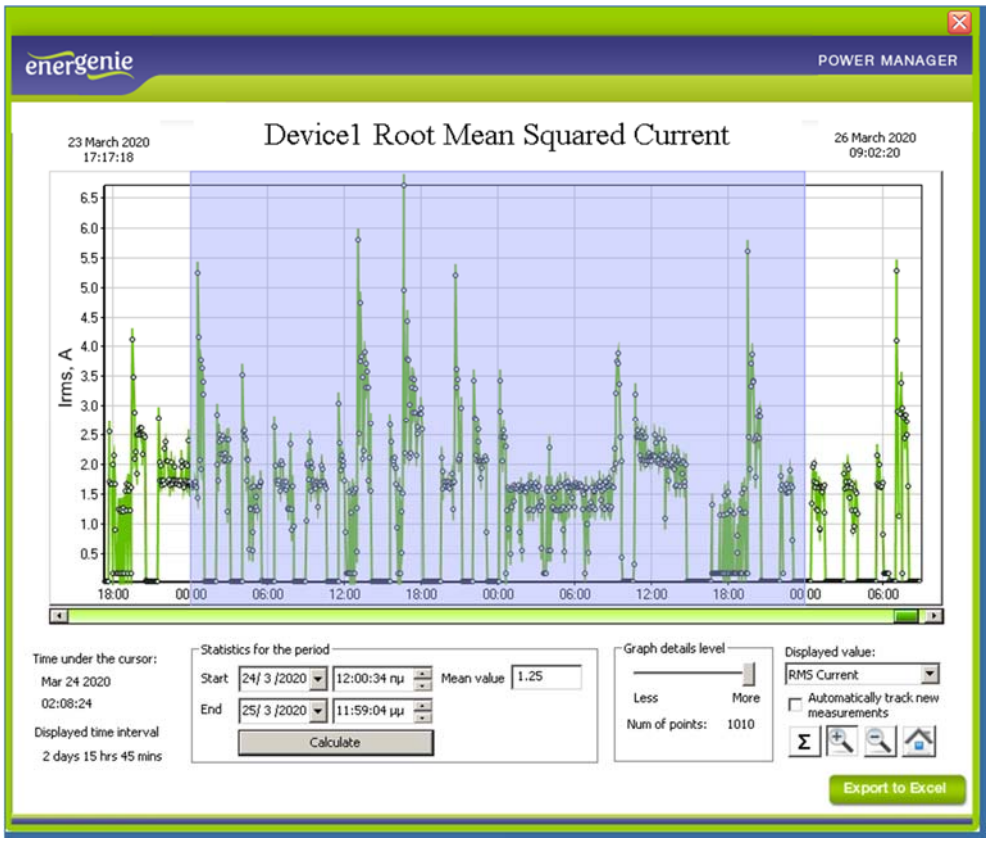
Εικόνα 5.22 – Συχνότητα ρεύματος με συνεχές λειτουργία 48h (7-3-2020 και 8-3-2020) (χωρίς την IoT συσκευή)



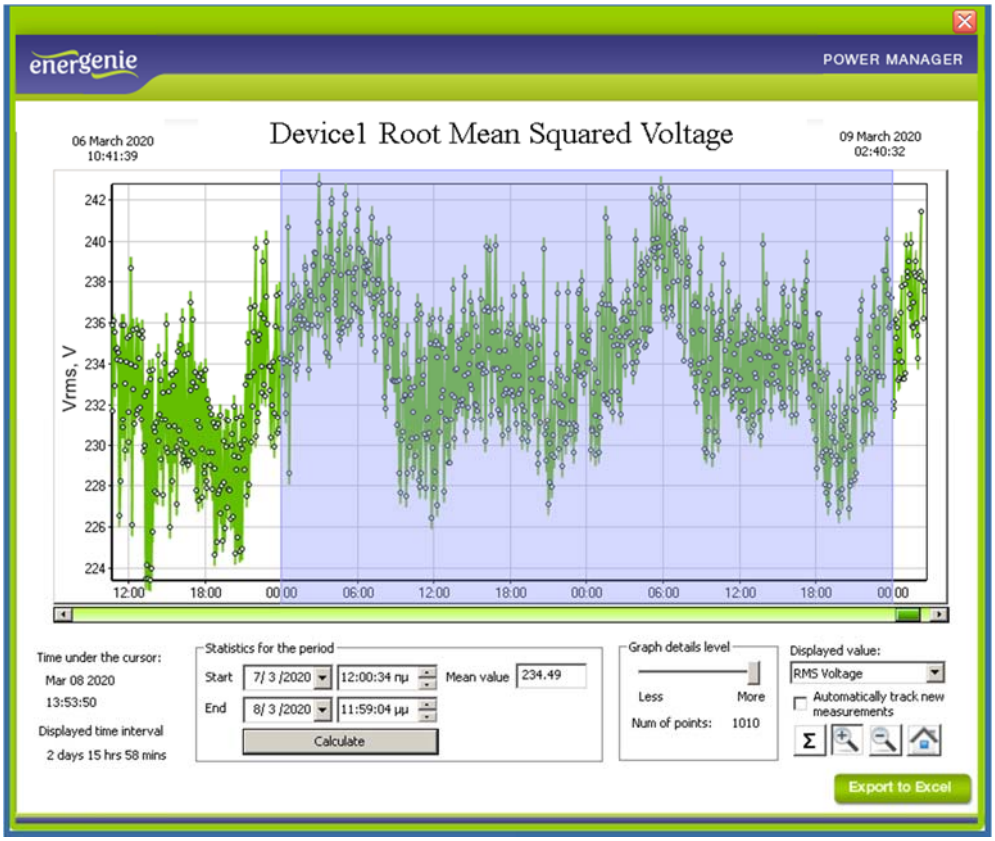
Εικόνα 5.23 - Συχνότητα ρεύματος με αυτοματοποιημένη λειτουργία 48h (24-3-2020 και 25-3-2020)



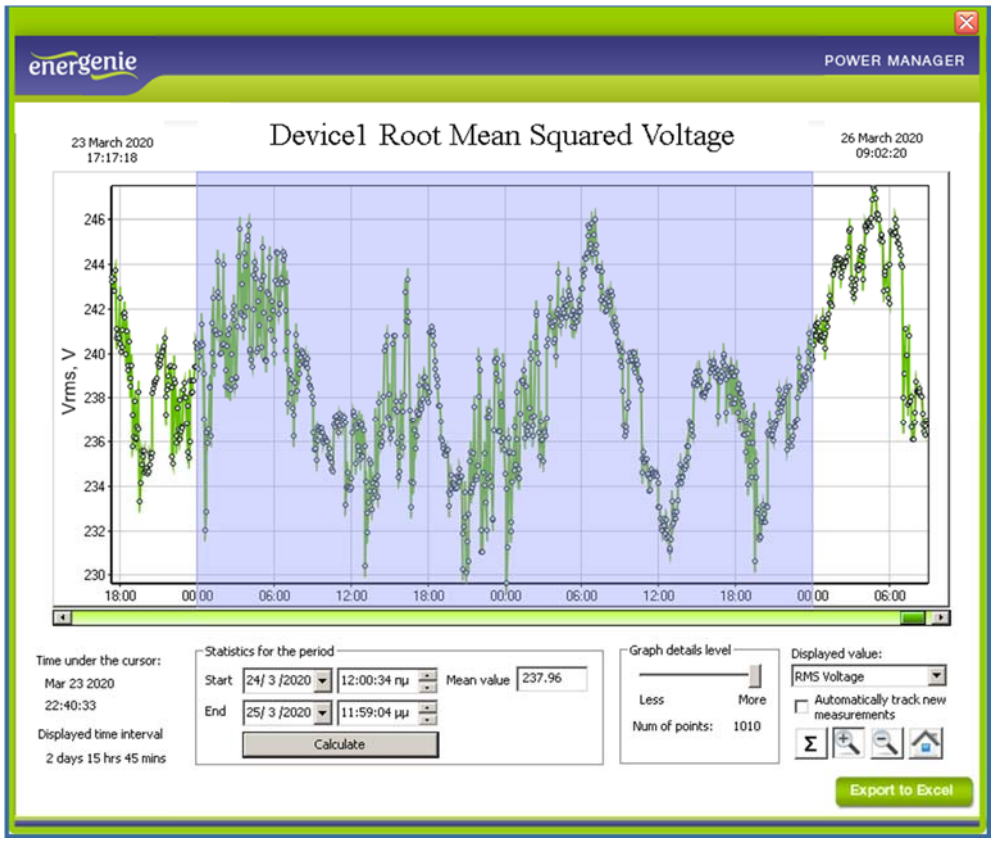
Εικόνα 5.24 - RMS Current με συνεχές λειτουργία 48h (7-3-2020 και 8-3-2020) (χωρίς την IoT συσκευή)



Εικόνα 5.25 - RMS Current με αυτοματοποιημένη λειτουργία 48h (24-3-2020 και 25-3-2020)



Εικόνα 5.26 - RMS Voltage με συνεχές λειτουργία 48h (7-3-2020 και 8-3-2020) (χωρίς την IoT συσκευή)



Εικόνα 5.27 - RMS Voltage με αυτοματοποιημένη λειτουργία 48h (24-3-2020 και 25-3-2020)

5.2 Συμπεράσματα

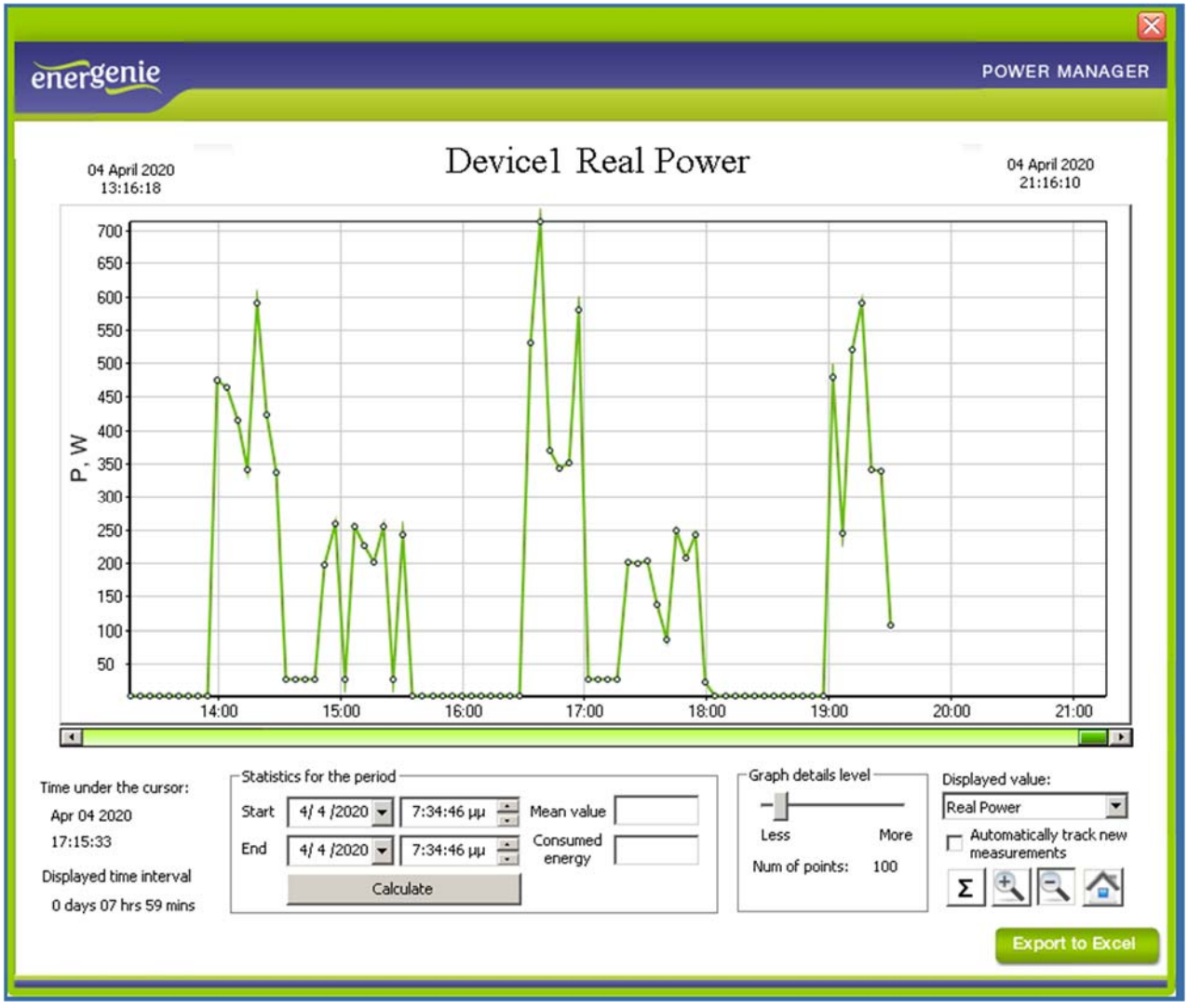
Κατά την υλοποίηση, η android εφαρμογή σχεδιάστηκε να επικοινωνεί και να ελέγχει μια συσκευή IoT. Μπορεί όμως πολύ εύκολα να υποστηρίξει και μεγαλύτερο αριθμό συσκευών IoT, αφού η κάθε μία συνδέεται με ξεχωριστή ip στο δίκτυο. Κατ' επέκταση, θα ήταν εφικτό να ελέγχονται διαφορετικά δωμάτια σε διαφορετικούς ορόφους ενός κτιρίου και να πραγματοποιείται επιπλέον εξοικονόμηση ενέργειας συλλογικά στο κτίριο.

Όπως φαίνεται στα γραφήματα της προηγουμένης ενότητας, η αναπτυσσόμενη συσκευή με την αυτοματοποιημένη λειτουργία κατάφερε για τον ίδιο στόχο θερμοκρασίας δωματίου να έχει κατανάλωση μικρότερη κατά 2/3 από την κατανάλωση που θα απαιτούνταν χωρίς την αυτοματοποιημένη συσκευή. Τα συγκρίσιμα γραφήματα αφορούν δυο διαφορετικά 48ώρα χρήσης με την συσκευή και χωρίς την συσκευή και στα οποία παρουσιάζεται η κατανάλωση σε Watt και το κόστος αυτής σε ευρώ με οριζόμενη τιμή kWh 0.115 ευρώ.
Με τους ιδίους στόχους θερμοκρασίας στο δωμάτιο (22 βαθμούς κελσίου) **οι μετρήσεις κόστους σε διάστημα 48ωρών (7/3/2020 – 8/3/2020) συνεχούς λειτουργίας (χωρίς IoT συσκευή), είναι συνολικά 4,95 ευρώ. Ενώ για μετρήσεις κόστους σε διάστημα 48ωρών (24/3/2020 – 25/3/2020) αυτοματοποιημένης λειτουργίας, είναι συνολικά 1,45 ευρώ με μείωση ποσοστού στα 70,7%.**
Τα ίδια ποσοστά εξοικονόμησης βλέπουμε και στις μετρήσεις των kWh που καταναλώθηκα στο 48ωρο διάστημα. **Με μετρήσεις Watt σε διάστημα 48ωρών (7/3/2020 – 8/3/2020) συνεχούς λειτουργίας, περίπου 43kWh. Ενώ για μετρήσεις Watt σε διάστημα 48ωρών (24/3/2020 – 25/3/2020) αυτοματοποιημένης λειτουργίας, περίπου 12.6kWh μείωση 70,7%.**

Συμπεραίνουμε από την σύγκριση των μετρήσεων ότι το παραδοτέο ολοκληρωμένο σύστημα IoT ανταπεξέρχεται πλήρως στους στόχους του. Προσφέροντας μια αυτοματοποιημένη διαδικασία χωρίς ανθρώπινη παρέμβαση καταφέρνει να πραγματοποιεί σημαντική εξοικονόμηση ενέργειας. Το ποσοστό εξοικονόμησης δεν αναμένετε να είναι όσο υψηλό τις ημέρες που η εξωτερική θερμοκρασία είναι υψηλή (καλοκαιρινή περίοδος). Παρόλα αυτά αναμένετε να είναι μεγαλύτερη του 50%. Η συσκευή θα παραμείνει σε λειτουργία στον χώρο που εγκαταστάθηκε με παράλληλη πρόταση στην Διεύθυνση για επιπλέον ανάπτυξη συσκευών και εμπλουτισμού του υπάρχοντος οικοσυστήματος IoT. Με μελλοντικό στόχο την επέκταση του συστήματος σε ολόκληρο το κτίριο, για τους χώρους που υπάρχει τεχνολογικός εξοπλισμός με συνεχή κλιματιστική υποστήριξη.

Στην γραφική παράσταση που ακολουθεί (εικόνα 5.28) παρουσιάζεται αναλυτικά η εξοικονόμηση της ενέργεια που γίνεται σε watt καθώς και η κατανάλωση του κλιματιστικό χωρίς να «ζορίζεται» να αλλάξει μια θερμοκρασία απότομα. Αναλυτικότερα:

- 14:00 ενεργοποιείται η μονάδα και έχουμε άνοδο της κατανάλωσης
- 14:30 αλλάζει η θερμοκρασία του κλιματιστικού σε κοντινότερη θερμοκρασία που υπάρχει στο χώρο με πτώση της κατανάλωσης
- 15:30 με 16:30 το κλιματιστικό απενεργοποιείται (μηδενική κατανάλωση)
- 16:30 ενεργοποιείται η μονάδα με άνοδο της κατανάλωσης
- 17:00 αλλάζει η θερμοκρασία του κλιματιστικού σε κοντινότερη θερμοκρασία που υπάρχει στο χώρο με πτώση της κατανάλωσης
- 18:00 με 19:00 απενεργοποίηση με μηδενική κατανάλωση



Εικόνα 5.28 – Γράφημα με μικρότερο χρονικό περιορισμό για παρουσίαση απενεργοποίησης κλιματιστικού

Από τα μέσα του 2019 για τα Arduino boards υποστηρίζονται πλέον οι IoT υλοποιήσεις και με IoT Cloud δίνοντας στου διαχειριστές την δυνατότητα να ελέγχουν πολυάριθμες συσκευές που βρίσκονται σε διαφορετικά σημεία εγκαταστημένες καθώς και να ανανεώνουν απομακρυσμένα τον κώδικα τους. Κατά την συγγραφή της διατριβής δεν είχε προστεθεί η υποστήριξη του cloud computing στο αναπτυσσόμενο board που χρησιμοποιήθηκε, αλλά αναμένεται να υποστηριχθεί σύμφωνα με ανακοίνωση της κατασκευάστριας εταιρίας στο IoT Cloud [19].

5.2.1 Μελλοντική επεκτασιμότητα

Μια μελλοντική επεκτασιμότητα της αναπτυσσόμενης συσκευής μπορεί να είναι η διερεύνηση για δυνατότητα αλλαγής των τηλεχειριστηρίων των κλιματιστικών μονάδων από IR (Infrared) τεχνολογίας σε RF (Radio Frequency). Τα RF τηλεχειριστήρια χρησιμοποιούνται όλο και περισσότερο, αντικαθιστώντας σιγά σιγά την τεχνολογία IR. Στην ουσία η διαδικασία που ακολουθείται όταν πραγματοποιείται ένα πάτημα ενός πλήκτρου σε τηλεχειριστήριο RF είναι ότι η «εντολή» στέλνεται μέσω ραδιοσυχνοτήτων σε έναν RF δέκτη και εκεί αφού επεξεργαστεί με τη σειρά του μεταφράζεται σε εντολή IR και προωθείται εσωτερικά στο κομμάτι του εξοπλισμού που προσπαθούμε να έχουμε έλεγχο. Τα RF τηλεχειριστήρια τα συναντούσαμε συνήθως στα χειριστήρια από τις πόρτες garage, αλλά τα τελευταία χρόνια υιοθετούνται και στις τηλεοράσεις Samsung (εικόνα 5.29).

Εικόνα 5.29 - Τηλεχειριστήριο τηλεόρασης νέου τύπου με RF χρήση

[19] https://www.arduino.cc/en/Create/FAQ#toc13

Με πιο πρόσφατες εφαρμογές της αντικατάστασης από IR σε RF , την κατασκευή συσκευών που με χρήση Wi-Fi λαμβάνουνε τις ανάλογες εντολές από απομακρυσμένες (remote) πηγές και τις εκπέμπουν είτε με Radio Frequency είτε με Multiple-direction Infrared (γωνία 360 μοιρών με 7 πομπούς υπέρυθρης ακτινοβολίας). Ένα τέτοιο παράδειγμα συσκευής φαίνεται στην εικόνα που ακολουθεί (εικόνα 5.30). Τα πλεονέκτημα που προσφέρει ένα χειριστήριο RF είναι ότι δεν χρειάζεται να είναι καν στο ίδιο δωμάτιο με την συσκευή που ελέγχουμε, σε σχέση με το IR χειριστήριο που θα πρέπει να είναι κοντά στην συσκευή και χωρίς να υπάρχουν εμπόδια αναμεσά στο χειριστήριο και τη συσκευή.

Εικόνα 5.30 - BroadLink RM Pro+ ,IR and RF Smart Remote WiFi Controller

5.2.2 Βελτιώσεις των κλιματιζόμενων αιθουσών για περεταίρω εξοικονόμηση ενέργειας

Η συντήρηση, που δεν πραγματοποιείται πολύ συχνά στα εταιρικά κτίρια, παίζει σημαντικό ρόλο στην εξοικονόμηση ενέργειας. Παραδείγματα βελτίωσης:

- Αντικαθιστώντας ένα βρώμικο, φραγμένο φίλτρο αέρα με ένα καθαρό μπορεί να μειώσει την κατανάλωση ενέργειας του κλιματιστικού κατά 5-15 τοις εκατό[20],
- Η χρήση ανεμιστήρα οροφής βοηθάει σε ορισμένες περιπτώσεις στην καλύτερη κίνηση του αέρα στο κλιματισμένο χώρο.
- Μόνωση τους εσωτερικούς τοίχους με θερμομονωτικά υλικά, πχ. τύπου kraft energy save[21]

[20] https://www.energy.gov/articles/energy-saver-101-infographic-home-cooling
[21] https://kraftpaints.gr/product/energy-save-interior/

- Μόνωση εξωτερικών τοίχων, αν και στις περισσότερες περιπτώσεις είναι δύσκολο με μεγάλο κόστος.
- Εγκατάσταση / αντικατάσταση υαλοπινάκων με ενεργειακά τζάμια, τα οποία επιτρέπουν την διείσδυση του φωτός και εμποδίζουν την θερμότητα από τον ήλιο.
- Εγκατάσταση ενεργειακών film[22], μεμβράνες πάνω στα υπάρχοντα παράθυρα οι οποίες μπορούν να μειώσουν την θερμότητα από 55% μέχρι 80% με υψηλής ποιότητας μεμβράνες, φτάνοντας σε ενεργειακή εξοικονόμηση από 35 έως 50%.

[22] http://precisefilm.com/blog/2011/01/10/how-much-energy-does-window-film-save/

Βιβλιογραφία

[01] Baraka, K., Ghobril, M., Malek, S., Kanj, R., & Kayssi, A. (2013). Low cost Arduino/Android-based Energy-Efficient Home Automation System with Smart Task Scheduling. *Fifth International Conference on Computational Intelligence, Communication Systems and Networks* (σσ. 296-301). IEEE. doi:DOI 10.1109/CICSYN.2013.47

[02] Brambley, M., Haves, P., McDonald, S., Torcellini, P., Hansen, D., Holmberg, D., & Roth, K. (2005). *Advanced Sensors and Controls for Building Applications: Market Assessment and Potential R&D Pathways.* (U. D. Energy, Επιμ.) Springfield: National Technical Information Service, U.S. Department of Commerce.

[03] Chaudhari, S., Rathod, P., Shaikh, A., Vora, D., & Ahir, J. (2017). Smart Energy Meter Using Arduino and GSM. *International Conference on Trends in Electronics and Informatics (ICEI)* (σσ. 598-601). IEEE. doi:doi 10.1109_ICOEI.2017.8300772

[04] Colubri, A. (2017). *Processing for Android - Create Mobile, Sensor-Aware, and VR Applications Using Processing.* A-press.

[05] Dukish, B. (2018). *Coding the Arduino - Building Fun Programs, Games, and Electronic Projects.* A-Press.

[06] Gravina, R., Palau, C., Manso, M., Liotta, A., & Fortino, G. (2018). *Integration, Interconnection, and Interoperability of IoT Systems.* Switzerland: Springer International Publishing.

[07] Horalek, J., & Sobeslav, V. (2016). Measuring of Electric Energy Consumption in Households by Means of Arduino Platform. Στο H. Sulaiman, *Advanced Computer and Communication* (Τόμ. Lecture Notes in Electrical Engineering 362, σσ. 819-830). Switzerland: Springer International Publishing. doi:doi 10.1007_978-981-10-4852-4_6

[08] Isikdag, U. (2015). *Enhanced Building Information Models - Using IoT Services and Integration Patterns.* Springer Briefs in Computer Science.

[09] Jeyanthi, N., Abraham, A., & Mcheick, H. (2019). *Ubiquitous Computing and Computing Security of IoT.* Springer Nature Switzerland AG.

[10] Klemenjak, C., & Elmenreich, W. (2017). YaY - An Open-Hardware Energy Measurement System for Feedback and Appliance Detection based on the Arduino Platform. *13th Workshop on Intelligent Solutions in Embedded Systems (WISES).* IEEE. doi:doi 10.1109_WISES.2017.7986924

[11] Mahmood, Z. (2019). *Guide to Ambient Intelligence in the IoT Environment.* (C. C. Networks, Επιμ.) Switzerland: Springer.

[12] Mitton, N., Chaouchi, H., Noel, T., Watteyne, T., & Gabillon, A. (2016). *Interoperability, Safety and Security in IoT.* (P. Capolsini, Επιμ.) Switzerland: Springer International Publishing AG.

[13] Quintana, M., Lange, H., & Bergés, M. (2017). Demo Abstract: Design and Implementation of a Low-cost Arduino-based High-Frequency ACWaveform Meter Board for the Raspberry Pi. *BuildSys'17, November 8–9, 2017, Delft, The Netherlands.* ACM . doi:https://doi.org/10.1145/3137133.3141441

[14] Rahman, M., Jannat, N.-E., Ohidul , I., & Serazus , S. (2015). Arduino and GSM Based Smart Energy Meter for Advanced Metering and Billing System. *2nd Int'l Conf. on Electrical Engineering and Information & Communication Technology (ICEEICT).* Bangladesh: IEEE. doi:doi 10.1109_ICEEICT.2015.7307498

[15] Raju, L., Amalraj Morais, A., & Milton, R. (2018). Advanced Energy Management of a Micro-grid Using Arduino and Multi-agent System. *Intelligent and Efficient Electrical Systems*(Volume 446), σσ. 65-76. doi:https://doi.org/10.1007/978-981-10-4852-4_6

[16] Raza, A., Ikram, A., Amin, A., & Ikram, A. (2016). A Review of Low Cost and Power Efficient Development Boards for IoT Applications. *Future Technologies Conference* (σσ. 786-790). San Francisco: IEEE. doi:doi 10.1109_FTC.2016.7821693

[17] Risk, A. (2019). *I Own Your Building (Management System) - Building Management Systems Security Research.* Amsterdam, Netherlands: Applied Risk.

[18] Roth, K., Westphalen, D., Dieckmann, J., Hamilton, S., & Goetzler, W. (2002). *Energy Consumption Characteristics of Commercial Building HVAC Systems Volume III: Energy Savings Potential.* (J. Brodrick, Επιμ.) Cambridge: National Technical Information Service (NTIS) U.S. Department of Commerce.

[19] Santis, D., Giampetruzzi, D., Abbatantuono, G., & Scala, M. (2016). Smart Metering for Low Voltage Electrical Distribution System using Arduino Due. IEEE. doi:doi 10.1109_EESMS.2016.7504847

[20] Siozios, K., Anagnostos, D., Soudris, D., & Kosmatopoulos, E. (2019). *IoT for Smart Grids - Design Challenges and Paradigms.* Springer Nature Switzerland AG.

[21] Siu, C., & Iniewski, K. (2018). *IoT and Low-Power - Devices, Circuits, and Systems.* Taylor & Francis Group.

[22] Tripathy, B., & Anuradha, J. (2018). *INTERNET OF THINGS (IoT) - Technologies, Applications, Challenges, and Solutions.* CRC Press, Taylor & Francis Group.

[23] Ventje , A., Deisi , M., Mucdar , P., & Yoice , P. (2018). Utilization of Solar Cells as Energy Sources for Heating and Fan (Exhouse) in White Copra Dryers with Arduino Uno as Temperature Control. *International Conference on Applied Science and Technology (iCAST)* (σσ. 521-525). IEEE. doi:doi 10.1109_iCAST1.2018.8751614

[24] Zapata, B., & Niñirola, A. (2014). *Testing and Securing Android Studio Applications.* Packt Publishing Ltd.

Παράρτημα Α
Κώδικας Arduino

Α.1 Κώδικας για εξαγωγή IR σήματος

Ο παρακάτω κώδικας χρησιμοποιείται ως προεργασία στην παρούσα διατριβή ώστε να γίνει εξαγωγή των εντολών IR του τηλεχειριστηρίου σε μορφή κειμένου, για αυτό το λόγο δεν θεωρήθηκε σωστό να ενσωματωθεί και να επιβαρύνει τον τελικό κώδικα Α2.

Ο κώδικάς διατίθεται από την κατασκευάστρια εταιρία του Module που εγκαταστάθηκε πάνω στο Arduino Board με σκοπό να καταγραφεί και να αναμεταδώσει το IR σήμα με την χρήση 2 πλήκτρων που βρίσκονται πάνω στο module. Ο κώδικας χρησιμοποιήθηκε ώστε να καταγραφούν οι εντολές του τηλεχειριστήριου ανά πλήκτρο για περεταίρω χρήση τους.

Record_and_playback_IR_signals_v3.ino

```
/*
 * IR Remote Module
 *
 * Copyright (C) Libelium Comunicaciones Distribuidas S.L.
 * http://www.libelium.com
 *
 * This program is free software: you can redistribute it and/or modify
```

```
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program.  If not, see http://www.gnu.org/licenses/.
 *
 * Version:        1.0
 * Design:         David Gascón
 * Implementation:   Luis Martin
 */

// We need to use the 'raw' pin reading methods
// because timing is very important here and the digitalRead()
// procedure is slower!

#define IRpin_PIN PIND
#define IRpin 2

// The maximum pulse we'll listen for - 65 milliseconds is a long time
#define MAXPULSE 65000

// Timing resolution
#define RESOLUTION 20
// αρχικοποίηση ψηφιακών εξόδων
int IRledPin =  3;    // IR emitter LED connected to digital pin 3
int orangeLedPin =  7;  // Orange LED connected to digital pin 7
int buttonPin1 = 4;    // "Send" push-button connected to digital pin 4
int blueLedPin =  6;   // Blue LED connected to digital pin 6
```

```
int buttonPin2 = 5;    // "Receive" push-button connected to digital pin 5

int buttonState1 = 0;
int buttonState2 = 0;

// Store up to 100 pulse pairs
uint16_t pulses[100][2]; // Pair is high and low pulse
uint8_t currentpulse = 0; // Index for pulses we're storing
uint8_t sendpulse = 0; // Index for send pulses

void setup(void) {
  // Initialize the Orange LED pin as an output
  pinMode(orangeLedPin, OUTPUT);
  // Initialize the "Send" push-button pin as an input
  pinMode(buttonPin1, INPUT);
  // Initialize the Orange LED pin as an output
  pinMode(blueLedPin, OUTPUT);
  // Initialize the "Send" push-button pin as an input
  pinMode(buttonPin2, INPUT);
  // Set uart baudrate
  Serial.begin(9600);
}

void loop(void) {
   // Read the state of the "Send" push-button value
  buttonState1 = digitalRead(buttonPin1);
  // Read the state of the "Receive" push-button value
  buttonState2 = digitalRead(buttonPin2);

if (buttonState1 == HIGH) {
    Serial.println("Sending IR signal");
    digitalWrite(orangeLedPin, HIGH);
    delay(200);
    digitalWrite(orangeLedPin, LOW);
```

```
  delay(200);

  for (int i = 0; i <= 1; i++) {
  SendIRCode();
  delay(2000);
 }

//   SendIRCode();
//delay(15);  // wait 15 milliseconds before sending it again
 //SendIRCode();  // repeat IR code if it is neccesary
//   delay(5000);  // wait 5 seconds to resend the code
  }

 if ((buttonState2==HIGH) || (currentpulse != 0)){

  if (buttonState2==HIGH){
  Serial.println("Ready to decode IR!");
  digitalWrite(blueLedPin, HIGH);
  delay(200);
  digitalWrite(blueLedPin, LOW);
  delay(200);
 }

 uint16_t highpulse, lowpulse; // Temporary storage timing
 highpulse = lowpulse = 0; // Start out with no pulse length

  while (IRpin_PIN & (1 << IRpin)) {
   // count off another few microseconds
   highpulse++;
   delayMicroseconds(RESOLUTION);

   // If the pulse is too long, we 'timed out' - either nothing
   // was received or the code is finished, so print what
   // we've grabbed so far, and then reset
```

```
    if ((highpulse >= MAXPULSE) && (currentpulse != 0)) {
      printpulses();
      sendpulse=currentpulse;
      currentpulse=0;
      return;
    }
  }
  // we didn't time out so lets stash the reading
  pulses[currentpulse][0] = highpulse;

  // same as above
  while (! (IRpin_PIN & _BV(IRpin))) {
    // pin is still LOW
    lowpulse++;
    delayMicroseconds(RESOLUTION);
    if ((lowpulse >= MAXPULSE) && (currentpulse != 0)) {
      printpulses();
      sendpulse=currentpulse;
      currentpulse=0;
      return;
    }
  }
  pulses[currentpulse][1] = lowpulse;

  // we read one high-low pulse successfully, continue!
  currentpulse++;

 }
}

void printpulses(void) {
  Serial.println("\n\r\n\rReceived:");
  for (uint8_t i = 0; i < currentpulse; i++) {
```

```
  if (i != 0){
  Serial.print("delayMicroseconds(");
  Serial.print(pulses[i][0] * RESOLUTION);
  Serial.println(");");
  }
  Serial.print("pulseIR(");
  Serial.print(pulses[i][1] * RESOLUTION);
  Serial.println(");");
 }
 delay(1000);
 digitalWrite(orangeLedPin, HIGH);
 delay(200);
 digitalWrite(orangeLedPin, LOW);
 delay(200);

}

// This procedure sends a 38KHz pulse to the IRledPin for a certain # of microseconds.
void pulseIR(long microsecs) {

 cli();  // Turn off any background interrupts

 while (microsecs > 0) {
  // 38 kHz is about 13 microseconds high and 13 microseconds low
  digitalWrite(IRledPin, HIGH);  // 3 microseconds
  delayMicroseconds(10);       /* hang out for 10 microseconds,
                        you can also change this to 9 if its not working */
  digitalWrite(IRledPin, LOW);   // 3 microseconds
  delayMicroseconds(10);       /* hang out for 10 microseconds,
                        you can also change this to 9 if its not working */

  // So 26 microseconds altogether
  microsecs -= 26;
 }
```

```
  sei();  // Turn them back on
}
// δοκιμή καταγεγραμμένης εντολής για αποστολή με το πλήκτρο send-IR
void SendIRCode() {

pulseIR(8980);
delayMicroseconds(4180);
pulseIR(600);
delayMicroseconds(1500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(480);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(1580);
pulseIR(520);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(480);
pulseIR(560);
delayMicroseconds(480);
pulseIR(540);
delayMicroseconds(500);
```

```
pulseIR(560);
delayMicroseconds(480);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(1540);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(600);
delayMicroseconds(1480);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(1580);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(520);
pulseIR(8980);
delayMicroseconds(4180);
pulseIR(600);
```

```
delayMicroseconds(1500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(480);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(1580);
pulseIR(520);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(480);
pulseIR(560);
delayMicroseconds(480);
pulseIR(540);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(480);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(500);
pulseIR(560);
```

```
delayMicroseconds(1540);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(600);
delayMicroseconds(1480);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(1580);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(520);
}
```

A.2 Τελικός Κώδικας

Ο τελικός κώδικας που φορτώνεται στην Arduino συσκευή και συμπεριλαμβάνει τον διακομιστή web, αποστολή των σημάτων IR, επαναλαμβανόμενου βρόχου, και συναρτήσεις ασφαλείας για αξιόπιστη λειτουργικότητα.

WebServerWiFi-with-commands-final-version.ino

```
/*********************************************************************************
-Έκδοση κωδικά για χρήση με Arduino UNO Wi-Fi Rev 2-- version 7 last updated 11-3-2020
-Implementation: Πατρινός Δημήτρης
```

-Η έκδοση 7 περιέχει νέους IR code Signals

-connects to predefine wifi network with predefine IP address

- print out the ip address for accessing the Arduino module directly

- send ir commands directly to AC

- For the specific AC model, we send twice the ir command

- connect to remote server for storing temperature data for future analysis

- looping procedure for sending data at specific time period outside of main Loop function

- reconnect to wifi if connection to server fails

- soft reset and reconnect after 3, 1/2 hours, for cleaning buffers and stabilizing the board for overflow

- improve messages to the serial monitor.

**********/

```
//Libraries for Serial Peripheral Interface (SPI)
#include <SPI.h>
//Libraries for WiFi
#include <WiFiNINA.h>
//Libraries for temperature sensor
#include <DHT.h>
#include <avr/wdt.h>
#include "arduino_secrets.h"
//********** include library file where wifi sensitive data are stored tab/arduino_secrets.h
char ssid[] = SECRET_SSID;      // your network SSID (name)
char pass[] = SECRET_PASS;      // your network password (use for WPA, or use as key for WEP)
int keyIndex = 0;               // your network key Index number (needed only for WEP)
// ********************************************************************************
// Constants for ir transmitter module
int IRledPin =  3;    // IR emitter LED connected to digital pin 3
int orangeLedPin =  7;  // Orange LED connected to digital pin 7
int buttonPin1 = 4;    // "Send" push-button connected to digital pin 4
int buttonState1 = 0;
// ********************************************************************************
//Constants for temperature sensor
#define DHTPIN 13       // what pin we're connected to
#define DHTTYPE DHT22    // DHT 22  (AM2302)
```

```
DHT dht(DHTPIN, DHTTYPE); // Initialize DHT sensor for normal 16mhz Arduino
// ****************************************************************************************
//Variables for temperature sensor
float humidityData;    //Stores humidity value
float temperatureData;  //Stores temperature value
// ****************************************************************************************

//Variables for looping procedure to send data to remote server
int sendState = LOW;
unsigned long previousMillis = 0;
// reference address for trying different time loops
//https://convertlive.com/el/u/μετατροπή/ώρες/σε/χιλιοστά-του-δευτερολέπτου#1
// one hour --- const long interval = 3600000;
// 1/2 hour --- const long interval = 1800000;
// 1min --- const long interval = 60000;
// 5min --- const long interval = 300000;
// 10min --- const long interval = 600000;
// 15min --- const long interval = 900000;
const long interval = 1800000;
// ****************************************************************************************
int reset_counter=0;
// Initialize Variables for WiFi
IPAddress ip;
IPAddress staticIP(**********);
IPAddress gateway(**********);
IPAddress subnet(**********);
IPAddress dns1(**********);
IPAddress dns2(**********);
int status = WL_IDLE_STATUS;
WiFiServer server(80);
WiFiClient client;
char server_remote[] = "**********";
// ****************************************************************************************
```

```
void setup() {
// ************************************************************************************************
// Initialize the Orange LED pin as an output
 pinMode(orangeLedPin, OUTPUT);
// Initialize the "Send" push-button pin as an input
 pinMode(buttonPin1, INPUT);
// Initialize the IR digital pin as an output
 pinMode(IRledPin, OUTPUT);
// ************************************************************************************************
Serial.begin(9600);  // initialize serial communication
dht.begin();
connectWifi();
}
// ************************************************************************************************
void(* resetFunc) (void) = 0;//declare reset function at address 0
// ************************************************************************************************
void loop() {
// ************************************************************************************************
 if (WiFi.RSSI() == 0) // Checks if Wifi signal lost.
 { resetFunc(); //Calls Reset
 }
// ************************************************************************************************
if (reset_counter == 5){
 resetFunc(); //Calls Reset
}
// ************************************************************************************************
// Read the state of the "Send" push-button value
// for testing functionality
 buttonState1 = digitalRead(buttonPin1);

 if (buttonState1 == HIGH) {
  Serial.println("Sending IR signal with button");
  digitalWrite(orangeLedPin, HIGH);
  delay(200);
```

```
  digitalWrite(orangeLedPin, LOW);
  delay(200);

  SendIRCode_on();
  delay(15);  // wait 15 milliseconds before sending it again
  SendIRCode_on();  // repeat IR code it is necessary for the specific A/C model
  }
// ************************************************************************************************
// ************************************************************************************************
 WiFiClient client = server.available();   // listen for incoming clients

 if (client) {
  // if you get a client,
  if (WiFi.RSSI() == 0) // Checks if Wifi signal lost.
 { resetFunc(); //Calls Reset
 }
  Serial.println("new client");         // print a message out the serial port
  String currentLine = "";              // make a String to hold incoming data from the client
  while (client.connected()) {          // loop while the client's connected
   if (client.available()) {            // if there's bytes to read from the client,
    char c = client.read();             // read a byte, then
    Serial.write(c);                    // print it out the serial monitor
    if (c == '\n') {                    // if the byte is a newline character

     // if the current line is blank, you got two newline characters in a row.
     // that's the end of the client HTTP request, so send a response:
     if (currentLine.length() == 0) {
     // HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)
     // and a content-type so the client knows what's coming, then a blank line:
     client.println(" HTTP/1.1 200 OK");
     client.print("<head>");
     client.print("<TITLE />AC management via WiFi-with-auto-refresh</title>");
     client.print("</head>");
     client.println("Content-type:text/html");
```

```
client.println();

humidityData = dht.readHumidity();
temperatureData= dht.readTemperature();

client.print("<table border='1' align='center' width='400'>");
client.print("<tr><td colspan='2' align='center'>Remote air conditioning management via WiFi<br></td></tr>");
client.print("<tr><td colspan='2'><hr></td></tr>");
client.print("<tr><td><strong>Humidity</strong></td><td>");
client.print(humidityData);
client.print("</td></tr>");
client.print("<tr><td><strong>Temperature</strong></td><td>");
client.print(temperatureData);
client.print(" Celsius</td></tr>");
client.print("<tr><td colspan='2'><a href=\"/R\">Refresh values on Temperature and Humidity</a></td></tr>");
client.print("</table>");
client.print("<hr>");
client.print("<table border='1' align='center' width='400'>");
client.print("<tr><td colspan='2' align='center'>Control Links<br></td></tr>");
client.print("<tr><td colspan='2'>Click <strong><a href=\"/RESET\">RESET</a></strong> for Soft Reset</td></tr>");
client.print("<tr><td colspan='2'><hr></td></tr>");
client.print("<tr><td colspan='2'>Click <strong><a href=\"/ON\">ON</a></strong> to power on AC</td></tr>");
client.print("<tr><td colspan='2'>Click <strong><a href=\"/OFF\">OFF</a></strong> to power off AC</td></tr>");
client.print("<tr><td colspan='2'><hr></td></tr>");
client.print("<tr><td colspan='2'>Click <strong><a href=\"/18\">18</a></strong> for temperature 18</td></tr>");
client.print("<tr><td colspan='2'>Click <strong><a href=\"/19\">19</a></strong> for temperature 19</td></tr>");
```

```
client.print("<tr><td colspan='2'>Click <strong><a href=\"/20\">20</a></strong> for temperature 20</td></tr>");
client.print("<tr><td colspan='2'>Click <strong><a href=\"/21\">21</a></strong> for temperature 21</td></tr>");
client.print("<tr><td colspan='2'>Click <strong><a href=\"/22\">22</a></strong> for temperature 22</td></tr>");
client.print("<tr><td colspan='2'>Click <strong><a href=\"/23\">23</a></strong> for temperature 23</td></tr>");
client.print("<tr><td colspan='2'>Click <strong><a href=\"/24\">24</a></strong> for temperature 24</td></tr>");
client.print("<tr><td colspan='2'>Click <strong><a href=\"/25\">25</a></strong> for temperature 25</td></tr>");
client.print("<tr><td colspan='2'>Click <strong><a href=\"/26\">26</a></strong> for temperature 26</td></tr>");
client.print("<tr><td colspan='2'>Click <strong><a href=\"/27\">27</a></strong> for temperature 27</td></tr>");
client.print("<tr><td colspan='2'>Click <strong><a href=\"/28\">28</a></strong> for temperature 28</td></tr>");
client.print("<tr><td colspan='2'>Click <strong><a href=\"/29\">29</a></strong> for temperature 29</td></tr>");
client.print("<tr><td colspan='2'>Click <strong><a href=\"/30\">30</a></strong> for temperature 30</td></tr>");
client.print("</table>");

        // The HTTP response ends with another blank line:
        client.println();
        // break out of the while loop:
        break;
      } else {    // if you got a newline, then clear currentLine:
        currentLine = "";
      }
    } else if (c != '\r') {  // if you got anything else but a carriage return character,
      currentLine += c;      // add it to the end of the currentLine
    }
```

```
      if (currentLine.endsWith("GET /R")) {
      // refresh page
      }
      if (currentLine.endsWith("GET /RESET")) {
      // reset function
      resetFunc();
      }
     // Check to see if the client request was "GET /??"

// ********************************************************************************************
/* the code that is repeated inside each function calling is for visual confirmation on arduino board.
digitalWrite(orangeLedPin, HIGH);
delay(200);
digitalWrite(orangeLedPin, LOW);
delay(200);
*/
// ********************************************************************************************

    if (currentLine.endsWith("GET /ON")) {
      Serial.println(" ");
      Serial.println("Sending IR signal on");
      digitalWrite(orangeLedPin, HIGH);
      delay(200);
      digitalWrite(orangeLedPin, LOW);
      delay(200);
      SendIRCode_on();            // call SendIRCode_on function
      delay(15);                // wait 15 milliseconds before sending it again
      SendIRCode_on();            // repeat send IR code neccesary for the AC that we developing
      // delay(2000);              // wait 2 seconds to return to listening status
     }
    if (currentLine.endsWith("GET /OFF")) {
      Serial.println(" ");
      Serial.println("Sending IR signal off");
```

```
  digitalWrite(orangeLedPin, HIGH);
  delay(200);
  digitalWrite(orangeLedPin, LOW);
  delay(200);
  SendIRCode_off();          // call SendIRCode_off function
  delay(15);               // wait 15 milliseconds before sending it again
  SendIRCode_off();          // repeat send IR code neccesary for the AC that we developing
  // delay(2000);            // wait 2 seconds to return to listening status
 }
if (currentLine.endsWith("GET /18")) {
  Serial.println(" ");
  Serial.println("Sending IR signal 18");
  digitalWrite(orangeLedPin, HIGH);
  delay(200);
  digitalWrite(orangeLedPin, LOW);
  delay(200);
  SendIRCode_18();          // call SendIRCode_18 function
  delay(15);               // wait 15 milliseconds before sending it again
  SendIRCode_18();          // repeat send IR code neccesary for the AC that we developing
  // delay(2000);            // wait 2 seconds to return to listening status
 }
if (currentLine.endsWith("GET /19")) {
  Serial.println(" ");
  Serial.println("Sending IR signal 19");
  digitalWrite(orangeLedPin, HIGH);
  delay(200);
  digitalWrite(orangeLedPin, LOW);
  delay(200);
  SendIRCode_19();          // call SendIRCode_19 function
  delay(15);               // wait 15 milliseconds before sending it again
  SendIRCode_19();          // repeat send IR code neccesary for the AC that we developing
  // delay(2000);            // wait 2 seconds to return to listening status
 }
if (currentLine.endsWith("GET /20")) {
```

```
  Serial.println(" ");
  Serial.println("Sending IR signal 20");
  digitalWrite(orangeLedPin, HIGH);
  delay(200);
  digitalWrite(orangeLedPin, LOW);
  delay(200);
  SendIRCode_20();          // call SendIRCode_20 function
  delay(15);              // wait 15 milliseconds before sending it again
  SendIRCode_20();          // repeat send IR code neccesary for the AC that we developing
  // delay(2000);             // wait 2 seconds to return to listening status
 }
if (currentLine.endsWith("GET /21")) {
  Serial.println(" ");
  Serial.println("Sending IR signal 21");
  digitalWrite(orangeLedPin, HIGH);
  delay(200);
  digitalWrite(orangeLedPin, LOW);
  delay(200);
  SendIRCode_21();          // call SendIRCode_21 function
  delay(15);              // wait 15 milliseconds before sending it again
  SendIRCode_21();          // repeat send IR code neccesary for the AC that we developing
  //delay(2000);             // wait 2 seconds to return to listening status
 }
if (currentLine.endsWith("GET /22")) {
  Serial.println(" ");
  Serial.println("Sending IR signal 22");
  digitalWrite(orangeLedPin, HIGH);
  delay(200);
  digitalWrite(orangeLedPin, LOW);
  delay(200);
  SendIRCode_22();          // call SendIRCode_22 function
  delay(15);              // wait 15 milliseconds before sending it again
  SendIRCode_22();          // repeat send IR code neccesary for the AC that we developing
  // delay(2000);             // wait 2 seconds to return to listening status
```

```
 }
if (currentLine.endsWith("GET /23")) {
  Serial.println(" ");
  Serial.println("Sending IR signal 23");
  digitalWrite(orangeLedPin, HIGH);
  delay(200);
  digitalWrite(orangeLedPin, LOW);
  delay(200);
  SendIRCode_23();          // call SendIRCode_23 function
  delay(15);              // wait 15 milliseconds before sending it again
  SendIRCode_23();          // repeat send IR code neccesary for the AC that we developing
  // delay(2000);            // wait 2 seconds to return to listening status
 }
if (currentLine.endsWith("GET /24")) {
  Serial.println(" ");
  Serial.println("Sending IR signal 24");
  digitalWrite(orangeLedPin, HIGH);
  delay(200);
  digitalWrite(orangeLedPin, LOW);
  delay(200);
  SendIRCode_24();          // call SendIRCode_24 function
  delay(15);              // wait 15 milliseconds before sending it again
  SendIRCode_24();          // repeat send IR code neccesary for the AC that we developing
  //delay(2000);            // wait 2 seconds to return to listening status
 }
if (currentLine.endsWith("GET /25")) {
  Serial.println(" ");
  Serial.println("Sending IR signal 25");
  digitalWrite(orangeLedPin, HIGH);
  delay(200);
  digitalWrite(orangeLedPin, LOW);
  delay(200);
  SendIRCode_25();          // call SendIRCode_25 function
  delay(15);              // wait 15 milliseconds before sending it again
```

```
    SendIRCode_25();          // repeat send IR code neccesary for the AC that we developing
    //delay(2000);            // wait 2 seconds to return to listening status
   }
  if (currentLine.endsWith("GET /26")) {
    Serial.println(" ");
    Serial.println("Sending IR signal 26");
    digitalWrite(orangeLedPin, HIGH);
    delay(200);
    digitalWrite(orangeLedPin, LOW);
    delay(200);
    SendIRCode_26();          // call SendIRCode_26 function
    delay(15);              // wait 15 milliseconds before sending it again
    SendIRCode_26();          // repeat send IR code neccesary for the AC that we developing
   // delay(2000);            // wait 2 seconds to return to listening status
   }
  if (currentLine.endsWith("GET /27")) {
    Serial.println(" ");
    Serial.println("Sending IR signal 27");
    digitalWrite(orangeLedPin, HIGH);
    delay(200);
    digitalWrite(orangeLedPin, LOW);
    delay(200);
    SendIRCode_27();          // call SendIRCode_27 function
    delay(15);              // wait 15 milliseconds before sending it again
    SendIRCode_27();          // repeat send IR code neccesary for the AC that we developing
    //delay(2000);            // wait 2 seconds to return to listening status
   }
  if (currentLine.endsWith("GET /28")) {
    Serial.println(" ");
    Serial.println("Sending IR signal 28");
    digitalWrite(orangeLedPin, HIGH);
    delay(200);
    digitalWrite(orangeLedPin, LOW);
    delay(200);
```

```
    SendIRCode_28();          // call SendIRCode_28 function
    delay(15);              // wait 15 milliseconds before sending it again
    SendIRCode_28();          // repeat send IR code neccesary for the AC that we developing
    //delay(2000);            // wait 2 seconds to return to listening status
   }
  if (currentLine.endsWith("GET /29")) {
    Serial.println(" ");
    Serial.println("Sending IR signal 29");
    digitalWrite(orangeLedPin, HIGH);
    delay(200);
    digitalWrite(orangeLedPin, LOW);
    delay(200);
    SendIRCode_29();          // call SendIRCode_29 function
    delay(15);              // wait 15 milliseconds before sending it again
    SendIRCode_29();          // repeat send IR code neccesary for the AC that we developing
    //delay(2000);            // wait 2 seconds to return to listening status
   }
  if (currentLine.endsWith("GET /30")) {
    Serial.println(" ");
    Serial.println("Sending IR signal 30");
    digitalWrite(orangeLedPin, HIGH);
    delay(200);
    digitalWrite(orangeLedPin, LOW);
    delay(200);
    SendIRCode_30();          // call SendIRCode_30 function
    delay(15);              // wait 15 milliseconds before sending it again
    SendIRCode_30();          // repeat send IR code neccesary for the AC that we developing
    //delay(2000);            // wait 2 seconds to return to listening status
   }


  }
 }
// close the connection:
// ********************************************************************************************
```

```
  delay(10);

  client.stop();
  Serial.println("client disonnected after loading arduino page");
 }
// ********************************************************************************************
// repeated time loop where we don't freeze the local webserver
unsigned long currentMillis = millis();
 if (currentMillis - previousMillis >= interval) {
// save the last time we use sendState
   previousMillis = currentMillis;

// if the sendState is low turn it high and vice-versa:
 if (sendState == LOW)
 { // start if the sendState is low
   sendState = HIGH;
   reset_counter= reset_counter+1;
   humidityData = dht.readHumidity();
   temperatureData= dht.readTemperature();
   Serial.println("VALUE OF THE reset_counter:");
   Serial.println(reset_counter);

// ********************************************************************************************
//Sending sensor data to remote server start
Serial.println("before first if of the time interval");
//if client active local arduino connection
if (client.connect(server_remote, 8080)) {
 Serial.println("inside first time interval, client connected");
  ip = WiFi.localIP();
   Serial.println("storing dynamic ip from WiFi for connecting directly to arduino");
   Serial.println(ip);
   Serial.println("Sending sensor data to remote server for future analysis");
    // Make a HTTP request:
    if (reset_counter != 5)
```

```
    {
    Serial.print("GET /dipatrin/android/pages/dht.php?humidity=");
    client.print("GET /dipatrin/android/pages/dht.php?humidity=");    //YOUR URL
    Serial.print(humidityData);
    client.print(humidityData);
    Serial.print("&temperature=");
    client.print("&temperature=");
    Serial.print(temperatureData);
    client.print(temperatureData);
    Serial.print("&ip=");
    client.print("&ip=");
    Serial.println(ip);
    client.print(ip);
    client.print(" ");     //SPACE BEFORE HTTP/1.1
    client.print("HTTP/1.1");
    client.println();
    client.println("Host: **********");
    client.println("Connection: close");
    client.println();
    }
 } else {
   // end else client active local arduino connection
   // if you didn't get a connection to the server:
   Serial.println("before first else of the time interval");
   Serial.println("connection to remote server failed");
// ********************************************************************************
connectWifi();
Serial.println("Reconnection to WiFi");
WiFiClient client = server.available();   // listen for incoming clients
client.connect(server_remote, 8080);
if (client.connect(server_remote, 8080)) {
Serial.println("Reconnection to remote server");
ip = WiFi.localIP();
   Serial.println("storing dynamic ip from WiFi for connecting directly to arduino");
```

```
   Serial.println(ip);
Serial.println("Sending sensor data to remote server for future analysis after reconnection");
    // Make a HTTP request:
    if (reset_counter != 5)
    {
    Serial.print("GET /dipatrin/android/pages/dht.php?humidity=");
    client.print("GET /dipatrin/android/pages/dht.php?humidity=");    //YOUR URL
    Serial.print(humidityData);
    client.print(humidityData);
    Serial.print("&temperature=");
    client.print("&temperature=");
    Serial.print(temperatureData);
    client.print(temperatureData);
    Serial.print("&ip=");
    client.print("&ip=");
    Serial.println(ip);
    client.print(ip);
    client.print(" ");    //SPACE BEFORE HTTP/1.1
    client.print("HTTP/1.1");
    client.println();
    client.println("Host: **********");
    client.println("Connection: close");
    client.println();
    }

}
// ********************************************************************************************
 } //end else client active local arduino connection
// ********************************************************************************************
   }  // end if the sendState is low
   else // else from the LOW if
   { // start else from the LOW if
    sendState = LOW;
    reset_counter= reset_counter+1;
```

```
    humidityData = dht.readHumidity();
    temperatureData= dht.readTemperature();

    Serial.println("VALUE OF THE reset_counter:");
    Serial.println(reset_counter);
Serial.println("before second if of the time interval");
// ********************************************************************************************
//Sending sensor data to remote server start
//if client active local arduino connection
 if (client.connect(server_remote, 8080))
  {
    Serial.println("inside second time interval, client connected");
    ip = WiFi.localIP();
    Serial.println("storing dynamic ip from WiFi for connecting directly to arduino");
    Serial.println(ip);
    Serial.println("Sending sensor data to remote server for future analysis");
    // Make a HTTP request:
    if (reset_counter != 5)
    {
    Serial.print("GET /dipatrin/android/pages/dht.php?humidity=");
    client.print("GET /dipatrin/android/pages/dht.php?humidity=");    //YOUR URL
    Serial.print(humidityData);
    client.print(humidityData);
    Serial.print("&temperature=");
    client.print("&temperature=");
    Serial.print(temperatureData);
    client.print(temperatureData);
    Serial.print("&ip=");
    client.print("&ip=");
    Serial.println(ip);
    client.print(ip);
    client.print(" ");     //SPACE BEFORE HTTP/1.1
    client.print("HTTP/1.1");
    client.println();
```

```
    client.println("Host: **********");
    client.println("Connection: close");
    client.println();
    }
   }
   else // start else client active local arduino connection
   {
    // if you didn't get a connection to the server:
   Serial.println("before second else of the time interval");
   Serial.println("connection to remote server failed");
// ***************************************************************************************
connectWifi();
Serial.println("Reconnection to WiFi");
WiFiClient client = server.available();   // listen for incoming clients
client.connect(server_remote, 8080);
if (client.connect(server_remote, 8080)) {
Serial.println("Reconnection to remote server");
ip = WiFi.localIP();
  Serial.println("storing dynamic ip from WiFi for connecting directly to arduino");
   Serial.println(ip);
Serial.println("Sending sensor data to remote server for future analysis after reconnection");
    // Make a HTTP request:
    if (reset_counter != 5)
    {
    Serial.print("GET /dipatrin/android/pages/dht.php?humidity=");
    client.print("GET /dipatrin/android/pages/dht.php?humidity=");    //YOUR URL
    Serial.print(humidityData);
    client.print(humidityData);
    Serial.print("&temperature=");
    client.print("&temperature=");
    Serial.print(temperatureData);
    client.print(temperatureData);
    Serial.print("&ip=");
    client.print("&ip=");
```

```
    Serial.println(ip);
    client.print(ip);
    client.print(" ");     //SPACE BEFORE HTTP/1.1
    client.print("HTTP/1.1");
    client.println();
    client.println("Host: **********");
    client.println("Connection: close");
    client.println();
    }
}
// ************************************************************************************************
  } //end else client active local arduino connection
// ************************************************************************************************
  } // end else from the LOW if
// ************************************************************************************************

} // end time Loop for sending data to database
// ************************************************************************************************

} // end of loop stage
// ************************************************************************************************

void connectWifi() {
// initializing Variables
status=WiFi.disconnect();
int status = WL_IDLE_STATUS; //second time?
// check for the WiFi module:
 if (WiFi.status() == WL_NO_MODULE) {
  Serial.println("Communication with WiFi module failed!");
 while (true);
 }

// ************************************************************************************************
String fv = WiFi.firmwareVersion();
```

```
  if (fv < WIFI_FIRMWARE_LATEST_VERSION) {
    Serial.println("Please upgrade the firmware");
  }
// ************************************************************************************************
// attempt to connect to Wifi network:
  while (status != WL_CONNECTED) {
    Serial.print("Attempting to connect to Network named: ");
    Serial.println(ssid);                   // print the network name (SSID);
    // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
  WiFi.config(staticIP, dns1, gateway, subnet);
    status = WiFi.begin(ssid, pass);
    // wait 10 seconds for connection:
    delay(10000);
  }
  Serial.println("Connected to wifi");
  server.begin();                           // start the web server on port 80
  printWifiStatus();                        // you're connected now, so print out the status
}

// ************************************************************************************************
void printWifiStatus() {
  // print the SSID of the network you're attached to:
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());

  // print your board's IP address:
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);

  // print the received signal strength:
  long rssi = WiFi.RSSI();
  Serial.print("signal strength (RSSI):");
  Serial.print(rssi);
```

```
 Serial.println(" dBm");
 // print where to go in a browser:
 Serial.print("To see this page in action, open a browser to http://");
 Serial.println(ip);
}
// ************************************************************************************************
// This procedure sends a 38KHz pulse to the IRledPin for a certain # of microseconds.
void pulseIR(long microsecs) {

 cli();  // Turn off any background interrupts

 while (microsecs > 0) {
  // 38 kHz is about 13 microseconds high and 13 microseconds low
  digitalWrite(IRledPin, HIGH);  // 3 microseconds
  delayMicroseconds(9);       /* hang out for 10 microseconds,
                      you use thos to 9 for Arduino Uno WiFi edition 10 for Arduino Uno*/
  digitalWrite(IRledPin, LOW);   // 3 microseconds
  delayMicroseconds(9);       /* hang out for 10 microseconds,
                      you use thos to 9 for Arduino Uno WiFi edition 10 for Arduino Uno */

  // So 26 microseconds altogether
  microsecs -= 26;
 }
 //Serial.println("inside function pulseIR");
 sei();  // Turn them back on
}
// ************************************************************************************************


// ************************************************************************************************
void SendIRCode_on() {
/*****************************************************
 * IR CODE FOR ON COMMAND TO 22 TEMPERATURE COLD MODE *
 * ***************************************************/
```

```
pulseIR(8980);
delayMicroseconds(4180);
pulseIR(600);
delayMicroseconds(1500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(480);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(1580);
pulseIR(520);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(480);
pulseIR(560);
delayMicroseconds(480);
pulseIR(540);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(480);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
```

```
pulseIR(520);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(1540);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(600);
delayMicroseconds(1480);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(1580);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(520);
}
// ************************************************************************************************


// ************************************************************************************************
void SendIRCode_off() {
/*****************************************************
 * IR CODE FOR OFF COMMAND                    *
 * ***************************************************/
```

```
pulseIR(8960);
delayMicroseconds(4180);
pulseIR(600);
delayMicroseconds(1500);
pulseIR(580);
delayMicroseconds(460);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(560);
delayMicroseconds(1520);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(580);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(560);
delayMicroseconds(460);
pulseIR(520);
delayMicroseconds(520);
```

```
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(580);
delayMicroseconds(440);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(1580);
pulseIR(560);
delayMicroseconds(460);
pulseIR(520);
delayMicroseconds(1580);
pulseIR(580);
delayMicroseconds(440);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(560);
}
// ************************************************************************************************

// ************************************************************************************************
void SendIRCode_18() {
/*************************************************
 * IR CODE FOR COMMAND TO on-18 TEMPERATURE COLD MODE   *
 * ************************************************/
pulseIR(8980);
```

```
delayMicroseconds(4160);
pulseIR(600);
delayMicroseconds(1500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(500);
delayMicroseconds(540);
pulseIR(500);
delayMicroseconds(540);
pulseIR(520);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(500);
pulseIR(500);
delayMicroseconds(540);
pulseIR(520);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
```

```
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(520);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(1580);
pulseIR(500);
delayMicroseconds(520);
pulseIR(560);
delayMicroseconds(480);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(580);
}
// ************************************************************************************************

// ************************************************************************************************
void SendIRCode_19() {
/*************************************************
 * IR CODE FOR COMMAND TO on-19 TEMPERATURE COLD MODE   *
 * ************************************************/
pulseIR(8980);
delayMicroseconds(4180);
```

```
pulseIR(580);
delayMicroseconds(1520);
pulseIR(520);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(500);
delayMicroseconds(540);
pulseIR(540);
delayMicroseconds(480);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
```

```
pulseIR(520);
delayMicroseconds(1580);
pulseIR(520);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
}
// ********************************************************************************************

// ********************************************************************************************
void SendIRCode_20() {
/*************************************************
 * IR CODE FOR COMMAND TO on-20 TEMPERATURE COLD MODE   *
 * ************************************************/
pulseIR(8960);
delayMicroseconds(4180);
pulseIR(600);
```

```
delayMicroseconds(1500);
pulseIR(540);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(540);
pulseIR(500);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(480);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(560);
delayMicroseconds(480);
pulseIR(540);
```

```
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(520);
pulseIR(480);
delayMicroseconds(1580);
pulseIR(540);
delayMicroseconds(520);
pulseIR(560);
delayMicroseconds(480);
pulseIR(560);
delayMicroseconds(1500);
pulseIR(540);
delayMicroseconds(520);
pulseIR(500);
delayMicroseconds(540);
pulseIR(520);
delayMicroseconds(1540);
pulseIR(540);
}
// ********************************************************************************************

// ********************************************************************************************
void SendIRCode_21() {
/*************************************************
 * IR CODE FOR COMMAND TO on-21 TEMPERATURE COLD MODE   *
 * ************************************************/
pulseIR(8960);
delayMicroseconds(4180);
pulseIR(600);
delayMicroseconds(1520);
```

```
pulseIR(500);
delayMicroseconds(520);
pulseIR(580);
delayMicroseconds(460);
pulseIR(520);
delayMicroseconds(520);
pulseIR(500);
delayMicroseconds(1580);
pulseIR(540);
delayMicroseconds(500);
pulseIR(500);
delayMicroseconds(540);
pulseIR(500);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(500);
pulseIR(500);
delayMicroseconds(540);
pulseIR(500);
delayMicroseconds(540);
pulseIR(500);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(500);
delayMicroseconds(540);
pulseIR(500);
delayMicroseconds(540);
pulseIR(520);
delayMicroseconds(1560);
```

```
pulseIR(500);
delayMicroseconds(1580);
pulseIR(600);
delayMicroseconds(440);
pulseIR(520);
delayMicroseconds(520);
pulseIR(500);
delayMicroseconds(1580);
pulseIR(600);
delayMicroseconds(440);
pulseIR(500);
delayMicroseconds(540);
pulseIR(580);
delayMicroseconds(1500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(500);
delayMicroseconds(1580);
pulseIR(520);
delayMicroseconds(520);
pulseIR(600);
}
// ************************************************************************************

// ************************************************************************************
void SendIRCode_22() {
/*************************************************
 * IR CODE FOR COMMAND TO on-22 TEMPERATURE COLD MODE   *
 * ************************************************/
pulseIR(8960);
delayMicroseconds(4200);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(500);
```

```
delayMicroseconds(520);
pulseIR(600);
delayMicroseconds(440);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(600);
delayMicroseconds(440);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(500);
pulseIR(500);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(480);
pulseIR(520);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(520);
```

```
delayMicroseconds(1580);
pulseIR(580);
delayMicroseconds(1500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(600);
delayMicroseconds(1500);
pulseIR(500);
}
// ************************************************************************************************

// ************************************************************************************************
void SendIRCode_23() {
/*************************************************
 * IR CODE FOR COMMAND TO on-23 TEMPERATURE COLD MODE   *
 * ************************************************/
pulseIR(9000);
delayMicroseconds(4160);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(500);
```

```
pulseIR(520);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(500);
pulseIR(500);
delayMicroseconds(1580);
pulseIR(540);
delayMicroseconds(500);
pulseIR(580);
delayMicroseconds(460);
pulseIR(520);
delayMicroseconds(520);
pulseIR(500);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(500);
pulseIR(500);
delayMicroseconds(540);
pulseIR(500);
delayMicroseconds(540);
pulseIR(500);
delayMicroseconds(520);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
```

```
pulseIR(520);
delayMicroseconds(520);
pulseIR(500);
delayMicroseconds(540);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(500);
delayMicroseconds(540);
pulseIR(500);
delayMicroseconds(540);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(500);
delayMicroseconds(1580);
pulseIR(600);
delayMicroseconds(440);
pulseIR(500);
delayMicroseconds(540);
pulseIR(520);
}
// ************************************************************************************************

// ************************************************************************************************
void SendIRCode_24() {
/*************************************************
 * IR CODE FOR COMMAND TO 24 TEMPERATURE COLD MODE   *
 * ************************************************/
pulseIR(8980);
delayMicroseconds(4160);
pulseIR(600);
delayMicroseconds(1500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
```

```
delayMicroseconds(500);
pulseIR(580);
delayMicroseconds(460);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(500);
pulseIR(580);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(480);
pulseIR(540);
delayMicroseconds(500);
pulseIR(580);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(580);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
```

```
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(580);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
}
// ************************************************************************************************

// ************************************************************************************************
void SendIRCode_25() {
/*************************************************
 * IR CODE FOR COMMAND TO 25 TEMPERATURE COLD MODE   *
 * ************************************************/
pulseIR(8980);
delayMicroseconds(4180);
pulseIR(600);
delayMicroseconds(1500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
```

```
pulseIR(560);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(1580);
pulseIR(520);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(580);
delayMicroseconds(460);
pulseIR(520);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(560);
delayMicroseconds(480);
pulseIR(520);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1580);
pulseIR(500);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(500);
delayMicroseconds(540);
```

```
pulseIR(520);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(560);
delayMicroseconds(480);
pulseIR(520);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(520);
delayMicroseconds(520);
pulseIR(560);
}
// ************************************************************************************************

// ************************************************************************************************
void SendIRCode_26() {
/*************************************************
 * IR CODE FOR COMMAND TO 26 TEMPERATURE COLD MODE   *
 * ************************************************/
pulseIR(8980);
delayMicroseconds(4180);
pulseIR(600);
delayMicroseconds(1500);
pulseIR(520);
delayMicroseconds(520);
pulseIR(520);
delayMicroseconds(500);
pulseIR(540);
```

```
delayMicroseconds(500);
pulseIR(580);
delayMicroseconds(1520);
pulseIR(520);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(580);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(560);
delayMicroseconds(480);
pulseIR(560);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(560);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
```

```
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(1520);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(520);
}
// ************************************************************************************************

// ************************************************************************************************
void SendIRCode_27() {
/*************************************************
 * IR CODE FOR COMMAND TO 27 TEMPERATURE COLD MODE   *
 * ************************************************/
pulseIR(8980);
delayMicroseconds(4180);
pulseIR(600);
delayMicroseconds(1500);
pulseIR(540);
delayMicroseconds(480);
pulseIR(540);
delayMicroseconds(500);
pulseIR(580);
delayMicroseconds(460);
```

```
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(500);
pulseIR(580);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(560);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
```

```
pulseIR(540);
delayMicroseconds(1560);
pulseIR(560);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(560);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(580);
}
// ************************************************************************************************

// ************************************************************************************************
void SendIRCode_28() {
/*************************************************
 * IR CODE FOR COMMAND TO 28 TEMPERATURE COLD MODE   *
 * ************************************************/
pulseIR(8980);
delayMicroseconds(4180);
pulseIR(600);
delayMicroseconds(1500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(480);
pulseIR(580);
delayMicroseconds(460);
pulseIR(560);
```

```
delayMicroseconds(1520);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(580);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(560);
delayMicroseconds(1520);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
```

```
delayMicroseconds(1540);
pulseIR(580);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(580);
delayMicroseconds(460);
pulseIR(520);
delayMicroseconds(520);
pulseIR(560);
delayMicroseconds(1520);
pulseIR(580);
}
// ************************************************************************************************

// ************************************************************************************************
void SendIRCode_29() {
/*************************************************
 * IR CODE FOR COMMAND TO 29 TEMPERATURE COLD MODE   *
 * ************************************************/
pulseIR(9000);
delayMicroseconds(4160);
pulseIR(600);
delayMicroseconds(1500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(480);
pulseIR(540);
delayMicroseconds(1540);
```

```
pulseIR(540);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(480);
pulseIR(540);
delayMicroseconds(480);
pulseIR(560);
delayMicroseconds(480);
pulseIR(560);
delayMicroseconds(480);
pulseIR(580);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(580);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(480);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1560);
```

```
pulseIR(540);
delayMicroseconds(480);
pulseIR(560);
delayMicroseconds(480);
pulseIR(560);
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(480);
pulseIR(560);
delayMicroseconds(1540);
pulseIR(580);
delayMicroseconds(460);
pulseIR(560);
}
// ************************************************************************************************

// ************************************************************************************************
void SendIRCode_30() {
/*************************************************
 * IR CODE FOR COMMAND TO 30 TEMPERATURE COLD MODE   *
 * ************************************************/
pulseIR(9000);
delayMicroseconds(4160);
pulseIR(600);
delayMicroseconds(1500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(540);
```

```
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(580);
delayMicroseconds(460);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(1520);
pulseIR(540);
```

```
delayMicroseconds(500);
pulseIR(540);
delayMicroseconds(500);
pulseIR(560);
delayMicroseconds(1520);
pulseIR(540);
delayMicroseconds(500);
pulseIR(520);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(1540);
pulseIR(540);
}
// ********************************************************************************************
```

Παράρτημα Β
Κώδικας εξωτερικού Web διακομιστή

Β.1 Κώδικας τρέχουσας κατάστασης συσκευών

Ο παρακάτω κώδικας χρησιμοποιείται για ενημέρωση του χρήστη για την κατάσταση του κλιματιστικού και της IoT συσκευής.

home.php

```
<?php
//*************************************
// status of the ac
// implementation: Patrinos Dimitris
//*************************************
$servername = "localhost";
$username = "**********";
$password = " ********** ";
$dbname = " ********** ";
```

```
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
$sql = "select * from ac_automation ORDER BY id DESC LIMIT 1";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
  // output data of each row
  while($row = $result->fetch_assoc()) {
  //echo "latest stored targeted record set[id: " . $row["id"]. " - target_temperature: " .
$row["target_temperature"]. " - status:". $row["status"]."]<br>";
        $target_temperature_control = $row["target_temperature"];
        $target_status = $row["status"];
        $last_send_temp = $row["last_send_temp"];
  }
} else {
  //echo "0 results";
}
$sql_ip = "select * from ac_temp_log ORDER BY id DESC LIMIT 1";
$result_ip = $conn->query($sql_ip);
if ($result_ip->num_rows > 0) {
  // output data of each row
  while($row = $result_ip->fetch_assoc()) {
  //echo "last ip [ip address: " . $row["ip"]. "]<br>";
        $stored_ip_address = $row["ip"];
        $last_stored_temperature = $row["'Temperature"];
  }
} else {
  //echo "0 results";
}
//************************************************************
```

```
// put an @ in front of function call to suppress all error messages.
// check if the arduino device is alive, with ping the last ip address
// reference from coding php forum
// https://stackoverflow.com/questions/1239068/ping-site-and-return-result-in-php
$arduino_status = @ping($stored_ip_address, 80, 10);

function ping($host, $port, $timeout) {
 $arduino_status = 3;
 $tB = microtime(true);
 $fP = fSockOpen($host, $port, $errno, $errstr, $timeout);
 if (!$fP) {
            //echo "<br>arduino down check power supply <br>";
            //echo "<br>**************************<br>";
            $arduino_status = 0;
            return $arduino_status;
            }
 $tA = microtime(true);
           //echo "<br>arduino is running ok<br>";
           //echo "<br>**************************<br>";
           $arduino_status = 1;
           //the ping time in ms
           //return round((($tA - $tB) * 1000), 0)." ms<br>";
           return $arduino_status;
}
//************************************************************

?>
<html>
 <head> </head>
 <body>
  <h2>Τρέχουσα κατάσταση συσκευών</h2>
  <hr>
   <table style="width:300px" style="height:700px" border ="1">
  <tr>
```

```
  <th>Arduino Status</th>
  <th>AC Last Status</th>
  <th>Στατιστικά Θερμοκρασίας</th>
 </tr>
 <tr> <td>
<?php
if ($arduino_status == 0)
{?> <img src="../icons/plug-off.png" width="130" height="176">
<?php  }
elseif ($arduino_status == 1)
{ ?>
<img src="../icons/plug-on.png" width="130" height="176">
<?php  } ?>
</td>  <td>
<?php   if ($target_status == 0)  { ?>
<img src="../icons/off.png" width="130" height="176">
<?php  }  elseif ($target_status == 1) { ?>
<img src="../icons/on.png" width="130" height="176">
<?php  } ?>
  </td>   <td>
  <?php
echo "Επιθυμητός στόχος θερμοκρασίας δωματίου:" . $target_temperature_control;
?>  <hr>
  <?php
echo " Τελευταία θερμοκρασία που στάλθηκε:" . $last_send_temp;
?>
 </td> </tr> <tr>   <td>
 <?php   if ($arduino_status == 0)  {  ?>
<div align="center">Η συσκευή έχει πρόβλημα στην τροφοδοσία</div>
<?php  }  elseif ($arduino_status == 1)  {  ?>
<div align="center">Η συσκευή λειτουργεί</div>
<?php  }  ?>
```

```
  </td>   <td>  <?php      if ($target_status == 0) { ?>
<div align="center">Η μονάδα του κλιματιστικού δεν λειτουργεί.</div>
<?php   } elseif ($target_status == 1) {
?>
<div align="center">Η μονάδα του κλιματιστικού λειτουργεί.</div>
<?php
}
?>
  </td>
  <td>
<?php
echo "Τελευταία καταγραφή θερμοκρασία δωματίου:" . $last_stored_temperature;
?></td>
 </tr>
 <tr>
<td colspan=2>
 Η συσκευή λειτουργεί σε Cool Mode λόγω της συγκεκριμένης εφαρμογής (Server Room) της μονάδας κλιματισμού
 </td>
 <td align="center"><img src="../icons/cool-small.png" width="100" height="100"></td>
 </tr>
</table>


<?php
$conn->close();
?>
 <hr>
 <font size="2">Automatic air condition IOT device (2019-2020)</font>
 </body>
</html>
```

B.2 Κώδικας χειροκίνητης παράκαμψης αυτοματισμού συστήματος

Ο παρακάτω κώδικας χρησιμοποιείται για χειροκίνητη παράκαμψη αυτοματισμού συστήματος από τον χρήστη.

override.php

```
<?php
//************************************
// status of the ac + override by user
// implementation: Patrinos Dimitris
//************************************
$servername = "localhost";
$username = "*******";
$password = " ******* ";
$dbname = " ******* ";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
$sql = "select * from ac_automation ORDER BY id DESC LIMIT 1";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
   // output data of each row
   while($row = $result->fetch_assoc()) {
   //echo "latest stored targeted record set[id: " . $row["id"]. " - target_temperature: " . $row["target_temperature"]. " - status:". $row["status"]."]<br>";
            $target_temperature_control = $row["target_temperature"];
            $target_id = $row["id"];
            $target_status = $row["status"];
            $last_send_temp = $row["last_send_temp"];
```

```
  }
} else {
  //echo "0 results";
}

$sql_ip = "select * from ac_temp_log ORDER BY id DESC LIMIT 1";
$result_ip = $conn->query($sql_ip);

if ($result_ip->num_rows > 0) {
  // output data of each row
  while($row = $result_ip->fetch_assoc()) {
  //echo "last ip [ip address: " . $row["ip"]. "]<br>";
        $stored_ip_address = $row["ip"];
        $last_stored_temperature = $row["Temperature"];
  }
} else {
  //echo "0 results";
}
//************************************************
// put an @ in front of function call to suppress all error messages.
// check if the arduino device is alive, with ping the last ip address
// reference from coding php forum
// https://stackoverflow.com/questions/1239068/ping-site-and-return-result-in-php
$arduino_status = @ping($stored_ip_address, 80, 10);

function ping($host, $port, $timeout) {
 $arduino_status = 3;
 $tB = microtime(true);
 $fP = fSockOpen($host, $port, $errno, $errstr, $timeout);
 if (!$fP) {
        //echo "<br>arduino down check power supply <br>";
        //echo "<br>**************************<br>";
        $arduino_status = 0;
        return $arduino_status;
```

```
            }
 $tA = microtime(true);
            //echo "<br>arduino is running ok<br>";
            //echo "<br>**************************<br>";
            $arduino_status = 1;
            //the ping time in ms
            //return round((($tA - $tB) * 1000), 0)." ms<br>";
            return $arduino_status;
}
//************************************************
?>
<html>
 <head> </head>
 <body>
  <h2>Χειροκίνητη παράκαμψη αυτοματοποιημένου συστήματος</h2>
  <hr>
<table style="width:300px" style="height:700px" border ="1">
  <tr>
    <th>Arduino Status</th>
    <th>AC Last Status</th>
    <th>Ρυθμίσεις Θερμοκρασίας</th>
  </tr>
<tr>
 <td>
<?php
   if ($arduino_status == 0)
{
?>
<img src="../icons/plug-off.png" width="130" height="176">
<?php
            }
elseif ($arduino_status == 1)
{
?>
```

```
<img src="../icons/plug-on.png" width="130" height="176">
<?php
}
?>
</td>
<td>
<?php
if ($target_status == 0 and $arduino_status == 0)
{
?>
<a href="#"><img src="../icons/off-grey.png" width="130" height="176"></a>
<?php
}
elseif ($target_status == 0 and $arduino_status == 1)
{
?>
<a href="http:// **********/dipatrin/android/commands/on.php"><img src="../icons/off.png"
width="130" height="176"></a>
<?php
}
elseif ($target_status == 1 and $arduino_status == 0)
{
?>
<a href="#"><img src="../icons/on-grey.png" width="130" height="176"></a>
<?php
}
elseif ($target_status == 1 and $arduino_status == 1)
{
?>
<a href="http:// **********/dipatrin/android/commands/off.php"><img src="../icons/on.png"
width="130" height="176"></a>
<?php
}
?>
```

```
</td>
  <td rowspan="4">
<?php
  if ($arduino_status == 0)
{
?>
<table border="1">
<tbody>
<tr>
<td height="50" width="50"><a href="#"><img src="../icons/18-off.png" width="50" height="50"></a></td>
<td height="50" width="50"><a href="#"><img src="../icons/19-off.png" width="50" height="50"></a></td>
</tr>
<tr>
<td height="50" width="50"><a href="#"><img src="../icons/20-off.png" width="50" height="50"></a></td>
<td height="50" width="50"><a href="#"><img src="../icons/21-off.png" width="50" height="50"></a></td>
</tr>
<tr>
<td height="50" width="50"><a href="#"><img src="../icons/22-off.png" width="50" height="50"></a></td>
<td height="50" width="50"><a href="#"><img src="../icons/23-off.png" width="50" height="50"></a></td>
</tr>
<tr>
<td height="50" width="50"><a href="#"><img src="../icons/24-off.png" width="50" height="50"></a></td>
<td height="50" width="50"><a href="#"><img src="../icons/25-off.png" width="50" height="50"></a></td>
</tr>
<tr>
```

```
<td height="50" width="50"><a href="#"><img src="../icons/26-off.png" width="50" height="50"></a></td>
<td height="50" width="50"><a href="#"><img src="../icons/27-off.png" width="50" height="50"></a></td>
</tr>
<tr>
<td height="50" width="50"><a href="#"><img src="../icons/28-off.png" width="50" height="50"></a></td>
<td height="50" width="50"><a href="#"><img src="../icons/29-off.png" width="50" height="50"></a></td>
</tr>
</tbody>
</table>
<?php
        }
elseif ($arduino_status == 1)
{
?>
<table border="1">
<tbody>

<tr>
<td height="50" width="50">
<?php
if ($last_send_temp == 18)
{
?>
<a href="http://**********/dipatrin/android/commands/18.php"><img src="../icons/18-pressed-green.png" width="50" height="50"></a>
<?php
}
else
{
?>
```

```
<a href="http:// **********/dipatrin/android/commands/18.php"><img src="../icons/18.png"
width="50" height="50"></a>
<?php
}
?>
</td>

<td height="50" width="50">
<?php
if ($last_send_temp == 19)
{
?>
<a href="http://**********/dipatrin/android/commands/19.php"><img src="../icons/19-
pressed-green.png" width="50" height="50"></a>
<?php
}
else
{
?>
<a href="http:// **********/dipatrin/android/commands/19.php"><img src="../icons/19.png"
width="50" height="50"></a>
<?php
}
?>
</td>
</tr>
<tr>
<td height="50" width="50">
<?php
if ($last_send_temp == 20)
{
?>
<a href="http://**********/dipatrin/android/commands/20.php"><img src="../icons/20-
pressed-green.png" width="50" height="50"></a>
```

```
<?php
}
else
{
?>
<a href="http:// **********/dipatrin/android/commands/20.php"><img src="../icons/20.png"
width="50" height="50"></a>
<?php
}
?>
</td>

<td height="50" width="50">
<?php
if ($last_send_temp == 21)
{
?>
<a href="http://**********/dipatrin/android/commands/21.php"><img src="../icons/21-
pressed-green.png" width="50" height="50"></a>
<?php
}
else
{
?>
<a href="http:// **********/dipatrin/android/commands/21.php"><img src="../icons/21.png"
width="50" height="50"></a>
<?php
}
?>
</td>
</tr>

<tr>
<td height="50" width="50">
```

```
<?php
if ($last_send_temp == 22)
{
?>
<a href="http://**********/dipatrin/android/commands/22.php"><img src="../icons/22-pressed-green.png" width="50" height="50"></a>
<?php
}
else
{
?>
<a href="http://**********/dipatrin/android/commands/22.php"><img src="../icons/22.png" width="50" height="50"></a>
<?php
}
?>
</td>

<td height="50" width="50">
<?php
if ($last_send_temp == 23)
{
?>
<a href="http://**********/dipatrin/android/commands/23.php"><img src="../icons/23-pressed-green.png" width="50" height="50"></a>
<?php
}
else
{
?>
<a href="http://**********/dipatrin/android/commands/23.php"><img src="../icons/23.png" width="50" height="50"></a>
<?php
}
```

```
?>
</td>
</tr>
<tr>
<td height="50" width="50">
<?php
if ($last_send_temp == 24)
{
?>
<a href="http://**********/dipatrin/android/commands/24.php"><img src="../icons/24-pressed-green.png" width="50" height="50"></a>
<?php
}
else
{
?>
<a href="http:// **********/dipatrin/android/commands/24.php"><img src="../icons/24.png" width="50" height="50"></a>
<?php
}
?>
</td>

<td height="50" width="50">
<?php
if ($last_send_temp == 25)
{
?>
<a href="http://**********/dipatrin/android/commands/25.php"><img src="../icons/25-pressed-green.png" width="50" height="50"></a>
<?php
}
else
{
```

```
?>
<a href="http:// **********/dipatrin/android/commands/25.php"><img src="../icons/25.png"
width="50" height="50"></a>
<?php
}
?>
</td>
</tr>

<tr>
<td height="50" width="50">
<?php
if ($last_send_temp == 26)
{
?>
<a href="http://**********/dipatrin/android/commands/26.php"><img src="../icons/26-
pressed-green.png" width="50" height="50"></a>
<?php
}
else
{
?>
<a href="http:// **********/dipatrin/android/commands/26.php"><img src="../icons/26.png"
width="50" height="50"></a>
<?php
}
?>
</td>

<td height="50" width="50">
<?php
if ($last_send_temp == 27)
{
?>
```

```
<a href="http://**********/dipatrin/android/commands/27.php"><img src="../icons/27-pressed-green.png" width="50" height="50"></a>
<?php
}
else
{
?>
<a href="http:// **********/dipatrin/android/commands/27.php"><img src="../icons/27.png" width="50" height="50"></a>
<?php
}
?>
</td>
</tr>

<tr>
<td height="50" width="50">
<?php
if ($last_send_temp == 28)
{
?>
<a href="http://**********/dipatrin/android/commands/28.php"><img src="../icons/28-pressed-green.png" width="50" height="50"></a>
<?php
}
else
{
?>
<a href="http:// **********/dipatrin/android/commands/28.php"><img src="../icons/28.png" width="50" height="50"></a>
<?php
}
?>
</td>
```

```
<td height="50" width="50">
<?php
if ($last_send_temp == 29)
{
?>
<a href="http://**********/dipatrin/android/commands/29.php"><img src="../icons/29-pressed-green.png" width="50" height="50"></a>
<?php
}
else
{
?>
<a href="http:// **********/dipatrin/android/commands/29.php"><img src="../icons/29.png" width="50" height="50"></a>
<?php
}
?>
</td>
</tr>

</tbody>
</table>
<?php
}
?>
</td>
 </tr>
 <tr>
  <td><?php
  if ($arduino_status == 0)
{
?>
<div align="center">Η συσκευή έχει πρόβλημα στην τροφοδοσία</div>
```

```
<?php  }
elseif ($arduino_status == 1)
{ ?>
<div align="center">Η συσκευή λειτουργεί</div>
<?php  }
?> </td>
  <td>Η κατάσταση του A/C και οι θερμοκρασίες είναι ενεργές στον χρήστη</td>
 </tr> <tr>
  <td><?php
        echo "Τελευταία καταγραφή θερμοκρασία δωματίου:" . $last_stored_temperature;
        ?></td>
  <td><?php
  if ($target_status == 0)
{
?>
<div align="center">Η μονάδα του κλιματιστικού δεν λειτουργεί.</div>
<?php  }
elseif ($target_status == 1)
{ ?>
<div align="center">Η μονάδα του κλιματιστικού λειτουργεί.</div>
<?php
}
?></td>
 </tr> <tr>  <td><?php
        echo "<strong>Τελευταία θερμοκρασία που στάλθηκε:" . $last_send_temp."</strong>";
        ?></td>  <td><?php
        echo "Επιθυμητός στόχος θερμοκρασίας δωματίου:" . $target_temperature_control;
        ?></td>
 </tr> </table>
<?php  $conn->close();  ?>
<hr>
 <font size="2">Automatic air condition IOT device (2019-2020)</font>
</body></html>
```

B.3 Κώδικας ορισμού επιθυμητής θερμοκρασίας

Ο παρακάτω κώδικας χρησιμοποιείται για τον ορισμό της επιθυμητής θερμοκρασίας που θα συνεργάζεται στενά με την αυτοματοποιημένη διαδικασία.

temperature_threshold.php

```
<?php
//************************************
// define temperature threshold
// implementation: Patrinos Dimitris
//************************************
$servername = "localhost";
$username = "*******";
$password = " ******* ";
$dbname = " ******* ";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}

$sql = "select * from ac_automation ORDER BY id DESC LIMIT 1";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
   // output data of each row
   while($row = $result->fetch_assoc()) {
   //echo "latest stored targeted record set[id: " . $row["id"]. " - target_temperature: " . $row["target_temperature"]. " - status:". $row["status"]."]<br>";
            $target_temperature_control = $row["target_temperature"];
            $target_id = $row["id"];
            $target_status = $row["status"];
            $last_send_temp = $row["last_send_temp"];
   }
```

```
} else {
  //echo "0 results";
}

$sql_ip = "select * from ac_temp_log ORDER BY id DESC LIMIT 1";
$result_ip = $conn->query($sql_ip);

if ($result_ip->num_rows > 0) {
  // output data of each row
  while($row = $result_ip->fetch_assoc()) {
  //echo "last ip [ip address: " . $row["ip"]. "]<br>";
        $stored_ip_address = $row["ip"];
        $last_stored_temperature = $row["Temperature"];
  }
} else {
  //echo "0 results";
}
?>
<html>
 <head>
 <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.0/jquery.min.js"></script>
 //*****************************************************************************

<!--  START
//*****************************************************************************
// The MIT License (MIT)
Copyright (c) 2013 Anthony Terrien
http://anthonyterrien.com/demo/knob/
Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:
```

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
//**********************************************************************
-->
    <script src="dist/jquery.knob.min.js"></script>
    <script>
      $(function($) {

        $(".knob").knob({
          change : function (value) {
          },
          release : function (value) {
            console.log("release : " + value);
          },
          cancel : function () {
            console.log("cancel : ", this);
          },
          draw : function () {

            // "tron" case
            if(this.$.data('skin') == 'tron') {

              this.cursorExt = 0.3;

              var a = this.arc(this.cv)  // Arc
                , pa                     // Previous arc
```

```
                , r = 1;

            this.g.lineWidth = this.lineWidth;

            if (this.o.displayPrevious) {
              pa = this.arc(this.v);
              this.g.beginPath();
              this.g.strokeStyle = this.pColor;
              this.g.arc(this.xy, this.xy, this.radius - this.lineWidth, pa.s, pa.e, pa.d);
              this.g.stroke();
            }

            this.g.beginPath();
            this.g.strokeStyle = r ? this.o.fgColor : this.fgColor ;
            this.g.arc(this.xy, this.xy, this.radius - this.lineWidth, a.s, a.e, a.d);
            this.g.stroke();

            this.g.lineWidth = 2;
            this.g.beginPath();
            this.g.strokeStyle = this.o.fgColor;
            this.g.arc( this.xy, this.xy, this.radius - this.lineWidth + 1 + this.lineWidth * 2 / 3, 0, 2 *
Math.PI, false);
            this.g.stroke();

            return false;
          }
        }
      });

    });
  </script>

 <!--  END
//*************************************************************************
```

// The MIT License (MIT)

Copyright (c) 2013 Anthony Terrien

http://anthonyterrien.com/demo/knob/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

//**

-->

```
<style>
        body{
          padding: 0;
          margin: 0px 10px;
          font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
          font-weight: 300;
          text-rendering: optimizelegibility;
        }
        p{font-size: 30px; line-height: 30px}
        div.slider{text-align: center; width: 320px; float: left}
        div.slider > p{font-size: 20px}
</style>
```

```
</head>
<body>

<h2>Ορισμός επιθυμητής θερμοκρασίας</h2>
<hr>
<table style="width:350px" style="height:700px" border ="1">
 <tr>
  <th>Επιλογή επιθυμητής θερμοκρασίας</th>
 </tr>
<tr>
<td>
<br>
<div class="slider">
<form action="store_temperature_threshold.php">
<input id="target_temperature" name="target_temperature" class="knob" data-min=18 data-max=29 data-angleOffset=-125 data-angleArc=250 data-fgColor="#F80606" data-rotation="clockwise" value="<?= $target_temperature_control ?>">
<img src="../icons/hot-spa.jpg" width="130" height="170"><input type="submit" value="Αποθήκευση">
</form>
</div>
<br>
<?php
echo "<strong>Επιθυμητός στόχος θερμοκρασίας δωματίου (threshold):" . $target_temperature_control."</strong>";
?>
</td></tr>
<tr><td>
<?php
echo "<font size=2>Τελευταία καταγραφή θερμοκρασία δωματίου:" . $last_stored_temperature."</font>";
?>
<br>
<?php
```

```
echo "<font size=2>Τελευταία θερμοκρασία που στάλθηκε:" . $last_send_temp."</font>";
?>
</td>
</tr>

 </table>

<?php
$conn->close();
?>
 <hr>
 <font size="2">Automatic air condition IOT device (2019-2020)
   </body>
</html>
//*****************************************************************************
//*****************************************************************************
```

store_temperature_threshold.php

```
<?php
// accept data for storing to the database (target_temperature)
// implementation: Patrinos Dimitris
class temperature_automation{
 public $link='';
 function __construct($target_temperature){
$servername = "localhost";
$username = "******";
$password = " ****** ";
$dbname = " ****** ";
$last_temperature;
$status;
  // Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
```

```
   die("Connection failed: " . $conn->connect_error);
}
$sql = "select * from ac_automation ORDER BY id DESC LIMIT 1";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
   // output data of each row
   while($row = $result->fetch_assoc()) {
      echo "latest stored targeted record set: id: " . $row["id"]. " - target_temperature: " .
$row["target_temperature"]. " <br>";
      $last_temperature = $row["last_send_temp"];
      $status = $row["status"];
   }
} else {    echo "0 results"; }
 $this->connect();
 $this->storeInDB($target_temperature,$status,$last_temperature);
 }
 function connect(){
 $this->link = mysqli_connect('localhost','*****','******') or die('Cannot connect to the DB');
 mysqli_select_db($this->link,'datalog') or die('Cannot select the DB');
 }
 function storeInDB($target_temperature,$status,$last_temperature){
 $query = "insert into ac_automation set target_temperature='".$target_temperature."',
status='".$status."',last_send_temp='".$last_temperature."'";
 $result = mysqli_query($this->link,$query) or die('Errant query:  '.$query);
 }

}
if($_GET['target_temperature'] != ''){
 $temperature_automation=new temperature_automation($_GET['target_temperature']);
}
header('Location: /dipatrin/android/pages/temperature_threshold.php');
?>
```

Β.4 Κώδικας γραφήματος θερμοκρασίας με παρέμβαση χρήστη

Ο παρακάτω κώδικας χρησιμοποιείται για την δημιουργία γραφήματος με ημερολογιακό ορισμό από τον χρήστη .

graph_temperature_custom.php

```
<?php
// graphs with custom user definition of dates
// implementation: Patrinos Dimitris
//*****************************************************************************
$servername = "localhost";
$username = "*****";
$password = " ***** ";
$dbname = "v";
$temp_date_1;
$temp_date_2;
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT DATE_FORMAT(date, '%Y-%m-%d') AS date FROM ac_temp_log ORDER BY id ASC
LIMIT 1";
$result1 = $conn->query($sql);

if ($result1->num_rows > 0) {
   // output data of each row
   while($row1 = $result1->fetch_assoc()) {
      //echo "first date : " . $row1["date"]. " <br>";
      $first_date = $row1["date"];
      $temp_date_1 = $first_date;
   }
```

```
} else {
  echo "0 results";
}
//*******************************************************************************
//*******************************************************************************
$sql2 = "SELECT DATE_FORMAT(date, '%Y-%m-%d') AS date FROM ac_temp_log ORDER BY id
DESC LIMIT 1";
$result2 = $conn->query($sql2);
if ($result2->num_rows > 0) {
  // output data of each row
  while($row2 = $result2->fetch_assoc()) {
    //echo "latest date : " . $row2["date"]. " <br>";
    $latest_date = $row2["date"];
    $temp_date_2 = $latest_date;
  }
} else {
  echo "0 results";
}
//*******************************************************************************
//*******************************************************************************

@$start_date = $_GET['start_date'];
@$end_date = $_GET['end_date'];

//echo "<br> start_date:".$start_date."<br>";
//echo "<br> end_date:".$end_date."<br>";
//between '2020-02-23' and '2020-02-25'
$connect = mysqli_connect("localhost", " ***** ", " ***** ", " ***** ");

// testing query. for specific date
$query = 'select Temperature,UNIX_TIMESTAMP(CONCAT_WS(" ", Date)) AS datetime from
ac_temp_log where date between "'.$start_date.'" and "'.$end_date.'"';

//echo "<br> query:".$query."<br>";
```

```
$result = mysqli_query($connect, $query);
$rows = array();
$table = array();

$table['cols'] = array(
 array(
  'label' => 'Date Time',
  'type' => 'datetime'
 ),
 array(
  'label' => 'Temperature (°C)',
  'type' => 'number'
 )
);

while($row = mysqli_fetch_array($result))
{
 $sub_array = array();
 $datetime = explode(".", $row["datetime"]);
 $sub_array[] =  array(
     "v" => 'Date(' . $datetime[0] . '000)'
    );
 $sub_array[] =  array(
     "v" => $row["Temperature"]
    );
 $rows[] =  array(
    "c" => $sub_array
   );
}
$table['rows'] = $rows;
$jsonTable = json_encode($table);
```

```
?>


<html>
 <head>
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
  <script type="text/javascript"
src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>

  <script type="text/javascript">
   google.charts.load('current', {'packages':['corechart']});
   google.charts.setOnLoadCallback(drawChart);
   function drawChart()
   {
    var data = new google.visualization.DataTable(<?php echo $jsonTable; ?>);

    var options = {
     title:'Δεδομένα Αισθητήρα Θερμοκρασίας',
     legend:{position:'bottom'},
     chartArea:{width:'90%', height:'65%'}
    };

    var chart = new google.visualization.LineChart(document.getElementById('line_chart'));

    chart.draw(data, options);
   }

  </script>
  <style>
  .page-wrapper
  {
   width:600px;
   margin:0 auto;
```

```
}
</style>
</head>  <body>
<table border=1>
        <tr>
        <td rowspan="2">
        <a                  href="http://*****/dipatrin/android/pages/graph_4days.php"><img
src="../icons/cost.png" width="80" height="80"></a><hr>
        <a                  href="http://*****/dipatrin/android/pages/graph_rms_current.php"><img
src="../icons/rms.png" width="80" height="80"></a><hr>
        <a                  href="http://*****/dipatrin/android/pages/graph_temperature.php"><img
src="../icons/temperature.png" width="80" height="80"></a>
        </td>
        <td>
        <form action="graph_temperature_custom.php">
        <table>
        <tr>
        <td><font size="2">Start Date</font></td>
        <td><font size="2">End Date</font></td>
        <td></td>
        </tr>
        <tr>
        <td><input type="date" id="start_date" name="start_date"
    value="2020-02-23"
    min="<?php echo $temp_date_1; ?>" max="<?php echo $temp_date_2; ?>"></td>
    <td>
    <input type="date" id="end_date" name="end_date"
    value="2020-02-25"
    min="<?php echo $temp_date_1; ?>" max="<?php echo $temp_date_2; ?>"></td>
    <td><input type="submit" value="View Specific Range"></td>
    </tr></table>
    </form>
        <div class="page-wrapper">
```

```
                    <div id="line_chart" style="height:250px; width:600px"></div>
            </div>
            </td>
            <td rowspan="2">
            <a              href="http://******/dipatrin/android/pages/graph_humid-temp.php"><img
src="../icons/temp-humid.png" width="80" height="80"></a><hr>
            <a                    href="http://*****/dipatrin/android/pages/graph_power.php"><img
src="../icons/real-power.png" width="80" height="80"></a><hr>
            <a    href="http://*****/dipatrin/android/pages/graph_temperature_custom.php"><img
src="../icons/edit-temp-green.png" width="80" height="80"></a>
            </td>
            </tr>
            </table>
 <!-- <hr>
   <font size="2">Automatic air condition IOT device (2019-2020)</font>
 -->
</body></html>
```

B.5 Κώδικας γραφήματος θερμοκρασίας και υγρασίας

Ο παρακάτω κώδικας χρησιμοποιείται για την δημιουργία γραφήματος θερμοκρασίας και υγρασίας.

graph_humid-temp.php

```
<?php
// graph with two different lines, one temperature one humidity
// implementation: Patrinos Dimitris
// ***************************************************************
$connect = mysqli_connect("localhost", " ***** ", " ***** ", " ***** ");

$query = '
SELECT Humidity,
UNIX_TIMESTAMP(CONCAT_WS(" ", Date)) AS datetime,
Temperature
FROM ac_temp_log where date >= "2020-01-21"';
```

```
$result = mysqli_query($connect, $query);
$rows = array();
$table = array();

$table['cols'] = array(
 array(
  'label' => 'Date Time',
  'type' => 'datetime'
 ),
 array(
  'label' => 'Humidity (%)',
  'type' => 'number'
 ),
 array(
  'label' => 'Temperature (°C)',
  'type' => 'number'
 )
);

while($row = mysqli_fetch_array($result))
{
 $sub_array = array();
 $datetime = explode(".", $row["datetime"]);
 $sub_array[] =  array(
     "v" => 'Date(' . $datetime[0] . '000)'
    );
 $sub_array[] =  array(
     "v" => $row["Humidity"]
    );
  $sub_array[] =  array(
     "v" => $row["Temperature"]
    );
 $rows[] =  array(
```

```
    "c" => $sub_array
   );
}
$table['rows'] = $rows;
$jsonTable = json_encode($table);

?>
<html> <head>
 <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
 <script type="text/javascript" src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
 <script type="text/javascript">
  google.charts.load('current', {'packages':['corechart']});
  google.charts.setOnLoadCallback(drawChart);
  function drawChart()
  {
   var data = new google.visualization.DataTable(<?php echo $jsonTable; ?>);

   var options = {
    title:'Συνολικά Δεδομένα Αισθητήρα Θερμοκρασίας & Υγρασίας (Από τις 21/1/2020 έως σήμερα)',
    legend:{position:'bottom'},
    chartArea:{width:'90%', height:'65%'}
   };

   var chart = new google.visualization.LineChart(document.getElementById('line_chart'));

   chart.draw(data, options);
  }
 </script>
 <style>
 .page-wrapper
 {
  width:600px;
```

```
  margin:0 auto;
 }
 </style>
</head>  <body>
        <table border=1>
        <tr><td rowspan="2">
        <a                href="http://*********/dipatrin/android/pages/graph_4days.php"><img
src="../icons/cost.png" width="80" height="80"></a><hr>
        <a      href="http://**********/dipatrin/android/pages/graph_rms_current.php"><img
src="../icons/rms.png" width="80" height="80"></a><hr>
        <a   href="http://**********/dipatrin/android/pages/graph_temperature.php"><img
src="../icons/temperature.png" width="80" height="80"></a>
        </td><td>
        <div class="page-wrapper">

                        <div id="line_chart" style="height:250px; width:600px"></div>
        </div></td>
        <td rowspan="2">
        <a        href="http://**********/dipatrin/android/pages/graph_humid-temp.php"><img
src="../icons/temp-humid-green.png" width="80" height="80"></a><hr>
        <a          href="http://**********/dipatrin/android/pages/graph_power.php"><img
src="../icons/real-power.png" width="80" height="80"></a><hr>
        <a href="http://*******/dipatrin/android/pages/graph_temperature_custom.php"><img
src="../icons/edit-temp.png" width="80" height="80"></a>
        </td>
        </tr>

        </table>
  <hr>
  <font size="2">Automatic air condition IOT device (2019-2020)</font>
 </body>
</html>
```

B.6 Κώδικας γραφήματος ενέργειας

Ο παρακάτω κώδικας χρησιμοποιείται για την δημιουργία γραφήματος ενέργειας.

graph_power.php

```
<?php
// graph for power line
// implementation: Patrinos Dimitris
$connect = mysqli_connect("localhost", "******", "******", "*******");
$query = '
SELECT RealPower,RealPower_1,DATE_FORMAT(timeValue1,"%H:%i:%s") TIMEONLY
FROM power_data LIMIT 447';

$result = mysqli_query($connect, $query);
$rows = array();
$table = array();

$table['cols'] = array(
 array(
  'label' => 'TIMEONLY',
  'type' => 'string'
 ),
 array(
  'label' => 'Cost with continue operation',
  'type' => 'number'
 ),
 array(
  'label' => 'Cost with automation device',
  'type' => 'number'
 )
);

while($row = mysqli_fetch_array($result))
{
 $sub_array = array();
```

```
 //$datetime = explode(".", $row["unitofmeasure"]);
 $sub_array[] =  array(
    "v" => $row["TIMEONLY"]
   );
 $sub_array[] =  array(
    "v" => $row["RealPower"]
   );
  $sub_array[] =  array(
    "v" => $row["RealPower_1"]
   );
 $rows[] =  array(
   "c" => $sub_array
  );
}
$table['rows'] = $rows;
$jsonTable = json_encode($table);

?>
<html>
 <head>
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
  <script                                                                    type="text/javascript"
src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
  <script type="text/javascript">
   google.charts.load('current', {'packages':['corechart']});
   google.charts.setOnLoadCallback(drawChart);
   function drawChart()
   {
    var data = new google.visualization.DataTable(<?php echo $jsonTable; ?>);

    var options = {
     title:'Γραφική διαφορά Ενέργειας (Watt) με και χωρίς την συσκευή 24-hour',
     legend:{position:'bottom'},
     chartArea:{width:'90%', height:'65%'}
```

```
  };

  var chart = new google.visualization.LineChart(document.getElementById('line_chart'));

  chart.draw(data, options);
 }
</script>
<style>
.page-wrapper
{
 width:600px;
 margin:0 auto;
}
</style>
</head>
<body>
        <table border=1>
        <tr>
        <td rowspan="2">
        <a        href="http://**********/dipatrin/android/pages/graph_4days.php"><img
src="../icons/cost.png" width="80" height="80"></a><hr>
        <a   href="http://**********/dipatrin/android/pages/graph_rms_current.php"><img
src="../icons/rms.png" width="80" height="80"></a><hr>
        <a   href="http://**********/dipatrin/android/pages/graph_temperature.php"><img
src="../icons/temperature.png" width="80" height="80"></a>
        </td>

        <td>
                                                                        <font
size=2>Πρώτες 12 ώρες</font>
                                                                         <a
href="http://**********/dipatrin/android/pages/graph_power_part1.php"><img
src="../icons/first-half.png" width="40" height="40"></a>
```

```
           <font
size=2>Επόμενες 12 ώρες</font>
                                                                        <a
href="http://***************/dipatrin/android/pages/graph_power_part2.php"><img
src="../icons/second-half.png" width="40" height="40"></a>

        <div class="page-wrapper">

                <div id="line_chart" style="height:250px; width:600px"></div>
        </div>
        </td>
        <td rowspan="2">
        <a  href="http://***************/dipatrin/android/pages/graph_humid-temp.php"><img
src="../icons/temp-humid.png" width="80" height="80"></a><hr>
        <a         href="http://*************/dipatrin/android/pages/graph_power.php"><img
src="../icons/real-power-green.png" width="80" height="80"></a><hr>
        <a href="http://*******/dipatrin/android/pages/graph_temperature_custom.php"><img
src="../icons/edit-temp.png" width="80" height="80"></a>
        </td>
        </tr>
        </table>
 <hr>
  <font size="2">Automatic air condition IOT device (2019-2020)</font>
 </body>
</html>
```

graph_power_part1.php

```
<?php
// same graph with less data for more details
// implementation : Patrinos Dimitris
$connect = mysqli_connect("localhost", "*********", "***********", "**********");
$query = '
SELECT RealPower,RealPower_1,DATE_FORMAT(timeValue1,"%H:%i:%s") TIMEONLY
FROM power_data LIMIT 224';
```

```
$result = mysqli_query($connect, $query);
$rows = array();
$table = array();

$table['cols'] = array(
 array(
  'label' => 'TIMEONLY',
  'type' => 'string'
 ),
 array(
  'label' => 'Cost with continue operation',
  'type' => 'number'
 ),
 array(
  'label' => 'Cost with automation device',
  'type' => 'number'
 )
);
while($row = mysqli_fetch_array($result))
{
 $sub_array = array();
 //$datetime = explode(".", $row["unitofmeasure"]);
 $sub_array[] =  array(
    "v" => $row["TIMEONLY"]
   );
 $sub_array[] =  array(
    "v" => $row["RealPower"]
   );
  $sub_array[] =  array(
    "v" => $row["RealPower_1"]
   );
 $rows[] =  array(
   "c" => $sub_array
```

```
  );
}
$table['rows'] = $rows;
$jsonTable = json_encode($table);

?>
<html>
 <head>
 <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
 <script type="text/javascript" src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
 <script type="text/javascript">
  google.charts.load('current', {'packages':['corechart']});
  google.charts.setOnLoadCallback(drawChart);
  function drawChart()
  {
   var data = new google.visualization.DataTable(<?php echo $jsonTable; ?>);

   var options = {
    title:'Γραφική διαφορά Ενέργειας (Watt) με και χωρίς την συσκευή [Πρώτες 12 ώρες]',
    legend:{position:'bottom'},
    chartArea:{width:'90%', height:'65%'}
   };

   var chart = new google.visualization.LineChart(document.getElementById('line_chart'));

   chart.draw(data, options);
  }
 </script>
 <style>
 .page-wrapper
 {
  width:600px;
  margin:0 auto;
```

```
}
 </style>
</head>  <body>

        <table border=1>
        <tr>
        <td rowspan="2">
        <a        href="http://**********/dipatrin/android/pages/graph_4days.php"><img src="../icons/cost.png" width="80" height="80"></a><hr>
        <a    href="http://**********/dipatrin/android/pages/graph_rms_current.php"><img src="../icons/rms.png" width="80" height="80"></a><hr>
        <a   href="http://**********/dipatrin/android/pages/graph_temperature.php"><img src="../icons/temperature.png" width="80" height="80"></a>
        </td>

        <td>
                                                                <font size=2>Πρώτες 12 ώρες</font>
                                                                 <a href="http://**********/dipatrin/android/pages/graph_power_part1.php"><img src="../icons/first-half.png" width="40" height="40"></a>

                  <font size=2>Επόμενες 12 ώρες</font>
                                                                <a href="http://**********/dipatrin/android/pages/graph_power_part2.php"><img src="../icons/second-half.png" width="40" height="40"></a>

        <div class="page-wrapper">

                        <div id="line_chart" style="height:250px; width:600px"></div>

        </div>
        </td>
```

```
<td rowspan="2">
<a href="http://**********/dipatrin/android/pages/graph_humid-temp.php"><img src="../icons/temp-humid.png" width="80" height="80"></a><hr>
<a href="http://**********/dipatrin/android/pages/graph_power.php"><img src="../icons/real-power-green.png" width="80" height="80"></a><hr>
<a href="http://******/dipatrin/android/pages/graph_temperature_custom.php"><img src="../icons/edit-temp.png" width="80" height="80"></a>
</td>
</tr>
</table>
<hr>
<font size="2">Automatic air condition IOT device (2019-2020)</font>
</body>
</html>
```

graph_power_part2.php

```
<?php
// same graph with less data for more details
// implementation: Patrinos Dimitris
$connect = mysqli_connect("localhost", "******", "******", "*******");
$query = "
SELECT RealPower,RealPower_1,DATE_FORMAT(timeValue1,'%H:%i:%s') TIMEONLY
FROM power_data where temp_id > 224 LIMIT 223";
$result = mysqli_query($connect, $query);
$rows = array();
$table = array();

$table['cols'] = array(
 array(
  'label' => 'TIMEONLY',
  'type' => 'string'
 ),
 array(
  'label' => 'Cost with continue operation',
```

```
 'type' => 'number'
 ),
 array(
  'label' => 'Cost with automation device',
  'type' => 'number'
 )
);

while($row = mysqli_fetch_array($result))
{
 $sub_array = array();
 //$datetime = explode(".", $row["unitofmeasure"]);
 $sub_array[] =  array(
    "v" => $row["TIMEONLY"]
   );
 $sub_array[] =  array(
    "v" => $row["RealPower"]
   );
  $sub_array[] =  array(
    "v" => $row["RealPower_1"]
   );
 $rows[] =  array(
   "c" => $sub_array
  );
}
$table['rows'] = $rows;
$jsonTable = json_encode($table);
?>
<html>
 <head>
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
  <script                                                                    type="text/javascript"
src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
  <script type="text/javascript">
```

```
  google.charts.load('current', {'packages':['corechart']});
  google.charts.setOnLoadCallback(drawChart);
  function drawChart()
  {
   var data = new google.visualization.DataTable(<?php echo $jsonTable; ?>);

   var options = {
    title:'Γραφική διαφορά Ενέργειας (Watt) με και χωρίς την συσκευή [Επόμενες 12 ώρες]',
    legend:{position:'bottom'},
    chartArea:{width:'90%', height:'65%'}
   };

   var chart = new google.visualization.LineChart(document.getElementById('line_chart'));

   chart.draw(data, options);
  }
 </script>
  <style>
  .page-wrapper
  {
   width:600px;
   margin:0 auto;
  }
  </style>
 </head>  <body>
            <table border=1>
            <tr>
            <td rowspan="2">
            <a              href="http://**********/dipatrin/android/pages/graph_4days.php"><img
src="../icons/cost.png" width="80" height="80"></a><hr>
            <a    href="http://**********/dipatrin/android/pages/graph_rms_current.php"><img
src="../icons/rms.png" width="80" height="80"></a><hr>
            <a    href="http://**********/dipatrin/android/pages/graph_temperature.php"><img
src="../icons/temperature.png" width="80" height="80"></a>
```

```
        </td>

        <td>
        <font size=2>Πρώτες 12 ώρες</font>
<a href="http://********/dipatrin/android/pages/graph_power_part1.php"><img src="../icons/first-half.png" width="40" height="40"></a>
                  <font size=2>Επόμενες 12 ώρες</font>
                                                                    <a href="http://********/dipatrin/android/pages/graph_power_part2.php"><img src="../icons/second-half.png" width="40" height="40"></a>
        <div class="page-wrapper">
                        <div id="line_chart" style="height:250px; width:600px"></div>
        </div>
        </td>

        <td rowspan="2">
        <a href="http://*******/dipatrin/android/pages/graph_humid-temp.php"><img src="../icons/temp-humid.png" width="80" height="80"></a><hr>
        <a href="http://**********/dipatrin/android/pages/graph_power.php"><img src="../icons/real-power-green.png" width="80" height="80"></a><hr>
        <a href="http://******/dipatrin/android/pages/graph_temperature_custom.php"><img src="../icons/edit-temp.png" width="80" height="80"></a>
        </td> </tr>
        </table>
<hr>
 <font size="2">Automatic air condition IOT device (2019-2020)</font>
</body></html>
```

B.7 Κώδικας γραφήματος κόστους

Ο παρακάτω κώδικας χρησιμοποιείται για την δημιουργία γραφήματος κόστους.

graph_4days.php

```
<?php
// graph with cost data
// implementation: Patrinos Dimitris
$connect = mysqli_connect("localhost", "******", "******", "*******");
$query = '
SELECT moneyspent,moneyspent2,DATE_FORMAT(timeValue1,"%H:%i:%s") TIMEONLY
FROM 2days_data LIMIT 447';

$result = mysqli_query($connect, $query);
$rows = array();
$table = array();

$table['cols'] = array(
 array(
  'label' => 'TIMEONLY',
  'type' => 'string'
 ),
 array(
  'label' => 'Cost with continue operation',
  'type' => 'number'
 ),
 array(
  'label' => 'Cost with automation device',
  'type' => 'number'
 )
);

while($row = mysqli_fetch_array($result))
{
 $sub_array = array();
 //$datetime = explode(".", $row["unitofmeasure"]);
 $sub_array[] =  array(
    "v" => $row["TIMEONLY"]
   );
```

```
 $sub_array[] =  array(
    "v" => $row["moneyspent"]
   );
 $sub_array[] =  array(
    "v" => $row["moneyspent2"]
   );
 $rows[] =  array(
   "c" => $sub_array
  );
}
$table['rows'] = $rows;
$jsonTable = json_encode($table);

?>
<html>
 <head>
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
  <script                                                                    type="text/javascript"
src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
  <script type="text/javascript">
   google.charts.load('current', {'packages':['corechart']});
   google.charts.setOnLoadCallback(drawChart);
   function drawChart()
   {
    var data = new google.visualization.DataTable(<?php echo $jsonTable; ?>);

    var options = {
     title:'Γραφική διαφορά πραγματικού κόστους με και χωρίς την συσκευή 24-hour',
     legend:{position:'bottom'},
     chartArea:{width:'90%', height:'65%'}
    };

    var chart = new google.visualization.LineChart(document.getElementById('line_chart'));
```

```
  chart.draw(data, options);
 }
</script>
<style>
.page-wrapper
{
 width:600px;
 margin:0 auto;
}
</style>
</head>
<body>

        <table border=1>
        <tr>
        <td rowspan="2">
        <a                href="http://*******/dipatrin/android/pages/graph_4days.php"><img
src="../icons/cost-green.png" width="80" height="80"></a><hr>
        <a          href="http://*********/dipatrin/android/pages/graph_rms_current.php"><img
src="../icons/rms.png" width="80" height="80"></a><hr>
        <a       href="http://**********/dipatrin/android/pages/graph_temperature.php"><img
src="../icons/temperature.png" width="80" height="80"></a>
        </td>
        <td>
                                                                                <font
size=2>Πρώτες 12 ώρες</font>
                                                                                  <a
href="http://**********/dipatrin/android/pages/graph_4days_part1.php"><img
src="../icons/first-half.png" width="40" height="40"></a>

                   <font
size=2>Επόμενες 12 ώρες</font>
```

```
                                                                    <a
href="http://********/dipatrin/android/pages/graph_4days_part2.php"><img
src="../icons/second-half.png" width="40" height="40"></a>

        <div class="page-wrapper">

                        <div id="line_chart" style="height:250px; width:600px"></div>

        </div>
        </td>

        <td rowspan="2">
        <a        href="http://********/dipatrin/android/pages/graph_humid-temp.php"><img
src="../icons/temp-humid.png" width="80" height="80"></a><hr>
        <a                href="http://*******/dipatrin/android/pages/graph_power.php"><img
src="../icons/real-power.png" width="80" height="80"></a><hr>
        <a href="http://*******/dipatrin/android/pages/graph_temperature_custom.php"><img
src="../icons/edit-temp.png" width="80" height="80"></a>
        </td>
        </tr>
        </table>
<hr>
  <font size="2">Automatic air condition IOT device (2019-2020)</font>
 </body>
</html>
```

graph_4days_part1.php

```
<?php
// graph with cost data with less data for more details
// implementation: Patrinos Dimitris
$connect = mysqli_connect("localhost", "*******", "******", "*******");

$query = '
SELECT moneyspent,moneyspent2,DATE_FORMAT(timeValue1,"%H:%i:%s") TIMEONLY
FROM 2days_data LIMIT 224';
```

```
$result = mysqli_query($connect, $query);
$rows = array();
$table = array();

$table['cols'] = array(
 array(
  'label' => 'TIMEONLY',
  'type' => 'string'
 ),
 array(
  'label' => 'Cost with continue operation',
  'type' => 'number'
 ),
 array(
  'label' => 'Cost with automation device',
  'type' => 'number'
 )
);

while($row = mysqli_fetch_array($result))
{
 $sub_array = array();
 //$datetime = explode(".", $row["unitofmeasure"]);
 $sub_array[] =  array(
    "v" => $row["TIMEONLY"]
   );
 $sub_array[] =  array(
    "v" => $row["moneyspent"]
   );
  $sub_array[] =  array(
    "v" => $row["moneyspent2"]
   );
 $rows[] =  array(
```

```
    "c" => $sub_array
   );
}
$table['rows'] = $rows;
$jsonTable = json_encode($table);

?>
<html> <head>
 <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
 <script type="text/javascript"
src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
 <script type="text/javascript">
  google.charts.load('current', {'packages':['corechart']});
  google.charts.setOnLoadCallback(drawChart);
  function drawChart()
  {
   var data = new google.visualization.DataTable(<?php echo $jsonTable; ?>);

   var options = {
    title:'Γραφική διαφορά πραγματικού κόστους με και χωρίς την συσκευή [Πρώτες 12 ώρες]',
    legend:{position:'bottom'},
    chartArea:{width:'90%', height:'65%'}
   };

   var chart = new google.visualization.LineChart(document.getElementById('line_chart'));

   chart.draw(data, options);
  }
 </script>
 <style>
 .page-wrapper
 {
  width:600px;
  margin:0 auto;
```

```
}
 </style>
</head>
<body>
        <table border=1>
        <tr>
        <td rowspan="2">
        <a href="http://**********/dipatrin/android/pages/graph_4days.php"><img src="../icons/cost-green.png" width="80" height="80"></a><hr>
        <a  href="http://**********/dipatrin/android/pages/graph_rms_current.php"><img src="../icons/rms.png" width="80" height="80"></a><hr>
        <a   href="http://**********/dipatrin/android/pages/graph_temperature.php"><img src="../icons/temperature.png" width="80" height="80"></a>
        </td>

        <td>
                                                                <font size=2>Πρώτες 12 ώρες</font>
                                                                 <a href="http://********/dipatrin/android/pages/graph_4days_part1.php"><img src="../icons/first-half.png" width="40" height="40"></a>

                  <font size=2>Επόμενες 12 ώρες</font>
                                                                <a href="http://********/dipatrin/android/pages/graph_4days_part2.php"><img src="../icons/second-half.png" width="40" height="40"></a>

        <div class="page-wrapper">

                        <div id="line_chart" style="height:250px; width:600px"></div>

        </div>
        </td>
```

```
<td rowspan="2">
<a href="http://******/dipatrin/android/pages/graph_humid-temp.php"><img
src="../icons/temp-humid.png" width="80" height="80"></a><hr>
<a href="http://***************/dipatrin/android/pages/graph_power.php"><img
src="../icons/real-power.png" width="80" height="80"></a><hr>
<a
href="http://********/dipatrin/android/pages/graph_temperature_custom.php"><img
src="../icons/edit-temp.png" width="80" height="80"></a>
</td>
</tr>
</table>
<hr>
<font size="2">Automatic air condition IOT device (2019-2020)</font>
</body>
</html>
```

graph_4days_part2.php

```
<?php
// graph with cost data with less data for more details
// implementation: Patrinos Dimitris
$connect = mysqli_connect("localhost", "*****", "******", "******");

$query2 = "
SELECT moneyspent,moneyspent2,DATE_FORMAT(timeValue1,'%H:%i:%s') TIMEONLY
FROM 4days_data where timeValue1 like '2020-03-07%'
order by TimeValue1 desc LIMIT 223";

$query = "
SELECT moneyspent,moneyspent2,DATE_FORMAT(timeValue1,'%H:%i:%s') TIMEONLY
FROM 2days_data where temp_id > 224 LIMIT 223";

$result = mysqli_query($connect, $query);
$rows = array();
$table = array();
```

```
$table['cols'] = array(
 array(
  'label' => 'TIMEONLY',
  'type' => 'string'
 ),
 array(
  'label' => 'Cost with continue operation',
  'type' => 'number'
 ),
 array(
  'label' => 'Cost with automation device',
  'type' => 'number'
 )
);

while($row = mysqli_fetch_array($result))
{
 $sub_array = array();
 //$datetime = explode(".", $row["unitofmeasure"]);
 $sub_array[] =  array(
    "v" => $row["TIMEONLY"]
   );
 $sub_array[] =  array(
    "v" => $row["moneyspent"]
   );
  $sub_array[] =  array(
    "v" => $row["moneyspent2"]
   );
 $rows[] =  array(
   "c" => $sub_array
  );
}
$table['rows'] = $rows;
```

```
$jsonTable = json_encode($table);

?>
<html>
 <head>
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
  <script type="text/javascript"
src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
  <script type="text/javascript">
   google.charts.load('current', {'packages':['corechart']});
   google.charts.setOnLoadCallback(drawChart);
   function drawChart()
   {
    var data = new google.visualization.DataTable(<?php echo $jsonTable; ?>);

    var options = {
     title:'Γραφική διαφορά πραγματικού κόστους με και χωρίς την συσκευή [Επόμενες 12 ώρες]',
     legend:{position:'bottom'},
     chartArea:{width:'90%', height:'65%'}
    };
    var chart = new google.visualization.LineChart(document.getElementById('line_chart'));
    chart.draw(data, options);
   }
  </script>
  <style>
  .page-wrapper
  {
   width:600px;
   margin:0 auto;
  }
  </style>
 </head>
 <body>
```

```
<table border=1>
<tr>
<td rowspan="2">
<a href="http://***/dipatrin/android/pages/graph_4days.php"><img src="../icons/cost-green.png" width="80" height="80"></a><hr>
<a href="http://*********/dipatrin/android/pages/graph_rms_current.php"><img src="../icons/rms.png" width="80" height="80"></a><hr>
<a href="http://**********/dipatrin/android/pages/graph_temperature.php"><img src="../icons/temperature.png" width="80" height="80"></a>
</td>

<td>
<font size=2>Πρώτες 12 ώρες</font>
<a href="http://********/dipatrin/android/pages/graph_4days_part1.php"><img src="../icons/first-half.png" width="40" height="40"></a>

           <font size=2>Επόμενες 12 ώρες</font>
<a href="http://*******/dipatrin/android/pages/graph_4days_part2.php"><img src="../icons/second-half.png" width="40" height="40"></a>

<div class="page-wrapper">

<div id="line_chart" style="height:250px; width:600px"></div>

</div>
</td>

<td rowspan="2">
<a href="http://*******/dipatrin/android/pages/graph_humid-temp.php"><img src="../icons/temp-humid.png" width="80" height="80"></a><hr>
```

```
        <a href="http://***/dipatrin/android/pages/graph_power.php"><img src="../icons/real-power.png" width="80" height="80"></a><hr>
        <a  href="http://*******dipatrin/android/pages/graph_temperature_custom.php"><img src="../icons/edit-temp.png" width="80" height="80"></a>
        </td>
        </tr>
        </table>
<hr>
  <font size="2">Automatic air condition IOT device (2019-2020)</font>
 </body>
</html>
```

B.8 Κώδικας γραφήματος RMS

Ο παρακάτω κώδικας χρησιμοποιείται για την δημιουργία γραφήματος RMS.

graph_rms_current.php

```
<?php
// graph with rms data
// implementation: Patrinos Dimitris
$connect = mysqli_connect("localhost", "****", "***", "******");

$query = '
SELECT RootMeanSquaredCurrent,
UNIX_TIMESTAMP(CONCAT_WS(" ", timeValue1)) AS datetime
FROM rms_current ORDER BY id DESC LIMIT 420';

$result = mysqli_query($connect, $query);
$rows = array();
$table = array();

$table['cols'] = array(
 array(
  'label' => 'Date Time',
  'type' => 'datetime'
```

```
 ),
 array(
  'label' => 'Root Mean Squared Current (Irms, A)',
  'type' => 'number'
 )
);

while($row = mysqli_fetch_array($result))
{
 $sub_array = array();
 $datetime = explode(".", $row["datetime"]);
 $sub_array[] =  array(
    "v" => 'Date(' . $datetime[0] . '000)'
   );
 $sub_array[] =  array(
    "v" => $row["RootMeanSquaredCurrent"]
   );
 $rows[] =  array(
   "c" => $sub_array
  );
}
$table['rows'] = $rows;
$jsonTable = json_encode($table);

?>
<html> <head>
 <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
 <script                                                                    type="text/javascript"
src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
 <script type="text/javascript">
  google.charts.load('current', {'packages':['corechart']});
  google.charts.setOnLoadCallback(drawChart);
  function drawChart()
  {
```

```
  var data = new google.visualization.DataTable(<?php echo $jsonTable; ?>);

  var options = {
   title:'Συνολικά Δεδομένα 24/3 έως 1/4 RMS Current (Irms, A)',
   legend:{position:'bottom'},
   chartArea:{width:'90%', height:'65%'}
  };

  var chart = new google.visualization.LineChart(document.getElementById('line_chart'));

  chart.draw(data, options);
 }


</script>
<style>
.page-wrapper
{
 width:600px;
 margin:0 auto;
}
</style>
</head>
<body>

        <table border=1>
        <tr>
        <td rowspan="2">
        <a                href="http://*****/dipatrin/android/pages/graph_4days.php"><img
src="../icons/cost.png" width="80" height="80"></a><hr>
        <a                href="http://******/dipatrin/android/pages/graph_rms_current.php"><img
src="../icons/rms-green.png" width="80" height="80"></a><hr>
        <a                href="http://******/dipatrin/android/pages/graph_temperature.php"><img
src="../icons/temperature.png" width="80" height="80"></a>
```

```
        </td>

        <td>
        <div class="page-wrapper">
                        <div id="line_chart" style="height:250px; width:600px"></div>
        </div>
        </td>

        <td rowspan="2">
        <a href="http://******/dipatrin/android/pages/graph_humid-temp.php"><img
src="../icons/temp-humid.png" width="80" height="80"></a><hr>
        <a href="http://*****/dipatrin/android/pages/graph_power.php"><img
src="../icons/real-power.png" width="80" height="80"></a><hr>
        <a href="http://*****/dipatrin/android/pages/graph_temperature_custom.php"><img
src="../icons/edit-temp.png" width="80" height="80"></a>
        </td>
        </tr>
        </table>
<hr>
  <font size="2">Automatic air condition IOT device (2019-2020)</font>
 </body>
</html>
```

B.9 Κώδικας γραφήματος θερμοκρασίας

Ο παρακάτω κώδικας χρησιμοποιείται για την δημιουργία γραφήματος θερμοκρασίας με κώδικα τελικής περιόδου έως σήμερα.

graph_temperature.php

```
<?php
// graph with temperature data
// implementation: Patrinos Dimitris
$connect = mysqli_connect("localhost", "*****", "******", "******");

$query = '
```

```
SELECT Temperature,
UNIX_TIMESTAMP(CONCAT_WS(" ", Date)) AS datetime
FROM ac_temp_log';
// testing query. for specific date

$query22 = 'select Temperature,UNIX_TIMESTAMP(CONCAT_WS(" ", Date)) AS datetime from ac_temp_log where date >= "2020-03-11"';

$result = mysqli_query($connect, $query22);
$rows = array();
$table = array();

$table['cols'] = array(
 array(
  'label' => 'Date Time',
  'type' => 'datetime'
 ),
 array(
  'label' => 'Temperature (°C)',
  'type' => 'number'
 )
);

while($row = mysqli_fetch_array($result))
{
 $sub_array = array();
 $datetime = explode(".", $row["datetime"]);
 $sub_array[] =  array(
    "v" => 'Date(' . $datetime[0] . '000)'
   );
 $sub_array[] =  array(
    "v" => $row["'Temperature"]
   );
 $rows[] =  array(
```

```
    "c" => $sub_array
   );
}
$table['rows'] = $rows;
$jsonTable = json_encode($table);
?>
<html> <head>
 <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
 <script type="text/javascript"
src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
 <script type="text/javascript">
  google.charts.load('current', {'packages':['corechart']});
  google.charts.setOnLoadCallback(drawChart);
  function drawChart()
  {
   var data = new google.visualization.DataTable(<?php echo $jsonTable; ?>);
   var options = {
    title:'Συνολικά Δεδομένα Αισθητήρα Θερμοκρασίας (Από 11 Μαρτίου 2020 έως σήμερα)',
    legend:{position:'bottom'},
    chartArea:{width:'90%', height:'65%'}
   };

   var chart = new google.visualization.LineChart(document.getElementById('line_chart'));
   chart.draw(data, options);
  }

 </script>
 <style>
 .page-wrapper
 {
  width:600px;
  margin:0 auto;
 }
 </style> </head>
```

```
<body> <table border=1>
        <tr> <td rowspan="2">
        <a                  href="http://******/dipatrin/android/pages/graph_4days.php"><img
src="../icons/cost.png" width="80" height="80"></a><hr>
        <a            href="http://*******/dipatrin/android/pages/graph_rms_current.php"><img
src="../icons/rms.png" width="80" height="80"></a><hr>
        <a            href="http://*******/dipatrin/android/pages/graph_temperature.php"><img
src="../icons/temperature-green.png" width="80" height="80"></a>
        </td>
<td>
        <div class="page-wrapper">

                    <div id="line_chart" style="height:250px; width:600px"></div>
        </div>
        </td>

        <td rowspan="2">
        <a            href="http://*******/dipatrin/android/pages/graph_humid-temp.php"><img
src="../icons/temp-humid.png" width="80" height="80"></a>
<hr>
        <a                  href="http://******/dipatrin/android/pages/graph_power.php"><img
src="../icons/real-power.png" width="80" height="80"></a><hr>
        <a  href="http://******/dipatrin/android/pages/graph_temperature_custom.php"><img
src="../icons/edit-temp.png" width="80" height="80"></a>
        </td>
        </tr>
</table>

<hr>
  <font size="2">Automatic air condition IOT device (2019-2020)</font>
 </body>
</html>
```

Β.10 Κώδικας ιστορικό μετρήσεων και εντολών

Ο παρακάτω κώδικας χρησιμοποιείται για την ενημέρωση και παρουσίαση των τελευταίων κινήσεων και καταγραφών του αυτοματισμού.

code.php

```
<html> <head></head>
<body>
<h2>Ιστορικό εντολών</h2>
<h3>Ιστορικό 10 τελευταίων καταμετρήσεων</h3>
 <hr>
<?php
//*******************************************************************************************
// status of the ac and database with history records
// implementation : Patrinos Dimitris
//*******************************************************************************************
$servername = "localhost";
$username = "*****";
$password = "****";
$dbname = "******";
//*******************************************************************************************
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
//*******************************************************************************************
$sql_last_10 = "select * from ac_temp_log ORDER BY id DESC LIMIT 10";
$result_last_10 = $conn->query($sql_last_10);
//echo "<h2>last 10 measurements ac_temp_log table</h2>";
?>
<table border=1>
<tr>
<td>Timestamp</td>
```

```
<td>Record id</td>
<td>Arduino IP</td>
<td>Temp (°C)</td>
<td>Humid (%)</td>
</tr>
<?php
if ($result_last_10->num_rows > 0) {
  // output data of each row
  while($row = $result_last_10->fetch_assoc()) {
  ?>
  <tr>
        <td>
  <?php
  echo  "". $row["Date"]." </td><td>" . $row["id"]. "</td><td>" . $row["ip"]. " </td><td>" .
$row["Temperature"]. " </td><td>". $row["Humidity"]."</td>";
  ?>
  </tr>
  <?php
  }
} else {
  //echo "0 results";
}
?>
  </table>
  <h3>Ιστορικό 10 τελευταίων αυτοματοποιημένων εντολών</h3>
  <hr>
  <?php
//*****************************************************************************************
$sql_last_10_set2 = "select * from ac_automation ORDER BY id DESC LIMIT 10";
$result_last_10_set2 = $conn->query($sql_last_10_set2);
//echo "<h2>last 10 automates commands ac_automation table</h2>";
?>
<table border=1>
<tr>
```

```
<td>Record id</td>
<td>Target Temperature</td>
<td>A/C status</td>
<td>Last Temperature Send</td>
</tr>
<?php
if ($result_last_10_set2->num_rows > 0) {
  // output data of each row
  while($row = $result_last_10_set2->fetch_assoc()) {
  ?>
  <tr>  <td>
  <?php
  // use ternary operator which acts as a shortened IF/Else statement
  echo "" . $row["id"]. " </td><td>" . $row["target_temperature"]. " </td><td>".
(($row["status"]=='1')?'<font color="green">On</font>':'<font color="red">Off</font>'). "
</td><td>". $row["last_send_temp"]. "</td>";
    ?>
  </tr>
  <?php
  }
} else {
  //echo "0 results";
}
//*****************************************************************************************
?>

</table>
<?php
$conn->close();
?>
 <h3>Live στιγμιότυπο κλιματιστικής μονάδας</h3>
 <hr>
 <iframe src="http://*******" width='800px' height='800px' />
```

```
 <font size="2">Automatic air condition IOT device (2019-2020)</font>
 </body>
</html>
```

B.11 Κώδικας About

Ο παρακάτω κώδικας χρησιμοποιείται για την ενημέρωση της χρήσης με και χωρίς την IoT συσκευή .

about.php

```
<?php
//**************************************
// πληροφοριακή σελίδα με αποτελέσματα συγκρίσεων με και χωρίς χρήση αυτοματισμού
// implementation : Patrinos Dimitris
//**************************************
$servername = "localhost";
$username = "*****";
$password = "******";
$dbname = "******";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}

$sql = "select * from ac_automation ORDER BY id DESC LIMIT 1";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
   // output data of each row
   while($row = $result->fetch_assoc()) {
   //echo "latest stored targeted record set[id: " . $row["id"]. " - target_temperature: " . $row["target_temperature"]. " - status:". $row["status"]."]<br>";
```

```
            $target_temperature_control = $row["target_temperature"];
            $target_status = $row["status"];
            $last_send_temp = $row["last_send_temp"];
    }
} else {
    //echo "0 results";
}

$sql_ip = "select * from ac_temp_log ORDER BY id DESC LIMIT 1";
$result_ip = $conn->query($sql_ip);

if ($result_ip->num_rows > 0) {
    // output data of each row
    while($row = $result_ip->fetch_assoc()) {
    //echo "last ip [ip address: " . $row["ip"]. "]<br>";
            $stored_ip_address = $row["ip"];
            $last_stored_temperature = $row["Temperature"];
    }
} else {
    //echo "0 results";
}
// alteration for test purposes
// $stored_ip_address = "10.96.27.18";
//***********************************************
// put an @ in front of function call to suppress all error messages.
// echo @ping($stored_ip_address, 80, 10);
$arduino_status = @ping($stored_ip_address, 80, 10);

function ping($host, $port, $timeout) {
  $arduino_status = 3;
  $tB = microtime(true);
  $fP = fSockOpen($host, $port, $errno, $errstr, $timeout);
  if (!$fP) {
            //echo "<br>arduino down check power supply <br>";
```

```
            //echo "<br>**************************<br>";
            $arduino_status = 0;
            return $arduino_status;
            }
 $tA = microtime(true);
            //echo "<br>arduino is running ok<br>";
            //echo "<br>**************************<br>";
            $arduino_status = 1;
            //the ping time in ms
            //return round((($tA - $tB) * 1000), 0)." ms<br>";
            return $arduino_status;
}

?>


<html>
 <head>


 </head>
 <body>

  <h2>About Section</h2>
  <hr>

   <table style="width:300px" style="height:700px" border ="1">
  <tr>
    <th colspan=3 align="center">Περιληπτικά αποτελέσματα εφαρμογής</th>
  </tr>


<tr>
            <td colspan=3 align="justify">
```

<blockquote>Όπως παρουσιάζεται μέρος των μετρήσεων, στα γραφήματα που ακολουθούν

η αναπτυσσόμενη συσκευή με την αυτοματοποίηση του κλιματιστικού κατάφερε

να λειτουργήσει την μονάδα στα ίδια όρια θερμοκρασίας με κατανάλωση

μικρότερη από το 1/3 από την κατανάλωση που θα απαιτούνταν χωρίς την αυτοματοποιημένη συσκευή.

Οι μετρήσεις αφορούν δυο διαφορετικά 48ώρα χρήσης με την συσκευή και χωρίς την συσκευή και παρουσιάζονται η κατανάλωση σε Watt και κοστολόγηση σε ευρώ με οριζόμενη τιμή kWh 0.115 ευρώ.

</blockquote>

</td>

</tr>

<tr>

<td colspan=3>

<blockquote>Μετρήσεις κόστους σε διάστημα 48ωρών (7/3/2020 – 8/3/2020)
Σύνολο 4,95 ευρώ, συνεχούς λειτουργίας.

</blockquote>

</td>

</tr>

<tr>

<td colspan=3>

<blockquote>Μετρήσεις κόστους σε διάστημα 48ωρών (24/3/2020 – 25/3/2020)
Σύνολο 1,45 ευρώ, αυτοματοποιημένης λειτουργίας.

</blockquote>

```
</td>
</tr>


<tr>
        <td colspan=3>
<img src="../graphs/large/real-power-7-8.png" width="760px">
<blockquote><em>Μετρήσεις Watt σε διάστημα 48ωρών (7/3/2020 – 8/3/2020) <br>Περίπου 43kWh κατανάλωσης με συνθήκες συνεχούς λειτουργίας.</em>
</blockquote>
</td>
</tr>


<tr>
        <td colspan=3>
<img src="../graphs/large/real-power-24-25.png" width="760px">
<blockquote><em>Μετρήσεις Watt σε διάστημα 48ωρών (24/3/2020 – 25/3/2020) <br>Περίπου 12.6kWh κατανάλωσης με συνθήκες αυτοματοποιημένης λειτουργίας.</em>
</blockquote>
</td>
</tr>



</table>


<hr>
<font size="2">Automatic air condition IOT device (2019-2020)</font>
</body>
</html>
```

Β.12 Κώδικας αποθήκευσης και αυτοματισμού

Ο παρακάτω κώδικας στην ουσία είναι ο κυριότερος, αφού περιέχει τον κεντρικό αυτοματισμό και την αποθήκευση στην βάση δεδομένων.

dht.php

```
<?php
// automation control

$servername = "localhost";
$username = "******";
$password = "******";
$dbname = "******";

$temperature = $_GET['temperature'];
$latest_ip = $_GET['ip'];
$humidity = $_GET['humidity'];

$temperature = str_replace(',', '.', $temperature);
$original_value = round($temperature);
$original_value_plus1 = $original_value +1;
$original_value_minus1 = $original_value -3;
// minus 3
$target_temperature_control;
$target_id;
$last_send_temp;

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
```

```
$sql_insert = "insert into ac_temp_log set humidity='".$humidity."', temperature='".$temperature."',
ip='".$latest_ip."'";
$insert_result = $conn->query($sql_insert);

//*******************************************************************************************
$sql = "select * from ac_automation ORDER BY id DESC LIMIT 1";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
  // output data of each row
  while($row = $result->fetch_assoc()) {
    echo "latest stored targeted record set: id: " . $row["id"]. " - target_temperature: " .
$row["target_temperature"]. " <br>";
    $target_temperature_control = $row["target_temperature"];
    $target_id = $row["id"];
    $target_temperature_minus1 = $target_temperature_control -1;
    $last_send_temp = $row["last_send_temp"];
  }
} else {
  echo "0 results";
}
//*******************************************************************************************
//temperature room is equal or less of the target temperature off
if ($original_value == $target_temperature_control || $original_value <
$target_temperature_control)
{
echo "<br>**********************************<br>";
echo "room temperature equal to target temperature room <br>";
echo "redirect to off command <br>";
echo "room temperature: ". $original_value;
echo "<br>target_temperature_control: ". $target_temperature_control;
$automate_command = file_get_contents("http://".$latest_ip."/OFF");
ob_clean();
echo "<br>call page: http://".$latest_ip."/OFF";
```

```
echo "<br>**********************************";
$sql_off = "insert into ac_automation set status = 0, last_send_temp=0, target_temperature='".$target_temperature_control."'";
$temp_result = $conn->query($sql_off);
}
else
{
echo "<br>***************************";
echo "room temperature ". $original_value ." not reached target keep it on";
echo "<br>***************************";

// in case we try to send temperature outside accepted temperatures
if ($original_value_minus1 <18)
{
        $original_value_minus1 = 18;
        echo "<br>***************************";
        echo "<br>*****outside accepted temperatures send temperature: ".$original_value_minus1;
        echo "<br>***************************";
        }
// check room temperature and send minus one command in order to reach target more economical.
switch ($original_value_minus1) {
case 18:
        echo "<br>send 18";
        $automate_command = file_get_contents("http://".$latest_ip."/18");
        echo "<br> http://".$latest_ip."/18";
        ob_clean();
        $sql_on = "insert into ac_automation set status = 1, last_send_temp=18, target_temperature='".$target_temperature_control."'";
        $temp_result = $conn->query($sql_on);
        break;
case 19:
        echo "<br>send 19";
```

```
            $automate_command = file_get_contents("http://".$latest_ip."/19");
            echo "<br> http://".$latest_ip."/19";
            ob_clean();
            $sql_on = "insert into ac_automation set status = 1, last_send_temp=19, target_temperature='".$target_temperature_control."'";
            $temp_result = $conn->query($sql_on);
            break;
case 20:
  echo "<br>send 20";
            $automate_command = file_get_contents("http://".$latest_ip."/20");
            echo "<br> http://".$latest_ip."/20";
            ob_clean();
            $sql_on = "insert into ac_automation set status = 1, last_send_temp=20, target_temperature='".$target_temperature_control."'";
            $temp_result = $conn->query($sql_on);
            break;
case 21:
  echo "<br>send 21";
            $automate_command = file_get_contents("http://".$latest_ip."/21");
            echo "<br> http://".$latest_ip."/21";
            ob_clean();
            $sql_on = "insert into ac_automation set status = 1, last_send_temp=21, target_temperature='".$target_temperature_control."'";
            $temp_result = $conn->query($sql_on);
            break;
case 22:
  echo "<br>send 22";
            $automate_command = file_get_contents("http://".$latest_ip."/22");
            echo "<br> http://".$latest_ip."/22";
            ob_clean();
            $sql_on = "insert into ac_automation set status = 1, last_send_temp=22, target_temperature='".$target_temperature_control."'";
            $temp_result = $conn->query($sql_on);
            break;
```

```
case 23:
  echo "<br>send 23";
        $automate_command = file_get_contents("http://".$latest_ip."/23");
        echo "<br> http://".$latest_ip."/23";
        ob_clean();
        $sql_on = "insert into ac_automation set status = 1, last_send_temp=23, target_temperature='".$target_temperature_control."'";
        $temp_result = $conn->query($sql_on);
        break;
case 24:
  echo "<br>send on";
        $automate_command = file_get_contents("http://".$latest_ip."/ON");
        echo "<br> http://".$latest_ip."/ON";
        ob_clean();
        sleep(2);
        echo "<br>send 24";
        $automate_command = file_get_contents("http://".$latest_ip."/24");
        echo "<br> http://".$latest_ip."/24";
        ob_clean();
        $sql_on = "insert into ac_automation set status = 1, last_send_temp=24, target_temperature='".$target_temperature_control."'";
        $temp_result = $conn->query($sql_on);
        break;
case 25:
  echo "<br>send on";
        $automate_command = file_get_contents("http://".$latest_ip."/ON");
        echo "<br> http://".$latest_ip."/ON";
        ob_clean();
        sleep(2);
        echo "<br>send 25";
        $automate_command = file_get_contents("http://".$latest_ip."/25");
        echo "<br> http://".$latest_ip."/25";
        ob_clean();
```

```
        $sql_on = "insert into ac_automation set status = 1, last_send_temp=25, target_temperature='".$target_temperature_control."'";
        $temp_result = $conn->query($sql_on);
        break;
case 26:
  echo "<br>send on";
        $automate_command = file_get_contents("http://".$latest_ip."/ON");
        echo "<br> http://".$latest_ip."/ON";
        ob_clean();
        sleep(2);
        echo "<br>send 26";
        $automate_command = file_get_contents("http://".$latest_ip."/26");
        echo "<br> http://".$latest_ip."/26";
        ob_clean();
        $sql_on = "insert into ac_automation set status = 1, last_send_temp=26, target_temperature='".$target_temperature_control."'";
        $temp_result = $conn->query($sql_on);
        break;
case 27:
  echo "<br>send on";
        $automate_command = file_get_contents("http://".$latest_ip."/ON");
        echo "<br> http://".$latest_ip."/ON";
        ob_clean();
        sleep(2);
        echo "<br>send 27";
        $automate_command = file_get_contents("http://".$latest_ip."/27");
        echo "<br> http://".$latest_ip."/27";
        ob_clean();
        $sql_on = "insert into ac_automation set status = 1, last_send_temp=27, target_temperature='".$target_temperature_control."'";
        $temp_result = $conn->query($sql_on);
        break;
case 28:
  echo "<br>send on";
```

```
            $automate_command = file_get_contents("http://".$latest_ip."/ON");
            echo "<br> http://".$latest_ip."/ON";
            ob_clean();
            sleep(2);
            echo "<br>send 28";
            $automate_command = file_get_contents("http://".$latest_ip."/28");
            echo "<br> http://".$latest_ip."/28";
            ob_clean();
            $sql_on = "insert into ac_automation set status = 1, last_send_temp=28, target_temperature='".$target_temperature_control."'";
            $temp_result = $conn->query($sql_on);
            break;
case 29:
  echo "<br>send on";
            $automate_command = file_get_contents("http://".$latest_ip."/ON");
            echo "<br> http://".$latest_ip."/ON";
            ob_clean();
            sleep(2);
            echo "<br>send 29";
            $automate_command = file_get_contents("http://".$latest_ip."/29");
            echo "<br> http://".$latest_ip."/29";
            ob_clean();
            $sql_on = "insert into ac_automation set status = 1, last_send_temp=29, target_temperature='".$target_temperature_control."'";
            $temp_result = $conn->query($sql_on);
            break;
case 30:
  echo "<br>send on";
            $automate_command = file_get_contents("http://".$latest_ip."/ON");
            echo "<br> http://".$latest_ip."/ON";
            ob_clean();
            sleep(2);
            echo "<br>send 30";
            $automate_command = file_get_contents("http://".$latest_ip."/30");
```

```
        echo "<br> http://".$latest_ip."/30";
        ob_clean();
        $sql_on = "insert into ac_automation set status = 1, last_send_temp=30, target_temperature='".$target_temperature_control."'";
        $temp_result = $conn->query($sql_on);
        echo "<br>***database statues updated********";
        echo "<br>**********************************";
        break;
default:
  echo "<br>send on";
        $automate_command = file_get_contents("http://".$latest_ip."/ON");
        ob_clean();
        $sql_on = "insert into ac_automation set status = 1, last_send_temp=22, target_temperature='".$target_temperature_control."'";
        $temp_result = $conn->query($sql_on);
} // end of switch case

} // end of else keep auto selecting correct temperature

$conn->close();
?>
```

18.php → 29.php

```
<?php
$servername = "localhost";
$username = "******";
$password = "*****";
$dbname = "******";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
```

```
$sql = "select * from ac_automation ORDER BY id DESC LIMIT 1";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
  // output data of each row
  while($row = $result->fetch_assoc()) {
  //echo "latest stored targeted record set[id: " . $row["id"]. " - target_temperature: " . $row["target_temperature"]. " - status:". $row["status"]."]<br>";
        $target_id = $row["id"];
  }
} else {
  //echo "0 results";
}
$sql_ip = "select * from ac_temp_log ORDER BY id DESC LIMIT 1";
$result_ip = $conn->query($sql_ip);

if ($result_ip->num_rows > 0) {
  // output data of each row
  while($row = $result_ip->fetch_assoc()) {
  //echo "last ip [ip address: " . $row["ip"]. "]<br>";
        $stored_ip_address = $row["ip"];
        }
} else {   //echo "0 results";
}
$sql_update_status = "update ac_automation set status = 1, last_send_temp=18 where id =".$target_id;
$temp_result = $conn->query($sql_update_status);
//$automate_command = file_get_contents("http://".$stored_ip_address."/ON");
//sleep(2);
$automate_command = file_get_contents("http://".$stored_ip_address."/18");
header('Location: /dipatrin/android/pages/override.php');
?>
```

Οι σελίδες 19.php έως 29.php και οι (On.php, Off.php) είναι ίδιες με διαφορά την εντολή που στέλνουν.

Παράρτημα Γ
Κώδικας εφαρμογής Android

Γ.1 Κώδικας AndroidManifest.xml

Κώδικας από το android studio.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.example.myapplication">

  <!--All Access-->
  <uses-permission android:name="android.permission.INTERNET" />

  <application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Automatic A/C device"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme"
```

```
    android:usesCleartextTraffic="true">
    <activity
      android:name=".MainActivity"
      android:label="Automatic A/C device"
      android:theme="@style/AppTheme.NoActionBar">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>

</manifest>
```

Γ.2 Κώδικας GalleryFragment.java

Κώδικας από το android studio.

```
package com.example.myapplication.ui.gallery;

import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;

import androidx.annotation.Nullable;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
```

```
import androidx.lifecycle.Observer;
import androidx.lifecycle.ViewModelProviders;

import com.example.myapplication.R;

public class GalleryFragment extends Fragment {
  public WebView mWebView;

  private GalleryViewModel galleryViewModel;

  public View onCreateView(@NonNull LayoutInflater inflater,
                ViewGroup container, Bundle savedInstanceState) {
    galleryViewModel =
        ViewModelProviders.of(this).get(GalleryViewModel.class);

getActivity().setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);

View v=inflater.inflate(R.layout.fragment_slideshow, container, false);
    mWebView = (WebView) v.findViewById(R.id.webview);
    mWebView.loadUrl("http://*******/dipatrin/android/pages/graph_humidity.php");

    // Enable Javascript
    WebSettings webSettings = mWebView.getSettings();
    webSettings.setJavaScriptEnabled(true);

    // Force links and redirects to open in the WebView instead of in a browser
    mWebView.setWebViewClient(new WebViewClient());

    return v;
      }


}
```

Γ.3 Κώδικας HomeFragment.java

Κώδικας από το android studio.

```
package com.example.myapplication.ui.home;

import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.TextView;

import androidx.annotation.Nullable;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.Observer;
import androidx.lifecycle.ViewModelProviders;

import com.example.myapplication.R;

public class HomeFragment extends Fragment {
  public WebView mWebView;
  private HomeViewModel homeViewModel;

  public View onCreateView(@NonNull LayoutInflater inflater,
                ViewGroup container, Bundle savedInstanceState) {
    homeViewModel =
        ViewModelProviders.of(this).get(HomeViewModel.class);
    getActivity().setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
    View v=inflater.inflate(R.layout.fragment_slideshow, container, false);
    mWebView = (WebView) v.findViewById(R.id.webview);
```

```
        mWebView.loadUrl("http://*******/dipatrin/android/pages/home.php");

        // Enable Javascript
        WebSettings webSettings = mWebView.getSettings();
        webSettings.setJavaScriptEnabled(true);

        // Force links and redirects to open in the WebView instead of in a browser
        mWebView.setWebViewClient(new WebViewClient());

        return v;
    }
}
```

Γ.4 Κώδικας SendFragment.java

Κώδικας από το android studio.

```
package com.example.myapplication.ui.send;

import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.TextView;

import androidx.annotation.Nullable;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.Observer;
import androidx.lifecycle.ViewModelProviders;
```

```
import com.example.myapplication.R;

public class SendFragment extends Fragment {
  public WebView mWebView;
  private SendViewModel sendViewModel;

  public View onCreateView(@NonNull LayoutInflater inflater,
                ViewGroup container, Bundle savedInstanceState) {
    sendViewModel =
        ViewModelProviders.of(this).get(SendViewModel.class);

    getActivity().setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
    View v=inflater.inflate(R.layout.fragment_slideshow, container, false);
    mWebView = (WebView) v.findViewById(R.id.webview);
    mWebView.loadUrl("http://*******/dipatrin/android/pages/temperature_threshold.php");

    // Enable Javascript
    WebSettings webSettings = mWebView.getSettings();
    webSettings.setJavaScriptEnabled(true);

    // Force links and redirects to open in the WebView instead of in a browser
    mWebView.setWebViewClient(new WebViewClient());

    return v;
  }
}
```

Γ.5 Κώδικας ShareFragment.java

Κώδικας από το android studio.

```
package com.example.myapplication.ui.share;

import android.content.pm.ActivityInfo;
import android.os.Bundle;
```

```
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.TextView;

import androidx.annotation.Nullable;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.Observer;
import androidx.lifecycle.ViewModelProviders;

import com.example.myapplication.R;

public class ShareFragment extends Fragment {
  public WebView mWebView;
  private ShareViewModel shareViewModel;

  public View onCreateView(@NonNull LayoutInflater inflater,
                ViewGroup container, Bundle savedInstanceState) {
    shareViewModel =
        ViewModelProviders.of(this).get(ShareViewModel.class);

    getActivity().setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
    View v=inflater.inflate(R.layout.fragment_slideshow, container, false);
    mWebView = (WebView) v.findViewById(R.id.webview);
    mWebView.loadUrl("http://**********/dipatrin/android/pages/about.php");

    // Enable Javascript
    WebSettings webSettings = mWebView.getSettings();
    webSettings.setJavaScriptEnabled(true);
```

```
        // Force links and redirects to open in the WebView instead of in a browser
        mWebView.setWebViewClient(new WebViewClient());

        return v;
    }
}
```

Γ.6 Κώδικας SlideshowFragment.java

Κώδικας από το android studio.

```
package com.example.myapplication.ui.slideshow;

import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.TextView;

import androidx.annotation.Nullable;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.Observer;
import androidx.lifecycle.ViewModelProviders;

import com.example.myapplication.R;

public class SlideshowFragment extends Fragment {
    public WebView mWebView;

    private SlideshowViewModel slideshowViewModel;
```

```
  public View onCreateView(@NonNull LayoutInflater inflater,
                ViewGroup container, Bundle savedInstanceState) {
    slideshowViewModel =
        ViewModelProviders.of(this).get(SlideshowViewModel.class);
    getActivity().setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
    View v=inflater.inflate(R.layout.fragment_slideshow, container, false);
    mWebView = (WebView) v.findViewById(R.id.webview);
    mWebView.loadUrl("http://*********/dipatrin/android/pages/code.php");

    // Enable Javascript
    WebSettings webSettings = mWebView.getSettings();
    webSettings.setJavaScriptEnabled(true);

    // Force links and redirects to open in the WebView instead of in a browser
    mWebView.setWebViewClient(new WebViewClient());

    return v;
  }
}
```

Γ.7 Κώδικας ToolsFragment.java

Κώδικας από το android studio.

```
package com.example.myapplication.ui.tools;

import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebSettings;
import android.webkit.WebView;
```

```
import android.webkit.WebViewClient;
import android.widget.TextView;
import androidx.annotation.Nullable;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.Observer;
import androidx.lifecycle.ViewModelProviders;
import com.example.myapplication.R;

public class ToolsFragment extends Fragment {
   public WebView mWebView;
   private ToolsViewModel toolsViewModel;

   public View onCreateView(@NonNull LayoutInflater inflater,
                    ViewGroup container, Bundle savedInstanceState) {
      toolsViewModel =
           ViewModelProviders.of(this).get(ToolsViewModel.class);

      getActivity().setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

View v=inflater.inflate(R.layout.fragment_slideshow, container, false);
      mWebView = (WebView) v.findViewById(R.id.webview);
      mWebView.loadUrl("http://*******/dipatrin/android/pages/override.php");

      // Enable Javascript
      WebSettings webSettings = mWebView.getSettings();
      webSettings.setJavaScriptEnabled(true);

      // Force links and redirects to open in the WebView instead of in a browser
      mWebView.setWebViewClient(new WebViewClient());
      return v;
   }
}
```

Γ.8 Κώδικας activity_main_drawer.xml

Κώδικας από το android studio.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  tools:showIn="navigation_view">

  <group android:checkableBehavior="single">
    <item
      android:id="@+id/nav_home"
      android:icon="@drawable/ic_menu_send"
      android:title="@string/menu_home" />
    <item
      android:id="@+id/nav_gallery"
      android:icon="@drawable/ic_menu_gallery"
      android:title="Γραφήματα" />
    <item
      android:id="@+id/nav_slideshow"
      android:icon="@drawable/ic_menu_gallery"
      android:title="Ιστορικό" />
    <item
      android:id="@+id/nav_share"
      android:icon="@drawable/ic_menu_share"
      android:title="About" />

  </group>

  <item android:title="Διαχείριση">
    <menu>
      <item
        android:id="@+id/nav_tools"
        android:icon="@drawable/ic_menu_manage"
```

```
            android:title="Απομακρυσμένες εντολές" />

        <item
            android:id="@+id/nav_send"
            android:icon="@drawable/ic_menu_send"
            android:title="Ορισμός στόχου" />
    </menu>
  </item>

</menu>
```

Γ.9 Κώδικας strings.xml

Κώδικας από το android studio.

```
<resources>
  <string name="app_name">My Application</string>
  <string name="navigation_drawer_open">Open navigation drawer</string>
  <string name="navigation_drawer_close">Close navigation drawer</string>
  <string name="nav_header_title">Απομακρυσμένη διαχείριση μονάδας κλιματισμού</string>
  <string name="nav_header_subtitle">dimitrios.patrinos1@st.ouc.ac.cy</string>
  <string name="nav_header_desc">Navigation header</string>
  <string name="action_settings">Settings</string>

  <string name="menu_home">A/C Status Device</string>
  <string name="menu_gallery">Γραφήματα</string>
  <string name="menu_slideshow">Ιστορικό εντολών</string>
  <string name="menu_tools">Απομακρυσμένες εντολές</string>
  <string name="menu_share">About</string>
  <string name="menu_send">Ορισμός στόχου</string>

</resources>
```

Γ.10 Κώδικας MainActivity.java

Κώδικας από το android studio.

```
package com.example.myapplication;

import android.os.Bundle;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.snackbar.Snackbar;
import android.view.View;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;
import com.google.android.material.navigation.NavigationView;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import android.view.Menu;

public class MainActivity extends AppCompatActivity {
private AppBarConfiguration mAppBarConfiguration;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toolbar toolbar = findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    DrawerLayout drawer = findViewById(R.id.drawer_layout);
    NavigationView navigationView = findViewById(R.id.nav_view);
    // Passing each menu ID as a set of Ids because each
    // menu should be considered as top level destinations.
```

```
        mAppBarConfiguration = new AppBarConfiguration.Builder(
                R.id.nav_home, R.id.nav_gallery, R.id.nav_slideshow,
                R.id.nav_tools, R.id.nav_share, R.id.nav_send)
                .setDrawerLayout(drawer)
                .build();
        NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment);
        NavigationUI.setupActionBarWithNavController(this, navController, mAppBarConfiguration);
        NavigationUI.setupWithNavController(navigationView, navController);

    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
    @Override
    public boolean onSupportNavigateUp() {
        NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment);
        return NavigationUI.navigateUp(navController, mAppBarConfiguration)
                || super.onSupportNavigateUp();
    }
}
```

Παράρτημα Δ

Κώδικας Database table schemas

Δ.1 Κώδικας table 1

```
CREATE TABLE `ac_temp_log` (
 `Date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
 `Temperature` float NOT NULL,
 `Humidity` float NOT NULL,
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `ip` varchar(20) DEFAULT NULL,
 PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
```

Δ.2 Κώδικας table 2

```
CREATE TABLE `ac_automation` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `target_temperature` int(11) DEFAULT NULL,
```

```
  `status` int(11) DEFAULT NULL,
  `last_send_temp` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
```

Δ.3 Κώδικας table 3

```
CREATE TABLE `power_data` (
  `TimeValue1` datetime DEFAULT NULL,
  `TimeValue2` datetime DEFAULT NULL,
  `RealPower` decimal(28,10) DEFAULT NULL,
  `TimeValue1_1` datetime DEFAULT NULL,
  `TimeValue2_1` datetime DEFAULT NULL,
  `RealPower_1` decimal(28,10) DEFAULT NULL,
  `UnitOfMeasure` varchar(50) DEFAULT NULL,
  `temp_id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Δ.4 Κώδικας table 4

```
CREATE TABLE `2days_data` (
  `TimeValue1` datetime DEFAULT NULL,
  `TimeValue2` datetime DEFAULT NULL,
  `MoneySpent` decimal(28,10) DEFAULT NULL,
  `TimeValue3` datetime DEFAULT NULL,
  `TimeValue4` datetime DEFAULT NULL,
  `MoneySpent2` decimal(28,10) DEFAULT NULL,
  `UnitOfMeasure` varchar(50) DEFAULT NULL,
  `temp_id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Δ.5 Κώδικας table 5

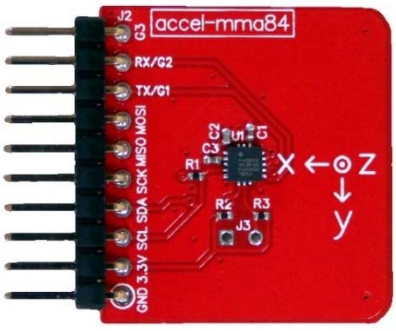
```
CREATE TABLE `rms_current` (
  `TimeValue1` datetime DEFAULT NULL,
  `TimeValue2` datetime DEFAULT NULL,
  `RootMeanSquaredCurrent` decimal(28,10) DEFAULT NULL,
  `UnitOfMeasure` varchar(50) DEFAULT NULL,
  `id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Παράρτημα Ε
Modules από το Board Tessel 2

Ε.1 Tessel 2 modules

Accelerometer

Εντοπισμός προσανατολισμού και κίνησης του board με τη μέτρηση της βαρύτητας / επιτάχυνσης.

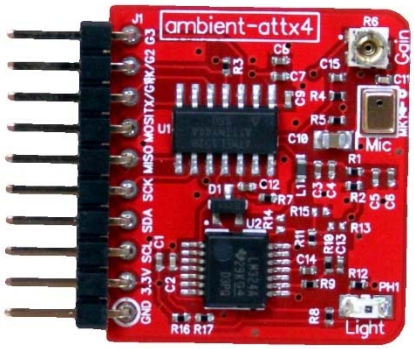


Εικόνα Ε.1 - Tessel 2 **Accelerometer**

Ambient Light+sound

Ο αισθητήρας περιβάλλοντος μπορεί να ανιχνεύσει επίπεδα του περιβάλλοντος φωτός και ήχου. Παράδειγμα εφαρμογής, παλαμάκια για να ενεργοποιήσουμε την τηλεόραση (σε συνδυασμό με

module υπέρυθρων) ή δυνατότητα ενημέρωσης μέσω εφαρμογής ιστού εάν τα φώτα είναι αναμμένα στο σπίτι.



Εικόνα Ε.2 - Tessel 2 Ambient Light+sound

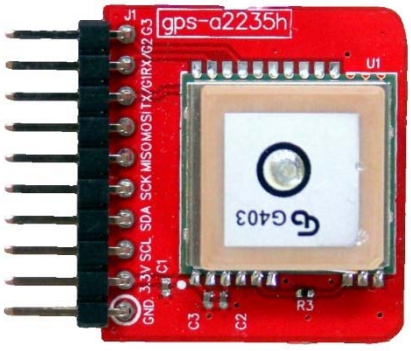
Climate

Μέτρηση υγρασίας και θερμοκρασίας στο περιβάλλον.

Εικόνα Ε.3 - Tessel 2 Climate

GPS

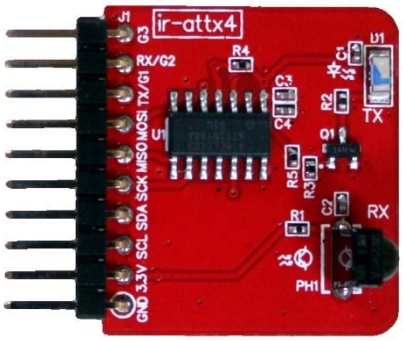
Εντοπισμός παγκόσμιας θέσης.



Εικόνα Ε.4 - Tessel 2 GPS

Infrared

Το υπέρυθρο (IR) Module μπορεί να στείλει και να ανιχνεύει / λαμβάνει σήματα IR



Εικόνα E.5 - Tessel 2 Infrared

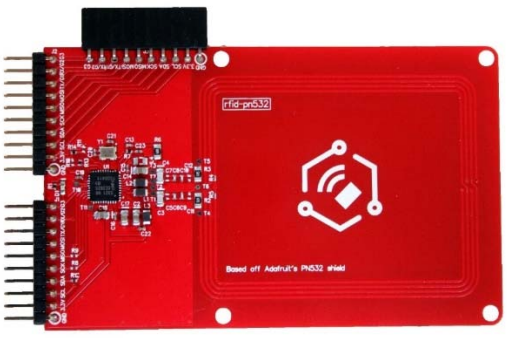
Relay

Έλεγχος ρεύματος σε συσκευές. Παράδειγμα εφαρμογής είναι η έναρξη λειτουργίας καφετιέρας όταν η αντίστοιχη μονάδα περιβάλλοντος (Ambient module) ανιχνεύσει φως.

Εικόνα E.6 - Tessel 2 Relay

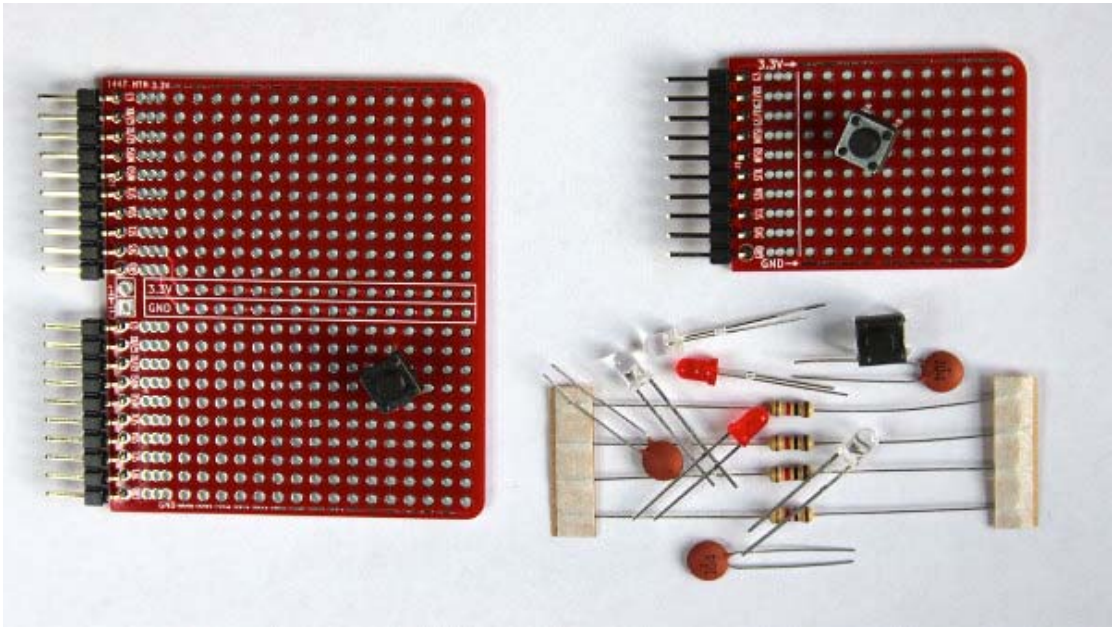
RFID

Διαβάζει RFID κάρτες.



Εικόνα E.7 - Tessel 2 RFID

Το Tessel παρέχει επίσης Prototyping board (ProtoBoard) για να μπορέσει κάποιος να αναπτύξει ατομικά modules.



Εικόνα Ε.8 - Tessel 2 Prototyping board (ProtoBoard)