



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Μελέτη και κατασκευή συστήματος μείωσης της
κατανάλωσης σε μονάδες ψύξης βασιζόμενο σε IoT
τεχνολογία»

Του φοιτητή
Νικόλαου Χατζηλάμπρου
Αρ. Μητρώου: 516159

Επιβλέπουσα
Μαρία Παπαδοπούλου
Βαθμίδα: Επίκουρη Καθηγήτρια

16-06-2023

Τίτλος Δ.Ε.: Μελέτη και κατασκευή συστήματος μείωσης της κατανάλωσης σε μονάδες ψύξης
βασιζόμενο σε ΙοΤ τεχνολογία

Κωδικός Δ.Ε.: 22278

Όνοματεπώνυμο φοιτητή: Νικόλαος Χατζηλάμπρου
Όνοματεπώνυμο εισηγητή/τριας: Μαρία Παπαδοπούλου

Ημερομηνία ανάληψης Δ.Ε.: 22-10-2022

Ημερομηνία περάτωσης Δ.Ε.: 16-06-2023

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Νικόλαου Χατζηλάμπρου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Πρόλογος

Τα τελευταία χρόνια η ανθρωπότητα αντιμετωπίζει ένα από τα δυσκολότερα και πιο σύγχρονα προβλήματα, την περιβαλλοντολογική κρίση. Οι επιστήμονες ψάχνουν πυρετωδώς τρόπους για να μειώσουν το αποτύπωμα που αφήνουμε από τη βιομηχανική επανάσταση μέχρι και σήμερα στο περιβάλλον. Όμως, και ατομικά ο καθένας οφείλει να λάβει τα κατάλληλα μέτρα για να βοηθήσουμε στην επιβίωση του πλανήτη μας. Η συγκεκριμένη πτυχιακή εργασία, με τίτλο «Μελέτη και κατασκευή συστήματος μείωσης της κατανάλωσης σε μονάδες ψύξης βασισμένο σε IoT τεχνολογία», στοχεύει στη σχεδίαση, μελέτη, κατασκευή και βελτιστοποίηση ενός συστήματος ικανού να παρακολουθεί και να ελέγχει τη λειτουργία πολλαπλών μονάδων ψύξης, με σκοπό την ελαχιστοποίηση της καταναλισκόμενης ενέργειας από αυτές.

Περίληψη

Επί δεκαετίες, οι κατασκευαστές συσκευών ψύξης (ψυγεία, καταψύκτες κ.τ.λ.) έχουν θέσει ως υψηλή προτεραιότητα την αύξηση απόδοσης των προϊόντων τους. Πρόκειται για μεγάλες συσκευές που καταναλώνουν τεράστια ποσά ενέργειας, αφήνουν μεγάλο ενεργειακό αποτύπωμα και έχουν σημαντικό κόστος συντήρησης και λειτουργίας. Το τελευταίο διάστημα, και ειδικότερα λόγω του παγκόσμιου προβλήματος της ενεργειακής κρίσης, η ζήτηση για συσκευές ψύξης με χαμηλή κατανάλωση, όχι μόνο από τα απλά νοικοκυριά αλλά κυρίως από τις επιχειρήσεις του κλάδου λιανικής πώλησης, έχει αυξηθεί κατακόρυφα. Επίσης, έρευνες δείχνουν ότι η παραγωγή ηλεκτρισμού στα εργοστάσια παραγωγής ηλεκτρικού ρεύματος καταναλώνει το 54% του συνολικού καυσίμου και σχεδόν το σύνολο του ορυκτού άνθρακα στον πλανήτη. Συνεπάγεται λοιπόν η απελευθέρωση διοξειδίου του άνθρακα και ατμοσφαιρικών ρύπων στην ατμόσφαιρα με τις επακόλουθες επιπτώσεις στην ποιότητα του αέρα και το περιβάλλον. Οι πρακτικές που εφαρμόζονται στην παραγωγή ψυκτικών μονάδων για την επίλυση του προβλήματος αφορούν, κατά κύριο λόγο, την καλύτερη μόνωση και τους συμπιεστές μεταβλητής συχνότητας, λύνοντας το πρόβλημα μόνο σε ένα μικρό ποσοστό.

Η παρούσα πτυχιακή εργασία στοχεύει στη μελέτη και κατασκευή ενός συστήματος αισθητήρων για τον έλεγχο ψυκτικών θαλαμών βασιζόμενος σε IoT τεχνολογία σε έναν επαγγελματικό χώρο με πολλές ψυκτικές μονάδες, όπως ένα super market ή μια βιομηχανία κρεάτων. Ο χρήστης (ο υπεύθυνος ή ο υπάλληλος του καταστήματος) μπορεί να διαβάσει τις ενδείξεις της θερμοκρασίας και υγρασίας του περιβάλλοντος του θαλάμου, καθώς και να ελέγχει χειροκίνητα τη λειτουργία του. Η συσκευή ψύξης ενεργοποιείται και απενεργοποιείται αυτόματα με βάση τη τιμή κατωφλίου που έχει καθοριστεί ανάλογα με τις ανάγκες ψύξης της εκάστοτε μονάδας. Ο χρήστης βλέπει τις μετρήσεις από ένα διακομιστή ιστού ο οποίος “τρέχει” σε έναν επεξεργαστή ESP32, που είναι και ο κεντρικός επεξεργαστής. Οι περιφερειακοί αισθητήρες είναι τοποθετημένοι μέσα στις μονάδες ψύξης και επικοινωνούν ασύρματα με τον κεντρικό επεξεργαστή, στον οποίο στέλνουν τα δεδομένα ανά τακτά χρονικά διαστήματα. Το υπόλοιπο χρονικό διάστημα που οι συσκευές βρίσκονται σε αδράνεια, δηλαδή μπαίνουν σε λειτουργία “ύπνου” για εξοικονόμηση ενέργειας. Με αυτόν τον τρόπο οι ψυκτικές μονάδες λειτουργούν όταν οι συνθήκες στο εσωτερικό τους το απαιτούν, με αποτέλεσμα τη σημαντική μείωση της κατανάλωσης ενέργειας της επιχείρησης.

«STUDY AND CONSTRUCTION OF A SYSTEM TO REDUCE CONSUMPTION IN REFRIGERATION UNITS BASED ON IoT TECHNOLOGY»

«NIKOLAOS CHATZILAMPROU»

Abstract

For the past decades, freezer unit manufacturers (fridges, refrigerators, etc.) have set the performance increase of their products as a high priority. These are sizeable devices that consume a lot of electric energy, leave immense amounts of energy footprint and demand high maintenance and operating costs. During the last months, especially because of the global energy crisis, the demand for low-cost freezer units, not just from households but from businesses as well, has grown exponentially. Also, research has shown that the production of electric energy in power plant stations consumes 54% of the total fuel and almost all the fossil carbon on the planet. Thus, it results in the release of carbon dioxide and atmospheric pollutants into the atmosphere with subsequent effects on air quality and the environment. The practices applied in the production of refrigeration units for solving the problem mainly concern insulation efficiency, using polyurethane foam, and variable speed compressors, solving this issue partly.

The present thesis aims to study and implement a smart sensors system, able to control the cooling mechanism of multiple freezer units, based on the Internet of Things (IoT) in a retail store, such as a supermarket or a meat industry. The user (store manager or an employee) will be able to read the indications of the temperature and humidity inside the unit and manually control its operation. The freezer unit will also automatically activate and deactivate based on the threshold value determined by the temperature requirements of each unit. The user will be able to watch the measurements from a web server which will “run” in an ESP32 micro-controller (main controller). The peripheral sensors will be placed inside the freezer units and will wirelessly communicate to the main controller to whom they will regularly send the sensor data. The sensors will be in a state of inaction, and they will switch to deep sleep mode to conserve power. So, the freezer units will refrigerate only when the conditions demand it, resulting in a significant reduction in the company's energy consumption.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την οικογένειά μου για τη στήριξή τους, καθώς και τους συναδέλφους μου για την πολύτιμη παροχή επιστημονικών γνώσεων. Ειδικές ευχαριστίες στο φίλο και συμφοιτητή Γιώργο, που μου παρείχε τον 3D εκτυπωτή του.

Περιεχόμενα

Πρόλογος.....	iv
Περίληψη	v
Abstract	vi
Ευχαριστίες	vii
Περιεχόμενα	viii
Κατάλογος Σχημάτων	x
Κατάλογος Πινάκων	xi
Συνομογραφίες.....	xi
Κεφάλαιο 1ο: Εισαγωγή.....	1
1.1 Το φαινόμενο του θερμοκηπίου	1
1.2 Η ενεργειακή κρίση	2
1.3 Ο ρόλος των ψυκτικών μονάδων.....	3
1.4 Ανάλυση και συνεισφορά της εργασίας	4
Κεφάλαιο 2ο: Hardware	6
2.1 Το Ηλεκτρονικό Κύκλωμα και η Πλακέτα	6
2.1.1 Εισαγωγή.....	6
2.1.2 Το Ηλεκτρονικό Κύκλωμα.....	6
2.1.3 Η πλακέτα PCB	9
2.2 Ο μικροελεγκτής	12
2.2.1 Εισαγωγή.....	12
2.2.2 Χαρακτηριστικά Στοιχεία	13
2.2.3 Ηλεκτρικά Στοιχεία.....	14
2.2.4 Είσοδοι / Έξοδοι (GPIOs)	15
2.2.5 Λειτουργία ύπνου	16
2.3 Ο αισθητήρας.....	18
2.3.1 Εισαγωγή.....	18
2.3.2 Συνδέσεις.....	18
2.3.3 Χαρακτηριστικά Στοιχεία	19
2.4 Το Κύκλωμα Φόρτισης.....	20
2.4.1 Εισαγωγή.....	20
2.4.2 MCP73831	20
2.4.3 Ανάλυση Κυκλώματος Φόρτισης	22

2.5	Το Κύκλωμα Σταθεροποίησης.....	25
2.5.1	Εισαγωγή.....	25
2.5.2	TPS631000.....	25
2.5.3	Μέθοδοι Λειτουργίας.....	28
2.5.4	Ανάλυση Κυκλώματος.....	29
2.6	Το Κύκλωμα Ρελέ	32
2.6.1	Εισαγωγή.....	32
2.6.2	PE014F03.....	32
2.6.3	Ανάλυση Κυκλώματος.....	33
2.7	Σχεδίαση και Κατασκευή Πλαστικού Καλουπιού.....	40
2.8	Επίλογος.....	41
Κεφάλαιο 3ο:	Software	40
3.1	Εισαγωγή	40
3.2	Ο Κώδικας του Διακομιστή	40
3.2.1	Γενικά.....	40
3.2.2	Η λειτουργία του Web Server.....	42
3.2.3	Επικοινωνία Ιστοσελίδας-Διακομιστή	49
3.2.4	Επικοινωνία Διακομιστή-Αισθητήριων Συσκευών.....	56
3.3	Ο κώδικας των Αισθητήριων Συσκευών	61
3.3.1	Γενικά.....	61
3.3.2	Η επικοινωνία Αισθητήριων Συσκευών.....	62
3.3.3	Επικοινωνία με τις Περιφερειακές Συσκευές.....	66
3.4	Επίλογος.....	71
Κεφάλαιο 4ο:	Συμπεράσματα και προτάσεις βελτίωσης.....	73
4.1	Εισαγωγή	73
4.2	Πειραματικές Μετρήσεις και Συμπεράσματα	73
4.2.1	Κατανάλωση της Συσκευής.....	73
4.2.2	Φόρτιση Μπαταρίας.....	74
4.3	Σύγκριση Πραγματικής Τιμής Τάσης με την Τιμή στην Ιστοσελίδα	75
4.4	Σύγκριση Δισταθούς Ρελέ με Κανονικό Ρελέ	75
4.5	Μέτρηση Διάρκειας Παλμού Ρελέ με Παλμογράφο	76
4.6	Σύγκριση Θερμοκρασίας - Υγρασίας με Δεύτερο Θερμόμετρο.....	77
4.7	Χρήση Θερμικής Κάμερας	78
4.8	Μελλοντικές Επεκτάσεις	80
4.9	Επίλογος.....	81

Βιβλιογραφία.....	82
ΠΑΡΑΡΤΗΜΑ Α: ΚΟΣΤΟΛΟΓΙΟ.....	84

Κατάλογος Σχημάτων

Σχήμα 2.1: Το ηλεκτρονικό κύκλωμα.....	8
Σχήμα 2.2: Η πλακέτα pcb	9
Σχήμα 2.3: Η πλακέτα ασυναρμολόγητη	10
Σχήμα 2.4: Η πλακέτα συναρμολογημένη	11
Σχήμα 2.5: Ο ESP32	12
Σχήμα 2.6: Μπλοκ – διάγραμμα του ESP32.	14
Σχήμα 2.7: ESP32 - Pinout Reference	16
Σχήμα 2.8: Ο DHT22.....	18
Σχήμα 2.9: Το κύκλωμα σύνδεσης DHT22 με τον ESP32.....	20
Σχήμα 2.10: Το Μπλοκ Διάγραμμα του MCP73831	21
Σχήμα 2.11: Το κύκλωμα φορτιστή.....	23
Σχήμα 2.12: Το κύκλωμα επιλογής ρεύματος φόρτισης.....	24
Σχήμα 2.13: Διαιρέτης Τάσης Μπαταρίας	25
Σχήμα 2.14: TPS631000 - Το Μπλοκ Διάγραμμα.....	27
Σχήμα 2.15: Επιλογή λειτουργίας συσκευής από το χρήστη	29
Σχήμα 2.16: Το κύκλωμα σταθεροποίησης της αισθητήριας συσκευής.....	31
Σχήμα 2.17: PE014F03 - Οι δύο καταστάσεις του	33
Σχήμα 2.18: Προσομοίωση του κυκλώματος ρελέ στο PSPICE στην κατάσταση 1.....	36
Σχήμα 2.19: Προσομοίωση του κυκλώματος ρελέ στο PSPICE στη κατάσταση 2.....	37
Σχήμα 2.20: Προσομοίωση του κυκλώματος ρελέ στο PSPICE στη κατάσταση 3.....	38
Σχήμα 2.21: Προσομοίωση του κυκλώματος ρελέ στο PSPICE στη κατάσταση 4.....	39
Σχήμα 2.22: Το πλαστικό καλούπι και η πλακέτα.....	40
Σχήμα 2.23: Η συσκευή από πάνω και από τα πλάγια.....	41
Σχήμα 3.1: Μπλοκ Διάγραμμα της επικοινωνίας Διακομιστή-Συσκευών	42
Σχήμα 3.2: Η ιστοσελίδα χωρισμένη στους “placeholders”	49
Σχήμα 3.3: Ο Web Server	55
Σχήμα 4.1: Οι 4 καταναλώσεις στις 4 διαφορετικές καταστάσεις.....	74
Σχήμα 4.2: Οι δύο καταστάσεις φόρτισης της συσκευής.....	74
Σχήμα 4.3: Σύγκριση τάσης ιστοσελίδας με τάση πολυμέτρου.....	75
Σχήμα 4.4: Η κατανάλωση της συσκευής με ένα κανονικό ρελέ.	76
Σχήμα 4.5: Ο παλμός του ρελέ όταν η μαγνητική επαφή είναι ανοιχτή	76
Σχήμα 4.6: Ο παλμός του ρελέ όταν η μαγνητική επαφή είναι κλειστή	77
Σχήμα 4.7: Σύγκριση μετρήσεων DHT22.....	78
Σχήμα 4.8: Η θερμοκρασία του ESP32 εκτός μονάδας ψύξης.....	79
Σχήμα 4.9: Η μέση θερμοκρασία φόρτισης της συσκευής και η θερμοκρασία του τσιπ φόρτισης.....	79

Κατάλογος Πινάκων

Πίνακας 2.1: ESP32 - Μέγιστες Επιτρεπτές Τιμές.....	14
Πίνακας 2.2: ESP32 - Συνιστώμενες Συνθήκες Λειτουργίας.....	15
Πίνακας 2.3: ESP32 - DC Χαρακτηριστικά.....	15
Πίνακας 2.4: Pinouts του ESP32 που χρησιμοποιούνται.....	16
Πίνακας 2.5: DHT22 - Λειτουργία Εξωτερικών Pin	19
Πίνακας 2.6: DHT22 - Τεχνικά Χαρακτηριστικά.....	19
Πίνακας 2.7: MCP73831 – Ηλεκτρικά Χαρακτηριστικά.....	22
Πίνακας 2.8: MCP73831 - Λειτουργία Εξωτερικών Pin	22
Πίνακας 2.9: Κατάσταση LED2 ανάλογα με την κατάσταση φόρτισης μπαταρίας	23
Πίνακας 2.10: Λειτουργία TPS631000 ανάλογα με τη τάση εισόδου	26
Πίνακας 2.11: TPS631000 - Μέγιστες Επιτρεπτές Τιμές	27
Πίνακας 2.12: TPS631000 - Προτεινόμενες Συνθήκες Λειτουργίας.....	27
Πίνακας 2.13: TPS631000 - Λειτουργία Εξωτερικών Pin.....	28
Πίνακας 2.14: PE014F03 - DC χαρακτηριστικά	33
Πίνακας 2.15: BC807 - Μέγιστες Επιτρεπτές Τιμές.....	34
Πίνακας 2.16: BC807 - DC Χαρακτηριστικά	34
Πίνακας 2.17: BC807 – Μέγιστες Επιτρεπτές Τιμές.....	34
Πίνακας 2.18: BC817 – Ηλεκτρικά Χαρακτηριστικά.....	35
Πίνακας 2.19: BAV99 – Μέγιστες Επιτρεπτές Τιμές.....	35
Πίνακας 2.20: BAV99 - DC Χαρακτηριστικά	35

Συντομογραφίες

ΔΙΠΙΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
IoT	Internet of Things
IEA	International Energy Agency
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
SMD	Surface-Mounted Device
PNP	Positive-Negative - Positive
NPN	Negative-Positive - Negative
GPIO	General Purpose Input-Output
PCB	Printed Circuit Board
RF	Radio Frequency

SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
I2S	Inter-IC Sound
DAC	Digital to Analog Converter
ACD	Analog to Digital Converter
CPU	Central Processing Unit
ROM	Read-Only Memory
SRAM	Static Random Access Memory
WLAN	Wireless Local Area Network
MAC	Media Access Control
BLE	Bluetooth Low Energy
UART	Universal Asynchronous Receiver-Transmitter
WDT	Watchdog Timer
RTC	Real-Time Clock
PWM	Pulse Width Modulation
SDIO	Secure Digital Input Output
IR	InfraRed
NTC	Negative Temperature Coefficient
LCD	Liquid Crystal Display
GND	Ground
DC	Direct Current
LED	Light-Emitting Diode
EMI	Electromagnetic interference
FB	Feedback
IDE	Integrated Development Environment
PFM	Pulse frequency modulation
SSE	Security Service Edge
URL	Uniform Resource Locators
HTTP	Hypertext Transfer Protocol

Κεφάλαιο 1ο: Εισαγωγή

1.1 Το φαινόμενο του θερμοκηπίου

Τα τελευταία 20 έτη, ένα πρόβλημα το οποίο απασχολεί την επιστημονική κοινότητα, τα μέσα μαζικής ενημέρωσης και γενικά την κοινή γνώμη είναι το φαινόμενο του θερμοκηπίου. Το φαινόμενο του θερμοκηπίου είναι ένα φυσικό φαινόμενο, που παρατηρήθηκε για πρώτη φορά από το Γάλλο μαθηματικό και φυσικό Joseph Fourier το 1824, κατά το οποίο η Γη απορροφά την υπέρυθη ακτινοβολία που εκπέμπει καθώς θερμαίνεται από τον Ήλιο. Υπό κανονικές συνθήκες, μεγάλο τμήμα αυτής της ακτινοβολίας φεύγει στο διάστημα, αλλά λόγω ανθρωπίνων δραστηριοτήτων διάφορα αέρια, όπως το διοξείδιο του άνθρακα (CO₂), το μεθάνιο (CH₄) και οι υδροφθοράνθρακες (HFCs) συσσωρεύονται στην ατμόσφαιρα της Γης, με αποτέλεσμα να απορροφούν αυτήν την ακτινοβολία και η θερμότητα που εκπέμπεται να συγκρατείται στην ατμόσφαιρα του πλανήτη. Αυτό έχει ως αποτέλεσμα την αύξηση της θερμοκρασίας στην επιφάνεια της Γης [1], [2].

Όπως αναφέραμε προηγουμένως, μπορεί το φαινόμενο του θερμοκηπίου να είναι ένα φυσικό φαινόμενο, ωστόσο ενισχύεται από την ανθρώπινη δραστηριότητα. Κύριος παράγοντας που ενισχύει το φαινόμενο του θερμοκηπίου είναι η εκπομπή καυσαερίων λόγω της καύσης ορυκτών καυσίμων και η εκπομπή διοξειδίου του άνθρακα που είναι αποτέλεσμα της καύσης αυτής. Είναι γεγονός ότι η χρήση ορυκτών καυσίμων αποτελεί τα 3/4 των εκπομπών διοξειδίου του άνθρακα της ανθρωπότητας και η χρήση αυτών των ορυκτών καυσίμων σε βιομηχανικό επίπεδο ξεκινάει από τη βιομηχανική επανάσταση στα τέλη του 18^{ου} και αρχές του 19^{ου} αιώνα. Είναι γνωστό ότι τα δέντρα και ιδιαίτερα τα δάση έχουν την ιδιότητα να απορροφούν μεγάλες ποσότητες διοξειδίου του άνθρακα που εκλύονται στην ατμόσφαιρα. Επομένως, ένας άλλος σημαντικός παράγοντας για την ενίσχυση του φαινομένου του θερμοκηπίου είναι η αποψίλωση δασών ή μεγάλου μέρους αυτών είτε για τη χρήση του ξύλου, είτε για τη δημιουργία καλλιεργήσιμων εκτάσεων γης που θα χρησιμοποιηθεί από τη βιομηχανία τροφίμων. Ιδιαίτερα, όταν τα δάση καίγονται απελευθερώνουν κατά την καύση διοξείδιο του άνθρακα και έτσι επιβαρύνεται περαιτέρω η κατάσταση. Έρευνες και στατιστικά δείχνουν ότι περίπου 600 εκατομμύρια με 2,6 δισεκατομμύρια τόνοι διοξειδίου του άνθρακα απελευθερώνονται κάθε χρόνο από την καύση ορυκτών καυσίμων και τις πυρκαγιές δασών [3], [4].

Αδιαμφισβήτητα, όσο αυξάνεται ο πληθυσμός των ανθρώπων αυξάνεται και η ανάγκη για τρόφιμα, με αποτέλεσμα η γεωργική και κτηνοτροφική βιομηχανία να χρειάζεται όλο και περισσότερους πόρους για να ανταπεξέλθει στις ανάγκες αυτές. Για αυτό το λόγο, χρησιμοποιούνται λιπάσματα για την καλλιέργεια τροφής καθώς και εντομοαπωθητικά και εντομοκτόνα σε μορφή σπρέι για να κρατήσουν μακριά ανεπιθύμητα έντομα ή τρωκτικά που μπορεί να καταστρέψουν τη σοδειά των γεωργών. Τα λιπάσματα αυτά περιέχουν άζωτο και παράγουν εκπομπές υποξειδίου του αζώτου, ενώ τα εντομοκτόνα φθοριούχα αέρια. Και τα δύο αέρια που αναφέρονται συμβάλλουν στην ενίσχυση του φαινομένου του θερμοκηπίου [3], [4].

Τέλος, μεγάλη ευθύνη έχει και η κτηνοτροφική βιομηχανία, καθώς τα οικόσιτα ζώα που εκτρέφονται με σκοπό να θρέψουν δισεκατομμύρια ανθρώπους εκπέμπουν αέρια όπως το μεθάνιο. Το μεθάνιο είναι το δεύτερο πιο σημαντικό αέριο που συμβάλλει στο φαινόμενο του θερμοκηπίου μετά το διοξείδιο του άνθρακα. Αυτές οι εκπομπές μεθανίου σχετίζονται άμεσα με τα ζώα καθώς παράγονται από εντερική ζύμωση τροφής από βακτήρια και άλλα μικρόβια στο πεπτικό σύστημα των ζώων. Σύμφωνα με τελευταίες μελέτες τα οικόσιτα ζώα όπως αγελάδες, πρόβατα, κατσίκες και χοίροι

ευθύνονται για περίπου το 1/4 των εκπομπών μεθανίου που ανέρχονται συνολικά σε περίπου 100 εκατομμύρια τόνους ετησίως [3], [4].

Το φαινόμενο του θερμοκηπίου, είναι ένα σύγχρονο πρόβλημα του οποίου η επίλυση είναι υψίστης προτεραιότητας, καθώς οι συνέπειες που προκύπτουν μπορεί να αποβούν καταστροφικές σε πολλούς τομείς, περιβαλλοντολογικούς, οικονομικούς, πολιτικούς και κοινωνικούς. Κάποιοι από αυτούς είναι η αύξηση της θερμοκρασίας του πλανήτη, η ρύπανση της ατμόσφαιρας και του αέρα που αναπνέουμε από τους επικίνδυνους ρύπους που εκπέμπονται από την καύση ορυκτών καυσίμων, το λιώσιμο των πάγων στο Βόρειο Πόλο και την Ανταρκτική και η άνοδος της στάθμης των θαλασσών και ωκεανών. Το τελευταίο έχει ως συνέπεια πολλές παραθαλάσσιες πόλεις σε χαμηλό υψόμετρο όπως διαφορές περιοχές της Ολλανδίας και ευρωπαϊκές πόλεις όπως η Βενετία στην Ιταλία, σε λίγες δεκαετίες να εξαφανιστούν κάτω από το νερό.

1.2 Η ενεργειακή κρίση

Η ενεργειακή κρίση είναι ένα φαινόμενο που απασχολεί την ανθρωπότητα τα τελευταία χρόνια, κυρίως λόγω της μεγάλης ζήτησης σε ηλεκτρική ενέργεια. Ειδικότερα, τα τελευταία 3 χρόνια από το 2019 μέχρι και σήμερα η κρίση αυτή έχει ενταθεί περισσότερο λόγω της πανδημίας του COVID-19 και της εισβολής της Ρωσίας στην Ουκρανία το Φεβρουάριο του 2022, καθώς η αντίθεσή τους στην εισβολή και οι οικονομικοί και πολιτικοί περιορισμοί που έθεσε η Ευρωπαϊκή Ένωση στη Ρωσία ως αντίδραση προώθησε τους Ρώσους να μειώσουν τις προμήθειες φυσικού αερίου σε ποσοστά μεγαλύτερα του 80%.

Βέβαια, η τιμή του φυσικού αερίου άρχισε να ανεβαίνει και πριν από την εισβολή αυτή, την περίοδο που σταδιακά τέλειωναν τα lockdowns και άρχισαν να αίρονται τα μέτρα που είχαν παρθεί από τις κυβερνήσεις παγκοσμίως κατά του κορωνοϊού, καθώς οι επιχειρήσεις και τα μεγάλα εργοστάσια προκειμένου να αναπληρώσουν τις απώλειες των προηγούμενων 2 ετών χρησιμοποιούσαν περισσότερη ενέργεια από αυτή που χρησιμοποιούσαν υπό κανονικές συνθήκες, προ πανδημίας. Είναι γεγονός, ότι οι τιμές του πετρελαίου, φυσικού αερίου και ηλεκτρικής ενέργειας έχουν εκτοξευτεί σε ποσά ρεκόρ και αυτή η κρίση πλήττει ιδιαίτερα την Ευρώπη, καθώς μεγάλο ποσοστό αυτών των αγαθών δεν παράγεται, αλλά εισάγεται από τη Ρωσία. Αυτό έχει ως αποτέλεσμα, πολλά νοικοκυριά και επιχειρήσεις σε όλο τον κόσμο να βλέπουν τους λογαριασμούς φυσικού αερίου και ηλεκτρισμού να εκτοξεύονται και να μην μπορούν να ανταπεξέλθουν οικονομικά [5], [6], [7].

Σύμφωνα με τον Παγκόσμιο Οργανισμό Ενέργειας (IEA), το 2022, περίπου 75 εκατομμύρια άνθρωποι και χιλιάδες νοικοκυριά σε ανεπτυγμένες και αναπτυσσόμενες χώρες, δεν μπορούν να πληρώσουν τον πάροχο ηλεκτρικής ενέργειας και 100 εκατομμύρια άνθρωποι μπορεί να μην έχουν πλέον τη δυνατότητα να χρησιμοποιούν υγραέριο για βασικές ανάγκες, όπως το μαγείρεμα και μπορεί να χρειαστεί να χρησιμοποιούν άλλους τρόπους όπως καυσόξυλα. Μπορεί το πρόβλημα της ενεργειακής κρίσης να είναι ένα σύγχρονο και σχετικά καινούργιο φαινόμενο που προέκυψε υπό πολύ συγκεκριμένες καταστάσεις, αυτό όμως δεν σημαίνει ότι είναι και παροδικό.[9] Μελέτες δείχνουν ότι η αυξανόμενη ζήτηση για ενεργειακές υπηρεσίες, θα αποτελέσει πρωτογενή παράγοντα για την συνεχώς αυξανόμενη χρήση ηλεκτρικής ενέργειας με μέσο όρο περίπου 2,8% ετησίως έως το 2050. Ταυτόχρονα η αύξηση του παγκόσμιου πληθυσμού σε περίπου 10 δισεκατομμύρια μέχρι το 2050 κάνει πιο ρεαλιστικό το σενάριο αυτό. Αυτές λοιπόν οι οικονομικές και δημογραφικές αλλαγές που θα γίνουν στα επόμενα 25 χρόνια, αποτελούν σημαντικό παράγοντα για τη μεγιστοποίηση του προβλήματος της ενεργειακής κρίσης παγκοσμίως. [6]

Για την καταπολέμηση του προβλήματος, τα σημαντικότερα μέτρα που προβλέπονται από το Ευρωπαϊκό Συμβούλιο είναι η διασφάλιση εισφοράς αλληλεγγύης από τις επιχειρήσεις και τα εργοστάσια ορυκτών καυσίμων, οι οποίες έχουν διασφαλίσει επιπλέον έσοδα από τις υψηλές χρηματικές τιμές ενέργειας και μείωση χρήσης της ηλεκτρικής ενέργειας. Ειδικά το τελευταίο μέτρο είναι πάρα πολύ σημαντικό καθώς μπορεί να αποβεί καταλυτικός παράγοντας και το μεγαλύτερο όπλο για να καταφέρουμε ως ανθρωπότητα να περιορίσουμε και αργότερα να καταπολεμήσουμε αποδοτικά αυτό το φαινόμενο. Αυτό μπορεί να επιτευχθεί με εθελοντικά μέτρα που θα παρθούν από νοικοκυριά και επιχειρήσεις για τη μείωση ηλεκτρικής ενέργειας κατά 10% μέχρι τα μέσα του 2023 και υποχρέωση των ευρωπαϊκών χωρών να μειώσουν την κατανάλωση ηλεκτρικής ενέργειας, τουλάχιστον 5% σε ώρες αιχμής [8].

1.3 Ο ρόλος των ψυκτικών μονάδων

Ψυγείο ή ψυκτική μονάδα είναι ένα μέσο αποθήκευσης προϊόντων, συνήθως τροφίμων, τα οποία πρέπει να διατηρούνται σε χαμηλές θερμοκρασίες και όχι σε θερμοκρασία δωματίου γιατί αλλιώς θα χαλάσουν. Συνηθέστερα, οι ψυκτικές μονάδες χρησιμοποιούνται από τα νοικοκυριά, τις βιομηχανίες ή τις επιχειρήσεις εστίασης ως μέσο διατήρησης τροφίμων και ποτών. Ένα κλασικό οικιακό ψυγείο αποτελείται από 2 κύριους χώρους τη συντήρηση και την κατάψυξη. Στη συντήρηση αποθηκεύονται προϊόντα και τρόφιμα που προορίζεται να καταναλωθούν εντός λίγων εβδομάδων και υπάρχει χαμηλή θερμοκρασία, αλλά όχι κάτω από μηδέν βαθμούς Κελσίου (συνήθως 3-5°C). Στην κατάψυξη συντηρούνται τρόφιμα τα οποία μπορούν να καταναλωθούν μετά από μήνες και στο εσωτερικό τους υπάρχουν πολύ χαμηλότερες θερμοκρασίες (περίπου -16 έως -20°C). οι πρώτες ψυκτικές μονάδες λειτουργούσαν με πάγο ενώ τα πρώτα ηλεκτρικά ψυγεία κατασκευάστηκαν στα τέλη της δεκαετίας του '20 και λειτουργούσαν με φρέον. Το φρέον πλέον έχει απαγορευθεί καθώς αποδείχθηκε ότι παίζει σημαντικό ρόλο στη δημιουργία της τρύπας του όζοντος. Τα ψυγεία λειτουργούν με τη χρήση ενός κυκλώματος αερίου, που κυκλοφορεί σε μια σειρά από σωληνώσεις συμπιέζεται και επεκτείνεται μέσω ενός συμπιεστή. Καθώς το αέριο που κυκλοφορεί στις σωλίνες συμπιέζεται, απορροφά θερμότητα μειώνοντας έτσι τη θερμοκρασία στο εσωτερικό του ψυγείου. Ο συμπιεστής ρυθμίζεται ανάλογα με τη θερμοκρασία στο εσωτερικό του ψυγείου, η οποία με τη σειρά της παρακολουθείται από έναν θερμοστάτη. Ταυτόχρονα, η μόνωση του ψυγείου διατηρεί τη θερμοκρασία στο εσωτερικό του σταθερή [9][10].

Ένα από τα σημαντικότερα χαρακτηριστικά ενός ψυγείου είναι η ενεργειακή απόδοση του. Η απόδοση ενός ψυγείου βασίζεται στην ενέργεια που καταναλώνεται ετησίως ως προς τον όγκο της μονάδας και εκφράζεται σε όγκο ψύξης ανά μονάδα ηλεκτρικής ενέργειας ανά ημέρα. Ο όγκος μετριέται σε κυβικά μέτρα ενώ η ηλεκτρική ενέργεια που καταναλώνεται σε κιλοβατώρες. Την ενεργειακή απόδοση μπορούμε εύκολα να τη διακρίνουμε από τη χαρακτηριστική ετικέτα η οποία είναι κολλημένη πάνω σε κάθε ψυκτική μονάδα και παρέχει μια σαφή ένδειξη της ενεργειακής απόδοσης της. Επίσης ένα σημαντικό χαρακτηριστικό που επηρεάζει την ενεργειακή απόδοση του θαλάμου είναι η εσωτερική θερμοκρασία που έχει ορίσει ο χρήστης. Η βέλτιστη θερμοκρασία στο εσωτερικό ενός ψυγείου είναι 5°C ενώ στο εσωτερικό μιας κατάψυξης είναι -18°C. Επομένως, η σωστή ρύθμιση, καθώς και η συνεχής παρακολούθηση της θερμοκρασίας από το χρήστη είναι υψίστης σημασίας για την αποδοτικότερη λειτουργία της ψυκτικής μονάδας [11].

Επιπρόσθετα, η σωστή μόνωση μιας θαλάμης είναι ένα χαρακτηριστικό που επηρεάζει τη λειτουργία της. Ο συνηθέστερος τρόπος μόνωσης του εσωτερικού της μονάδας είναι μέσω της χρήσης

κάποιου μονωτικού υλικού, συνήθως κατασκευασμένου από αφρό υαλοβάμβακα. Αυτή η μόνωση βοηθά στην παγίδευση του κρύου αέρα που παράγεται από το σύστημα ψύξης μέσα στο ψυγείο, εμποδίζοντας τον να διαφύγει στο ζεστό περιβάλλον. Επιπλέον, η πόρτα του ψυγείου σφραγίζεται καλά για να αποτρέψει την είσοδο θερμού αέρα και τη διαφυγή κρύου αέρα διατηρώντας περαιτέρω σταθερή τη θερμοκρασία. Ένας άλλος αρκετά διαδεδομένος τρόπος μόνωσης της θερμοκρασίας στο εσωτερικό της μονάδας, είναι οι συμπιεστές μεταβλητής συχνότητας γνωστοί και ως συμπιεστές inverter, οι οποίοι προσαρμόζουν την ταχύτητα του κινητήρα του συμπιεστή ώστε να ταιριάζει κατάλληλα με τη ζήτηση ψύξης του ψυγείου. Οι παραδοσιακή συμπιεστές λειτουργούν με σταθερή ταχύτητα γεγονός που μπορεί να οδηγήσει σε υπερβολική κατανάλωση ενέργειας και διακυμάνσεις της θερμοκρασίας στο εσωτερικό της μονάδας. Αντίθετα, ένα συμπιεστής μεταβλητής συχνότητας χρησιμοποιεί έναν αισθητήρα θερμοκρασίας για να ανιχνεύσει τη θερμοκρασία στο εσωτερικό του θαλάμου και να προσαρμόσει ανάλογα την ταχύτητα του κινητήρα. Με αυτόν τον τρόπο ο συμπιεστής λειτουργεί σε χαμηλότερη ταχύτητα όταν απαιτείται λιγότερη ψύξη, γεγονός που εξοικονομεί ενέργεια και μειώνει τη φθορά του συμπιεστή [11].

Η καλή μόνωση και η σφιχτή σφράγιση της πόρτας είναι υψίστης σημασίας για τη διατήρηση της κατάλληλης θερμοκρασίας, καθώς μόνο έτσι προλαμβάνεται αποτελεσματικά η αλλοίωση των αποθηκευμένων τροφίμων και ποτών και αποφεύγεται η κατανάλωση περισσότερης ηλεκτρικής ενέργειας από αυτή που χρειάζεται στην πραγματικότητα η μονάδα για να λειτουργήσει. Ταυτόχρονα καθένας από εμάς, πρέπει να προλαμβάνει και να μην κρατάει την πόρτα για πολύ ώρα ανοιχτή καθώς κάθε φορά που η πόρτα της μονάδας ανοίγει ο κρύος αέρας ξεφεύγει και εισέρχεται ζεστός ατμοσφαιρικός αέρας από το περιβάλλον. Ως συνέπεια, για να αντισταθμιστεί οι αύξηση της θερμοκρασίας στο εσωτερικό της, η ψυκτική μονάδα πρέπει να καταναλώσει περισσότερη ενέργεια για μειώσει ξανά τη θερμοκρασία στο επιθυμητό.

1.4 Ανάλυση και συνεισφορά της εργασίας

Όπως αναφέραμε και στο κεφάλαιο 1.3 υπάρχει πληθώρα εφαρμογών για τη βέλτιστη ενεργειακή απόδοση των ψυκτικών θαλάμων, αλλά οι σημαντικότερες και πιο κρίσιμες είναι η κατάλληλη ρύθμιση και συνεχής παρακολούθηση της θερμοκρασίας στο εσωτερικό τους, καθώς και η κατάσταση της πόρτας, αν δηλαδή είναι για πολύ ώρα ανοιχτή. Η συγκεκριμένη πτυχιακή εργασία στοχεύει στη χρήση μιας σχετικά καινούργιας τεχνολογίας, το Διαδίκτυο των Πραγμάτων (Internet of Things ή εν συντομία IoT) για την παρακολούθηση και ρύθμιση της θερμοκρασίας από έναν χρήστη, τη λειτουργία αυτής της ψυκτικής μονάδας αυτομάτως, ανάλογα με την επιλογή της θερμοκρασίας κατωφλίου, καθώς και τη χειροκίνητη απενεργοποίηση της σε περίπτωση που ο χρήστης το επιθυμεί. Το σύστημα το οποίο έχει κατασκευαστεί αποτελείται από 3 αισθητήριες συσκευές, οι οποίες θα είναι εγκατεστημένες μέσα στην ψυκτική μονάδα και θα λαμβάνουν μετρήσεις θερμοκρασίας και υγρασίας και από έναν διακομιστή ο οποίος θα φιλοξενεί έναν web server. Οι αισθητήριες συσκευές μαζί με το διακομιστή θα επικοινωνούν μεταξύ τους μέσω ενός πρωτοκόλλου επικοινωνίας.

Κάθε μία από τις αισθητήριες συσκευές θα έχει ως πυρήνα έναν επεξεργαστή ESP32, ο οποίος θα κάνει όλη την επεξεργασία των δεδομένων και θα ρυθμίζει την επικοινωνία μεταξύ θερμοστάτη και διακομιστή. Οι γλώσσες προγραμματισμού που έχουν επιλεγεί για τη σύνταξη του προγράμματος στον επεξεργαστή είναι C++ καθώς και JavaScript, HTML και CSS για τη σύνταξη του web server και το προγραμματιστικό περιβάλλον για τη συγγραφή του κώδικα είναι το Arduino IDE (έκδοση 2.0.4). Να σημειωθεί ότι έγινε επίσης χρήση των πλατφορμών Visual Studio (έκδοση 1.74) και CodeLite (έκδοση

17.2.0) για τη σύνταξη κάποιων κομματιών του κώδικα σε C++, καθώς και της διαδικτυακής πλατφόρμας W3Schools για τις απαιτούμενες δοκιμές που έπρεπε να γίνουν στη σύνταξη του κώδικα για τη λειτουργία και την εμφάνιση του web server. Οι αισθητήριες συσκευές τροφοδοτούνται η κάθε μια από μία μπαταρία λιθίου 18650 3.7 V. Ο σχεδιασμός του κυκλώματος και της πλακέτας έγινε στη διαδικτυακή πλατφόρμα EasyEDA και η πλακέτα κατασκευάστηκε από τον κατασκευαστή JLCPCB. Η πλακέτα αποτελείται από 3 υποκυκλώματα. Το πρώτο υποκύκλωμα είναι ο φορτιστής της μπαταρίας 18650 και χρησιμεύει στη φόρτιση της μπαταρίας, από το σταθεροποιητή ο οποίος προμηθεύει στον επεξεργαστή σταθερή τάση 3.3V και το υποκύκλωμα ρελέ για τον έλεγχο της εξόδου της ψυκτικής μονάδας. Κάθε συσκευή θα μπαίνει σε λειτουργία ύπνου για όσο χρονικό διάστημα δεν στέλνει δεδομένα στο διακομιστή για την εξοικονόμηση ενέργειας.

Μέσω του web server, ο χρήστης θα μπορεί να παρακολουθεί τη θερμοκρασία στο εσωτερικό του ψυγείου σε βαθμούς Κελσίου και Fahrenheit, να ρυθμίζει τη θερμοκρασία κατωφλίου ανάλογα με τις απαιτήσεις της εκάστοτε μονάδας, να την απενεργοποιεί σε περίπτωση που το επιθυμήσει καθώς και να βλέπει διάφορα status για τη λειτουργία της, όπως την κατάσταση της κάθε ψυκτικής μονάδας, αν δηλαδή είναι ενεργοποιημένη ή απενεργοποιημένη, καθώς και την κατάσταση της πόρτας αν δηλαδή είναι ανοιχτή ή κλειστή. Η τελευταία λειτουργία είναι ένας μηχανισμός ασφάλειας έτσι ώστε, αν αφήσει κάποιος ανοιχτή την πόρτα, ο χρήστης να ειδοποιηθεί για να την κλείσει. Ο web server θα είναι προστατευμένος με κωδικό και η είσοδος του χρήστη σε αυτήν θα μπορεί να γίνεται μόνο εντός τοπικού δικτύου. Τέλος, η επικοινωνία μεταξύ αισθητήριας συσκευής και του server που φιλοξενεί το διακομιστή ιστού είναι αμφίδρομη. Κάθε συσκευή στέλνει μετρήσεις θερμοκρασίας, υγρασίας, τάση και ποσοστού της μπαταρίας στο server, ενώ ο ιδίως ο server θα στέλνει τις ρυθμίσεις θερμοκρασίας κατωφλίου και ενεργοποίησης/απενεργοποίησης της ψυκτικής μονάδας έτσι ώστε να γίνεται ο κατάλληλος έλεγχος στον αισθητήρα.

Η παραπάνω κατασκευή έχει πολλές χρήσεις και μπορεί να εγκατασταθεί σε διάφορα καταστήματα εστίασης όπως ένα super market, μια βιομηχανία κρεάτων ή μια αποθήκη τροφίμων. Με αυτό τον τρόπο, έχοντας δηλαδή ο χρήστης, ο οποίος θα μπορεί να είναι κάποιος υπεύθυνος ή ένας υπάλληλος, τη δυνατότητα να παρακολουθεί σε μόνιμη βάση καθεμία από τις ψυκτικές μονάδες που έχει εγκατεστημένες στο κατάστημά του και να τις ελέγχει εξ αποστάσεως, παραδείγματος χάριν από το γραφείο του. Εν κατακλείδι, έχει γίνει η βέλτιστη προσπάθεια για να εξοικονομηθεί ενέργεια όσο το δυνατόν είναι αυτό εφικτό δεδομένου των γνώσεων που υπήρχαν και των οικονομικών πόρων που διατέθηκαν για την παρούσα πτυχιακή εργασία.

Κεφάλαιο 2ο: Hardware

2.1 Το Ηλεκτρονικό Κύκλωμα και η Πλακέτα

2.1.1 Εισαγωγή

Το ηλεκτρονικό κύκλωμα της αισθητήριας συσκευής αποτελείται από τρία υποκυκλώματα και κάθε ένα από αυτά θα εξηγηθεί και θα αναλυθεί λεπτομερώς στα επόμενα κεφάλαια. Το πρώτο κύκλωμα είναι το κύκλωμα φόρτισης, το οποίο είναι υπεύθυνο για τη φόρτιση της μπαταρίας, το δεύτερο κύκλωμα είναι το κύκλωμα σταθεροποίησης, το οποίο μετατρέπει τη σταθερή τάση της μπαταρίας σε σταθερή τάση που μπορεί να τροφοδοτηθεί το κύκλωμα και το τρίτο κύκλωμα είναι το κύκλωμα ρελέ, το οποίο χρησιμοποιείται κάθε φορά που απαιτείται να ενεργοποιηθεί ή να απενεργοποιηθεί η ψυκτική μονάδα στην οποία έχει συνδεθεί η συσκευή. Ο μικροελεγκτής που χρησιμοποιείται για την υλοποίηση των διαφόρων λειτουργιών της συσκευής είναι ο ESP32, ο οποίος είναι ένας ελεγκτής κατάλληλος για τον έλεγχο ενσωματωμένων κυκλωμάτων και εφαρμογών Διαδικτύου των Πραγμάτων (IoT) που απαιτούνται λειτουργίες Wi-Fi ή Bluetooth. Ο αισθητήρας θερμοκρασίας και υγρασίας είναι ο DHT22, ο οποίος είναι ένας οικονομικός αισθητήρας κατάλληλος για τέτοιου είδους ερευνητικά σχέδια. Το κύκλωμα έχει σχεδιαστεί και υλοποιηθεί στο πρόγραμμα EasyEDA. Το EasyEDA είναι ένα διαδικτυακό, εύχρηστο πρόγραμμα σχεδίασης κυκλωμάτων και ηλεκτρονικών πλακετών που επιτρέπει στους μηχανικούς να σχεδιάζουν, να προσομοιώνουν και να μοιράζονται σχηματικά, προσομοιώσεις και πλακέτες τυπωμένων κυκλωμάτων. Το EasyEDA επιτρέπει τη δημιουργία και την επεξεργασία σχηματικών διαγραμμάτων, την προσομοίωση SPICE μικτών αναλογικών και ψηφιακών κυκλωμάτων και τη δημιουργία και επεξεργασία διατάξεων πλακέτας τυπωμένων κυκλωμάτων.

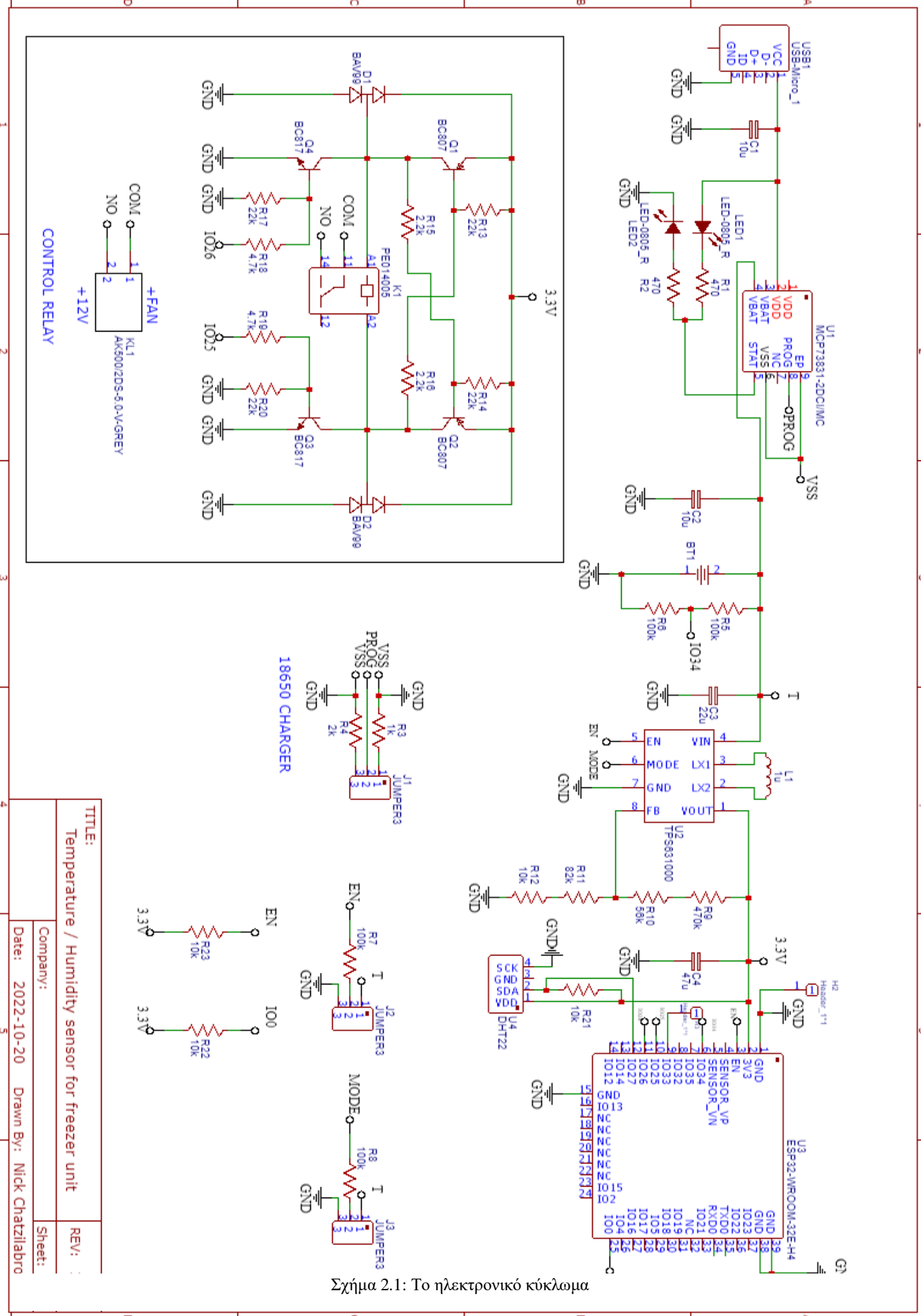
2.1.2 Το Ηλεκτρονικό Κύκλωμα

Το ηλεκτρονικό κύκλωμα, όπως αναφέρθηκε και στην εισαγωγή, αποτελείται από τρία κυκλώματα. Το πρώτο κύκλωμα είναι το κύκλωμα φόρτισης. Γενικότερα, η αισθητήρια συσκευή θα τροφοδοτείται από μία επαναφορτιζόμενη μπαταρία ιόντων λιθίου 18650, η οποία θα πρέπει να φορτίζει ανά τακτά χρονικά διαστήματα, προκειμένου να ανταπεξέλθει στις απαιτήσεις του κυκλώματος. Το κύκλωμα του φορτιστή λοιπόν, είναι υπεύθυνο γι' αυτήν ακριβώς τη δουλειά. Το κύκλωμα αποτελείται κυρίως από ένα ολοκληρωμένο MCP83831. Ο MCP73831 (U1) είναι ένας εξαιρετικά προηγμένος ελεγκτής γραμμικής διαχείρισης φόρτισης και χρησιμοποιείται σε εφαρμογές οι οποίες δεν είναι κοστοβόρες και είναι μικρές σε μέγεθος. Όταν ο χρήστης συνδέει τη συσκευή στο δίκτυο για να τη φορτίσει, τα 5V συνεχούς τάσης από το φορτιστή εισέρχονται στην είσοδο του ελεγκτή και η συνδεδεμένη στην έξοδο του κυκλώματος μπαταρία φορτίζει με σταθερή τάση περίπου 4.2V. Όσον αφορά τη φόρτιση της συσκευής, ο χρήστης έχει δύο επιλογές. Κανονική Φόρτιση και Γρήγορη Φόρτιση. Στην Κανονική Φόρτιση η μπαταρία φορτίζει με μέγιστο ρεύμα 500 mA ενώ στη Γρήγορη Φόρτιση η μπαταρία φορτίζει με μέγιστο ρεύμα 1 A. Η εναλλαγή αυτή γίνεται μέσω του Jumper J1 το οποίο μπορεί να βρεθεί πάνω στη πλακέτα (σχήμα 2.2).

Το δεύτερο κύκλωμα είναι το κύκλωμα σταθεροποίησης της τάσης. Επάνω στο κύκλωμα υπάρχουν πολλά υλικά τα οποία χρειάζονται μία σταθερή τάση για να λειτουργήσουν ομαλά. Αυτή τη σταθερή τάση δε μπορούν να τη λάβουν από τη μπαταρία ιόντων λιθίου, καθώς αυτή έχει ένα εύρος τάσης εξόδου από τα 4.2V – 2.8V. Αν για παράδειγμα προσπαθήσουμε να τροφοδοτήσουμε το μικροελεγκτή ESP32 απευθείας από τη μπαταρία οι πιθανότητες ο επεξεργαστής να καεί ή να μη λειτουργεί καθόλου είναι αυξημένες αφού το φύλλο δεδομένων του κατασκευαστή του ESP32 ορίζει

ως φυσιολογική τάση λειτουργίας του ελεγκτή τα $3V - 3.6V$. Κρίνεται λοιπόν αναγκαία η σύνδεση ενός κυκλώματος σταθεροποίησης DC – DC για τη παροχή σταθερής τάσης στο υπόλοιπο κύκλωμα. Αυτή η σταθεροποίηση θα παρέχει $3.3V$ στον επεξεργαστή, στο ρελέ και στα περιφερειακά SMD υλικά. Το ολοκληρωμένο το οποίο θα χρησιμοποιηθεί για αυτή τη δουλειά είναι ο TPS631000. Ο TPS631000 είναι ένας μετατροπέας σταθερής – σταθερής τάσης με μέγιστο ρεύμα περίπου τα $3A$. Η δυνατότητα του να παρέχει μέχρι και $3A$ εξασφαλίζει ότι ο σταθεροποιητής θα μπορεί να ανταπεξέλθει και στις μέγιστες απαιτήσεις του κυκλώματος. Η τάση λοιπόν της μπαταρίας εφαρμόζεται στην είσοδο του κυκλώματος και τη σταθεροποιεί στα $3.3V$. Αυτή η σταθεροποίηση μπορεί να είναι δύο ειδών. Είτε η τάση ενισχύεται και σταθεροποιείται στα $3.3V$, είτε η τάση υποβιβάζεται και σταθεροποιείται στα $3.3V$. Πιο συγκεκριμένα, αν το εύρος της τάσης της μπαταρίας είναι μεταξύ $2.8V - 3.3V$ η τάση ενισχύεται στην ονομαστική τιμή λειτουργίας, ενώ αν το εύρος της τάσης μπαταρίας είναι μεταξύ $3.3V - 4.2V$, η τάση υποβιβάζεται στην ονομαστική τιμή λειτουργίας. Μια επιπλέον λειτουργία του σταθεροποιητή είναι η επιλογή εξοικονόμησης ενέργειας ή η επιλογή εξαναγκασμένης λειτουργίας PWM για τη μέγιστη απόδοση μετατροπής.

Τέλος το τρίτο κύκλωμα είναι το κύκλωμα ρελέ. Το κύκλωμα ρελέ είναι υπεύθυνο για τον έλεγχο της κάθε ψυκτικής μονάδας, δηλαδή για την ενεργοποίηση και απενεργοποίηση ανάλογα με τις συνθήκες του εσωτερικού της. Το ρελέ που χρησιμοποιείται για την κατασκευή είναι ένα δισταθές ρελέ που παίρνει δύο καταστάσεις ανάλογα με την πολικότητα τάσης που θα εφαρμοστεί στα δύο πόδια εισόδου του. Αυτό σημαίνει ότι το ρελέ δεν χρειάζεται μόνιμα μια συνεχή τάση να εφαρμόζεται στην είσοδό του για να αλλάξει κατάσταση, αλλά κάθε φορά που αλλάζει η πολικότητα της τάσης που εφαρμόζεται στις δύο εισόδους του, αλλάζει και η κατάσταση του ρελέ. Με αυτό τον τρόπο υπάρχει εξοικονόμηση ενέργειας, καθώς η μπαταρία δε χρειάζεται να τροφοδοτεί συνέχεια το ρελέ για να το “κρατήσει” ανοιχτό. Το ρελέ που χρησιμοποιείται είναι το PEO14F03. Το κύκλωμα διαθέτει τέσσερα τρανζίστορ, 2 NPN και 2 PNP, που λειτουργούν σαν διακόπτες, και το ρελέ σπλίζει και αποπλίζει ανάλογα με την τάση στα 2 pins του ESP32 που είναι συνδεδεμένα στο κύκλωμα. Όταν το GPIO26 είναι σε λογικό 1 και το GPIO25 είναι λογικό 0 τότε άγει το τρανζίστορ Q2 και Q4, ενώ όταν το GPIO26 είναι σε λογικό 0 και το GPIO25 είναι λογικό 1 τότε άγει το τρανζίστορ Q1 και Q3. Στο σχήμα 2.1 απεικονίζεται το ηλεκτρονικό κύκλωμα της αισθητήριας συσκευής.

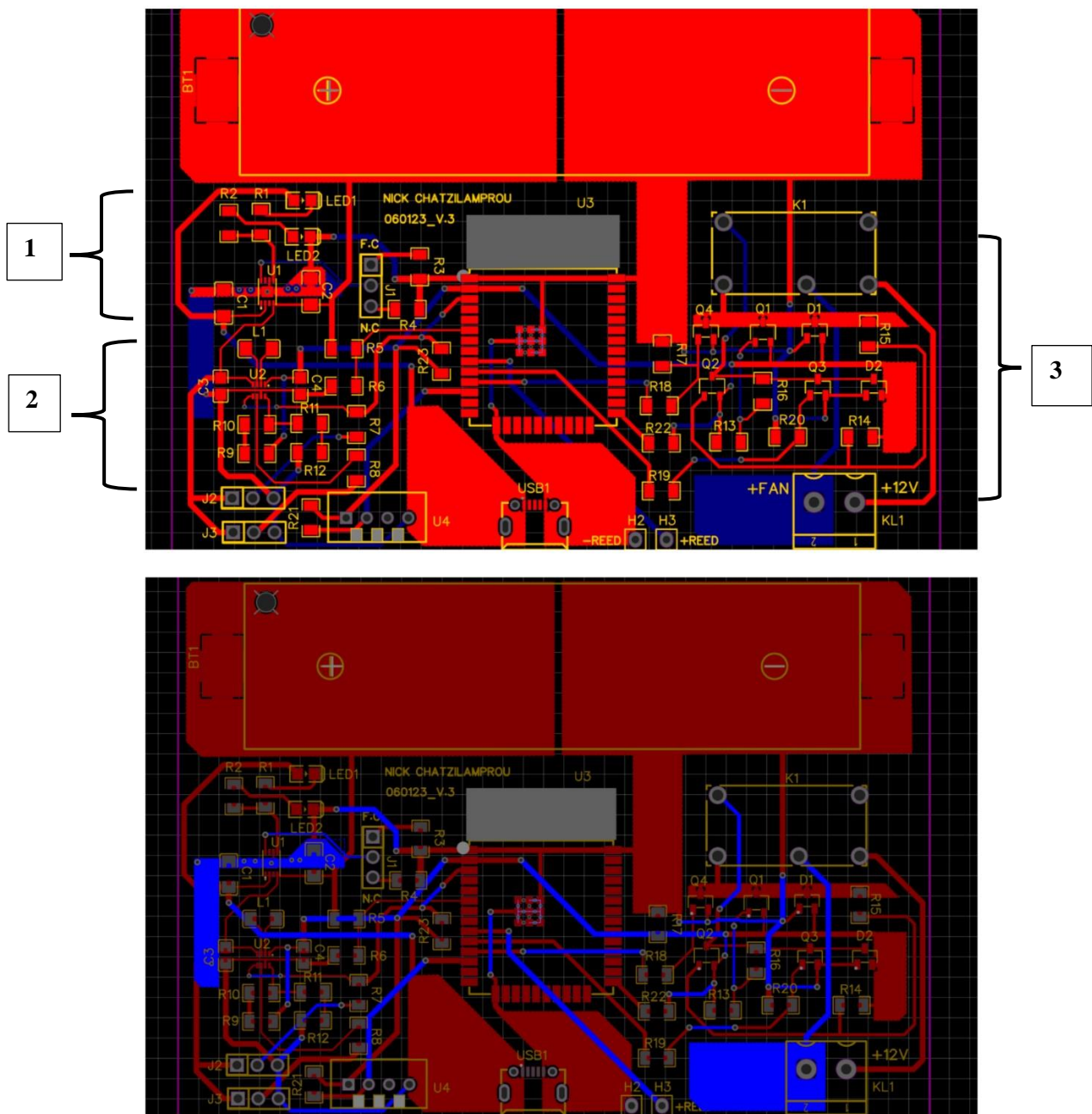


Σχήμα 2.1: Το ηλεκτρονικό κύκλωμα

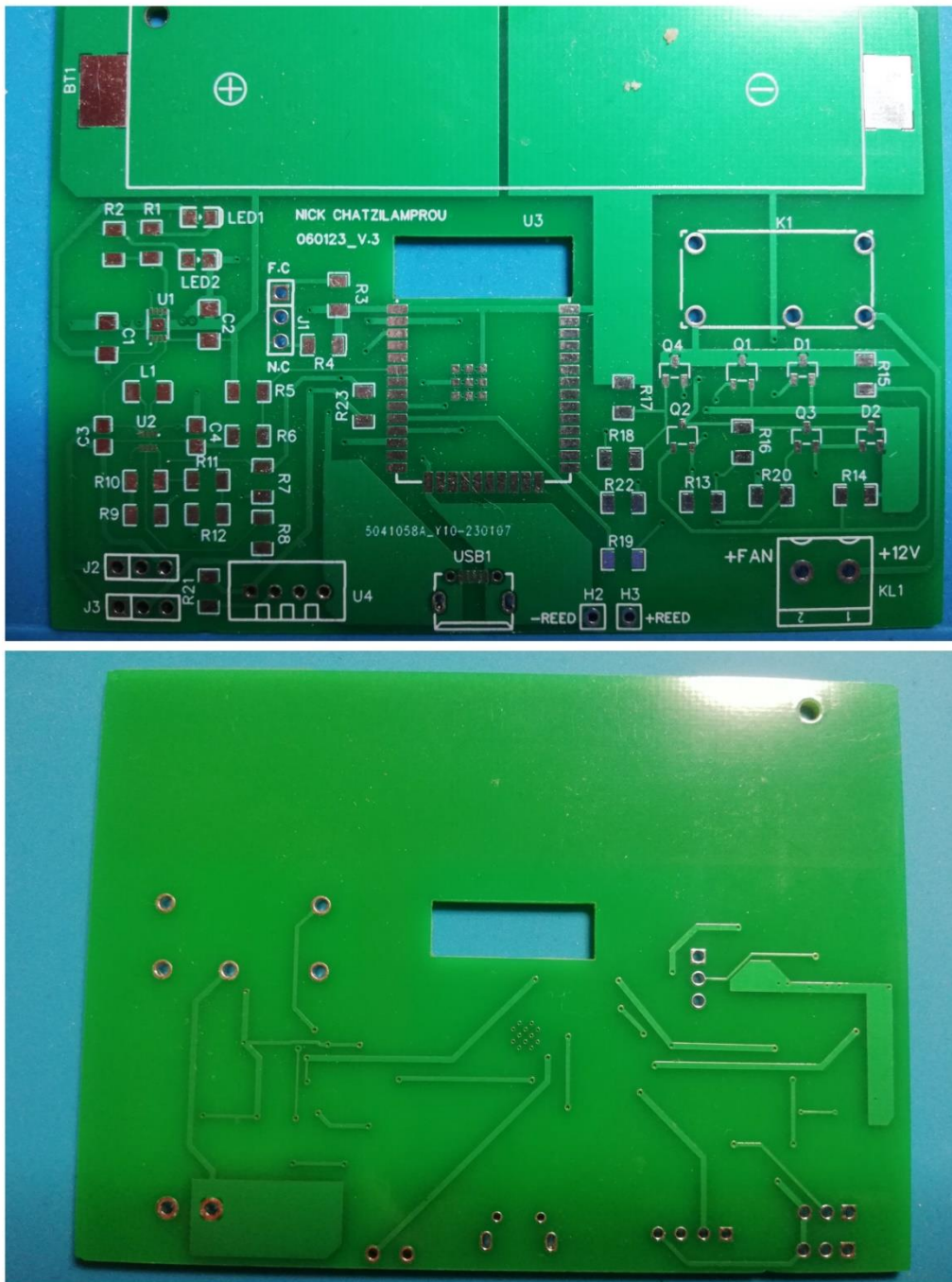
TITLE:		REV:
Temperature / Humidity sensor for freezer unit		
Company:	Date: 2022-10-20	Drawn By: Nick Chatzilabro
Sheet:		

2.1.3 Η πλακέτα PCB

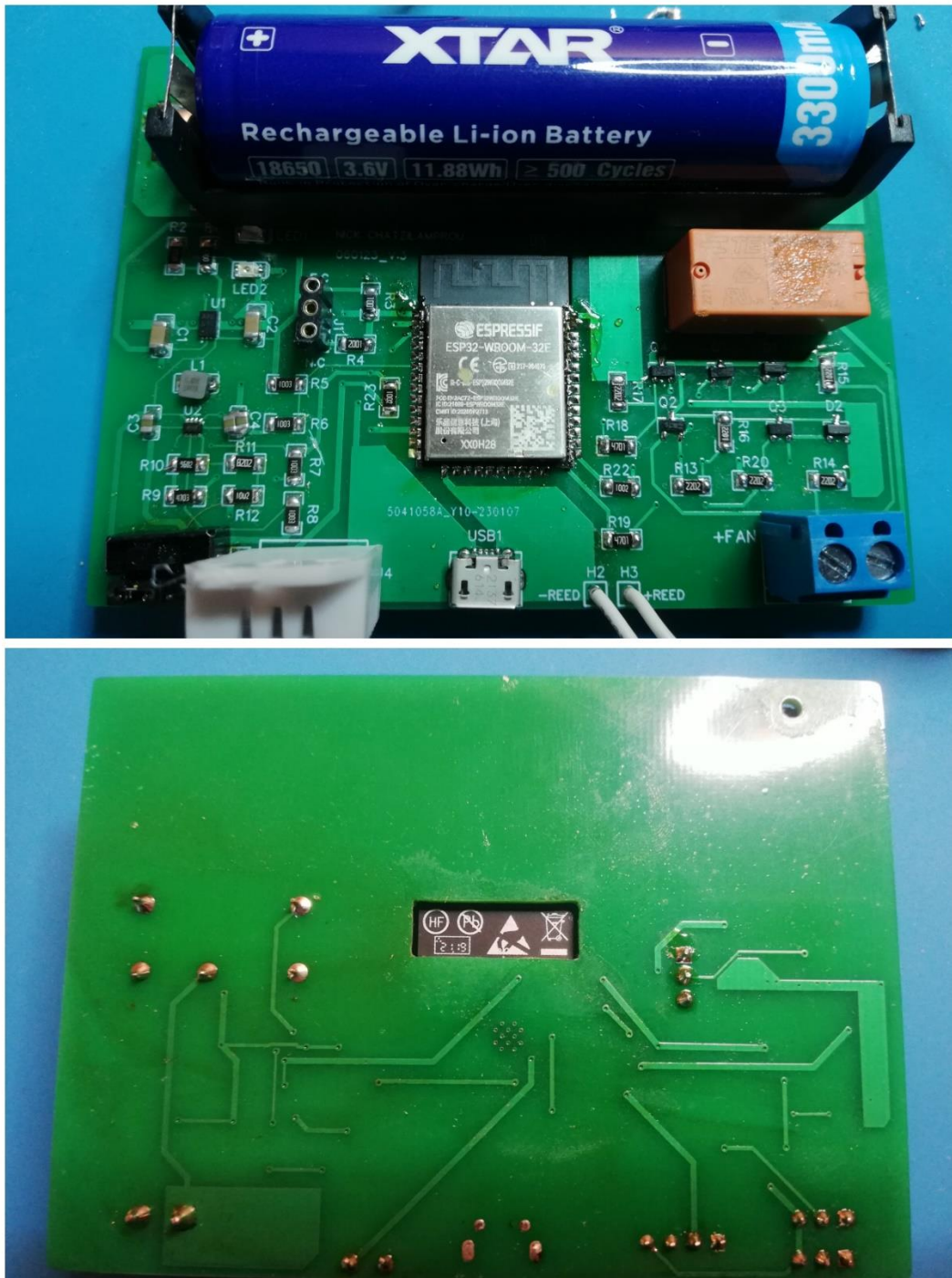
Το τυπωμένο ηλεκτρονικό κύκλωμα pcb σχεδιάστηκε όπως και το κύκλωμα στο διαδικτυακό πρόγραμμα EasyEDA και κατασκευάστηκε από την εταιρία JLCPCB. Μόλις τελείωσε η σχεδίαση του κυκλώματος έγινε μετατροπή του σχηματικού σε pcb. Η πλακέτα είναι 2 διαστάσεων και οι διαστάσεις της πλακέτας είναι 98mm – 68mm. Το πλάτος των γραμμών κυμαίνονται από 0.3mm έως και 0.7mm, η διάμετρος των τρυπανιών 0.8mm και το ελάχιστο μέγεθος τρυπανιού είναι 0.4mm. Παρακάτω, στο σχήμα 2.2, απεικονίζεται το σχηματικό της πλακέτας. Στο σημείο 1 βρίσκεται το κύκλωμα του φορτιστή, στο σημείο 2 βρίσκεται το κύκλωμα σταθεροποίησης και στο σημείο 3 βρίσκεται το κύκλωμα ρελέ. Στο σχήμα 2.3 απεικονίζεται η πλακέτα ασυναρμολόγητη και στο σχήμα 2.4 απεικονίζεται συναρμολογημένη (η πάνω εικόνα είναι το top layer ενώ στην κάτω εικόνα είναι το bottom layer).



Σχήμα 2.2: Η πλακέτα pcb



Σχήμα 2.3: Η πλακέτα ασυναρμολόγητη



Σχήμα 2.4: Η πλακέτα συναρμολογημένη

2.2 Ο μικροελεγκτής

2.2.1 Εισαγωγή

Ο ESP32 είναι ένας μικροελεγκτής χαμηλού κόστους και χαμηλής κατανάλωσης και είναι ο διάδοχος του μικροελεγκτή ESP8266. Κατασκευάστηκε και αναπτύσσεται από την Espressif Systems, η οποία είναι μια κινέζικη εταιρεία με έδρα στη Σαγκάη. Διαθέτει έναν διπύρηνο επεξεργαστή Tensilica Xtensa 32-bit LX6 με συχνότητα ρολογιού μέχρι και 240 MHz, αλλά υπάρχουν και παραλλαγές με μονοπύρηνο επεξεργαστή. Διαθέτει λειτουργίες Wi-Fi (150.0 Mbps ρυθμό δεδομένων) και 2 λειτουργίες Bluetooth, Bluetooth BLE και κλασικό Bluetooth. Ταυτόχρονα περιλαμβάνει ενσωματωμένους διακόπτες κεραίας, ενισχυτές ισχύος, ενισχυτές λήψης χαμηλού θορύβου, μετασηματιστές ραδιοσυχνότητας RF Balun, φίλτρα και μονάδες διαχείρισης ενέργειας. Το κόστος του μικροελεγκτή ξεκινάει από τα 2-3 ευρώ, ενώ το κόστος των αναπτυξιακών πλακετών, ξεκινάει από τα 7 ευρώ. Θα εξηγήσουμε τι είναι οι αναπτυξιακές πλακέτες στο κεφάλαιο 2.3. Ο ESP32 καταναλώνει πολύ λίγη ενέργεια συγκριτικά με άλλους μικροελεγκτές και υποστηρίζει λειτουργίες χαμηλής κατανάλωσης για εξοικονόμηση ενέργειας.



Σχήμα 2.5: Ο ESP32

2.2.2 Χαρακτηριστικά Στοιχεία

Ο ESP32 έχει πολλές παραπάνω λειτουργίες συγκριτικά με τον προκάτοχο του και κάποια από τα χαρακτηριστικά του μικροελεγκτή συνοψίζονται παρακάτω [12]:

Διπύρηνος επεξεργαστής Xtensa® Dual-Core 32-bit LX6 600 DMIPS. Ο ESP32 λειτουργεί από τα 160 MHz έως και τα 240 MHz. Οι 2 πυρήνες είναι ουσιαστικά υπεύθυνοι για διαφορετικές διεργασίες. Ο πρώτος πυρήνας ονομάζεται Protocol CPU (PRO CPU) και είναι υπεύθυνος για τη λειτουργία του Wi-Fi, Bluetooth και για όλες τις περιφερειακές εισόδους – εξόδους όπως SPI, I2C, καθώς και DAC και ADC. Ο δεύτερος πυρήνας ονομάζεται Application CPU (APP CPU) και χειρίζεται τον κώδικά τον οποίο έχει γράψει ο χρήστης.

Εσωτερική μνήμη 448 KB ROM που χρησιμοποιείται κατά την εκκίνηση της μικροελεγκτή και για τις λειτουργίες των πυρήνων.

Εσωτερική μνήμη 520 KB SRAM που χρησιμοποιείται για αποθήκευση δεδομένων.

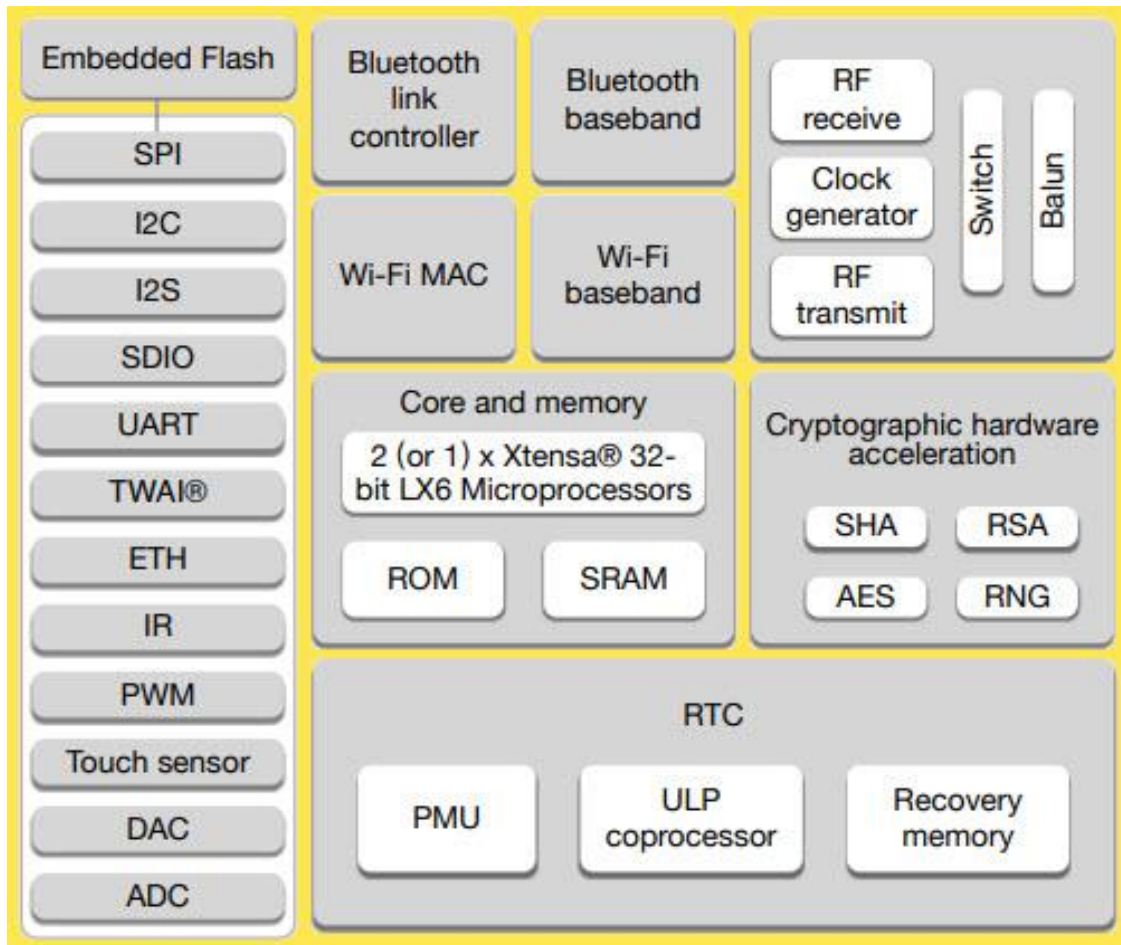
Εσωτερική μνήμη 16 KB RTC SRAM.

Wi-Fi Direct και TCP/IP, full 802.11 b/g/n/e/i WLAN MAC πρωτόκολλο επικοινωνίας. Ο ESP32 μπορεί να επικοινωνεί σχεδόν με τα περισσότερα μοντέλα Wi-Fi Routers όταν λειτουργεί ως πελάτης, αλλά μπορεί να λειτουργήσει και ως Access Point. Οι ταχύτητα του Wi-Fi μπορεί να ξεπεράσει τα 150 Mbps. Ο ESP32 υποστηρίζει επίσης Wi-Fi direct, το οποίο είναι μία πολύ καλή επιλογή για peer-to-peer επικοινωνία λόγω των υψηλών ταχυτήτων και ευκολίας χρήσης σε σχέση με άλλους τρόπους ανταλλαγής δεδομένων όπως το Bluetooth.

Bluetooth και Bluetooth BLE. Ο ESP32 υποστηρίζει λειτουργία Classic Bluetooth v4.2 καθώς και Bluetooth BLE. Η υποστήριξη και των 2 παραπάνω λειτουργιών ουσιαστικά σημαίνει ότι ο μικροελεγκτής μπορεί να επικοινωνεί με παλιές αλλά και με καινούργιες συσκευές που υποστηρίζουν Bluetooth.

34 προγραμματιζόμενες εισόδους – εξόδους. Ο ESP32 διαθέτει 34 GPIOs για προγραμματισμό και επικοινωνία του μικροελεγκτή με άλλες περιφερειακές συσκευές όπως έναν αισθητήρα. Πιο συγκεκριμένα, ο ESP32 διαθέτει:

- Χρονιστές και χρονομετρητές επαγρύπνησης (WDT watchdog timer)
- RTC (Real Time Clock)
- ADC (Analog to Digital Converter)
- DAC (Digital to Analog Converter)
- Αισθητήρες αφής
- Διεπαφή Ethernet MAC
- UART (Universal Asynchronous Receiver Transmitter)
- Διεπαφή I2C
- Διεπαφή I2S
- Διεπαφή SPI
- IR Remote
- Μετρητής παλμών
- PWM
- SDIO



Σχήμα 2.6: Μπλοκ – διάγραμμα του ESP32.

2.2.3 Ηλεκτρικά Στοιχεία

Παραπάνω αναλύθηκαν συνοπτικά τα τεχνικά χαρακτηριστικά του ESP32 και εν συνεχεία παρατίθενται τα ηλεκτρικά χαρακτηριστικά του ακριβώς όπως καταγράφονται στο επίσημο φύλλο δεδομένων της Espressif.

Απόλυτα Μέγιστες Τιμές: Συνθήκες πάνω από τις παρακάτω τιμές μπορεί να προκαλέσουν μόνιμη βλάβη στο μικροελεγκτή.

Πίνακας 2.1: ESP32 - Μέγιστες Επιτρεπτές Τιμές

Symbol	Parameter	Min	Max	Unit
VDDA, VDD3P3, VDD3P3_RTC, VDD3P3_CPU, VDD_SDIO	Voltage applied to power supply pins per power domain	-0.3	3.6	V
I_{output}^*	Cumulative IO output current	-	1200	mA
T_{store}	Storage temperature	-40	150	°C

* The chip worked properly after a 24-hour test in ambient temperature at 25 °C, and the IOs in three domains (VDD3P3_RTC, VDD3P3_CPU, VDD_SDIO) output high logic level to ground.

Πίνακας 2.2: ESP32 - Συνιστώμενες Συνθήκες Λειτουργίας

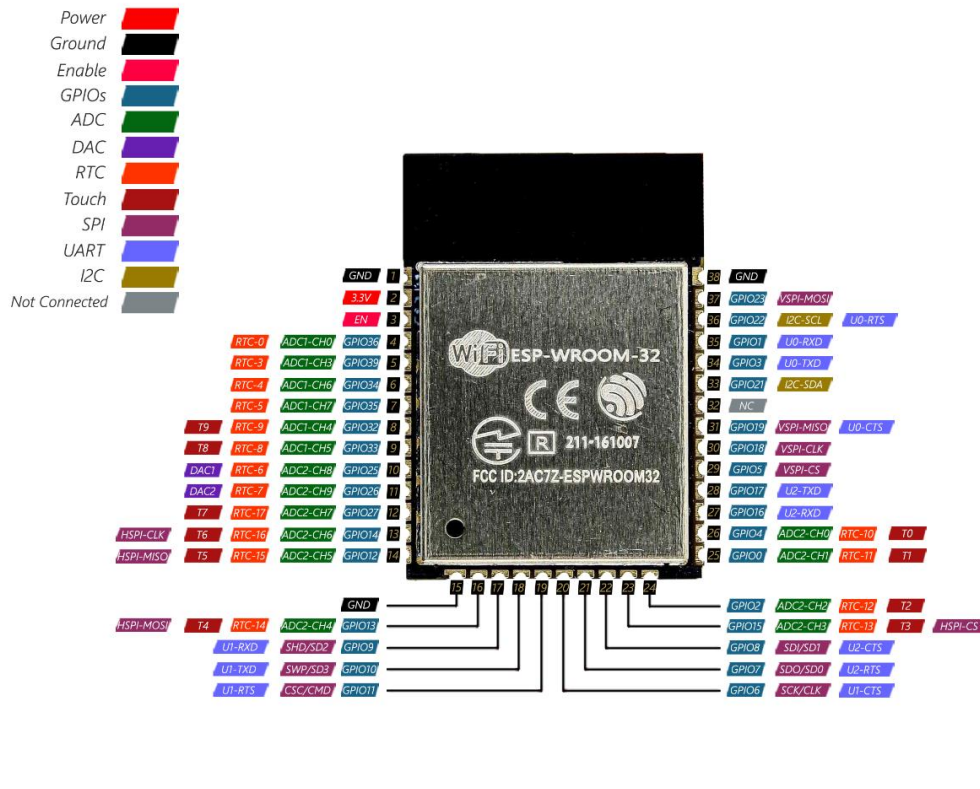
Symbol	Parameter	Min	Typ	Max	Unit
VDDA, VDD3P3_RTC, ^{note 1} VDD3P3, VDD_SDIO (3.3 V mode) ^{note 2}	Voltage applied to power supply pins per power domain	2.3/3.0 ^{note 3}	3.3	3.6	V
VDD3P3_CPU	Voltage applied to power supply pin	1.8	3.3	3.6	V
I_{VDD}	Current delivered by external power supply	0.5	-	-	A
T ^{note 4}	Operating temperature	-40	-	125	°C

Πίνακας 2.3: ESP32 - DC Χαρακτηριστικά

Symbol	Parameter	Min	Typ	Max	Unit	
C_{IN}	Pin capacitance	-	2	-	pF	
V_{IH}	High-level input voltage	$0.75 \times VDD^1$	-	$VDD^1 + 0.3$	V	
V_{IL}	Low-level input voltage	-0.3	-	$0.25 \times VDD^1$	V	
I_{IH}	High-level input current	-	-	50	nA	
I_{IL}	Low-level input current	-	-	50	nA	
V_{OH}	High-level output voltage	$0.8 \times VDD^1$	-	-	V	
V_{OL}	Low-level output voltage	-	-	$0.1 \times VDD^1$	V	
I_{OH}	High-level source current ($VDD^1 = 3.3$ V, $V_{OH} \geq 2.64$ V, output drive strength set to the maximum)	VDD3P3_CPU power domain ^{1, 2}	-	40	-	mA
		VDD3P3_RTC power domain ^{1, 2}	-	40	-	mA
		VDD_SDIO power domain ^{1, 3}	-	20	-	mA
I_{OL}	Low-level sink current ($VDD^1 = 3.3$ V, $V_{OL} = 0.495$ V, output drive strength set to the maximum)	-	28	-	mA	
R_{PU}	Resistance of internal pull-up resistor	-	45	-	k Ω	
R_{PD}	Resistance of internal pull-down resistor	-	45	-	k Ω	
V_{IL_nRST}	Low-level input voltage of CHIP_PU to power off the chip	-	-	0.6	V	

2.2.4 Είσοδοι / Έξοδοι (GPIOs)

Όπως αναφέρεται και στην παραπάνω υποενότητα, ο ESP32 έχει μεγάλη πληθώρα εισόδων – εξόδων συγκριτικά με άλλους μικροελεγκτές της κατηγορίας του. Παρακάτω φαίνονται όλα τα GPIOs του ESP32, καθώς και η λειτουργία του καθενός.



Σχήμα 2.7: ESP32 - Pinout Reference

Παρόλα αυτά, συνολικά για τη συγκεκριμένη κατασκευή, θα χρησιμοποιηθούν τα παρακάτω pinouts.

Πίνακας 2.4: Pinouts του ESP32 που χρησιμοποιούνται

GPIO	Περιγραφή
3V3	Pin για τη τροφοδοσία 3.3V
GND	Pin για τη γείωση
IO25	Pin για τον έλεγχο κυκλώματος ρελέ
IO26	Pin για τον έλεγχο κυκλώματος ρελέ
IO27	Pin για την επικοινωνία με τον αισθητήρα θερμοκρασίας / υγρασίας DHT22
IO33	Pin για την ανάγνωση κατάστασης μαγνητικής επαφής
IO34	Pin για την ανάγνωση τάσης μπαταρίας 18650

2.2.5 Λειτουργία ύπνου

Ο μικροελεγκτής ESP32 είναι αδιαμφισβήτητα ένας ελεγκτής με πάρα πολλές λειτουργίες αλλά είναι επίσης ένας ελεγκτής που καταναλώνει μεγάλες ποσότητες ενέργειας και συνήθως απαιτεί πολύ ρεύμα για να λειτουργήσει [13]. Γι' αυτό το λόγο, διαθέτει τη λειτουργία ύπνου, μια λειτουργία που επιτρέπει τον ESP32 να καταναλώνει ελάχιστα ποσά ενέργειας απενεργοποιώντας εσωτερικά περιφερειακά όπως τις Bluetooth και Wi-Fi, το RTC ή ακόμα και τον επεξεργαστή του. Υπάρχουν πολλών ειδών λειτουργίες ύπνου, που κάθε μία από αυτές έχει διαφορετική κατανάλωση. Στο σχήμα 2.6, απεικονίζεται το μπλοκ-διάγραμμα του ESP32 που περιέχει τα εσωτερικά περιφερειακά συστήματα

του, και έτσι μέσω της απεικόνισης αυτών είναι ευκολότερη κατανόηση αυτών που θα αναλυθούν παρακάτω.

Ο ESP32 διαθέτει πέντε είδη λειτουργίας:

- Ενεργή Λειτουργία (Active Mode)
- Λειτουργία Ύπνου Modem (Modem Sleep Mode)
- Λειτουργία Ελαφρύ Ύπνου (Light Sleep Mode)
- Λειτουργία Βαθύ Ύπνου (Deep Sleep Mode)
- Λειτουργία Αδρανοποίησης (Hibernation Mode)

Ενεργή Λειτουργία

Σε αυτή τη λειτουργία βρίσκεται ο ελεγκτής, όταν δουλεύει σε φυσιολογική κατάσταση και είναι ενεργοποιημένες όλες οι περιφερειακές λειτουργίες του ESP32. Σε αυτή τη περίπτωση είναι ενεργοποιημένα το Bluetooth, το Wi-Fi, η κεραία RF, οι περιφερειακές επαφές, οι δύο επεξεργαστές με τη μνήμη και το RTC. Η κατανάλωση κυμαίνεται περίπου στα 160-300mA, αλλά σε πολλές περιπτώσεις φτάνει και σε τιμές άνω των 500mA, ειδικά αν είναι ενεργοποιημένα το Bluetooth και το Wi-Fi ταυτόχρονα.

Λειτουργία Ύπνου Modem

Σε αυτή τη λειτουργία βρίσκεται ο ελεγκτής και είναι ενεργοποιημένα οι δύο επεξεργαστές με τη μνήμη και το RTC και η κατανάλωση κυμαίνεται περίπου στα 3-20mA.

Λειτουργία Ελαφρύ Ύπνου

Η λειτουργία ελαφρύ ύπνου είναι παρόμοια με τη λειτουργία ύπνου modem, καθώς το τσιπ ακολουθεί το μοτίβο ύπνου συσχέτισης. Κατά τη λειτουργία ελαφρύ ύπνου, η CPU τίθεται σε παύση απενεργοποιώντας τον παλμό ρολογιού της. Ο συνεπεξεργαστής RTC και ULP, από την άλλη πλευρά, παραμένουν ενεργοί. Αυτό έχει ως αποτέλεσμα χαμηλότερη κατανάλωση ενέργειας από τη λειτουργία ύπνου μόντεμ, η οποία είναι περίπου 0,8mA.

Λειτουργία Βαθύ Ύπνου

Στη λειτουργία βαθιάς αναστολής λειτουργίας, οι CPU, το μεγαλύτερο μέρος της μνήμης RAM και όλα τα ψηφιακά περιφερειακά είναι απενεργοποιημένα, ενώ ο επεξεργαστής ULP παραμένει ενεργοποιημένος και μπορεί να αφυπνίσει την CPU όταν χρειάζεται. Μαζί με τη CPU, απενεργοποιείται και η μνήμη του τσιπ, επομένως οτιδήποτε αποθηκεύεται εκεί διαγράφεται. Παρόλα αυτά, η μνήμη RTC διατηρείται ενεργή, και τα περιεχόμενά της διατηρούνται ακόμη και κατά τη διάρκεια του βαθύ ύπνου και μπορούν να ανακτηθούν μόλις αφυπνιστεί το τσιπ. Σε αυτή τη κατάσταση, ο επεξεργαστής καταναλώνει περίπου 10μΑ.

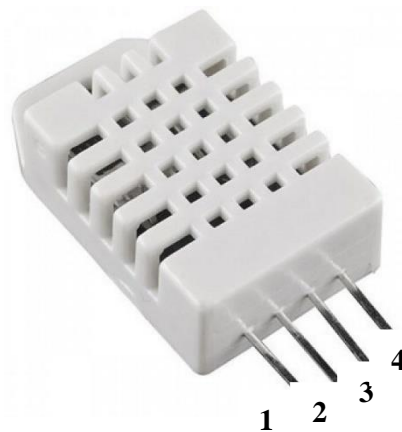
Λειτουργία Αδρανοποίησης

Η λειτουργία αδρανοποίησης μοιάζει πολύ με τη λειτουργία βαθύ ύπνου. Η μόνη διαφορά είναι ότι στη λειτουργία αδρανοποίησης, το τσιπ απενεργοποιεί τον εσωτερικό ταλαντωτή 8MHz, καθώς και τον επεξεργαστή ULP, αφήνοντας μόνο έναν χρονοδιακόπτη RTC (σε αργό ρολόι) και μερικές RTC επαφές για να ξυπνήσουν το τσιπ. Επίσης και η μνήμη είναι απενεργοποιημένη, επομένως οτιδήποτε αποθηκεύεται εκεί θα διαγραφεί. Σε αυτή τη κατάσταση, ο επεξεργαστής καταναλώνει περίπου 2.5μΑ.

2.3 Ο αισθητήρας

2.3.1 Εισαγωγή

Ο AM2302, ή αλλιώς DHT22, είναι ένας χαμηλού κόστους ψηφιακός αισθητήρας μέτρησης θερμοκρασίας και υγρασίας. Χρησιμοποιεί ένα θερμίστορ για τη μέτρηση της θερμοκρασίας και ένα χωρητικό αισθητήρα υγρασίας για τη μέτρηση της υγρασίας, ενώ εκπέμπει ένα ψηφιακό σήμα στον ακροδέκτη δεδομένων. Έπειτα, λαμβάνει αυτό το σήμα ο ESP32 και το αποθηκεύει ως μια μεταβλητή την οποία μπορεί μετά ο χρήστης να καλέσει και να απεικονίσει σε μία LCD οθόνη, ή όπως στην περίπτωση της συγκεκριμένης εργασίας, σε έναν Web Server. Πιο συγκεκριμένα, για τη μέτρηση της θερμοκρασίας, ο DHT22 διαθέτει στο εσωτερικό του ένα θερμίστορ. Το θερμίστορ ή αλλιώς NTC, είναι στην ουσία μια θερμική αντίσταση, της οποίας η τιμή αλλάζει αυτόματα, ανάλογα με τη θερμοκρασία του περιβάλλοντος. Η τιμή της αντίστασης του θερμίστορ μειώνεται όσο αυξάνεται η θερμοκρασία περιβάλλοντος, σε αντίθεση με τις μεταλλικές αντιστάσεις, των οποίων η αντίσταση αυξάνεται με την αύξηση της θερμοκρασίας. Αυτό οφείλεται στα κεραμικά ημιαγωγικά υλικά, από τα οποία είναι κατασκευασμένο και επομένως τα θερμίστορ έχουν αρνητικό συντελεστή θερμοκρασίας. Τέλος, το εύρος των τιμών της θερμοκρασίας που μπορεί να μετρήσει ο αισθητήρας κυμαίνεται από -40°C έως $+80^{\circ}\text{C}$. Επομένως ο DHT22 είναι ιδανικός για να χρησιμοποιηθεί στη συγκεκριμένη εργασία ως αισθητήριο όργανο, καθώς η αισθητήρια συσκευή θα βρίσκεται σε περιβάλλον χαμηλής θερμοκρασίας, που πολλές φορές μπορεί να είναι και αρνητική.



Σχήμα 2.8: Ο DHT22

2.3.2 Συνδέσεις

Στο Pin 1 του αισθητήρα συνδέεται η τροφοδοσία του. Ο DHT22 για να λειτουργήσει χρειάζεται 3.3V έως και 6V συνεχούς τάσης. Στο Pin 2 του αισθητήρα χρησιμοποιείται για τη μεταφορά δεδομένων μεταξύ αισθητήρα και μικροελεγκτή και πρέπει πάντα να παρεμβάλλεται με μία pull-up αντίσταση. Τέλος, το Pin 3 μένει ασύνδετο, ενώ το Pin 4 συνδέεται στη γείωση του κυκλώματος. Στον πίνακα 2.5 γίνεται η περιγραφή των εξωτερικών ηλεκτρικών συνδέσμων.

Πίνακας 2.5: DHT22 - Λειτουργία Εξωτερικών Pin

Pin	Λειτουργία
1	V_{DD} - Τροφοδοσία
2	Δεδομένα
3	Ασύνδετο
4	Γείωση GND

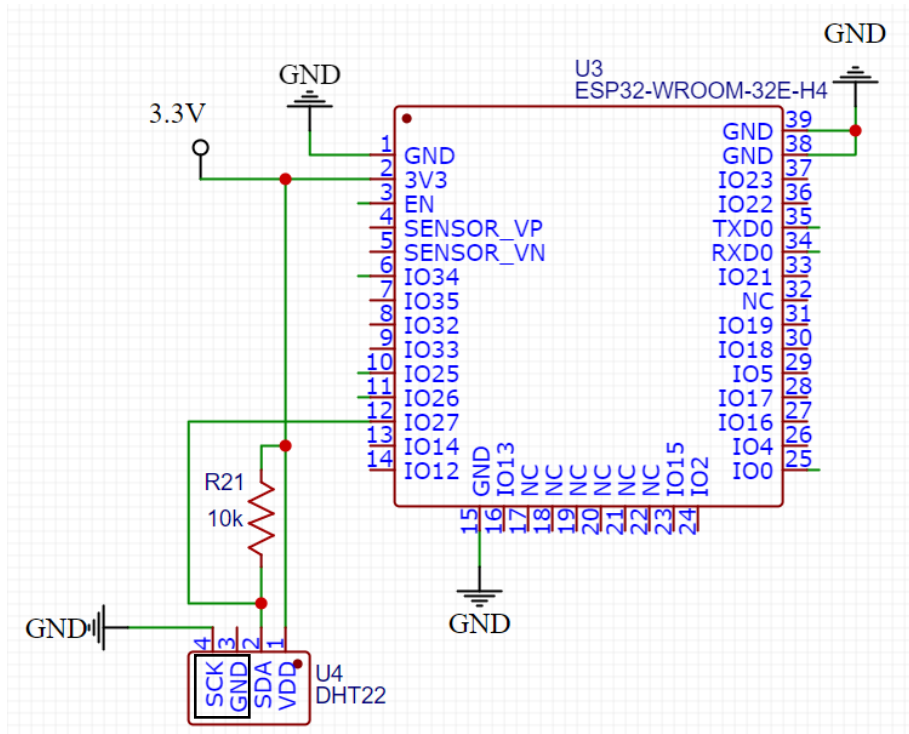
2.3.3 Χαρακτηριστικά Στοιχεία

Στον πίνακα 2.6 παρατίθενται τα βασικότερα από τα ηλεκτρικά DC χαρακτηριστικά από το επίσημο φύλλο δεδομένων του ολοκληρωμένου.

Πίνακας 2.6: DHT22 - Τεχνικά Χαρακτηριστικά

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

Στο σχήμα 2.9 απεικονίζεται το κομμάτι του κυκλώματος που είναι συνδεδεμένος ο DHT22 με τον ESP32. Η αντίσταση $R21 = 10k$ είναι η προαναφερθείσα pull-up αντίσταση και είναι συνδεδεμένη στο pin 2 του αισθητήρα.



Σχήμα 2.9: Το κύκλωμα σύνδεσης DHT22 με τον ESP32

Σημείωση:

Στο σχήμα 2.9 παραπάνω φαίνεται ο U4, ο οποίος είναι ο αισθητήρας θερμοκρασίας. Στο σχέδιο του αισθητήρα φαίνεται ότι η σύνδεση της γείωσης είναι το pin 3 και η ασύνδετη σύνδεση είναι το pin 4 (όπως φαίνεται και στο πλαίσιο πάνω στον U4) και το παραπάνω έρχεται σε αντίθεση με τις πληροφορίες στον πίνακα 2.5. Στην πραγματικότητα αυτό είναι ένα λάθος της βιβλιοθήκης που χρησιμοποιήθηκε για τους σκοπούς της σχεδίασης του ηλεκτρονικού κυκλώματος και όπως φαίνεται από τις συνδέσεις έχει ληφθεί υπ' όψιν.

2.4 Το Κύκλωμα Φόρτισης

2.4.1 Εισαγωγή

Το κύκλωμα του φορτιστή είναι υπεύθυνο για τη φόρτιση της μπαταρίας 18650, η οποία τροφοδοτεί το κύκλωμα. Στο κύκλωμα εφαρμόζεται σταθερή τάση 5V από έναν απλό φορτιστή 5V και φορτίζει τη μπαταρία με σταθερή τάση 4.2V. Το ρεύμα εξόδου και ως κατά συνέπεια ο ρυθμός φόρτισης της μπαταρίας μπορεί να επιλεγεί από το χρήστη, έχοντας την επιλογή των 1 Ampere και 0.5 Ampere.

2.4.2 MCP73831

Στην “καρδιά” του κυκλώματος υπάρχει το ολοκληρωμένο MCP73831. Το ολοκληρωμένο αυτό είναι ένας γραμμικός ελεγκτής φόρτισης μπαταριών και μπορεί να χρησιμοποιηθεί σε εφαρμογές, όπου απαιτείται κύκλωμα φόρτισης μπαταριών λιθίου όπως μία ψηφιακή κάμερα, ένα MP3 Player ή ακόμα και μία “έξυπνη” συσκευή όπως ένα smartphone. Ο MCP73831 τροφοδοτείται με σταθερή τάση 3.75V – 6V και παρέχει σταθερή τάση 4.20V για να φορτίσει τη μπαταρία του κυκλώματος. Το σταθερό ρεύμα φόρτισης ρυθμίζεται με μία εξωτερική αντίσταση R_{PROG} . Παρόλα αυτά η συσκευή έχει

Πίνακας 2.7: MCP73831 – Ηλεκτρικά Χαρακτηριστικά

Electrical Specifications: Unless otherwise indicated, all limits apply for $V_{DD} = [V_{REG}(\text{typical}) + 1.0V]$ to 6V, $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$. Typical values are at $+25^\circ\text{C}$.						
Parameters	Sym.	Min.	Typ.	Max.	Units	Conditions
Supply Input						
Supply Voltage	V_{DD}	3.75	—	6	V	
Supply Current	I_{SS}	—	510	1500	μA	Charging
		—	53	200	μA	Charge Complete, No Battery
		—	25	50	μA	PROG Floating
		—	1	5	μA	$V_{DD} \leq (V_{BAT} - 50 \text{ mV})$
		—	0.1	2	μA	$V_{DD} < V_{STOP}$
UVLO Start Threshold	V_{START}	3.3	3.45	3.6	V	V_{DD} Low-to-High
UVLO Stop Threshold	V_{STOP}	3.2	3.38	3.5	V	V_{DD} High-to-Low
UVLO Hysteresis	V_{HYS}	—	70	—	mV	

Στον πίνακα 2.8 γίνεται η περιγραφή των εξωτερικών ηλεκτρικών συνδέσμων.

Πίνακας 2.8: MCP73831 - Λειτουργία Εξωτερικών Pin

Pin No.	Σύμβολο	Λειτουργία
1	V_{DD}	Τροφοδοσία Εισόδου Ολοκληρωμένου
2	V_{DD}	Τροφοδοσία Εισόδου Ολοκληρωμένου
3	V_{BAT}	Τάση Εξόδου Τροφοδοσίας Μπαταρίας
4	V_{BAT}	Τάση Εξόδου Τροφοδοσίας Μπαταρίας
5	STAT	Κατάσταση Φόρτισης Μπαταρίας
6	V_{SS}	0V Αναφορά
7	NC	Not Connected
8	PROG	Ρύθμιση Ρεύματος Τροφοδοσίας
9	EP	Εκτεθειμένο Θερμικό Pad

2.4.3 Ανάλυση Κυκλώματος Φόρτισης

Αρχικά, η τάση εφαρμόζεται μέσω του βύσματος micro USB. Αυτή η τάση είναι 5V και πρέπει να κυμαίνεται στο εύρος των επιθυμητών τιμών που ορίζει ο κατασκευαστής του controller (3.75V – 6V).

2.4.3.1 Εξωτερικοί Πυκνωτές

Οι 2 εξωτερικοί κεραμικοί πυκνωτές SMD μεγέθους 1206 C1 και C2 στο σχήμα 2.11, είναι υπεύθυνοι για την αντιστάθμιση σε περίπτωση που δεν υπάρχει φορτίο στην έξοδο του κυκλώματος, η οποία παρέχεται από τη χωρητικότητα των 2 πυκνωτών στην είσοδο και έξοδο του κυκλώματος. Με αυτό τον τρόπο, διατηρείται καλή σταθερότητα AC στη λειτουργία σταθερής τάσης. Επιπλέον, η μπαταρία και οι διασυνδέσεις του κυκλώματος παρουσιάζουν επαγωγική συμπεριφορά σε υψηλές συχνότητες. Αυτά τα στοιχεία, βρίσκονται στο βρόχο ανάδρασης κατά τη διάρκεια λειτουργίας σταθερής τάσης και επομένως, η χωρητικότητα των δύο πυκνωτών είναι απαραίτητη για την αντιστάθμιση της επαγωγικής φύσης της μπαταρίας.

2.4.3.3 Φόρτιση Συσκευής

Ο χρήστης έχει δύο επιλογές όσον αφορά τη φόρτιση της συσκευής. Αυτές οι δύο επιλογές είναι “Κανονική Φόρτιση” και “Γρήγορη Φόρτιση”. Αυτή η λειτουργία επιλέγεται μέσω του J1(σχήμα 2.12) από την πλακέτα pcb. Ουσιαστικά πρόκειται για ένα faston jumper, το οποίο βραχυκυκλώνει το μεσαίο pin του jumper που είναι συνδεδεμένο με το pin “PROG”, με μία από τις δύο αντιστάσεις R_3 , R_4 . Η αντίσταση $R_3 = 1k$ και η αντίσταση $R_4 = 2k$. Η αντίσταση R_3 είναι η αντίσταση για την επιλογή “Γρήγορη Φόρτιση”, ενώ η αντίσταση R_4 είναι η αντίσταση για την επιλογή “Κανονική Φόρτιση”.

Το ρεύμα φόρτισης I_{REG} με $R_{PROG} = R_3$ υπολογίζεται από τον τύπο:

$$I_{REG} = \frac{1000V}{R_3} = \frac{1000}{1} = 1000mA \quad (2.1)$$

Ενώ το ρεύμα φόρτισης I_{REG} με $R_{PROG} = R_4$ υπολογίζεται από τον τύπο:

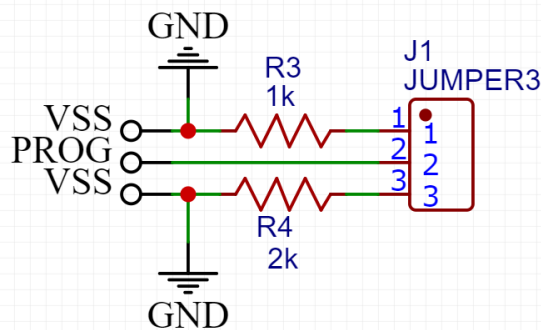
$$I_{REG} = \frac{1000V}{R_4} = \frac{1000}{2} = 500mA \quad (2.2)$$

Οπου:

$$R_{PROG} = kOhms$$

$$I_{REG} = milliampere$$

Στο σχήμα 2.12 απεικονίζεται το κύκλωμα επιλογής τύπου φόρτισης ανάλογα με το που θα τοποθετηθεί το jumper J1



18650 CHARGER

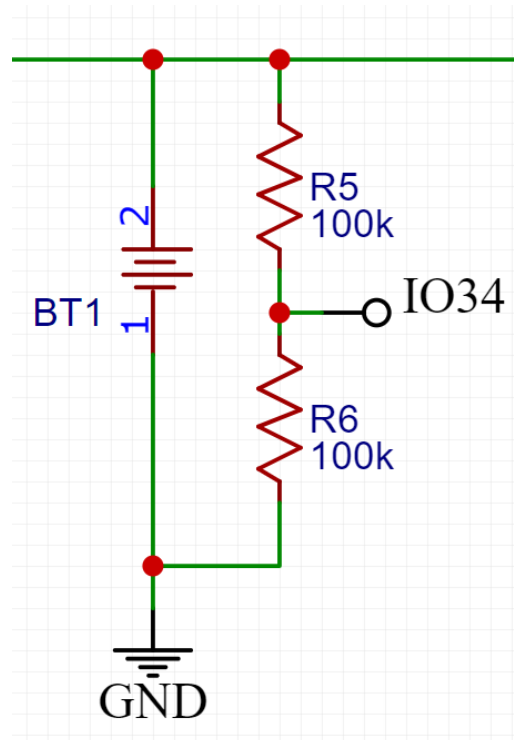
Σχήμα 2.12: Το κύκλωμα επιλογής ρεύματος φόρτισης

Ανεξάρτητα από το κύκλωμα φόρτισης, πάνω στην μπαταρία, υπάρχει ένας διαιρέτης τάσης όπως φαίνεται στο σχήμα 2.13. Ανάμεσα από τις δύο αντιστάσεις R_5 και R_6 βρίσκεται η επαφή GPIO34 του ESP32, η οποία είναι μια ADC είσοδος-έξοδος. Γενικότερα, στις επαφές του μικροελεγκτή δεν πρέπει να εφαρμοστεί τάση μεγαλύτερη των 3.6V γιατί υπάρχει κίνδυνος καταστροφής του τσιπ. Για αυτόν το λόγο υπάρχει ο διαιρέτης τάσης έτσι ώστε να μειώνει την τάση της μπαταρίας στο μισό. Με αυτόν τον τρόπο η τάση που εφαρμόζεται στη συγκεκριμένη επαφή του ESP32 δεν ξεπερνάει τα $4.2/2 = 2.1V$. Η τιμή των αντιστάσεων που έχουν επιλεγεί, έχει γίνει με σκοπό την ακριβέστερη

μέτρηση της τάσης και ταυτόχρονα τη χαμηλότερη κατανάλωση όσον αφορά το ρεύμα που διαρρέει αυτές τις δύο αντιστάσεις. Με $R_5 = R_6 = 100k\Omega$ η μέγιστη κατανάλωση θα είναι περίπου ίση με:

$$I = \frac{V_{BAT}}{R_5 + R_6} = \frac{4.2V}{200k\Omega} = 21\mu A \quad (2.3)$$

Στο σχήμα 2.13 απεικονίζεται το κύκλωμα διαιρέτη τάσης ανάμεσα στην μπαταρία.



Σχήμα 2.13: Διαίρετης Τάσης Μπαταρίας

2.5 Το Κύκλωμα Σταθεροποίησης

2.5.1 Εισαγωγή

Το κύκλωμα του σταθεροποιητή τροφοδοτεί με σταθερή τάση τον ESP32 και κατ' επέκταση ολόκληρο το κύκλωμα. Το κύκλωμα τροφοδοτείται από τη μπαταρία η οποία παρέχει μία σταθερή τάση με ένα αρκετά ικανοποιητικό εύρος τιμών τάσης περίπου από τα 2.8V έως και τα 4.2V. Η τάση εξόδου του κυκλώματος, δηλαδή η τάση που εφαρμόζεται στο μικροελεγκτή είναι 3.35V και ελέγχεται μέσω ενός διαιρέτη τάσης που είναι συνδεδεμένος στο feedback pin του ολοκληρωμένου που χρησιμοποιείται.

2.5.2 TPS631000

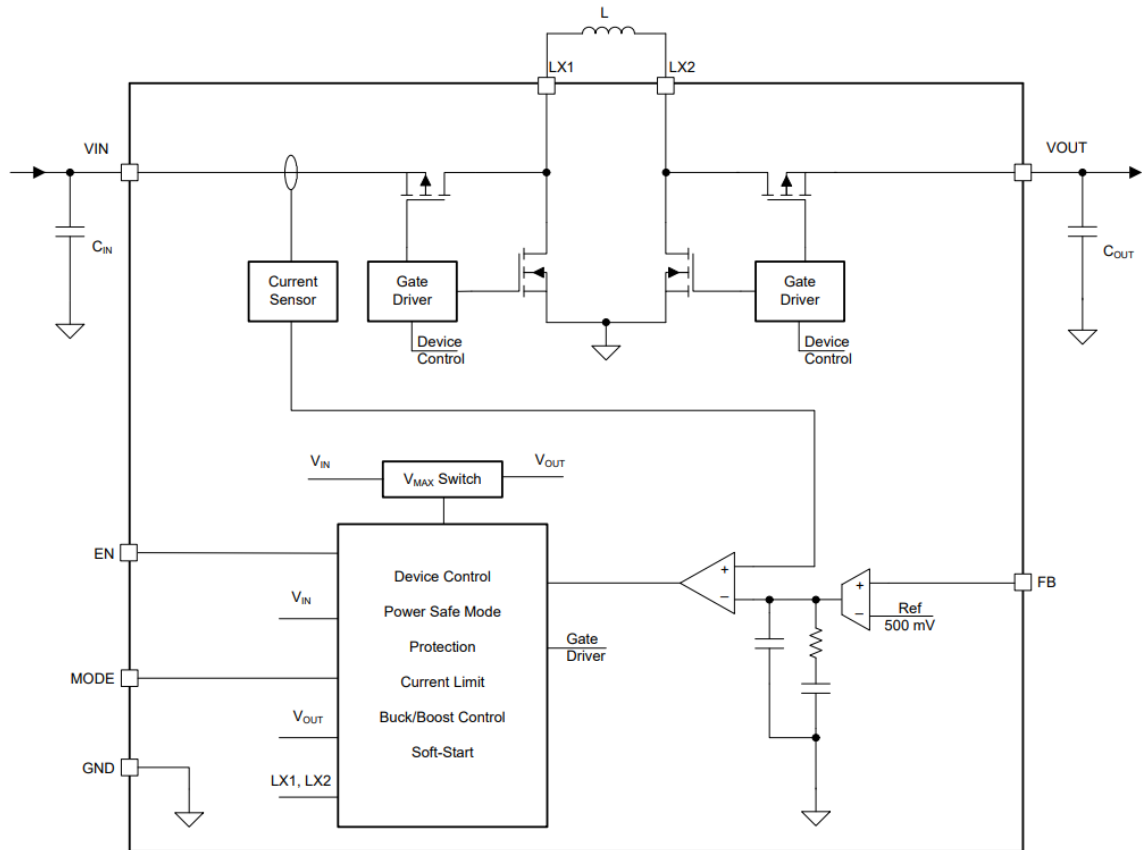
Στο επίκεντρο του κυκλώματος σταθεροποίησης βρίσκεται το ολοκληρωμένο TPS631000. Το TPS631000 είναι buck-boost DC-DC μετατροπέας συνεχούς ρεύματος. Η συσκευή μπορεί να τροφοδοτήσει το κύκλωμα μέχρι και 3A ενώ το εύρος τιμών τάσης που μπορεί να εφαρμοστεί στην είσοδο είναι από 1.6V μέχρι και 5.5V. Το TPS631000 παρέχει στην έξοδο του σταθερή τάση, ανάλογα με τις ανάγκες του κάθε κυκλώματος. Μία σημαντική ιδιαιτερότητα του TPS631000 είναι ότι ανάλογα την τάση εισόδου του, μπορεί να λειτουργήσει σαν boost μετατροπέας κάνοντας step – up την τάση εισόδου, αλλά μπορεί να λειτουργήσει και σαν back μετατροπέας, κάνοντας δηλαδή step – down

μετατροπή της τάσης εισόδου. Αυτές οι μεταβάσεις μεταξύ των δύο λειτουργιών του σταθεροποιητή πραγματοποιούνται σε ένα καθορισμένο κύκλο λειτουργίας, για την ανεπιθύμητη εναλλαγή μεταξύ αυτών των λειτουργιών και την κυμάτωση της τάσης εξόδου. Η συσκευή μπορεί να λειτουργήσει σε εύρος θερμοκρασίας από -40°C έως $+85^{\circ}\text{C}$ και όπως αναφέρθηκε και στην προηγούμενη ενότητα, η θερμοκρασία λειτουργίας είναι πολύ σημαντική στη συγκεκριμένη κατασκευή, καθώς η αισθητήρια συσκευή θα λειτουργεί σε συνθήκες χαμηλής θερμοκρασίας. Ο TPS631000 μπορεί να χρησιμοποιηθεί σε πολλές εφαρμογές οι οποίες απαιτούν τροφοδοσία από μία μπαταρία λιθίου, η οποία δεν έχει σταθερή τάση εξόδου και το κύκλωμα απαιτεί χαμηλές τιμές συνεχούς τάσης για παράδειγμα για την τροφοδοσία ενός μικροελεγκτή. Ο τρόπος λειτουργίας του στη συγκεκριμένη πτυχιακή εργασία είναι παρόμοια. Η μπαταρία που χρησιμοποιούμε είναι μία μπαταρία λιθίου 18650 με μέγιστη τάση 4.2V. Όσο η μπαταρία τροφοδοτεί το κύκλωμα, εκφορτίζεται, με αποτέλεσμα να πέφτει η τάση εξόδου της με συγκεκριμένο ρυθμό. Η τάση αποκοπής της μπαταρίας είναι περίπου τα 2.8V, επομένως είναι εύκολα κατανοητό ότι το εύρος τιμών τάσης που μπορούμε να αξιοποιήσουμε είναι από τα 4.2V μέχρι τα 2.8V. Επίσης το κύκλωμα χρειάζεται 3.3V για να τροφοδοτηθούν ο ESP32 και το κύκλωμα ελέγχου του ρελέ. Όταν το εύρος τιμών τάσης της μπαταρίας κυμαίνεται από τα 4.2V μέχρι και τα 3.3V ο σταθεροποιητής λειτουργεί σαν Back Converter και κάνει χαμήλωμα της τάσης στα 3.3V. Όταν το εύρος τιμών τάσης της μπαταρίας κυμαίνεται από τα 3.3V μέχρι και τα 2.8V ο σταθεροποιητής λειτουργεί σαν Boost Converter και κάνει ανύψωση της τάσης στα 3.3V. Στον πίνακα 2.10 γίνεται πιο ξεκάθαρη η λειτουργία του σταθεροποιητή.

Πίνακας 2.10: Λειτουργία TPS631000 ανάλογα με τη τάση εισόδου

Εύρος Τιμών Τάσης Εισόδου	Λειτουργία TPS631000	Λειτουργία
4.2V – 3.3V	Back Converter	Γίνεται χαμήλωμα της τάσης στα 3.3V
3.3V	---	Η τιμή της τάσης παραμένει ίδια
3.3V – 2.8V	Boost Converter	Γίνεται ανύψωση της τάσης στα 3.3V

Κεφάλαιο 2



Σχήμα 2.14: TPS631000 - Το Μπλοκ Διάγραμμα

Στον πίνακα 2.11 παρατίθενται οι απόλυτα μέγιστες τιμές, ενώ στον πίνακα 2.12 παρατίθενται τα βασικότερα από τα ηλεκτρικά DC χαρακτηριστικά από το επίσημο φύλλο δεδομένων του ολοκληρωμένου.

Πίνακας 2.11: TPS631000 - Μέγιστες Επιτρεπτές Τιμές

		MIN	MAX	UNIT
V_I	Input voltage (VIN, LX1, LX2, VOUT, EN, FB, MODE) ⁽²⁾	-0.3	6	V
	Input voltage for less than 10 ns (LX1, LX2) ⁽²⁾	-0.3	7	V
T_J	Operating junction temperature	-40	150	°C
T_{stg}	Storage temperature	-65	150	°C

Πίνακας 2.12: TPS631000 - Προτεινόμενες Συνθήκες Λειτουργίας

		MIN	NOM	MAX	UNIT	
V_I	Supply voltage	1.6		5.5	V	
V_O	Output voltage	1.2		5.3	V	
C_I	Input capacitance	$V_I = 1.6\text{ V to } 5.5\text{ V}$			μF	
C_O	Output capacitance	$1.2\text{ V} \leq V_O \leq 3.6\text{ V}$, nominal value at $V_O = 3.3\text{ V}$		10.4	16.9	μF
		$3.6\text{ V} < V_O \leq 5.3\text{ V}$, nominal value at $V_O = 5\text{ V}$		7.95	10.6	μF
L	Inductance	0.7	1	1.3	μH	
T_J	Operating junction temperature range	-40		125	°C	

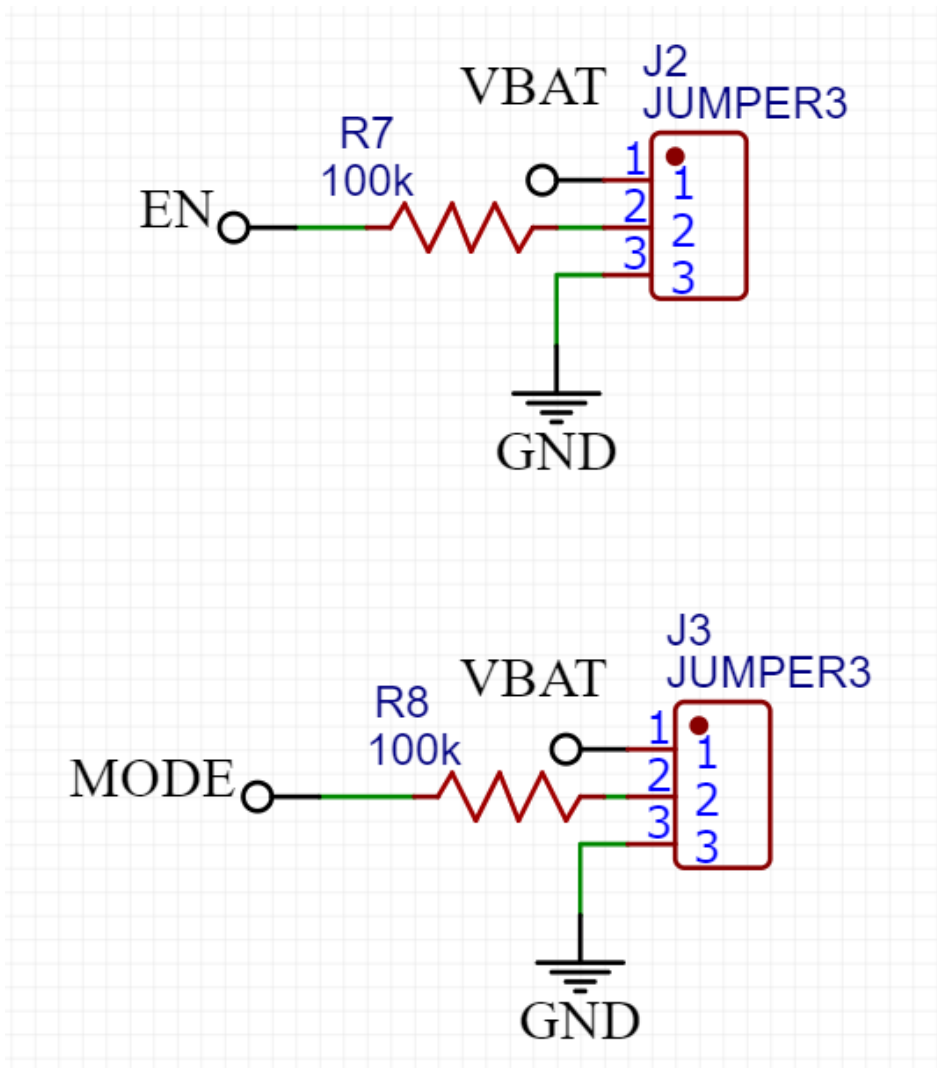
Στον πίνακα 2.13 γίνεται η περιγραφή των εξωτερικών ηλεκτρικών συνδέσεων.

Πίνακας 2.13: TPS631000 - Λειτουργία Εξωτερικών Pin

PIN		I/O	Περιγραφή
Όνομα	Pin No.		
V_{OUT}	1	PWR	Τάση εξόδου
LX2	2	PWR	Κόμβος Μεταγωγής Πηνίου του Σταδίου Ενίσχυσης
LX1	3	PWR	Κόμβος Μεταγωγής Πηνίου του Σταδίου Πτώσης
V_{IN}	4	PWR	Τάση Εισόδου
EN	5	I	Device Enable. Λογικό 1 για ενεργοποίηση και λογικό 0 για απενεργοποίηση.
MODE	6	I	Επιλογή PFM/PWM. Λογικό 1 για εξαναγκασμένη λειτουργία PWM και λογικό 0 για λειτουργία εξοικονόμησης ενέργειας.
GND	7	PWR	Γείωση
FB	8	I	Τάση Feedback

2.5.3 Μέθοδοι Λειτουργίας

Στο σχήμα 2.15 απεικονίζεται η επιλογή που έχει ο χρήστης όσον αφορά την επιλογή για τα pins “EN” και “MODE”. Η επιλογή για το pin “EN” γίνεται μέσω του Jumper J2. Αν ο χρήστης βραχυκυκλώσει την ένωση V_{BAT} και EN, η συσκευή είναι ενεργοποιημένη και πλήρως λειτουργική αφού όλα τα επιμέρους του κυκλώματος τροφοδοτούνται με σταθερή τάση και στο κύκλωμα διαρρέει το απαιτούμενο ρεύμα, ενώ αν ο χρήστης βραχυκυκλώσει την ένωση V_{BAT} και GND, η συσκευή είναι απενεργοποιημένη και δε δουλεύει. Αντίστοιχα, η επιλογή για το pin “MODE” γίνεται μέσω του Jumper J3. Αν ο χρήστης βραχυκυκλώσει την ένωση V_{BAT} και MODE, η συσκευή λειτουργεί σε εξαναγκασμένη λειτουργία PWM ανεξάρτητα από το ρεύμα εξόδου για να εξασφαλιστεί η ελάχιστη κυμάτωση εξόδου. Αν ο χρήστης βραχυκυκλώσει την ένωση V_{BAT} και GND, η συσκευή λειτουργεί σε αυτόματη PFM λειτουργία. Πιο συγκεκριμένα, Ο TPS631000 μπορεί να τεθεί σε λειτουργία εξοικονόμησης ενέργειας, ώστε να διατηρηθεί η υψηλότερη απόδοση σε σχέση με το όλο το εύρος του ρεύματος εξόδου του κυκλώματος.



Σχήμα 2.15: Επιλογή λειτουργίας συσκευής από το χρήστη

2.5.4 Ανάλυση Κυκλώματος

2.5.4.1 Εξωτερικοί Πυκνωτές

Αρχικά, η τάση της μπαταρίας εφαρμόζεται στην είσοδο του κυκλώματος σταθεροποίησης. Αυτή η τάση, όπως αναφέρθηκε και παραπάνω έχει εύρος $4.2V - 2.8V$. Οι 2 εξωτερικοί κεραμικοί πυκνωτές SMD μεγέθους 0805 C3 και C4, είναι υπεύθυνοι για τη βελτίωση της μεταβατικής συμπεριφοράς του σταθεροποιητή και της συμπεριφοράς EMI του συνολικού κυκλώματος τροφοδοσίας. Οι τιμές των δύο πυκνωτών έχουν επιλεγθεί με βάση το φύλλο δεδομένων του TPS631000.

2.5.4.2 Πηνίο

Το πηνίο L1 παρέχει σταθερό ρεύμα στο φορτίο καθώς ο PWM παλμός εναλλάσσει. Ο κύκλος λειτουργίας του PWM επιτρέπει στο χρήστη να ελέγξει την τάση εξόδου του σταθεροποιητή και η τιμή του πηνίου καθορίζει το ρεύμα κυμάτωσης του πηνίου. Όσο μεγαλύτερη είναι η τιμή του πηνίου τόσο μικρότερο είναι το ρεύμα κυμάτωσης και το αντίθετο. Ο κύκλος λειτουργίας σε λειτουργία ενίσχυσης υπολογίζεται από τη σχέση:

$$D = \frac{V_{OUT} - V_{IN}}{V_{OUT}} = \frac{3.3V - 2.8V}{3.3V} \approx 0.15 = 15\% \quad (2.4)$$

Όπου:

$D = Duty\ Cycle$ σε λειτουργία ενίσχυσης

$V_{OUT} =$ Τάση Εξόδου

$V_{IN} =$ Ελάχιστη Τάση Εισόδου

Για να αποφευχθεί ο κορεσμός του πηνίου, πρέπει να επιλεγθεί σωστή τιμή του μέγιστου ρεύματος του πηνίου, το οποίο ισούται με:

$$I_{PEAK} = \frac{I_{OUT}}{n(1-D)} + \frac{V_{IN} * D}{2 * f * L} = \frac{160mA}{0.9(1-0.15)} + \frac{2.8 * 0.15}{2 * 2.2 * 10^6 * 10^{-6}} \sim 0.3A \quad (2.5)$$

Όπου:

$f =$ Συχνότητα σταθεροποιητή (τυπικά 2.2 MHz)

$L =$ Τιμή πηνίου

$n =$ εκτιμώμενη απόδοση μετατροπέα (Σύμφωνα με το φύλλο δεδομένων ισούται με 0.9) [19]

2.5.4.3 Διαιρέτης τάσης

Τέλος, οι τέσσερις αντιστάσεις καθορίζουν τάση εξόδου του μετατροπέα. Ο διαιρέτης τάσης συνδέεται ανάμεσα στα pins V_{OUT} , FB και GND. Εδώ αξίζει να σημειωθεί ότι ο διαιρέτης τάσης θα υπολογισθεί παρακάτω με 2 αντιστάσεις. Ο λόγος που στη συγκεκριμένη κατασκευή χρησιμοποιήθηκαν τέσσερις αντιστάσεις (2 + 2 σε σειρά), είναι επειδή οι δύο παρακάτω τιμές που θα υπολογιστούν δεν ήταν δυνατό να αγοραστούν και για οικονομία χρόνου και χρήματος χρησιμοποιούνται αντιστάσεις που προϋπήρχαν στην κατοχή μου.

Η τιμή των αντιστάσεων R11 και R12 παραμένουν σταθερές. Παρακάτω υπολογίζεται η συνολική τιμή των αντιστάσεων R9 και R10. Προκειμένου να αντισταθμίσουμε τις αιχμές τάσης που παρουσιάζονται κατά την εκκίνηση της συσκευής και για να μη ζοριστεί ο σταθεροποιητής σε περίπτωση που το κύκλωμα απαιτεί παραπάνω ρεύμα από το μέσο ρεύμα, το άθροισμα των δύο, θα υπολογιστεί με τάση εξόδου $V_{OUT} = 3,35V$.

$$R_1 = R_2 \left(\frac{V_{OUT}}{V_{FB}} - 1 \right) = 91k \left(\frac{3.35V}{0.5V} - 1 \right) = 518.7k\Omega \quad (2.6)$$

Όπου:

$$R_1 = R_9 + R_{10}$$

$$R_2 = R_{11} + R_{12}$$

Κεφάλαιο 2

$$V_{FB} = 0.5V$$

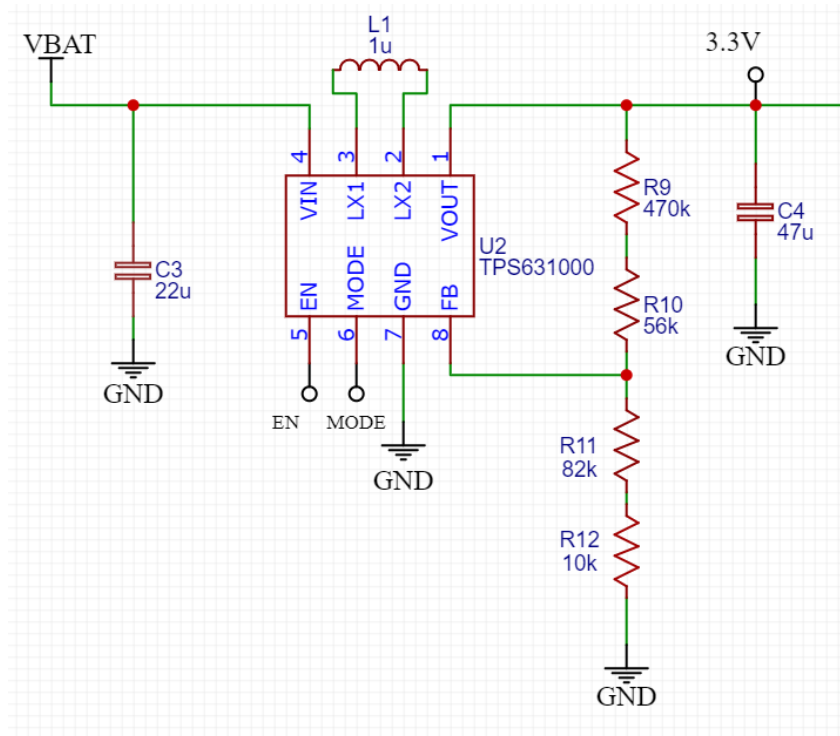
Σύμφωνα με τη παραπάνω σχέση και τη τιμή της αντίστασης που υπολογίσθηκε, οι τιμές που επιλέχθηκαν για τις αντιστάσεις R9 και R10 θα ισούνται με 470 kΩ και 56 kΩ. Οι δύο αυτές αντιστάσεις συνδεδεμένες σε σειρά, βγάζουν ένα τελικό αποτέλεσμα

$$R_2 = R_9 + R_{10} = 470k\Omega + 56k\Omega = 526k\Omega \quad (2.7)$$

αποτέλεσμα που είναι αποδεκτό. Η αντίσταση R2 αναφέρθηκε προηγουμένως ότι έχει σταθερή τιμή 92 kΩ, επομένως οι τιμές των αντιστάσεων R11 και R12 ισούνται με:

$$R_2 = R_{11} + R_{12} = 82k\Omega + 10k\Omega = 92k\Omega \quad (2.8)$$

Στα σχήματα 2.15 και 2.16 απεικονίζεται συνολικά το κύκλωμα σταθεροποίησης



Σχήμα 2.16: Το κύκλωμα σταθεροποίησης της αισθητήριας συσκευής

2.6 Το Κύκλωμα Ρελέ

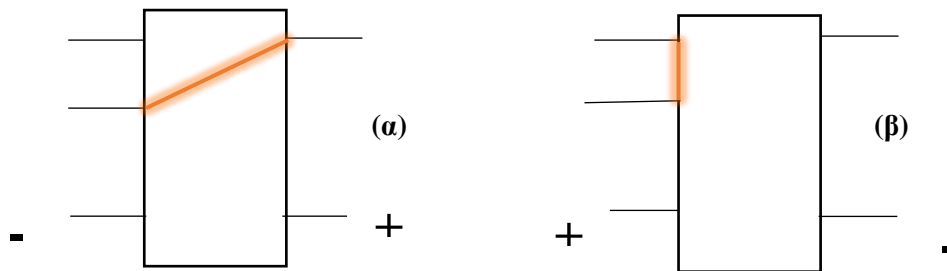
2.6.1 Εισαγωγή

Το κύκλωμα ρελέ είναι υπεύθυνο για τη λειτουργία και το χειρισμό του δισταθούς ρελέ, το οποίο οπλίζει και αποπλίζει ανάλογα με τη λειτουργία της αισθητήριας συσκευής, που είναι τοποθετημένη σε κάθε μία από τις ψυκτικές μονάδες ξεχωριστά. Το κύκλωμα είναι στην ουσία μία συστοιχία τεσσάρων τρανζίστορ που λειτουργούν ως διακόπτες αλλάζοντας την πολικότητα της τάσης κάθε φορά που το ρελέ πρέπει να αλλάξει κατάσταση και ο σχεδιασμός του είναι ένα “H-Bridge”. Αλλάζοντας έτσι κατάσταση το ρελέ, ο συνδεδεμένος στην έξοδο του ανεμιστήρας της κάθε ψυκτικής μονάδας ανοίγει και κλείνει για να ψύξει το εσωτερικό της θαλάμης, εφόσον οι συνθήκες το απαιτούν. Το κύκλωμα είναι μελετημένο και κατασκευασμένο έτσι ώστε να εξυπηρετεί δύο σκοπούς: Εξοικονόμηση ενέργειας και ορθή λειτουργία του κυκλώματος, ακόμα και όταν ο επεξεργαστής βρίσκεται σε λειτουργία ύπνου.

2.6.2 PE014F03

Στο επίκεντρο του κυκλώματος είναι το δισταθές ρελέ PE014F03. Ένα δισταθές ρελέ διαφέρει αρκετά από ένα μονοσταθές ρελέ και η βασική τους διαφορά είναι ο τρόπος με τον οποίο οπλίζουν και αποπλίζουν. Ένα μονοσταθές ρελέ απαιτεί μόνιμα να εφαρμόζεται μία τιμή τάσης στην είσοδο του για να παραμένει το ρελέ ανοιχτό (ή κλειστό ανάλογα με το είδος του ρελέ). Αν αυτή η τάση σταματήσει να εφαρμόζεται στο ρελέ τότε αυτό, θα αλλάξει κατάσταση. Επίσης αυτή η τάση εφαρμόζεται στην είσοδο του ρελέ με συγκεκριμένη πολικότητα. Από την άλλη μεριά, ένα δισταθές ρελέ δεν απαιτεί μόνιμως μια τιμή τάσης να εφαρμόζεται στην είσοδό του, έτσι ώστε να παραμείνει σε μία συγκεκριμένη κατάσταση. Αντιθέτως, για να αλλάξει κατάσταση αρκεί να εφαρμοστεί στιγμιαία η ονομαστική τιμή της τάσης με συγκεκριμένη πολικότητα. Αυτό σημαίνει ότι όταν μία τιμή τάσης εφαρμόζεται στο ρελέ αυτό αλλάζει κατάσταση και όταν εφαρμόζεται η ίδια τιμή τάσης στο ίδιο ρελέ αλλά με διαφορετική πολικότητα, τότε θα αλλάξει ξανά κατάσταση. Η συγκεκριμένη λειτουργία δουλεύει ως εξής: Το ρελέ διαθέτει στο εσωτερικό του ένα πηνίο και έναν μαγνήτη. Όταν εφαρμόζεται στο ρελέ τάση με συγκεκριμένη πολικότητα τότε το ρεύμα που διαρρέει αυτό το πηνίο δημιουργεί ένα ηλεκτρομαγνητικό πεδίο το οποίο ενεργοποιεί το μαγνήτη. Όταν η πολικότητα της τάσης στην είσοδο του ρελέ αλλάξει, τότε στο πηνίο διαρρέει αντίθετης κατεύθυνσης ρεύμα, το οποίο μαγνητίζει τον μαγνήτη από την αντίθετη μεριά και αυτό έχει ως συνέπεια να αλλάζει η κατάστασή του ρελέ. Το πλεονέκτημα σε αυτή τη λειτουργία είναι το γεγονός ότι το ρελέ καταναλώνει ενέργεια μόνο όταν θα εφαρμοστεί αυτή η στιγμιαία τάση και θα αλλάξει η κατάσταση του, επομένως το κύκλωμα απαιτεί πολύ λιγότερο ρεύμα για να λειτουργήσει σε σύγκριση με ένα μονοσταθές ρελέ, που πρέπει να διαρρέει ρεύμα μόνιμα, για να παραμείνει το ρελέ στην επιθυμητή κατάστασή του. Ένα ακόμα πλεονέκτημα του δισταθούς ρελέ που χρησιμοποιείται στη συγκεκριμένη κατασκευή είναι ότι δε θερμαίνεται, αφού καταναλώνει ελάχιστα ποσά ενέργειας. Το συγκεκριμένο είναι ιδιαιτέρως σημαντικό πλεονέκτημα, αν σκεφτεί κανείς ότι η κατασκευή στη παρούσα πτυχιακή είναι στην ουσία ένας θερμοστάτης, ο οποίος για να λειτουργεί σωστά και να μετράει όσο το δυνατόν ακριβέστερες μετρήσεις δε πρέπει να επηρεάζεται από τα γύρω υλικά που εκπέμπουν θερμότητα. Τέλος, όπως αναφέρθηκε και στην προηγούμενη παράγραφο, το συγκεκριμένο ρελέ είναι ιδανικό για μία βασική λειτουργία του μικροελεγκτή της αισθητήριας συσκευής. Η λειτουργία “βαθύ” ύπνου, επιτρέπει τον ESP32 να βρίσκεται σε μία κατάσταση, στην οποία καταναλώνει ελάχιστα ποσά ενέργειας, καθώς απενεργοποιούνται βασικές λειτουργίες και περιφερειακά του ελεγκτή. Αν χρησιμοποιούσαμε ένα μονοσταθές ρελέ τότε κάθε φορά που ο ESP32 βρισκόταν σε λειτουργία βαθύ” ύπνου και το ρελέ ήταν κλειστό (δηλαδή ο ανεμιστήρας της ψυκτικής

μονάδας λειτουργεί), τότε το ρελέ θα άλλαζε κατάσταση ακόμα και αν οι συνθήκες της θαλάμης δεν το επέτρεπαν, αφού σε αυτή τη λειτουργία ο ελεγκτής σταματάει να λειτουργεί και δεν θα τροφοδοτεί πια το ρελέ για να παραμένει κλειστό. Χρησιμοποιώντας έτσι το δισταθές ρελέ, ακόμα και όταν ESP32 βρίσκεται σε λειτουργία βαθύ ύπνου, ο ανεμιστήρας θα παραμείνει ανοιχτός, εφόσον η θερμοκρασία στο εσωτερικό της θαλάμης είναι μεγαλύτερη από την τιμή κατωφλίου που έχει θέσει ο χρήστης. Στο σχήμα 2.17 φαίνεται η λειτουργία του ρελέ στις δύο περιπτώσεις, ανάλογα με την πολικότητα της τάσης που εφαρμόζεται στο ρελέ.



Σχήμα 2.17: PE014F03 - Οι δύο καταστάσεις του

Στον πίνακα 2.14, παρατίθενται τα σημαντικότερα ηλεκτρονικά χαρακτηριστικά του ρελέ PE014F03. Το κείμενο υπογραμμισμένο με κίτρινο είναι ο κωδικός του ρελέ που χρησιμοποιείται για τη κατασκευή.

Πίνακας 2.14: PE014F03 - DC χαρακτηριστικά

Coil Data						
Magnetic system		bistable, polarized				
Coil voltage range		2.2 to 48VDC				
Operative range, IEC 61810		2				
Reset voltage max., % of rated coil voltage		120% at -40°C				
Min./Max. energization duration		20ms ¹⁾ /1min at <10% duty factor				
1) Information on reduced pulse duration with higher energization voltages on demand.						
Coil versions, bistable 1 coil						
Coil code ²⁾		Rated voltage VDC	Set voltage VDC	Reset voltage VDC	Coil resistance Ω±10%	Rated coil power mW
F02	H02	2.2	1.65	1.65	22	220
F03	H03	3	2.25	2.25	41	220
F05	H05	5	3.75	3.75	125	200
F06	H06	6	4.5	4.5	180	200
F12	H12	12	9.0	9.0	650	222
F24	H24	24	18.0	18.0	2750	209

²⁾ Coil codes F. and H..have opposite polarity; refer to coil operation table.

All figures are given for coil without pre-energization, at ambient temperature +23°C.

Other coil voltages on request.

2.6.3 Ανάλυση Κυκλώματος

Το κύκλωμα ελέγχεται από τον ESP32 μέσω του παλμού που δίνει από τα GPIO25 και GPIO26(σχήματα 2.18 – 2.21). Ο παλμός αυτός ελέγχεται και φτάνει στις εισόδους του ρελέ για να το

οπλίσει ή αφοπλίσει με τη χρήση τεσσάρων τρανζίστορ που λειτουργούν ως διακόπτες. Τα δύο τρανζίστορ Q1 και Q2 στο πάνω μέρος του κυκλώματος είναι δύο PNP τρανζίστορ, ενώ τα δύο τρανζίστορ Q3 και Q4 στο κάτω μέρος του κυκλώματος είναι δύο NPN τρανζίστορ. Για τα δύο PNP τρανζίστορ χρησιμοποιήθηκε το τρανζίστορ BC507, ενώ για τα 2 NPN τρανζίστορ, χρησιμοποιήθηκε το BC817. Στους παρακάτω πίνακες παρατίθενται τα τεχνικά χαρακτηριστικά των δύο τρανζίστορ από τα φύλλα δεδομένων του κατασκευαστή αντίστοιχα.

Πίνακας 2.15: BC807 - Μέγιστες Επιτρεπτές Τιμές

Rating	Symbol	Value	Unit
Collector – Emitter Voltage	V_{CEO}	-45	V
Collector – Base Voltage	V_{CBO}	-50	V
Emitter – Base Voltage	V_{EBO}	-5.0	V
Collector Current – Continuous	I_C	-500	mAdc

Πίνακας 2.16: BC807 - DC Χαρακτηριστικά

ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise noted.)

Characteristic	Symbol	Min	Typ	Max	Unit
OFF CHARACTERISTICS					
Collector – Emitter Breakdown Voltage ($I_C = -10\text{ mA}$)	$V_{(BR)CEO}$	-45	-	-	V
Collector – Emitter Breakdown Voltage ($V_{EB} = 0, I_C = -10\ \mu\text{A}$)	$V_{(BR)CES}$	-50	-	-	V
Emitter – Base Breakdown Voltage ($I_E = -1.0\ \mu\text{A}$)	$V_{(BR)EBO}$	-5.0	-	-	V
Collector Cutoff Current ($V_{CB} = -20\text{ V}$) ($V_{CB} = -20\text{ V}, T_J = 150^\circ\text{C}$)	I_{CBO}	-	-	-100 -5.0	nA μA
ON CHARACTERISTICS					
DC Current Gain ($I_C = -100\text{ mA}, V_{CE} = -1.0\text{ V}$) ($I_C = -500\text{ mA}, V_{CE} = -1.0\text{ V}$)	h_{FE}	160 250 40	- - -	400 600 -	-
Collector – Emitter Saturation Voltage ($I_C = -500\text{ mA}, I_B = -50\text{ mA}$)	$V_{CE(sat)}$	-	-	-0.7	V
Base – Emitter On Voltage ($I_C = -500\text{ mA}, V_{CE} = -1.0\text{ V}$)	$V_{BE(on)}$	-	-	-1.2	V

Πίνακας 2.17: BC807 – Μέγιστες Επιτρεπτές Τιμές

Absolute Maximum Ratings (@ $T_A = +25^\circ\text{C}$, unless otherwise specified.)

Characteristic	Symbol	Value	Unit
Collector-Base Voltage	V_{CBO}	50	V
Collector-Emitter Voltage	V_{CEO}	45	V
Emitter-Base Voltage	V_{EBO}	5.0	V
Collector Current	I_C	0.5	A
Peak Collector Current	I_{CM}	1.0	A
Peak Base Current	I_{BM}	200	mA

Πίνακας 2.18: BC817 – Ηλεκτρικά Χαρακτηριστικά

Electrical Characteristics (@T _A = +25°C, unless otherwise specified.)						
Characteristic	Symbol	Min	Typ	Max	Unit	Test Condition
Collector-Base Breakdown Voltage	BV _{CB0}	50	—	—	V	I _C = 100μA
Collector-Emitter Breakdown Voltage	BV _{CEO}	45	—	—	V	I _C = 10mA
Emitter-Base Breakdown Voltage	BV _{EBO}	5	—	—	V	I _C = 100μA
Collector-Emitter Cut-Off Current	I _{CES}	—	—	100 5.0	nA μA	V _{CE} = 45V V _{CE} = 25V, T _J = +150°C
Emitter-Base Cut-Off Current	I _{EBO}	—	—	100	nA	V _{EB} = 5.0V
DC Current Gain (Note 9)	BC817-16 BC817-25 BC817-40	h _{FE}	—	100 160 250	—	V _{CE} = 1.0V, I _C = 100mA
	BC817-16 BC817-25 BC817-40			60 100 170		
Collector-Emitter Saturation Voltage (Note 9)	V _{CE(SAT)}	—	—	0.7	V	I _C = 500mA, I _B = 50mA
Base-Emitter Voltage (Note 9)	V _{BE}	—	—	1.2	V	V _{CE} = 1.0V, I _C = 300mA
Gain Bandwidth Product	f _T	100	—	—	MHz	V _{CE} = 5.0V, I _C = 10mA, f = 50MHz
Collector-Base Capacitance	C _{CB0}	—	—	12	pF	V _{CB} = 10V, f = 1.0MHz

Note: 9. Measured under pulsed conditions. Pulse width ≤ 300μs. Duty cycle ≤ 2%.

Οι δύο διόδοι D1 και D2 είναι διόδοι προστασίας των τρανζίστορ και παρέχουν μια ασφαλή διαδρομή του ρεύματος που επιστρέφει από το ρελέ. Χωρίς αυτές πολύ πιθανόν τα τρανζίστορ να καιγόντουσαν. Για τους σκοπούς της κατασκευής έχει επιλεγθεί το BAV99, το οποίο είναι ένα smd υλικό που στο εσωτερικό του έχει δύο διόδους μεταγωγής σήματος συνδεδεμένες σε σειρά. Στους πίνακες 2.19 και 2.20, παρατίθενται τα τεχνικά χαρακτηριστικά από το φύλλο δεδομένων του κατασκευαστή.

Πίνακας 2.19: BAV99 – Μέγιστες Επιτρεπτές Τιμές

ABSOLUTE MAXIMUM RATINGS (T _{amb} = 25 °C, unless otherwise specified)				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
Non repetitive peak reverse voltage		V _{RM}	100	V
Repetitive peak reverse voltage = working peak reverse voltage = DC blocking voltage		V _R = V _{RRM} = V _{RWM} = V _R	70	
Peak forward surge current	t _p = 1 s	I _{FSM}	1	A
	t _p = 1 μs		4.5	
Average forward current	Half wave rectification with resistive load and f ≥ 50 MHz, on ceramic substrate 10 mm x 8 mm x 0.7 mm	I _{F(AV)}	150	mA
Forward current	On ceramic substrate 10 mm x 8 mm x 0.7 mm	I _F	250	
Power dissipation	On ceramic substrate 10 mm x 8 mm x 0.7 mm	P _{tot}	300	mW

Πίνακας 2.20: BAV99 - DC Χαρακτηριστικά

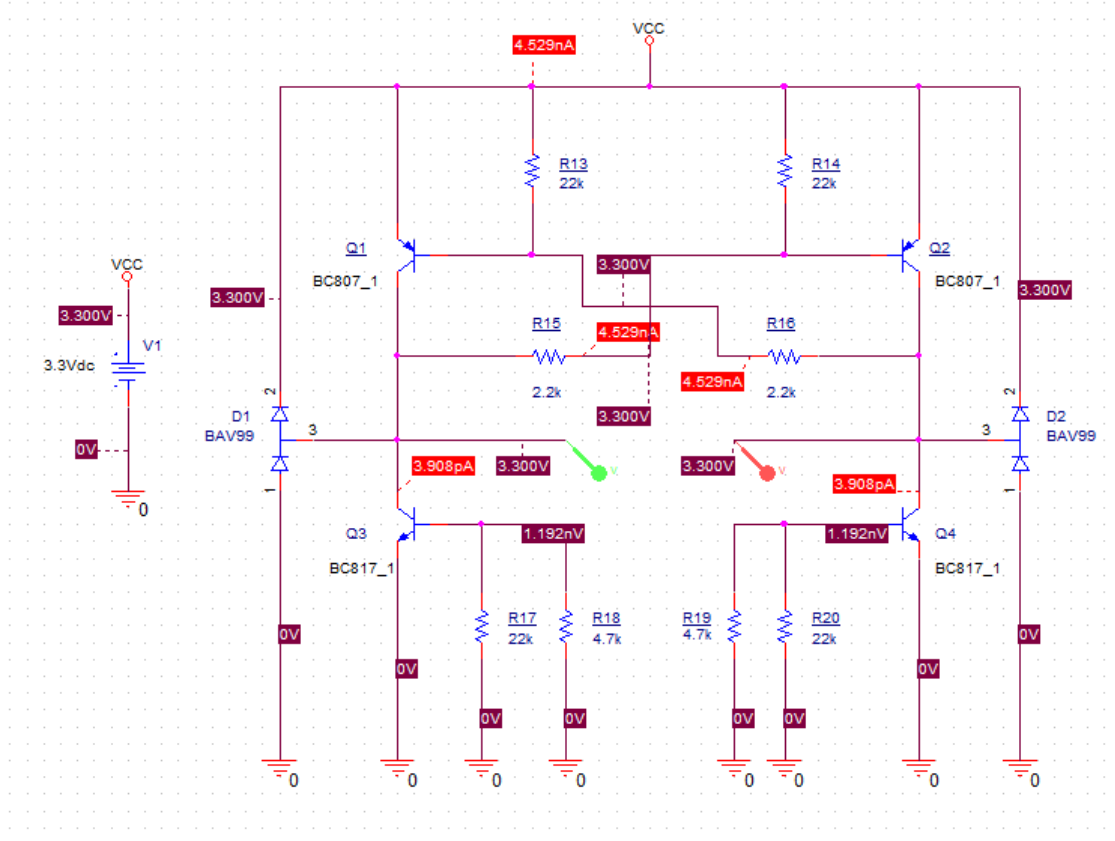
ELECTRICAL CHARACTERISTICS (T _{amb} = 25 °C, unless otherwise specified)						
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
Forward voltage	I _F = 1 mA	V _F			0.715	V
	I _F = 10 mA				0.855	V
	I _F = 50 mA				1	V
	I _F = 150 mA				1.25	V
Reverse current	V _R = 70 V	I _R			2500	nA
	V _R = 70 V, T _J = 150 °C				50	μA
	V _R = 25 V, T _J = 150 °C				30	μA
Diode capacitance	V _R = 0, f = 1 MHz	C _D			1.5	pF
Reverse recovery time	I _F = 10 mA to i _R = 1 mA, V _R = 6 V, R _L = 100 Ω	t _{rr}			6	ns

Αφού το κύκλωμα ελέγχεται από δύο καταστάσεις εισόδων τότε υπάρχουν $2^n = 4$ περιπτώσεις για τις δύο αυτές καταστάσεις. Παρακάτω εξηγείται και αναλύεται το κύκλωμα για κάθε μία από αυτές τις καταστάσεις. Επίσης, έχει γίνει προσομοίωση του κυκλώματος ρελέ στο λογισμικό **OrCAD PSPICE A/D**, το οποίο είναι ένα πρόγραμμα προσομοίωσης ηλεκτρονικών κυκλωμάτων που στόχο έχει να βοηθήσει το χρήστη να σχεδιάσει, να ελέγξει και να βελτιώσει το κύκλωμα χωρίς να χρειαστεί να το κατασκευάσει.

Περίπτωση 1:

$GPIO25 = 0$ $GPIO26 = 0$

Σε αυτήν την περίπτωση τα 2 τρανζίστορ Q3 και Q4 βρίσκονται σε αποκοπή ενώ το τρανζίστορ Q1 και Q2 άγουν, αφού στη βάση των 2 τρανζίστορ εφαρμόζεται πάντα χαμηλότερο δυναμικό από αυτό που εφαρμόζεται στον εκπομπό. Αυτό σημαίνει ότι και στις 2 συνδέσεις του ρελέ εφαρμόζεται τάση $V_{A1} = V_{A2} = 3.3V$. Όταν συμβαίνει αυτό το ρελέ δεν αλλάζει σε κάποια συγκεκριμένη κατάσταση αλλά παραμένει στην κατάσταση που βρισκόταν προηγουμένως. Στην περίπτωση αυτή θα βρισκεται το κύκλωμα το μεγαλύτερο χρονικό διάστημα, επειδή αυτή η κατάσταση είναι και η κατάσταση ηρεμίας. Παρακάτω φαίνεται η σχεδίαση και προσομοίωση του κυκλώματος ρελέ καθώς και οι τάσεις που εφαρμόζονται και τα ρεύματα που διαρρέουν στις συνδέσεις του κυκλώματος.



Σχήμα 2.18: Προσομοίωση του κυκλώματος ρελέ στο PSPICE στην κατάσταση 1

Περίπτωση 2:

$$GPIO25 = 0 \quad GPIO26 = 1$$

Σε αυτήν τη περίπτωση η τάση των 3.3V εφαρμόζεται στην αντίσταση R18 και το Q4 άγει. Η τάση της βάσης του τρανζίστορ ισούται με $V_B = 0.7V$. Η αντίσταση R19 ρυθμίζει το ρεύμα της βάσης του τρανζίστορ Q4 και ισούται με:

$$I_B = \frac{V_{CC} - V_{BE}}{R_{19}} = \frac{3.3 - 0.7}{4.7k} \sim 553\mu A \quad (2.9)$$

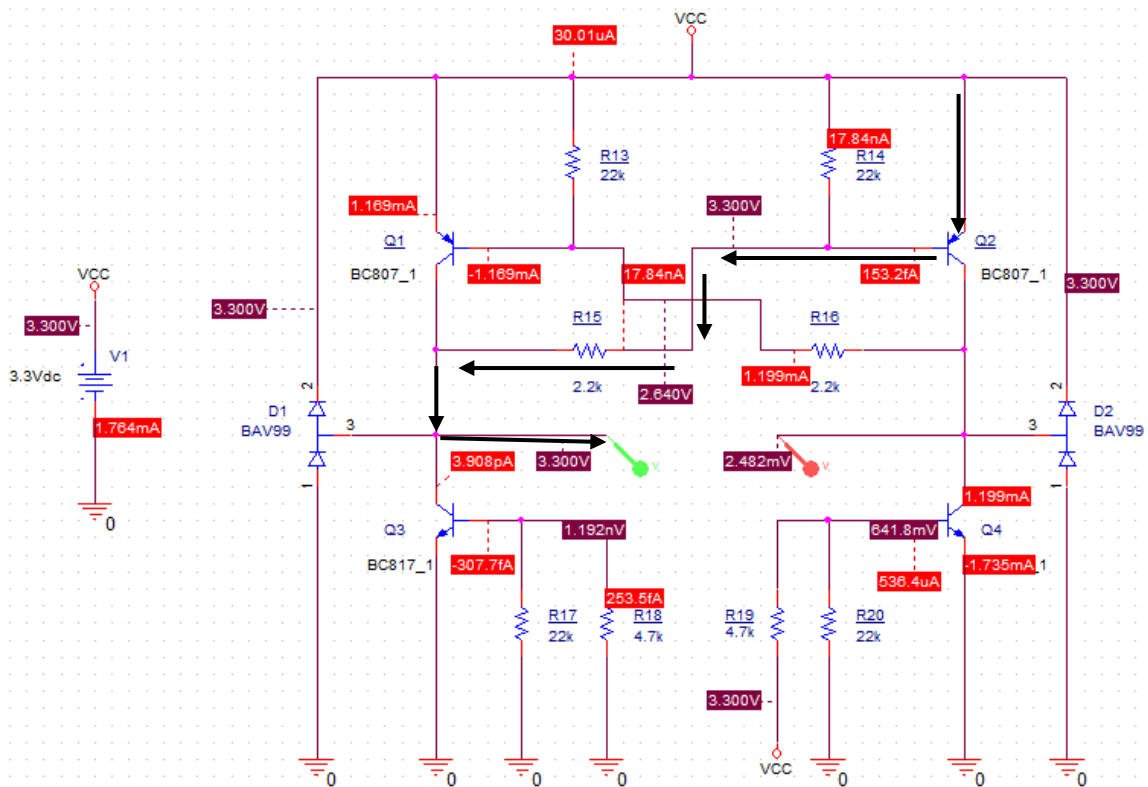
Το τρανζίστορ Q1 επίσης άγει και η τάση στη βάση του ισούται με

$$V_B = 3.3 - 0.7V = 2.6V \quad (2.10)$$

Επομένως, το ρεύμα που διαρρέει την αντίσταση R16 είναι ίσο με το ρεύμα συλλέκτη του Q4 και ισούται με:

$$I = \frac{V}{R_{16}} = \frac{2.6V}{2.2k} \sim 1.2mA \quad (2.11)$$

Αφού το Q4 άγει, τότε το χαμηλό δυναμικό των 0V από τη γείωση, εφαρμόζεται στην επαφή A2 του ρελέ K1 και επομένως η επαφή βρίσκεται σε λογικό 0. Αντιθέτως, το τρανζίστορ Q3 βρίσκεται σε αποκοπή, αφού δεν εφαρμόζεται τάση στη βάση του. Αυτό σημαίνει ότι επειδή το τρανζίστορ Q1 άγει τότε στην επαφή A1 εφαρμόζονται $V_{A1} = 3.3V$, επομένως η επαφή βρίσκεται σε λογικό 1. Όταν συμβαίνει αυτό, το ρελέ βρίσκεται στην κατάσταση β του σχήματος 2.17. Τα βέλη στο σχήμα 2.19 δείχνουν την πορεία του ρεύματος στη συγκεκριμένη κατάσταση.



Σχήμα 2.19: Προσομοίωση του κυκλώματος ρελέ στο PSpice στη κατάσταση 2

Περίπτωση 3:

$$GPIO25 = 1 \quad GPIO26 = 0$$

Σε αυτήν τη περίπτωση η τάση των 3.3V εφαρμόζεται στην αντίσταση R_{19} και το Q3 άγει. Η τάση της βάσης του τρανζίστορ ισούται με $V_B = 0.7V$. Η αντίσταση R_{18} ρυθμίζει το ρεύμα της βάσης του τρανζίστορ Q3 και ισούται με:

$$I_B = \frac{V_{CC} - V_{BE}}{R_{18}} = \frac{3.3 - 0.7}{4.7k} \sim 553\mu A \quad (2.12)$$

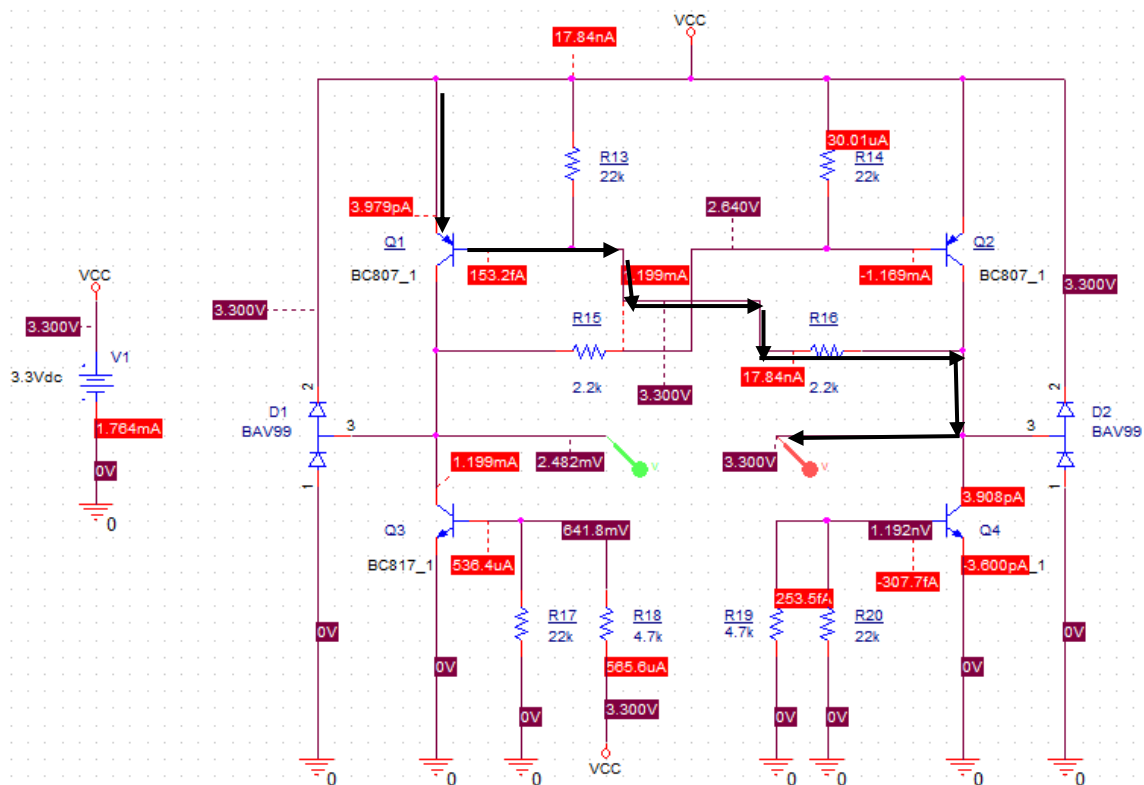
Το τρανζίστορ Q2 επίσης άγει και η τάση στη βάση του ισούται με

$$V_B = 3.3 - 0.7V = 2.6V \quad (2.13)$$

Επομένως, το ρεύμα που διαρρέει την αντίσταση R_{15} είναι ίσο με το ρεύμα συλλέκτη του Q3 και ισούται με:

$$I = \frac{V}{R_{15}} = \frac{2.6V}{2.2k} \sim 1.2mA \quad (2.14)$$

Αφού το Q3 άγει, τότε το χαμηλό δυναμικό των 0V από τη γείωση, εφαρμόζεται στην επαφή A1 του ρελέ K1 και επομένως η επαφή βρίσκεται σε λογικό 0. Αντιθέτως, το τρανζίστορ Q4 βρίσκεται σε αποκοπή, αφού δεν εφαρμόζεται τάση στη βάση του. Αυτό σημαίνει ότι επειδή το τρανζίστορ Q1 άγει τότε στην επαφή A2 εφαρμόζονται $V_{A2} = 3.3V$, επομένως η επαφή βρίσκεται σε λογικό 1. Όταν συμβαίνει αυτό, το ρελέ βρίσκεται στην κατάσταση α του σχήματος 2.21. Τα βέλη στο σχήμα 2.20 δείχνουν την πορεία του ρεύματος στη συγκεκριμένη κατάσταση.

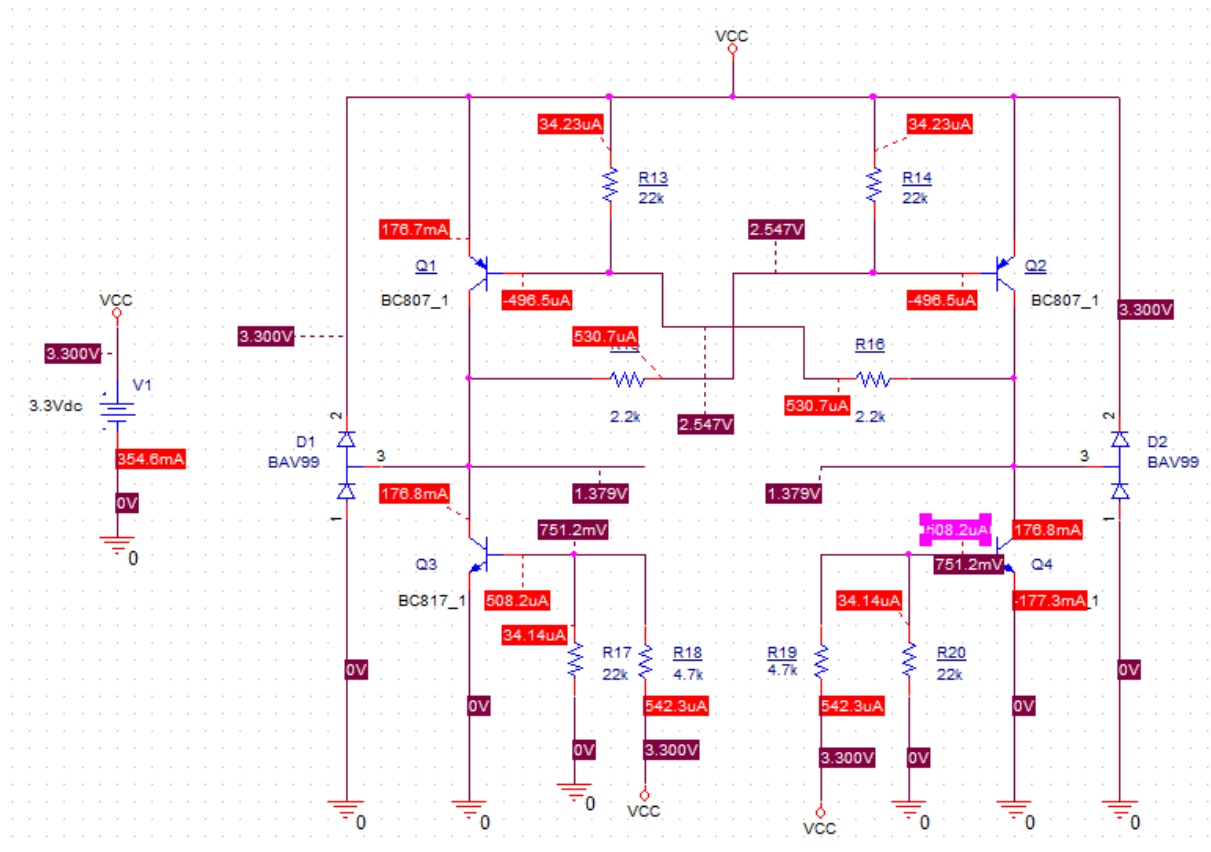


Σχήμα 2.20: Προσομοίωση του κυκλώματος ρελέ στο PSPICE στη κατάσταση 3

Περίπτωση 4:

$$GPIO25 = 1 \quad GPIO26 = 1$$

Σε αυτήν την περίπτωση και τα 4 τρανζίστορ άγουν επομένως έχουμε βραχυκύκλωμα, γιατί θα υπάρξει απευθείας ένωση της τροφοδοσίας με τη γείωση και η μόνη αντίσταση μεταξύ τους θα είναι η ελάχιστη αντίσταση εκπομπού-συλλέκτη των 4 τρανζίστορ. Σε μια τέτοια κατάσταση το πιθανότερο είναι ότι τα τρανζίστορ θα καταστραφούν επομένως σε καμία περίπτωση δεν πρέπει οι 2 επαφές GPIO25 και GPIO26 να βρεθούν ταυτόχρονα σε λογικό 1.



Σχήμα 2.21: Προσομοίωση του κυκλώματος ρελέ στο PSPICE στη κατάσταση 4

2.7 Σχεδίαση και Κατασκευή Πλαστικού Καλούπιού

Σε αυτήν την ενότητα θα γίνει ανάλυση του τρόπου τύπωσης του πλαστικού καλούπιού της κάθε αισθητήριας συσκευής. Το πλαστικό τυπώθηκε στο λογισμικό Fusion 360 (έκδοση 2.0.16009). Το Fusion 360 είναι μια πλατφόρμα λογισμικού τρισδιάστατης μοντελοποίησης, για τον επαγγελματικό σχεδιασμό και κατασκευή προϊόντων. Το είδος του υλικού που χρησιμοποιήθηκε για την εκτύπωση του πλαστικού είναι πολυγαλακτικό οξύ (PLA). Η επιλογή του υλικού δεν είναι τυχαία, καθώς το PLA είναι τεχνικά "ουδέτερο ως προς το διοξείδιο του άνθρακα", προέρχεται από ανανεώσιμες πηγές απορρόφησης άνθρακα και δεν εκπέμπει τοξικά αέρια όταν αποτεφρώνεται. Με αυτόν τον τρόπο η συσκευή συμβάλλει με έναν ακόμη τρόπο στη μείωση των εκπομπών αερίων θερμοκηπίου. Ο εκτυπωτής που χρησιμοποιήθηκε είναι ο Creality3d ender 3. Το καλούπι συνολικά αποτελείται από δύο σχέδια, τα οποία κουμπώνουν μεταξύ τους. Το ένα είναι το κυρίως σώμα, στο οποίο κουμπώνει η πλακέτα σε ένα συρόμενο διάδρομο με δύο εσοχές, για να μπαίνει η πλακέτα. Το κύριο σώμα, διαθέτει τέσσερις τρύπες για να είναι εκτεθειμένα τα 4 περιφερειακά που είναι κολλημένα πάνω στην πλακέτα. Ο πρώτος είναι ο αισθητήρας θερμοκρασίας, η δεύτερη είναι η θύρα φόρτισης USB, η τρίτη είναι η μαγνητική επαφή και η τέταρτη είναι η κλέμα στην οποία συνδέεται ο ανεμιστήρας και τα 12V. Το δεύτερο κομμάτι, είναι το πλαστικό καπάκι που εφάπτεται πάνω στο κυρίως σώμα του πλαστικού και κουμπώνει για να μην ανοίγει το πλαστικό και εκτίθεται η πλακέτα. Το καπάκι κουμπώνει πάνω στο κυρίως σώμα, μέσω 2 προεξοχών που μπαίνουν μέσα σε δυο τρύπες, που είναι τοποθετημένες στο κύριο σώμα. Το κόστος κυμαίνεται περίπου στα 2 ευρώ ανά συσκευή.

Στο σχήμα 2.22 απεικονίζονται τα δύο μέρη του πλαστικού καλούπιού και η πλακέτα, η οποία είναι στερεωμένη στο κύριο σώμα.



Σχήμα 2.22: Το πλαστικό καλούπι και η πλακέτα

Στο σχήμα 2.23 απεικονίζονται δύο εικόνες. Στην πρώτη (πάνω) φαίνεται η συσκευή από πάνω ενώ στη δεύτερη (κάτω) φαίνεται η συσκευή από τα πλάγια. Απεικονίζονται οι τέσσερις τρύπες που άνοιξαν για τις 4 περιφερειακές συνδέσεις



Σχήμα 2.23: Η συσκευή από πάνω και από τα πλάγια

2.8 Επίλογος

Σε αυτό το κεφάλαιο καλύφτηκε οτιδήποτε αφορά το υλικό και τα εξαρτήματα της κατασκευής. Έγινε μία εισαγωγή στην αισθητήρια συσκευή και τον τρόπο λειτουργίας του υλικού της. Κατόπιν, έγινε μία γενική ανάλυση του ολοκληρωμένου κυκλώματος και της τυπωμένης πλακέτας και έπειτα μία λεπτομερής ανάλυση των 2 βασικών υλικών του κυκλώματος· του επεξεργαστή και του αισθητήρα θερμοκρασίας και υγρασίας. Στη συνέχεια, αναλύθηκε το κάθε υποκύκλωμα. Αυτά είναι το κύκλωμα φόρτισης, το οποίο είναι υπεύθυνο για τη φόρτιση της μπαταρίας της κάθε αισθητήριας συσκευής, το κύκλωμα σταθεροποίησης, το οποίο είναι υπεύθυνο για τη σταθεροποίηση της τάσης στην επιθυμητή για την τροφοδοσία σταθερής τάσης των εξαρτημάτων και τέλος το κύκλωμα ρελέ που ελέγχει την αντίστοιχη ψυκτική μονάδα που είναι συνδεδεμένη η κάθε συσκευή. Πιο συγκεκριμένα, έγινε αναφορά στα βασικά εξαρτήματα του κάθε κυκλώματος ξεχωριστά, μέσω των φύλλων δεδομένων του κάθε εξαρτήματος καθώς και λεπτομερής ανάλυση του κυκλώματος και του τρόπου λειτουργίας του. Τέλος, η τελευταία ενότητα, αφιερώθηκε στη σχεδίαση και ανάπτυξη του πλαστικού καλουπιού που περικλείει το τυπωμένο κύκλωμα.

Κεφάλαιο 3ο: Software

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα αναλυθούν τα δύο προγράμματα που έχουν γραφτεί για τη λειτουργία της συσκευής. Ο κώδικας έχει γραφτεί σε C++, εκτός από το κομμάτι που ρυθμίζει τη λειτουργία της ιστοσελίδας. Το προγραμματιστικό περιβάλλον στο οποίο έχει γραφτεί ο κώδικας είναι το Arduino IDE (V.2.1.0), αλλά έχουν χρησιμοποιηθεί κι άλλα για αποσφαλμάτωση (Microsoft Visual Studio, CodeLite). Τα δύο αυτά προγράμματα ελέγχουν τη λειτουργία και τον έλεγχο των ψυκτικών μονάδων, τη λειτουργικότητα και την εμφάνιση της ιστοσελίδας, την επικοινωνία που υπάρχει μεταξύ ιστότοπου και διακομιστή καθώς και την επικοινωνία ανάμεσα σε διακομιστή και αισθητήριων συσκευών. Ο πρώτος κώδικας ελέγχει τη λειτουργία του ESP32 που λειτουργεί ως web server και είναι υπεύθυνος για τη φιλοξενία της ιστοσελίδας. Το αρχείο περιλαμβάνει όλο το front-end που έχει γραφτεί σε τρεις γλώσσες προγραμματισμού (JavaScript, HTML, CSS), καθώς και το κομμάτι του προγράμματος, που ευθύνεται για την επικοινωνία ανάμεσα σε ιστοσελίδα–διακομιστή (TCP/IP) και διακομιστή-συσκευών (ESP-NOW). Ο δεύτερος κώδικας, περιλαμβάνει τη λογική για τον έλεγχο του ρελέ, μέσω του παλμού που παράγει ο μικροελεγκτής, καθώς και το κομμάτι του προγράμματος που ευθύνεται για την επικοινωνία με τον ESP32 που χειρίζεται την ιστοσελίδα. Από εδώ και πέρα, ο ESP32 που λειτουργεί ως διακομιστής και φιλοξενεί την ιστοσελίδα, θα αναφέρεται ως “M”, εννοώντας ότι είναι ο κύριος (Main) σύνδεσμος του συστήματος. Οι αισθητήριες συσκευές θα αναφέρονται ως S1, S2, και S3 για τη κάθε μια ξεχωριστά, εννοώντας ότι είναι οι δευτερεύων κόμβοι (subnodes) του συστήματος.

3.2 Ο Κώδικας του Διακομιστή

3.2.1 Γενικά

Ο πρώτος κώδικας που θα αναλυθεί είναι ο κώδικας που έχει γραφτεί για τον web server. Ο συγκεκριμένος κώδικας δεν είναι περασμένος σε κάποια από τις αισθητήριες συσκευές, αλλά σε ένα αναπτυξιακό κύκλωμα ESP32, το οποίο θα είναι συνέχεια συνδεδεμένο στο ρεύμα (αν δεν τροφοδοτείται, τότε ο server δε θα λειτουργεί). Οι λειτουργίες του, είναι να ελέγχει τη λειτουργία της ιστοσελίδας, την επικοινωνία της ιστοσελίδας με τον server και την επικοινωνία του server με κάθε μία από τις αισθητήριες συσκευές.

Η πρώτη επικοινωνία μεταξύ διακομιστή και αισθητήριας συσκευής γίνεται μέσω ESP-NOW. Το ESP-NOW είναι ένα ασύρματο πρωτόκολλο επικοινωνίας που έχει σχεδιαστεί και αναπτυχθεί από την Espressif και επιτρέπει την επικοινωνία ανάμεσα σε πολλαπλές συσκευές χωρίς τη χρήση Wi-Fi, αλλά με τη χρήση ασύρματης σύνδεσης χαμηλής κατανάλωσης στα 2.4 GHz. Πριν ξεκινήσει η επικοινωνία μεταξύ των συνδεδεμένων συσκευών, απαιτείται πρώτα να υπάρξει σύζευξη ανάμεσά τους. Αφού ολοκληρωθεί η σύζευξη, οι πληροφορίες στέλνονται αμφίδρομα, δηλαδή τα δεδομένα ρέουν ταυτόχρονα από το διακομιστή στην κάθε αισθητήρια συσκευή και το αντίστροφο. Στην πρώτη περίπτωση τα δεδομένα, τα οποία αποστέλλονται από τον M στους S1, S2 και S3 είναι:

- Τιμή κατωφλίου (Threshold Value)
- Πλαίσιο Ελέγχου (Checkbox)

Η τιμή κατωφλίου είναι η τιμή που θέτει ο χρήστης για την ελάχιστη τιμή θερμοκρασίας που θέλει να υπάρχει στο εσωτερικό της ψυκτικής θαλάμης. Αν ξεπεραστεί αυτή, ο ανεμιστήρας τους ψυγείου ανάβει. Το checkbox είναι ένα πλαίσιο, το οποίο όταν είναι επιλεγμένο το σύστημα δουλεύει αυτόματα, δηλαδή σύμφωνα με τη τιμή κατωφλίου που έχει οριστεί. Όταν είναι απενεργοποιημένο, ανεξάρτητα από τη τιμή κατωφλίου, η ψυκτική μονάδα θα παραμείνει απενεργοποιημένη. Αυτά αποστέλλονται στις συσκευές για να γίνουν οι αναγκαίοι έλεγχοι για τον έλεγχο του ρελέ. Στη δεύτερη περίπτωση τα δεδομένα, τα οποία αποστέλλονται από τις S1, S2 και S3 στον M είναι οι εξής:

- Θερμοκρασία (Κελσίου)
- Θερμοκρασία (Fahrenheit)
- Υγρασία
- Τάση Μπαταρίας
- Ποσοστό Μπαταρίας
- Κατάσταση Μαγνητικής Επαφής

Οι τρεις τιμές θερμοκρασίας και υγρασίας είναι οι τιμές που “διαβάζει” ο DHT22 και η τιμή της τάσης και ποσοστού μπαταρίας θα εμφανίζονται δίπλα από τις παραπάνω τιμές. Η κατάσταση της μαγνητικής επαφής δηλώνει αν η πόρτα του ψυγείου είναι ανοιχτή ή κλειστή.

Η δεύτερη επικοινωνία μεταξύ ιστοσελίδας-διακομιστή γίνεται μέσω TCP/IP. Το TCP/IP είναι ένα πρωτόκολλο επικοινωνίας που χρησιμοποιείται για να συνδέσει ασύρματα συσκευές στο διαδίκτυο. Έχει σχεδιαστεί για να μεταφέρει πακέτα στο διαδίκτυο και διασφαλίζει την αξιόπιστη αυτή μεταφορά σε όλα τα δίκτυα τα οποία εμπλέκονται. Μέσω αυτής της επικοινωνίας, οι πληροφορίες στέλνονται αμφίδρομα, δηλαδή τα δεδομένα ρέουν ταυτόχρονα από την ιστοσελίδα στο διακομιστή και από το διακομιστή στην ιστοσελίδα. Στην πρώτη περίπτωση τα δεδομένα, τα οποία αποστέλλονται από την ιστοσελίδα στο M είναι οι εξής:

- Τιμή κατωφλίου (Threshold Value)
- Πλαίσιο Ελέγχου (Checkbox)
- Κουμπί κατάθεσης (Submit)

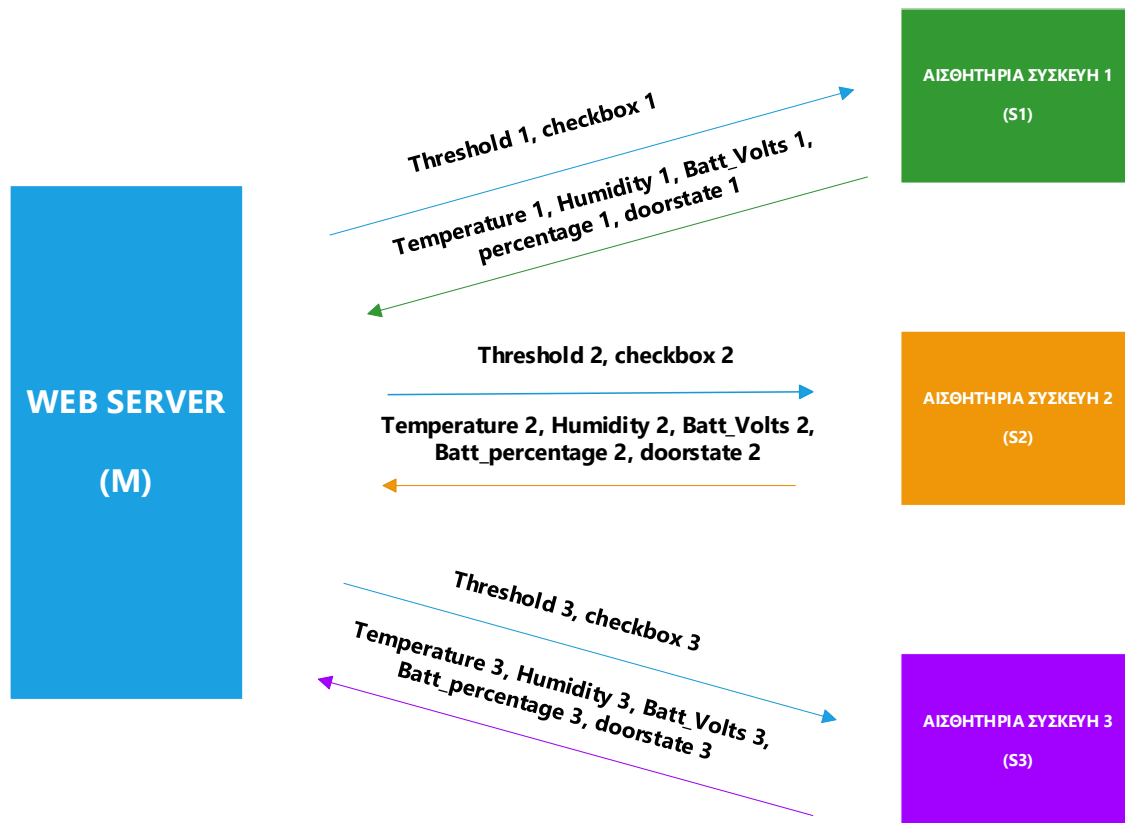
Τα δύο πρώτα δεδομένα είναι αυτά που αναφέρθηκαν παραπάνω στην επικοινωνία server-συσκευών. Το κουμπί κατάθεσης, καταθέτει όλες τις τιμές που έχει θέσει ο χρήστης στο server. Στη δεύτερη περίπτωση τα δεδομένα, τα οποία αποστέλλονται από το M στην ιστοσελίδα είναι οι εξής:

- Θερμοκρασία (Κελσίου)
- Θερμοκρασία (Fahrenheit)
- Υγρασία
- Τάση Μπαταρίας
- Ποσοστό Μπαταρίας
- Κατάσταση Μαγνητικής Επαφής
- Κατάσταση Ανεμιστήρα
- Κατάσταση μπαταρίας

Οι έξι πρώτες τιμές που αναγράφονται, είναι οι τιμές που αποστέλλονται από κάθε αισθητήρια συσκευή μέσω ESP-NOW και αναφέρθηκαν παραπάνω, στη περίπτωση μεταφοράς δεδομένων από κάθε συσκευή στον server. Η κατάσταση του ανεμιστήρα, δηλώνει αν ο ανεμιστήρας του ψυγείου είναι ανοιχτός ή κλειστός και η κατάσταση της μπαταρίας είναι ουσιαστικά μία μεταβλητή που ελέγχει τη τάση της μπαταρίας και παράγει ένα κείμενο. Οι δύο παραπάνω καταστάσεις στην ουσία δεν είναι αριθμητικές τιμές που αποστέλλονται στην ιστοσελίδα όπως για παράδειγμα η θερμοκρασία, αλλά είναι

μεταβλητές που ελέγχουν την κάθε μία από αυτές τις καταστάσεις και αναλόγως παράγουν ένα κείμενο για να διαβάσει ο χρήστης και να γνωρίζει ανά πάσα στιγμή τη κατάσταση των παραπάνω.

ESP-NOW Wireless Communication Protocol



Σχήμα 3.1: Μπλοκ Διάγραμμα της επικοινωνίας Διακομιστή-Συσκευών

3.2.2 Η λειτουργία του Web Server

Σε αυτήν την ενότητα, ο κώδικας που θα εξηγηθεί είναι γραμμένος σε HTML, JavaScript και CSS. Η HTML είναι μία γλώσσα σήμανσης, και είναι η βάση κάθε ιστοσελίδας, ασχέτως με τη πολυπλοκότητα. Χρησιμοποιεί ετικέτες για να προσδιορίσει διαφορετικούς τύπους περιεχομένου, καθώς και τους σκοπούς που εξυπηρετεί ο καθένας στην ιστοσελίδα. Η JavaScript, είναι μία γλώσσα προγραμματισμού πιο σύνθετη από την HTML και επιτρέπει τη σχεδίαση διαδραστικών αντικειμένων όπως ένα κουμπί ή ένα πτυσσόμενο μενού. Στη JavaScript, οφείλεται το μεγαλύτερο μέρος της δυναμικής συμπεριφοράς σε μια ιστοσελίδα. Τέλος, η CSS είναι μια γλώσσα προγραμματισμού που υπαγορεύει πώς πρέπει να εμφανίζονται πραγματικά τα στοιχεία της HTML ενός ιστότοπου στο frontend της σελίδας αυτής. Όλος ο κώδικας για τη λειτουργία της ιστοσελίδας είναι γραμμένος μέσα σε ένα αρχείο index.html.

Αρχικά, ορίζουμε την κλίμακα της ιστοσελίδας με την εντολή στο 100% με την εντολή "initial-scale=1" και ορίζουμε το πλάτος της προβολής στο πλάτος της κάθε οθόνης συσκευής με την εντολή content="width=device-width.

```
<!DOCTYPE HTML><html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <script src="https://kit.fontawesome.com/7ae1530eb9.js"
crossorigin="anonymous"></script>
```

Έπειτα, πρέπει να αρχικοποιήσουμε βασικές πληροφορίες όπως το μέγεθος και το είδος της γραμματοσειράς, το χρώμα και τη στοίχιση των κειμένων που θα χρησιμοποιηθούν. Κάθε μία από τις παρακάτω παραγράφους έχουν διαφορετικά χαρακτηριστικά, τα οποία δηλώνονται μέσα στις παρενθέσεις. Για παράδειγμα η παράγραφος “.units”, δηλώνει τα χαρακτηριστικά του κειμένου για τις μονάδες μέτρησης (°C, °F, V), ενώ η παράγραφος “.dht-labels”, δηλώνει τα χαρακτηριστικά του κειμένου για τις αριθμητικές τιμές του DHT22 και της τάσης της μπαταρίας.

```
html {
  font-family: Arial;
  display: inline-block;
  margin: 0px auto;
  text-align: center;
}
h2 { font-size: 3.0rem; }
p { font-size: 3.0rem; }
p2 {font-size: 1.1rem;}
p3 {font-size: 1.1rem;
  color:#4CAF50;}
p4 {font-size: 1.5rem;
  vertical-align:middle;
  padding-bottom: 15px;
}
p5 {font-size: 1.5rem; }
.units { font-size: 1.2rem; }
.dht-labels{
  font-size: 1.5rem;
  vertical-align:middle;
  padding-bottom: 15px;
}
```

Μετά, ορίζουμε τα χαρακτηριστικά του κουμπιού “Submit”.

```
input[type="submit"]{
  background: #0066A2;
  color: white;
  border-style: outset;
  border-color: #0066A2;
  height: 50px;
  width: 100px;
  font: bold15px arial,sans-serif;
  text-shadow: none;}
```

Τέλος, ορίζουμε τα χαρακτηριστικά του κάθε πλαισίου μέσα στα οποία απεικονίζονται οι αριθμητικές τιμές. Υπάρχουν τρεις αισθητήριες συσκευές και για κάθε συσκευή απεικονίζονται δύο πλαίσια (στο πρώτο οι τιμές του αισθητήρα και στο δεύτερο οι τιμές της μπαταρίας), άρα στο σύνολο υπάρχουν έξι πλαίσια. Παρακάτω παρατίθεται ο τρόπος σύνταξης του ενός για οικονομία χώρου, αλλά και τα υπόλοιπα πέντε στοιχίζονται με παρόμοιο τρόπο αλλάζοντας μόνο το χρώμα του πλαισίου.

```
.element1 {
  display: inline-block;
  background-color: lightgrey;
  width:500px;
  border-width: 5px;
  border-color: #4CA64C;
  border-style: solid;
  text-align: center;
}
```

Αφού έχει τελειώσει η δήλωση των χαρακτηριστικών της ιστοσελίδας, σειρά έχει η συγγραφή της. Παρακάτω, μέσα στη παράγραφο <h2>, ορίζεται ο τίτλος της ιστοσελίδας που απεικονίζεται πάνω και έπειτα το πρώτο πλαίσιο που θα περιέχει τις τιμές θερμοκρασίας και υγρασίας μέσα στην ετικέτα “element1”. Μέσα σε αυτή υπάρχουν τρεις παράγραφοι <p> και μέσα στη κάθε μια καλείται η κάθε εικόνα που χρησιμοποιείται <i class>, το κείμενο , η αριθμητική τιμή και η μονάδα μέτρησης <sup class>. Η πρώτη παράγραφος είναι το κομμάτι απεικόνισης θερμοκρασίας σε βαθμούς Κελσίου, η δεύτερη παράγραφος είναι το κομμάτι απεικόνισης θερμοκρασίας σε βαθμούς Fahrenheit και η τρίτη παράγραφος είναι το κομμάτι απεικόνισης της υγρασίας.

```
<body>
  <h2> FREEZER UNIT CONTROL SERVER </h2>
  <div class="element1">
    <p4>
      <br><strong> INDOOR</strong></p4>
    <p>
      <i class="fas fa-thermometer-half" style="color:#059e8a;"></i>
      <span class="dht-labels">Temperature in Celsius</span>
      <span id="tc1"></span>
      <sup class="units">&deg;C</sup>
    </p>
    <p>
      <i class="fas fa-thermometer-half" style="color:#059e8a;"></i>
      <span class="dht-labels">Temperature in Fahrenheit</span>
      <span id="tf1"></span>
      <sup class="units">&deg;F</sup>
    </p>
    <p>
      <i class="fas fa-tint" style="color:#00add6;"></i>
      <span class="dht-labels">Humidity</span>
      <span id="h1"></span>
    </p>
  </div>
```

```

    <sup class="units">&percent;</sup>
  </p>
</div>

```

Στη συνέχεια, ορίζεται το δεύτερο πλαίσιο που απεικονίζει τις αριθμητικές τιμές τάσης, ποσοστού μπαταρίας και του κειμένου για τη κατάσταση μπαταρίας. Περισσότερες λεπτομέρειες για αυτό το κείμενο θα αναλυθούν σε άλλο κομμάτι. Μέσα στην ετικέτα `<element2>` που ορίζει και το πλαίσιο υπάρχουν τρεις παράγραφοι `<p>` και μέσα στη κάθε μια καλείται η κάθε εικόνα που χρησιμοποιείται `<i class>`, το κείμενο ``, η αριθμητική τιμή `` και η μονάδα μέτρησης `<sup class>`. Η πρώτη παράγραφος είναι το κομμάτι απεικόνισης τάσης μπαταρίας σε Volts, η δεύτερη παράγραφος είναι το κομμάτι απεικόνισης ποσοστού μπαταρίας και η τρίτη παράγραφος είναι το κομμάτι απεικόνισης του κειμένου για τη κατάσταση της μπαταρίας.

```

</div>
<div class="element2">
  <p4>
    <br><strong><i class="fa-solid fa-battery-full"
style="color:#008000;"></i> BATTERY STATUS</strong></p4>

```

```

<p>
  <i class="fa-solid fa-bolt" style="color:#FFA500"></i>
  <span class="dht-labels"> Battery Voltage:</span>
  <span id="v1"></span>
  <span class="units">V </span>
</p>
<p>
  <i class="fa-solid fa-percent" style="color:#800000"></i>
  <span class="dht-labels"> Battery Percentage: </span>
  <span id="p1"></span>
  <sup class="units">&percent;</sup>
</p>
<p>
<p2> <strong><span id="batStatus1"></span></strong></p2>
</p>
</div>

```

Τέλος, παρακάτω υπάρχουν οι εξής λειτουργίες. Μέσα στη πρώτη ελέγχεται η λειτουργία του πλαισίου ελέγχου, η δεύτερη ελέγχει τη λειτουργία επιλογής τιμής καταφλίου, η τρίτη απεικονίζει ένα κείμενο που επιτρέπει το χρήστη να καταλάβει εάν η ψυκτική μονάδα δουλεύει σε αυτόματη λειτουργία, η τέταρτη απεικονίζει το κείμενο κατάστασης ανεμιστήρα, η πέμπτη απεικονίζει το κείμενο κατάστασης της πόρτας της θαλάμης και τέλος, η έκτη ελέγχει τη λειτουργία του κουμπιού “Submit”. Πρέπει να δοθεί ιδιαίτερη προσοχή στο γεγονός ότι αν ο χρήστης δεν πατήσει το κουμπί, δε θα γίνει καμία αλλαγή στις καινούργιες τιμές και ο web server θα συνεχίσει να λειτουργεί με τις παλιές!

```
<input type="checkbox" name="enable_arm_input1" value="true"
id="checkbox1" %ENABLE_ARM_INPUT1%></span>
```

```
<span class="dht-labels"><strong>Freezer 1 </strong> set to: <input
type="number" step="0.1" name="threshold_input1" value="%THRESHOLD1%"
required></span> <br><br>
```

```
<p3 id="message1"></p3> <br><br>
```

```
<p4><strong>Freezer 1 is <spanid="fanStatus1">%FAN_STATUS1%
</span>.</strong></p4> <br><br>
```

```
<p4><strong><span id="doorStatus1"></span></strong></p4> <p></p>
```

```
<p2><input type="submit" value="Submit"></p2>
```

Το τελευταίο κομμάτι που μένει είναι να κατασκευαστούν οι συναρτήσεις μέσω των οποίων όλες οι λειτουργίες που δηλώθηκαν παραπάνω θα είναι λειτουργικές. Αρχικά, πρέπει να κατασκευαστεί μια συνάρτηση, η οποία θα απεικονίζει όλες τις τιμές που έρχονται από τις αισθητήριες συσκευές στο σωστό σημείο. Το παρακάτω απόσπασμα κώδικά κάνει ακριβώς αυτή τη δουλειά. Πιο συγκεκριμένα, δημιουργείται ένα αντικείμενο 'EventSource' που συνδέεται σε ένα σημείο εξυπηρέτησης του διακομιστή και ελέγχει αν ο περιηγητής υποστηρίζει λειτουργία SSE πριν δημιουργήσει αυτό το αντικείμενο. Οι γραμμές 'source.addEventListener' προσθέτουν ακροατές συμβάντων για να χειριστούν τα διάφορα συμβάντα που αποστέλλονται από το διακομιστή μέσω της σύνδεσης SSE. Αυτά τα συμβάντα έχουν τέσσερις καταστάσεις.

- **'open'**: Αυτό το συμβάν εκτελείται, όταν η σύνδεση με τον διακομιστή είναι ενεργή. Όταν συμβεί αυτό, η κονσόλα καταγράφει το μήνυμα "Events Connected".
- **'error'**: Αυτό το συμβάν εκτελείται, όταν συμβεί κάποιο σφάλμα με τη σύνδεση SSE. Εάν η σύνδεση δεν είναι ανοιχτή, η κονσόλα καταγράφει το μήνυμα "Events Disconnected".
- **message**: Αυτό το συμβάν εκτελείται, όταν λαμβάνεται ένα γενικό μήνυμα από το διακομιστή μέσω SSE. Όταν συμβεί αυτό, η κονσόλα καταγράφει το μήνυμα "message" και τα δεδομένα ('data') του συμβάντος.
- **'new_readings'**: Αυτό το συμβάν εκτελείται, όταν λαμβάνονται νέα δεδομένα αισθητήρων από το διακομιστή μέσω SSE. Όταν συμβαίνει αυτό, η κονσόλα καταγράφει το μήνυμα "new_readings" και τα δεδομένα που μεταφέρονται στο "φορτίο" του συμβάντος, το οποίο είναι μια JSON συμβολοσειρά που περιέχει τις αναγνώσεις των αισθητήρων. Στη συνέχεια, ο κώδικας εξάγει τις αναγνώσεις των αισθητήρων από τη JSON συμβολοσειρά και ενημερώνει τις τιμές που εμφανίζονται στην ιστοσελίδα για κάθε αισθητήρα ('tc', 'tf', 'h', 'v' και 'p'). Αυτές οι τιμές πρέπει να είναι ίδιες με αυτές στο προηγούμενο κομμάτι που εξηγήθηκε σχετικά με την απεικόνιση των αριθμητικών τιμών στην ετικέτα . Τέλος, δηλώνεται και με πόσα δεκαδικά ψηφία θα απεικονίζονται οι αριθμητικές τιμές. Για παράδειγμα, η θερμοκρασία απεικονίζεται με ακρίβεια ενός δεκαδικού, ενώ η τάση της μπαταρίας με ακρίβεια δύο δεκαδικών.

```

if (!!window.EventSource) {
  var source = new EventSource('/events');

  source.addEventListener('open', function(e) {
    console.log("Events Connected");
  }, false);

  source.addEventListener('error', function(e) {
    if (e.target.readyState !== EventSource.OPEN) {
      console.log("Events Disconnected");
    }
  }, false);

  source.addEventListener('message', function(e) {
    console.log("message", e.data);
  }, false);

  source.addEventListener('new_readings', function(e) {
    console.log("new_readings", e.data);
    var obj = JSON.parse(e.data);
    document.getElementById("tc"+obj.id).innerHTML =
      obj.temperaturec.toFixed(1);
    document.getElementById("tf"+obj.id).innerHTML =
      obj.temperaturef.toFixed(1);
    document.getElementById("h"+obj.id).innerHTML = obj.humidity.toFixed(1);
    document.getElementById("v"+obj.id).innerHTML = obj.volts.toFixed(2);
    document.getElementById("p"+obj.id).innerHTML =
      obj.percentage.toFixed(2);
  }, false);
}

```

Το παρακάτω κομμάτι ελέγχει τη λειτουργία του κειμένου όταν ο χρήστης τσεκάρει κάποιο από τα τρία πλαίσια ελέγχου. Όταν δηλαδή το checkbox επιλεγθεί αυτόματα εμφανίζεται το κείμενο «The freezer unit is working in automatic operation.»

```

var checkbox1 = document.getElementById("checkbox1");
var message1 = document.getElementById("message1");

setInterval(function() {
  if (checkbox1.checked) {
    message1.innerHTML = "The freezer unit is working in automatic
      operation";
  } else {
    message1.innerHTML = "";
  }
}, 50); // Check every 50ms

```

Τέλος, η παρακάτω συνάρτηση ελέγχει το κείμενο που απεικονίζεται για τη κατάσταση του ανεμιστήρα, της πόρτας και της τάσης μπαταρίας της κάθε ψυκτικής θαλάμης στέλνοντας ένα HTTP GET αίτημα κάθε μισό δευτερόλεπτο. Όταν ο εξυπηρετητής απαντήσει με έναν κωδικό κατάστασης

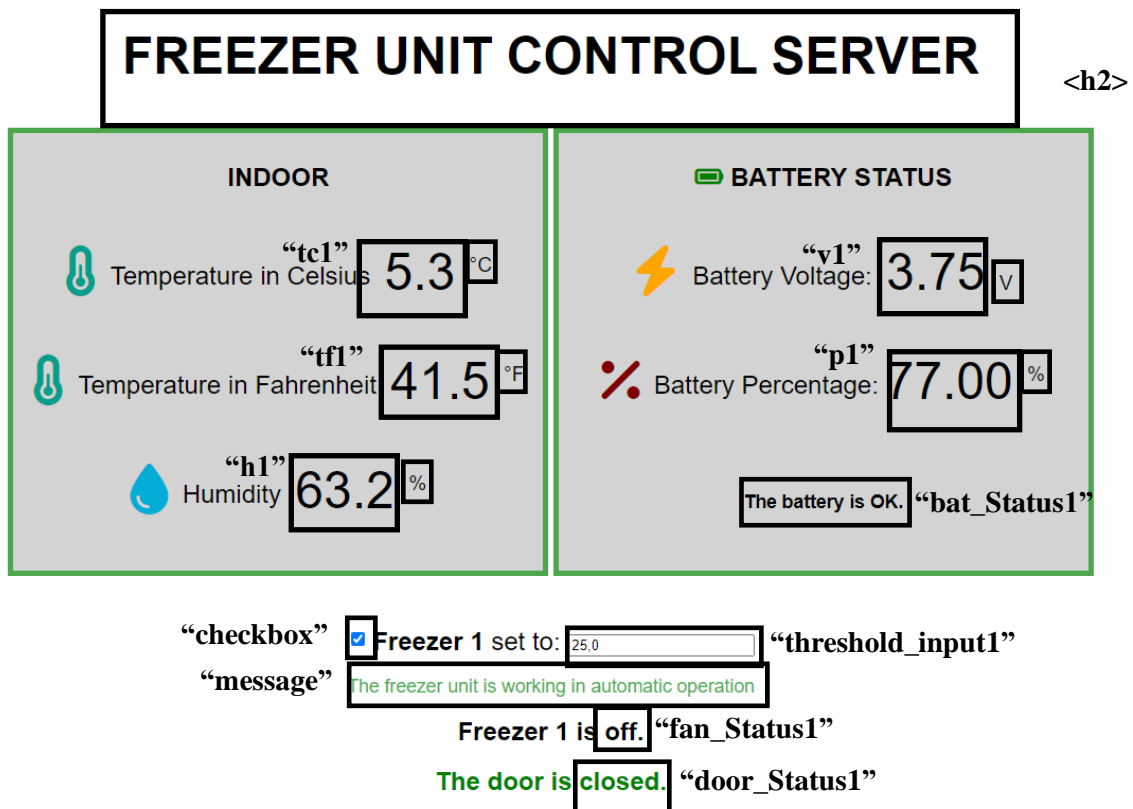
200 (OK) και ένα κείμενο απάντησης, η λειτουργία εξάγει το κείμενο απάντησης και ορίζει το 'innerHTML' ενός στοιχείου HTML με το κάθε αναγνωριστικό ID ('fanStatus1', 'drStatus1', 'btStatus1') στο κείμενο απάντησης. Το στοιχείο 'fanStatus1' είναι για τη κατάσταση του ανεμιστήρα, το στοιχείο 'drStatus1' είναι για τη κατάσταση της πόρτας και το στοιχείο 'btStatus1' είναι για τη κατάσταση της μπαταρίας. Το αναγνωριστικό '1' δηλώνει ότι αναφερόμαστε στη πρώτη ψυκτική μονάδα. Συνολικά υπάρχουν εννιά τέτοιες συναρτήσεις, αλλά παρακάτω παρατίθενται μόνο τρεις από αυτές για οικονομία χώρου (μία για κάθε κατάσταση στο πρώτο ψυγείο), παρόλα αυτά ο τρόπος συγγραφής της κάθε μίας είναι ο ίδιος.

```
setInterval(function ( ) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("fanStatus1").innerHTML = this.responseText;
        }
    };
    xhttp.open("GET", "/status1", true);
    xhttp.send();
}, 500 ) ;
```

```
setInterval(function ( ) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("doorStatus1").innerHTML = this.responseText;
        }
    };
    xhttp.open("GET", "/drstatus1", true);
    xhttp.send();
}, 500 ) ;
```

```
setInterval(function ( ) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("batStatus1").innerHTML = this.responseText;
        }
    };
    xhttp.open("GET", "/btStatus1", true);
    xhttp.send();
}, 500 ) ;
```

Στην παρακάτω εικόνα, γίνεται μία προσπάθεια διαχωρισμού των κομματιών ενός εκ των τριών πλαισίων ελέγχου της πρώτης ψυκτικής μονάδας.



Σχήμα 3.2: Η ιστοσελίδα χωρισμένη στους “placeholders”

3.2.3 Επικοινωνία Ιστοσελίδας-Διακομιστή

Σε αυτήν την ενότητα θα αναλυθεί ο κώδικας που έχει γραφτεί για την αμφίδρομη επικοινωνία ανάμεσα στην ιστοσελίδα και το διακομιστή. Αρχικά, πρέπει να δηλωθούν τα διαπιστευτήρια του δικτύου στο οποίο θα είναι συνδεδεμένος μέσω Wi-Fi ο server (για λόγους ασφαλείας έχουν αντικατασταθεί με αστερίσκους) και έπειτα πρέπει να δηλωθούν οι μεταβλητές που θα χρησιμοποιηθούν. Στον κώδικα παρακάτω παρατίθενται και σχόλια για τη χρησιμότητα της κάθε μεταβλητής.

```
//declare router credentials. For safety reasons mine are hidden.
const char* ssid = "*****";
const char* password = "*****";

//threshold_input and enable_arm_input are threshold value and checkbox
const char* PARAM_INPUT_1 = "threshold_input1";
const char* PARAM_INPUT_2 = "enable_arm_input1";
const char* PARAM_INPUT_3 = "threshold_input2";
const char* PARAM_INPUT_4 = "enable_arm_input2";
const char* PARAM_INPUT_5 = "threshold_input3";
const char* PARAM_INPUT_6 = "enable_arm_input3";
```

```

#define DHTPIN 27    // Digital pin connected to the DHT sensor
#define DHTTYPE    DHT22 //define sensor type
DHT dht(DHTPIN, DHTTYPE);

String threshold1 = "25.0"; //Set initial temperature threshold 1
String enableArmChecked1 = "checked"; //Set initial checkbox 1 status to
checked
String inputMessage1 = "true"; //string to set the value "true" if
checkbox 1 is checked
String fanStatus1; //string to save the fan status 1
String threshold2 = "25.0"; //Set initial temperature threshold 2
String enableArmChecked2 = "checked"; //Set initial checkbox 2 status to
checked
String inputMessage2 = "true"; //string to set the value "true" if
checkbox 2 is checked
String fanStatus2; //string to save the fan status 2
String threshold3 = "25.0"; //Set initial temperature threshold 3
String enableArmChecked3 = "checked"; //Set initial checkbox 3 status to
checked
String inputMessage3 = "true"; //string to set the value "true" if
checkbox 3 is checked
String fanStatus3; //string to save the fan status 3
int doorState; //doorstate either open or closed
float tempc1; //temperature value from sensor device 1
float tempc2; //temperature value from sensor device 2
float tempc3; //temperature value from sensor device 3

int drst1; //door state from sensor device 1
int drst2; //door state from sensor device 2
int drst3; //door state from sensor device 3

```

Έπειτα, πρέπει να τεθεί μία στατική IP για να γνωρίζει ο χρήστης σε ποια διεύθυνση βρίσκεται η ιστοσελίδα του και να δημιουργηθεί ένα αντικείμενο 'AsyncWebServer' στη πόρτα 80 του πρωτοκόλλου HTTP. Η στατική IP που χρησιμοποιείται στο παρακάτω κώδικα είναι η **192.168.1.10**. Το αντικείμενο 'AsyncWebServer' συνδέεται με το τύπο της ιστοσελίδας που έχει κατασκευαστεί. Η ιστοσελίδα λειτουργεί "ασύγχρονα" εννοώντας ότι τα δεδομένα που απεικονίζονται ανανεώνονται αυτόματα και δε χρειάζεται ο χρήστης να κάνει ανανέωση συνέχεια.

```

// Set Static IP address
IPAddress local_IP(192, 168, 1, 10);
// Set Gateway IP address
IPAddress gateway(192, 168, 1, 1);

IPAddress subnet(255, 255, 0, 0);
IPAddress primaryDNS(8, 8, 8, 8); //optional
IPAddress secondaryDNS(8, 8, 4, 4); //optional

AsyncWebServer server(80); //// Create AsyncWebServer object on port 80
AsyncEventSource events("/events");

```

Έπειτα, δημιουργείται μία συνάρτηση επεξεργασίας (processor), η οποία αντικαθιστά τους placeholders που δημιουργήθηκαν στο αρχείο HTML στην προηγούμενη ενότητα με τις συμβολοσειρές που αντιστοιχούν για τη κάθε μεταβλητή. Για παράδειγμα, στο σημείο στο οποίο χρήστης θέτει μία τιμή κατωφλίου για τη ψυκτική μονάδα 1, υπάρχει ένας placeholder που υποδεικνύει τη θέση της συμβολοσειράς που έχει φτιαχτεί για τη τιμή κατωφλίου της μονάδας 1.

```
//function that replaces all the %PLACEHOLDERS% on the HTML
String processor(const String& var){
  if(var == "THRESHOLD1"){ //if placeholder is the specific variable,
replace with var threshold1
    return threshold1;
  }
  else if(var == "ENABLE_ARM_INPUT1"){ //if placeholder is the specific
variable, replace with var enableArmChecked1
    return enableArmChecked1;
  }
  else if(var == "THRESHOLD2"){ //if placeholder is the specific variable,
replace with var threshold2
    return threshold2;
  }
  else if(var=="ENABLE_ARM_INPUT2"){ //if placeholder is the specific
variable, replace with var enableArmChecked2
    return enableArmChecked2;
  }

  else if(var == "THRESHOLD3"){ //if placeholder is the specific variable,
replace with var threshold3
    return threshold3;
  }
  else if(var=="ENABLE_ARM_INPUT3"){ //if placeholder is the specific
variable, replace with var enableArmChecked3
    return enableArmChecked3;
  }
  return String();
}
```

Σε περίπτωση κάποιου προβλήματος φόρτωσης της ιστοσελίδας επειδή ο πελάτης ζητήσει ένα URL που δεν υπάρχει, καλείται η παρακάτω συνάρτηση και στέλνει Error 404.

```
//Handle this in case where the url is not found on the server.
void notFound(AsyncWebServerRequest *request) {
  request->send(404, "text/plain", "Not found");
}
```

Μέσα στη συνάρτηση 'setup()' δίνεται η εντολή για να ξεκινήσει η σύνδεση του ESP32 στο Wi-Fi χρησιμοποιώντας τα διαπιστευτήρια που θέσαμε προηγουμένως. Αφού γίνει αυτό ο ESP32 ορίζεται ως Wi-Fi Station.

```
// Set device as a Wi-Fi Station
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.println("Setting as a Wi-Fi Station..");
}
```

Στη συνέχεια, οι παρακάτω συναρτήσεις δηλώνουν τι γίνεται όταν ο ESP32 λάβει μία αίτηση HTTP. Πιο συγκεκριμένα, η συνάρτηση ορίζει έναν χειριστή HTTP GET για τον εξυπηρετητή, ο οποίος ακούει στο root path ("/"). Όταν λαμβάνει αίτημα GET στο root path, αποστέλλει στον πελάτη το αρχείο index.html ως απόκριση με κωδικό κατάστασης 200 (OK). Το index.html επεξεργάζεται με τη χρήση της συνάρτησης επεξεργασίας (processor) που αναλύθηκε προηγουμένως που περνάει ως παράμετρο στη συνάρτηση send_P.

```
//Handle requests when http req is received for the processor string
server.on("/", HTTP_GET, [])(AsyncWebServerRequest *request){
  request->send_P(200, "text/html", index_html, processor);
});
```

Έπειτα, κατασκευάζονται τρεις συναρτήσεις για τον έλεγχο του κειμένου κατάστασης του ανεμιστήρα της κάθε μονάδας. Πιο συγκεκριμένα, η συνάρτηση ορίζει έναν χειριστή HTTP GET για τον εξυπηρετητή, ο οποίος ακούει στο root path ("/status1", "/status2", "/status3") που δηλώθηκε στο αρχείο index.html. Όταν λαμβάνει αίτημα GET στο root path, πρώτα δηλώνεται μία μεταβλητή 'fanStatus1' για τη πρώτη ψυκτική μονάδα, 'fanStatus2' για τη δεύτερη και 'fanStatus3' για την τρίτη σε έναν υποθετικό τελεστή. Αν πληρούνται οι προϋποθέσεις το κείμενο που παράγεται είναι «The fan is **on**». Διαφορετικά το κείμενο που παράγεται είναι «The fan is **off**». Τέλος, αποστέλλει στον πελάτη τη κατάσταση του ανεμιστήρα ως την αντίστοιχη μεταβλητή.

```
//Handle requests when http req is received for fan status 1
server.on("/status1", HTTP_GET, [])(AsyncWebServerRequest *request){
  // Return the fan status for freezer unit 1
  String fanStatus1 = (tempc1 > myData.thres && myData.checkbox ==
"true" && drst1==LOW) ? "on" : "off";
  // Send the HTML response with the appropriate HTTP status code
  request->send(200, "text/plain", fanStatus1);
});
```

Στη συνέχεια, κατασκευάζονται τρεις συναρτήσεις για τον έλεγχο του κειμένου κατάστασης της πόρτας της κάθε μονάδας. Πιο συγκεκριμένα, η συνάρτηση ορίζει έναν χειριστή HTTP GET για τον εξυπηρετητή, ο οποίος ακούει στο root path ("/doorStatus1", "/doorStatus2", "/doorStatus3") που δηλώθηκε στο αρχείο index.html. Όταν λαμβάνει αίτημα GET στο root path, πρώτα δηλώνεται μία μεταβλητή 'doorStatus1' για τη πρώτη ψυκτική μονάδα, 'doorStatus2' για τη δεύτερη και 'doorStatus3' για την τρίτη σε δύο υποθετικούς τελεστές. Ο πρώτος ρυθμίζει το χρώμα του κειμένου και ο δεύτερος το ίδιο το κείμενο. Αν πληρούνται οι προϋποθέσεις το κείμενο που παράγεται είναι «**The door is open**» σε κόκκινο χρώμα για να είναι πιο εμφανές. Διαφορετικά το κείμενο που παράγεται είναι «The door is closed». Τέλος, αποστέλλει στον πελάτη τη κατάσταση της πόρτας ως την μεταβλητή 'htmlResponse' που αντιπροσωπεύει το κείμενο.

```
//Handle requests when http req is received for door status 1
server.on("/doorstatus1", HTTP_GET, []) (AsyncWebServerRequest *request){
  // Determine the state of the magnetic reed switch
  bool doorOpen = (drst1 == HIGH);
  // Set the text color based on the state of the door
  String textColor = doorOpen ? "red" : "green";
  // Create an HTML response with the appropriate text color
  String htmlResponse = "<html><body><p4 style=\"color: " + textColor +
  "\">The door is " + (doorOpen ? "open." : "closed.") +
  "</p4></body></html>";
  // Send the HTML response with the appropriate HTTP status code
  request->send(200, "text/html", htmlResponse);
};
```

batt_value1' για τη πρώτη ψυκτική μονάδα, 'batt_value2' για τη δεύτερη και 'batt_value3' για την τρίτη σε μία συνθήκη 'Αν', και καθεμία από αυτές εξισώνεται με τη τιμή της τάσης της μπαταρίας της κάθε μονάδες που λαμβάνεται μέσω ESP-NOW. Αν η τάση της μπαταρίας είναι μεταξύ 3.9V – 4.3V το κείμενο που παράγεται είναι «Battery is fully charged!». Αν η τάση της μπαταρίας είναι μεταξύ 3.2V – 3.9V το κείμενο που παράγεται είναι «Battery is OK.». !». Αν η τάση της μπαταρίας είναι μεταξύ 3.2V – 2.9V το κείμενο που παράγεται είναι «Battery needs to be charged.». Αν η τάση της μπαταρίας είναι μεταξύ 2.5V – 2.9V το κείμενο που παράγεται είναι «Battery is critical! Please charge.». Τέλος γίνεται ένας τελευταίος έλεγχος που αν το κείμενο είναι το τελευταίο που αναφέρθηκε, θα απεικονίζεται σε χρώμα κόκκινο, και αποστέλλει στον πελάτη τη κατάσταση της πόρτας ως την μεταβλητή 'htmlResponse που αντιπροσωπεύει το κείμενο.

```
//Handle requests when http req is received for battery voltage voltage 1
server.on("/batstatus1", HTTP_GET, [])(AsyncWebServerRequest *request) {
  float batt_value1 = bat_Status1; //float variable to check the batt
  voltage of sensor device 1
  String text1; //string for th displayed text on the server
  if (batt_value1 >= 4.30 && batt_value1 <= 3.90) {
    text1 = "Battery is fully charged!";
  } else if (batt_value1 >= 3.20 && batt_value1 < 3.90) {
    text1 = "Battery is OK.";
  } else if (batt_value1 >= 2.90 && batt_value1 < 3.20) {
    text1 = "Battery needs to be charged";
  } else if (batt_value1 > 2.50 && batt_value1 < 2.90) {
    text1 = "Battery is critical! Please charge.";
  } else {
    text1 = "LOADING...";
  }
  // Create an HTML response with the appropriate text
  String htmlResponse = "<html><body><p2>" + text1 +
  "</p2></body></html>";
  // Send the HTML response with the appropriate HTTP status code
  request->send(200, "text/html", htmlResponse);
};
```

Τέλος, κάθε φορά που πατιέται το κουμπί “Submit” αποστέλλεται στο τσιπ μία φόρμα που περιλαμβάνει τις ανανεωμένες πληροφορίες. Γίνεται δηλαδή έλεγχος αυτής της αίτησης για τις παραμέτρους, και κάθε παράμετρος αποθηκεύεται ως η αντίστοιχη μεταβλητή. Υπάρχουν τρεις τέτοιες συναρτήσεις για κάθε μία από τις συσκευές.

```
server.on("/get", HTTP_GET, []) (AsyncWebServerRequest *request) {
  // if request has input parameter 1
  if (request->hasParam(PARAM_INPUT_1)) {
    // save this parameter into variable threshold1
    threshold1 = request->getParam(PARAM_INPUT_1)->value();
    // if request has input parameter 1
    if (request->hasParam(PARAM_INPUT_2)) {
      // save this parameter into variable inputMessage1
      inputMessage1 = request->getParam(PARAM_INPUT_2)->value();
      //change checkbox 1 status to checked
      enableArmChecked1 = "checked";
    }
    //else checkbox is not checked set string as false
    else {
      inputMessage1 = "false";
      enableArmChecked1= "";
    }
  }
  //after clicking the "submit" button display a page saying the req was
  sent succesfully
  request->send(200, "text/html", "HTTP GET request sent to your
  ESP.<br><a href=\"/\">Return to Home Page</a>");
});
server.begin(); //start server
```

Παρακάτω, στο σχήμα 3.3, απεικονίζεται ολοκληρωμένη ιστοσελίδα και καθεμία από τις ψυκτικές μονάδες βρίσκεται σε διαφορετική κατάσταση. Πιο συγκεκριμένα, στην πρώτη αισθητήρια συσκευή το πλαίσιο ελέγχου είναι απενεργοποιημένο, επομένως ανεξαρτήτως της θερμοκρασίας η μονάδα παραμένει απενεργοποιημένη. Επίσης, η πόρτα είναι κλειστή και η μπαταρία είναι πλήρως φορτισμένη. Στη δεύτερη αισθητήρια συσκευή η θερμοκρασία είναι άνω της θερμοκρασίας κατωφλίου και η πόρτα είναι κλειστή, επομένως ο ανεμιστήρας είναι ενεργοποιημένος. Η τάση της μπαταρίας που βρίσκεται σε φυσιολογικά επίπεδα. Στη τρίτη ψυκτική μονάδα, η θερμοκρασία είναι μεγαλύτερη από τη θερμοκρασία κατωφλίου, παρόλα αυτά επειδή η πόρτα είναι ανοιχτή ο ανεμιστήρας είναι απενεργοποιημένος. Η τάση της μπαταρίας είναι σε χαμηλά επίπεδα και άρα το μήνυμα που τυπώνεται, είναι ότι η μπαταρία πρέπει να φορτιστεί.

FREEZER UNIT CONTROL SERVER

INDOOR	BATTERY STATUS
Temperature in Celsius 23.6 °C	Battery Voltage: 3.98 V
Temperature in Fahrenheit 74.5 °F	Battery Percentage: 91.00 %
Humidity 54.1 %	Battery is fully charged!

Freezer 1 set to: 25.0

Freezer 1 is off.

The door is closed.

INDOOR	BATTERY STATUS
Temperature in Celsius 25.5 °C	Battery Voltage: 3.76 V
Temperature in Fahrenheit 77.9 °F	Battery Percentage: 73.00 %
Humidity 46.4 %	Battery is OK.

Freezer 2 set to: 25

The freezer unit is working in automatic operation

Freezer 2 is on.

The door is closed.

INDOOR	BATTERY STATUS
Temperature in Celsius 26.7 °C	Battery Voltage: 3.20 V
Temperature in Fahrenheit 80.1 °F	Battery Percentage: 11.00 %
Humidity 49.5 %	Battery needs to be charged

Freezer 3 set to: 25.0

The freezer unit is working in automatic operation

Freezer 3 is off.

The door is open.

Submit

Σχήμα 3.3: Ο Web Server

3.2.4 Επικοινωνία Διακομιστή-Αισθητήριων Συσκευών

Σε αυτή την ενότητα θα αναλυθεί ο κώδικας της επικοινωνίας μεταξύ διακομιστή και αισθητήριων συσκευών. Αρχικά, ορίζονται οι βιβλιοθήκες που απαιτούνται για να εδραιωθεί το πρωτόκολλο επικοινωνίας ESO-NOW.

```
#include <esp_now.h>
#include <Arduino_JSON.h>
#include <WiFi.h>
```

Μετά, ορίζονται δύο μεταβλητές για να τεθεί το χρονικό περιθώριο δεν θα αποστέλλονται τα δεδομένα από τον M στις υπόλοιπες συσκευές. Αυτό το χρονικό περιθώριο είναι τα 10 δευτερόλεπτα. Δηλαδή τα δεδομένα που στέλνει ο server σε κάθε μία από τις S1, S2 και S3 θα αποστέλλονται κάθε 10 δευτερόλεπτα. Έπειτα ορίζονται οι τρεις MAC διευθύνσεις των τριών ESP32 σε καθεμία από τις συσκευές.

```
unsigned long previousMillis = 0; // Stores last time temperature was
published
const long interval = 10000; // Interval at which to publish
sensor readings

//MAC Address of the receivers
uint8_t broadcastAddress1[] = {0xEC, 0x94, 0xCB, 0x6F, 0x79, 0x94};
uint8_t broadcastAddress2[] = {0xC8, 0xF0, 0x9E, 0xF2, 0xC6, 0x40};
uint8_t broadcastAddress3[] = {0x1C, 0x9D, 0xC2, 0x54, 0x6E, 0xD8};
```

Στη συνέχεια, θα οριστούν 2 δομές 'struct_message1' και 'struct_message2', για να αποστέλλονται και να λαμβάνονται τα δεδομένα. Στην πρώτη δομή ενσωματώνονται οι μεταβλητές που λαμβάνονται από τις αισθητήριες συσκευές και είναι ο αναγνωριστικός αριθμός ID της κάθε συσκευής, οι τιμές θερμοκρασίας, η τιμή της υγρασίας και η τάση και ποσοστό της μπαταρίας. Στη δεύτερη δομή ενσωματώνονται οι μεταβλητές που στέλνονται σε καθεμία από τις αισθητήριες συσκευές και είναι η τιμή θερμοκρασίας κατωφλίου και η κατάσταση του checkbox.

```
// Structure to receive data
typedef struct struct_message {
    int id; //id of each individual sensor device
    float tempc; //incoming temperature (in Celcius) readings
    float tempf; //incoming temperature (in Fahrenheit) readings
    float hum; //incoming humidity readings
    float volts; //incoming battery voltage readings
    float perc; //incoming battery percentage readings
    int drst; //incoming battery status readings
}
struct_message;
```

```
//Structure to send data
typedef struct struct_message2 {
    float thres; //outgoing threshold value for sensor device 1
    float thres2; //outgoing threshold value for sensor device 2
    float thres3; //outgoing threshold value for sensor device 3
    String checkbox; //outgoing checkbox state for sensor device 1
    String checkbox2; //outgoing checkbox state for sensor device 2
    String checkbox3; //outgoing checkbox state for sensor device 3
}
struct_message2;

struct_message incomingReadings; //link structure struct_mwssage1 to
incomingReadings object
struct_message2 myData; ////link structure struct_message2 to myData
object
```

Έπειτα, θα κατασκευάσουμε μια συνάρτηση 'OnDataRecv()', η οποία θα καλείται κάθε φορά που θα λαμβάνονται τα πακέτα. Αρχικά, μέσα σε αυτήν αντιγράφονται αυτά τα πακέτα που βρίσκονται στη μεταβλητή incomingData, μέσα στην δομή incomingReadings. Στη συνέχεια, δημιουργείται μια συμβολοσειρά JSON μαζί με τα ληφθέντα δεδομένα. Τέλος, αφού μαζευτούν όλα τα πακέτα μέσα στη μεταβλητή, αποστέλλονται οι πληροφορίες στην ιστοσελίδα, ως συμβάν 'new_readings' και τυπώνονται σειριακά.

```
// callback function that will be executed when data is received
void OnDataRecv(const uint8_t * mac_addr, const uint8_t *incomingData,
int len) {
    // Copies the sender mac address to a string
    char macStr[18];
    Serial.print("Packet received from: ");
    snprintf(macStr, sizeof(macStr), "%02x:%02x:%02x:%02x:%02x:%02x",
             mac_addr[0], mac_addr[1], mac_addr[2], mac_addr[3],
mac_addr[4], mac_addr[5]);
    Serial.println(macStr);
    memcpy(&incomingReadings, incomingData, sizeof(incomingReadings));

    //access member variable to incomingReadings object
    board["id"] = incomingReadings.id;
    board["temperaturec"] = incomingReadings.tempc;
    board["temperaturef"] = incomingReadings.tempf;
    board["humidity"] = incomingReadings.hum;
    board["volts"] = incomingReadings.volts;
    board["percentage"] = incomingReadings.perc;
    board["doorstate"] = incomingReadings.drst;
    String jsonString = JSON.stringify(board);
    events.send(jsonString.c_str(), "new_readings", millis());
```

Η δήλωση 'αν' παρακάτω, ελέγχει την τιμή του πεδίου "ID" στη μεταβλητή incomingReadings. Ανάλογα με την τιμή του "ID", σε διαφορετικές μεταβλητές (tempc, drst, volts) εκχωρούνται διαφορετικές τιμές για διαφορετικές συσκευές αισθητήρα. Οι μεταβλητές στις οποίες εκχωρούνται οι ληφθέντες τιμές δηλώνονται στην αρχή του κώδικα και αυτό φαίνεται στην προηγούμενη ενότητα.

```

//set temperature, door state and battery voltage readings to object
depending the ID of each sensor device
//variables for the first sensor device
if (incomingReadings.id == 1) {
  tempc1 = incomingReadings.tempc;
  drst1 = incomingReadings.drst;
  volts1 = incomingReadings.volts;
//variables for the second sensor device
} else if (incomingReadings.id == 2) {
  tempc2 = incomingReadings.tempc;
  drst2 = incomingReadings.drst;
  volts2 = incomingReadings.volts;
//variables for the third sensor device
} else if (incomingReadings.id == 3) {
  tempc3 = incomingReadings.tempc;
  drst3 = incomingReadings.drst;
  volts3 = incomingReadings.volts;
}

```

Μετά από τη συνάρτηση που ελέγχει την λήψη πακέτων από τις αισθητήριες συσκευές σειρά έχει η σύνταξη μιας συνάρτησης, η οποία θα ελέγχει την αποστολή των πακέτων στις συσκευές. Η παρακάτω συνάρτηση 'OnDataSent()' καλείται κάθε φορά που θα αποστέλλονται τα δεδομένα. Σε περίπτωση που τα πακέτα αποσταλούν με επιτυχία τυπώνεται σειριακά το μήνυμα Delivery Success. Διαφορετικά τυπώνεται σειριακά το μήνυμα Delivery Fail.

```

// callback function when data are sent
void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status)
{
  char macStr[18];
  Serial.print("Packet to: ");
  // Copies the sender mac address to a string
  snprintf(macStr, sizeof(macStr), "%02x:%02x:%02x:%02x:%02x:%02x",
           mac_addr[0], mac_addr[1], mac_addr[2], mac_addr[3],
mac_addr[4], mac_addr[5]);
  Serial.print(macStr);
  Serial.print(" send status:\t");
  Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Delivery Success" :
"Delivery Fail");
}

```

Οι παρακάτω γραμμές κώδικα ελέγχουν τη λειτουργία το πρωτόκολλου επικοινωνίας. Αρχικά μέσα στην συνάρτηση 'setup()' ενεργοποιείται το ESP-NOW. Αφού ενεργοποιηθεί, δηλώνεται η συνάρτηση που θα καλεί όταν τα δεδομένα ληφθούν και αποσταλούν. Αυτή η συνάρτηση είναι αντίστοιχα η 'OnDataRecv' και 'OnDataSent'.

```

// Initialize ESP-NOW
if (esp_now_init() != ESP_OK) {
    Serial.println("Error initializing ESP-NOW");
    return;
}

//callback function that will be called when a message is received
esp_now_register_recv_cb(OnDataRecv);
//callback function that will be called when a message is sent
esp_now_register_send_cb(OnDataSent);

```

Στη συνέχεια, θα πρέπει να “ζευγαρώσει” η κάθε αισθητήρια συσκευή με το διακομιστή. Παρακάτω, δημιουργούνται τρεις συναρτήσεις, οι οποίες δημιουργούν τρεις συνδέσεις για την κάθε συσκευή χρησιμοποιώντας τη διεύθυνση MAC της καθεμίας.

```

//callback function that will be called when a message is received
esp_now_register_recv_cb(OnDataRecv);
//callback function that will be called when a message is sent
esp_now_register_send_cb(OnDataSent);

// register peer
esp_now_peer_info_t peerInfo;
peerInfo.channel = 0;
peerInfo.encrypt = false;

//Add peer to send the data to sensor device 1
memcpy(peerInfo.peer_addr, broadcastAddress1, 6);
if (esp_now_add_peer(&peerInfo) != ESP_OK){
    Serial.println("Failed to add peer");
    return;
}

//Add peer to send the data to sensor device 2
memcpy(peerInfo.peer_addr, broadcastAddress2, 6);
if (esp_now_add_peer(&peerInfo) != ESP_OK){
    Serial.println("Failed to add peer");
    return;
}

//Add peer to send the data to sensor device 3
memcpy(peerInfo.peer_addr, broadcastAddress3, 6);
if (esp_now_add_peer(&peerInfo) != ESP_OK){
    Serial.println("Failed to add peer");
    return;
}

```

Αυτή η συνάρτηση δημιουργεί ένα ‘event listener’ για το συμβάν σύνδεσης σε έναν ‘AsyncEventSourceClient’. Όταν ένας πελάτης συνδεθεί, αυτό το event listener εκτελείται και επιστρέφει ένα μήνυμα "hello!" μαζί με τον αριθμό των δευτερολέπτων από την αρχή του χρόνου και ένα χρονικό διάστημα επανασύνδεσης του πελάτη σε περίπτωση απώλειας σύνδεσης.

```

//Set up the event source on the server
events.onConnect([](AsyncEventSourceClient *client){
    if(client->lastId()){
        Serial.printf("Client reconnected! Last message ID that it got
is: %u\n", client->lastId());
    } // send event with message "hello!", id current millis and set
reconnect delay to 1 second
    client->send("hello!", NULL, millis(), 10000);
});

```

Τέλος, μέσα στη συνάρτηση 'loop()' εκτελούνται οι εξής εντολές. Πρώτον, ο Μ στέλνει ping κάθε 5 δευτερόλεπτα για να ελέγξει αν ο διακομιστής εξακολουθεί να είναι ενεργοποιημένος. Έπειτα μέσα σε μια συνάρτηση 'αν' αποστέλλεται ένα πακέτο με τα δεδομένα θερμοκρασίας κατωφλίου και checkbox κάθε 10 δευτερόλεπτα και τυπώνεται σειριακά το αποτέλεσμα της αποστολής.

```

void loop() {
    //create an event every 10 secs
    static unsigned long lastEventTime = millis();
    static const unsigned long EVENT_INTERVAL_MS = 10000;
    if ((millis() - lastEventTime) > EVENT_INTERVAL_MS) {
        events.send("ping",NULL,millis());
        lastEventTime = millis();
    }
    unsigned long currentMillis = millis();
    //every 10 seconds send the outgoing data of threshold and checkbox
status to each sensor device
    if (currentMillis - previousMillis >= interval) {
        // Save the last time a new reading was published
        previousMillis = currentMillis;
        myData.thres = threshold1.toFloat();
        myData.checkbox = inputMessage1;
        myData.thres2 = threshold2.toFloat();
        myData.checkbox2 = inputMessage2;
        myData.thres3 = threshold3.toFloat();
        myData.checkbox3 = inputMessage3;
        //send the message structure via ESP-NOW to sensor device 1
        esp_err_t result1 = esp_now_send(broadcastAddress1, (uint8_t *)
&myData, sizeof(myData));
        if (result1 == ESP_OK) {
            Serial.println("Sent with success");
        }
        else {
            Serial.println("Error sending the data");}
    }
}

```

```

//send the message structure via ESP-NOW to sensor device 2
esp_err_t result2 = esp_now_send(broadcastAddress2, (uint8_t *)
&myData, sizeof(myData));
if (result2 == ESP_OK) {
    Serial.println("Sent with success");
}
else {
    Serial.println("Error sending the data");}

//send the message structure via ESP-NOW to sensor device 3
esp_err_t result3 = esp_now_send(broadcastAddress3, (uint8_t *)
&myData, sizeof(myData));
if (result3 == ESP_OK) {
    Serial.println("Sent with success");
}
else {
    Serial.println("Error sending the data");}
}

```

3.3 Ο κώδικας των Αισθητήριων Συσκευών

3.3.1 Γενικά

Ο δεύτερος κώδικας που θα αναλυθεί, είναι ο κώδικας που έχει γραφτεί για τις τρεις αισθητήριες συσκευές. Οι λειτουργίες του είναι να ελέγχει την επικοινωνία του server με την εκάστοτε συσκευή, να λαμβάνει τις τιμές θερμοκρασίας και υγρασίας από τον DHT22 και τάσης από τη μπαταρία, να διαβάζει τη κατάσταση της μαγνητικής επαφής και να ελέγχει το ρελέ που ανοίγει και κλείνει τον ανεμιστήρα της κάθε ψυκτικής μονάδας. Τέλος, ενεργοποιείται η λειτουργία ύπνου, στην οποία ο ESP32 μπαίνει σε μία κατάσταση αδράνειας για ένα χρονικό διάστημα, έτσι ώστε να εξοικονομήσει ενέργεια.

Η επικοινωνία μεταξύ διακομιστή-συσκευών γίνεται μέσω του πρωτοκόλλου επικοινωνίας ESP-NOW και η διαδικασία έχει αναλυθεί στην ενότητα 3.2.1. Ο μικροελεγκτής “διαβάζει” επίσης τον αισθητήρα και λαμβάνει τις αριθμητικές τιμές θερμοκρασίας σε βαθμούς Κελσίου και Fahrenheit, καθώς και υγρασίας σε ποσοστό επί τοις εκατό. Η μαγνητική επαφή είναι στην ουσία δύο μαγνήτες οι οποίες ανοίγουν και κλείνουν ένα διακόπτη εσωτερικά. Όταν οι μαγνήτες είναι ενωμένοι, ο διακόπτης είναι κλειστός και όταν έχουν απόσταση μεταξύ τους, ο διακόπτης είναι ανοιχτός. Η επαφή του τσιπ διαβάζει αυτή τη κατάσταση και μπορεί να καταλάβει με αυτό τον τρόπο αν η πόρτα των μονάδων είναι ανοιχτή ή κλειστή. Ο ελεγκτής, για να διαβάσει τη τιμή της τάσης, έχει μία επαφή του συνδεδεμένη στο διαιρέτη τάσης ανάμεσα στη μπαταρία και αυτή η τιμή πολλαπλασιάζεται επί δύο για να προκύψει η πραγματική τιμή της τάσης. Το ποσοστό προκύπτει από έναν πίνακα αντιστοίχισης τάσης-ποσοστού μέσα στη βιβλιοθήκη. Το ρελέ ελέγχεται μέσα στη ‘void loop()’. Για όσο η πόρτα είναι ανοιχτή, ανεξαρτήτως της θερμοκρασίας κατωφλίου ή της θερμοκρασίας στο εσωτερικό της θαλάμης, ο ανεμιστήρας παραμένει κλειστός για εξοικονόμηση ενέργειας, γιατί προφανώς το να δουλεύει ενώ η πόρτα είναι ανοιχτή και η ψύξη χάνεται είναι ανούσιο. Αν η πόρτα είναι κλειστή, γίνεται έλεγχος της θερμοκρασίας στο εσωτερικό και του checkbox. Αυτές τις τιμές τις έχει λάβει η συσκευή μέσω ESP-NOW από το διακομιστή. Πιο συγκεκριμένα, αν η τιμή θερμοκρασίας που έχει λάβει ο αισθητήρας

είναι μεγαλύτερη από τη τιμή κατωφλίου, το πλαίσιο ελέγχου είναι ενεργοποιημένο και η έξοδος δεν έχει ενεργοποιηθεί ακόμα, τότε ο ανεμιστήρας ενεργοποιείται μέσω του ρελέ. Αν τουλάχιστον μία από τις παραπάνω συνθήκες δεν ικανοποιείται, ο ανεμιστήρας παραμένει απενεργοποιημένος. Τέλος, ο ESP32 μπαίνει σε λειτουργία ελαφρύ ύπνου ανά 1 λεπτό και ξυπνάει σε δύο περιπτώσεις. Πρώτων, όταν περάσει 1 λεπτό έχει τεθεί ένας χρονιστής που ενεργοποιεί το τσιπ. Δεύτερον, αν ανοίξει η πόρτα της ψυκτικής μονάδας. Το τελευταίο συμβαίνει για να κλείσει ουσιαστικά ο ελεγκτής τον ανεμιστήρα, αφού η πόρτα ανοίγει. Αν δηλαδή η πόρτα άνοιγε και εκείνη την ώρα το τσιπ βρισκόταν σε λειτουργία ύπνου, δε θα μπορούσε να κλείσει την έξοδο και ο ανεμιστήρας θα παρέμενε ενεργοποιημένος.

3.3.2 Η επικοινωνία Αισθητήριων Συσκευών

Σε αυτή την ενότητα θα αναλυθεί ο κώδικας της επικοινωνίας μεταξύ αισθητήριων συσκευών και διακομιστή. Αρχικά, ορίζονται οι βιβλιοθήκες που απαιτούνται για να εδραιωθεί το πρωτόκολλο επικοινωνίας ESO-NOW.

```
#include <esp_now.h>
#include <WiFi.h>
#include <Arduino_JSON.h>
```

Μετά, ορίζονται δύο μεταβλητές για να τεθεί το χρονικό περιθώριο δεν θα αποστέλλονται τα δεδομένα από τη κάθε συσκευή στο διακομιστή. Αυτό το χρονικό περιθώριο είναι τα 10 δευτερόλεπτα. Δηλαδή τα δεδομένα που στέλνει ο S1, S2, S3 στο M, αποστέλλονται κάθε 10 δευτερόλεπτα. Έπειτα ορίζεται η MAC διεύθυνση του ESP32 του server. Πολύ σημαντικό είναι να οριστεί ο αναγνωριστικός αριθμός (ID) της κάθε αισθητήριας συσκευής. Χωρίς αυτόν ο κώδικας του M δεν θα μπορεί να ξεχωρίσει ποιες μετρήσεις έρχονται από ποια συσκευή.

```
unsigned long previousMillis = 0; // Stores last time temperature was
published
const long interval = 10000; // Interval at which to publish
sensor readings

//MAC Address of the receiver
uint8_t broadcastAddress[] = {0xE8, 0x31, 0xCD, 0x02, 0xC9, 0x0C};

// Set your Board ID (ESP32 Sensor device #1 = BOARD_ID 1, ESP32 Sensor
device #2 = BOARD_ID 2, ESP32 Sensor device #3 = BOARD_ID 3)
#define BOARD_ID 1
```

Στη συνέχεια, θα οριστούν 2 δομές 'struct_message1' και 'struct_message2', για να αποστέλλονται και να λαμβάνονται τα δεδομένα. Στην πρώτη δομή ενσωματώνονται οι μεταβλητές που αποστέλλονται από τις αισθητήριες συσκευές και είναι ο αναγνωριστικός αριθμός ID της κάθε συσκευής, οι τιμές θερμοκρασίας, η τιμή της υγρασίας και η τάση και ποσοστό της μπαταρίας. Στη δεύτερη δομή ενσωματώνονται οι μεταβλητές που λαμβάνονται σε καθεμία από τις αισθητήριες συσκευές και είναι η τιμή θερμοκρασίας κατωφλίου και η κατάσταση του checkbox.

```
typedef struct struct_message1 {
    int id;
    float tempc;
    float tempf;
    float hum;
    float volts;
    float perc;
    int drst;
    int readingId;
}
struct1_message;
```

```
typedef struct struct_message2 {
    float thres;
    float thres2;
    float thres3;
    String checkbox;
    String checkbox2;
    String checkbox3;
}
struct_message2;
```

```
//Create a struct_message called myData
```

```
struct_message1 myData;
struct_message2 incomingReadings;
```

Έπειτα, θα κατασκευάσουμε μια συνάρτηση ‘OnDataRecv()’, η οποία θα καλείται κάθε φορά που θα λαμβάνονται τα πακέτα. Αρχικά, μέσα σε αυτήν αντιγράφονται αυτά τα πακέτα που βρίσκονται στη μεταβλητή ‘incomingData’, μέσα στην δομή incomingReadings. Στη συνέχεια, δημιουργείται μια συμβολοσειρά JSON μαζί με τα ληφθέντα δεδομένα.

```
void OnDataRecv(const uint8_t * mac_addr, const uint8_t *incomingData,
int len) {
    // Copies the sender mac address to a string
    char macStr[18];
    Serial.print("Packet received from: ");
    snprintf(macStr, sizeof(macStr), "%02x:%02x:%02x:%02x:%02x:%02x",
        mac_addr[0], mac_addr[1], mac_addr[2], mac_addr[3],
mac_addr[4], mac_addr[5]);
    Serial.println(macStr);
    memcpy(&incomingReadings, incomingData, sizeof(incomingReadings));

    board["threshold"] = incomingReadings.thres;
    board["checkbox"] = incomingReadings.checkbox;
    board["threshold2"] = incomingReadings.thres2;
    board["checkbox2"] = incomingReadings.checkbox2;
    board["threshold3"] = incomingReadings.thres3;
    board["checkbox3"] = incomingReadings.checkbox3;
    String jsonString = JSON.stringify(board);}
```

Για να εδραιωθεί αποτελεσματικά η επικοινωνία μέσω ESP-NOW πρέπει να γίνει σύνδεση στο ίδιο Wi-Fi κανάλι του διακομιστή. Η παρακάτω συνάρτηση σκανάρει το δίκτυο στο οποίο έχει συνδεθεί και εκχωρεί αυτόματα σε κάθε μία από τις συσκευές το ίδιο κανάλι με το server.

```
constexpr char WIFI_SSID[] = "*****";

int32_t getWiFiChannel(const char *ssid) {
    if (int32_t n = WiFi.scanNetworks()) {
        for (uint8_t i=0; i<n; i++) {
            if (!strcmp(ssid, WiFi.SSID(i).c_str())) {
                return WiFi.channel(i);}}
    return 0;}

```

Η παρακάτω συνάρτηση onDataSent καλείται κάθε φορά που θα αποστέλλονται τα δεδομένα. Σε περίπτωση που τα πακέτα αποσταλούν με επιτυχία τυπώνεται σειριακά το μήνυμα Delivery Success. Διαφορετικά τυπώνεται σειριακά το μήνυμα Delivery Fail.

```
// callback when data is sent
void onDataSent(const uint8_t *mac_addr, esp_now_send_status_t status)
{
    Serial.print("\r\nLast Packet Send Status:\t");
    Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Delivery Success" :
"Delivery Fail");
}

```

Οι παρακάτω γραμμές κώδικα βρίσκονται μέσα στη συνάρτηση ‘setup()’. Αρχικά, θέτει την διεπαφή Wi-Fi σε αδιάκριτη κατάσταση λειτουργίας, για να μπορεί να διαβάσει κάθε συσκευή κάθε πακέτου του δικτύου που φτάνει. Έπειτα, καλείται η συνάρτηση ‘getWiFiChannel()’ που αναφέραμε παραπάνω και μετά απενεργοποιεί την αδιάκριτη κατάσταση λειτουργίας που ενεργοποιήθηκε προηγουμένως. Τέλος, καλείται η συνάρτηση ‘printDiag()’ της βιβλιοθήκης Wi-Fi για να τρέξει έναν διαγνωστικό έλεγχο του καναλιού.

```
int32_t channel = getWiFiChannel(WIFI_SSID);

esp_wifi_set_promiscuous(true); //Set the ESP32's WiFi interface to
promiscuous mode
esp_wifi_set_channel(channel, WIFI_SECOND_CHAN_NONE); //change the
channel on which the ESP32 is listening for WiFi traffic.
esp_wifi_set_promiscuous(false); //Disable promiscuous mode
WiFi.printDiag(Serial); // verify channel change

```

Στη συνέχεια, θα πρέπει να “ζευγαρώσει” η κάθε αισθητήρια συσκευή με το διακομιστή. Αρχικά ενεργοποιείται η λειτουργία ESP-NOW και στη συνέχεια καλούνται οι συναρτήσεις που δηλώθηκαν προηγουμένως για τα δεδομένα τα οποία αποστέλλονται και λαμβάνονται. Τέλος, δημιουργείται μια συνάρτηση μέσω της οποίας γίνεται σύνδεση κάθε αισθητήριας συσκευής με το διακομιστή χρησιμοποιώντας τη MAC διεύθυνση του.

```

//Init ESP-NOW
if (esp_now_init() != ESP_OK) {
    Serial.println("Error initializing ESP-NOW");
    return;
}

// get the status of Trasnmitted packet
esp_now_register_send_cb(OnDataSent);
esp_now_register_rcv_cb(OnDataRecv);

//Register peer
esp_now_peer_info_t peerInfo;
memcpy(peerInfo.peer_addr, broadcastAddress, 6);
peerInfo.encrypt = false;
//Add peer
if (esp_now_add_peer(&peerInfo) != ESP_OK){
    Serial.println("Failed to add peer");
    return;
}

```

Τέλος, μέσα σε μια συνάρτηση 'loop()' αποστέλλεται ένα πακέτο με τα δεδομένα θερμοκρασίας (και στις δύο μονάδες μέτρησης), υγρασίας, τάσης και ποσοστού μπαταρίας, κάθε 10 δευτερόλεπτα και τυπώνεται σειριακά το αποτέλεσμα της αποστολής.

```

unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval) {
    esp_sleep_enable_timer_wakeup(10 * 1000000); // wake up after 30
seconds
    // Save the last time a new reading was published
    previousMillis = currentMillis;
    //Set values to send
    myData.id = BOARD_ID;
    myData.tempc = readDHTTemperature();
    myData.tempf = readDHTTemperatureF();
    myData.hum = readDHTHumidity();
    myData.volts = readBatteryVolts();
    myData.perc = readBatterypercentage();
    myData.drst = digitalRead(DOOR_SENSOR_PIN);

    //Send message via ESP-NOW
    esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *)
&myData, sizeof(myData));
    if (result == ESP_OK) {
        Serial.println("Sent with success");
    }
    else {
        Serial.println("Error sending the data");}
}

```

3.3.3 Επικοινωνία με τις Περιφερειακές Συσκευές

Σε αυτήν την ενότητα θα αναλυθεί ο κώδικας που έχει θα γραφτεί για την επικοινωνία με τις περιφερειακές συσκευές των αισθητηρίων συσκευών. Αυτές περιλαμβάνουν τον αισθητήρα θερμοκρασίας και υγρασίας, τη μαγνητική επαφή που “διαβάζει” τη κατάσταση της πόρτας κάθε ψυκτικής θαλάμης και της μπαταρίας. Επίσης θα αναλυθεί ο κώδικας με τον οποίο ελέγχεται το ρελέ για τον έλεγχο της εξόδου των ψυγείων, καθώς και της λειτουργίας “βαθύ ύπνου” για την ελαχιστοποίηση της κατανάλωσης.

3.3.3.1 Επικοινωνία με τον DHT22

Αρχικά, πρέπει να δηλωθούν οι βιβλιοθήκες με τις οποίες “διαβάζουμε” τον DHT22 και να δηλωθεί ποιος αισθητήρας από την οικογένεια DHT χρησιμοποιείται και σε ποια επαφή του ESP32 είναι συνδεδεμένος.

```
#include <Adafruit_Sensor.h>
#include <DHT.h>

// Digital pin connected to the DHT sensor
#define DHTPIN 27

#define DHTTYPE    DHT22    // DHT 22 (AM2302)

DHT dht(DHTPIN, DHTTYPE);
```

Έπειτα, κατασκευάζονται τρεις συναρτήσεις, στις οποίες δηλώνεται μια δεκαδική μεταβλητή για κάθε μία από τις τρεις μετρήσεις. Αν η μέτρηση είναι ανεπιτυχής επιστρέφει κενό, ενώ αν είναι επιτυχής επιστρέφει τη μεταβλητή. Η πρώτη συνάρτηση είναι για τη θερμοκρασία σε βαθμούς Κελσίου, η δεύτερη συνάρτηση είναι για τη θερμοκρασία σε βαθμούς Φαρενάιτ, ενώ η τρίτη συνάρτηση είναι για την υγρασία σε ποσοστό %.

```
//function to read temperature in celcius
float readDHTTemperature() {
  float tc = dht.readTemperature();
  if (isnan(tc)) {
    Serial.println("Failed to read from DHT sensor!");
    return 0;
  }
  else {
    Serial.println(tc);
    return tc;}
}

//function to read temperature in fahrenheit
float readDHTTemperatureF() {
  float tf= dht.readTemperature(true);
  if (isnan(tf)) {
    Serial.println("Failed to read from DHT sensor!");
    return 0;
  }
```

```
    }
    else {
        Serial.println(tf);
        return tf;}
}
//function to read the humidity
float readDHTHumidity() {
    float h = dht.readHumidity();
    if (isnan(h)) {
        Serial.println("Failed to read from DHT sensor
        return 0;
    } else {
        Serial.println(h);
        return h;
    }
}
```

Το μόνο το οποίο μένει, είναι να ενεργοποιήσουμε τον αισθητήρα μέσα στη συνάρτηση ‘setup()’ μέσω της εντολής ‘`dht.begin()`’.

3.3.3.2 Επικοινωνία με τη Μαγνητική Επαφή

Το μόνο το οποίο χρειάζεται για να ‘διαβάσει’ ο ελεγκτής τη μαγνητική επαφή είναι να δηλωθεί σε ποια επαφή είναι συνδεδεμένος και έπειτα να τεθεί μέσα στη συνάρτηση ‘setup()’ ως είσοδος. Επίσης δηλώνεται και μια μεταβλητή ‘doorState’ η οποία αντιπροσωπεύει την κατάσταση της πόρτας.

```
const gpio_num_t DOOR_SENSOR_PIN = GPIO_NUM_33;
int doorState;

void setup() {
    pinMode(DOOR_SENSOR_PIN, INPUT_PULLUP);
}
```

3.3.3.3 Επικοινωνία με τη Μπαταρία

Για να ‘διαβάσει’ ο ελεγκτής την τιμή της τάσης της μπαταρίας 18650 γίνεται η χρήση της βιβλιοθήκης <Pangodream_18650_CL.h>. Μέσα σε αυτήν, υπάρχει ένας πίνακας σύμφωνα με τον οποίον, για κάθε τιμή τάσης που διαβάζει το τσιπ αντιστοιχεί το ποσοστό της μπαταρίας. Αυτή η αντιστοίχιση έχει γίνει με βάση το διάγραμμα ρεύματος αποφόρτισης-τάσης.

```

void Pangodream_18650_CL::_initVoltsArray(){
  _vs[0] = 2.800;
  _vs[1] = 2.840; _vs[2] = 2.880; _vs[3] = 2.930; _vs[4] = 2.970; _vs[5] = 3.000;
  _vs[6] = 3.050; _vs[7] = 3.100; _vs[8] = 3.140; _vs[9] = 3.175; _vs[10] = 3.200;
  _vs[11] = 3.220; _vs[12] = 3.240; _vs[13] = 3.260; _vs[14] = 3.280; _vs[15] = 3.290;
  _vs[16] = 3.300; _vs[17] = 3.320; _vs[18] = 3.330; _vs[19] = 3.340; _vs[20] = 3.350;
  _vs[21] = 3.360; _vs[22] = 3.370; _vs[23] = 3.380; _vs[24] = 3.390; _vs[25] = 3.400;
  _vs[26] = 3.405; _vs[27] = 3.410; _vs[28] = 3.420; _vs[29] = 3.430; _vs[30] = 3.440;
  _vs[31] = 3.445; _vs[32] = 3.450; _vs[33] = 3.460; _vs[34] = 3.470; _vs[35] = 3.480;
  _vs[36] = 3.483; _vs[37] = 3.488; _vs[38] = 3.493; _vs[39] = 3.497; _vs[40] = 3.500;
  _vs[41] = 3.505; _vs[42] = 3.510; _vs[43] = 3.518; _vs[44] = 3.525; _vs[45] = 3.530;
  _vs[46] = 3.535; _vs[47] = 3.540; _vs[48] = 3.550; _vs[49] = 3.560; _vs[50] = 3.570;
  _vs[51] = 3.575; _vs[52] = 3.580; _vs[53] = 3.590; _vs[54] = 3.600; _vs[55] = 3.610;
  _vs[56] = 3.615; _vs[57] = 3.620; _vs[58] = 3.630; _vs[59] = 3.640; _vs[60] = 3.650;
  _vs[61] = 3.660; _vs[62] = 3.670; _vs[63] = 3.680; _vs[64] = 3.690; _vs[65] = 3.700;
  _vs[66] = 3.710; _vs[67] = 3.720; _vs[68] = 3.730; _vs[69] = 3.740; _vs[70] = 3.750;
  _vs[71] = 3.755; _vs[72] = 3.760; _vs[73] = 3.765; _vs[74] = 3.770; _vs[75] = 3.780;
  _vs[76] = 3.795; _vs[77] = 3.805; _vs[78] = 3.820; _vs[79] = 3.830; _vs[80] = 3.840;
  _vs[81] = 3.850; _vs[82] = 3.865; _vs[83] = 3.880; _vs[84] = 3.890; _vs[85] = 3.900;
  _vs[86] = 3.910; _vs[87] = 3.920; _vs[88] = 3.930; _vs[89] = 3.940; _vs[90] = 3.950;
  _vs[91] = 3.960; _vs[92] = 3.970; _vs[93] = 3.980; _vs[94] = 3.990; _vs[95] = 4.000;
  _vs[96] = 4.040; _vs[97] = 4.080; _vs[98] = 4.112; _vs[99] = 4.116; _vs[100] = 4.200;
}

```

Επίσης ορίζονται 3 μεταβλητές και αρχικοποιείται η βιβλιοθήκη. Αυτές οι μεταβλητές είναι η επαφή του μικροελεγκτή με την οποία συνδέεται ή έξοδος του διαιρέτη τάσης της μπαταρίας και ως θέλεις της μετατροπής για τη μετατροπή της αναλογικής τιμής σε ψηφιακή.

```

#define ADC_PIN 34
#define CONV_FACTOR 1.7
#define READS 20

Pangodream_18650_CL BL(ADC_PIN, CONV_FACTOR, READS);

```

Τέλος, κατασκευάζονται δύο συναρτήσεις, στις οποίες δηλώνεται μια δεκαδική μεταβλητή για κάθε μία από τις δύο μετρήσεις. Αν η μέτρηση είναι ανεπιτυχής επιστρέφει κενό, ενώ αν είναι επιτυχής επιστρέφει τη μεταβλητή. Η πρώτη συνάρτηση είναι για την τάση της μπαταρίας σε Volts, ενώ η δεύτερη συνάρτηση είναι για το ποσοστό της μπαταρίας σε ποσοστό %.

```

//function to read battery voltage
float readBatteryVolts(){
  float v = BL.getBatteryVolts();
  if (isnan(v)){
    Serial.println("Failed to read battery volts");
    return 0;
  }
  else {
    Serial.println(v);
    return v;
  }
}

```

```
//function to read battery percentage
float readBatterypercentage(){
  float p = BL.getBatteryChargeLevel();
  if (isnan(p)){
    Serial.println("Failed to read battery percentage");
    return 0 ;
  }
  else {
    Serial.println(p);
    return p;
  }
}
```

3.3.3.4 Οδήγηση του Ρελέ και Λειτουργία Ύπνου

Αρχικά, δηλώνουμε τις βιβλιοθήκες και τις μεταβλητές που θα χρησιμοποιηθούν για τον έλεγχο του ρελέ. Η βιβλιοθήκη <esp_sleep.h> είναι για τη λειτουργία ύπνου, η μεταβλητή ‘triggerActive’ είναι μια μεταβλητή τύπου bool, ορίζει τη κατάσταση της εξόδου και αρχικοποιείται ως ψευδής (εννοώντας ότι η έξοδος στην αρχή είναι κλειστή) και η μεταβλητή ‘doorState’ είναι η κατάσταση της μαγνητικής επαφής (ανοιχτή-κλειστή). Επίσης ορίζεται σε ποιες επαφές θα είναι συνδεδεμένες οι δύο εισοδοί του ρελέ. Η βιβλιοθήκη <Preferences.h> επιτρέπει την αποθήκευση μεταβλητών στην εσωτερική μνήμη του ελεγκτή (EEPROM) σε περίπτωση που χάσει απότομα ενέργεια και κλείσει. Στη συγκεκριμένη περίπτωση, χρησιμοποιείται για την αποθήκευση των δύο μεταβλητών θερμοκρασίας κατωφλίου και κατάστασης πλαισίου ελέγχου, καθώς κάθε φορά που ESP32 βγαίνει από τη λειτουργία ύπνου κάνει επαναφορά, “ξεχνώντας” τις δύο μεταβλητές. Αν δεν χρησιμοποιηθεί η συγκεκριμένη βιβλιοθήκη, κάθε φορά που ο ανεμιστήρας θα είναι ανοιχτός και ο ελεγκτής θα βγαίνει από τη λειτουργία ύπνου, θα κλείνει.

```
#include <esp_sleep.h>
#include <Preferences.h>

const int RELAY_PIN1 = 25;
const int RELAY_PIN2 = 26;

bool triggerActive = false;
int doorState;
int relayState1;
int relayState2;
```

Έπειτα, μέσα στη συνάρτηση ‘setup()’ δηλώνονται οι επαφές 25 και 26 ως εξόδοι και αρχικοποιείται η κατάστασή τους. Με λογικό “0” η πρώτη είσοδος και λογικό “1” η δεύτερη είσοδος το ρελέ είναι απενεργοποιημένο. Επίσης, ενεργοποιείται η βιβλιοθήκη <preferences.h>. Τέλος, ορίζεται ο χρόνος που θα βρίσκεται ο ελεγκτής σε λειτουργία ύπνου τα 60 δευτερόλεπτα.

```
preferences.begin("relay_state", false);

relayState1 = preferences.getInt("relayState1", LOW);
relayState2 = preferences.getInt("relayState2", HIGH);
```

```
digitalWrite(RELAY_PIN1, relayState1);
digitalWrite(RELAY_PIN2, relayState2);

pinMode(RELAY_PIN1, OUTPUT);
pinMode(RELAY_PIN2, OUTPUT);

esp_sleep_enable_timer_wakeup(60 * 1000000);
```

Κατόπιν, μέσα στη συνθήκη if γίνεται ο έλεγχος του ρελέ. Όπως φαίνεται και παρακάτω, αν η πόρτα είναι κλειστή γίνεται έλεγχος της θερμοκρασίας σε σχέση με τη θερμοκρασία κατωφλίου και του πεδίου ελέγχου. Αν δηλαδή η θερμοκρασία στο εσωτερικό της θαλάμης είναι μεγαλύτερη από τη θερμοκρασία κατωφλίου, το checkbox είναι επιλεγμένο και η έξοδος δεν είναι ήδη ενεργοποιημένη, τότε ο ανεμιστήρας ανοίγει. Σε οποιαδήποτε άλλη περίπτωση, παραμένει απενεργοποιημένος. Αν η πόρτα είναι ανοιχτή, ανεξαρτήτου των παραπάνω συνθηκών, ο ανεμιστήρας παραμένει απενεργοποιημένος. Μέσα στη συγκεκριμένη συνθήκη υπάρχει και η εντολή `esp_sleep_enable_ext0_wakeup(GPIO_NUM_33, HIGH)`, η οποία “ξυπνάει” τον εκλεκτή σε περίπτωση που βρίσκεται σε λειτουργία ύπνου. Παρατηρείται επίσης ότι κάθε φορά που κάποια επαφή βρίσκεται σε κατάσταση “HIGH”, έπεται η εντολή ‘`delay(10)`’ και η ίδια η επαφή επιστρέφει σε κατάσταση “LOW”. Αυτή η εντολή είναι ουσιαστικά ο παλμός που δίνεται σε μια εκ των δύο εισόδων του ρελέ, ο οποίος διαρκεί 10 χιλιοστά του δευτερολέπτου. Τέλος, αφού πραγματοποιηθεί ο έλεγχος, αποθηκεύονται στη μνήμη του ESP32 οι δύο καταστάσεις εισόδου του ρελέ και μπαίνει σε λειτουργία ύπνου μέχρι να ξυπνήσει ξανά σε 1 λεπτό, μέσω της εντολής `esp_deep_sleep_start()`.

```
float temperature = dht.readTemperature(); // read temperature in Celsius
```

```
doorState = digitalRead(DOOR_SENSOR_PIN); // read state
```

```
if (doorState ==LOW){
  if (temperature > incomingReadings.thres3 &&
incomingReadings.checkbox3 == "true" && !triggerActive){
    Serial.println ("Turn the freezer unit on");
    triggerActive = true;
    digitalWrite (RELAY_PIN2, LOW); //turn on
    digitalWrite (RELAY_PIN1 ,HIGH);
    delay(10);
    digitalWrite (RELAY_PIN1 ,LOW);
  }
  else {
    Serial.println ("Turn the freezer unit off");
    triggerActive = false;
    digitalWrite (RELAY_PIN2, HIGH);
    delay(10);
    digitalWrite (RELAY_PIN2,LOW);
    digitalWrite (RELAY_PIN1 ,LOW);}}//turn off
```

```

if (doorState == HIGH) {
    esp_sleep_enable_ext0_wakeup(GPIO_NUM_33, HIGH);
    Serial.println("The door is open. Turn the freezer unit off");
    triggerActive = false;
    digitalWrite(RELAY_PIN2, HIGH);
    delay(10);
    digitalWrite(RELAY_PIN2, LOW);
    digitalWrite(RELAY_PIN1 , LOW);
}
// Save the state of the GPIO pins to the preferences memory
preferences.putInt("relayState1", digitalRead(RELAY_PIN1));
preferences.putInt("relayState2", digitalRead(RELAY_PIN2));
Serial.println(relayState1);
Serial.println(relayState2);
delay(1000);
esp_deep_sleep_start();
//esp_light_sleep_start();
}

```

3.4 Επίλογος

Σε αυτό το κεφάλαιο καλύφτηκε οτιδήποτε αφορά το πρόγραμμα που ελέγχει τον ελεγκτή του διακομιστή και των αισθητήριων συσκευών. Στην πρώτη ενότητα, έγινε μια εισαγωγή για τον τρόπο λειτουργίας του server. Αναλύθηκαν οι δύο μέθοδοι επικοινωνίας σε επίπεδο ιστοσελίδας-διακομιστή και διακομιστή-συσκευών, ποιες μεταβλητές ανταλλάσσονται και με ποιο τρόπο. Επίσης εξηγήθηκε ο τρόπος που λειτουργεί η ιστοσελίδα, πως χειρίζεται και πως στέλνει τις πληροφορίες που θέτει ο χρήστης. Στη δεύτερη ενότητα, έγινε μια εισαγωγή για τον τρόπο λειτουργίας του κώδικα των αισθητήριων συσκευών. Η καθεμία, λαμβάνει και στέλνει πληροφορίες από και προς το διακομιστή και αποθηκεύει κάθε μία από αυτές τις μεταβλητές που λαμβάνει, για να κάνει τους απαραίτητους ελέγχους για τη λειτουργία του ρελέ που χειρίζεται τον ανεμιστήρα της κάθε ψυκτικής μονάδας. Επίσης εξηγήθηκε πως λαμβάνει τιμές από τους περιφερειακούς αισθητήρες (αισθητήρας θερμοκρασίας-υγρασίας και μαγνητική επαφή), πως διαβάζει τη τιμή της τάσης της μπαταρίας 18650 και με ποιον τρόπο απεικονίζει το ποσοστό μπαταρίας. Τέλος, κάθε συσκευή εισέρχεται σε λειτουργία βαθύ ύπνου στην οποία απενεργοποιούνται το Wi-Fi, το Bluetooth, το RF και οι περιφερειακές επαφές. Με αυτόν τον τρόπο γίνεται σημαντική εξοικονόμηση ενέργειας και η μπαταρία δε χρειάζεται τόσο συχνά φόρτιση. Στο επόμενο και τελευταίο κεφάλαιο, θα δούμε πως όλα αυτά που εξηγήθηκαν και αναλύθηκαν σε αυτό και σε όλα τα προηγούμενα κεφάλαια, εφαρμόζονται πρακτικά, ποια είναι τα τελικά αποτελέσματα και κατά πόσο, όλες οι παραπάνω θεωρητικές προσεγγίσεις υλοποιούνται αποτελεσματικά.

Κεφάλαιο 4ο: Συμπεράσματα και προτάσεις βελτίωσης

4.1 Εισαγωγή

Στο τέταρτο και τελευταίο κεφάλαιο θα γίνουν όλες οι πειραματικές μετρήσεις και απαραίτητες δοκιμές για τη συνολική λειτουργία της ιστοσελίδας, επικοινωνίας και αισθητήριων συσκευών. Πιο συγκεκριμένα, θα συγκριθεί η θερμοκρασία που λαμβάνει η μία συσκευή, σε σχέση με ένα επαγγελματικό θερμόμετρο μέσα σε μία ψυκτική μονάδα, θα δοκιμαστεί η αντοχή της μπαταρίας (διάρκεια λειτουργίας συσκευής χωρίς φόρτιση μπαταρίας), θα ελεγχθεί η λειτουργία του κυκλώματος σταθεροποίησης και φόρτισης και θα απεικονιστεί ο παλμός του ρελέ και η διάρκεια του με παλμογράφο. Επίσης θα γίνει ένας συνολικός έλεγχος της λειτουργίας των συσκευών, μετρώντας συνολικά το ρεύμα κατανάλωσης σε κανονική λειτουργία και σε λειτουργία ύπνου. Τέλος, αφού η εργασία αποτελεί τη κατασκευή ενός οργάνου μέτρησης θερμοκρασίας που δε πρέπει να έχει παρεμβολές από το ίδιο το σύστημα (για παράδειγμα υπερθέρμανση κάποιου υλικού), θα γίνει μετρηθεί της θερμοκρασίας της συσκευής σε φόρτιση και μη με τη χρήση θερμοκάμερας. Αφού γίνουν όλες οι παραπάνω πειραματικές δοκιμές και μετρήσεις, θα προκύψουν τα αντίστοιχα συμπεράσματα για τη λειτουργία του συστήματος. Τέλος, θα γίνουν προτάσεις για τη βελτίωση του συστήματος ως πιθανές μελλοντικές επεκτάσεις και η εργασία θα κλείσει με τον επίλογο, ο οποίος θα περιλαμβάνει μία σύνοψη των λειτουργιών που αναλύθηκαν στην παρούσα πτυχιακή εργασία.

4.2 Πειραματικές Μετρήσεις και Συμπεράσματα

4.2.1 Κατανάλωση της Συσκευής

Η κάθε αισθητήρια συσκευή καταναλώνει ενέργεια σε δύο διαφορετικές συνθήκες. Όταν βρίσκεται σε κανονική λειτουργία, δηλαδή ο ESP32 δεν βρίσκεται σε λειτουργία ύπνου, και σε λειτουργία χαμηλής κατανάλωσης, όταν δηλαδή ο ESP32 βρίσκεται σε λειτουργία ύπνου. Επίσης η συσκευή καταναλώνει διαφορετική ενέργεια ανάλογα με την τάση της μπαταρίας. Όσο μικρότερη είναι η τάση της μπαταρίας τόσο μεγαλύτερη είναι η κατανάλωση. Παρακάτω θα εξετάσουμε την κατανάλωση για τέσσερις διαφορετικές καταστάσεις, **(1)** πλήρως φορτισμένη μπαταρία (4.2V), **(2)** μπαταρία σε κανονική κατάσταση (3.7V), **(3)** μπαταρία που χρειάζεται φόρτιση (2.8V) και **(4)** ESP32 σε λειτουργία ύπνου.

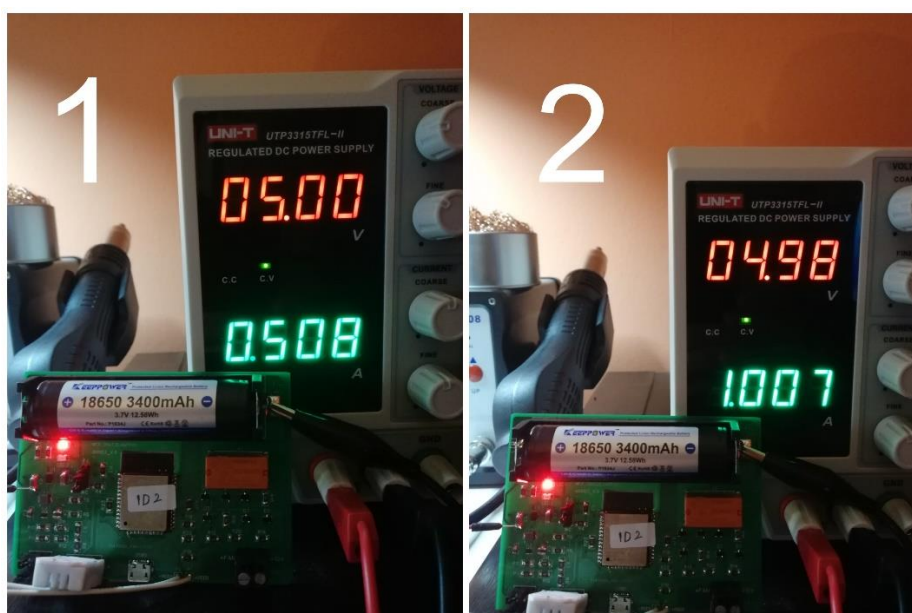


Σχήμα 4.1: Οι 4 καταναλώσεις στις 4 διαφορετικές καταστάσεις

Στο σχήμα 4.1 παρατηρούμε ότι με πλήρως φορτισμένη μπαταρία η κατανάλωση είναι περίπου 110mA (σχήμα 4.1(1)), η κατανάλωση με μπαταρία στην κανονική της κατάσταση είναι περίπου 130mA (σχήμα 4.1(2)), και η κατανάλωση με μη φορτισμένη μπαταρία είναι περίπου 170mA (σχήμα 4.1(3)). Τέλος, όσο ο ελεγκτής βρίσκεται σε κατάσταση βαθύ ύπνου η κατανάλωση δεν ξεπερνάει τα 10mA (σχήμα 4.1(4)).

4.2.2 Φόρτιση Μπαταρίας

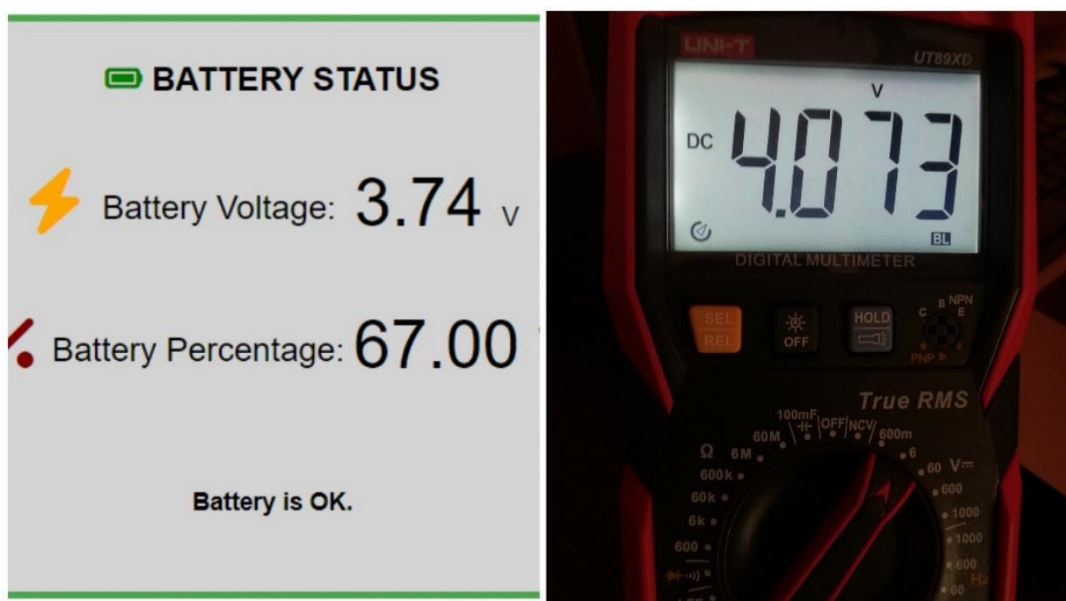
Όπως αναφέρθηκε στην ενότητα 2.4.3.3, ο χρήστης έχει την επιλογή για κανονική φόρτιση και γρήγορη φόρτιση. Στην κανονική φόρτιση, η συσκευή φορτίζει με 500mA (σχήμα 4.2(1)), ενώ στη γρήγορη φόρτιση, η αισθητήρια συσκευή φορτίζει με 1A (σχήμα 4.2(2)). Στο σχήμα 4.2 απεικονίζεται το ρεύμα φόρτισης στις 2 αυτές οι καταστάσεις, ενώ δίπλα βρίσκεται η συσκευή με αναμμένο το ενδεικτικό led.



Σχήμα 4.2: Οι δύο καταστάσεις φόρτισης της συσκευής

4.3 Σύγκριση Πραγματικής Τιμής Τάσης με την Τιμή στην Ιστοσελίδα

Στο σχήμα 4.3, απεικονίζεται η τιμή της τάσης που φαίνεται στον web server σε σχέση με την πραγματική τιμή τάσης που μετράμε με το πολύμετρο. Η διαφορά ανάμεσα σε αυτές τις 2 τιμές είναι της τάξης των 0.33V. Γενικότερα, είναι γνωστό ότι οι ADC εισοδοί του ESP32 μπορούν να έχουν σημαντικές αποκλίσεις, καθώς τα γραμμικά χαρακτηριστικά τιμής τάσης 0 – 3.3V σε σχέση με τις αναλογικές τιμές 0 – 4095 είναι φτωγά. Σύμφωνα με τα φύλλο δεδομένων του εκλεκτή, η απόκλιση είναι της κλίμακας του +/- 6%. Στον κώδικα, ο παράγοντας μετατροπής `“#define CONV_FACTOR 1.7”` ισούται με 1.7 και σύμφωνα με δοκιμές που πραγματοποιήθηκαν με άλλες τιμές, αυτή είναι η πλησιέστερη στην πραγματικότητα.



Σχήμα 4.3: Σύγκριση τάσης ιστοσελίδας με τάση πολυμέτρου

4.4 Σύγκριση Δισταθούς Ρελέ με Κανονικό Ρελέ

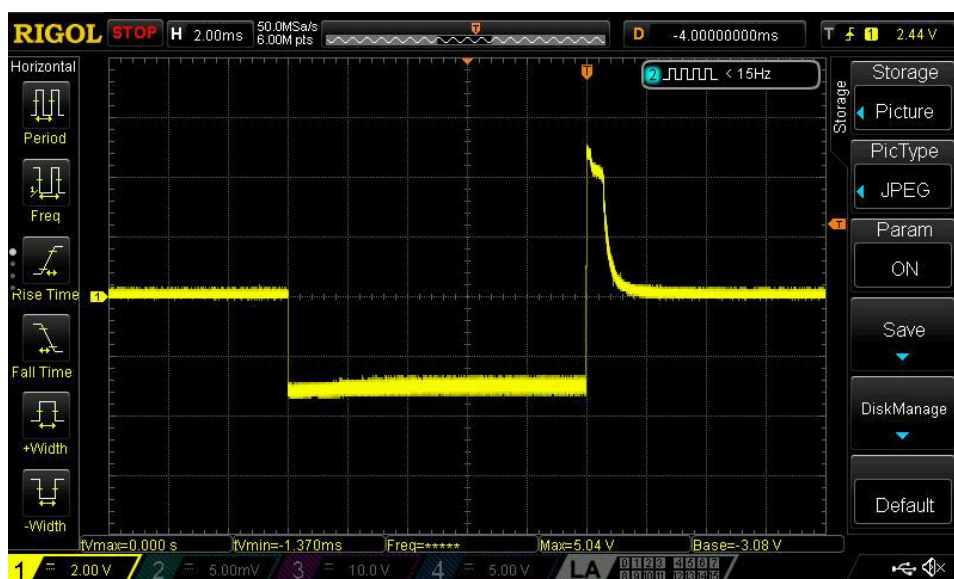
Στο σχήμα 4.4 απεικονίζεται η κατανάλωση της συσκευής όταν λειτουργεί με ένα κανονικό ρελέ, το οποίο απαιτεί συνεχώς μια τιμή τάσης να εφαρμόζεται στην είσοδο του για να παραμείνει ανοιχτό. Η κατανάλωση σε αυτή την περίπτωση είναι 210mA και μπορούμε να τη συγκρίνουμε με τις τιμές κατανάλωσης στις τρεις πρώτες περιπτώσεις του σχήματος 4.1. Παρατηρούμε ότι ιδιαίτερα στις δύο πρώτες περιπτώσεις, όπου η τιμή τάσης της μπαταρίας είναι σε φυσιολογικά πλαίσια, η εξοικονόμηση ενέργειας που γίνεται με το δισταθές ρελέ είναι σημαντική, αφού είναι σχεδόν η μισή της κατανάλωσης με ένα κανονικό ρελέ.



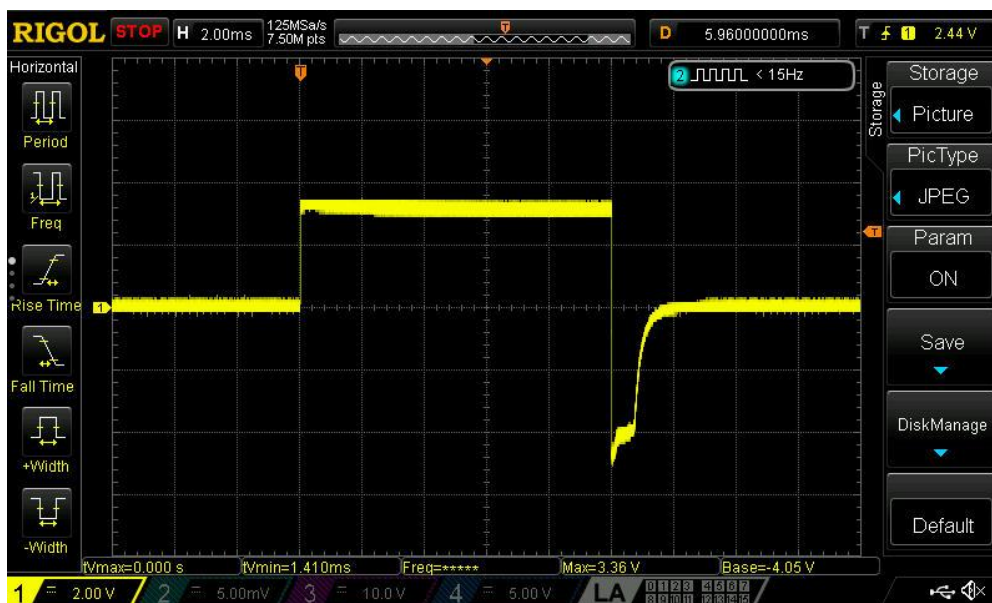
Σχήμα 4.4: Η κατανάλωση της συσκευής με ένα κανονικό ρελέ.

4.5 Μέτρηση Διάρκειας Παλμού Ρελέ με Παλμογράφο

Όπως προαναφέρθηκε, το ρελέ που χρησιμοποιείται για την εναλλαγή κατάστασης ου ανεμιστήρα των μονάδων ψύξης, είναι ένα δισταθές ρελέ που χρειάζεται ένα μοναδικό παλμό για να οπλίσει και να αφοπλίσει. Αυτός ο παλμός έχει ορισθεί μέσα από τον κώδικα να έχει διάρκεια 10ms. Στα σχήματα 4.5 και 4.6 απεικονίζει ο παλμός και η διάρκεια του κατά την όπλιση και αφόπλιση του ρελέ. Στον άξονα των x βρίσκεται η τάση, η οποία είναι στα 3.3V ενώ στον άξονα των y είναι ο χρόνος, όπου στα δύο σχήματα φαίνεται ξεκάθαρα ότι ο παλμός διαρκεί ακριβώς 10ms. Στο σχήμα 4.5 απεικονίζεται ο παλμός όταν η μαγνητική επαφή είναι ανοιχτή ενώ στο σχήμα 4.6 απεικονίζεται ο παλμός όταν η μαγνητική επαφή είναι κλειστή.



Σχήμα 4.5: Ο παλμός του ρελέ όταν η μαγνητική επαφή είναι ανοιχτή

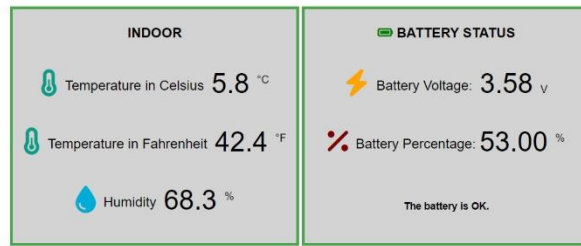


Σχήμα 4.6: Ο παλμός του ρελέ όταν η μαγνητική επαφή είναι κλειστή

4.6 Σύγκριση Θερμοκρασίας - Υγρασίας με Δεύτερο Θερμόμετρο

Για να βεβαιωθούμε ότι ο DHT22 λαμβάνει σωστές μετρήσεις θερμοκρασίας και υγρασίας, καθώς και ότι το πρωτόκολλο επικοινωνίας της αισθητήριας συσκευής με το διακομιστή, δεν θα παρεμποδίζεται όταν η συσκευή θα βρίσκεται μέσα στην ψυκτική θαλάμη, πραγματοποιήθηκε μια δοκιμή, στην οποία τοποθετήθηκε ένα επαγγελματικό θερμόμετρο και μία από τις συσκευές σε ένα ψυγείο για περίπου δύο ώρες. Στο σχήμα 4.7 απεικονίζεται η θερμοκρασία που μετράει ο DHT 22 (πάνω) και η θερμοκρασία που μετράει το θερμόμετρο (κάτω). Η απόκλιση της θερμοκρασίας είναι στους 0.5°C, ενώ η απόκλιση στην υγρασία είναι κατά προσέγγιση στο 3%, καθώς ο DHT22 μετράει περίπου 68% υγρασία, ενώ το θερμόμετρο μετράει 71% υγρασία. Εκτός από σύγκριση ενδείξεων θερμοκρασίας και υγρασίας ανάμεσα στις δύο συσκευές, δοκιμάστηκε και η επικοινωνία ανάμεσα στον διακομιστή και στην αισθητήρια συσκευή, για τυχόν παρεμβολές ή διακοπές στην επικοινωνία, που θα μπορούσαν να προκύψουν λόγω της απόστασης μεταξύ των δύο, καθώς και λόγω των τοιχωμάτων της μονάδας ψύξης που έγινε η δοκιμή. Το αποτέλεσμα ήταν θετικό, καθώς δεν παρουσιάστηκε καμία διακοπή ούτε παρεμβολή στην επικοινωνία των δύο. Αξίζει να σημειωθεί ότι η απόσταση μεταξύ διακομιστή και ψυγείου είναι περίπου 8 μέτρα και υπάρχει ένας τοίχος ανάμεσα στις συσκευές.

FREEZER UNIT CONTROL SERVER



Freezer 1 set to: 25.0

The freezer unit is working in automatic operation

Freezer 1 is off.

The door is closed.



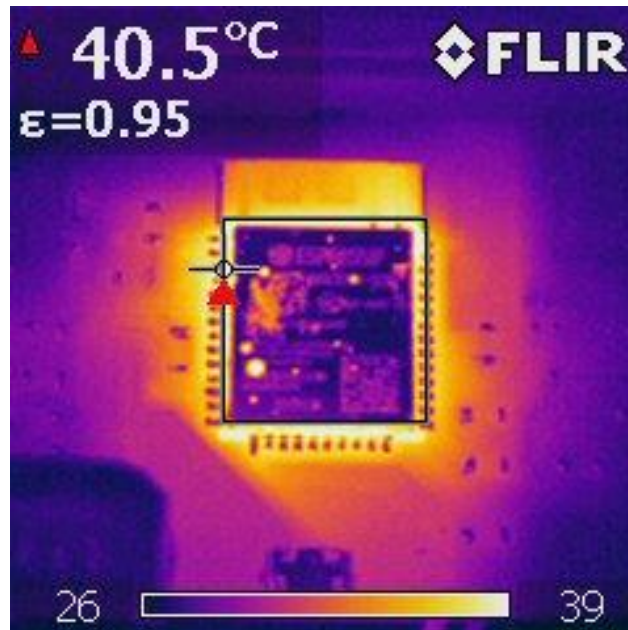
Σχήμα 4.7: Σύγκριση μετρήσεων DHT22

Σημείωση:

Το θερμόμετρο που χρησιμοποιήθηκε είναι το Bradas White Line WL-M21 με τιμή εμπορίου περίπου τα 10€

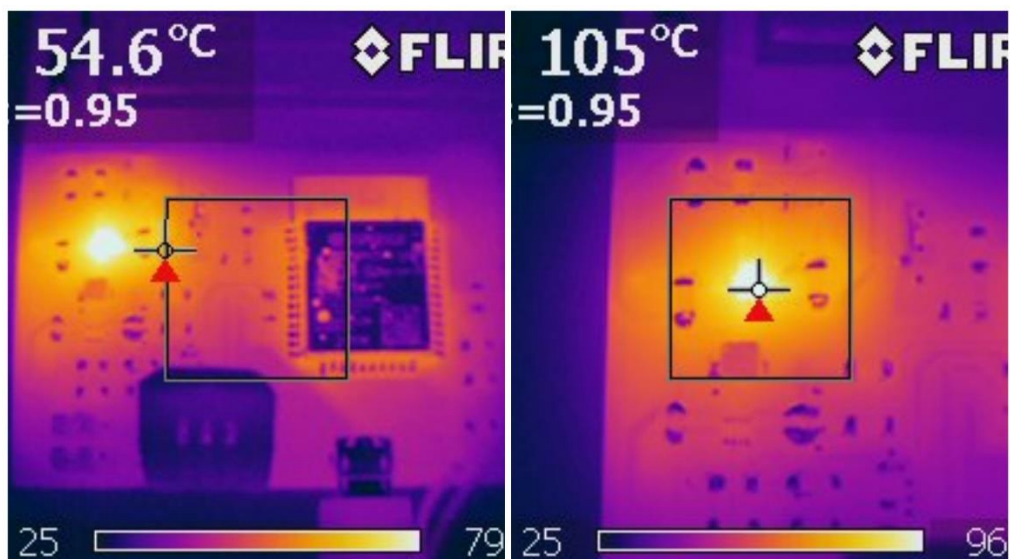
4.7 Χρήση Θερμικής Κάμερας

Επιπλέον, ένα πείραμα που έγινε είναι η μέτρηση της θερμοκρασίας της αισθητήρια συσκευής με τη χρήση θερμικής κάμερας όταν η συσκευή βρίσκεται σε ηρεμία και όταν φορτίζει **εκτός** ψυκτικής μονάδας. Ο λόγος είναι επειδή η συσκευή είναι μια αισθητήρια μονάδα που μετράει η θερμοκρασία δεν πρέπει να έχει παρεμβολές από άλλα στοιχεία του κυκλώματος που θα μπορούσαν να παραποιήσουν τις μετρήσεις του αισθητήρα. Στο σχήμα 4.8 απεικονίζεται η θερμοκρασία του θερμότερου στοιχείου της πλακέτας και αυτό είναι ο ελεγκτής ο οποίος όπως φαίνεται παρακάτω έχει θερμοκρασία περίπου 40.5°C.



Σχήμα 4.8: Η θερμοκρασία του ESP32 εκτός μονάδας ψύξης

Τέλος, στο σχήμα 4.9 απεικονίζεται η μέση θερμοκρασία της συσκευής όταν αυτή βρίσκεται σε φόρτιση με ρεύμα φόρτισης 1A. Αυτή η θερμοκρασία είναι 54.6°C εκτός ψυκτικής μονάδας (αριστερά). Παρατηρήθηκε επίσης ότι η θερμοκρασία του τσιπ φόρτισης MCP73831 μετά από περίπου μία ώρα φόρτισης είναι ιδιαίτερα αυξημένη και επομένως ενδείκνυται η φόρτιση να γίνεται εκτός ψυκτικής μονάδας όταν η συσκευή δεν λειτουργεί για να μην επηρεάζονται οι ενδείξεις θερμοκρασίας του αισθητήρα από το τσιπ (δεξιά).



Σχήμα 4.9: Η μέση θερμοκρασία φόρτισης της συσκευής και η θερμοκρασία του τσιπ φόρτισης

4.8 Μελλοντικές Επεκτάσεις

Οι μελλοντικές επεκτάσεις αποτελούν προτάσεις για τη βελτίωση και αναβάθμιση των αισθητήριων συσκευών και του συστήματος γενικότερα.

- **Λειτουργία του Web Server εκτός τοπικού δικτύου:** Αρχικά, μια πολύ βοηθητική και φιλική για το χρήστη λειτουργία, είναι η εκτέλεση της ιστοσελίδας και εκτός τοπικού δικτύου. Όλη η λειτουργία του συστήματος που επεξηγήθηκε μέχρι τώρα, βασίζεται στο γεγονός ότι ο χρήστης μπορεί να χειρίζεται τη κάθε ψυκτική μονάδα μέσω της ιστοσελίδας μόνο εντός τοπικού δικτύου, δηλαδή μόνο όταν είναι συνδεδεμένος στο Wi-Fi που είναι συνδεδεμένος και ο διακομιστής. Θα μπορούσε όμως να κάνει τον απαραίτητο έλεγχο και εκτός αυτού, (για παράδειγμα από το σπίτι του ή με τη χρήση δεδομένων). Αυτό μπορεί να επιτευχθεί με τη χρήση του προγράμματος ngrok. Το Ngrok είναι ένα λογισμικό που δημιουργεί έναν ασφαλή διαμοιραστή τοπικού διακομιστή στο διαδίκτυο. Συγκεκριμένα, το Ngrok διευκολύνει την πρόσβαση σε τοπικούς υπολογιστές και συσκευές που βρίσκονται πίσω από NATs και firewalls, δίνοντας τους δυνατότητα να εκτελούν εφαρμογές και να επικοινωνούν με άλλες συσκευές στο διαδίκτυο. Οι χρήστες του Ngrok μπορούν να δημιουργήσουν τοπικά URLs και να τα μοιραστούν με άλλους χρήστες του διαδικτύου για αναπαραγωγή των τοπικών τους σελίδων και υπηρεσιών στον παγκόσμιο ιστό. Το Ngrok διαθέτει επίσης πολλά χαρακτηριστικά ασφάλειας, όπως το HTTPS και η πιστοποίηση ταυτότητας του χρήστη, για να διασφαλίζει ότι οι εφαρμογές του χρήστη παραμένουν ιδιωτικές και ασφαλείς. Το μόνο που χρειάζεται είναι η δημιουργία λογαριασμού από την ιστοσελίδα του λογισμικού, η εγκατάσταση του στον υπολογιστή, η δημιουργία ενός AUTH token και η διαμόρφωση του κώδικα.
- **Αλλαγή αισθητήρα θερμοκρασίας – υγρασίας:** Γενικότερα, ο DHT22 δε φημίζεται ούτε για την ακρίβεια, ούτε για τη ταχύτητα του και σίγουρα δεν είναι η καλύτερη επιλογή για τη κατασκευή ενός συστήματος βιομηχανικής κλίμακας. Επομένως, η αλλαγή του με έναν αισθητήρα μεγαλύτερης, θα είναι μία αξιόλογη αναβάθμιση της συσκευής. Πιθανοί ADS18B20, ο BME280 και ο TMP75. Όλοι οι παραπάνω είναι αισθητήρες με καλύτερη ακρίβεια και χαμηλότερη κατανάλωση από τον DHT22.
- **Δυναμικά Διαπιστευτήρια:** Έχει γίνει αντιληπτό ότι στη συγκεκριμένη πτυχιακή τα διαπιστευτήρια του δικτύου (όνομα και κωδικός) στο οποίο είναι συνδεδεμένος ο server ορίζονται στατικά, μέσω του κώδικά. Ορίζονται και αποθηκεύονται δηλαδή σαν δύο μεταβλητές και χρησιμοποιούνται από τον κώδικα για να συνδεθεί ο ESP32 στο τοπικό δίκτυο. Αυτό σημαίνει ότι αν ο χρήστης θέλει να αλλάξει δίκτυο, ή θέλει να αλλάξει το όνομα και το κωδικό του router του, θα πρέπει να κάνει τις αντίστοιχες αλλαγές και στο κώδικα, κάτι που δεν είναι και πολύ πρακτικό. Μέσω της βιβλιοθήκης 'WiFi Manager'. Η συγκεκριμένη βιβλιοθήκη κατά την ενεργοποίηση του συστήματος, θέτει τον εκλεκτή ως Access Point μέσα από το οποίο, ο χρήστης μπορεί να δει όλα τα διαθέσιμα δίκτυα, να επιλέξει το δίκτυο του και να βάλει τον κωδικό του. Μετά από αυτή τη διαδικασία γίνεται η σύνδεση του ESP32 στο συγκεκριμένο δίκτυο. Αν το δίκτυο δεν είναι διαθέσιμο ή τα διαπιστευτήρια αλλάξουν, ο χρήστης έχει τη δυνατότητα να επαναλάβει εύκολα τη διαδικασία.
- **Αναπαράσταση Μετρήσεων σε Γραφικές Παραστάσεις:** Μια πολύ καλή ιδέα, είναι η αναπαράσταση των μετρήσεων θερμοκρασίας και υγρασίας σε ένα γραφικό περιβάλλον όπως μια γραφική παράσταση σε συνάρτηση με το χρόνο ή ένας γραφικός μετρητής. Με αυτό τον τρόπο, ο χρήστης θα μπορεί πολύ εύκολα να διαπιστώσει αν υπάρχουν διαρροές της ψύξης, διακοπές ρεύματος και γενικότερα να έχει μια γενική εικόνα των συνθηκών στο εσωτερικό κάθε θαλάμης.

- **Περισσότερες Συσκευές:** Το πρωτόκολλο ESP-NOW, σύμφωνα με το επίσημο documentation της Espressif μπορεί να υποστηρίξει έως 20 συσκευές συνδεδεμένες μεταξύ τους. Επομένως, αν ο χρήστης έχει έναν μεγάλο βιομηχανικό χώρο με πολλές ψυκτικές μονάδες έχει την επιλογή να αυξήσει τις αισθητήριες συσκευές.
- **Εφαρμογή:** Θα μπορούσε να κατασκευαστεί μια εφαρμογή για τον έλεγχο των θαλάμων που θα έκανε ακόμα ευκολότερη τη χρήση του συστήματος από το χρήστη. Αυτή η εφαρμογή πέρα από το πλεονέκτημα της προσβασιμότητας, έχει κι άλλα πλεονεκτήματα, όπως η γρηγορότερη αντίληψη μίας δυσχερούς κατάστασης από το χρήστη μέσω ειδοποιήσεων από το κινητό. Αν για παράδειγμα η πόρτα μιας ψυκτικής θαλάμης παραμείνει ανοικτή για μεγάλο χρονικό διάστημα, μπορεί να σταλθεί μια ειδοποίηση στο κινητό του χρήστη, για να το ειδοποιήσει να τη κλείσει.
- **Καταγραφικό:** Σε κάθε αισθητήρια συσκευή, θα μπορούσε να είναι εγκατεστημένο ένα σύστημα καταγραφής και αποθήκευσης των ενδείξεων θερμοκρασίας και υγρασίας σε μία κάρτα μνήμης, serial data logger ή σε ένα υπολογιστικό νέφος (Cloud). Θα μπορούσε για παράδειγμα να εγκατασταθεί στη πλακέτα, μία θύρα SD, στην οποία ο χρήστης θα μπορεί να βάλει μια κάρτα μνήμης για να αποθηκεύει τις μετρήσεις του δίπλα σε μία χρονική σήμανση για να γνωρίζει την ακριβή ώρα της μέτρησης, καθώς και να υπάρχει ένα μενού κατώτερης-άνωτερης καταγραφής μετρήσεων. Κάνοντας ένα βήμα παρακάτω θα μπορούσε να δημιουργηθεί ένα cloud, στο οποίο θα “ανεβαίνουν” οι μετρήσεις θερμοκρασίας, πληροφορίες για αυτές και για τα ψυγεία, καθώς και διαφόρων ειδών γραφικές παραστάσεις ή γραφικών μετρητών (gauges).
- **Εγκατάσταση Οθόνης LCD:** Τέλος, ένα βοηθητικό για το χρήστη σενάριο, θα ήταν η εγκατάσταση μίας οθόνης LCD για την απεικόνιση των μετρήσεων θερμοκρασίας-υγρασίας. Με αυτό τον τρόπο, ο χρήστης μπορεί κατευθείαν να δει τις ενδείξεις, χωρίς να χρειάζεται να ανοίξει την ιστοσελίδα. Παρόλα αυτά, η παραπάνω λειτουργία θα αυξήσει την κατανάλωση της συσκευής, καθώς απαιτείται περισσότερη ενέργεια, για η τροφοδοσία της οθόνης.

4.9 Επίλογος

Αδιαμφισβήτητα, το άκρως επίκαιρο πρόβλημα του φαινομένου του θερμοκηπίου και της ενεργειακής κρίσης, είναι ένα πρόβλημα που πρέπει να καταπολεμηθεί το αμεσότερο δυνατό, κάνοντας μικρά ή μεγάλα βήματα προς την επίτευξή του. Όλες οι τεχνικές, διαδικασίες και λειτουργίες που αναλύθηκαν στη παρούσα πτυχιακή εργασία, είχαν ως σκοπό να εκπληρώσουν το κύριο μέλημα της εργασίας: τη μείωση κατανάλωσης σε μία ψυκτική μονάδα, κάνοντας μικρά βήματα προς την επίτευξη αυτού του στόχου. Από την επιλογή του διασταθούς ρελέ και τη λειτουργία του με έναν μικρό παλμό, μέχρι την επιλογή του πρωτοκόλλου επικοινωνίας ως ένα πρωτόκολλο χαμηλής κατανάλωσης, έγινε η βέλτιστη προσπάθεια μείωσης της κατανάλωσης της κάθε αισθητήριας συσκευής, κάνοντας παράλληλα τη λειτουργία της όσο το δυνατό αποδοτικότερη. Στο πρώτο κεφάλαιο γίνεται μία γενική εισαγωγή των προβλημάτων που “γέννησαν” την ιδέα της κατασκευής και μία πολύ γενικευμένη εισαγωγή στο τρόπο λειτουργίας της. Το δεύτερο κεφάλαιο είναι αφιερωμένο στο hardware των αισθητήριων συσκευών. Πιο συγκεκριμένα, γίνεται μία προσπάθεια προσέγγισης του κάθε κυκλώματος σε θεωρητικό επίπεδο και αναλύεται ο τρόπος λειτουργίας του καθενός από αυτά. Στο τρίτο κεφάλαιο επεξηγούνται οι δύο κώδικες. Ο πρώτος κώδικας είναι του διακομιστή και ο δεύτερος των συσκευών. Καθένας από αυτούς είναι υπεύθυνος για την επικοινωνία του server με την ιστοσελίδα και την επικοινωνία του server με τις περιφερειακές συσκευές, για τον έλεγχο αλληλεπίδρασης της ιστοσελίδας με το χρήστη και τον έλεγχο των περιφερειακών υλικών πάνω στη πλακέτα. Τέλος στο τέταρτο

κεφάλαιο, γίνεται μία ένωση όλων αυτών που αναλύθηκαν και μέσω των τελικών πειραματικών μετρήσεων και δοκιμών που γίνονται, προκύπτουν τα τελικά συμπεράσματα.

Βιβλιογραφία

Internet Site

- [1] Europa, Το ανώμαλο φαινόμενο του θερμοκηπίου. Available: https://wayback.archive-it.org/12090/20201015130120/https://ec.europa.eu/environment/archives/youth/air/air_abnormalgh_el.html
- [2] Wikipedia, Greenhouse Effect. Available: https://en.wikipedia.org/wiki/Greenhouse_effect
- [3] European Commission, Causes of climate change. Available: https://climate.ec.europa.eu/climate-change/causes-climate-change_en
- [4] Climate Change Information Sheet 22. Available: <https://unfccc.int/cop3/fccc/climate/fact22.htm>
- [5] IEA, The global energy crisis Available: <https://www.iea.org/reports/world-energy-outlook-2022/the-global-energy-crisis>
- [6] BBC, Why is there a global energy crisis and who might suffer most from it? Available: <https://www.bbc.com/news/world-63430824>
- [7] IEA, Global Energy Crisis. Available: https://www.iea.org/topics/global-energy-crisis?utm_source=google&utm_medium=search&utm_campaign=energy+crisis+hub
- [8] Council of the European Union. Available: <https://www.consilium.europa.eu/en/infographics/eu-measures-to-cut-down-energy-bills/>
- [9] ΕΗΣ, Η ιστορία του Ψυγείου. Available: <https://www.eis-service.gr/el/new.aspx?id=46>
- [10] ECEP, Energy Efficiency of a Refrigerator. Available: <https://www.e-education.psu.edu/egee102/node/2010#:~:text=When%20selecting%20a%20refrigerator%2C%20remember,than%20side%2Dby%2Dside.>
- [11] ECEP, Energy Efficiency of a Refrigerator. Available: <https://www.e-education.psu.edu/egee102/node/2010>
- [12] The Internet of Things with ESP32. Available: <http://esp32.net/>
- [13] Last Minute Engineers. Available: <https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/>

Βιβλία

- [14] Albert Malvino | David Bates, *Ηλεκτρονική 8^η έκδοση*. Εκδόσεις Τζιόλα, 2017.
- [15] Herbert Schildt, *Μάθετε τη C++, 3^η Αμερικάνικη Έκδοση*. Εκδόσεις Κλειδάριθμος, 2004.
- [16] Rui Santos | Sara Santos. *Learn ESP32 with Arduino IDE 2nd Edition*. E-Book.

Datasheets

- [17] Espressif Systems, ESP32 Series Version 4.2, ESP32 datasheet, Copyright 2023 https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- [18] Microchip, Miniature Single-Cell, Fully Integrated Li-Ion, Li-Polymer Charge Management Controllers, MCP73831 datasheet, Copyright 2020: <https://ww1.microchip.com/downloads/en/DeviceDoc/MCP73831-Family-Data-Sheet-DS20001984H.pdf>
- [19] Texas Instruments, 1.5-A Output Current, High Power Density Buck-Boost Converter, TPS631000, January 2022 <https://pdf1.alldatasheet.com/datasheet-pdf/view/1371735/TI/TPS631000.html>
- [20] TE Connectivity Miniature PCB Relay PE bistable, PE014F03 datasheet, December 2021 https://gr.mouser.com/datasheet/2/418/8/ENG_DS_PE_bistable_1221-1130489.pdf
- [21] Aosong Electronics Co., Ltd Digital-output relative humidity & temperature sensor/module, DHT22 <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>
- [22] Onsemi, General Purpose Transistors PNP Silicon, BC807 datasheet, March 2022 <https://www.onsemi.com/pdf/datasheet/bc807-25w-d.pdf>
- [23] Diodes Incorporated, 45V NPN Small Signal Transistor In SOR23, BC817 datasheet, June 2016 <https://www.diodes.com/assets/Datasheets/ds11107.pdf>
- [24] Vishay Semiconductors, Small Signal Switching Diode, Dual in Series, BAV99 datasheet, 13 February 2018: <https://www.vishay.com/docs/85718/bav99.pdf>

ΠΑΡΑΡΤΗΜΑ Α: ΚΟΣΤΟΛΟΓΙΟ

ESP32 THERMOSTAT

ΚΟΣΤΟΛΟΓΙΟ

ΟΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΠΟΣΟΤΗΤΑ	ΤΙΜΗ ΜΟΝ (€)	ΣΥΝΟΛΟ (€)
MCP73831-5ACI/MC	ΦΟΡΤΙΣΤΗΣ	1,00	0,83 €	0,83 €
TPS631000DRLR	ΣΤΑΘΕΡΟΠΟΙΗΤΗΣ	1,00	1,77 €	1,77 €
PE014F03	ΡΕΛΕ	1,00	6,39 €	6,39 €
Keppower Build-in Protection 3400mAh	ΜΠΑΤΑΡΙΑ 18650	1,00	11,29 €	11,29 €
Μπαταριοθήκη 18650 με καλώδια	ΜΠΑΤΑΡΙΟΘΗΚΗ	1,00	1,20 €	1,20 €
AM2302 (DHT22)	ΑΙΣΘΗΤΗΡΑΣ ΘΕΡΜ/ ΥΓΡΑΣ.	1,00	6,90 €	6,90 €
ESP32-WROOM-32	ΜΙΚΡΟΕΛΕΓΚΤΗΣ	1,00	3,79 €	3,79 €
ΠΛΑΚΕΤΑ PCB	ΠΛΑΚΕΤΑ	1,00	1,88 €	1,88 €
USB micro smd	ΘΗΡΑ USB	1,00	0,80 €	0,80 €
GRM319R61E106KA12D	ΠΥΚΝΩΤΗΣ SMD 10uF 1206	2,00	1,06 €	1,06 €
WU-1-315-7 SICK	LED SMD κόκκινο 0805	1,00	0,23 €	0,23 €
WU-1-315-7 SICK	LED SMD πράσινο 0805	1,00	0,23 €	0,23 €
ERJ-UP8F4700V	ΑΝΤΙΣΤΑΣΗ SMD 470Ω 1206	2,00	0,34 €	0,68 €
CRM1206QFX-1003ELF	ΑΝΤΙΣΤΑΣΗ SMD 100ΚΩ 1206	4,00	0,21 €	0,84 €
GRM219R60J476ME44J	ΠΥΚΝΩΤΗΣ SMD 47uF 0805	1,00	0,10 €	0,10 €
GRM187R61A226ME15D	ΠΥΚΝΩΤΗΣ SMD 22uF 0805	1,00	0,33 €	0,33 €
DFE252012P-1R0M=P2	ΠΗΝΙΟ 1uF 4,2A 1008	1,00	0,29 €	0,29 €
CMP1206AFX-4703ELF	ΑΝΤΙΣΤΑΣΗ SMD 470ΚΩ 1206	1,00	0,46 €	0,46 €
WCR1206-56KFI	ΑΝΤΙΣΤΑΣΗ SMD 56ΚΩ 1206	1,00	0,13 €	0,13 €
AC1206FR-0782KL	ΑΝΤΙΣΤΑΣΗ SMD 82ΚΩ 1206	1,00	0,10 €	0,10 €
CRGH1206F10K	ΑΝΤΙΣΤΑΣΗ SMD 10ΚΩ 1206	4,00	0,18 €	0,72 €
CRM1206QFX-1001ELF	ΑΝΤΙΣΤΑΣΗ SMD 1ΚΩ 1206	1,00	0,21 €	0,21 €
CRGCQ1210F2K7	ΑΝΤΙΣΤΑΣΗ SMD 2ΚΩ 1206	1,00	0,31 €	0,31 €
BC-807	ΤΡΑΝΖΙΣΤΟΡ SMD PNP	2,00	0,23 €	0,46 €
BC-817	ΤΡΑΝΖΙΣΤΟΡ SMD NPN	2,00	0,23 €	0,46 €
BAV99	ΔΙΟΔΟΣ ΠΡΟΣΤΑΣΙΑΣ SMD	2,00	0,24 €	0,48 €
WCR1206-22KFI	ΑΝΤΙΣΤΑΣΗ SMD 22ΚΩ 1206	4,00	0,13 €	0,52 €
CRGP1206F2K2	ΑΝΤΙΣΤΑΣΗ SMD 2.2ΚΩ 1206	2,00	0,31 €	0,62 €
CR1206-FX-4701ELF	ΑΝΤΙΣΤΑΣΗ SMD 4.7ΚΩ 1206	2,00	0,09 €	0,18 €
AK500/2DS-5,0V	ΚΛΕΜΑ ΤΥΠΩΜΕΝΟΥ 2P	1,00	0,17 €	0,34 €
Mini SPDT - 0.3A/30VDC	ΔΙΑΚΟΠΤΗΣ ON/OFF	1,00	0,20 €	0,20 €
Νήμα PLA 1.75mm Cotton White	ΠΛΑΣΤΙΚΟ ΚΑΛΟΥΠΙ	1,00	2,00 €	2,00 €
ΣΥΝΟΛΟ ΜΟΝΑΔΑΣ				45,80
ΣΥΝΟΛΟ 3 ΜΟΝΑΔΩΝ				€ 137,39
ESP32-DEVKIT V1				€ 9,90
ΛΟΙΠΕΣ ΔΑΠΑΝΕΣ				--
TOTAL €				147,29