

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Διπλωματική Εργασία

Ανάπτυξη συστήματος λήψης
μετρήσεων και ελέγχου με χρήση
Arduino.

2019

Ντουκάκης Παναγιώτης

Επιβλέπων: Καθ. Ι.Κ. Νικολός

Χανιά, Ιανουάριος 2019

Περιεχόμενα

Κεφάλαιο 1 - Εφαρμογές με χρήση πλακέτας Arduino	4
1.1 Μετρήσεις Καιρικών Συνθηκών	4
1.2 Απομακρυσμένος Έλεγχος Ηλεκτρικών Συσκευών	5
1.3 Αυτόνομο Ρομποτικό Όχημα	5
1.4 Έλεγχος αυτόνομου οχήματος	6
1.5 Σύστημα συναγερμού οικίας	7
1.6 Ποδήλατο με ηλεκτρική υποβοήθηση	8
1.7 Σύστημα μέτρησης ροής υγρού	8
1.8 Παρακολούθηση ασθενούς	9
1.9 Αυτόματο πότισμα	10
1.10 Παρακολούθηση ύπνου ασθενούς	10
Κεφάλαιο 2 – Η πλατφόρμα Arduino	12
2.1 Περιγραφή του Arduino	12
2.1.1 Υλικό (Hardware)	13
2.1.2 Shields	13
2.1.3 Λογισμικό	13
2.2 Ιστορική αναδρομή	14
Κεφάλαιο 3 – Τα μετρούμενα μεγέθη	18
3.1 Θερμοκρασία	18
3.2 Υγρασία	20
Κεφάλαιο 4 - Processing	22
4.1 Γενικά	22
4.2 IDLE	22
Κεφάλαιο 5 – Το ψυγείο	24
5.1 Γενικά	24
5.2 Ιστορία	24
5.3 Αρχή λειτουργίας	24
Κεφάλαιο 6 – Κατασκευή της διάταξης	26
6.1 Υλικός εξοπλισμός	26
6.2 Συνδεσμολογία	29
6.2.1 Breadboard	29

6.2.2 Αισθητήρας DHT11	30
6.2.3 Οθόνη LCD 1602.....	31
6.2.4 Το ρελέ	32
Κεφάλαιο 7 - Προγραμματισμός Arduino.....	34
7.1 Προγραμματισμός πλακέτας Arduino	34
7.2 Βιβλιοθήκες και δηλώσεις μεταβλητών	34
7.3 Συνάρτηση void setup()	36
7.4 Συνάρτηση void loop()	37
Κεφάλαιο 8 - Προγραμματισμός Processing	40
8.1 Αρχικοποιήσεις-Δηλώσεις	40
8.2 Συνάρτηση void setup()	40
8.3 Συνάρτηση void init_graph().....	41
8.4 Συνάρτηση void draw()	42
Κεφάλαιο 9 – Συμπεράσματα.....	45
9.1 Λειτουργία του ψυγείου	45
9.2 Τα θετικά χαρακτηριστικά της προτεινόμενης διάταξης	51
9.2.1 Επεκτάσιμη διάταξη.....	51
9.2.2 Προσαρμοστικότητα	52
Βιβλιογραφία	53

Κεφάλαιο 1 - Εφαρμογές με χρήση πλακέτας Arduino

Στην παρούσα εργασία θα παρουσιαστεί μία μεθοδολογία για τον έλεγχο της λειτουργίας ενός ψυγείου μέσω της πλακέτας Arduino Uno, ώστε να μπορεί να γίνεται αυτόματη ρύθμιση της θερμοκρασίας του, καθώς και της εκκίνησης και της παύσης του ηλεκτροκινητήρα του. Επίσης θα μπορούμε να λαμβάνουμε μετρήσεις και να τις απεικονίζουμε σε πραγματικό χρόνο με την χρήση κατάλληλης εφαρμογής. Θα δούμε δηλαδή πως μπορούν να συνδυαστούν το πρόγραμμα για την λήψη και απεικόνιση δεδομένων Processing, το οποίο είναι γραμμένο σε Java, με την πλακέτα Arduino Uno, ώστε να εμφανίζονται στην οθόνη διαγράμματα σχετικά με τις μετρήσεις της θερμοκρασίας. Θα παρέχεται επίσης η δυνατότητα να παρατηρούμε πώς αυτή μεταβάλλεται με το χρόνο αλλά και ανάλογα με το θερμοκρασιακό περιβάλλον που επικρατεί στο χώρο λειτουργίας του ψυγείου. Στη συνέχεια θα αναλυθούν τόσο η διαδικασία κατασκευής της ηλεκτρονικής διάταξης, όσο και η διαδικασία προγραμματισμού. Η σημασία της συγκεκριμένης εργασίας βρίσκεται στις σημαντικές δυνατότητες που παρέχει για τη δημιουργία εύκολα αναβαθμιζόμενων συστημάτων ελέγχου οικιακών συσκευών, οι οποίες μπορούν να αποκτήσουν και ευφυή χαρακτηριστικά, ενώ σε δεύτερο επίπεδο μπορούν να επικοινωνούν μεταξύ τους για βελτιστοποίηση της λειτουργίας τους. Φυσικά, στα πλαίσια της παρούσας εργασίας, πραγματοποιήθηκε μόνο ένα αρχικό βήμα προς την κατεύθυνση αυτή.

Για να γίνει κατανοητή η ευκολία και η διάδοση της χρήσης της πλατφόρμας Arduino, θα γίνει στη συνέχεια μια παρουσίαση αντίστοιχων εργασιών, που πραγματοποιήθηκαν από άλλους φοιτητές στο παρελθόν.

1.1 Μετρήσεις Καιρικών Συνθηκών

Ο Μπούλης Σταμάτιος [Μπούλης, 2014] από το ΤΕΙ της Καβάλας χρησιμοποίησε μια πλακέτα Arduino για να μπορεί να λαμβάνει μετρήσεις σχετικά με τις καιρικές συνθήκες που επικρατούν στην πόλη. Πιο συγκεκριμένα, χρησιμοποίησε μια πλακέτα Arduino Ethernet, με την οποία υπάρχει δυνατότητα επικοινωνίας μέσω ίντερνετ, καθώς και συγκεκριμένους αισθητήρες. Ο μικροελεγκτής δέχεται τα δεδομένα από τους αισθητήρες, τα επεξεργάζεται και στην συνέχεια τα ανεβάζει στην ιστοσελίδα που έχει κατασκευάσει ο φοιτητής. Με τον τρόπο αυτό, οποιοσδήποτε επισκεφτεί την ιστοσελίδα μπορεί να πάρει

πληροφορίες για την θερμοκρασία, την υγρασία και την φωτεινότητα της περιοχής, καθώς και τα ποσοστά του μονοξειδίου του άνθρακα στην ατμόσφαιρα. Οι αισθητήρες που χρησιμοποιήθηκαν είναι ένας ψηφιακός αισθητήρας, που παίρνει τιμές για την ατμοσφαιρική πίεση, ένας ψηφιακός αισθητήρας, που λαμβάνει τιμές για τη θερμοκρασία και την υγρασία και έχει τροφοδοσία στα 5 V, και ένας αναλογικός αισθητήρας που μετράει την φωτεινότητα. Τέλος, συνδέθηκε ένας αναλογικός αισθητήρας, που μετράει την ποσότητα μονοξειδίου του άνθρακα στην ατμόσφαιρα.

1.2 Απομακρυσμένος Έλεγχος Ηλεκτρικών Συσκευών

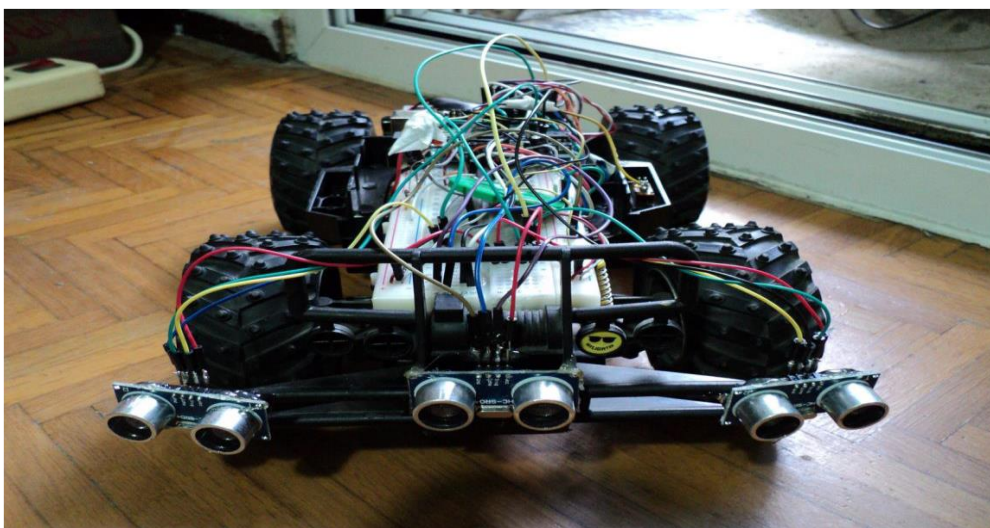
Ένα άλλο παράδειγμα χρήσης της πλακέτας Arduino είναι η διπλωματική εργασία του Χρήστου Κατσανού [Κατσανός, 2015] από το Πανεπιστήμιο της Δυτικής Μακεδονίας, ο οποίος κατάφερε να κάνει απομακρυσμένο έλεγχο ηλεκτρικών συσκευών μέσω Arduino.

Πιο συγκεκριμένα, μέσω μια πλακέτας Arduino Uno και ενός Ethernet shield υλοποίησε απομακρυσμένο έλεγχο συγκεκριμένων ρελέ. Το Ethernet shield είναι μία πλακέτα η οποία προσαρμόζεται πάνω στα pin του Arduino Uno και παρέχει τη δυνατότητα σύνδεσης του Arduino με το διαδίκτυο. Το shield δεν απαιτεί τροφοδοσία, αφού τροφοδοτείται από το ίδιο το Arduino. Η λειτουργία της συσκευής αυτής είναι η εξής: συνδέεται αρχικά στο διαδίκτυο με τον σέρβερ και στέλνει ενημέρωση σχετικά με την κατάσταση των ρελέ από τους αισθητήρες ρεύματος (αμπερόμετρο). Εν συνεχεία κλείνει την σύνδεση με την ιστοσελίδα για 4 δευτερόλεπτα, ενώ στη συνέχεια επαναλαμβάνει τη διαδικασία, κάνοντας όποια αλλαγή κατάστασης της έχει ζητηθεί από τον σέρβερ. Αν η συσκευή δεν έχει συνδεθεί για πάνω από 8 δευτερόλεπτα με τον σέρβερ τότε γίνεται επανεκκίνηση της συσκευής. Επίσης η συσκευή διαθέτει διάφορα leds και beeper για να ειδοποιεί σχετικά με την κατάσταση λειτουργίας της (αν είναι συνδεδεμένη στο διαδίκτυο ή όχι, κατάσταση οικονομικής λειτουργίας κ.λπ.).

1.3 Αυτόνομο Ρομποτικό Όχημα

Το επόμενο παράδειγμα χρήσης της πλακέτας Arduino είναι αυτό της Μακράς Αικατερίνης [Μακρά, 2016] από το ΤΕΙ Κεντρικής Μακεδονίας, η οποία κατασκεύασε ένα μικρό αυτόνομο ρομποτικό όχημα με έναν μικροελεγκτή Arduino. Παίρνοντας ένα τηλεκατευθυνόμενο αυτοκίνητο, προσθέτοντάς του μερικούς αισθητήρες και προγραμματίζοντας κατάλληλα την πλακέτα Arduino, κατάφερε να μπορεί το όχημα να κινείται και να αποφεύγει τυχόν εμποδία που παρουσιάζονται στο δρόμο του.

Τα pin του Arduino Uno δεν μπορούν να δώσουν το απαιτούμενο ρεύμα στα μοτέρ του τηλεκατευθυνόμενου οχήματος που χρησιμοποιήθηκε. Για αυτό το λόγο χρησιμοποιήθηκε το L293D, το οποίο κρύβει στο εσωτερικό του τέσσερα transistor και είναι ένα κατάλληλο ολοκληρωμένο για τον έλεγχο έως και 2 DC ηλεκτροκινητήρων. Το L293D είναι εύκολο στην συνδεσμολογία και μπορεί να δώσει αρκετά υψηλές τάσεις έως 35 V, ένα πλεονέκτημα που όμως πρέπει να προσέξει κανείς για να μην οδηγηθεί στο κάψιμο των ηλεκτροκινητήρων. Στην συνέχεια η κ. Μακρά προσάρμοσε κάποιους αισθητήρες πάνω στην κατασκευή για να μπορέσει να ελέγξει την κίνηση του. Πιο συγκεκριμένα, χρησιμοποίησε τρεις αισθητήρες υπερήχων HC-SR04, οι οποίοι με υπέρηχους ανιχνεύουν αν υπάρχει κάποιο εμπόδιο κατά την κίνηση του οχήματος. Χρησιμοποίησε επίσης ένα ηχείο, το οποίο ειδοποιεί ηχητικά αν προκύψει κάποιο εμπόδιο. Η υλοποίηση παρουσιάζεται στην επόμενη Εικόνα.



Εικόνα 1.1: Αυτόνομο ρομποτικό όχημα με χρήση Arduino.

1.4 Έλεγχος αυτόνομου οχήματος

Το παράδειγμα στη συνέχεια αναφέρεται και αυτό στον έλεγχο ενός οχήματος μέσω μιας πλακέτας Arduino. Ο Πετράκης Ζαχαρίας του ΤΕΙ Κρήτης, [Πετράκης, 2014] στα πλαίσια της πτυχιακής του εργασίας, υλοποίησε τον τηλεκατευθυνόμενο έλεγχο μικρού οχήματος, και συγκεκριμένα να εκτελεί τις βασικές εντολές κίνησης εμπρός, πίσω, αριστερά, δεξιά. Ο έλεγχος του οχήματος γίνεται ασύρματα με Bluetooth, μέσω μιας εφαρμογής android, που έχει εγκατασταθεί σε ένα tablet. Πάνω στο σασί του οχήματος υπάρχουν 4 dc motors, ένα σε κάθε τροχό. Χρησιμοποιήθηκε επίσης το L298N Dual H-Bridge Motor Driver για τον έλεγχο των ηλεκτροκινητήρων (όπως και στο προηγούμενο παράδειγμα όπου χρησιμοποιήθηκε ένα παρόμοιο μοντέλο), ένας αισθητήρας απόστασης, μία οθόνη LCD,

ένα Bluetooth module HC-06, καθώς και μπαταρίες σε ειδικές θήκες για την τροφοδοσία τόσο του Arduino αλλά και του motor shield, που χρησιμοποιήθηκε. Το motor shield, που χρησιμοποιήθηκε, βασίζεται στο L298N, με το οποίο μπορούμε να ελέγξουμε έως 2 dc motors και να ρυθμίσουμε ξεχωριστά την ταχύτητά τους. Ο αισθητήρας υπερήχων χρησιμοποιείται και σε αυτό το παράδειγμα για να γίνει διακοπή της κίνησης από το Arduino εάν ο αισθητήρας αντιληφθεί κάποιο εμπόδιο. Επίσης έχουν χρησιμοποιηθεί δύο διακόπτες για να ανοίγει και να κλείνει η τροφοδοσία του ρεύματος, στο Arduino και στο shield αντίστοιχα.

Στην συνέχεια, αφού πραγματοποιήθηκε η συνδεσμολογία και ο προγραμματισμός με τις κατάλληλες βιβλιοθήκες των παραπάνω εξαρτημάτων, υλοποιήθηκε η εφαρμογή για τον απομακρυσμένο έλεγχο του οχήματος. Μέσω κάποιων συναρτήσεων (incomingByte), γίνεται έλεγχος των δεδομένων (bytes) που προέρχονται από την εφαρμογή android και μεταφράζεται από το Arduino στην κίνηση που επιθυμεί ο χειριστής της εφαρμογής να εκτελέσει το όχημα.

1.5 Σύστημα συναγερμού οικίας

Ο κ. Φωτεινογιαννόπουλος Ανδρέας από το Πανεπιστήμιο Πειραιά [Φωτεινογιαννόπουλος, 2016] υλοποίησε μια κατασκευή με Arduino με στόχο την προστασία οικιών από κινδύνους, όπως πυρκαγιά, ληστεία κ.λπ. Ο μηχανισμός αυτός με την βοήθεια διαφόρων αισθητήρων παρέχει τη δυνατότητα, μέσω ενός server, να ενημερώνει τους ιδιοκτήτες-ενοίκους της οικίας για πιθανό κίνδυνο, ώστε να αποτρέψουν τα χειρότερα. Επίσης δίνει τη δυνατότητα στον χρήστη μέσω διαδικτύου να ελέγχει κάποιες λειτουργίες που γίνονται μέσα στο σπίτι, καθώς επίσης και να επιβλέπει τις λειτουργίες αυτές σε πραγματικό χρόνο.

Για τις ανάγκες της εργασίας αυτής, μερικοί από τους βασικούς αισθητήρες που χρησιμοποιήθηκαν είναι ένας αισθητήρας κίνησης, ένας αισθητήρας θερμοκρασίας, ένας αισθητήρας φωτεινότητας, διάφορα led, καθώς και μια σειρήνα (buzzer) για να παράγει ήχους όταν είναι αυτό επιθυμητό. Για την σύνδεση του Arduino με το διαδίκτυο χρησιμοποιήθηκε η πλακέτα HanRun Ethernet, που διαθέτει πολλά pins εισόδου-εξόδου. Επιπλέον, χρησιμοποιήθηκε ένα GSM Shield, το οποίο παρέχει τη δυνατότητα να γίνει χρήση του Arduino σαν ένα κανονικό τηλέφωνο (λήψη αποστολή μηνυμάτων GPS κτλ). Τέλος, για τον απομακρυσμένο έλεγχο του συστήματος, χρησιμοποιήθηκε ένα Infrared Control.

Κάποιες από τις εργασίες που εκτελεί το προτεινόμενο σύστημα είναι να ελέγχει την θερμοκρασία του χώρου και να ειδοποιεί με συναγερμό τον ιδιοκτήτη όταν αυτή ξεπεράσει ένα όριο (πυρκαγιά), να ελέγχει την φωτεινότητα και όταν αυτή πέσει κάτω από ένα όριο ανάβει ένα φως για να φαίνεται από το εξωτερικό του σπιτιού ότι κατοικείται για να επιτυγχάνεται αποτροπή από τυχόν ληστεία, με έναν αισθητήρα επιτάχυνσης ειδοποιεί τον ιδιοκτήτη με συναγερμό σε περίπτωση σεισμού. Όλες αυτές οι λειτουργίες μπορούν να ελέγχονται μέσω του server που έχει κατασκευαστεί, ενώ επίσης μπορεί να οριστεί και ένας αριθμός κινητού τηλεφώνου, όπου μπορούν να στέλνονται τα αποτελέσματα. Τέλος χρησιμοποιήθηκε ένα KeyPad στην πόρτα της οικίας με κωδικό πρόσβασης. Όταν το keypad είναι ενεργό λειτουργούν όλοι οι αισθητήρες. Ο χρήστης έχει τη δυνατότητα να απενεργοποιήσει τον εσωτερικό αισθητήρα κίνησης όταν βρίσκεται μέσα στο σπίτι και επιθυμεί να κινείται ελεύθερα μέσα σε αυτό.

1.6 Ποδήλατο με ηλεκτρική υποβοήθηση

Ο κ. Κορέτσης Γεώργιος από το Αλεξάνδρειο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Θεσσαλονίκης [Κορέτσης, 2013] στα πλαίσια της Πτυχιακής του εργασίας έκανε χρήση της πλακέτας Arduino σε τρίκυκλο ποδήλατο με ηλεκτρική υποβοήθηση. Χρησιμοποίησε το Arduino για να ελέγχει τις στροφές που θα δίνει ο κάθε ένας από τους δύο κινητήρες στις πίσω ρόδες το ποδηλάτου, με σκοπό το ποδήλατο να μην βγαίνει εκτός τροχιάς. Υπολόγισε πόση διαφορά στροφών θα πρέπει να έχει ο εξωτερικός από τον εσωτερικό τροχό σε μία στροφή και με την βοήθεια ενός αισθητήρα κλίσης και ενός ποτενσιόμετρου γκαζιού κατάφερε προγραμματίζοντας κατάλληλα το Arduino να ελέγχει επιτυχώς το τρίκυκλο. Χρησιμοποίησε επίσης ένα datalogger, ένα μικροτσίπ, το οποίο σε συνεργασία με το Arduino αποθηκεύει κάποια δεδομένα όπως η ταχύτητα, οι στροφές του κάθε τροχού, το ρεύμα, η τάση μπαταρίας, η γωνία στροφή σε ένα αρχείο excel. Έτσι, ο οδηγός μπορεί να συνδέσει την κάρτα μνήμης του μικροτσίπ με το υπολογιστή, να μελετήσει τα δεδομένα και να εξάγει διάφορα συμπεράσματα.

1.7 Σύστημα μέτρησης ροής υγρού

Ο κ. Βλαγοΐδης Νικόλαος από το Τμήμα Πληροφορικής του ΑΠΘ [Βλαγοΐδης, 2016] κατασκεύασε ένα σύστημα μέτρησης ροής υγρού σε ένα σωλήνα. Το σύστημα στέλνει και αποθηκεύει τις μετρήσεις σε cloud server (cloud Thingspeak.com), που χρησιμοποιείται για την συλλογή και επεξεργασία δεδομένων μέσω διαδικτύου.

Για την υλοποίηση της εφαρμογής γίνεται η χρήση μίας πλακέτας Arduino Uno, ένα Wifi Module ESP8266 για την σύνδεση του Arduino με το διαδίκτυο, ένα Power Bank Xiaomi, η πλατφόρμα cloud Thingspeak.com και ένα ροόμετρο-αισθητήρα φαινομένου Hall. Με την βοήθεια της συνάρτησης pulseCounter() μετρώνται οι παλμοί από τον αισθητήρα, πραγματοποιούνται οι απαραίτητοι υπολογισμοί και εξάγεται το αποτέλεσμα, δηλαδή η παροχή του υγρού (λίτρα/ λεπτό) που περνά από τον σωλήνα. Με την βοήθεια του καναλιού που κατέχει στο Thingspeak.com μπορεί να εξάγει αποτελέσματά σχετικά με το μέσο όρο, αλλά και διάφορα γραφήματα που περιγράφουν την ροή με το χρόνο κ.λπ..



Εικόνα 1.2: Σύστημα μέτρησης παροχής υγρού με χρήση Arduino.

1.8 Παρακολούθηση ασθενούς

Ο Yuhua Hu από το Kennesaw State University [Yuhua, 2015] χρησιμοποίησε την πλακέτα Arduino για ιατρικό σκοπό. Στόχος του ήταν η παρακολούθηση ενός ασθενή ασύρματα μέσω υπολογιστή, χωρίς να είναι απαραίτητη η συνεχής φυσική παρουσία μιας νοσοκόμας. Έτσι, χρησιμοποιώντας μια κάμερα, έναν αισθητήρα θερμοκρασίας και έναν αισθητήρα υγρασίας έφτιαξε ένα σύστημα παρακολούθησης του ασθενή. Αρχικά ο φοιτητής επιχείρησε να κάνει τον έλεγχο μόνο με την πλακέτα Arduino, όμως παρατήρησε ότι η ποιότητα του βίντεο δεν ήταν η επιθυμητή. Έτσι, για να λύσει αυτό το πρόβλημα, αποφάσισε να προβεί στην ακόλουθη λύση, να χρησιμοποιήσει το Arduino Mega 2560R3 για τον έλεγχο των αισθητήρων και των σερβο-κινητήρων, καθώς και μία Raspberry Pi πλακέτα με μία κατάλληλη κάμερα, για να επιτύχει καλύτερη ποιότητα εικόνας. Η

ασύρματη επικοινωνία με την πλακέτα Raspberry PI επιτυγχάνεται με ένα Wi-Fi module, ενώ η επικοινωνία με την πλακέτα Arduino γίνεται με την βοήθεια ενός Bluetooth module.

Οι πλακέτες τύπου Raspberry PI είναι open source μικροεπεξεργαστές Βρετανικής καταγωγής και η χρήση τους παρόμοια με αυτήν των πλακετών Arduino. Η διαφορά τους είναι ότι αυτές οι πλακέτες προσφέρουν μεγαλύτερες ταχύτητες επεξεργασίας από τις Arduino. Για αυτό το λόγο προσέφεραν στο project του φοιτητή καλύτερη ποιότητα εικόνας με την χρήση της κατάλληλης κάμερας (Raspberry Pi 5 MegaPixel).

1.9 Αυτόματο πότισμα

Το παράδειγμα αυτό αναφέρεται στην χρήση του Arduino για γεωργικούς σκοπούς. Πιο συγκεκριμένα, ο κ. Γουργιώτης Κοσμάς στα πλαίσια της Πτυχιακής του εργασίας [Γουργιώτης, 2016], υλοποίησε μια κατασκευή με σκοπό το πότισμα των καλλιεργειών ενός αγρότη από το σπίτι, χωρίς να χρειάζεται η μετακίνηση του στο χωράφι του. Χρησιμοποίησε μια ηλεκτροβάννα, έναν αισθητήρα υγρασίας και ένα τρανζίστορ ρελέ. Για τον απομακρυσμένο έλεγχο της πλακέτας χρησιμοποίησε ένα internet shield και δημιούργησε μια απλή ιστοσελίδα, μέσω της οποίας ο χρήστης θα ελέγχει την κατάσταση του ποτίσματος. Αναλυτικότερα, ο χρήστης, μπαίνοντας στην ιστοσελίδα, μπορεί να δει το ποσοστό υγρασίας που διατηρεί το χώμα και να πατήσει να ξεκινήσει το πότισμα. Αν ο χρήστης πατήσει να ξεκινήσει το πότισμα, τότε ενεργοποιείται η ηλεκτροβάννα μέσω του Arduino και το πότισμα ξεκινάει. Στην συνέχεια ο ενδιαφερόμενος μπορεί να σταματήσει το πότισμα με το πάτημα ενός κουμπιού μόλις δει το επιθυμητό επίπεδο υγρασίας στην οθόνη του. Μια επιπλέον επιλογή που έχει ο χρήστης είναι η επιλογή αυτόματου ποτίσματος, όπου σε αυτή την περίπτωση αποφασίζει το Arduino πότε θα ξεκινήσει το πότισμα, παρατηρώντας το ποσοστό υγρασίας του χώματος.

1.10 Παρακολούθηση ύπνου ασθενούς

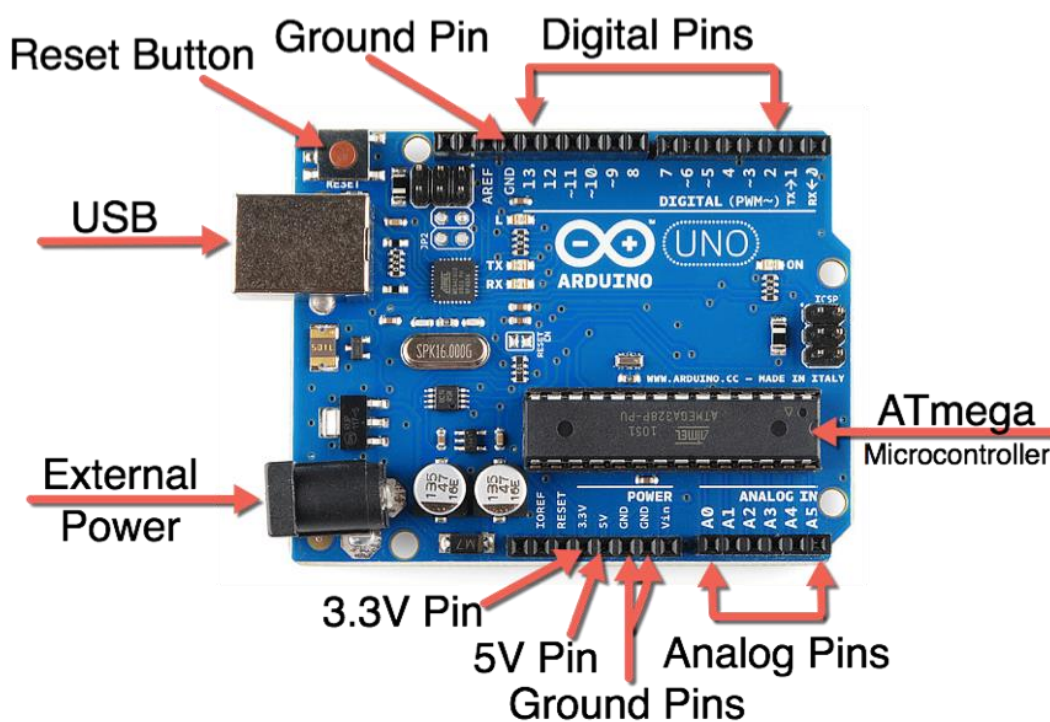
Ο κ. Vladyslav Larionov από το Πολυτεχνείο της Τσεχίας του Τμήματος Ηλεκτρολόγων Μηχανικών [Vladyslav, 2017], χρησιμοποίησε την πλακέτα Arduino για ιατρικό σκοπό. Χρησιμοποίησε την πλακέτα και διάφορους αισθητήρες (κίνησης, θερμοκρασίας) για να μπορεί να παρακολουθεί την διάρκεια ύπνου ενός ασθενή. Υλοποίησε δηλαδή ένα <<έξυπνο>> κρεβάτι. Χρησιμοποιώντας αισθητήρες κίνησης (στα χέρια και στα πόδια του ασθενή), θερμοκρασίας, θορύβου και φωτισμού, προσπαθεί να κάνει διαχωρισμό από τον ξεκούραστο ύπνο και τον ύπνο με διαταραχές. Στο πείραμα πήραν μέρος 2 εθελοντές ηλικίας 19-23 χρονών και μπόρεσε να τους παρακολουθεί κατά την διάρκεια του ύπνου και

να ελέγχει τα επίπεδα θορύβου, φωτισμού, έτσι ώστε να εξάγει συμπεράσματα για το πώς επηρεάζουν το κάθε άτομο στον ύπνο του. Χρησιμοποιώντας εοιταχυνσιόμετρα και αισθητήρες θερμοκρασίας στα χέρια και στα πόδια του κάθε ασθενή προσπαθεί να κάνει διαχωρισμό από τον ξεκούραστο ύπνο και τον ύπνο με διαταραχές.

Κεφάλαιο 2 – Η πλατφόρμα Arduino

2.1 Περιγραφή του Arduino

Η πλακέτα αποτελείται από τον βασικό μικροελεγκτή (Atmega) της εταιρείας Atmel και είναι βασισμένος στην 8,16,32 bit αρχιτεκτονική. Η πλακέτα έχει 12 digital pins και 6 analog pins. Τα digital pins δέχονται ή λαμβάνουν μεμονωμένες τιμές ενώ τα analog pins δέχονται ή στέλνουν ένα εύρος τιμών. Και τα δύο χρησιμοποιούνται για επικοινωνία με εξωτερικές συσκευές, ανάλογα τις ανάγκες του project. Όπως βλέπουμε και στην Εικόνα 2.1, η πλακέτα διαθέτει ένα εξωτερικό κουμπί reset για να πραγματοποιείται επανεκκίνηση αν χρειαστεί, χωρίς να είναι απαραίτητη η χρήση εξωτερικά συνδεδεμένου Η/Υ.



Εικόνα 2.1: Η πλακέτα Arduino.

Ο προγραμματισμός της πλακέτας γίνεται μέσω ενός καλωδίου USB, το οποίο συνδέεται στην USB θύρα της πλακέτας και στον υπολογιστή του προγραμματιστή. Η πλακέτα προγραμματίζεται από την γλώσσα προγραμματισμού Arduino, η οποία ουσιαστικά είναι η γλώσσα προγραμματισμού C++, εμπλουτισμένη με διάφορες συναρτήσεις, οι οποίες διευκολύνουν τον χρήστη να χρησιμοποιεί την πλακέτα σε διάφορες περιπτώσεις ελέγχου σε ένα ολοκληρωμένο προγραμματιστικό περιβάλλον IDE (Integrated Development Environment).

2.1.1 Υλικό (Hardware)

Όλα τα μοντέλα τύπου Arduino αποτελούνται από έναν μικροελεγκτή **Atmel** (ATmega328 και ATmega168) και βοηθητικά εξαρτήματα, όπως έναν ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16 MHz. Ο μικροελεγκτής είναι ουσιαστικά μια παραλλαγή μικροεπεξεργαστή, ο οποίος λόγω των πολλών εσωτερικών υποσυστημάτων που διαθέτει, μπορεί να λειτουργήσει σχεδόν αυτόνομα. Χρησιμοποιείται σε πολλά ηλεκτρονικά εξαρτήματα και αυτοματισμούς, καθώς και σε κάθε είδους αυτοκινούμενα οχήματα. Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα bootloader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής. Ο κρυσταλλικός ταλαντωτής αποτελείται στο εσωτερικό του από έναν κρύσταλλο μαζί με το κύκλωμα του ταλαντωτή που παράγει τους παλμούς. Έτσι απλά του δίνουμε τάση και μας επιστρέφει παλμούς.

2.1.2 Shields

Τα shields είναι ιδιαίτερα σημαντικά εξαρτήματα, τα οποία ταιριάζουν σε πολλά projects που είναι ανάγκη η ασύρματη επικοινωνία της πλακέτας Arduino για την λήψη πληροφοριών. Ουσιαστικά είναι τυπωμένα boards επεκτάσεων κυκλωμάτων, τα οποία συνδέονται μέσω των pins με το Arduino και μας δίνουν δυνατότητες, όπως σύνδεση στο διαδίκτυο, έλεγχο σε κινητήρες, GPS και LCD εικόνας.

2.1.3 Λογισμικό

Το ολοκληρωμένο περιβάλλον ανάπτυξης κώδικα (IDE) του Arduino είναι γραμμένο σε Java, με σκοπό να εισάγει στον προγραμματισμό όσους δεν είναι εξοικειωμένοι με την ανάπτυξη κώδικα. Διαθέτει αποσφαλμάτωση, αυτόματη αναπαραγωγή κώδικα και έναν πηγαίο επεξεργαστή κώδικα, που συνδυάζει τις αγκύλες και κάνει επισήμανση σύνταξης για την διευκόλυνση του χρήστη. Η γλώσσα προγραμματισμού είναι η C++ και στελεχώνεται από έναν τεράστιο αριθμό έτοιμων βιβλιοθηκών, που βοηθούν τον χρήστη στον προγραμματισμό των διαφόρων αισθητήρων ή εξαρτημάτων που χρησιμοποιεί. Το πρόγραμμα ονομάζεται sketch και μπορεί πολύ εύκολα να διορθωθεί (εντοπισμός λαθών σύνταξης) και να φορτωθεί στην πλακέτα με το πάτημα ενός κουμπιού.

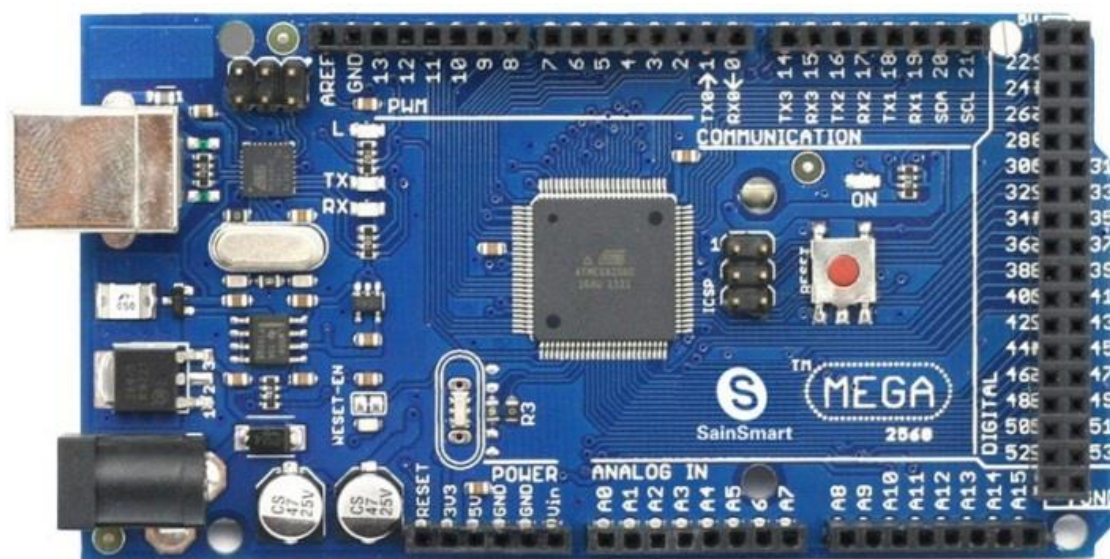
Το πρόγραμμα χωρίζεται σε δύο συναρτήσεις:

1. Την κλάση void setup(): όπου όλες οι εντολές μέσα σε αυτήν την συνάρτηση εκτελούνται μόνο και μία φορά.
2. Την κλάση void loop(): όπου όλες οι εντολές της συνάρτησης αυτής εκτελούνται συνεχόμενα όσο η πλακέτα μας βρίσκεται σε τροφοδοσία ρεύματος.

2.2 Ιστορική αναδρομή

Η πλακέτα Arduino έχει τις ρίζες της στην Ιταλία όταν οι δύο εφευρέτες της Massimo Banzi και David Cueartielles το 2005 ξεκίνησαν να παράγουν μικρές πλακέτες, προκειμένου να πετύχουν τη διαδραστική επικοινωνία του προγραμματισμού με τους μαθητές. Ξεκίνησαν να παράγουν το πρώτο μοντέλο, το οποίο το ονόμασαν Arduino της Inrea σε μια μικρή αποθήκη στην Ιβρέα της βορειοδυτικής Ιταλίας. Ανά τακτά χρονικά διαστήματα κατασκευάζονταν νέες εκδόσεις της πλακέτας, με σκοπό την κάλυψη κάποιου συγκεκριμένου χαρακτηριστικού, π.χ. την ανάγκη η πλακέτα να υπάρχει σε μικρότερο μέγεθος (Arduino mini, Σεπτέμβριος 2006). Έχουν δημιουργηθεί αρκετά μοντέλα μετά την κατασκευή του πρώτου μοντέλου. Μερικά από αυτά είναι.

- Arduino Mega (Μάρτιος 2009) (Εικόνα 2.2)



Εικόνα 2.2: Arduino Mega.

Μία πλακέτα με περισσότερες θέσεις (54 I/O pins, 16 analog pins), που δίνει τη δυνατότητα για πιο πολύπλοκα projects. Δουλεύει όπως και το Arduino Uno, με τη διαφορά τους να είναι ότι χρησιμοποιεί διαφορετικό επεξεργαστή (ATmega2560), που του δίνει τη δυνατότητα για ταχύτερους υπολογισμούς και πιο πολύπλοκα προγράμματα.

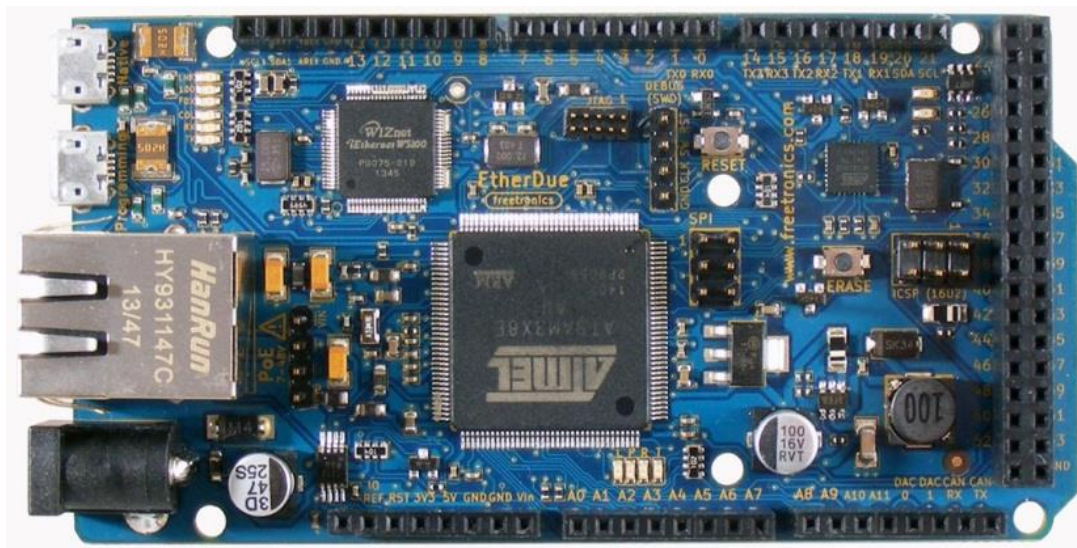
- Arduino Leonardo (Ιούλιος 2012) (Εικόνα 2.3)



Εικόνα 2.3: Arduino Leonardo.

Το Arduino Leonardo είναι ένας πίνακας μικροελεγκτών, που βασίζεται στο ATmega32u4. Διαθέτει 20 digital pins από τα οποία τα 12 μπορούν να χρησιμοποιηθούν ως analog input pins. Διαθέτει επίσης μία micro USB θύρα και μία ενσωματωμένη επικοινωνία USB, που του επιτρέπει να ανιχνεύεται ως ένα ποντίκι ή πληκτρολόγιο από τον υπολογιστή και όχι ως μια εικονική σειριακή θύρα COM/PORT.

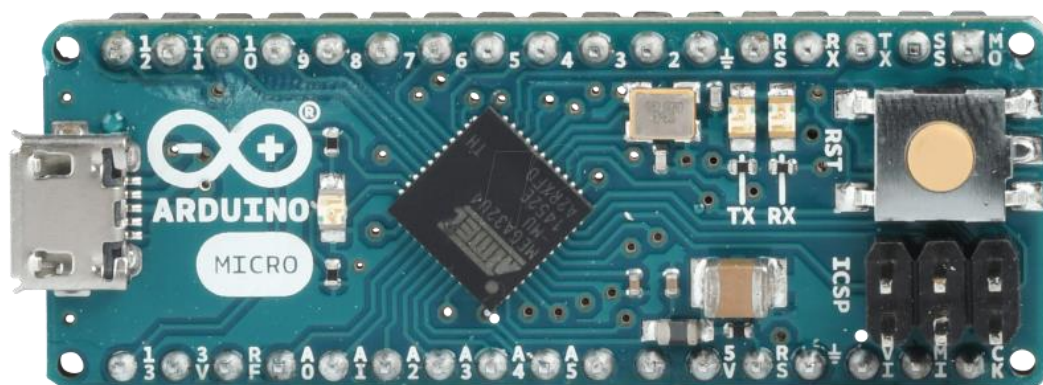
- Arduino Due (Οκτώβριος 2012) (Εικόνα 2.4)



Εικόνα 2.4: Arduino Due.

Είναι η πρώτη πλακέτα η οποία βασίστηκε σε 32bit αρχιτεκτονική με 54 digital pins και 12 analog inputs. Η μεγαλύτερη τάση εξόδου που μπορεί να διαχειριστεί αυτός ο τύπος πλακέτας είναι 3.3 volt, οποιαδήποτε τιμή παραπάνω από αυτήν μπορεί να οδηγήσει σε καταστροφή της πλακέτας.

- Arduino Micro (Νοέμβριος 2012) (Εικόνα 2.5)

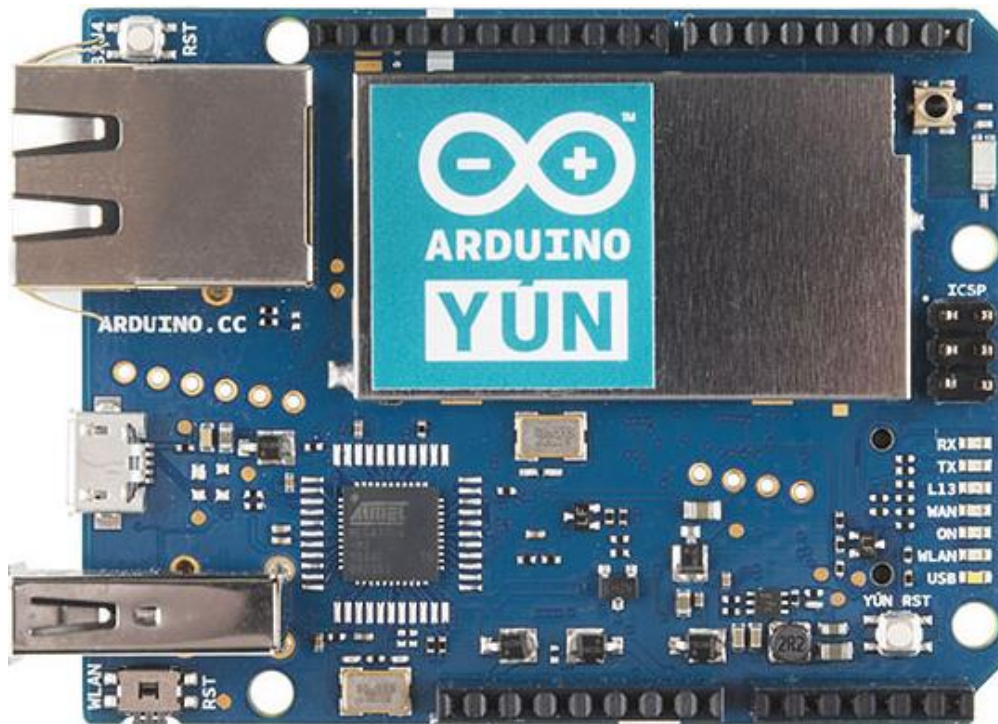


Εικόνα 2.5: Arduino Micro.

Το Arduino Micro είναι το μικρότερο μέλος της οικογένειας των πλακετών Arduino. Είναι εύκολο να το ενσωματώσει κάποιος σε καθημερινές συσκευές για να τους δώσει μία

διαδραστική μορφή. Το Micro βασίζεται στον μικροελεγκτή ATmega32U4, που διαθέτει ενσωματωμένο USB, το οποίο κάνει την πλακέτα να διαβάζεται από τον υπολογιστή ως κάποιο ποντίκι ή πληκτρολόγιο και όχι ως μια εικονική σειριακή θύρα COM/PORT. Διαθέτει 20 digital pins, από τα οποία τα 12 μπορούν να χρησιμοποιηθούν ως analog inputs.

- Arduino Yun (Μάιος 2013) (Εικόνα 2.6)



Εικόνα 2.6: Arduino Yun.

Το Arduino Yun είναι ένας πίνακας μικροελεγκτών με βάση το ATmega32u4 και το Atheros AR9331, που υποστηρίζει το λειτουργικό Linux. Η πλακέτα διαθέτει ενσωματωμένη υποστήριξη Ethernet και WiFi, θύρα USB-A, υποδοχή κάρτας micro-SD, 20 digital και 12 analog pins. Σε πολλές χώρες απαγορεύεται η πώληση συσκευών με εγκατεστημένο το WIFI και η συγκεκριμένη πλακέτα χρήζει ακόμη πιστοποίησης. Το Yun είναι παρόμοιο με το Leonardo με το ATmega32u4.

Κεφάλαιο 3 – Τα μετρούμενα μεγέθη

3.1 Θερμοκρασία

Ο όρος θερμοκρασία είναι αρκετά κοινός στην καθημερινή μας ζωή. Το ανθρώπινο σώμα, μέσω των κατάλληλων αισθητηρίων, έχει την ικανότητα να διακρίνει τα αντικείμενα με τα οποία έρχεται σε επαφή, σε θερμότερα ή ψυχρότερα από αυτό. Εκτός της επαφής, το ανθρώπινο σώμα έχει τη δυνατότητα να διακρίνει τη διαφορά θερμοκρασίας και από απόσταση, λόγω της θερμικής ακτινοβολίας, την οποία εκπέμπουν ή απορροφούν τα διάφορα αντικείμενα και το ανθρώπινο σώμα. Επίσης, από την καθημερινή πρακτική γνωρίζουμε ότι δύο σώματα με διαφορετική θερμοκρασία, όταν έρθουν σε επαφή για ικανό χρονικό διάστημα, οι θερμοκρασίες τους τελικά τείνουν να γίνουν ίσες.

Τα αισθητήρια όργανα του ανθρώπινου σώματος όμως δεν μπορούν ουσιαστικά να μετρήσουν τη θερμοκρασία. Στην πραγματικότητα αισθάνονται την ένταση της ροής θερμότητας από το αντικείμενο προς το σώμα ή το αντίστροφο. Έτσι, στην περίπτωση που έχουμε δύο αντικείμενα ίσης θερμοκρασίας (χαμηλότερης του σώματος) από ξύλο και μέταλλο, το μεταλλικό αντικείμενο θα μας φανεί πιο ψυχρό, διότι σε αυτό η ένταση της ροής της θερμότητας από το σώμα είναι μεγαλύτερη.

Το προηγούμενο παράδειγμα δείχνει τη δυσκολία που υπάρχει ως προς τον ορισμό της θερμοκρασίας με βάση τις παρατηρήσεις μας. Ο αυστηρός ορισμός της θερμοκρασίας μπορεί να γίνει μόνο με βάση αξιώματα της Θερμοδυναμικής. Δεν περιγράφει τη φύση της θερμοκρασίας, αλλά καθορίζει ακριβή τρόπο μέτρησης και σύγκρισης των θερμοκρασιών διαφόρων σωμάτων ή συστημάτων.

Η θερμοκρασία ως μέγεθος είναι βασικό στοιχείο της Θερμοδυναμικής επιστήμης, αλλά ταυτόχρονα χρησιμοποιείται ευρέως στην καθημερινή ζωή. Επειδή δεν είναι παράγωγο μέγεθος, η μονάδα μέτρησής του δεν μπορεί να προκύψει από άλλες μονάδες. Απαιτείται συνεπώς ορισμός της μονάδας μέτρησης της θερμοκρασίας. Για τον ορισμό μιας θερμοκρασιακής κλίμακας απαιτούνται δύο προϋποθέσεις:

- Ύπαρξη φαινομένων, τα οποία (υπό καθορισμένες συνθήκες) λαμβάνουν χώρα πάντα υπό την ίδια σταθερή θερμοκρασία (σταθερά σημεία θερμοκρασίας).
- Ύπαρξη ιδιότητας του σώματος που χρησιμοποιείται για τη μέτρηση της θερμοκρασίας, η οποία να είναι μονοσήμαντη συνεχής συνάρτηση της θερμοκρασίας.

Η πρώτη προϋπόθεση πληρούται από πολλά φαινόμενα. Δύο κλασικά παραδείγματα είναι:

α) το μίγμα νερού και πάγου υπό πίεση 760 mmHg σε ισορροπία και

β) το μίγμα νερού και υδρατμού υπό την ίδια πίεση σε ισορροπία.

Στα δύο αυτά σημεία δόθηκαν αυθαίρετα οι τιμές 0 βαθμοί και 100 βαθμοί για την κλίμακα **Celsius**, ενώ για την κλίμακα **Fahrenheit** δόθηκαν αυθαίρετα οι τιμές 32 βαθμοί και 212 βαθμοί αντίστοιχα. Στην πρώτη κλίμακα ο συμβολισμός γίνεται με $^{\circ}\text{C}$, ενώ στη δεύτερη με $^{\circ}\text{F}$.

Η δεύτερη προϋπόθεση ικανοποιείται επίσης από πολλά σώματα και πολλές ιδιότητές τους, μερικές από τις οπο

- Μεταβολή όγκου υγρού.
- Μεταβολή της πίεσης αερίου σταθερού όγκου.
- Μεταβολή του όγκου αερίου σταθερής πίεσης.
- Μεταβολή μήκους μεταλλικής ράβδου.
- Μεταβολή ηλεκτρικής αντίστασης μεταλλικού σύρματος.
- Μεταβολή χρώματος μετάλλου ή άλλου υλικού.
- Μεταβολή ηλεκτρικών ιδιοτήτων ημιαγωγών.
- Ηλεκτρεγερτική δύναμη ζεύγους θερμοηλεκτρικών στοιχείων.

Αν χρησιμοποιήσουμε την πρώτη ιδιότητα και κατασκευάσουμε ένα θερμόμετρο διαστολής υγρού, αντιστοιχούμε την ένδειξη των 0°C στο σημείο όπου έχουμε θερμική ισορροπία με το σύστημα νερού – πάγου και την ένδειξη των 100°C στο σημείο όπου έχουμε θερμική ισορροπία με το σύστημα νερού – υδρατμού. Στη συνέχεια διαιρούμε το μήκος μεταξύ των θέσεων 0 και 100 σε 100 ίσα μέρη. Προφανώς η κλίμακα μήκους που θα προκύψει, εξαρτάται από το υγρό που θα χρησιμοποιήσουμε (συντελεστής διαστολής) και από τη μορφή της διάταξης. Οι κλίμακες θερμοκρασίας που προκύπτουν με τον παραπάνω τρόπο καλούνται εμπειρικές κλίμακες θερμοκρασίας.

Η Θερμοδυναμική Κλίμακα Θερμοκρασιών απαιτεί για τον ορισμό της ένα μόνο σταθερό σημείο θερμοκρασίας. Το σημείο που χρησιμοποιείται είναι το **τριπλό σημείο του ύδατος** (το σημείο ισορροπίας των τριών φάσεων του νερού). Η πολυπλοκότητα της εφαρμογής της οδήγησε στην υιοθέτηση, σε πρακτικό επίπεδο, της **Διεθνούς Κλίμακας Θερμοκρασιών**, η οποία προσεγγίζει τη Θερμοδυναμική Κλίμακα Θερμοκρασιών. Η κλίμακα αυτή χρησιμοποιεί περισσότερα σταθερά σημεία, καθώς επίσης και τη μεταβολή της ηλεκτρικής αντίστασης σύρματος πλατίνας.

Η απόλυτη κλίμακα που σχετίζεται με την κλίμακα Κελσίου είναι η κλίμακα **Kelvin**, η οποία χρησιμοποιεί το σύμβολο K, χωρίς την προσθήκη του συμβόλου του βαθμού. Η σχέση μεταξύ των δύο κλιμάκων είναι

$$K = ^\circ C + 273.15$$

Η απόλυτη κλίμακα που σχετίζεται με την κλίμακα Fahrenheit είναι η κλίμακα **Rankine**, η οποία χρησιμοποιεί το σύμβολο R. Η σχέση μεταξύ των δύο κλιμάκων είναι

$$R = ^\circ F + 459.67$$

3.2 Υγρασία

Υγρασία καλούμε το ποσό του ύδατος που είναι διαλυμένο στην ατμόσφαιρα. Όπως είναι γνωστό, από την θερμότητα που εκλύεται από τον ήλιο ένα μέρος από τις διάφορες υδάτινες επιφάνειες (λίμνες, θάλασσες κ.λπ.) εξατμίζεται και εκλύεται στην ατμόσφαιρα. Έτσι ο ατμοσφαιρικός αέρας αποτελείται από ένα ποσοστό νερού (υδρατμούς). Αυτή η ποσότητα των υδρατμών στον αέρα καλείται υγρασία.

Ο αέρας που περιλαμβάνει την μέγιστη ποσότητα υδρατμών ονομάζεται **κεκορεσμένος (σημείο κόρου)**. Η τάση των υδρατμών στο σημείο κορεσμού ονομάζεται **μέγιστη τάση υδρατμών**. Αποτελεί συνάρτηση της θερμοκρασίας και αυξάνει με αυτήν. Η ικανότητα του αέρα να συγκρατεί υγρασία είναι ανάλογη με την θερμοκρασία. Σε μια συγκεκριμένη θερμοκρασία η ποσότητα των υδρατμών που μπορεί να συγκρατήσει έχει μια μέγιστη τιμή.

Σημείο δρόσου ή Σημείο υγροποίησης ή σημείο κόρου ατμόσφαιρας χαρακτηρίζεται το σημείο εκείνο της θερμοκρασίας που όταν οι υδρατμοί ψυχθούν δημιουργούν το φαινόμενο της δρόσου, δηλαδή τις σταγόνες δρόσου. Στη θερμοκρασία αυτή εξυπακούεται πως ο αέρας είναι κορεσμένος και δεν μπορεί να συγκρατήσει άλλους υδρατμούς με αποτέλεσμα η σχετική υγρασία να είναι 100%. Σημειώνεται όμως ότι η θερμοκρασία κορεσμού της ατμόσφαιρας ή του σημείου δρόσου μπορεί να είναι οποιαδήποτε θερμοκρασία, πάνω από το 0 °C. Η θερμοκρασία αυτή εξαρτάται και μόνο από την ποσότητα των υδρατμών που περιέχει 1 κυβικό μέτρο αέρος, συνεπώς εξαρτάται από την απόλυτη υγρασία.

Απόλυτη υγρασία ονομάζεται ο λόγος της μάζας των υδρατμών προς τον όγκο του αέρα στον οποίο περιέχονται οι υδρατμοί αυτοί. Η απόλυτη υγρασία και η πυκνότητα των υδρατμών θα μπορούσαμε να πούμε ότι είναι έννοιες ταυτόσημες, αφού και στις δύο περιπτώσεις καλύπτουν το σύνολο του όγκου του αέρα που τους προσφέρεται.

Καλούμε **σχετική υγρασία** τον λόγο της μάζας των υδρατμών που περιέχει ο αέρας προς την ποσότητα των υδρατμών που μπορεί να συμπεριλάβει έως ότου κεκορεστεί υπό την ίδια πίεση και θερμοκρασία.

Όπως είναι αντιληπτό ένα ποσοστό 100% αντιστοιχεί σε κεκορεσμένη κατάσταση αέρα ενώ ένα ποσοστό 0% σε αέρα με χωρίς ίχνος υγρασίας. Τα δελτία καιρού συνήθως εκφράζουν την σχετική υγρασία για την ενημέρωση των καιρικών συνθηκών.

Η μέτρηση της υγρασίας γίνεται από ειδικά μετεωρολογικά εργαλεία, όπως τα υγρόμετρα, οι υγραγράφοι και τα ψυχρόμετρα.

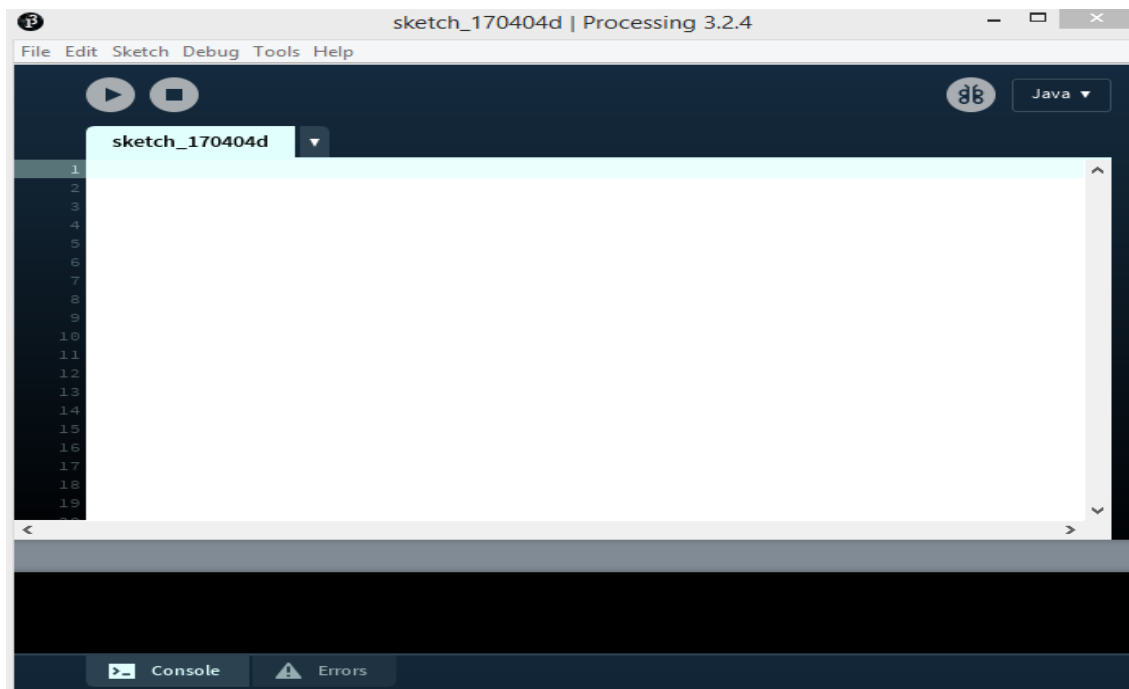
Κεφάλαιο 4 - Processing

4.1 Γενικά

Το Processing είναι μία ανοιχτού κώδικα γλώσσα προγραμματισμού, η οποία χρησιμοποιείται τόσο για επαγγελματική όσο και εκπαιδευτική χρήση για τον προγραμματισμό εικόνων, ήχων και animation (Εικόνα 4.1). Δημιουργήθηκε από δύο απόφοιτους του πανεπιστημίου MIT το 2001 σε Java. Ο προγραμματισμός της γίνεται σε μια απλούστερη μορφή της γλώσσας Java, ενσωματώνοντας μια συλλογή με περισσότερες από 100 βιβλιοθήκες και μια παγκόσμια κοινότητα προγραμματιστών για την υποστήριξη οποιουδήποτε αρχάριου θέλει να πειραματιστεί με την γλώσσα και να δημιουργήσει γραφικό περιβάλλον.

4.2 IDLE

Ο προγραμματισμός σε περιβάλλον Processing αποτελείται από δύο βασικά μέρη, την συνάρτηση `setup()` και τη συνάρτηση `draw()`. Όπως και στο Arduino έτσι και εδώ, η συνάρτηση `setup()` εκτελείται μία φορά στο πρόγραμμα, ενώ η συνάρτηση `draw()` αποτελεί έναν βρόγχο που εκτελείται συνεχώς έως ότου ο χρήστης επιλέξει να σταματήσει το πρόγραμμα. Στο IDLE του processing γράφουμε το πρόγραμμα και στη συνέχεια το εκτελούμε, επιλέγοντας το κουμπί RUN από την γραμμή εργαλείων στην κορυφή της οθόνης. Στη συνέχεια, εάν θέλουμε να σταματήσουμε να τρέχει το πρόγραμμα, επιλέγουμε το stop αντίστοιχα. Σε επόμενο κεφάλαιο, όπου παρουσιάζεται ο προγραμματισμός του processing, θα δούμε αναλυτικότερα κάποιες από τις βασικές εντολές της γλώσσας.



Εικόνα 4.1: Γραφικό περιβάλλον του *Processing*.

Κεφάλαιο 5 – Το ψυγείο

5.1 Γενικά

Το ψυγείο είναι το μέσο το οποίο χρησιμοποιείται από τον άνθρωπο για την αποθήκευση και διατήρηση τροφών. Τα ψυγεία είναι κατασκευές οι οποίες διατηρούν συγκεκριμένες θερμοκρασίες στο εσωτερικό τους, με σκοπό οι διάφορες τροφές να αντέχουν περισσότερες μέρες χωρίς να αλλοιωθούν. Ένα απλό οικιακό ψυγείο αποτελείται από δύο διακριτούς θαλάμους, την συντήρηση και την κατάψυξη. Η συντήρηση είναι ένας θάλαμος ο οποίος λειτουργεί περίπου στους 4 - 5 °C και χρησιμοποιείται για την προσωρινή αποθήκευση (περίπου για μία εβδομάδα) των τροφών. Ο δεύτερος θάλαμος, η κατάψυξη, διατηρεί ένα περιβάλλον αρκετά κάτω από το 0 °C, με σκοπό την ψύξη των προϊόντων με για την περισσότερη δυνατή διατήρησή τους. Ψυγεία χρησιμοποιούνται σε πολλούς τομείς, όπως βιομηχανία, ιατρική, γεωργία κ.λπ., όπου διαφοροποιείται το μέγεθος και οι θερμοκρασίες λειτουργίας. Υπάρχουν ψυγεία συντήρησης ή κατάψυξης που φτάνουν το μέγεθος ολόκληρων δωματίων.

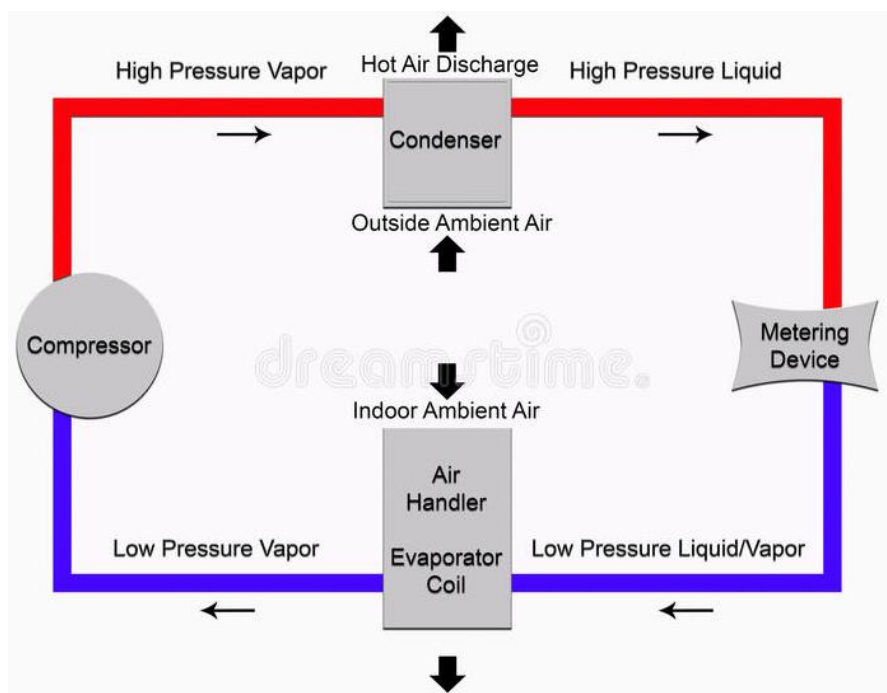
5.2 Ιστορία

Το πρώτο ηλεκτρικό ψυγείο κατασκευάστηκε το 1914 από τον Nathaniel Brackett Wales στο Ντιτρόιτ. Τα πρώτα ψυγεία που κατασκευαστήκαν είχαν δυνατότητα συντήρησης της θερμοκρασίας έως μία ημέρα, αφού λειτουργούσαν με πάγο. Με τον καιρό άρχισαν να εμφανίζονται ηλεκτρικά ψυγεία, που λειτουργούσαν ως αντλίες θερμότητας, τα οποία χρησιμοποιούσαν φρέον ως ψυκτικό μέσο. Τα τελευταία χρόνια όμως το φρέον έχει καταργηθεί λόγω του ότι συμβάλει στην τρύπα του όζοντος και την θέση τους αρχίζουν να παίρνουν τα ψυγεία που λειτουργούν με άλλα οικολογικά ψυκτικά μέσα.

5.3 Αρχή λειτουργίας

Γενικά η ψύξη επιτυγχάνεται μόνο με την εξάτμιση υγρού. Είναι μια δύσκολη διαδικασία και απαιτεί συγκεκριμένη διάταξη για να επιτευχθεί. Σε αυτήν την διαδικασία στηρίζεται και η λειτουργία του ψυγείου, που δεν είναι τίποτα άλλο από μία αντλία θερμότητας. Μέσω του ψυκτικού υγρού που κυκλοφορεί μέσα και έξω από την «καμπίνα» ή «θάλαμο συντήρησης-ψύξης» γίνεται μεταφορά θερμότητας από το εσωτερικό στο εξωτερικό περιβάλλον.

Αναλυτικότερα ο ηλεκτροκίνητος συμπιεστής ψυγείου συμπιέζει το εξαχνωμένο ψυκτικό μέσο, για να το επαναφέρει στην αρχική του κατάσταση ώστε να μπορεί να ξαναχρησιμοποιηθεί για την ψύξη. Λίγο πριν το υγρό εισέλθει στην καμπίνα του ψυγείου υπάρχει μια στραγγαλιστική βαλβίδα εκτόνωσης, η οποία μετατρέπει το υγρό σε αέριο, λόγω απότομης πτώσης της πίεσεως, που οδηγεί και σε μείωση της θερμοκρασίας του αερίου. Έτσι το ατμοποιημένο πλέον μέσο κυκλοφορεί στον εσωτερικό εναλλάκτη θερμότητας (σύστημα σωληνώσεων στο οποίο κυκλοφορεί το ψυκτικό μέσο και γίνεται μεταφορά θερμότητας), μειώνοντας την θερμοκρασία του αέρα στο εσωτερικό της καμπίνας. Στην συνέχεια, αφού το αέριο περάσει και ψύξει τον εσωτερικό εναλλάκτη οδηγείται στον συμπιεστή ο οποίος συμπιέζει το ψυκτικό μέσο και το οδηγεί στον συμπυκνωτή. Λόγω της συμπίεσης το ψυκτικό μέσο αποκτά υψηλή πίεση και θερμοκρασία. Η πίεση διατηρείται υψηλή μετά τον συμπιεστή λόγω της ύπαρξης της στραγγαλιστικής βαλβίδας. Το θερμό ψυκτικό μέσο οδηγείται στον συμπυκνωτή (εναλλάκτη θερμότητας στο εξωτερικό του ψυγείου), στον οποίο αποβάλλει θερμότητα προς το περιβάλλον και υγροποιείται. Το ψυκτικό μέσο στην είσοδο του συμπυκνωτή θα έχει θερμοκρασία της τάξης των 60 °C και αν η θερμοκρασία περιβάλλοντος είναι γύρω στους 30 °C, είναι εύκολο να μειωθεί στους 40 °C η θερμοκρασία του ψυκτικού μέσου αποβάλλοντας θερμότητα προς τον αέρα του περιβάλλοντος. Στην Εικόνα 5.1 παρουσιάζεται ο κύκλος ψύξης που περιγράψαμε παραπάνω.



Εικόνα 5.1: Ψυκτικός κύκλος.

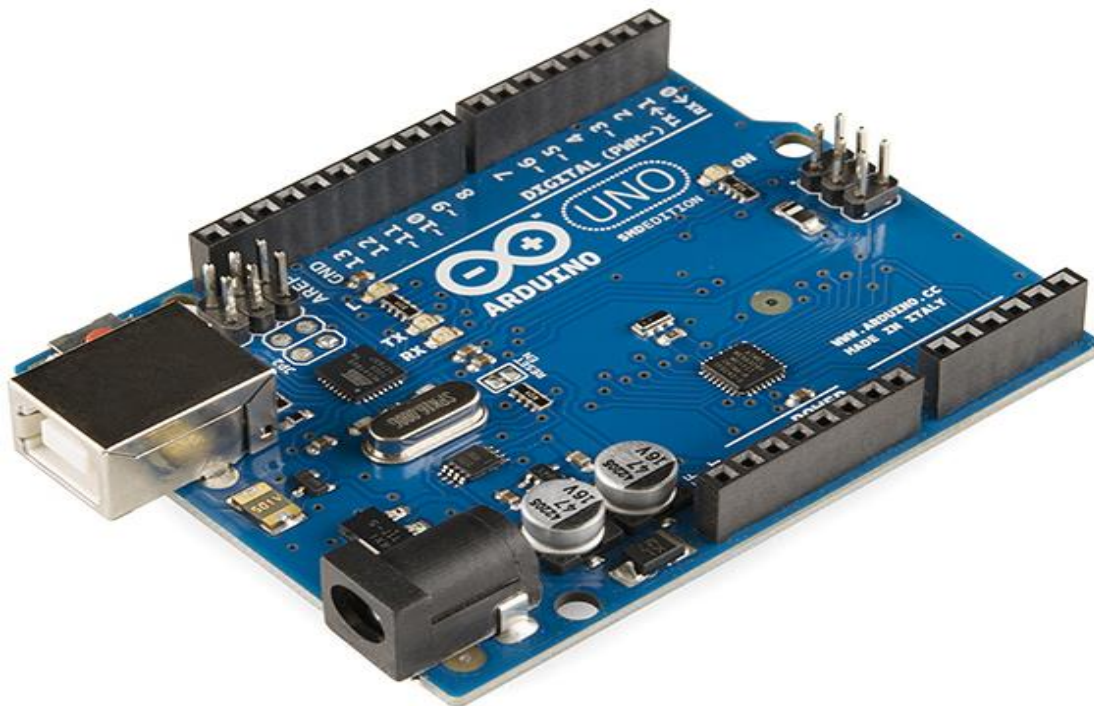
Κεφάλαιο 6 – Κατασκευή της διάταξης

Σε αυτό το κεφάλαιο παρουσιάζονται τα υλικά και τα εξαρτήματα που απαιτούνται για να υλοποιηθεί η παρούσα εργασία, καθώς και η συνδεσμολογία των εξαρτημάτων που εφαρμόστηκε.

6.1 Υλικός εξοπλισμός

Για τη υλοποίηση της κατασκευής θα χρειαστούν τα υλικά και προγράμματα που περιγράφονται στη συνέχεια.

- **Πλακέτα Arduino Uno** (22,50 Ευρώ) (Εικόνα 6.1).



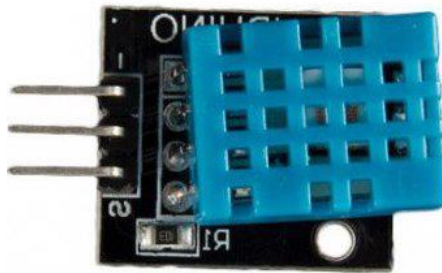
Εικόνα 6.1: Πλακέτα Arduino Uno.

- **Οθόνη LCD** (Εικόνα 6.2), για να μπορούμε να οπτικοποιούμε τη θερμοκρασία και την υγρασία εντός του ψυγείου (Module LCD 1602) (8,20 Ευρώ).



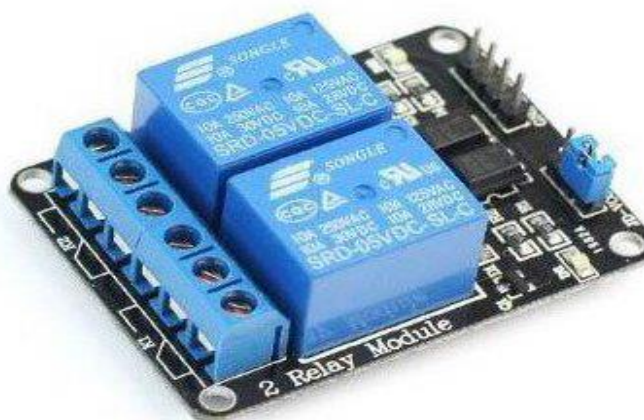
Εικόνα 6.2: Οθόνη LCD.

- Ψηφιακός αισθητήρα θερμοκρασίας-υγρασίας (Εικόνα 6.3) DHT11 (5,90 Ευρώ).



Εικόνα 6.3: Ψηφιακός αισθητήρας θερμοκρασίας-υγρασίας.

- Διπλό ρελέ (Εικόνα 6.4) τροφοδοσίας 5 V και μέγιστης τάσης εξόδου 250 V (6,90 Ευρώ).



Εικόνα 6.4: Διπλό ρελέ.

- Καλώδιο USB 2.0 (Εικόνα 6.5) A/M B/M σύνδεσης συσκευών (2,30 Ευρώ).



Εικόνα 6.5: Καλώδιο USB 2.0.

- **Πλακέτα δοκιμών breadboard** (Εικόνα 6.6) (17,90 Ευρώ).



Εικόνα 6.6: Πλακέτα δοκιμών breadboard.

- **Εγκατάσταση του προγράμματος προγραμματισμού** (Εικόνα 6.7) της πλακέτας Arduino, το οποίο μπορούμε να το βρούμε δωρεάν στην ιστοσελίδα www.arduinoprogramm.gr.



Εικόνα 6.7: Το περιβάλλον διεπαφής που χρησιμοποιείται για τον προγραμματισμό της πλακέτας Arduino.

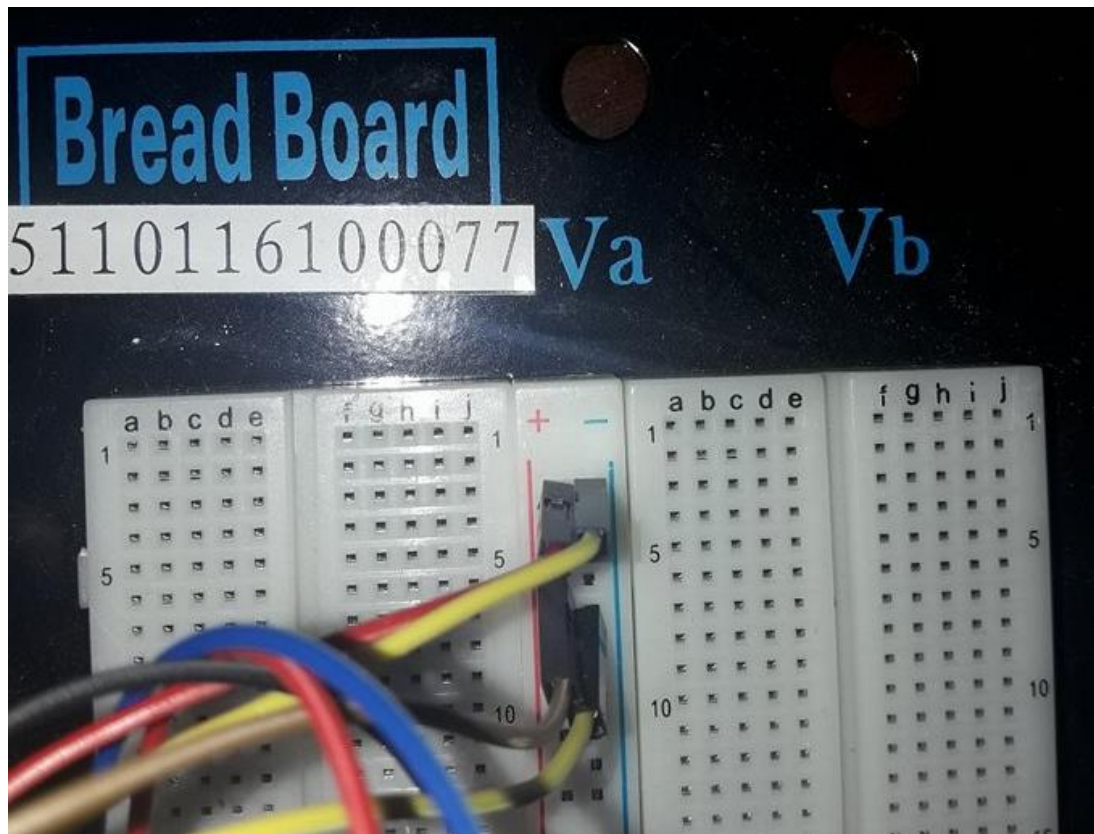
- Εγκατάσταση του προγράμματος της γλώσσας προγραμματισμού Processing από την ιστοσελίδα www.processing.org.

6.2 Συνδεσμολογία

Θα ασχοληθούμε στη συνέχεια με την συνδεσμολογία των παραπάνω εξαρτημάτων, για την ομαλή λειτουργία του project.

6.2.1 Breadboard

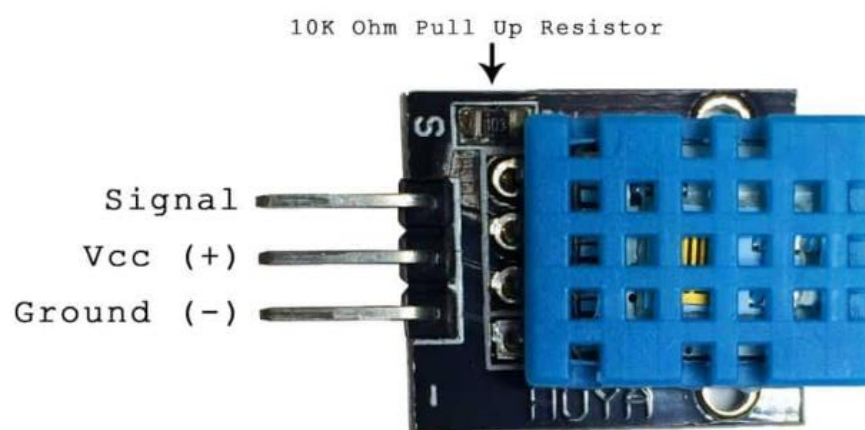
Αρχικά, μέσω του breadboard θα πρέπει να συνδέσουμε τάση και γείωση από την πλακέτα (Εικόνα 6.8). Στο breadboard οι 2 στήλες στο κέντρο (+, -) επικοινωνούν μεταξύ τους και εκεί συνδέουμε την τάση και την γείωση από την πλακέτα, ώστε να παρέχεται η δυνατότητα να τα τροφοδοτούνται όλα τα εξαρτήματα. Επίσης, όλες οι γραμμές που ανήκουν στην ίδια πλευρά (στην αριστερή ή στη δεξιά πλευρά) επικοινωνούν μεταξύ τους [(a-e) και (f-j)] ενώ η μία πλευρά του breadboard (αριστερή) δεν επικοινωνεί με την άλλη (δεξιά).



Εικόνα 6.8: Σύνδεση τάσης και γείωσης στο breadbord.

6.2.2 Αισθητήρας DHT11

Αρχικά συνδέεται ο αισθητήρας DHT11 (Εικόνα 6.9), ο οποίος χρειάζεται μια τάση λειτουργίας 5 V, μία επαφή για να στέλνει τα δεδομένα στην πλακέτα και μια γείωση.



Εικόνα 6.9: Ο αισθητήρας DHT11 με την ονοματολογία των επαφών του.

6.2.3 Οθόνη LCD 1602

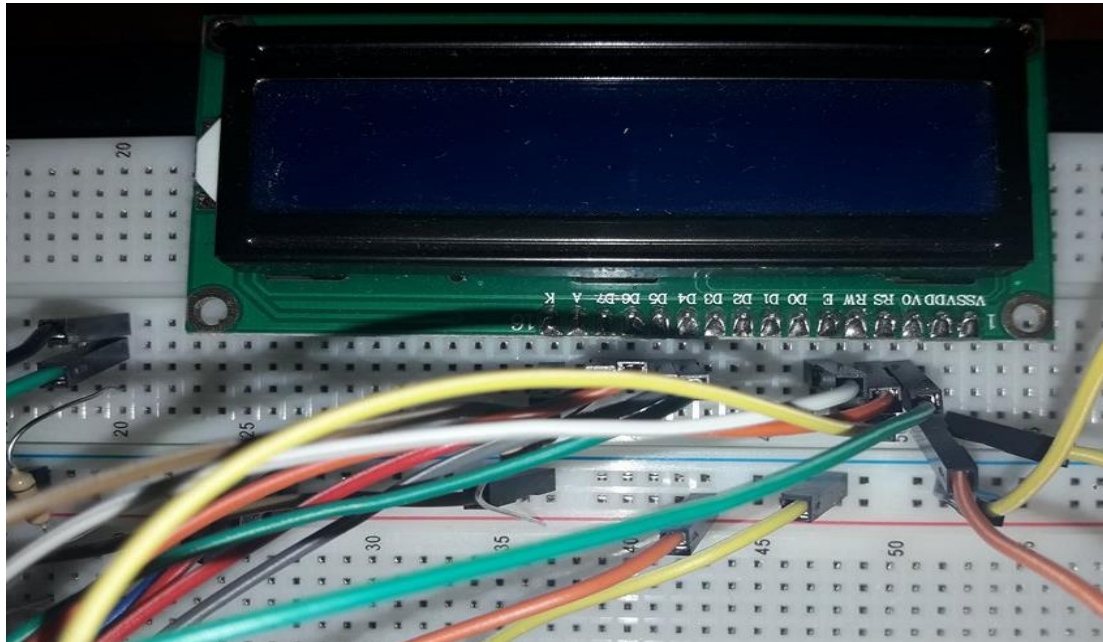
Στην συνέχεια συνδέεται η οθόνη led lcd1602 (Εικόνα 6.10), πάνω στην οποία θα αναγράφονται τα δεδομένα από τον αισθητήρα, δηλαδή η θερμοκρασία και η υγρασία. Αρχικά, ανοίγοντας την συσκευασία, θα δούμε ότι η οθόνη περιέχει μια σειρά από pins, τα οποία θα πρέπει με ειδική κόλληση (καλάι) να τα στερεοποιήσουμε στις 16 θέσεις πάνω στη πλακέτα της οθόνης. Έχοντας τελειώσει με την συγκόλληση των pins, συνδέουμε την οθόνη επάνω στο board, για να μπορέσουμε στην συνέχεια να την συνδέσουμε με την πλακέτα Arduino Uno. Η πρώτη επαφή από τα αριστερά είναι η γείωση (ground), την οποία συνδέουμε με την γείωση της πλακέτας. Αυτό (όπως αναφέραμε και παραπάνω) δεν είναι αναγκαίο να γίνει απευθείας στην πλακέτα, αλλά μπορεί να γίνει από το breadboard, αφού έχουμε τάση 5 V και γείωση στις στήλες με τα +/- αντίστοιχα. Η δεύτερη επαφή VCC συνδέεται στην τάση 5 V του πίνακα μας. Η τρίτη επαφή Vo είναι για να μπορούμε να προσθέτουμε ένα ποτενσιόμετρο, έτσι ώστε να ελέγχουμε την φωτεινότητα της οθόνης. Σε περίπτωση που δεν επιθυμούμε κάτι τέτοιο, για να μπορέσει να λειτουργήσει η οθόνη θα πρέπει να παρεμβάλουμε μια αντίσταση, έτσι ώστε να δημιουργηθεί ένα μικρό κύκλωμα, το οποίο θα αντιπροσωπεύει την ύπαρξη του ποτενσιόμετρου.

Η επόμενη επαφή RS (register select) χρησιμοποιείται για να στέλνουμε δεδομένα ή σχόλια στην οθόνη. Για παράδειγμα, αν την RS επαφή την βάλουμε στην κατάσταση LOW (0 V), τότε στέλνουμε σχόλια (όπως βάλε το κέρσορα σε μια συγκεκριμένη θέση ή καθάρισε την οθόνη από τα υπάρχοντα κ.λπ.). Αν θέσουμε την RS επαφή σε κατάσταση HIGH (5 V) στέλνουμε δεδομένα ή χαρακτήρες στην οθόνη. Αυτές οι ενέργειες πραγματοποιούνται χωρίς να εμπλέκεται άμεσα ο χρήστης, καθώς από τις βιβλιοθήκες που έχουμε κατεβάσει και χρησιμοποιούμε για την οθόνη, εκτελούνται αυτές οι εργασίες όταν εμείς απλά πληκτρολογούμε μια εντολή για τον καθαρισμό ή για να γράψουμε κάτι, όπως θα δούμε σε επόμενο κεφάλαιο. Μπορούμε να συνδέσουμε αυτήν την επαφή σε οποιοδήποτε digital pin επιθυμούμε.

Σειρά τώρα έχει η επαφή R/W (read/write), η οποία χρησιμοποιείται για να επιλέγει αν η οθόνη θα χρησιμοποιηθεί για να διαβάζει ή να γράφει δεδομένα. Είναι προφανές ότι στο παράδειγμά μας χρειαζόμαστε να γράφουμε και να στέλνουμε δεδομένα στην οθόνη.

Η επόμενη επαφή (enable) συνδέεται σε ένα digital pin στην πλακέτα μας και η χρήση της είναι να μπορούμε να γράφουμε στην οθόνη. Στην συνέχεια συνδέουμε και τις επαφές D4, D5, D6, D7 στην πλακέτα μας για να δημιουργήσουμε μια 4 bit συνδεσμολογία. Αν συνδέσουμε και τις 8 επαφές πετυχαίνουμε επικοινωνία 8 bit, το οποίο δεν είναι αναγκαίο, καθώς μπορούμε να γράψουμε σχεδόν τα πάντα με την πρώτη επιλογή.

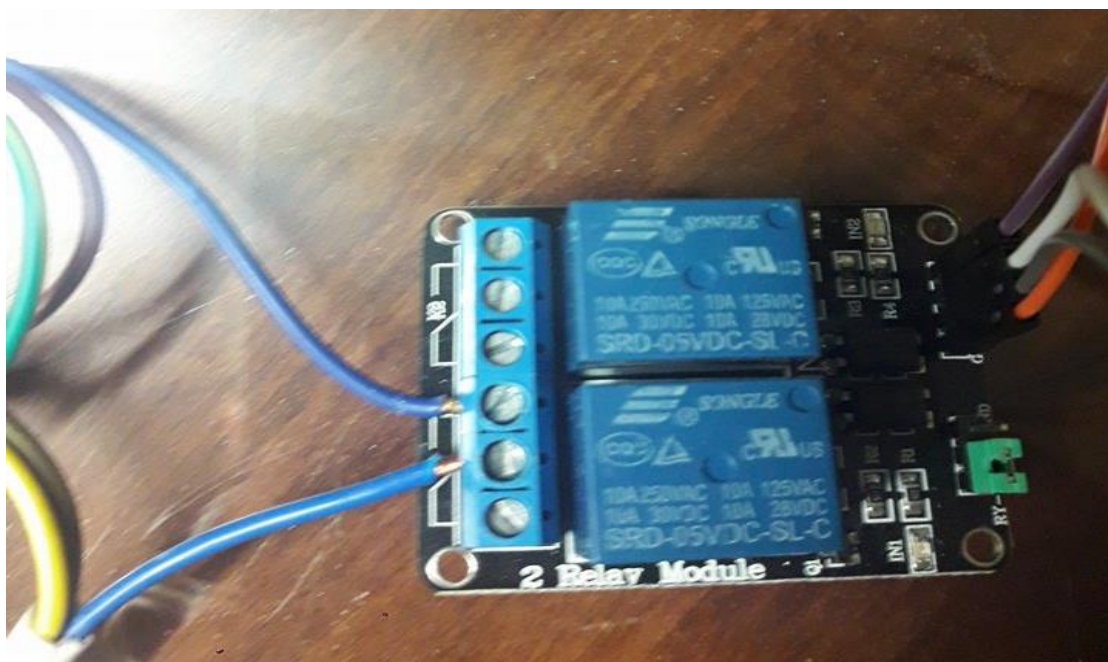
Οι δύο τελευταίες επαφές είναι η άνοδος και η κάθοδος, τα οποία χρησιμεύουν για τον background φωτισμό της οθόνης. Συνδέουμε την άνοδο στην τάση μας (5 V) και την κάθοδο στην γείωση στο breadboard.



Εικόνα 6.10: Σύνδεση της οθόνης LCD.

6.2.4 Το ρελέ

Το ρελέ που χρησιμοποιήσαμε στο πείραμα μας είναι δυνατότητας ελέγχου δύο συσκευών 220 V. Για να συνδέσουμε το ρελέ με τον ηλεκτροκινητήρα μας χρειαζόμαστε μόνο την μία πλευρά του ρελέ. Την μία πλευρά του ρελέ την παρεμβάλουμε στο κύκλωμα του ρεύματος του ηλεκτροκινητήρα. Επιλέγουμε την φάση (καφέ καλώδιο) του ρεύματος του ηλεκτροκινητήρα και το περνάμε μέσα από το ρελέ όπως θα δούμε και στην Εικόνα (6.11). Στην συνέχεια, από την άλλη πλευρά του ρελέ υπάρχουν τέσσερις επαφές, οι οποίες χρησιμεύουν στη λειτουργία και τον έλεγχο του ρελέ από την πλακέτα μας. Συνδέουμε την επαφή VCC με την τάση 5 V που έχουμε βάλει στην θετική στήλη του breadboard. Την άλλη επαφή GND την συνδέουμε στην αρνητική στήλη του breadboard, όπου έχουμε συνδέσει την γείωση από την πλακέτα μας. Οι άλλες δύο επαφές IN1, IN2 αφορούν στην επικοινωνία του ρελέ 1 και του ρελέ 2 αντίστοιχα με το Arduino.



Εικόνα 6.11: Σύνδεση του ρελέ.

Κεφάλαιο 7 - Προγραμματισμός Arduino

7.1 Προγραμματισμός πλακέτας Arduino


Ανοίγοντας κανείς το περιβάλλον προγραμματισμού της πλακέτας Arduino, θα διακρίνει ότι διαχωρίζεται σε δύο βασικά μέρη. Αυτά δεν είναι τίποτε άλλο από 2 συναρτήσεις τύπου void (που αναφέραμε και σε προηγούμενο κεφάλαιο), που σημαίνει ότι δεν επιστρέφουν κάποια μεταβλητή στο πρόγραμμα μας. Η μία συνάρτηση καλείται setup() και η άλλη loop().

Στην συνάρτηση setup() ο προγραμματιστής προγραμματίζει οτιδήποτε θέλει να εκτελεστεί μία φορά από την πλακέτα, όπως αρχικοποίηση κάποιας μεταβλητής, ή η συμπερίληψη κάποιας βιβλιοθήκης που απαιτείται, όπως θα δούμε και παρακάτω.

Στην συνέχεια, η συνάρτηση loop() είναι ο βασικός βρόγχος του προγράμματος μας, δηλαδή, ότι προγραμματίσουμε εκεί μέσα η πλακέτα το εκτελεί συνεχώς όσο τροφοδοτείται με ρεύμα.

7.2 Βιβλιοθήκες και δηλώσεις μεταβλητών

Όσον αναφορά τον κώδικα για την υλοποίηση της εν λόγω εργασίας, φαίνεται στη συνέχεια η αρχή του.



```

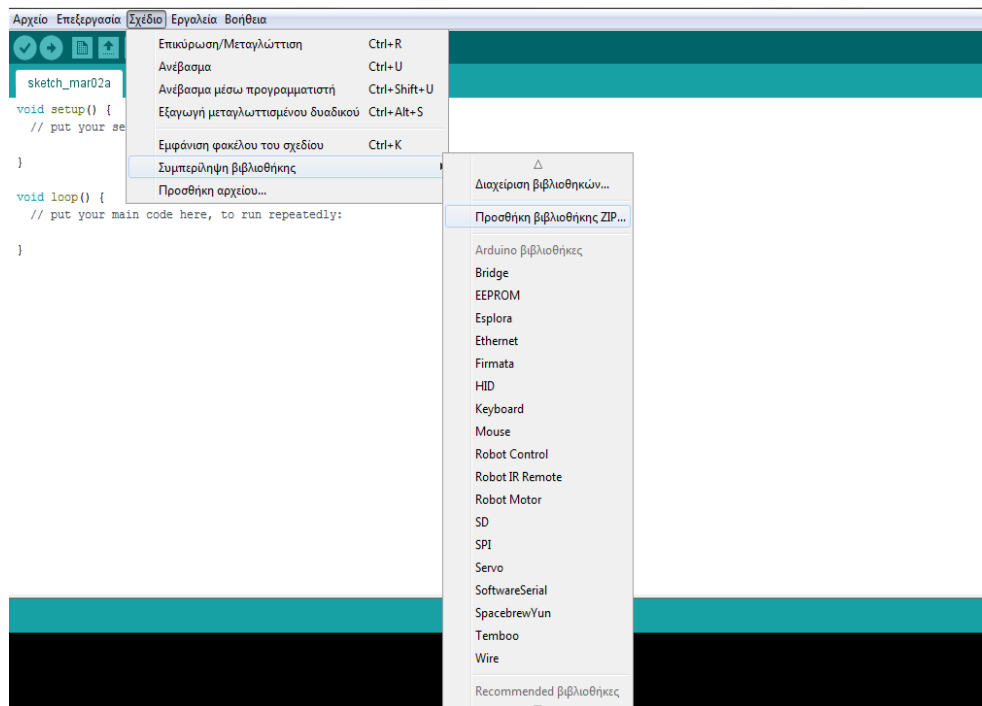
#include <LiquidCrystal.h>
#include <dht.h>
#define C1 0
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int greenPin = 7;
dht sensor;

//summerilamvano vivliothiki pou xreiazete o aisthitiras DHT11 gia na leitourgisei
// Arxikopoio stin A0 tin epikoinonia me to RELE 1
//dilono ta pin pou exo sundesei me tin leitourgia tis othonis me to arduino
//dilono to input tou aisthitira apo ekei tha pairno ta dedomena
  
```

Ξεκινώντας τον κώδικα μέσω της εντολής #include, πρέπει να συμπεριλάβουμε τις βιβλιοθήκες LiquidCrystal.h και dht.h, οι οποίες χρειάζονται για τη λειτουργία της οθόνης lcd και του αισθητήρα, αντίστοιχα. Πριν από αυτό όμως θα πρέπει να κατεβάσουμε τις δύο βιβλιοθήκες από το επίσημο site του Arduino ([liquidCrystal](#) [DHT11LIB](#)). Αφού κατεβάσουμε της 2 βιβλιοθήκες, εκτελούμε τις ενέργειες που περιγράφονται στη συνέχεια.

Πατάμε σχέδιο - συμπερίληψη βιβλιοθήκης - προσθήκη βιβλιοθήκης zip, για να συμπεριλάβουμε τη βιβλιοθήκη μας στο κατάλογο με τις βιβλιοθήκες, οι οποίες είναι γνωστές για το Arduino. Στην συνέχεια, αφού βρούμε που τις έχουμε αποθηκεύσει και

έχοντας συμπεριλάβει στις γνωστές βιβλιοθήκες, μπορούμε είτε μέσω της εντολής `include`, η πάλι με τον ίδιο τρόπο (σχέδιο συμπερίληψη βιβλιοθήκης και επιλογή της βιβλιοθήκης από την λίστα βιβλιοθηκών) να την εντάξουμε στο sketch μας (όπως φαίνεται στην ακόλουθη Εικόνα).



Η επόμενη εντολή (`#define C1 0`) χρησιμοποιείται για να αρχικοποιήσουμε την επικοινωνία του ρελέ με την A0 θύρα της πλακέτας, που όπως θα δούμε και αργότερα, μέσω αυτής της μεταβλητής γίνεται ο έλεγχος του μοτέρ μας, δίνοντας απλά τον χαρακτηρισμό `high` ή `low` στην μεταβλητή `C1`.

Με την εντολή

```
liquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

δηλώνουμε τα pin που έχουμε δεσμεύσει από το Arduino για την λειτουργία της οθόνης. Η εντολή

```
int greenPin = 7;
```

δεσμεύει την θύρα 7 ως input του αισθητήρα. Από αυτήν την θύρα θα λαμβάνουμε τις μετρήσεις του αισθητήρα και θα τις επεξεργαζόμαστε με συγκεκριμένες συναρτήσεις, όπως θα δούμε παρακάτω, για να εξάγουμε το αποτέλεσμα (θερμοκρασία). Όπως είναι γνωστό στον προγραμματισμό όλα τα δεδομένα από τους αισθητήρες έρχονται σε μορφή σημάτων, τα οποία πρέπει μετά να επεξεργαστούμε για να τους δώσουμε κάποια κατανοητή μορφή

για τον άνθρωπο. Στο παράδειγμα μας η τιμή που στέλνει ο αισθητήρας στην πλακέτα μας δεν είναι τίποτα άλλο από μια τιμή ρεύματος, την οποία εμείς με τις κατάλληλες συναρτήσεις την μετατρέπουμε σε έναν κατανοητό για εμάς αριθμό, ο οποίος αντιπροσωπεύει την θερμοκρασία του χώρου στον οποίο βρίσκεται ο αισθητήρας (εσωτερικό ψυγείου).

Συνεχίζοντας τον κώδικα, για να λειτουργήσει ο αισθητήρας και οι βιβλιοθήκες που τρέχουμε στο πρόγραμμα μας, θα πρέπει να δηλώσουμε την μεταβλητή, στην οποία θα αποθηκεύονται και θα επεξεργάζονται οι μετρήσεις μας. Έτσι, στην βιβλιοθήκη `dht` ορίζουμε την μεταβλητή `sensor`. Όπως θα δούμε και στην συνέχεια, με αυτήν την μεταβλητή θα εργαζόμαστε για την εμφάνιση και επεξεργασία των αποτελεσμάτων μας.

7.3 Συνάρτηση `void setup()`

Όπως αναφέρθηκε και παραπάνω, οι εντολές στην συνάρτηση `setup` εκτελούνται μία φορά κατά την εκτέλεση του κώδικα. Σε αυτό το σημείο του προγράμματος θα πρέπει να είμαστε ιδιαίτερα προσεκτικοί, διότι οι εντολές δεν αποτελούν τον κορμό της λειτουργίας του προγράμματος, όμως είναι απαραίτητες για την ομαλή λειτουργία του. Στην συνέχεια παρουσιάζονται οι εντολές που χρησιμοποιήθηκαν για το συγκεκριμένο project.

Αρχικά, με την εντολή

```
pinMode(C1,OUTPUT)
```

δηλώνουμε ότι η μεταβλητή `C1` θα χρησιμοποιηθεί ως εντολή εξόδου από το πρόγραμμα μας και το Arduino. Πιο πάνω είχαμε αρχικοποιήσει την μεταβλητή `C1` στην θέση `A0` του Arduino. Τη θέση `A0` την έχουμε συνδέσει με το ρελέ της κατασκευής μας. Έχοντας ορίσει έτσι την μεταβλητή `C1` ως μεταβλητή εξόδου, θα μπορούμε, όπως θα δούμε και παρακάτω, να δίνουμε ρεύμα και να ανοίγουμε ή να κλείνουμε το ρελέ μας και κατά συνέπεια να ξεκινάμε ή να σταματάμε το μοτέρ της κατασκευής μας (220 V).

Συνεχίζοντας, ενεργοποιούμε την οθόνη μας με την εντολή

```
lcd.begin(16,2).
```

Με την εντολή

```
lcd.setCursor(5,0)
```

επιλέγουμε το σημείο της οθόνης στο οποίο θέλουμε να τυπώσουμε μια φράση ή έναν αριθμό. Η οθόνη αποτελείται από 2 γραμμές και 16 στήλες. Έτσι, αν εμείς θέλουμε να τυπώσουμε κάτι στην αρχή θα εκτελέσουμε διαδοχικά

```
lcd.setCursor(1,0)
```

```
lcd.print("ergasia");
```

Αυτή η σειρά εντολών ξεκινάει να τυπώσει στην πρώτη στήλη στην πρώτη γραμμή την λέξη (ergasia). Αν επιθυμούσαμε να τυπώσουμε κάτι στη δεύτερη γραμμή ξεκινώντας από την πρώτη στήλη, τότε στην εντολή `lcd.setCursor` θα δηλώναμε

`lcd.setCursor(0,1).`

Η εντολή

`lcd.clear()`

χρησιμοποιείται για τον καθαρισμό της οθόνης από όλες τις υπάρχουσες τυπωμένες φράσεις ή αριθμούς έως εκείνη την στιγμή.

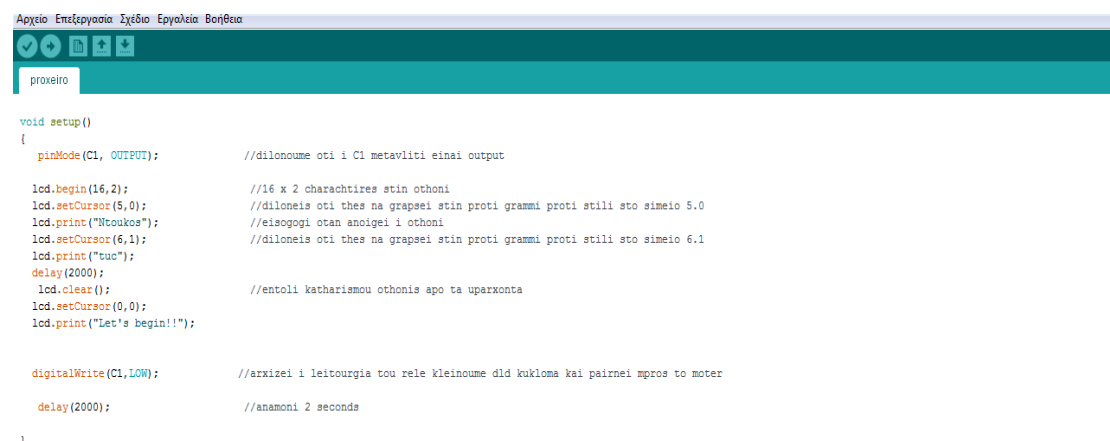
Τελειώνοντας, με την συνάρτηση `setup` εκτελούμε την εντολή

`digitalWrite(C1,LOW)`

για να κλείσουμε το ρελέ και να ξεκινήσει η λειτουργία του μοτέρ του ψυγείου μας. Γενικά η εντολή `digitalWrite()` χρησιμοποιείται για να δίνει σε μία μεταβλητή (δηλαδή στην θέση της πλακέτας που έχουμε αντιστοιχίσει την μεταβλητή μας κάθε φορά) τάση ή όχι, με τις εκφράσεις `HIGH` ή `LOW` αντίστοιχα.

Πολλές φορές επίσης σε sketches χρησιμοποιούμε την εντολή `delay()`. Αυτή η εντολή είναι μια χρονοκαθυστέρηση την οποία επιθυμούμε στο πρόγραμμα μας για κάποιο λόγο (π.χ. θέλουμε να παίρνουμε μετρήσεις από τον αισθητήρα κάθε 5 δευτερόλεπτα). Ως input δέχεται μια τιμή η οποία είναι σε ms. Οπότε, αν εμείς επιθυμούμε μια καθυστέρηση της τάξης των 2 δευτερολέπτων για κάποιο λόγο, θα εκτελέσουμε την εντολή

`delay(2000).`



```

void setup()
{
  pinMode(C1, OUTPUT);           //dilonoume oti i C1 metavliti einai output

  lcd.begin(16,2);               //16 x 2 charactires stin othoni
  lcd.setCursor(5,0);            //diloneis oti thes na grapsei stin proti grammi proti stili sto simio 5.0
  lcd.print("Ntoukos");          //eisogogi otan anoigei i othoni
  lcd.setCursor(6,1);            //diloneis oti thes na grapsei stin proti grammi proti stili sto simio 6.1
  lcd.print("tuc");
  delay(2000);
  lcd.clear();                   //entoli katharismou othonis apo ta uparkonta
  lcd.setCursor(0,0);
  lcd.print("Let's begin!!");

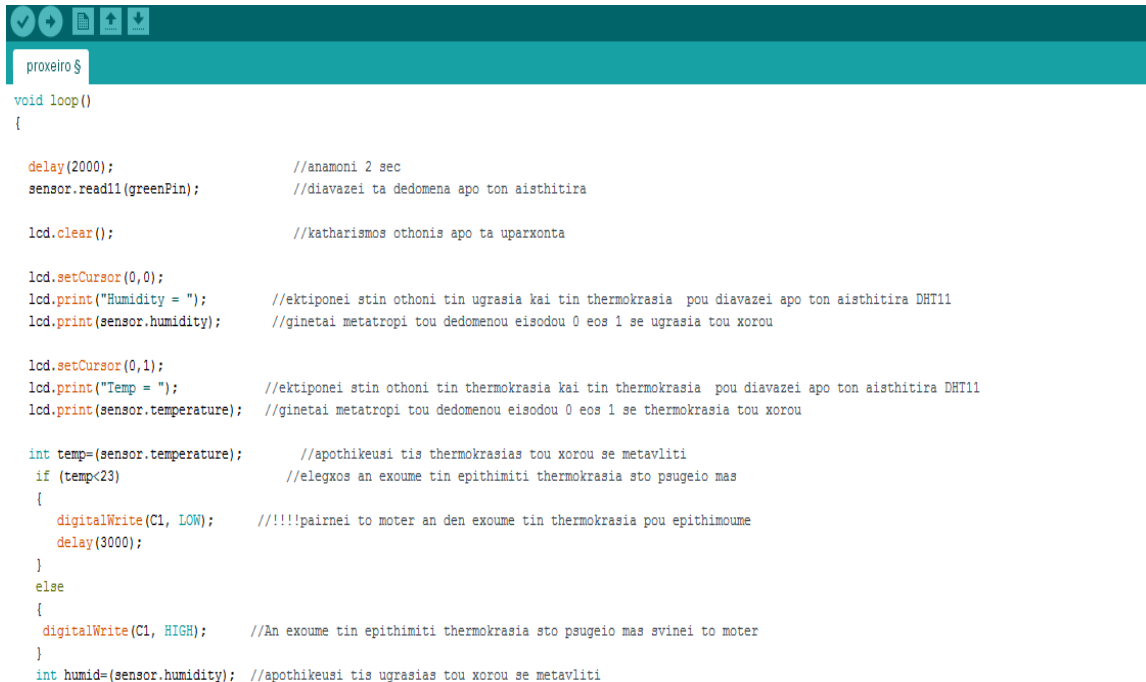
  digitalWrite(C1,LOW);          //arkizei i leitourgia tou rele kleinoume did kukloma kai pairnei mpros to moter

  delay(2000);                   //anamoni 2 seconds
}
  
```

7.4 Συνάρτηση void loop()

Στη συνέχεια θα αναλύσουμε την κατασκευή της συνάρτησης `void loop()` του προγράμματος μας. Η συνάρτηση `loop` σε ένα sketch αποτελεί τον βασικό εκτελεστικό κορμό. Μέσα σε αυτήν την συνάρτηση επαναλαμβάνονται όλες οι εντολές, όσο το Arduino

έχει τροφοδοσία ρεύματος. Όσον αναφορά το δικό μας πρόγραμμα, μέσα στη συνάρτηση loop παίρνουμε συνεχώς μετρήσεις από τον αισθητήρα και ανάλογα με τα αποτελέσματα των μετρήσεων ενεργοποιούμε ή απενεργοποιούμε το μοτέρ του ψυγείου. Γίνεται δηλαδή ένας διαρκής έλεγχος της θερμοκρασίας του ψυγείου και έλεγχος του μοτέρ του.



```

void loop()
{
    delay(2000);           //ανάμονη 2 sec
    sensor.read11(greenPin); //διαβάζει τα δεδομένα από τον αισθητήρα

    lcd.clear();           //katharismos othonis apo ta uparxonta

    lcd.setCursor(0,0);
    lcd.print("Humidity = "); //ektiponei stin othoni tin ugrasia kai tin thermokrasia pou diavazei apo ton αισθητήρα DHT11
    lcd.print(sensor.humidity); //ginetai metatropi tou dedomenou eisodou 0 eos 1 se ugrasia tou xorou

    lcd.setCursor(0,1);
    lcd.print("Temp = "); //ektiponei stin othoni tin thermokrasia kai tin thermokrasia pou diavazei apo ton αισθητήρα DHT11
    lcd.print(sensor.temperature); //ginetai metatropi tou dedomenou eisodou 0 eos 1 se thermokrasia tou xorou

    int temp=(sensor.temperature); //apothikeusi tis thermokrasias tou xorou se metavlititi
    if (temp>23) //elegxos an exoume tin epithimiti thermokrasia sto psugeio mas
    {
        digitalWrite(C1, LOW); //!!!!pairnei to moter an den exoume tin thermokrasia pou epithimoume
        delay(3000);
    }
    else
    {
        digitalWrite(C1, HIGH); //An exoume tin epithimiti thermokrasia sto psugeio mas svinei to moter
    }
    int humid=(sensor.humidity); //apothikeusi tis ugrasias tou xorou se metavlititi

```

Ξεκινάμε διαβάζοντας δεδομένα από τον αισθητήρα με την εντολή

`sensor.read11(greenPin);`.

Αυτή η εντολή προέρχεται από την βιβλιοθήκη του αισθητήρα DHT 11. Στην μεταβλητή `greenPin` έχουμε αντιστοιχίσει την θέση 7 του Arduino, η οποία συνδέεται με τον αισθητήρα μας. Στην μεταβλητή `sensor`, που αρχικοποιήσαμε στην αρχή, αποθηκεύεται το δεδομένο από τον αισθητήρα. Στην συνέχεια, αφού έχουμε πάρει το δεδομένο το οποίο όπως αναφέραμε είναι ένας αριθμός μη κατανοητός για εμάς, τον μετατρέπουμε σε μια κατανοητή μορφή και το εκτυπώνουμε στην οθόνη μας με την εντολή

`lcd.print(sensor.humidity);`

και

`lcd.print(sensor.temperature);`

για την υγρασία και την θερμοκρασία αντίστοιχα. Αυτές οι δύο συναρτήσεις (`sensor.humidity`, `sensor.temperature`) μετατροπής των δεδομένων τις χρησιμοποιούμε έτοιμες από τη βιβλιοθήκη του αισθητήρα. Κάνουν δηλαδή την μετατροπή χωρίς να χρειαστεί εμείς να κάνουμε μια σειρά από πράξεις και την κλιμάκωση σε βαθμούς Κελσίου.

Στην συνέχεια, αποθηκεύουμε τις δύο μεταβλητές, ώστε να μπορέσουμε να τις χρησιμοποιήσουμε για την κατασκευή των διαγραμμάτων αργότερα, καθώς επίσης και να κάνουμε τον έλεγχο για την λειτουργία του μοτέρ.

Σημειώνεται ότι το διάγραμμα της θερμοκρασίας ή της υγρασίας θα γίνεται με το processing, το οποίο απαιτεί την εκτύπωση στην σειριακή ενός και μόνο αριθμού χωρίς γράμματα και επεξηγήσεις. Οπότε εκτυπώνουμε με την εντολή

```
Serial.println (temp);
```

την θερμοκρασία στο μόνιτορ για την επεξεργασία και την κατασκευή του διαγράμματος θερμοκρασίας - χρόνου από το processing, όπως θα δούμε αναλυτικότερα στο επόμενο κεφάλαιο.

Τελειώνοντας, κάνουμε κάποιους ελέγχους σχετικά με την θερμοκρασία και αναλόγως ενεργοποιούμε το μοτέρ μας με την ήδη γνωστή εντολή digitalWrite(). Με έναν βρόγχο ελέγχου if, εξετάζουμε αν η θερμοκρασία του ψυγείου είναι κάτω από το όριο που επιθυμούμε (πχ 5 °C). Αν η θερμοκρασία είναι στο επιθυμητό επίπεδο, τότε απενεργοποιούμε το μοτέρ μας, εάν όχι ενεργοποιούμε το μοτέρ ώστε η θερμοκρασία να ξεκινήσει να μειώνεται.

Κεφάλαιο 8 - Προγραμματισμός Processing

Με το processing θα επιτύχουμε μια real time γραφική απεικόνιση της θερμοκρασίας του ψυγείου σε συνάρτηση με τον χρόνο.

8.1 Αρχικοποιήσεις-Δηλώσεις

Όπως σε κάθε πρόγραμμα έτσι και εδώ θα πρέπει να αρχικοποιήσουμε κάποιες μεταβλητές και να εισάγουμε κάποιες βιβλιοθήκες. Ξεκινάμε με την εντολή

```
import processing.serial.*;
```

για την εισαγωγή της βιβλιοθήκης, ώστε να ξεκινήσουμε την επικοινωνία με τη σειριακή θύρα, στην οποία αποστέλλονται τα δεδομένα της θερμοκρασίας από το Arduino στον υπολογιστή, όπως αναφέραμε σε προηγούμενο κεφάλαιο. Αρχικοποιούμε στη Serial την μεταβλητή myPort. Η Serial απαιτεί την αρχικοποίηση μιας μεταβλητής, η οποία αναφέρεται στην θύρα επικοινωνίας usb με την πλακέτα.

Συνεχίζουμε αρχικοποιώντας τις ακόλουθες μεταβλητές:

```
int xpos (θέση στον x άξονα),
```

```
int oldxpos=0 (παλιά θέση στον x άξονα),
```

```
int oldxtint (παλιά τιμή στο γράφημα),
```

int dx=10 (το κενό που αφήνουμε από το παράθυρο έως ο σημείο όπου ξεκινάει η γραφική απεικόνιση).

Όλες αυτές οι μεταβλητές έχουν σκοπό τη δημιουργία του γραφήματος, δημιουργώντας ένα ευθύγραμμο τμήμα από το παλιό σημείο απεικόνισης στο νέο.

```

1
2 import processing.serial.*;
3 Serial myPort;
4
5 int xpos; //trexousa thesi sto x a3ona
6 int oldxpos=0; //palia thesi ston x a3ona. to programma ennonei tin grammi me ton proigoumeno kai to neo simeio
7 int oldxtint=0; //apothikeusi tis palias timis pou apeikonizetai sto grafima
8 int dx=10; // apostasi apo to aristero kai to de3i perithorio tou parathirou grafikon
9

```

8.2 Συνάρτηση void setup()

Όπως αναφέρθηκε και προηγουμένως, η συνάρτηση setup αποτελεί ένα από τα δύο βασικά συστατικά του προγράμματος. Όλες οι εντολές εκτελούνται μία και μόνο φορά κατά την εκτέλεση του προγράμματος.

Κατά την εκτέλεση της συνάρτησης `setup()`, αυτό που κάνουμε είναι να δημιουργούμε το παράθυρο στο οποίο θα δημιουργούνται τα γραφήματα (real time), καθώς επίσης εγκαθιστούμε επικοινωνία με την θύρα `usb`, στην οποία είναι συνδεδεμένο το Arduino και αποστέλλονται τα δεδομένα από τον αισθητήρα. Αναλυτικότερα, ξεκινάμε με την εντολή

```
size(500,250)
```

με την οποία καθορίζουμε το μέγεθος του παραθύρου σε `pixel`, το οποίο θα ανοίγει για την απεικόνιση των γραφημάτων. Στην συνέχεια καλούμε τη συνάρτηση

```
void init_graph()
```

για το σχεδιασμό των αξόνων και την προσθήκη ετικετών στους άξονες. Έπειτα ακολουθεί η εντολή

```
xpos=dx
```

με την οποία αρχικοποιούμε την αρχή του γραφήματος στην αρχή του παραθύρου. Τελευταία αλλά πολύ σημαντική εντολή έρχεται η

```
myPort=new Serial(this,"COM3",9600)
```

η οποία ενεργοποιεί την επικοινωνία με τη θύρα `usb 3`, στην οποία έχουμε συνδέσει την πλακέτα μας. Ο αριθμός `9600` αναφέρεται στον ρυθμό επικοινωνίας που έχουμε επιλέξει να λειτουργεί ο υπολογιστής με το Arduino μέσα από το `IDLE` του προγραμματισμού της πλακέτας.

```
10 void setup()
11 {
12   size (500,250); //megethos parathirou
13   init_graph(); //klisi tis sinartisis gia to sxediasmo a3onon etiketon ktl
14   xpos=dx; //arxikopoioyme apo tin arxi tou parathirou (dx)
15   myPort=new Serial(this,"COM3",9600); //epikoīnonia me tin com3 pou einai sundedemenο to arduino mas
16
17 }
```

8.3 Συνάρτηση `void init_graph()`

Μέσα στη συνάρτηση `init_graph` έχουμε τις ακόλουθες εντολές. Η εντολή

```
Background (255,255,255)
```

καθορίζει το χρώμα του φόντου του παραθύρου με τις τιμές `0,0,0` να αντιστοιχούν στο μαύρο χρώμα, ενώ οι τιμές `255,255,255` αντιστοιχούν σε λευκό χρώμα. Οι επόμενες δύο εντολές,

```
line (0,230,500,230)
```

και

```
Line (10,250,10,0)
```

δημιουργούν τις δύο κάθετες γραμμές του καρτεσιανού συστήματος συντεταγμένων. Στην εντολή `line` δηλώνουμε ότι επιθυμούμε τη σχεδίαση μιας γραμμής από το σημείο $(x1,y1)$ στο σημείο $(x2,y2)$, η εντολή συντάσσεται δηλαδή ως `line(x1,y1,x2,y2)`. Ακολουθούν οι εντολές

```
    textSize (20)
```

και

```
    fill (0)
```

οι οποίες αναφέρονται στο μέγεθος της γραμματοσειράς, καθώς και στο χρώμα της, με το 0 να αντιστοιχεί στο μαύρο χρώμα. Τέλος, ακολουθούν οι εντολές

```
    text ("[Y]-θερμοκρασία",11,22)
```

και

```
    text ("[X]-ΧΡΟΝΟΣ",390,230)
```

οι οποίες τοποθετούν τις ετικέτες στους άξονες. Η εντολή `text` συντάσσεται τοποθετώντας μέσα στα άγκιστρα τη λέξη που επιθυμούμε και στη συνέχεια τις συντεταγμένες του σημείου στο χώρο, στο οποίο θέλουμε να τοποθετηθεί η λέξη αυτή.

```

62 void init_graph()           //sunartisi arxikopoiisis parathrou sxediasmou
63 {
64     background(255,255,255); //xroma fontou(leuko) oi times 0,0,0 antistoixoun sto mauro kai oi times 255,255,255 sto leuko
65     line(0,230,500,230);
66     line(10,250,10,0);       //sxediasi a3onon
67     textSize(20);            //megethos grammatoseiras
68     fill (0);                //xroma grammatoseiras 0=mauro
69
70     text ("[Y]- θερμοκρασια",11,20); //ETIKETES STOX X,Y A3ONA
71     text ("[X]- ΧΡΟΝΟΣ",390,230);
72
73 }
74
75
```

8.4 Συνάρτηση `void draw()`

Η συνάρτηση `draw()` αποτελεί τον βασικό βρόγχο του προγράμματός μας. Αυτή η συνάρτηση επαναλαμβάνεται επ' άπειρο, έως ότου τερματιστεί το πρόγραμμα. Ξεκινώντας, αρχικοποιούμε 2 μεταβλητές, οι οποίες χρησιμεύουν στο να αποθηκεύουμε τις τιμές της θερμοκρασίας, ώστε να τις σχεδιάζουμε κάθε φορά, όπως θα δούμε στις επόμενες γραμμές κώδικα. Στη συνέχεια δηλώνουμε ως `string` την αλφαριθμητική μεταβλητή `inString` και με την εντολή

```
    myPort.readStringUntil('n')
```

διαβάζουμε τα δεδομένα που έχουν σταλεί στην θύρα `usb COM3`, την οποία αρχικοποιήσαμε στη συνάρτηση `setup`, από την πλακέτα `Arduino` και τον αισθητήρα μέσα από το ψυγείο. Στη συνέχεια, με την εντολή

```
    if (inString != null)
```

ελέγχουμε αν έχει σταλθεί κάποιο δεδομένο στην θύρα για να συνεχίσουμε τη διαδικασία.

Αν μετά τον έλεγχο δούμε ότι έχει αποσταλεί κάποιο δεδομένο στη θύρα, τότε δηλώνουμε ως αλφαριθμητικό πινάκα num και με την εντολή

```
SplitTokens (inString, ".")
```

διαχωρίζουμε το αλφαριθμητικό στο σημείο όπου υπάρχει «.» και αποθηκεύουμε το πρώτο μέρος πριν την τελεία στο num[1] και το δεύτερο μέρος μετά την τελεία στο num[2]. Στη συνέχεια στην μεταβλητή n κρατάμε μόνο το ακέραιο μέρος που μας ενδιαφέρει (num[0]), πληκτρολογώντας την εντολή

```
n= Integer.parseInt(num[0])
```

και την εκτυπώνουμε στο μόνιτορ με την εντολή

```
println.
```

Για να είναι ευκρινέστερα τα σημεία στο γράφημα, βάζουμε έναν μικρό πολλαπλασιαστή (π.χ. 3) στην τιμή της θερμοκρασίας και αποθηκεύουμε τον αριθμό που προκύπτει στην μεταβλητή ixt, το οποίο θα είναι και το σημείο απεικόνισης.

Προχωράμε στη σχεδίαση αφού έχουμε φτιάξει όλα τα δεδομένα όπως παραπάνω. Καθορίζουμε το χρώμα σχεδίασης (άσπρο) με την εντολή

```
stroke(0,0,0)
```

και προχωράμε με την εντολή

```
line(oldxpos,150-oldxtint,xpos,150-ixt).
```

Με την εντολή αυτή ζητάμε από το πρόγραμμα να δημιουργήσει μια γραμμή από το παλιό σημείο (oldxpos,150-oldxtint) στο καινούριο (xpos,150-ixt). Το 150 χρησιμοποιείται για την προσαρμογή στις συντεταγμένες της οθόνης. Όσον αφορά στον γ-άξονα, είναι ανάλογο του μεγέθους το παραθύρου απεικόνισης που χρησιμοποιούμε κάθε φορά. Για την καλύτερη κατανόηση του γραφήματος εκτυπώνουμε πάνω στη γραμμή απεικόνισης την θερμοκρασία κάθε φορά που η τιμή της αλλάζει. Κάνουμε δηλαδή έναν έλεγχο για τον αν έχουμε αλλαγή στη τιμή της θερμοκρασίας με τον βρόγχο ελέγχου

```
if(xpos!=ixt)
```

και αν βγει θετικό τυπώνουμε τη θερμοκρασία στο σημείο που βρισκόμαστε την παρούσα χρονική στιγμή (xpos,ixt) με την εντολή

```
text(n,xpos,ixt).
```

Συνεχίζουμε αποθηκεύοντας τις καινούριες τιμές στις μεταβλητές oldxpos=xpos και oldxtint=ixt. Τέλος, έχουμε δημιουργήσει ένα βρόγχο ελέγχου

```
if(xpos>=width-dx)
```

με τον οποίο ελέγχουμε αν έχουμε ξεπεράσει το δεξί όριο του παραθύρου. Αν έχουμε υπερβεί αυτό το όριο, τότε αρχικοποιούμε πάλι τις μεταβλητές $xpos=dx$ και $oldxpos=0$ και ξανακαλούμε τη συνάρτηση

`init_graph()`

για τη δημιουργία του παραθύρου. Αν δεν έχουμε υπερβεί το δεξί όριο του παραθύρου προχωράμε στον υπολογισμό του επόμενου σημείο στον άξονα x με την εντολή

$xpos+=10$.

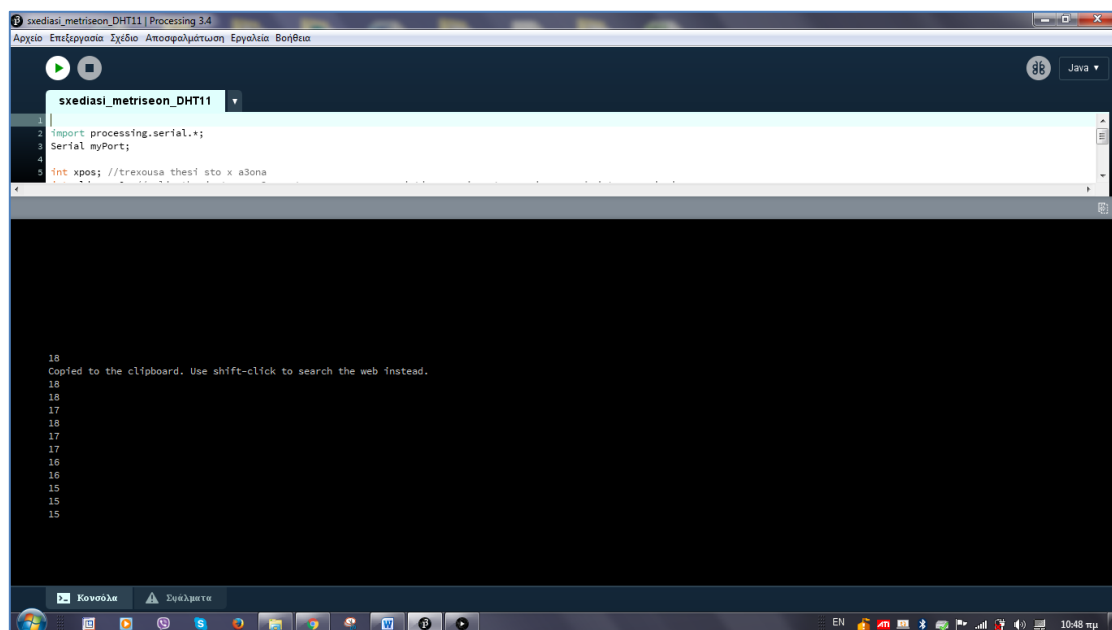
```
19 void draw()
20 {
21     delay (3000);
22     int ixt=0,n=0;
23     String inString=myPort.readStringUntil('\n'); //anagnosi mias grammis termatismenis me to \n kai apothikeusi stin metavliti inString
24     if (inString!=null) //elegxos gia to an exei apostalei kapoio dedomeno stin com3
25     {
26         String[] num=splitTokens(inString,"."); //me tin sunartisi splitTokens mporoume na diaxorisoume ena alfarithmitiko me vasi to xaraktira pou dinoume.
27         //stin periptosi mas to diaxORIZOUME mr ". ". Apothikeuetai diladi to akeraio meros tis thermokrasias sto num[1] kai to metavlito sto num[2]
28
29         n=Integer.parseInt(num[0]); //kratame to proto tmima pou mas endiaferi (akeraio)
30         ixt=(n*3); //bazoume ena mikro pollaplasiastei gia megalutero platos stin apeikonisi
31         println(n);
32
33
34
35         stroke (0,0,0); //kathorismos xromatos sxediasis
36         line(oldxpos,150-oldxtint,xpos,150-ixt); //sxediasmos mias grammis pou na enonei to palio me to neo simeio.
37         //To 150 xrisimopoietai gia tin prosarmogi stis suntetagmenes tin othonis sxetika me ton y a3ona
38         //(x1,y1,x2,y2) (x1,y1) proigoumeno simeio (x2,y2)epomeno simeio
39
40
41         if (oldxtint!=ixt)
42         {
43             text (n,xpos,ixt); //an uparxei diafora thermokrasias tin tuponei sti grammi tou graffimatos
44             textSize(10);
45         }
46         oldxpos=xpos; //apothikeusi trexousas timis
47
48
49         oldxtint=ixt; //apothikeusi trexousas timis
50
51         if (xpos>=width-dx) //an exoume 3eperasei to de3i orio tou parathirou arxikopoioume pali to aristero orio
52         {
53             xpos=dx;
54             oldxpos=0; //arxikopoioume pali tin metavliti tis proigoumenis timis oste na min ginei enosei me to teleutaio simeio tou x
55             init_graph();
56         }
57         else {xpos+=10;} //upologismos epomenou simeio ston a3ona x
58     }
59 }
```

Κεφάλαιο 9 – Συμπεράσματα

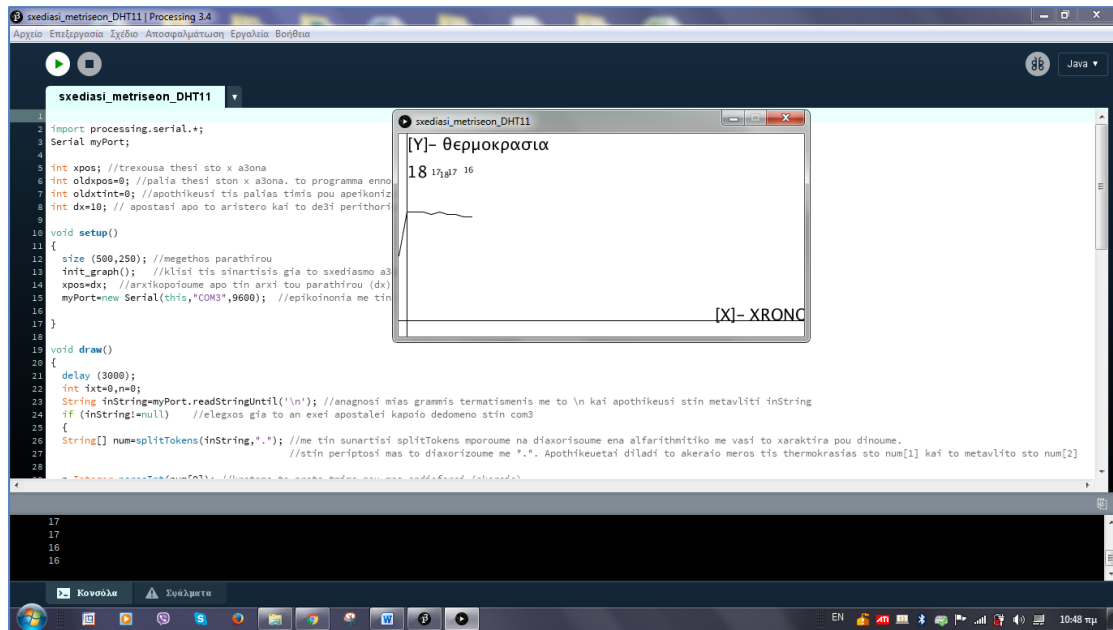
9.1 Λειτουργία του ψυγείου

Ο μικροελεγκτής, με την βοήθεια του αισθητήρα θερμοκρασίας και του ρελέ, είναι πλέον υπεύθυνος για την ομαλή λειτουργία του ψυγείου. Καθορίζει πότε θα ξεκινήσει το ψυγείο (ο ηλεκτροκινητήρας που κινεί τον συμπιεστή) και πότε θα σταματήσει, λαμβάνοντας υπόψη την θερμοκρασία λειτουργίας, την οποία έχουμε προσδιορίσει κατά τον προγραμματισμό της πλακέτας. Έχει αντικαταστήσει τον κοινό μηχανικό θερμοστάτη, ο οποίος ξεκινούσε και σταματούσε το μοτέρ πριν τη συγκεκριμένη τροποποίηση, μην μπορώντας να προσδιοριστεί μια συγκεκριμένη ακριβής θερμοκρασία λειτουργίας του ψυγείου. Επιπλέον, η πλακέτα λειτουργεί και σαν καταγραφικό, αφού μας δίνει πληροφορίες για το πώς μεταβάλλεται η θερμοκρασία στο χρόνο. Μέσω του προγράμματος Processing βλέπουμε πως μεταβάλλεται η θερμοκρασία, αφού όλες οι τιμές της παρουσιάζονται στην οθόνη του υπολογιστή.

Στην Εικόνα 9.1 βλέπουμε την αποτύπωση των θερμοκρασιών στην οθόνη του υπολογιστή μας μέσα από το πρόγραμμα processing. Στην Εικόνα 9.2 απεικονίζεται μαζί με το μόνιτορ και ένα άλλο παράθυρο, στο οποίο γίνεται γραφική απεικόνιση της θερμοκρασίας με τον χρόνο. Κάθε μεταβολή της θερμοκρασίας παρουσιάζεται τόσο σαν μια απλή τιμή στο μόνιτορ όσο και γραφικά στο διάγραμμα θερμοκρασίας-χρόνου.



Εικόνα 9.1: Απεικόνιση θερμοκρασιών στην κονσόλα.

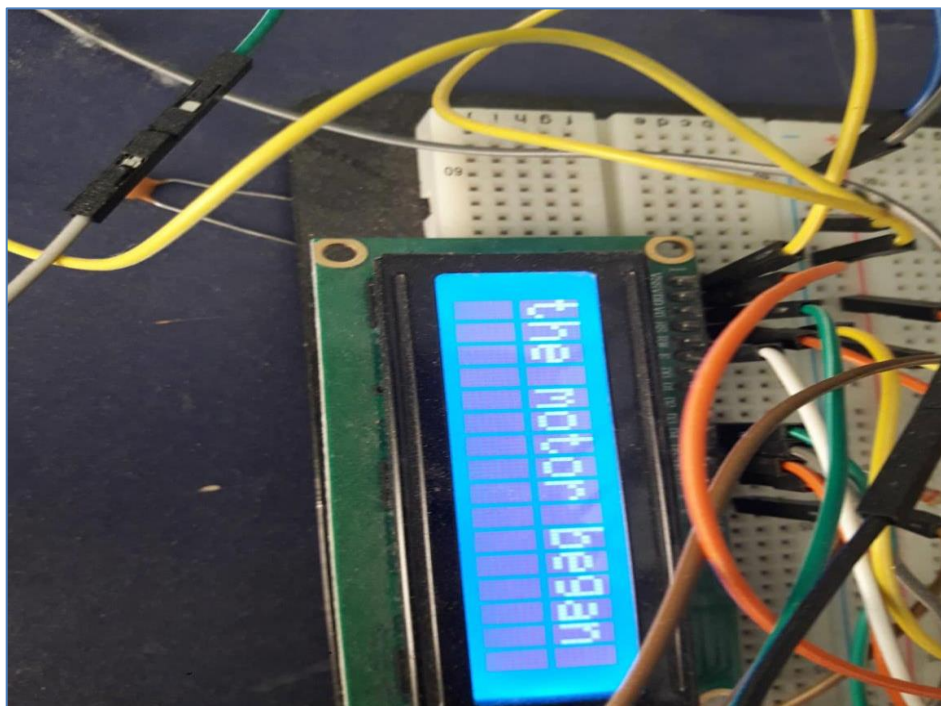


Εικόνα 9.2: Διάγραμμα θερμοκρασίας με το χρόνο.

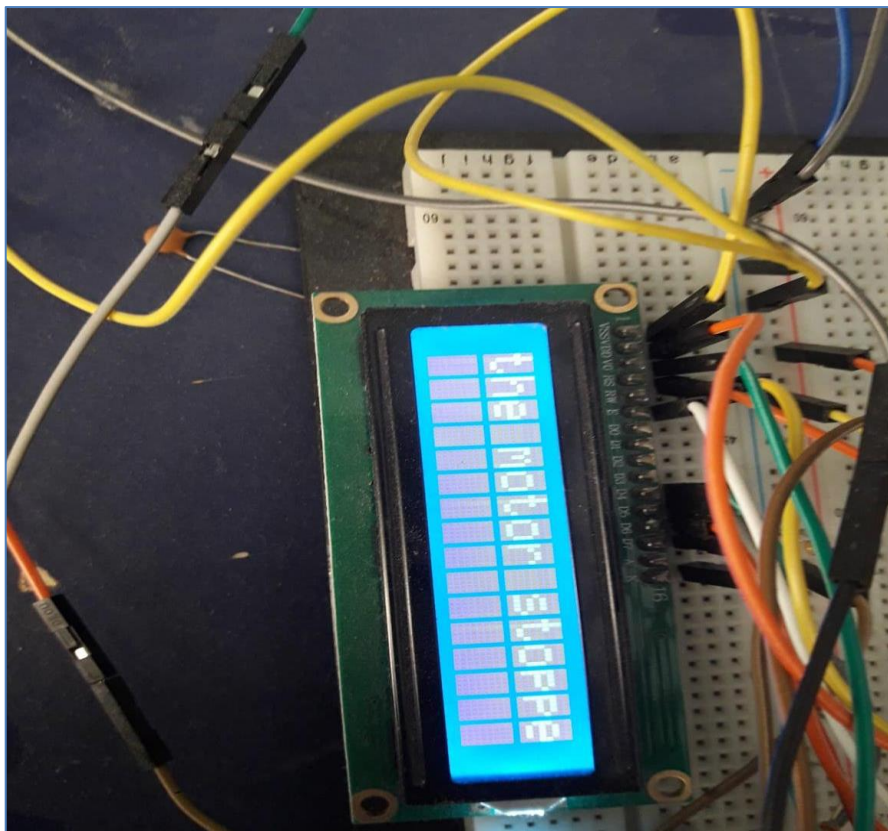
Εκτός όμως από την παρακολούθηση της κατάστασης του ψυγείου μέσω της οθόνης του υπολογιστή, μπορούμε να παρατηρούμε όλες τις απαραίτητες πληροφορίες μέσω της οθόνης lcd, που έχουμε εγκαταστήσει πάνω στο ψυγείο. Από την οθόνη του ψυγείου παίρνουμε πληροφορίες σχετικά με την θερμοκρασία και την υγρασία μέσα στο ψυγείο (Εικόνα 9.3) καθώς επίσης και πότε ο ηλεκτροκινητήρας βρίσκεται σε λειτουργία (Εικόνα 9.4) ή έχει σταματήσει (Εικόνα 9.5), όταν έχουμε επιτύχει την θερμοκρασία που επιθυμούμε μέσα στην καμπίνα του ψυγείου.



Εικόνα 9.3: Lcd οθόνη απεικόνισης θερμοκρασίας-υγρασίας στο ψυγείο.



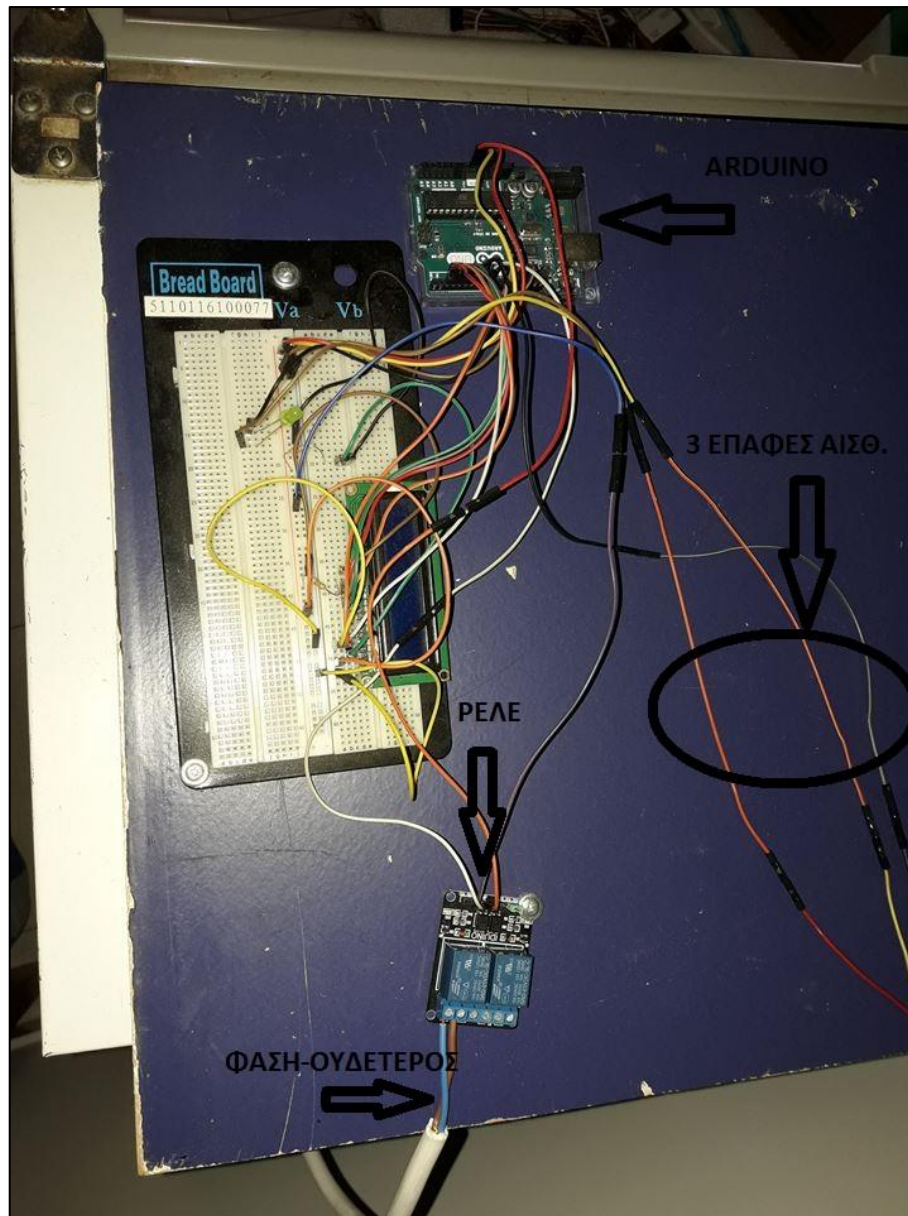
Εικόνα 9.4: Η lcd οθόνη όταν ο ηλεκτροκινητήρας ξεκινάει τη λειτουργία του.



Εικόνα 9.5: Η lcd οθόνη όταν ο ηλεκτροκινητήρας σταματάει τη λειτουργία του.

Μπορούμε δηλαδή να θέσουμε σε λειτουργία την συσκευή ακόμα και χωρίς την χρήση κάποιου Η/Υ, απλώς συνδέοντας με ρεύμα το ψυγείο και την πλακέτα Arduino. Υπάρχει ένας ειδικός μετασχηματιστής, ο οποίος μετασχηματίζει το ρεύμα 220Volt σε 12Volt, με τον οποίο μπορούμε να τροφοδοτήσουμε την πλακέτα. Έτσι η συσκευή δουλεύει κανονικά, με τη διαφορά ότι δεν συγκρατούνται οι μεταβολές τις θερμοκρασίας με το χρόνο για περαιτέρω ανάλυση.

Στην Εικόνα 9.6 βλέπουμε την τελική ολοκληρωμένη μορφή της κατασκευής μας. Η φωτογραφία απεικονίζει το επάνω μέρος του ψυγείου, το οποίο επιλέξαμε (χωρίς να είναι υποχρεωτικό) για να τοποθετήσουμε όλα τα απαραίτητα εξαρτήματα. Ένα σημαντικό σημείο αναφοράς είναι ο τρόπος σύνδεσης του ρελέ. Το ρελέ, για να μπορούμε να ορίζουμε πότε θα έχει ρεύμα το μοτέρ του ψυγείου, πρέπει να τοποθετηθεί «ανάμεσα» στην φάση και τον ουδέτερο του ψυγείου. Πιο αναλυτικά, παρεμβάλλοντας το ρελέ ανάμεσα στην φάση και τον ουδέτερο, μπορούμε (με την κατάλληλη εντολή) να αποφασίζουμε πότε θα κλείνει το κύκλωμα ρεύματος μοτέρ - πρίζα (δηλ. το μοτέρ θα λειτουργεί) και πότε θα ανοίγει (δηλ. το μοτέρ θα βρίσκεται εκτός λειτουργίας).



Εικόνα 9.6: Πανοραμική εικόνα της κατασκευής.

Στο εσωτερικό του ψυγείου βρίσκεται ο αισθητήρας θερμοκρασίας-υγρασίας (Εικόνα 9.7), ο οποίος έχει αντικαταστήσει τον μηχανικό θερμοστάτη που βρισκόταν εκεί.

Ο μηχανικός θερμοστάτης (Εικόνα 9.8), έχοντας την άκρη του μεταλλικού του ελάσματος μέσα στην καμπίνα, ελέγχει το πότε θα κλείσει το ηλεκτρικό κύκλωμα και πότε θα το ανοίξει. Στο εσωτερικό του μεταλλικού ελάσματος κυκλοφορεί ένα ψυκτικό υγρό, το οποίο ανάλογα την θερμοκρασία μεταβάλλει τον όγκο του. Έτσι, ο θερμοστάτης (ο οποίος έχει και την ιδιότητα του ρελέ), ανάλογα με την μεταβολή του όγκου του υγρού επιτρέπει την διέλευση ρεύματος, δηλαδή την λειτουργία του ηλεκτροκινητήρα.

Όπως είναι προφανές, ο παραδοσιακός θερμοστάτης δεν είναι τόσο ακριβής στον προσδιορισμό της θερμοκρασίας όσο ο αισθητήρας μας, καθώς χρειάζεται κάποια ώρα ώστε να εκκινήσει ο ηλεκτροκινητήρας. Με τον μηχανικό θερμοστάτη η θερμοκρασία του ψυγείου έχει ανεβεί παραπάνω από το επιθυμητό όριο μέχρι ο ηλεκτροκινητήρας να ενεργοποιηθεί. Σε αντίθεση με την δική μας περίπτωση, όπου μπορούμε να ορίσουμε ακριβές σημείο όπου το μοτέρ θα σταματήσει ή θα ξεκινήσει την λειτουργία του.



Εικόνα 9.7: Ο αισθητήρας στο εσωτερικό του ψυγείου.



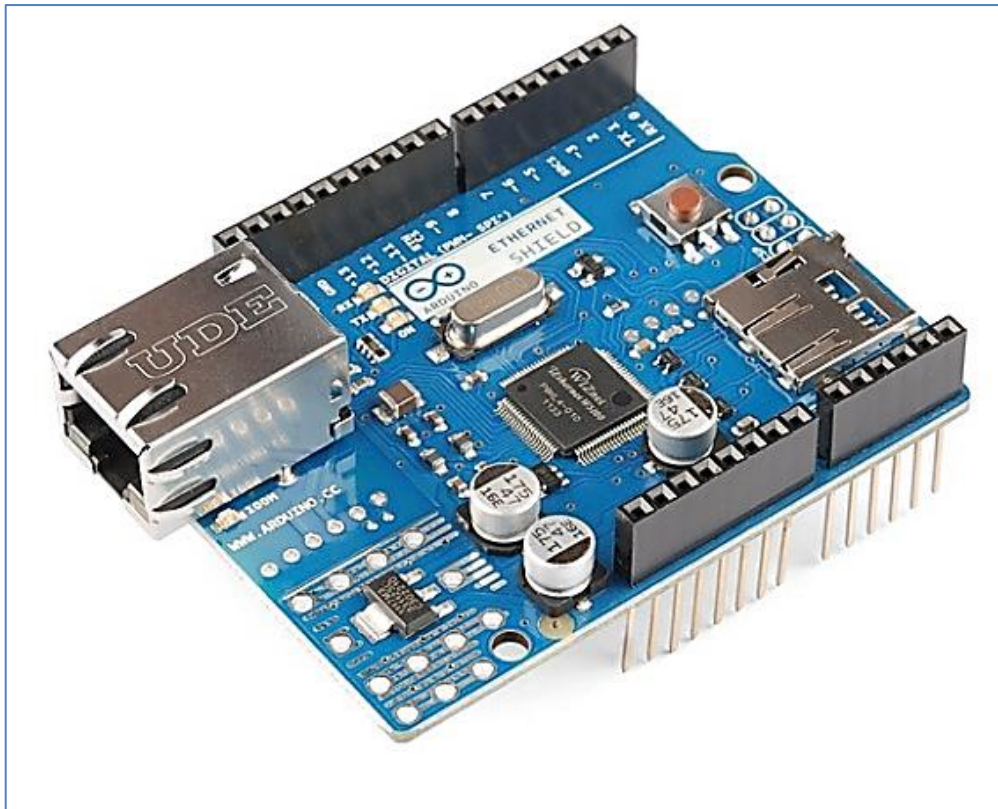
Εικόνα 9.8: Μηχανικός θερμοστάτης.

9.2 Τα θετικά χαρακτηριστικά της προτεινόμενης διάταξης

Από την παρούσα διπλωματική εργασία προέκυψαν διάφορα θετικά συμπεράσματα, τα οποία θα αναλύσουμε στη συνέχεια.

9.2.1 Επεκτάσιμη διάταξη

Κύριο χαρακτηριστικό της προτεινόμενης διάταξης είναι ότι έχει ένα μεγάλο εύρος επέκτασης της χρήσης της. Πιο συγκεκριμένα, θα μπορούσε με τις κατάλληλες προσθήκες (τόσο στο κώδικα όσο και με ορισμένα επιμέρους εξαρτήματα) η πλακέτα να στέλνει τα δεδομένα σε κάποια σελίδα στο internet, ώστε να μπορεί ο ενδιαφερόμενος να λαμβάνει πληροφορίες για την συσκευή που επιθυμεί εξ αποστάσεως. Αυτό μπορεί να γίνει με ένα internet shield (Εικόνα 9.9) όπως ονομάζεται, το οποίο προσαρμόζεται πάνω στην πλακέτα και μας δίνει την δυνατότητα η πλακέτα μας να συνδέεται στο internet. Βέβαια μετά χρειάζεται μια σειρά από προγραμματιστικές ενέργειες για τη δημιουργία ιστοσελίδας, όπου θα αποστέλλονται τα δεδομένα. Μπορεί να προχωρήσει ακόμα περισσότερο, αναπτύσσοντας μια εφαρμογή για κινητά τηλέφωνα, όπου θα μπορεί κάποιος να λαμβάνει πληροφορίες για την συσκευή που επιθυμεί ή ακόμα και να ελέγχει το αν η συσκευή δουλεύει ή όχι.



Εικόνα 9.9: Internet shield.

9.2.2 Προσαρμοστικότητα

Το επόμενο θετικό χαρακτηριστικό της κατασκευής είναι ότι μας δίνει την δυνατότητα εγκατάστασης της και σε άλλες ηλεκτρικές συσκευές. Είναι μια ηλεκτρονική διάταξη, η οποία μπορεί να μας δώσει πληροφορίες σχετικά με την θερμοκρασία και την υγρασία οποιουδήποτε μηχανήματος ή χώρου. Έχοντας πρόσβαση στο λογισμικό της πλακέτας, μπορούμε με μικρές αλλαγές και ελέγχουμε συσκευές των οποίων η λειτουργία καθορίζεται από την θερμοκρασία του περιβάλλοντος.

Έχουμε κατασκευάσει έτσι ένα εργαλείο σχετικά φτηνό, το οποίο μπορεί να μας βοηθήσει τόσο σε επίπεδο παραγωγής νέων προϊόντων, όσο και σε επίπεδο καθημερινότητας απλών ανθρώπων.

Βιβλιογραφία

Βλαγοΐδης, Ν., 2016, «Σύστημα Βασισμένο σε Arduino για την παρατήρηση ροής υγρού», Διπλωματική Εργασία, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Σχολή Θετικών Επιστημών, Τμήμα Πληροφορικής.

Γουργιώτης, Κ., 2016, «Εφαρμογή Arduino στην γεωργία», Διπλωματική Εργασία, Παν. Δυτ. Μακεδονίας, Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών.

Κατσάνος, Χ., 2016, «Σχεδιασμός και υλοποίηση απομακρυσμένου συστήματος ελέγχου ηλεκτρικής ισχύος», Διπλωματική Εργασία, Παν. Δυτ. Μακεδονίας, Τμήμα Μηχανικών Πληροφορικής και τηλεπικοινωνιών.

Κορέτσης, Γ., 2013, «Σχεδιασμός και κατασκευή τρίκυκλου ποδήλατου με ηλεκτρική υποβοήθηση», ΤΕΙ Θεσσαλονίκης, Τμήμα Οχημάτων.

Μάκρα, Αικ., 2016, «Υλοποίηση μικρού αυτόνομου ρομποτικού οχήματος με μικροελεγκτή Arduino», Διπλωματική Εργασία, ΤΕΙ Κεντ. Μακεδονίας, Τμήμα Μηχανικών Πληροφορικής Τ.Ε.

Μπούλης, Σ., 2014, «Έλεγχος και καταγραφή συνθηκών με online ενημέρωση στο διαδίκτυο μέσω Arduino», Διπλωματική Εργασία, ΤΕΙ Μακεδονίας και Θράκης, Σχολή Τεχνολογικών Εφαρμογών, Τμήμα Ηλεκτρολόγων Μηχανικών Τ.Ε.

Πετράκης, Ζ., 2014, «Ανάπτυξη εφαρμογής τηλεχειριζόμενου οχήματος με πλατφόρμα Arduino», Διπλωματική Εργασία, ΤΕΙ Κρήτης, Τμήμα Μηχανικών Πληροφορικής.

Φωτεινογιαννόπουλος, Α., 2016, «Ανάπτυξη υλικού και λογισμικού για συστήματα ασφαλείας και απομακρυσμένου ελέγχου κατοικίας σε περιβάλλον Arduino», Διπλωματική Εργασία, Παν. Πειραιά, Τμήμα Ψηφιακών Συστημάτων.

Vladyslav, L., 2017, «Intelligent Bed», Bachelor Project, Czech University in Prague, Faculty of Electrical Engineering.

Yuhua, Y., 2015, "Wirelessly Connected Sensor Acquisition System for Remote Nursing Applications", Bachelor Project, Kennesaw State University, Faculty of Electrical and Computer Engineering.