

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΟΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ 1507**

# **ΕΛΕΓΧΟΣ ΗΛΕΚΤΡΟΝΟΜΟΥ ΜΕ ΧΡΗΣΗ ΜΙΚΡΟΕΛΕΓΧΤΗ ARDUINO**

**ΣΠΟΥΔΑΣΤΗΣ:  
ΓΕΩΡΓΙΟΣ ΝΤΑΛΛΗΣ (5914)**

**ΕΙΣΗΓΗΤΗΣ:  
ΕΥΑΓΓΕΛΟΣ ΜΙΜΟΣ**

**ΠΑΤΡΑ 2015**



## Πρόλογος / Περίληψη

Η παρούσα πτυχιακή εργασία ασχολείται με τον έλεγχο ηλεκτρονόμου, που τροφοδοτεί μονοφασικό φορτίο, μέσω της πλατφόρμας Arduino με σκοπό την βελτίωση της προστασίας των ενεργητικών ή των παθητικών φορτίων του ηλεκτρονόμου σε περιπτώσεις υπερφόρτισης .

Στο πρώτο κεφάλαιο περιγράφονται αναλυτικά οι ηλεκτρονόμοι ως κομμάτι του κλασικού αυτοματισμού. Δίνεται η αρχή λειτουργίας τους, τα είδη τους με σκοπό την ανάδειξη των μειονεκτημάτων τους ως στοιχεία του κλασικού αυτοματισμού.

Στο δεύτερο κεφάλαιο περιγράφονται αναλυτικά τα είδη των Arduino. Γίνεται παρουσίαση του προγράμματος Arduino μέσω του οποίου προγραμματίστηκε ο Arduino. Τέλος δίνονται οι κανόνες με τους οποίους γράφονται τα προγράμματά Arduino.

Στο τρίτο κεφάλαιο εξηγείται ο λόγος επιλογής του το Arduino που χρησιμοποιήθηκε, και παρουσιάζονται τα περιφερειακά στοιχεία που χρησιμοποιήθηκαν για την επίτευξη της πειραματικής διάταξης.

Στο τέταρτο κεφάλαιο αρχικά δίνεται το ηλεκτρολογικό σχέδιο του κυκλώματος ισχύος και του βοηθητικού κυκλώματος του ηλεκτρονόμου με την πλατφόρμα Arduino. Στην συνέχεια δίδεται το διάγραμμα ροής του κώδικα Arduino που υλοποιήθηκε. Δίδεται κι εξηγείται η λογική με την οποία υλοποιήθηκε η πειραματική διάταξη και αναπτύχθηκε ο κώδικας για την προστασία του φορτίου έναντι υπερφόρτισης. Τέλος παρουσιάζεται φωτογραφικό υλικό της υλοποιημένης διάταξης κατά την διάρκεια των δοκιμών της.

Στο πέμπτο κεφάλαιο περιγράφεται ο τρόπος με τον οποίο ο χρήστης μπορεί να χειριστεί το Arduino μέσω του διαδικτύου ώστε να είναι εφικτός ο απομακρυσμένος έλεγχος και η επίβλεψη του ηλεκτρονόμου μέσω του προγράμματος LabView ή μέσω του υλικού (hardware), ethernet shield της πλατφόρμας Arduino.

Τέλος στο έκτο κεφάλαιο δίδονται τα συμπεράσματα της παρούσας πτυχιακής εργασίας και ο υλοποιημένος κώδικας δίνεται ως παράρτημα .



## Περιεχόμενα

<b>Πρόλογος / Περίληψη</b> .....	<b>iii</b>
<b>Εισαγωγή</b> .....	<b>1</b>
<b>Κεφάλαιο 1ο Ηλεκτρονόμοι</b> .....	<b>2</b>
1.1 Ορισμός και περιγραφή ηλεκτρονόμου.....	2
1.2 Αρχή λειτουργίας του ηλεκτρονόμου.....	4
1.3 Είδη και κατηγορίες ηλεκτρονόμων.....	6
1.4 Προστασία εγκαταστάσεων.....	8
1.5 Κύκλωμα ισχύος ηλεκτρονόμου .....	9
<b>Κεφάλαιο 2ο Arduino</b> .....	<b>11</b>
2.1 Ορισμός.....	11
2.2 Είδη Arduino.....	11
2.3 Προγραμματισμός Arduino.....	17
2.3.1 Εγκατάσταση προγράμματος Arduino IDE.....	17
2.3.2 Περιβάλλον Ανάπτυξης Arduino .....	17
2.3.3 Προγραμματισμός - Βασικές λειτουργίες.....	19
<b>Κεφάλαιο 3ο Arduino και περιφερειακά στοιχεία της πειραματικής διάταξης</b> .....	<b>23</b>
3.1 Επιλογή Arduino.....	23
3.2.1 Τεχνικά χαρακτηριστικά Arduino Mega 2560.....	24
3.2.2 Τάση και ρεύμα λειτουργίας.....	24
3.2.3 Θύρες επικοινωνίας .....	24
3.3 Περιφερειακά στοιχεία της πειραματικής διάταξης .....	25
3.3.1 Εισαγωγή.....	25
3.3.2 Κουμπιά -Buttons.....	25
3.3.3 Ποτενσιόμετρο .....	27
3.3.4 Οθόνη LCD.....	27
3.3.5 Φωτοдиодοι - LED .....	28
3.3.6 Ethernet Shield.....	29
3.3.7 Αισθητήρας θερμοκρασίας.....	30
<b>Κεφάλαιο 4ο</b> .....	<b>35</b>
4.1 Ηλεκτρολογικό σχέδιο πειραματικής διάταξης.....	35
4.2 Διάγραμμα Ροής.....	36
4.3. Πειραματική διαδικασία.....	37
4.3.1 Διαδικασία πειράματος.....	38
4.2.2 Μεθοδολογία μέτρησης από το δίκτυο.....	51
4.2.2.1 Μέτρηση τάσης με διαιρέτη τάσης .....	51
4.2.2.2 Μέτρηση με αισθητήρα Hall.....	51
<b>Κεφάλαιο 5ο Απομακρυσμένος έλεγχος του Arduino</b> .....	<b>53</b>
5.1 Arduino και LabView.....	53
5.1.1 Εγκατάσταση του LabView.....	53
5.1.2 Κατεβάζουμε το Arduino IDE.....	55
5.1.3 Εγκατάσταση του LINX.....	55
5.1.4 Έξοδο στο διαδίκτυο.....	62
5.2 Έξοδο στο διαδίκτυο με Ethernet Shield.....	64
<b>Κεφάλαιο 6ο Συμπεράσματα πτυχιακής εργασίας</b> .....	<b>65</b>
<b>Βιβλιογραφία</b> .....	<b>66</b>
<b>Παράρτημα</b> .....	<b>67</b>

## Εισαγωγή

Από την αρχή της βιομηχανικής εποχής μέχρι και τις μέρες μας το κύριο μέλημα των μηχανικών στον εργασιακό χώρο είναι η ασφάλεια και η προστασία τόσο των ανθρώπων όσο και του εξοπλισμού. Ο ηλεκτρονόμος ως βασικό στοιχείο του κλασικού αυτοματισμού από την αρχή της χρήσης του μέχρι και τις μέρες μας παρέμεινε ουσιαστικά ο ίδιος. Στο πέρασμα του χρόνου δημιουργήθηκε η ανάγκη για επιπρόσθετη προστασία, απομακρυσμένο έλεγχο και επίβλεψη του εξοπλισμού (π.χ. των μηχανών) και έτσι με την χρήση των ηλεκτρονόμων και διάφορων άλλων υλικών και προγραμμάτων ( SCADA , PLC) φτάσαμε στο σημείο να μπορούμε να ελέγχουμε ολόκληρα εργοστάσια μέσα από μερικούς ηλεκτρονικούς υπολογιστές. Η παρούσα πτυχιακή εργασία ασχολείται με τον έλεγχο ηλεκτρονόμου, που τροφοδοτεί μονοφασικό φορτίο, μέσω της πλατφόρμας Arduino με σκοπό την βελτίωση της προστασίας των ενεργητικών ή των παθητικών φορτίων του ηλεκτρονόμου σε περιπτώσεις υπερφόρτισης.

# Κεφάλαιο 1ο

## Ηλεκτρονόμοι

### 1.1 Ορισμός και περιγραφή ηλεκτρονόμου

Ο ηλεκτρονόμος, ρελαί (*relay*) ή ρελές είναι ένας ηλεκτρικός διακόπτης που ανοίγει και κλείνει ένα ηλεκτρικό κύκλωμα κάτω από τον έλεγχο ενός άλλου ηλεκτρικού κυκλώματος. Στην αρχική μορφή του, ένας ηλεκτρομαγνήτης ενεργοποιούσε το διακόπτη, με το άνοιγμα ή κλείσιμο μιας ή περισσότερων επαφών. Εφευρέθηκε από τον Τζόζεφ Χένρυ το 1835. Επειδή ένας ηλεκτρονόμος είναι ικανός να ελέγχει ένα κύκλωμα εξόδου υψηλότερης ισχύος από το κύκλωμα εισόδου, μπορεί να θεωρηθεί, γενικά, μια μορφή ηλεκτρικού ενισχυτή.

Ο ηλεκτρονόμος έχει διάφορων ειδών και λειτουργιών επαφές:

- Κύριες επαφές (υπάρχουν μόνο στο κύκλωμα ισχύος). Το κινούμενο μέρος της επαφής στηρίζεται στον οπλισμό, ενώ στο ένα άκρο του ακίνητου μέρους της επαφής συνδέεται το Δίκτυο (περιττοί αριθμοί) και στο άλλο το φορτίο (άρτιοι αριθμοί). Οι ακροδέκτες των κυρίων επαφών συμβολίζονται από μονοψήφιους αριθμούς (1 έως 6) για 3 πόλους, (1 έως 8) για 4 πόλους, (1 έως 10) για 5 πόλους, ή από τα γράμματα (R, S, T), ( $L_1, L_2, L_3$ ), ( $A_1, B_1, C_1$ ). Οι περιττοί αριθμοί και τα γράμματα τοποθετούνται στην κορυφή και η διαδικασία της αρίθμησης γίνεται από πάνω προς τα κάτω και από αριστερά προς τα δεξιά.
- Βοηθητικές επαφές (υπάρχουν μόνο στο κύκλωμα εντολής). Με τον οπλισμό του ηλεκτρονόμου επιτυγχάνετε άνοιγμα ή το κλείσιμο των βοηθητικών επαφών, που χρησιμοποιούνται είτε για την υλοποίηση του αυτοματισμού (π.χ. άνοιγμα κλείσιμο ηλεκτρονόμων, αυτοσυγκράτηση, ηλεκτρική μανδάλωση) είτε για τις βοηθητικές εργασίες (π.χ. φωτεινή ή ηχητική σήμανση). Οι βοηθητικές επαφές συμβολίζονται με δύο αριθμούς. Ο πρώτος αριθμός συμβολίζει την επαφή και ο δεύτερος την κατάσταση της επαφής σε κατάσταση ηρεμίας π.χ. η επαφή με τον συμβολισμό 21-22 είναι η βοηθητική επαφή No2 του ηλεκτρονόμου και σε κατάσταση ηρεμίας είναι κανονικά-κλειστή. οι βοηθητικές επαφές μπορούν να χωριστούν στις εξής κατηγορίες:
- Στιγμαίεις επαφές (λειτουργία όταν ενεργοποιείται ή μη ο ηλεκτρονόμος χωρίς χρονική καθυστέρηση) οπότε έχουμε τις :
  1. Μια επαφή Κανονικά-Ανοικτή (N/O) συνδέει το κύκλωμα όταν ο ηλεκτρονόμος ενεργοποιείται· το κύκλωμα αποσυνδέεται όταν ο ηλεκτρονόμος είναι ανενεργός. Μια τέτοια επαφή καλείται επίσης Επαφή Μορφής A ή επαφή "make". Η επαφή μορφής A είναι ιδανική για εφαρμογές που απαιτούν την ενεργοποίηση μιας πηγής υψηλής τάσης από απόσταση και συμβολίζονται με διψήφιους αριθμούς που τελειώνουν σε (3) και (4)
  2. Μια επαφή Κανονικά-Κλειστή (N/C) όπου αποσυνδέει το κύκλωμα όταν ο ηλεκτρονόμος ενεργοποιείται το κύκλωμα συνδέεται όταν ο ηλεκτρονόμος είναι ανενεργός. Μια τέτοια επαφή καλείται επίσης Επαφή Μορφής B ή επαφή "break". Η

επαφή μορφής B είναι ιδανική για εφαρμογές που απαιτούν το κύκλωμα να παραμένει κλειστό (ενεργό) μέχρι ο ηλεκτρονόμος να ενεργοποιηθεί που συμβολίζονται με διψήφιους αριθμούς που τελειώνουν σε (1) και (2)

- Επαφή χρονικής καθυστέρησης Κανονικά-Ανοικτή (N/O) και Κανονικά-Κλειστή (N/C), αποκαθιστούν ή διακόπτουν το βοηθητικό κύκλωμα ένα χρονικό διάστημα μετά το άνοιγμα (απενεργοποίηση), ή το κλείσιμο (ενεργοποίηση) του πηνίου του ηλεκτρονόμου, που τις ενεργοποιεί. Οι ακροδέκτες των βοηθητικών επαφών χαρακτηρίζονται από διψήφιους αριθμούς, και τα μεν πρώτα ψηφία καθορίζουν την ταυτότητα, τα δε δεύτερα ψηφία δείχνουν τον ρόλο – εργασία της βοηθητικής επαφής.
- Υπάρχουν και οι μεταγωγικές επαφές που συμβολίζονται με (C/O) και μπορεί να ελέγχει δύο κυκλώματα. Ισοδυναμεί με μια επαφή κανονικά-ανοικτή και μια επαφή κανονικά-κλειστή που έχουν ένα κοινό ακροδέκτη. Μια τέτοια επαφή καλείται επίσης Επαφή Μορφής C

Συνήθως ένας ηλεκτρονόμος αποτελείται από περισσότερες από μία ελεγχόμενες επαφές. Οι επαφές χωρίζονται σε κύριες και βοηθητικές. Οι κύριες διαρρέονται συχνά από ισχυρότερα ρεύματα και έτσι είναι αυτές που διακόπτουν το κύριο κύκλωμα και συνήθως είναι Κανονικά-Ανοικτές. Οι βοηθητικές έχουν όπως υπονοεί και το όνομά τους επικουρικό χαρακτήρα και ο ρόλος τους είναι να βοηθούν στον έλεγχο των αυτοματισμών (που είναι ο κύριος τομέας χρήσης των ηλεκτρονόμων). Για παράδειγμα βοηθούν στην ενεργοποίηση/απενεργοποίηση βοηθητικών κυκλωμάτων όπως ενδεικτικές λυχνίες.



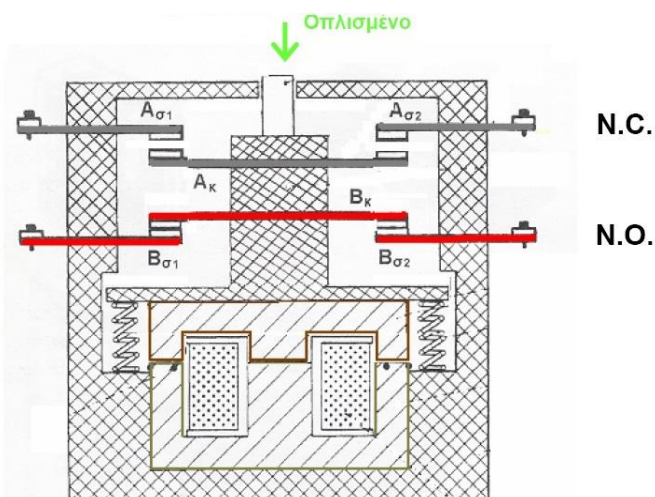
## 1.2 Αρχή λειτουργίας του ηλεκτρονόμου



Εικόνα 1: Βασικά μέρη του ηλεκτρονόμου

Οι ηλεκτρονόμοι λειτουργούν με την βοήθεια ενός ηλεκτρομαγνητικού κυκλώματος το οποίο αποτελείται από έναν ηλεκτρομαγνήτη ο οποίος είναι τοποθετημένος σε ένα σιδηρομαγνητικό υλικό (πυρήνας) και ένα κινητό μεταλλικό οπλισμό ο οποίος φέρει μαζί του και το κινητό μέρος των επαφών οι οποίες όμως είναι ηλεκτρικά απομονωμένες από αυτόν.

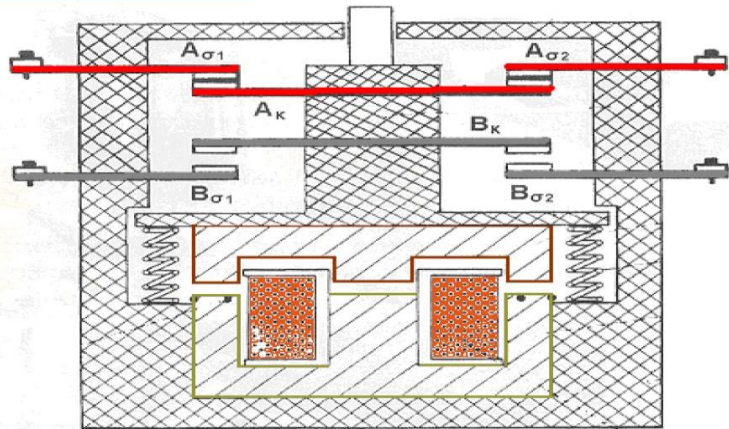
Με την διέγερση του τυλίγματος του ηλεκτρομαγνήτη (οπλισμός του ρελέ) θα έχουμε την έλξη του μεταλλικού οπλισμού και επόμενος το κλείσιμο η αντίστοιχα το άνοιγμα των επαφών του ηλεκτρονόμου.



Εικόνα 2: Ο ηλεκτρονόμος σε κατάσταση οπλισμού

Όταν πάψει να υφίσταται η διέγερση στο πηνίο, τότε όλες οι επαφές θα επανέλθουν στην κατάσταση ηρεμίας τους με την βοήθεια ενός επαναληπτικού ελατηρίου.

**Κατάσταση ηρεμίας:**



*Εικόνα 3: Ο ηλεκτρονόμος πάλι σε κατάσταση ηρεμίας*

### 1.3 Είδη και κατηγορίες ηλεκτρονόμων

Οι ηλεκτρονόμοι ανάλογα με το είδος της τάσης τροφοδοσίας αλλά και το είδος της τάσης την οποία ελέγχουν διακρίνονται σε ηλεκτρονόμους συνεχούς ρεύματος και ηλεκτρονόμους εναλλασσομένου ρεύματος. Επιπλέον οι ηλεκτρονόμοι ανάλογα με την εφαρμογή για την οποία είναι κατασκευασμένοι να λειτουργούν κατατάσσονται στις παρακάτω κατηγορίες:

- Στους ηλεκτρονόμους ισχύος (ρελέ ισχύος) οι οποίοι χρησιμοποιούνται για τον έλεγχο ηλεκτρικών κυκλωμάτων που τροφοδοτούνται από ισχυρά ρεύματα υψηλών τάσεων, δηλαδή χρησιμοποιούνται για τον έλεγχο των κυκλωμάτων ισχύος των εγκαταστάσεων αυτοματισμού.



Εικόνα 4: Ηλεκτρονόμος Ισχύος

- Σε βοηθητικούς ηλεκτρονόμους (ρελέ εντολοδότησης) οι οποίοι χρησιμοποιούνται στα κυκλώματα ελέγχου των εγκαταστάσεων αυτοματισμού.
- Επιτηρητής τάσης :Ο επιτηρητής τάσης κάνει έλεγχο της τάσης και ελέγχει αν αυτή είναι μέσα στα επιτρεπτά όρια και στα τριφασικά αν η διαδοχή των φάσεων είναι ορθή. Στην περίπτωση που δεν πληρούνται τα προηγούμενα, δίνει εντολή στο ρελέ φορτίου να κλείσει το κύκλωμα ή να το ανοίξει. Τέλος Υπάρχουν μονοφασικοί (φάση + ουδέτερος) και τριφασικοί με έλεγχο (3-φάσεων ή 3-φάσεων + ουδέτερο) . Ο



Εικόνα 5: Τριφασικός επιτηρητής τάσης

επιτηρητής έχει επάνω του 2 επαφές μια normally open & μια normally close με τις οποίες μπορούμε να κάνουμε τον έλεγχο του κυκλώματος μέσω ενός ρελαί.

- Σε ρελέ υπερφόρτισης (θερμικά ρελέ) τα οποία χρησιμοποιούνται στα κυκλώματα ισχύος των κινητήρων και ο σκοπός τους είναι να προστατεύσουν τους κινητήρες τους οποίους τροφοδοτούν από υπερφορτίσεις. Αυτές οι υπερφορτίσεις οφείλονται κυρίως σε κάποια δυσλειτουργία της ομαλής λειτουργίας του κινητήρα και μπορούν να προκαλέσουν υπερθέρμανση ή ακόμη και καταστροφή στις περιελίξεις των κινητήρων αυτών.



*Εικόνα 6: Θερμικός ηλεκτρονόμος*

- Τα χρονικά ρελέ τα οποία έχουν ως κύρια λειτουργία την μετάδοση του ηλεκτρικού σήματος που δέχτηκαν στο ηλεκτρικό κύκλωμα το οποίο ελέγχουν, μετά το πέρασ κάποιου συγκεκριμένου χρονικού διαστήματος το οποίο έχουμε ρυθμίσει εμείς εκ των προτέρων. Χρησιμοποιούνται κατά πλείστον σε κινητήρες για την μεταγωγή από αστέρα σε τρίγωνο.



*Εικόνα 7: Χρονικός ηλεκτρονόμος του εμπορίου*

#### 1.4 Προστασία εγκαταστάσεων

Σε μια ηλεκτρική εγκατάσταση μπορούν να προκύψουν ανωμαλίες ή σφάλματα τα οποία μπορούν να προκαλέσουν ζημιές στο κύκλωμα ισχύος-ελέγχου, στο εγκαταστημένο φορτίο, στην τροφοδοσία αλλά και θανατηφόρες ηλεκτροπληξίες και πυρκαγιές. Τα σφάλματα αυτά ανιχνεύονται από τα μέσα προστασίας όπου διακόπτουν την τροφοδότηση του κυκλώματος. Τα κύρια μέσα προστασίας ηλεκτρολογικών εγκαταστάσεων είναι τα τηκτά (ασφάλειες), οι μικροαυτόματοι και οι διακόπτες ισχύος, τα θερμικά, και διακόπτες διαφυγής έντασης (αντιηλεκτροπληξιακό ρελέ).

Ένα συχνό πρόβλημα στα ηλεκτρικά κυκλώματα κινητήρων είναι στη ζήτηση περισσότερης μηχανικής ισχύς από την ονομαστική ισχύ του κινητήρα. Αυτό έχει ως αποτέλεσμα την αύξηση του ρεύματος πέραν της ονομαστικής τιμής. Το αποτέλεσμα είναι η υπερφόρτιση του κυκλώματος και η υπερθέρμανση των τυλιγμάτων του κινητήρα.

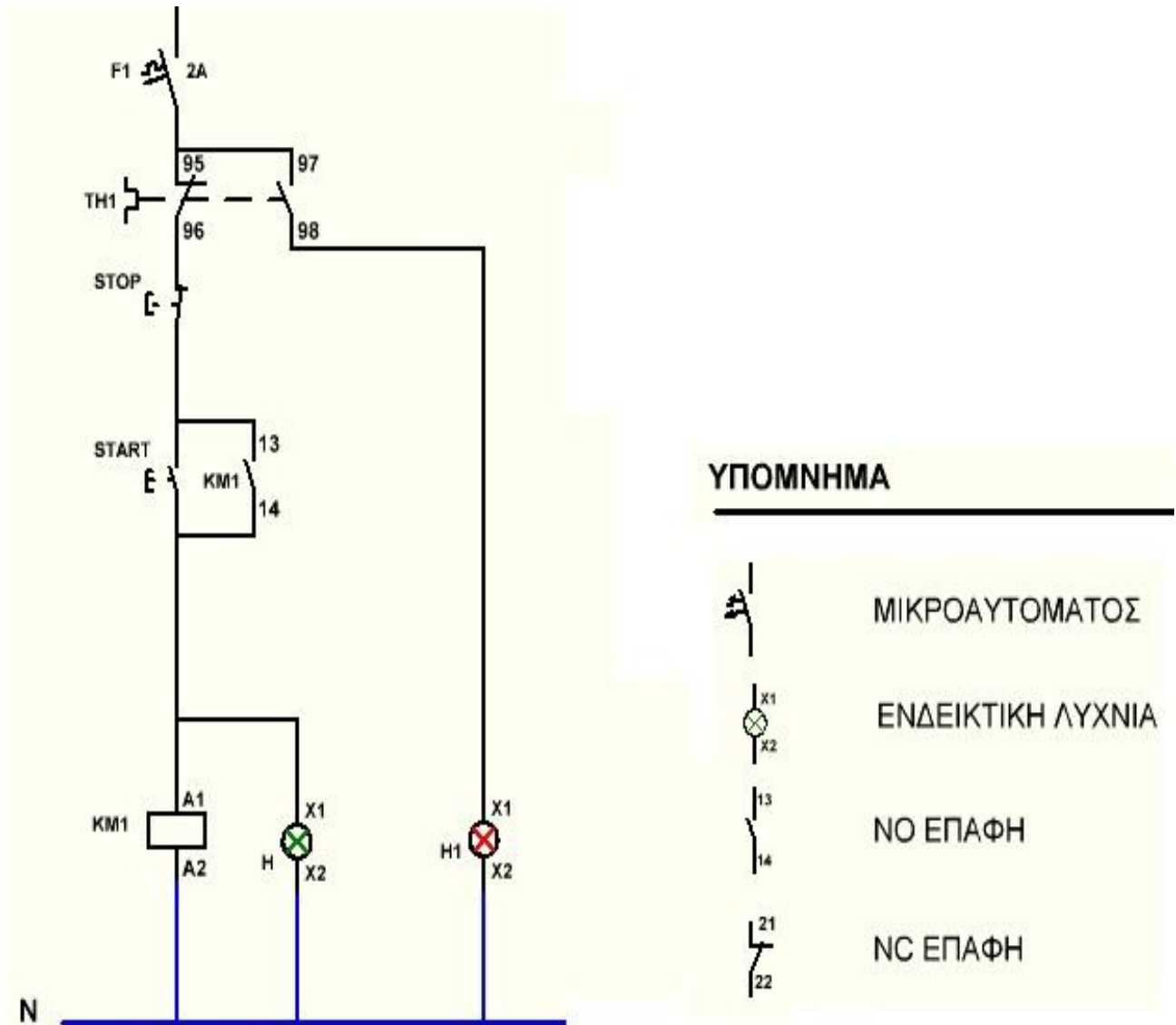
Η υπερθέρμανση ενός κινητήρα (10°C υπεράνω της θερμοκρασίας της κατηγορίας μόνωσης σημαίνει 50% μικρότερη διάρκεια ζωής του κινητήρα) μπορεί να προέλθει επίσης από ανεπαρκή ψύξη (όταν ο αέρας είναι αραιός) ή από βραχυκύκλωμα. Επομένως συνεχής υπερφόρτιση μπορεί να προκληθεί από τα εξής γεγονότα :

- μεγάλη ροπή φορτίου,
- μειωμένη τάση,
- βαριά εκκίνηση,
- επανειλημμένες εκκινήσεις, και
- έλλειψη μιας φάσης.

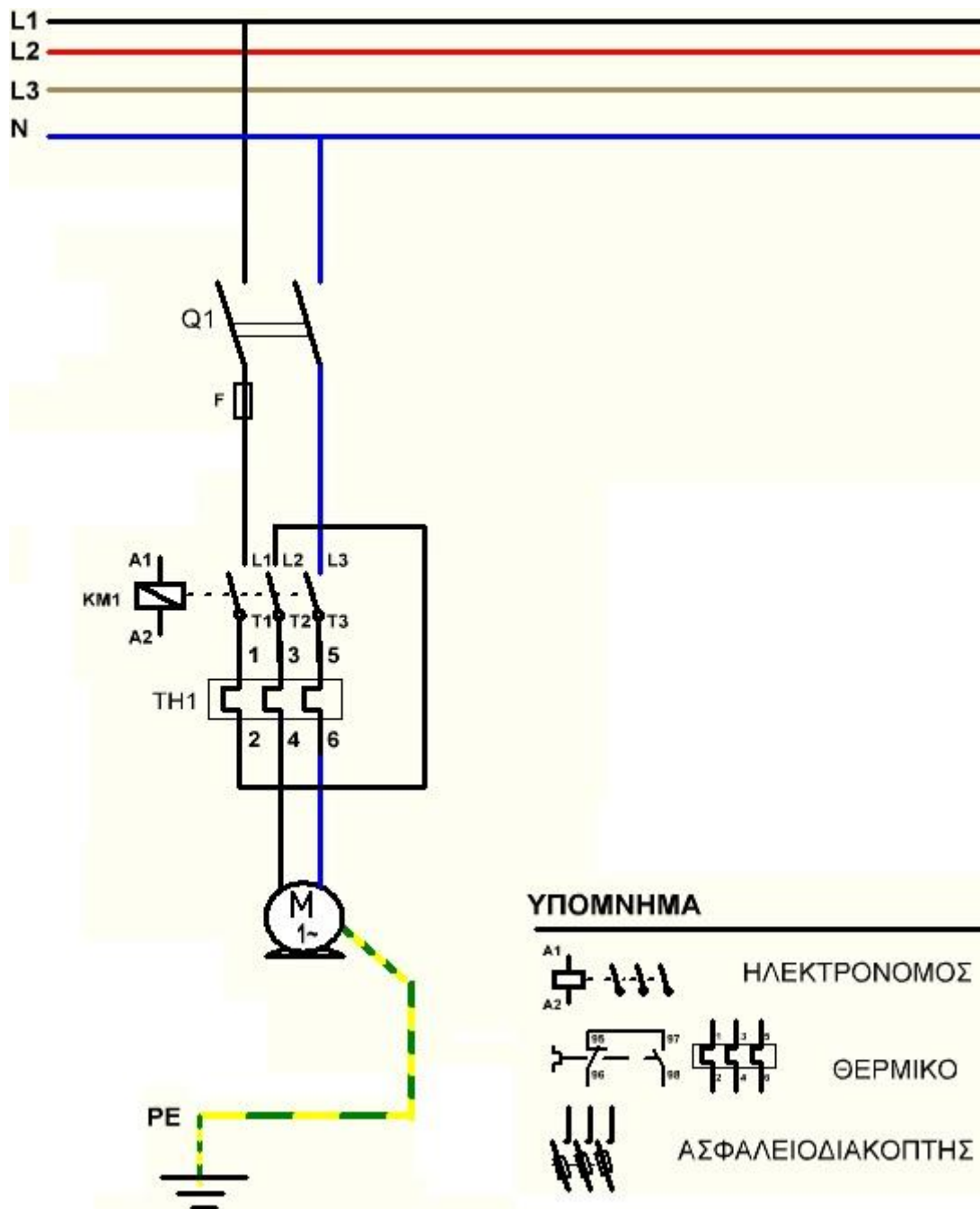
Άλλα σφάλματα που μπορούν να προκύψουν είναι η υπέρταση, υπόταση, η διακοπή φάσης, το βραχυκύκλωμα. Τα μέσα προστασίας πρέπει να προστατεύουν το ηλεκτρολογικό κύκλωμα από τα προαναφερθέντα σφάλματα.

### 1.5 Κύκλωμα ισχύος ηλεκτρονόμου

Στο παραπάνω σχήμα βλέπουμε ένα κλασικό κύκλωμα ελέγχου ενός πηνίου ηλεκτρονόμου .



Εικόνα 8: Βοηθητικό κύκλωμα ηλεκτρονόμου



Εικόνα 9: Κύκλωμα ισχύος

## Κεφάλαιο 2ο Arduino

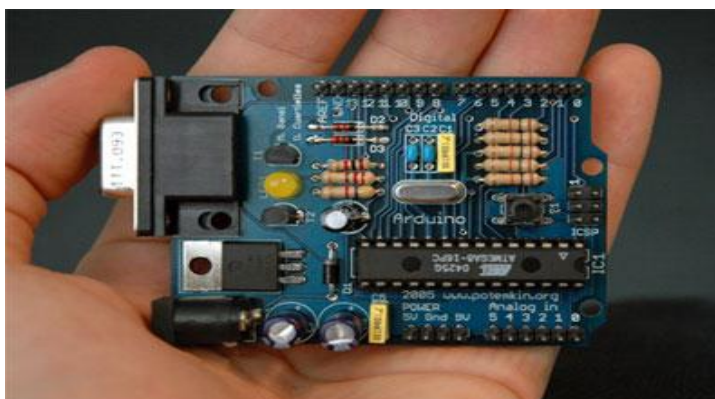
### 2.1 Ορισμός

Το Arduino είναι μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, Super Collider. Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προ-συναρμολογημένες· το διάγραμμα και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους.

### 2.2 Είδη Arduino

Το πρωτότυπο υλισμικό του Arduino κατασκευάζεται από την Ιταλική εταιρία Smart Projects. Κάποιες πλακέτες με την μάρκα του Arduino έχουν σχεδιαστεί από την Αμερικάνικη εταιρία SparkFun Electronics. Δεκαέξι εκδοχές του Arduino Hardware έχουν χρησιμοποιηθεί εμπορικά μέχρι τώρα:

1. Το Serial Arduino, προγραμματισμένο με μία σειριακή DE-9 σύνδεση χρησιμοποιώντας τεχνολογία ATmega8



Εικόνα 10: Το Serial Arduino

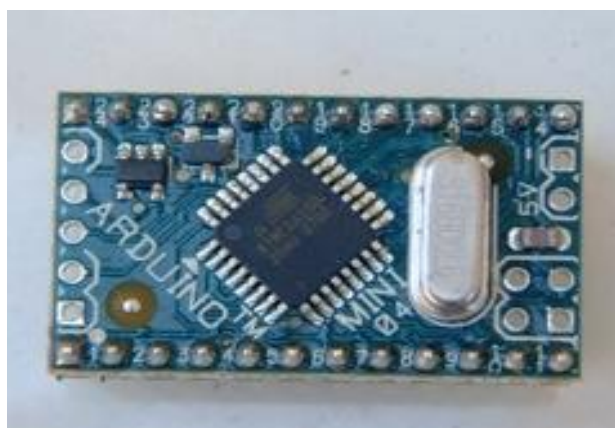
2. Το Arduino Extreme, με ένα USB interface για προγραμματισμό χρησιμοποιώντας τεχνολογία Atmega8





*Εικόνα 11: Το Arduino Extreme*

3. Το Arduino Mini, μία έκδοση μινιατούρας του Arduino χρησιμοποιώντας τεχνολογία surface-mounted Atmega168



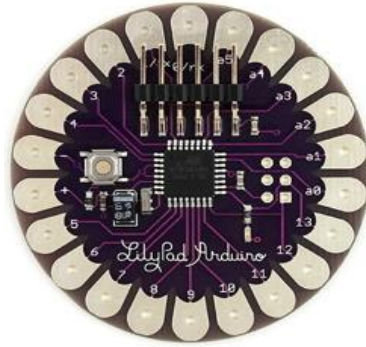
*Εικόνα 12: Το Arduino Mini*

4. Το Arduino Nano, ένα ακόμα πιο μικρό, USB τροφοδοτούμενη εκδοχή του Arduino χρησιμοποιώντας τεχνολογία surface-mounted ATmega168 (ATmega328 για την νεότερη έκδοση)



*Εικόνα 13: Το Arduino Nano*

5. Το LilyPad Arduino, ένα μινιμαλιστικό σχέδιο για εφαρμογές ένδυσης και E-textiles χρησιμοποιώντας τεχνολογία surface-mounted AT-mega328



*Εικόνα 14: Το LilyPad Arduino*

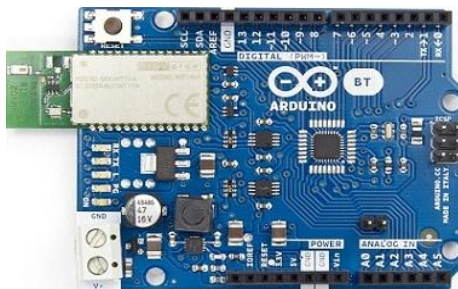
6. Το Arduino NG, με ένα USB interface για προγραμματισμό και χρησιμοποιώντας



*Εικόνα 15: Το Arduino NG*

τεχνολογία ATmega8

7. Το Arduino Bluetooth, με Bluetooth interface για προγραμματισμό χρησιμοποιώντας τεχνολογία Atmega168



*Εικόνα 16: Το Arduino Bluetooth*

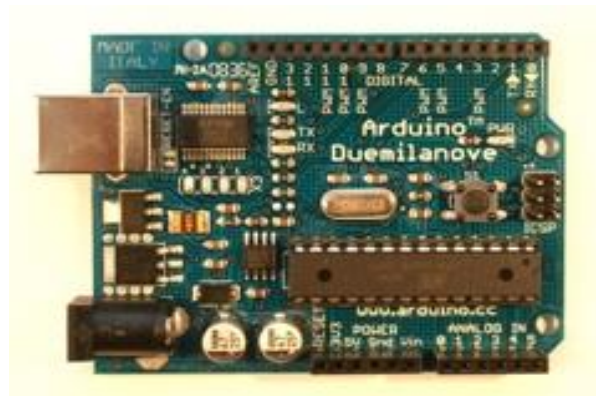
8 Το Arduino Diecimila, με ένα USB interface και χρησιμοποιεί τεχνολογία ATmega168 σε

ένα DIP28 πακέτο



*Εικόνα 17: Το Arduino Diecimila*

9 Το Arduino Duemilanove (“2009”), χρησιμοποιεί τεχνολογία ATmega168 (ATmega328 για την καινούργια έκδοση) και τροφοδοτείτε μέσω ενέργειας USB/DC, αυτόματα εναλλασσόμενης



*Εικόνα 18: Το Arduino Duemilanove*

10 Το Arduino Mega, χρησιμοποιώντας τεχνολογία surface-mounted ATmega1280 για περαιτέρω I/O και μνήμη



*Εικόνα 19: Το Arduino Mega*

11. Το Arduino Uno, χρησιμοποιώντας την ίδια τεχνολογία ATmega328 όπως το τελευταίο μοντέλο Duemilανone, αλλά ενώ το Duemilανone χρησιμοποιεί ένα FTDI chipset για το USB, το Uno χρησιμοποιεί τεχνολογία ATmega8U2 προγραμματισμένο ως σειριακός μετατροπέας



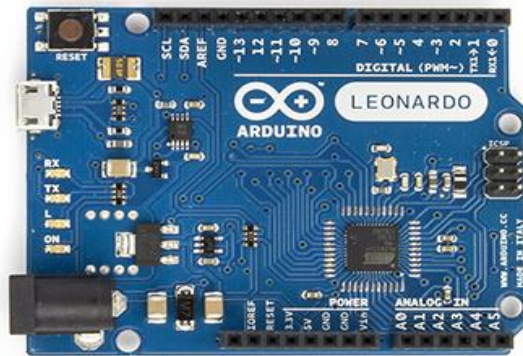
*Εικόνα 20: Το πιο συχνό Arduino, Arduino Uno*

12. Arduino Mega2560, χρησιμοποιεί τεχνολογία surface-mounted ATmega2560 φέρνοντας την ολική μνήμη στα 256kB. Επίσης ενσωματώνει την νέα τεχνολογία ATmega8U2 (ATmega16U2 σε αναθεώρηση τύπου 3) USB chipset.



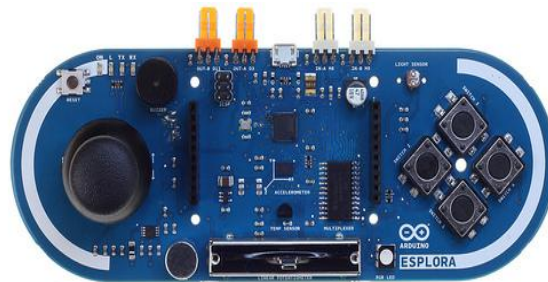
*Εικόνα 21: Arduino Mega2560*

13. Το Arduino Leonardo, με ένα ATmega32U4 chip που εξαλείφει την ανάγκη για συνδεσιμότητα μέσω USB και μπορεί να χρησιμοποιηθεί ως ψηφιακό πληκτρολόγιο ή ποντίκι. Κυκλοφόρησε στο Maker Faire Bay Area το 2012



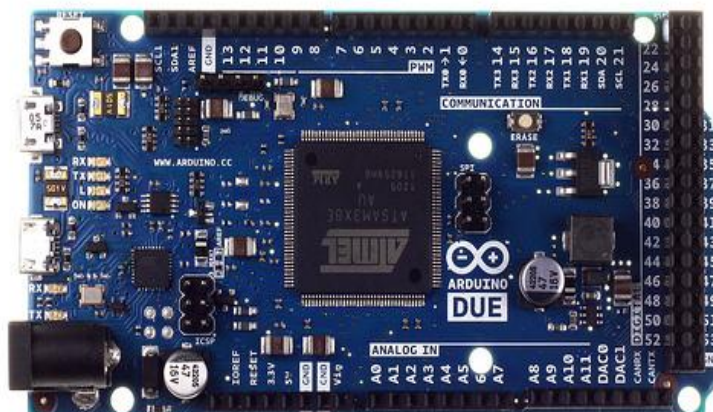
Εικόνα 22: Το Arduino Leonardo

14. Το Arduino Esplora, με εμφάνιση που παραπέμπει σε χειριστήριο κονσόλας βιντεοπαιχνιδιών με joystick και ενσωματωμένους αισθητήρες για ήχο, φως, θερμοκρασία και επιτάχυνση



Εικόνα 23: Το Arduino Esplora

15. Το Arduino Due είναι ένα μικροχειριστήριο board βασισμένο στην τεχνολογία Atmel SAM3X8E ARM Cortex-M3 CPU. Είναι το πρώτο board της Arduino βασισμένη σε επεξεργαστή 32-bit ARM microcontroller



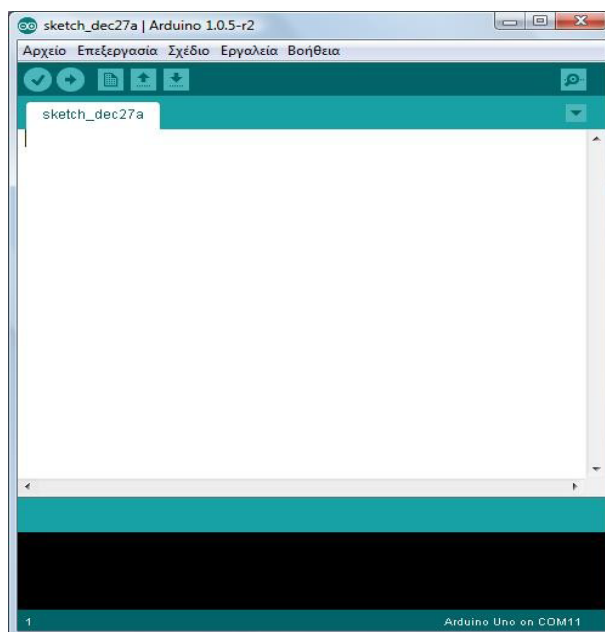
## 2.3 Προγραμματισμός Arduino.

### 2.3.1 Εγκατάσταση προγράμματος Arduino IDE

Για να προγραμματίσετε τη μονάδα σας θα χρειαστείτε το περιβάλλον προγραμματισμού Arduino IDE . Στο περιβάλλον αυτό γράφετε κώδικα (βασίζεται στη γλώσσα C/C++) τον οποίο μετά μεταγλωττίζετε και μεταφορτώνεται στη μονάδα σας. Το Arduino IDE υπάρχει σε εκδόσεις για Windows, Mac και Linux και μπορείτε να το κατεβάσετε εντελώς δωρεάν από την επίσημη ιστοσελίδα (<http://arduino.cc/en/Main/Software>).

### 2.3.2 Περιβάλλον Ανάπτυξης Arduino

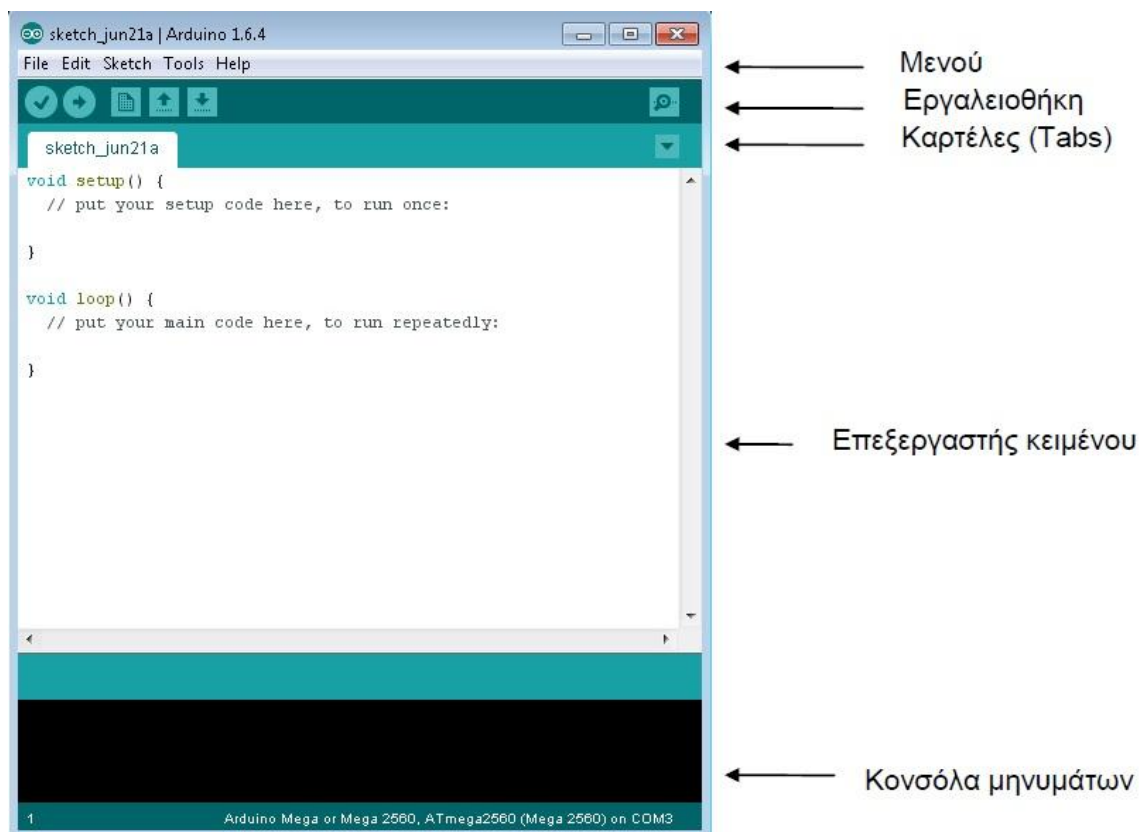
Το περιβάλλον ανάπτυξης Arduino περιέχει μια περιοχή επεξεργασίας κειμένου για τη συγγραφή κώδικα, μια περιοχή μηνυμάτων, ένα μενού, μια γραμμή εργαλείων με κουμπιά για κοινές λειτουργίες, καθώς και μια σειρά από μενού. Συνδέεται με το υλικό Arduino για τη φόρτωση προγραμμάτων και για να επικοινωνούν μεταξύ τους.



Εικόνα 24: Απλό παράθυρο της εφαρμογής Arduino IDE

Ένα ολοκληρωμένο πρόγραμμα συνήθως ονομάζεται sketch. Αυτό το sketch είναι γραμμένο με το πρόγραμμα επεξεργασίας κειμένου. Έχει δυνατότητες για την αντιγραφή/επικόλληση και για την αναζήτηση/αντικατάσταση κειμένου. Η κονσόλα απεικονίζει την έξοδο του κειμένου από το περιβάλλον Arduino συμπεριλαμβάνοντας πλήρη μηνύματα λάθους και άλλες πληροφορίες. Τα κουμπιά της γραμμής εργαλείων επιτρέπουν τον έλεγχο και το ανέβασμα των προγραμμάτων, τη δημιουργία νέου sketch, το άνοιγμα και την αποθήκευση των sketch και άνοιγμα της σειριακής οθόνης.

Στις παρακάτω φωτογραφίες βλέπουμε το περιβάλλον ανάπτυξης και τι κάνει αναλυτικά το κάθε κουμπί του προγράμματος.



Εικόνα 25: Παράθυρο της εφαρμογής Arduino IDE με σημειώσεις λειτουργίας

Τα κουμπιά της γραμμής εργαλείων:



Verify/Compile (Έλεγχος/Μεταγλώττιση): Έλεγχος για λάθη στον κώδικα



Upload: Ανέβασμα του κώδικα στον μικροελεγκτή



New(Νέο): Δημιουργεί ένα νέο sketch



Open(Άνοιγμα): Παρουσιάζει ένα μενού με όλα τα sketch, κάνοντας κλικ σε ένα από αυτά θα ανοίξει μέσα στο τρέχον παράθυρο



Save(Αποθήκευση): Αποθηκεύει το sketch



Serial Monitor(Σειριακή οθόνη): Ανοίγει την σειριακή οθόνη ώστε να μπορούμε να δώσουμε δεδομένα από το πληκτρολόγιο

Ει

Εικόνα 26: Αναλυτική λειτουργία κάθε κουμπιού του περιβάλλοντος Arduino IDE

### 2.3.3 Προγραμματισμός - Βασικές λειτουργίες

Μετά την εγκατάσταση του Arduino IDE μπορούμε να γράψουμε τα πρώτα μας τμήματα κώδικα. Η λογική του Arduino είναι πολύ απλή - στην ουσία υπάρχουν δύο βασικές συναρτήσεις, η `setup()` και η `loop()` οι οποίες δουλεύουν ως εξής:

- **setup()** - εδώ βάζουμε όλες τις εντολές που πρέπει να τρέξουν μία φορά, όταν ενεργοποιείται η μονάδα μας (όταν δηλαδή δίνουμε ρεύμα ή όταν πατηθεί το πλήκτρο `reset` που υπάρχει). Συνήθως μπαίνουν αρχικοποιήσεις τιμών μεταβλητών και οπωσδήποτε ο χαρακτηρισμός των εισόδων/εξόδων που θα χρησιμοποιήσουμε (αν δηλαδή ένα συγκεκριμένο Pin θα είναι είσοδος ή έξοδος).

- **loop()** - εδώ γράφουμε το πρόγραμμά μας. Οι εντολές που υπάρχουν θα τρέξουν κι όταν φτάσει στο τέλος θα ενεργοποιηθεί ξανά η `loop()`, συνεχίζοντας από την αρχή της, και ξανά. Αυτό θα συμβαίνει συνεχώς, όσο έχει ρεύμα το Arduino ή μέχρι να πατηθεί το πλήκτρο `reset`.

Έτσι, η βασική λειτουργία του Arduino είναι ότι τρέχει η συνάρτηση `setup()` μία φορά στην αρχή και ακολούθως η `loop()` ξανά και ξανά μέχρι να το κλείσουμε (να μην τροφοδοτείται με ρεύμα) ή να πατήσουμε το πλήκτρο `reset`. Στην περίπτωση του `Reset` ξανατρέχει η συνάρτηση `setup()` μία φορά και ακολούθως η `loop()` ξανά και ξανά, όπως δηλαδή ακριβώς και όταν αρχικά ενεργοποιείται με ρεύμα ο μικροελεγκτής. Στην περίπτωση που έχουμε κάνει αλλαγές στο πρόγραμμά μας και το φορτώσουμε στον μικροελεγκτή (θα δούμε παρακάτω τη διαδικασία αυτή) αρκεί να πατήσουμε το πλήκτρο `Reset` ώστε να φορτώσει το πρόγραμμά μας από την αρχή με τον τρόπο που περιγράφηκε. Ένα τυπικό πρόγραμμα έχει την παρακάτω δομή:

```
void setup() {  
/* οι εντολές εδώ θα τρέξουν μόνο στην ενεργοποίηση ή μετά από Reset */  
}  
void loop() {  
/* οι εντολές εδώ θα τρέχουν ξανά και ξανά,  
μέχρι να απενεργοποιηθεί ή να πατηθεί το Reset */  
}
```

### 2.3.4 Βασικές δομές ,λειτουργίες και σταθερές

Παρακάτω, ακολουθούν μερικές από τις πιο βασικές δομές και λειτουργίες που μπορεί να αξιοποιηθεί ως εργαλεία κατά την συγγραφή ενός προγράμματος Arduino :

Δομές ελέγχου ροής

- `if` (δομή ελέγχου μίας συνθήκης)
- `if ... else` (δομή ελέγχου πολλαπλών συνθηκών)
- `for` (δομή επαναληπτικού ελέγχου συνθήκης)
- `while` (δομή επαναληπτικού ελέγχου συνθήκης)
- `do ... while` (δομή επαναληπτικού ελέγχου συνθήκης)
- `switch ... case` (δομή ελέγχου περιπτώσεων)
- `break` (εντολή διακοπής μιας επαναληπτικής δομής)
- `continue` (εντολή παράλειψης της τρέχουσας επανάληψης)
- `return` (εντολή επιστροφής από μία συνάρτηση)
- `goto` (εντολή μετάβασης σε κάποιο σημείο του κώδικα)

Αριθμητικοί τελεστές

- `=` (τελεστής εκχώρησης)
- `+` (τελεστής πρόσθεσης)



- - (τελεστής αφαίρεσης)
- \* (τελεστής πολλαπλασιασμού)
- / (τελεστής διαίρεσης)
- % (τελεστής υπόλοιπου ακεραίας διαίρεσης)

#### Λογικοί τελεστές

- && (λογική σύζευξη)
- || (λογική διάζευξη)
- ! (λογική άρνηση)

#### Δυαδικοί τελεστές

- & (δυαδική σύζευξη)
- | (δυαδική διάζευξη)
- ^ (δυαδική αποκλειστική διάζευξη)
- ~ (δυαδική άρνηση)
- << (δυαδική αριστερή ολίσθηση)
- >> (δυαδική δεξιά ολίσθηση)

#### Τελεστές αύξησης και μείωσης

- ++ (αύξηση κατά μία ακέραιη μονάδα)
- -- (μείωση κατά μία ακέραιη μονάδα)

#### Σύνθετοι τελεστές

- +=, -=, \*=, /=, %= (σύνθετοι αριθμητικοί τελεστές)
- &=, |=, ^=, ~=, <<=, >>= (σύνθετοι δυαδικοί τελεστές)

#### Τελεστές σύγκρισης

- == (ισότητα)
- != (ανισότητα)
- < (μικρότερο)
- > (μεγαλύτερο)
- <= (μικρότερο ή ίσο)
- >= (μεγαλύτερο ή ίσο)

#### Τελεστές δεικτών

- \* (τελεστής απόκτησης περιεχομένου)
- & (τελεστής απόκτησης διεύθυνσης)

#### Σταθερές

- HIGH (τιμή υψηλής στάθμης για μία επαφή εισόδου ή εξόδου)
- LOW (τιμή χαμηλής στάθμης για μία επαφή εισόδου ή εξόδου)
- false (λογικό επίπεδο ψεύδους σε μία συνθήκη)
- true (λογικό επίπεδο αλήθειας σε μία συνθήκη)
- INPUT (χρησιμοποιείται για τον ορισμό μίας επαφής ως είσοδο)
- OUTPUT (χρησιμοποιείται για τον ορισμό μίας επαφής ως έξοδο)
- A0, ..., A15 (σύμβολο-σταθερές για τις αναλογικές επαφές εισόδου)
- 0, ..., 54 (σύμβολο-σταθερές για τις ψηφιακές επαφές εισόδου-εξόδου)

#### Τύποι δεδομένων

- boolean (λογική δυαδική τιμή)

- char (προσημασμένος χαρακτήρας 8 ψηφίων)
- unsigned char (μη προσημασμένος χαρακτήρας 8 ψηφίων)
- byte (μη προσημασμένος χαρακτήρας 8 ψηφίων)
- int (προσημασμένος ακέραιος αριθμός 16 ψηφίων)
- unsigned int (μη προσημασμένος ακέραιος αριθμός 16 ψηφίων)
- word (μη προσημασμένος ακέραιος αριθμός 16 ψηφίων)
- long (προσημασμένος ακέραιος αριθμός 32 ψηφίων)
- unsigned long (μη προσημασμένος ακέραιος αριθμός 32 ψηφίων)
- float, double (αριθμός κινητής υποδιαστολής απλής ακρίβειας)
- String (αντικείμενο αλφαριθμητικού με χρήσιμες μεθόδους)

#### Συναρτήσεις μετατροπής τύπων

- boolean, με τιμές το 0 και 1 (ή True – False)
- byte, με τιμές από 0 έως και 255
- int, ακέραιος με δυνατές τιμές από -32768 έως και 32767
- long, ακέραιος με δυνατές τιμές από -2147483648 έως και 2147483647
- float, δεκαδικοί αριθμοί
- char, ένας χαρακτήρας (μέγεθος ένα Byte)
- string, πίνακας χαρακτήρων

#### Συναρτήσεις εισόδου και εξόδου

- pinMode() (ορίζει μια επαφή ως είσοδο ή έξοδο)

#### Συναρτήσεις ψηφιακής εισόδου και εξόδου

- digitalWrite() (γράφει σε μία ψηφιακή επαφή εξόδου)
- digitalRead() (διαβάζει από μία ψηφιακή επαφή εισόδου)

#### Συναρτήσεις αναλογικής εισόδου και εξόδου

Το Arduino έχει 16 αναλογικές εισόδους, οι οποίες χαρακτηρίζονται με τα σύμβολα A0, A1, A2, A3,...,A14, A15. Μπορούμε να συνδέσουμε κάποιο αναλογικό εξάρτημα (π.χ. ένα ποτενσιόμετρο) και να το διαβάσουμε ως είσοδο. Αυτό γίνεται με χρήση της συνάρτησης **analogRead(Pin)**, όπου το όρισμα Pin αναφέρεται στο νούμερο της θύρας για την οποία θα πάρουμε είσοδο, ενώ η συνάρτηση επιστρέφει με το όνομά της την τιμή εισόδου. Η τιμή εισόδου κυμαίνεται από 0 μέχρι και 1023.

- analogReference() (ορίζει την τάση αναλογικής αναφοράς)
- analogWrite() (γράφει PWM σήματα σε μία επαφή εξόδου)
- analogRead() (διαβάζει από μία αναλογική επαφή εισόδου)

**Προσοχή:** Η αντίστοιχη θύρα θα πρέπει να έχει οριστεί ως είσοδο στη διαδικασία setup(), με χρήση της συνάρτησης pinMode.

#### Συναρτήσεις εισόδου και εξόδου

- pinMode() (ορίζει μια επαφή ως είσοδο ή έξοδο)

#### Προηγμένες συναρτήσεις εισόδου και εξόδου

- tone() (παράγει ένα τετραγωνικό σήμα ορισμένης συχνότητας)
- noTone() (διακόπτει την παραγωγή τετραγωνικών σημάτων)
- shiftOut() (ολισθαίνει τα ψηφία μιας τιμής σε μία επαφή εξόδου)
- pulseIn() (επιστρέφει την διάρκεια σε ms ενός παλμού HIGH ή LOW)

## Συναρτήσεις χρόνου

Το Arduino έχει ενσωματωμένο ρολόι, το οποίο μετράει το χρόνο από τη στιγμή που ενεργοποιείται (ή του γίνεται reset). Η πληροφορία αυτή μας είναι διαθέσιμη σε κάθε σημείο με κλήση της συνάρτησης `millis()`, η οποία μας επιστρέφει το χρόνο σε milliseconds (1/1000 sec) που έχει περάσει από την ενεργοποίηση της μονάδας μας. Αυτό μας βοηθά να μετράμε το χρόνο στα

προγράμματά μας, ειδικά στις περιπτώσεις που θέλουμε να “θυμόμαστε” πράγματα.

Για παράδειγμα, αν θέλουμε να ελέγξουμε αν έχει περάσει συγκεκριμένο διάστημα από τότε που

έγινε κάτι (π.χ. πατήθηκε τελευταία φορά ένα πλήκτρο), μπορούμε να καταγράψουμε το γεγονός σε

μια μεταβλητή χρόνου και τον χρόνο αυτό να τον αφαιρούμε από τον επόμενο κτλ.

- `millis()` (διάρκεια εκτέλεσης του προγράμματος σε ms)
- `micros()` (διάρκεια εκτέλεσης του προγράμματος σε μs)

Στο πρόγραμμά μας μπορούμε να ορίσουμε μια καθυστέρηση ώστε να διαρκέσει για το χρόνο που εμείς ορίζουμε ένα γεγονός. Αυτό το επιτυγχάνουμε με χρήση της συνάρτησης `delay(time)` όπου στη θέση `time` δίνουμε το χρόνο σε ms (1/1000 sec). Η εντολή `delay(time)` σημαίνει ότι σταματά στο σημείο αυτό η εκτέλεση του προγράμματός μας για το χρόνο `time`.

- `delay()` (παύση προγράμματος - η διάρκεια δίδεται σε ms)
- `delayMicroseconds()` (παύση προγράμματος - η διάρκεια δίδεται σε μs)

## Μαθηματικές και Τριγωνομετρικές συναρτήσεις

- `max()` (βρίσκει τον μεγαλύτερο ανάμεσα σε δύο αριθμούς)
- `min()` (βρίσκει τον μικρότερο ανάμεσα σε δύο αριθμούς)
- `abs()` (επιστρέφει την απόλυτη τιμή ενός αριθμού)
- `constrain()` (ελέγχει για υπερχείλιση ή υποχείλιση ορίων)
- `map()` (πραγματοποιεί γραμμικό μετασχηματισμό ορίων)
- `pow()` (επιστρέφει το αποτέλεσμα μίας δύναμης)

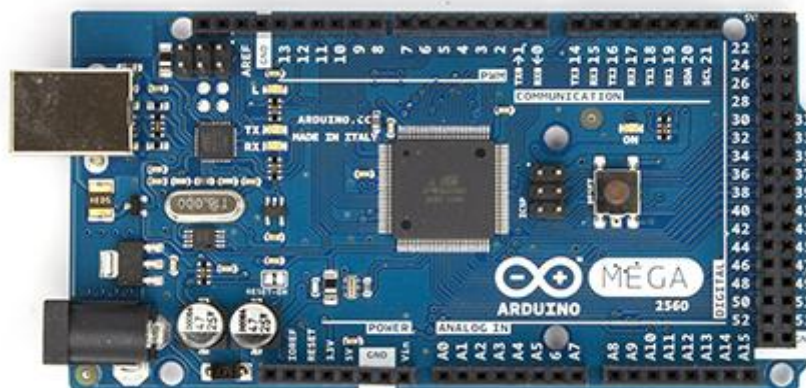
## Κεφάλαιο 3ο

### Arduino και περιφερειακά στοιχεία της πειραματικής διάταξης

#### 3.1 Επιλογή Arduino

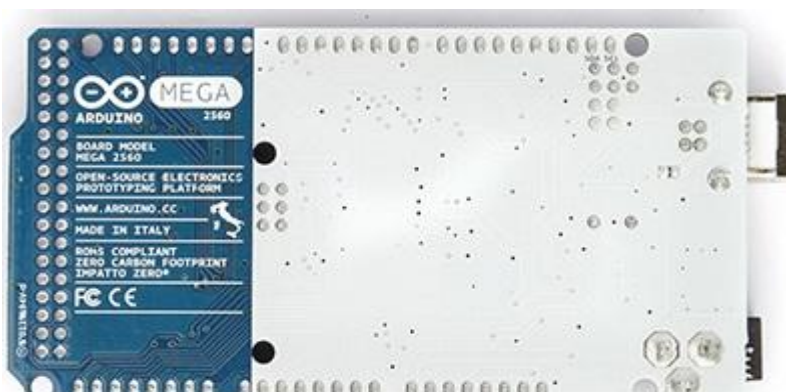
Για την παρούσα πτυχιακή εργασία θα χρησιμοποιηθεί ο Arduino Mega 2560. Ο λόγος είναι το μεγάλο πλήθος εισόδων και εξόδων που μας προσφέρει άλλα και το μεγάλο μέγεθος της μνήμης του επεξεργαστή. Αν και το πρόγραμμα που κατασκευάσαμε και η κατασκευή που υλοποιήσαμε θα μπορούσε να υλοποιηθεί και με έναν Arduino UNO , ο λόγος που επιλέξαμε τον Mega είναι ότι περίπτωση όπου είτε χρησιμοποιούσαμε το Ethernet Shield (πλακέτα ενσύρματης σύνδεσης Arduino με το διαδίκτυο) ή θέλαμε να επεκτείνουμε την λειτουργία του προγράμματος και να προσθέταμε λειτουργίες όπως αντίστροφη περιστροφή κινητήρα ή έλεγχος δεύτερου ηλεκτρονόμου , τότε θα είχαμε περιορισμό στις θύρες του Arduino αλλά και στην χωρητικότητα του προγράμματός μας . Για αυτό τον λόγο επιλέξαμε το συγκεκριμένο μοντέλο. Ας δούμε πιο αναλυτικά τον Arduino Mega 2560 και τι μας προσφέρει.

Το Arduino Mega 2560 από την επάνω όψη



Εικόνα 27: Η επάνω όψη του Arduino Mega 2560

Το Arduino Mega 2560 από την κάτω όψη



Εικόνα 28: Η κάτω όψη του Arduino Mega 2560

### 3.2.1 Τεχνικά χαρακτηριστικά Arduino Mega 2560

Όπως είπαμε και πιο πριν Arduino Mega2560, χρησιμοποιεί τεχνολογία Surface-mounted όπου είναι μια μέθοδος για την παραγωγή ηλεκτρονικών κυκλωμάτων στα οποία τα συστατικά τοποθετούνται ή έχουν τοποθετηθεί απευθείας πάνω στην επιφάνεια των τυπωμένων κυκλωμάτων ATmega2560 φέρνοντας την ολική μνήμη στα 256kB. Επίσης ενσωματώνει την νέα τεχνολογία ATmega8U2 (ATmega16U2 σε αναθεώρηση τύπου 3) και USB chipset με το οποίο μπορούμε να έχουμε επικοινωνία με το Arduino. Παρακάτω ακολουθεί αναλυτικός πίνακας στοιχείων του Arduino Mega 2560.

Επεξεργαστής	ATmega2560
Λειτουργική Τάση	5V
Τάση λειτουργίας (συνιστάται)	7-12V
Τάση λειτουργίας (όρια)	6-20V
Ψηφιακές I/O ακροδέκτες	54 (των οποίων 15 παρέχουν PWM έξοδο)
Ακροδέκτες αναλογικών εισόδων	16
Συνεχές ρεύμα ανα ακροδέκτη εισόδου/εξόδου	40 mA
Συνεχές ρεύμα για τον ακροδέκτη των 3.3V	50 mA
Flash μνήμη	256 KB των οποίων 8 KB χρησιμοποιείτε από τον bootloader
Στατική μνήμη (SRAM)	8 KB
EEPROM	4 KB
Ταχύτητα επεξεργαστή	16 MHz

### 3.2.2 Τάση και ρεύμα λειτουργίας.

Το Arduino μπορεί να δουλέψει με ρεύμα από τη USB θύρα του υπολογιστή σας ή με αυτόνομη παροχή ρεύματος από μπαταρία. Η μονάδα παρέχει σταθερά τάση 5V στις εξόδους της. Για παροχή ρεύματος στη μονάδα από εξωτερική πηγή δέχεται τροφοδοσία από εξωτερικό βύσμα -συνιστώμενη παρεχόμενη τάση λειτουργίας είναι στα 7V έως 12V, ώστε να μπορεί να λειτουργήσει και να δώσει σταθερά τα 5V στην έξοδο . Μπορείτε να συνδέσετε την παροχή ρεύματος απευθείας στα pins που προορίζονται για αυτό το σκοπό: (+) στο Pin VCC IN και (-) στο Gnd δίπλα του. Στην περίπτωση που είναι συνδεδεμένη η μονάδα σας μόνιμα με θύρα USB τότε δουλεύει χωρίς πρόβλημα με τα 5V που παρέχει η USB θύρα.

### 3.2.3 Θύρες επικοινωνίας .

Το Arduino Mega 2560 έχει 54 ψηφιακές θύρες εισόδου ή εξόδου (digital input/output pins) και δεκαέξι αναλογικές εισόδους (analog input pins). Οι 54 ψηφιακές θύρες ονομάζονται με νούμερα από το 0 έως το 53, ενώ οι δεκαέξι αναλογικές με το γράμμα A ακολουθούμενο από ένα νούμερο από 0 μέχρι το 15 (π.χ. A3). Στην έξοδο τα pins μπορούν να δώσουν 0 έως και 5V τάση. Από τις 54 ψηφιακές θύρες οι δεκαπέντε (2 έως 13 , 44 και 46)

είναι PWM θύρες (Pulse Width Modulation), δηλαδή μπορούν να προσομοιάσουν αναλογικές εξόδους.

Έτσι, συνοπτικά για την είσοδο και έξοδο έχουμε:

- Για **ψηφιακή είσοδο**, χρησιμοποιούμε τις 54 ψηφιακές 0..53. Όταν δουλεύουν ψηφιακά, η είσοδος μπορεί να είναι ή 0 ή 5V, με τον χαρακτηρισμό LOW ή HIGH όπως θα δούμε παρακάτω.
- Για **ψηφιακή έξοδο**, χρησιμοποιούμε τις 54 ψηφιακές 0..53. Όταν δουλεύουν ψηφιακά, η έξοδος μπορεί να είναι 0 ή 5V, με τον χαρακτηρισμό LOW ή HIGH όπως θα δούμε παρακάτω.
- Για **αναλογική είσοδο**, δηλαδή να διαβάσουμε τιμές ρεύματος στο διάστημα 0 έως 5V, χρησιμοποιούμε τις έξι αναλογικές θύρες A0..A15.
- Για **αναλογική έξοδο**, μπορούμε να χρησιμοποιήσουμε τις PWM ψηφιακές θύρες, οι οποίες θα μας δώσουν ρεύμα εξόδου όποιας τιμή θέλουμε στο διάστημα από 0 έως 5V με την εντολή `analogWrite()`.

Γράφοντας κώδικα θα πρέπει να αρχικοποιήσουμε τις θύρες που χρησιμοποιούμε με τη συνάρτηση `pinMode()`, δηλαδή να δίνουμε την πληροφορία για όποιες χρησιμοποιήσουμε αν θα είναι για είσοδο ή για έξοδο.

Όταν χρησιμοποιείται η σειριακή οθόνη παρακολούθησης της επικοινωνίας με τον υπολογιστή, χρησιμοποιούνται τα pins 0 και 1 για αυτό, οπότε προτείνουμε να μην τα χρησιμοποιείτε στις εφαρμογές σας, εκτός αν αυτό είναι απαραίτητο.

Επίσης, στη θύρα 13 υπάρχει συνήθως συνδεδεμένο ήδη ένα Led πάνω στην πλακέτα Arduino Mega 2560, κι έτσι μπορούμε να το χρησιμοποιούμε για σχετικές λειτουργίες.

### 3.3 Περιφεριακά στοιχεία της πειραματικής διάταξης

#### 3.3.1 Εισαγωγή

Τα Arduino μπορούν να χρησιμοποιούν την τεχνολογία των shields. Τα shields είναι τυπωμένες πλακέτες επεκτάσεων κυκλωμάτων που συνδέονται στα κανονικά παρεχόμενα των ακροδεκτών του Arduino. Τα shields μπορούν να παρέχουν έλεγχο κινητήρων, GPS, Internet μέσω Ethernet αλλά και μέσω Wifi, LCD εικόνες. Πέρα όμως από αυτά μπορούμε να τοποθετήσουμε και αρκετά στοιχεία όπως φωτοδιόδους, κουμπιά αλλά και απλές αντιστάσεις και πυκνωτές ανάλογα πάντα με την εφαρμογή που θέλουμε να κατασκευάσουμε.

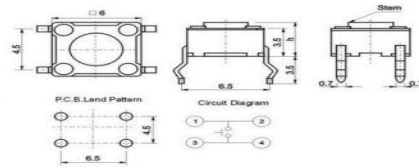
#### 3.3.2 Κουμπιά -Buttons

Στα κυκλώματά μας θα χρειαστεί πολλές φορές να χρησιμοποιήσουμε κουμπιά ώστε να μπορούμε να παρέμβουμε στο κύκλωμα όταν εμείς το θέλουμε, π.χ. για να δώσουμε ρεύμα σε να τμήμα του ή να διακόψουμε την τροφοδότηση με ρεύμα ενός άλλου τμήματος.



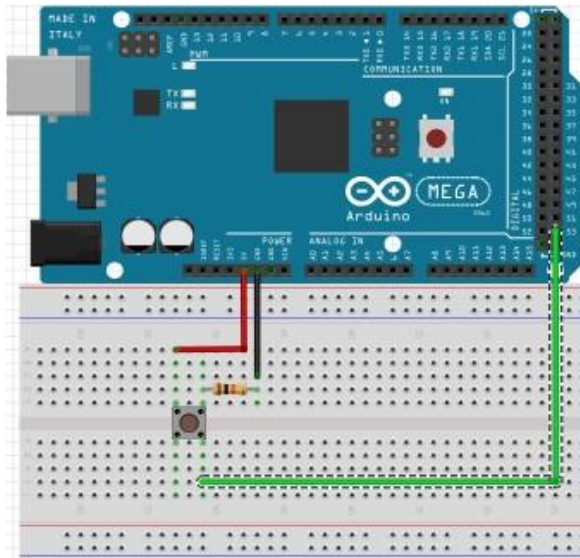
Εικόνα 29:

Πως απεικονίζεται το κουμπί σε παραδείγματα

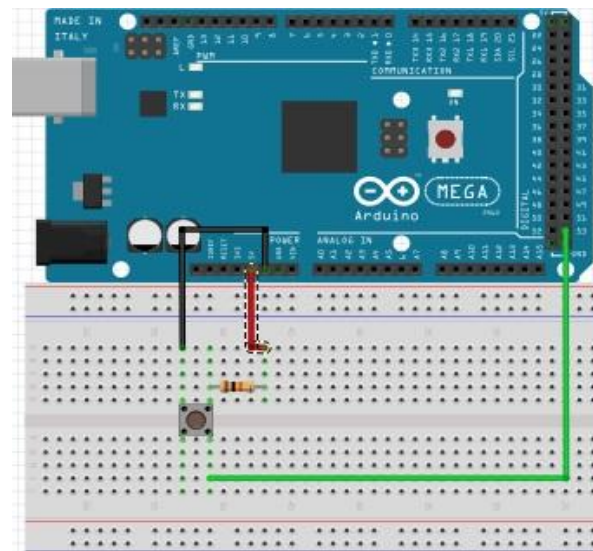


Εικόνα 30: Τα κουμπιά που θα χρησιμοποιήσουμε

Στα σχήματα που ακολουθούν μπορείτε να δείτε δύο κυκλώματα με κουμπιά. Το ένα αφήνει το ρεύμα να περνάει όταν πατηθεί το κουμπί (δηλαδή είναι πάντα στο LOW το input pin κι όταν πατηθεί το πλήκτρο γίνεται HIGH) – αυτό ονομάζεται pull-down resistor και το βλέπουμε στην εικόνα 30



Εικόνα 31: Με ενεργοποίηση με το πάτημα



Εικόνα 32: Απενεργοποίηση με το πάτημα

ενώ το άλλο διακόπτει τη ροή του ρεύματος όταν πατηθεί (δηλαδή είναι πάντα στο HIGH το input pin κι όταν πατηθεί το πλήκτρο γίνεται LOW) – αυτό ονομάζεται pull-up resistor και το βλέπουμε στην εικόνα 31 αντίστοιχα. Υπάρχει ένας κανόνας (Kirchoff) που λέει ότι το ρεύμα θα ακολουθήσει τη διαδρομή με τη μικρότερη αντίσταση. Το κουμπί έχει μια μικρή εσωτερική αντίσταση μέσα του. Έτσι, στο κύκλωμα της εικόνας 12 (pull-up) όταν πατηθεί το κουμπί και κλείσει το κύκλωμα, έχουμε ροή από τα 5V προς τη γείωση (μικρότερη αντίσταση), οπότε διακοπή του αρχικού κυκλώματος (5V – Input). Στο κύκλωμα της εικόνας 11 (pull-down), όταν πατηθεί το κουμπί, τότε έχουμε ροή από την πηγή προς την είσοδο (μικρότερη αντίσταση), δηλαδή ενεργοποίηση του κυκλώματος (5V – Input). Ανάλογα λοιπόν τί θέλουμε να κάνουμε στα κυκλώματά μας, ακολουθούμε την αντίστοιχη συνδεσμολογία.

### 3.3.3 Ποτενσιόμετρο

Ένα ποτενσιόμετρο μπορεί να περιγραφεί ως μια διάταξη με κύκλωμα μεταβλητής αντίστασης. Γυρίζοντας το χειριστήριο που έχει, αυξάνει ή μειώνει την αντίστασή του, οπότε περνάει λιγότερο ή περισσότερο ρεύμα από αυτό .



Εικόνα 33:

Σχεδιασμένο ένα ποτενσιόμετρο

Έχει τρία σημεία σύνδεσης – ένα πόδι συνδέεται στην πηγή (5V), ένα στη γείωση (GND) και το μεσαίο στο κύκλωμα που θέλουμε να πάρει τη μεταβλητή τιμή ρεύματος (αναλογικό pin στο Arduino). Το ποτενσιόμετρο δηλαδή δουλεύει ως ένας διαιρέτης τάσης, ο οποίος όμως μεταβάλλεται κάθε φορά που γυρίζουμε το χειριστήριό του. Ποτενσιόμετρα χρησιμοποιούνται σε πολλές εφαρμογές, όπως στην αυξομείωση του ήχου στα στερεοφωνικά ή την αυξομείωση της έντασης μιας λάμπας. Στην δική μας πειραματική διάταξη χρησιμοποιούμε το ποτενσιόμετρο για την ρύθμιση της φωτεινότητας της LCD οθόνης.

### 3.3.4 Οθόνη LCD

Η βιβλιοθήκη LiquidCrystal μας επιτρέπει να ελέγχουμε τις οθόνες LCD που είναι συμβατές με το πρόγραμμα οδήγησης της Hitachi HD44780. Υπάρχουν πολλά είδη και μπορείτε να τις βρούμε συνήθως με τη διασύνδεση των 16 ακροδεκτών.



Εικόνα 34: Μια απλή LCD οθόνη του εμπορίου

Αυτό είναι ένα παράδειγμα όπου τυπώνει "Hello World!" στην οθόνη LCD και δείχνει το χρόνο σε δευτερόλεπτα από όταν το Arduino τέθηκε σε λειτουργία.

Οι οθόνες LCD έχουν μια παράλληλη διασύνδεση, πράγμα που σημαίνει ότι ο μικροελεγκτής έχει να χειριστεί πολλές ακίδες διασύνδεσης ταυτόχρονα για τον έλεγχο της οθόνης. Η διεπαφή αποτελείται από τις ακόλουθες ακίδες:

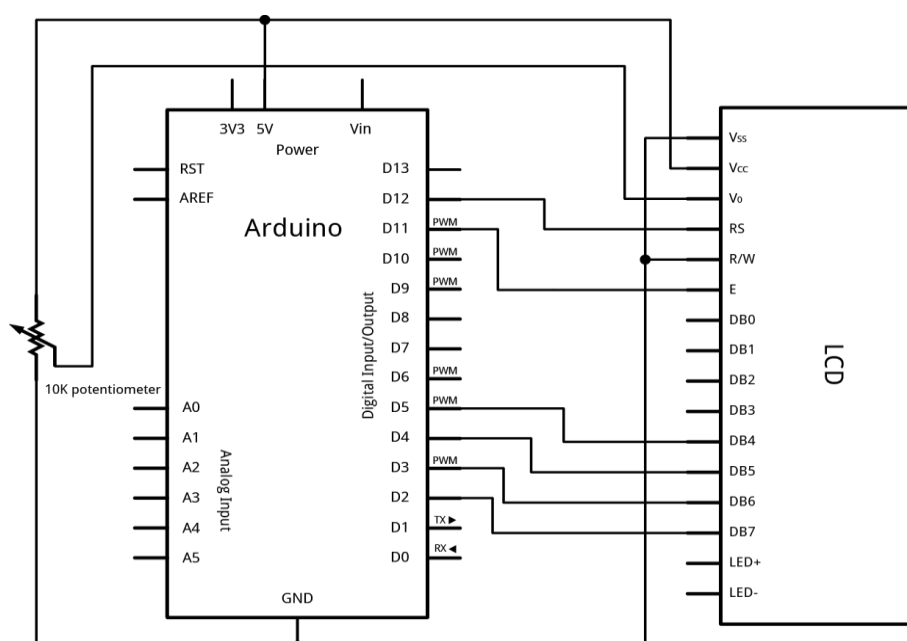
- Ένα καταχωρητή επιλογής (RS) ακροδέκτη που ελέγχει πού στη μνήμη της οθόνης LCD όπου γράφετε δεδομένα σε αυτόν. Μπορείτε να επιλέξετε είτε τον καταχωρητή δεδομένων, ο οποίος κατέχει το τι στέλνετε στην οθόνη, ή ένα καταχωρητή εντολών, ο



οποίος είναι αυτός όπου ο ελεγκτής της οθόνης LCD αναζητά οδηγίες για το τι θα κάνουμε στη συνέχεια.

- Ένα ανάγνωσης / εγγραφής (R / W) ακροδέκτη που επιλέγει την λειτουργία ανάμεσα σε ανάγνωσης ή γραφής
- Ένα PIN ενεργοποίησης όπου επιτρέπει την εγγραφή στο μητρώο
- 8 ακροδέκτες δεδομένων (D0 -D7). Οι καταστάσεις σε αυτούς τους ακροδέκτες (0 ή 5 Volt) είναι τα bits που γράφετε στο μητρώο όταν γράφετε, ή με τις τιμές που διαβάζετε σε κατάσταση “διαβάσματος”..
- Τέλος άλλους δύο ακροδέκτες για την αντίθεση στον φωτισμό της οθόνης.

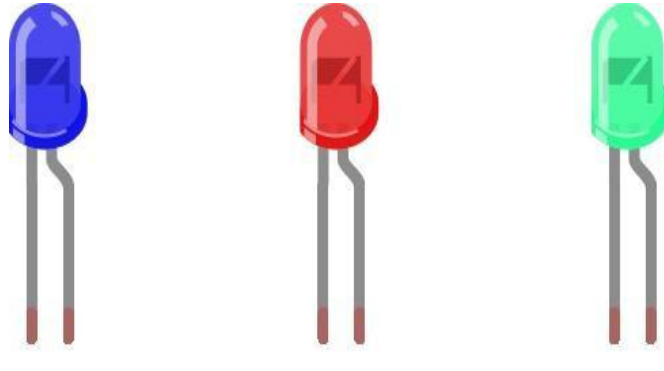
Στο εμπόριο μπορούμε να βρούμε διάφορες LCD οθόνες είτε ως προς το πίσω χρώμα (για παράδειγμα αυτή που θα χρησιμοποιήσουμε είναι χρώματος μπλε) αλλά και ως προς το μέγεθός της , μπορούμε να βρούμε οθόνες από μία σειρά μέχρι πολλές περισσότερες.



Εικόνα 35: Σχηματικό διάγραμμα σύνδεσης LCD οθόνης με το arduino

### 3.3.5 Φωτοдиодοι - LED

Οι φωτοдиодοι, ή τα leds όπως έχουν κυριαρχήσει, υπάρχουν σε διάφορα χρώματα και τάσεις λειτουργίας . Ανάλογα με την προβλεπόμενη τάση λειτουργίας του led που έχουμε πρέπει να χρησιμοποιήσουμε μαζί και την κατάλληλη αντίσταση, ώστε να αποφύγουμε κάποια καταστροφή από υπέρταση. Για να βρούμε την αντίσταση που χρειαζόμαστε αρκεί να θυμηθούμε τον τύπο  $R = (V\pi - V\lambda) / I$  όπου R είναι η αντίσταση που χρειαζόμαστε,  $V\pi$  η παρεχόμενη τάση από την πηγή μας (5V από το Arduino),  $V\lambda$  η τάση λειτουργίας του led και I το ρεύμα λειτουργίας του led. Τυπικά, μια αντίσταση γύρω στα 220 Ω καλύπτει τα περισσότερα led που έχουμε χρησιμοποιήσει.



Εικόνα 36: Leds σε διάφορα χρώματα

Επίσης, τα leds έχουν πολικότητα, δηλαδή δουλεύουν μόνο αν συνδεθούν στην κατάλληλη φορά ρεύματος. Συνήθως το πόδι που πρέπει να συνδεθεί στη θετική κατεύθυνση (+) είναι πιο μακρύ από το αντίστοιχο για την αρνητική φορά (-). Επίσης, το λαμπάκι από την αρνητική μεριά (-) είναι συνήθως επίπεδο κι όχι στρογγυλό όπως είναι από το άλλο πόδι (+). Μην ανησυχείτε όμως, μια ανάποδη σύνδεση δεν θα το καταστρέψει, απλά δεν θα ανάψει στην ανάποδη φορά (εκτός από την ιδιαίτερη περίπτωση να έχετε δώσει πολύ μεγάλη τάση).

### 3.3.6 Ethernet Shield

Το Arduino Ethernet Shield επιτρέπει σε μια πλακέτα Arduino για να συνδεθείτε στο διαδίκτυο. Βασίζεται στην αρχιτεκτονική Wiznet W5100 Ethernet chip. Η αρχιτεκτονική Wiznet W5100 παρέχει ένα διαδίκτυο (IP) συμβατό τόσο με TCP όσο και με UDP πρότυπα.

Υποστηρίζει έως και τέσσερις ταυτόχρονες συνδέσεις. Για να την λειτουργήσουμε πρέπει να προσθέσουμε τη βιβλιοθήκη Ethernet στο πρόγραμμά μας για να συνδεθούμε στο Internet χρησιμοποιώντας την Ethernet Shield. Η Ethernet Shield συνδέεται σε μια πλακέτα Arduino χρησιμοποιώντας τους αρσενικούς ακροδέκτες που εκτείνονται από μέσα προς τα κάτω του εξαρτήματος. Αυτό κρατά τη διάταξη άθικτη και επιτρέπει σε μια άλλη Shield να τοποθετηθεί στην κορυφή ή για να συνδέσουμε το υπόλοιπο κύκλωμά μας.

Το Ethernet Shield έχει ένα πρότυπο RJ-45 σύνδεσης, με ενσωματωμένο μετασχηματιστή γραμμής και την λειτουργία Power over Ethernet ενεργοποιημένη. Υπάρχει μια ενσωματωμένη υποδοχή κάρτας micro-SD, την οποία μπορούμε να χρησιμοποιήσουμε για την αποθήκευση αρχείων για την λειτουργία μέσω του δικτύου. Είναι συμβατό με το Arduino Uno και Mega (με τη χρήση της βιβλιοθήκης Ethernet). Ο ενσωματωμένος αναγνώστης καρτών microSD για να γίνει λειτουργικός πρέπει να προσθέσουμε τις βιβλιοθήκες για SD



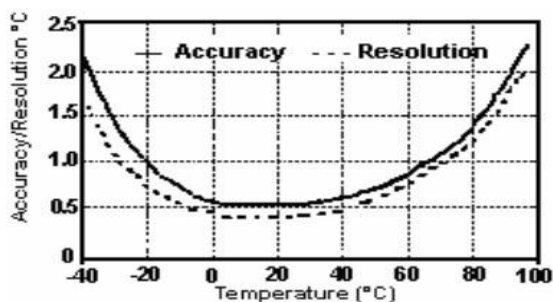
Εικόνα 37: Το επίσημο Ethernet Shield του Arduino

### 3.3.7 Αισθητήρας θερμοκρασίας

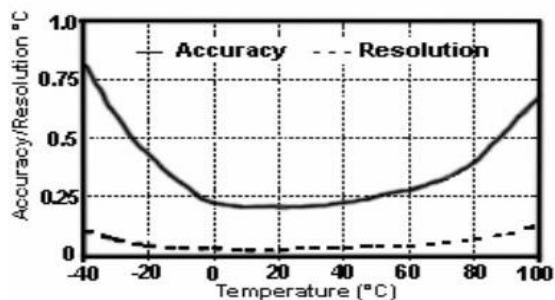
Για τον έλεγχο της θερμοκρασίας δεν χρησιμοποιήσαμε έναν απλό αισθητήρα θερμοκρασίας ή κάποιο θερμίστορ ή κάποιο θερμικό αναλογικό αισθητήρα ή ένα σειριακό αισθητήρα αλλά χρησιμοποιήσαμε ένα αισθητήρα θερμοκρασίας της σειράς Hobo. Πιο συγκεκριμένα χρησιμοποιήσαμε τον αισθητήρα TMC20HD ο οποίος είναι για χρήση στον αέρα, νερό και το έδαφος. Ο λόγος που τον χρησιμοποιήθηκε ο συγκεκριμένος αισθητήρας ήταν λόγω της κατοχής του από το εργαστήριο. Οποιοδήποτε άλλος αισθητήρας με τάση εξόδου έως 5 Volt θα μπορούσε να χρησιμοποιηθεί.

Τα χαρακτηριστικά του αισθητήρα είναι :

- Όρια μέτρησης:  $-40^{\circ}$  με  $50^{\circ}\text{C}$  ( $-40^{\circ}$  με  $122^{\circ}\text{F}$ ) στο νερό,  $-40^{\circ}$  με  $100^{\circ}\text{C}$  ( $-40^{\circ}$  με  $212^{\circ}\text{F}$ ) στον αέρα
- Ακρίβεια με καταγραφέα δεδομένων H8:  $\pm 0.5^{\circ}$  στους  $20^{\circ}\text{C}$  ( $\pm 0.9^{\circ}$  στους  $68^{\circ}\text{F}$ ) Δείτε το σχεδιάγραμμα Α
- Ακρίβεια με καταγραφέα δεδομένων U12:  $\pm 0.25^{\circ}$  στους  $20^{\circ}\text{C}$  ( $\pm 0.45^{\circ}$  στους  $68^{\circ}\text{F}$ ) Δείτε το σχεδιάγραμμα Β
- Απόκλιση :  $< 0.1^{\circ}\text{C}$  ( $< 0.2^{\circ}\text{F}$ ) ανά χρόνο λειτουργίας
- Χρόνος απόκρισης σε αέρα: 3 λεπτά για τιμή 90% της πραγματικής με ταχύτητα ανέμου 1 m/sec (2.2 mph)
- Χρόνος απόκρισης μέσα σε νερό : 30 δευτερόλεπτα για τιμή 90% της πραγματικής
- Χρόνος καλής λειτουργίας: το άκρο του αισθητήρα και το καλώδιο σε καθαρό νερό μέχρι  $50^{\circ}\text{C}$  ( $122^{\circ}\text{F}$ ) για ένα χρόνο
- Διαστάσεις αισθητήρα: 0.5 cm x 2.5 cm (0.200" x 1.0")
- Βάρος: 37 g (1.3 oz), 125 g (4.4 oz), ή 280 g (9.9 oz) ανάλογα το μήκος του καλωδίου



Σχεδιάγραμμα Α



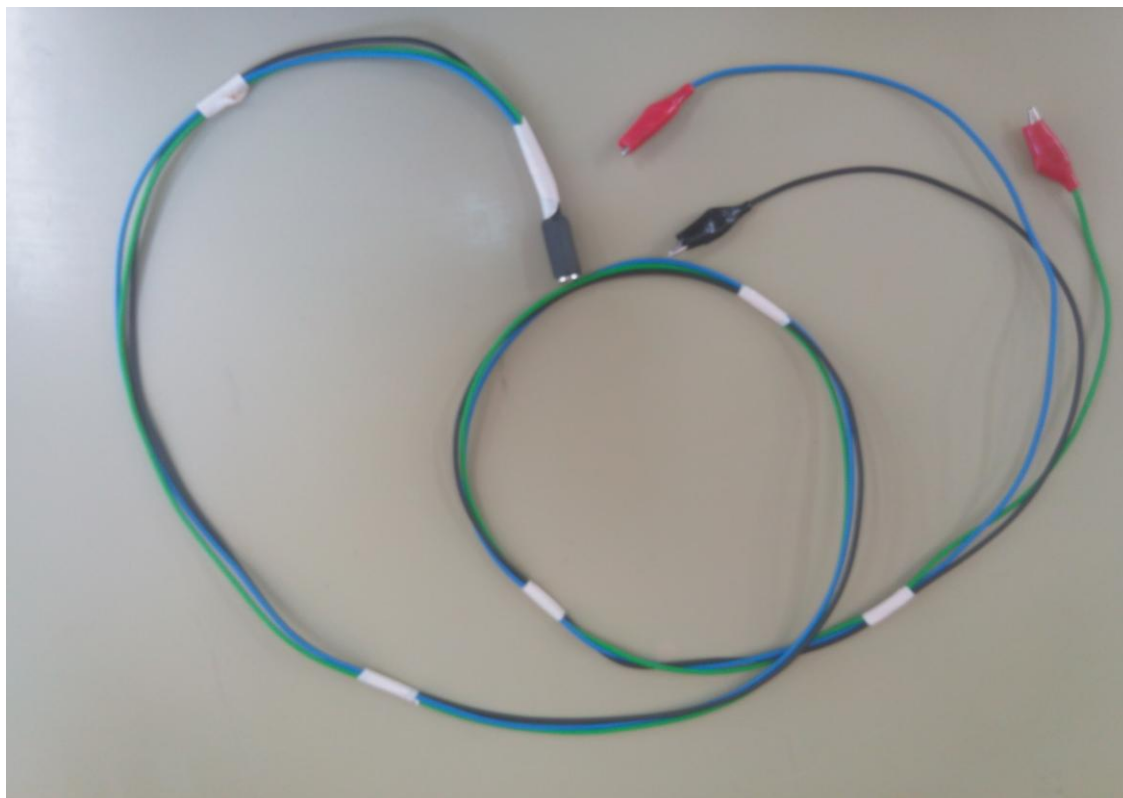
Σχεδιάγραμμα Β

Εικόνα 38: Σχεδιαγράμματα ακρίβειας ανάλογα τον δειγματολήπτη



Εικόνα 39: Ο αισθητήρας που χρησιμοποιήθηκε

Λόγο του ότι ο αισθητήρας έχει βύσμα jack 2,5mm και το arduino δεν έχει τέτοια υποδοχή και έχει μόνο ψηφιακούς και αναλογικούς ακροδέκτες έπρεπε να φτιάξουμε το δικό μας καλώδιο όπου θα περιλαμβάνει από την μια άκρη θηλυκό ακροδέκτη 2,5mm και από την άλλη τρεις ακροδέκτες τύπου δαγκάνας ώστε να έχουμε μέγιστη δυνατή επαφή και αγωγιμότητα με τα καλώδια που είναι συνδεδεμένα με το arduino .



*Εικόνα 40: Το καλώδιο όπου κατασκευάσαμε*

Τέλος το Arduino δεν μπορεί να διαβάσει κατευθείαν την θερμοκρασία από τον αισθητήρα διότι αυτό που βλέπει είναι μια τιμή τάσης που διαβάζεται από της αναλογικές εισόδους του Arduino (A0 έως A15) . Εφόσον η τάση εξόδου του αισθητήρα δεν έχει γραμμική σχέση με την θερμοκρασία, πρέπει να βρεθεί η σχέση που διέπει αυτές τις δύο μεταβλητές .

Πραγματοποιήθηκε μία πειραματική διάταξη στην οποία μετριόταν η θερμοκρασία νερού από τους 0°C έως τους 80°C μέγιστη μετρούμενη θερμοκρασία του αισθητήρα. Ταυτόχρονα πάρθηκαν οι τιμές της τάσης εξόδου του αισθητήρα, όπου και φτιάχτηκε ο παρακάτω πίνακας (εικόνα 43) :



Εικόνα 41: Ο αισθητήρας θερμοκρασίας με το HOBO

Με βάση αυτές τις μετρήσεις του παραπάνω πίνακα και χρησιμοποιώντας το πρόγραμμα υπολογισμού συναρτήσεων (origin) μπορέσαμε να βρούμε την σχέση που διέπει αυτές τις δύο μεταβλητές. Η συνάρτηση είναι η εξής :

Θερμοκρασία	Τιμή τάσης
1.8	749
2.6	741
8.5	684
10	666
19.5	566
25	510
30	456
32	439
34	420
36	404
38	385
40	364
41	361
42	348
44	332
46	316
48	300
50	284
52	271
53.5	259
56	242
58	231
60	219
62	207
64.5	195
65	187
66	184
68.5	179
74	149
79	132

Εικόνα 42: Πίνακας που δημιουργήθηκε από την δειγματοληψία του πειράματος

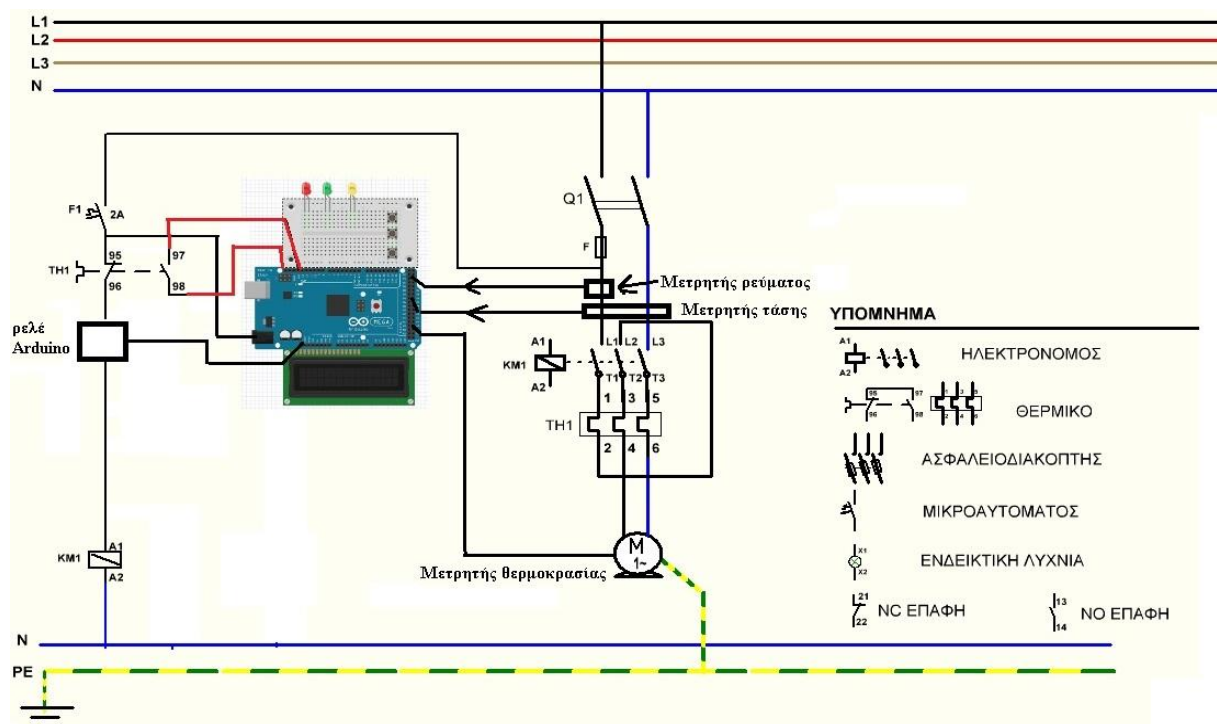
$$temp=0.000000000529387*x^4+0.00000118191*x^3+0.000979104*x^2-0.4556*x+123.9809$$

Όπου στην παραπάνω εξίσωση , temp είναι η τελική τιμή της θερμοκρασίας ενώ x είναι η τιμή τάσης που λαμβάνει το Arduino. Αξίζει να σημειώσουμε πως ο βαθμός του πολυωνύμου είναι ανάλογος με την ακρίβεια στο τελικό αποτέλεσμα , οπότε όσο μεγαλύτερος ο βαθμός του πολυωνύμου τόσο καλύτερη ακρίβεια έχουμε στην λήψη της θερμοκρασίας .Στην δική μας περίπτωση βρέθηκε ότι μας δίνει ικανοποιητική ακρίβεια πολυώνυμο 4ο βαθμού.

## Κεφάλαιο 4ο

### 4.1 Ηλεκτρολογικό σχέδιο πειραματικής διάταξης

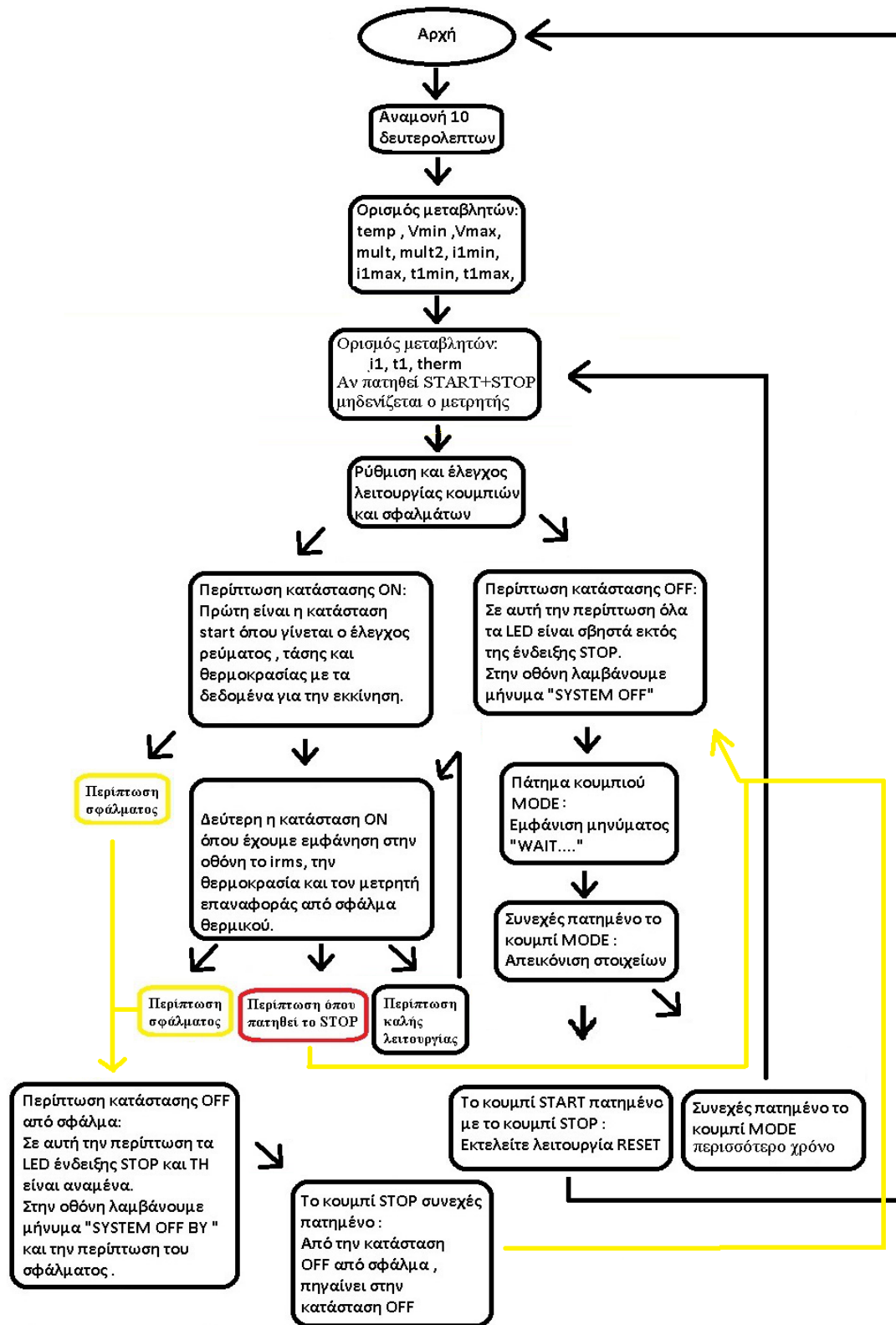
Στο παρακάτω σχήμα φαίνονται το κύκλωμα ισχύος, το βοηθητικό κύκλωμα του ηλεκτρονόμου στο οποίο συμπεριλαμβάνονται το Arduino με τα περιφερειακά του στοιχεία. Το κύκλωμα ισχύος προστατεύεται από βραχυκυκλώματα με την ασφάλεια F. Το κύκλωμα επίσης προστατεύεται από υπερφορτίσεις μέσω του θερμικού TH1 το οποίο είναι σε σειρά με το ρελέ του Arduino ως επιπρόσθετο μέρος προστασίας για την περίπτωση αστοχίας του arduino. Τέλος το Arduino μέσω των μετρητών τάσης, ρεύματος και θερμοκρασίας καθώς και με την κανονικά ανοιχτή επαφή του θερμικού 97-98 προστατεύει το κύκλωμα από υπερτάσεις και υποτάσεις, υπερθέρμανση του φορτίου (πχ κινητήρα) και υπερφορτίσεις μέσω του μετρητή ρεύματος και της κανονικά ανοιχτής επαφής του θερμικού. Με τον τρόπο αυτό προσφέρεται μεγαλύτερη προστασία από υπερφορτίσεις σε σχέση με τον κλασικό αυτοματισμό. Το κόστος που θα δαπανηθεί για την αγορά του μικροεπεξεργαστή Arduino και των περιφερειακών του είναι σχετικά χαμηλό και αντισταθμίζεται για την την προστασία που παρέχει.



Εικόνα 43: Ηλεκτρολογικό κύκλωμα ελέγχου ηλεκτρονόμου από Arduino



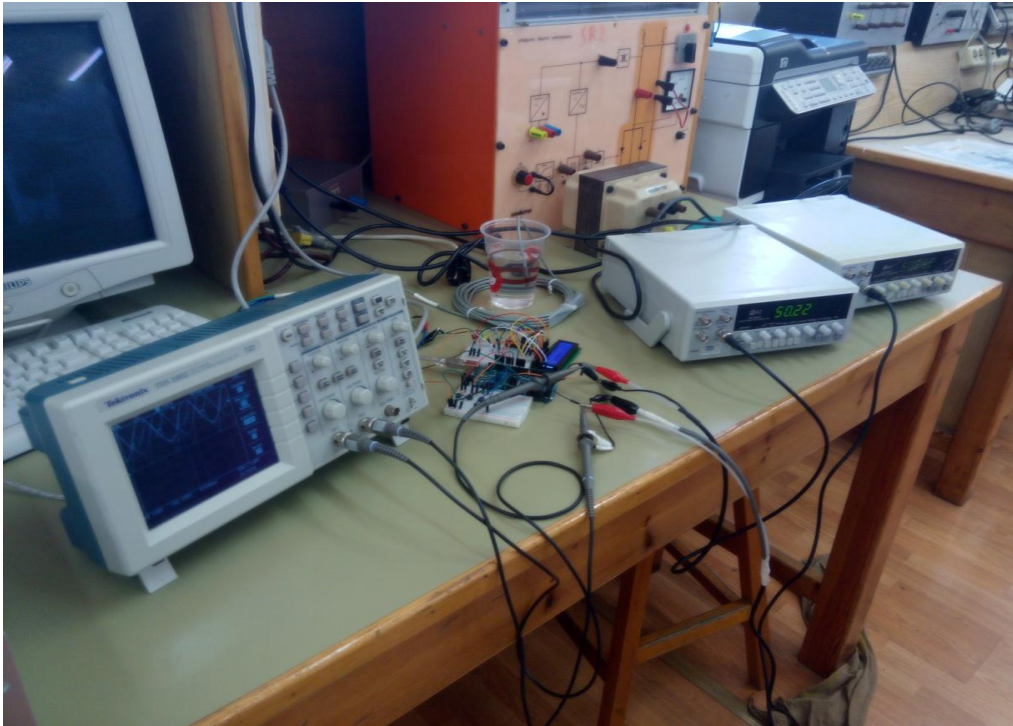
## 4.2 Διάγραμμα Ροής



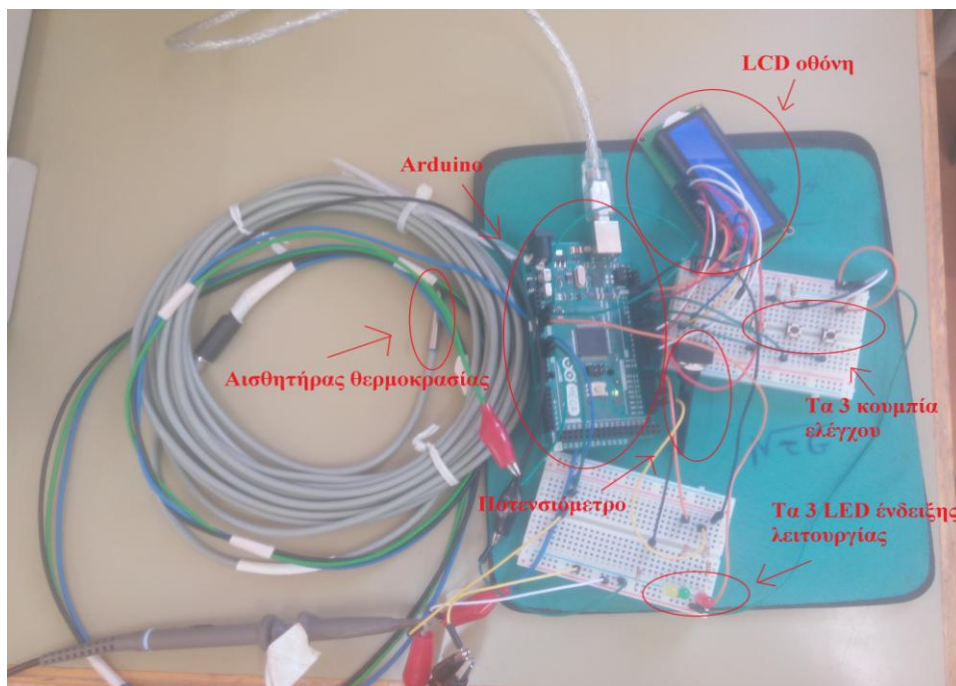
Εικόνα 44: Το διάγραμμα ροής του προγράμματός μας

### 4.3. Πειραματική διαδικασία

Στις παρακάτω εικόνες παρουσιάζεται το Arduino με όλα τα περιφερειακά στοιχεία του. Φαίνονται ο μικροελεγκτής Arduino , η οθόνη LCD , ο αισθητήρας θερμοκρασίας , τα μπουτόν και τις φωτοδιόδους(LED) καθώς επίσης και δύο γεννήτριες συχνοτήτων οι οποίες εξομοιώνουν αισθητήρες τάσης και έντασης .Φαίνεται επίσης και ο παλμογράφος όπου οπτικοποιούσε τα δύο σήματα .



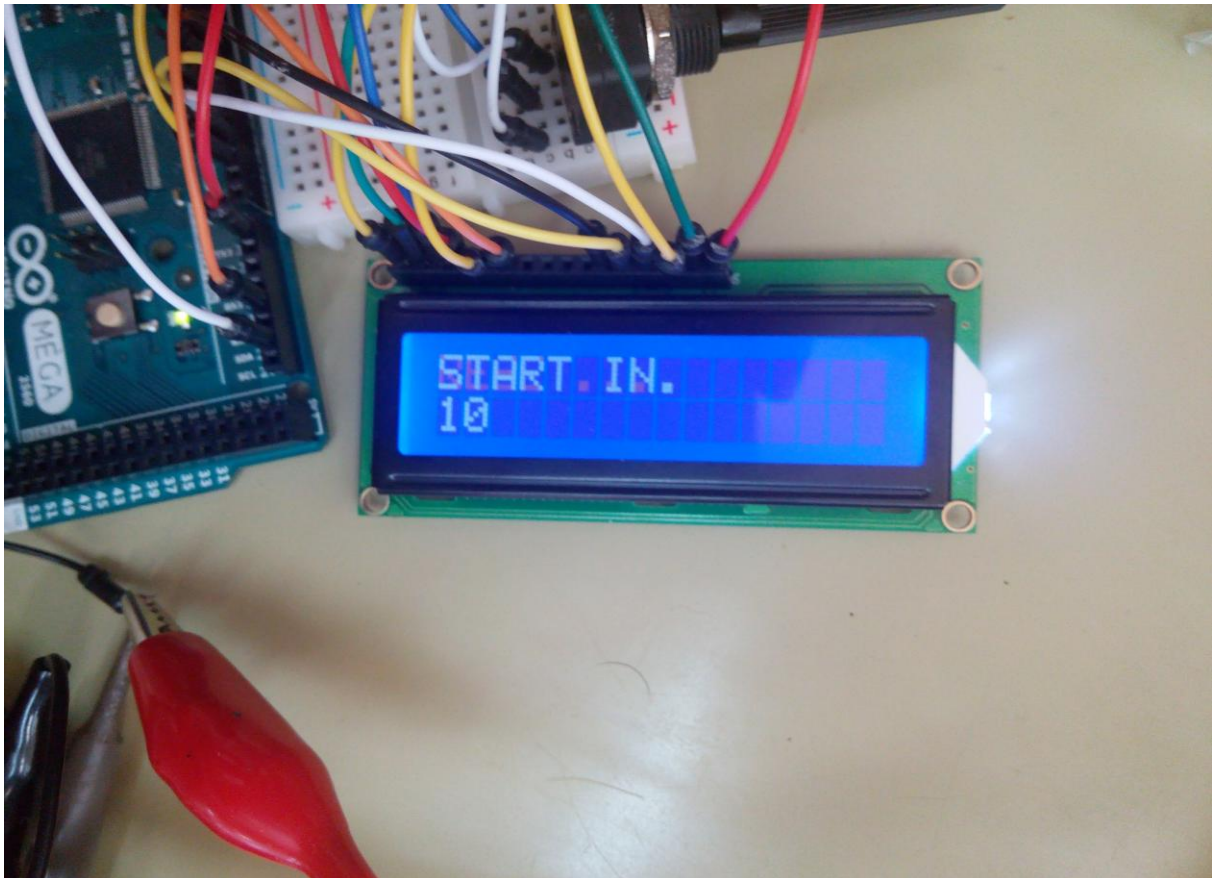
Εικόνα 45: Η υλοποιημένη πειραματική διάταξη



Εικόνα 46: Υλοποιημένη η διάταξη με σύμανση για το κάθε αντικείμενο

### 4.3.1 Διαδικασία πειράματος

Όπως είδαμε από το παραπάνω διάγραμμα ροής , με την εκκίνηση του προγράμματος υπάρχει ένας χρόνος δέκα δευτερολέπτων. Ο χρόνος αυτός είναι απαραίτητος για την ομαλοποίηση της τάσης σε περίπτωση επανατροφοδότησης του δικτύου.



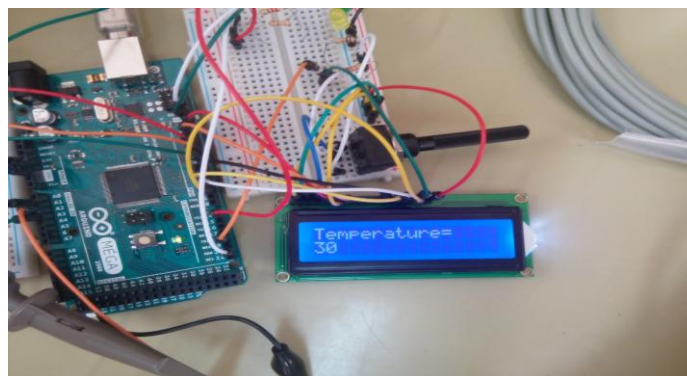
Εικόνα 47: Το αντίστοιχο μήνυμα εκτυπωμένο στην οθόνη μας

Το επόμενο στάδιο είναι να ορίσουμε (σαν εξειδικευμένοι εργάτες) κάποιες αρχικές ρυθμίσεις τις οποίες ο τελικός χρήστης δεν θα μπορεί να τις αλλάξει .

ΣΗΜΕΙΩΣΗ: αυτές οι ρυθμίσεις μπορούν να αλλάξουν μόνο στις εξής περιπτώσεις :

- αν θέσουμε το Arduino εκτός τροφοδοσίας και το επαναφέρουμε
- από συνδυασμό σε συγκεκριμένο σημείο του προγράμματος

Η πρώτη τιμή που βάζουμε είναι η μέγιστη επιτρεπόμενη τιμή της θερμοκρασίας του φορτίου.



Εικόνα 48: Εισαγωγή παραμέτρου θερμοκρασίας

Στην συνέχεια ορίζουμε την ελάχιστη επιτρεπόμενη τάση λειτουργίας.

Στο συμβόλαιο όπου συνάπτει ο καθένας μας με την Δημόσια Επιχείρηση Ηλεκτρισμού( ΔΕΗ εν συντομία), η ΔΕΗ δεσμεύεται ότι θα μας παρέχει με τάση με διακύμανση 5% στα 230 Volt . Παρόλα αυτά σε περιπτώσεις σφάλματος ή διακοπής ή εκκινήσεις μεγάλων κινητήρων η τάση μπορεί να είναι πέσει από αυτήν την τιμή. Αυτός είναι ο λόγος για τον οποίο θέτουμε αυτή την δικλίδα ασφαλείας. Στον κλασικό αυτοματισμό το όργανο που αναλαμβάνει αυτή την προστασία είναι ο ελεγκτής τάσης.

Ένας δεύτερος λόγος τον οποίο θέτουμε αυτήν την ασφάλεια είναι διότι μπορεί λόγο το φορτίο μας τεχνικών χαρακτηριστικών να παρουσιάζει προβλήματα σε μικρότερη ή μεγαλύτερη τιμή του 5% που προαναφέραμε όποτε σε αυτό το σημείο ο εργάτης ορίζει ανάλογα .



*Εικόνα 49: Ορισμός ελάχιστης τάσης στα 218 Volt*

Η επόμενη ρύθμιση που θα γίνει είναι για την μέγιστη επιτρεπόμενη τάση. Οι λόγοι της ρύθμισης αυτής είναι οι ίδιες με την προηγούμενη ρύθμιση αλλά ελέγχουμε το ανώτερο όριο της τάσης .



*Εικόνα 50: Ορισμός μέγιστης τάσης στα 242Volt*

Κατά την εκκίνηση του κινητήρα ο κινητήρας θα καταναλώσει μεγαλύτερο ρεύμα από ότι σε κανονική λειτουργία. Συνήθως η τιμή αυτή αγγίζει τις 8 φορές του ονομαστικού ρεύματος. Στα παθητικά στοιχεία δεν έχουμε τέτοια φαινόμενα. Για αυτό τον λόγο υπάρχει η επόμενη ρύθμιση όπου ο εργάτης ρυθμίζει το πολλαπλασιαστή του ρεύματος κατά την εκκίνηση. Η τιμή πρέπει να ρυθμιστεί στο 1 για τα παθητικά στοιχεία και ανάλογα τον κινητήρα μπορεί να πάρει οποιαδήποτε τιμή άνω την μονάδος. Συνήθως η τιμή κυμαίνεται μεταξύ 8 με 10.



*Εικόνα 51: Ρύθμιση πολλαπλασιαστή εκκίνησης στην τιμή 10*

Αντίστοιχα και στην κατάσταση λειτουργίας λόγω του γεγονότος ότι για κάποιο λόγο μπορεί η τιμή του ονομαστικού ρεύματος για κάποια αιτία να ξεπεραστεί, έχουμε την επόμενη ρύθμιση όπου ο εργάτης ρυθμίζει πόσες φορές πάνω από το ονομαστικό ρεύμα το φορτίο μας μπορεί να καταναλώσει έχοντας ως επιπρόσθετο μέσο προστασίας την θερμοκρασία του φορτίου.



*Εικόνα 52: Ρύθμιση πολλαπλασιαστή κανονικής λειτουργίας στην τιμή 1*

Η επόμενη ρύθμιση που ακολουθεί είναι για την τιμή του ελάχιστου ονομαστικού ρεύματος που μπορεί αργότερα ο χρήστης να ρυθμίσει για το φορτίο του.



*Εικόνα 53: Ρύθμιση της ελάχιστης επιτρεπόμενης τιμής του ρεύματος προστασίας στα 10 A*

Η επόμενη ρύθμιση που θα κάνουμε έρχεται να καλύψει το ανώτερο όριο της τιμής του ρεύματος που ο χρήστης θα μπορεί να βάλει στο σύστημα .



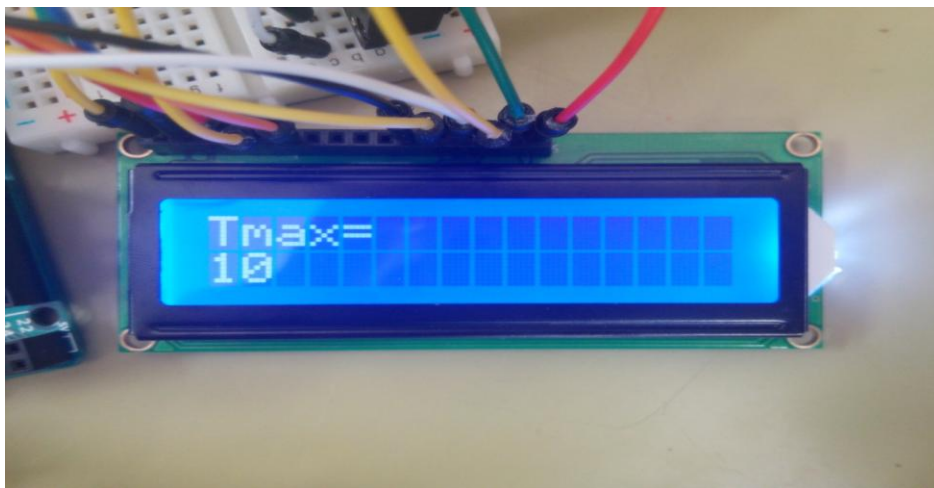
*Εικόνα 54: Ρύθμιση της μέγιστης επιτρεπόμενης τιμής του ρεύματος προστασίας στα 15 A*

Όπως με ρεύμα, έτσι και με τον χρόνο εκκίνησης, εργάτης ο οποίος θα κάνει την εγκατάσταση θα ρυθμίσει τα όρια μέσα στα οποία ο χρήστης θα μπορεί να ρυθμίζει τον χρόνο εκκίνησης του φορτίου. Πρώτα θα γίνει η ρύθμιση για το ελάχιστο όριο της τιμής αυτής.



*Εικόνα 55: Ρύθμιση της ελάχιστης επιτρεπόμενης τιμής του χρόνου εκκίνησης στα 0 sec*

Ακολουθεί αντίστοιχα και η τιμή για το μέγιστο χρόνο εκκίνησης που θα οριστεί από τον εργάτη για τον χρήστη.



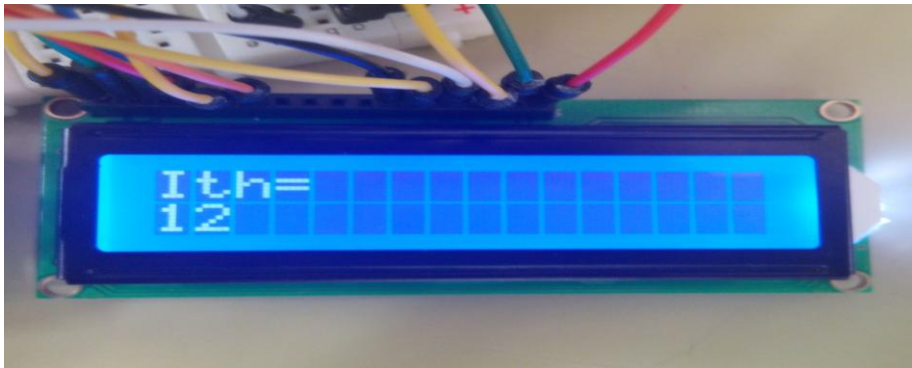
*Εικόνα 56: Ρύθμιση της μέγιστης επιτρεπόμενης τιμής του χρόνου εκκίνησης στα 10 sec*

Με την εισαγωγή της τιμής της μέγιστης επιτρεπόμενης τιμής του χρόνου εκκίνησης η εισαγωγή των εξειδικευμένων ρυθμίσεων όπου πρέπει να γίνουν μόνο από εξειδικευμένο προσωπικό έχει ολοκληρωθεί. Θεωρούμε πως ρυθμίσεις όπως το ανώτερο όριο της θερμοκρασίας, μέγιστη και ελάχιστη επιτρεπόμενη τάση, τα όρια για το μέγιστο και ελάχιστο όριο των ρευμάτων και του χρόνου εκκίνησης και τέλος τους πολλαπλασιαστές εκκίνησης και κανονικής λειτουργίας είναι μέγιστης σημασίας και για αυτό θα πρέπει να τις κάνει μόνο ειδικευμένος εργάτης. Ένα λάθος σε μία από τις παραπάνω ρυθμίσεις είναι επικίνδυνη και μπορεί να οδηγήσει σε μη επιθυμητές καταστάσεις.

Στην συνέχεια γίνεται η εισαγωγή των μη εξειδικευμένων ρυθμίσεων. Τις ρυθμίσεις αυτές μπορεί να τις εισάγει ο οποιοσδήποτε χρήστης.

Η επόμενη ρύθμιση είναι του επιτρεπόμενου ρεύματος προστασίας, δηλαδή μέγιστου ρεύματος που θα επιτρέπεται να διαρρέει το φορτίο μας. Η τιμή του επιτρεπόμενου ρεύματος προστασίας ανάλογα την κατάσταση λειτουργίας (κανονική ή εκκίνησης) θα ρυθμίζεται

αυτόματα με τον αντίστοιχο πολλαπλασιαστή.

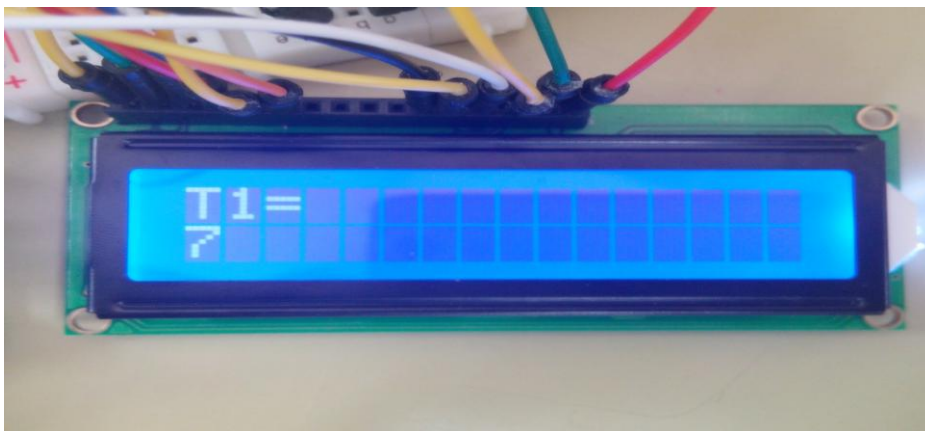


Εικόνα 57: Ρύθμιση της επιτρεπόμενης τιμής του ρεύματος προστασίας στα 12 A

Στην συνέχεια , η επόμενη ρύθμιση είναι η τιμή του χρόνου εκκίνησης. Ο χρόνος εκκίνησης είναι ο χρόνος εκείνος στον οποίο χρειάζεται ένας κινητήρας για να αποκτήσει τις ονομαστικές στροφές του.

Σημείωση :ο χρήστης δεν μπορεί να ξεπεράσει τα όρια που του έθεσε ο εργάτης πιο πριν.

Σημείωση :ο χρόνος που τοποθετούμε είναι σε δευτερόλεπτα.



Εικόν

α 58: Ρύθμιση του χρόνου εκκίνησης στα 7 sec

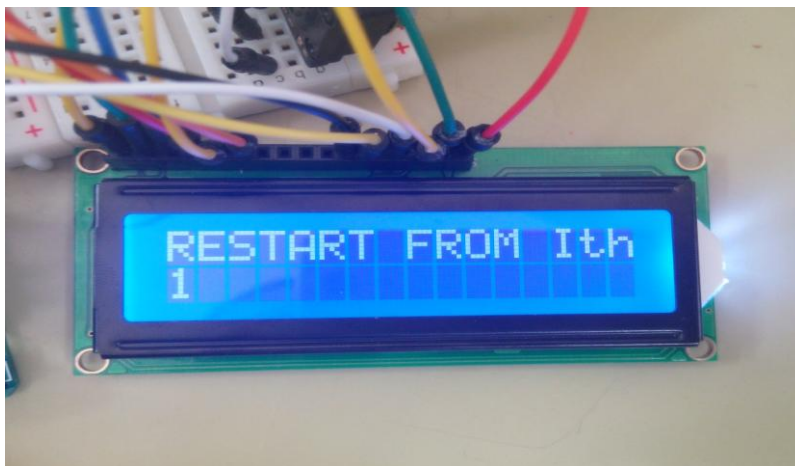
Τα θερμικά έχουν δύο λειτουργίες. Η λειτουργία της αυτόματης επαναφοράς και της μη αυτόματης επαναφοράς. Την κατάσταση αυτή εξομοιώνεται με την παρακάτω μεταβλητή. Ο χρήστης μπορεί να επιλέξει ανάμεσα στις επιλογές 0 και 1. Μηδέν για μη αυτόματη επαναφορά του συστήματος από θερμικό και 1 για την αυτόματη επαναφορά από θερμικό.





E

Εικόνα 59: Επιλογή μη αυτόματης επαναφοράς από θερμικό



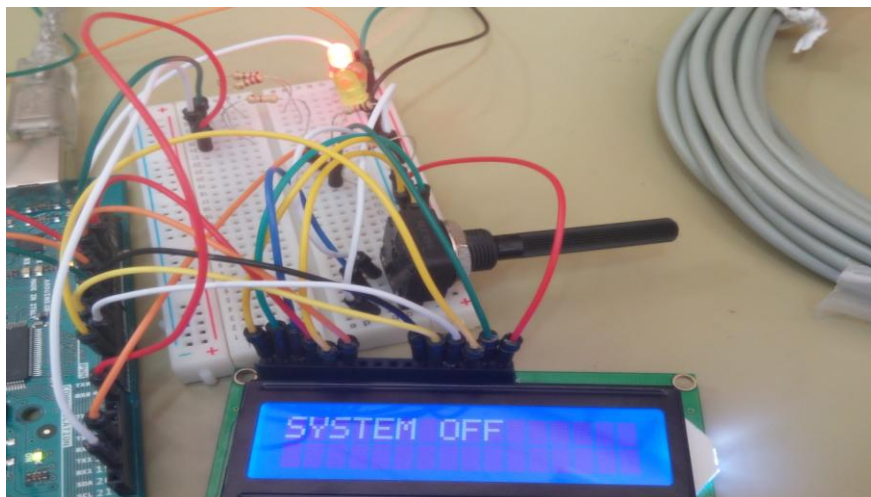
Εικόνα 60: Επιλογή αυτόματης επαναφοράς από θερμικό τιμή=1

Στο σημείο αυτό έχουμε την δυνατότητα αν πατηθούν ταυτόχρονα τα κουμπιά αύξησης και μείωσης των μεταβλητών να μηδενιστεί ο μετρητής σφαλμάτων από θερμικό.



*Εικόνα 61: Μηδενισμός του μετρητή σφαλμάτων από θερμικό*

Σε αυτό το σημείο φτάσαμε στο τέλος των ρυθμίσεων όπου ο arduino μπαίνει στην κατάσταση αναμονής και λαμβάνουμε το παρακάτω μήνυμα στην οθόνη μας



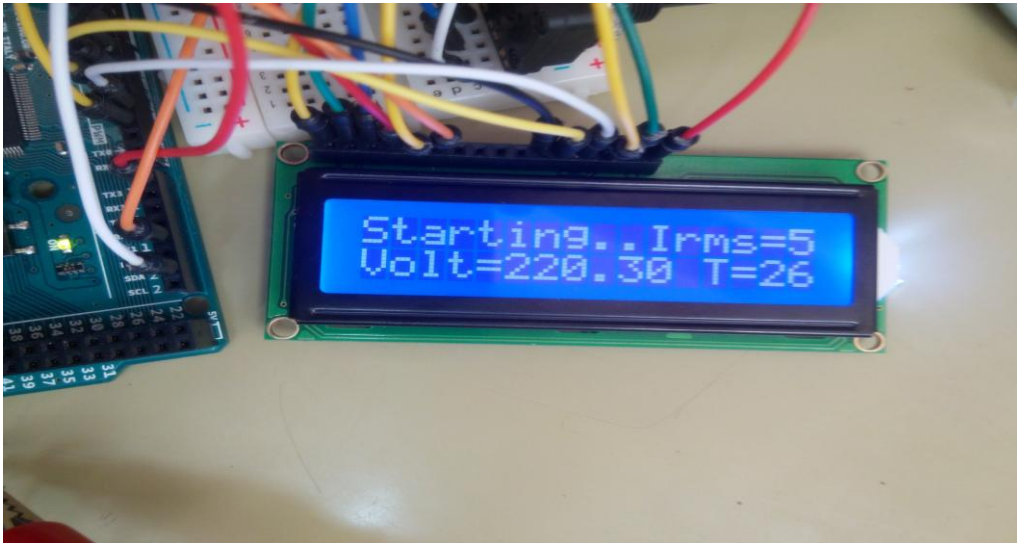
*Εικόνα 62: κατάσταση αναμονής*

Όπως βλέπουμε μας αναφέρει πως το σύστημα είναι σε κατάσταση κλεισίματος (αναμονής) και βλέπουμε την φωτοдиодο χρώματος κόκκινου και αντιστοιχεί στην κατάσταση “STOP” να είναι αναμμένο. Αυτή την στιγμή έχουμε 4 επιλογές.

- Επιλογή 1 : Να πατηθεί το κουμπί που αντιστοιχεί στο κουμπί επιλογής. Με αυτή την κίνηση βλέπουμε όλα τις μη εξειδικευμένες ρυθμίσεις.
- Επιλογή 2 : Να πατηθεί παρατεταμένα το κουμπί που αντιστοιχεί στο κουμπί επιλογής, επαναφερόμαστε στο σημείο όπου ο χρήστης ορίζει τις τιμές στο σύστημα.
- Επιλογή 3 : Να πατηθούν ταυτόχρονα τα κουμπιά START και STOP. Σε αυτή την περίπτωση στο πρόγραμμα γίνεται επανεκκίνηση.
- Επιλογή 4 : Να πατηθεί το κουμπί START.

Στην περίπτωση όπου πατηθεί το κουμπί START το σύστημα μπαίνει στην κατάσταση εκκίνησης. Η φωτοдиодος με το πράσινο χρώμα είναι αναμμένη, το ρελέ του Arduino σπλίζει

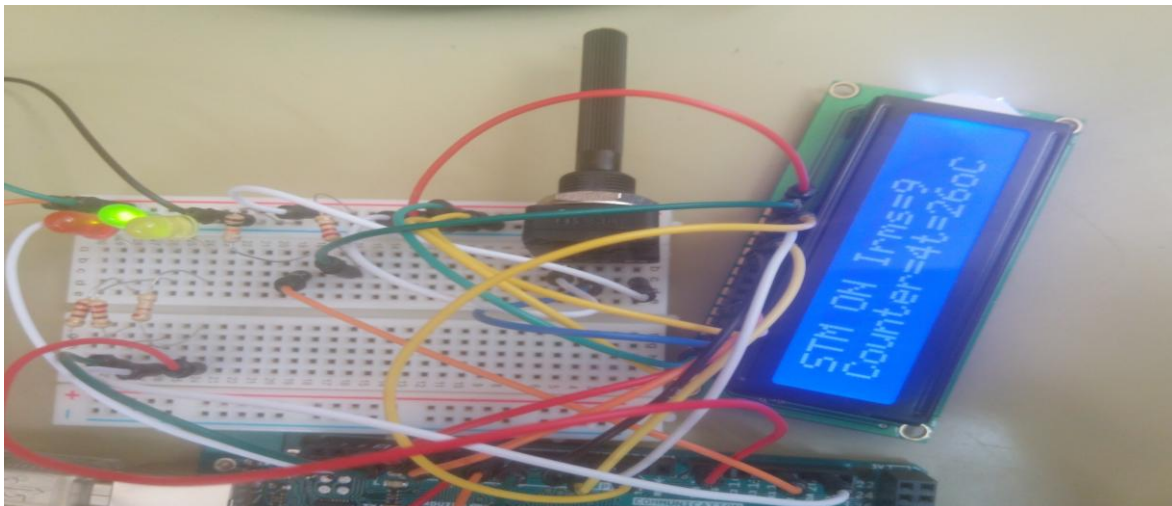
ενώ είναι σβηστές η κόκκινη και την κίτρινη φωτοδίοδος. Στην οθόνη λαμβάνουμε το αντίστοιχο μήνυμα .



*Εικόνα 63: Κατάσταση εκκίνησης.*

Στην οθόνη λαμβάνουμε το μήνυμα “STARTING.. IRMS= xx” στην επάνω γραμμή όπου μας ενημερώνει ότι είμαστε σε κατάσταση εκκίνησης και την τιμή του ρεύματος rms όπου διαρρέει το φορτίο μας . Στην κάτω σειρά βλέπουμε το μήνυμα “Volt = yy T=cc” όπου μας ενημερώνει για τις τιμές της τάσης rms και την θερμοκρασία εκείνη την στιγμή .

Μετά από χρονικό διάστημα ίσο με χρόνο t1 όπου ορίσαμε πιο πριν , το σύστημα μας μεταβαίνει στην κατάσταση κανονικής λειτουργίας και βλέπουμε το αντίστοιχο μήνυμα στην LCD οθόνη.



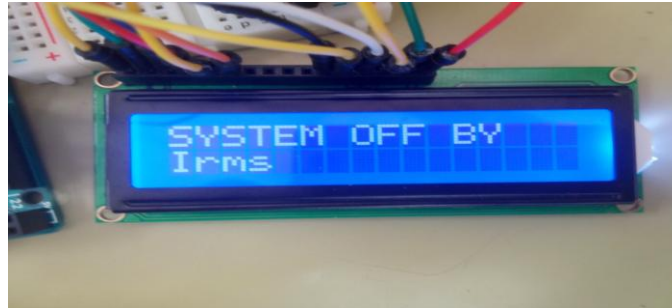
*Εικόνα 64: Κατάσταση κανονικής λειτουργίας*

Σε κατάσταση κανονικής λειτουργίας λαμβάνουμε τα δεδομένα από το ρεύμα rms που διαρρέει το φορτίο μας , τον μετρητή ώστε να έχουμε επίγνωση του αριθμού επανεκκινήσεων που έχουν γίνει απο θερμικό και τέλος την θερμοκρασία την δεδομένη στιγμή.

Στην κατάσταση κανονικής λειτουργίας και στην κατάσταση λειτουργία εκκίνησης , το

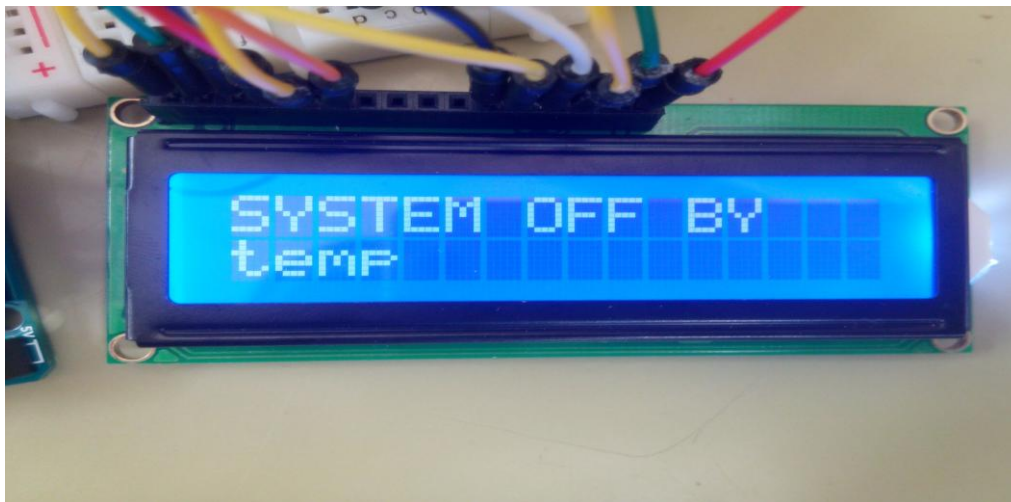
σύστημα μπορεί να διακοπεί από 5 αιτίες.

- Αιτία πρώτη: Ο χρήστης θέλει να σταματήσει το σύστημα. Η κίνηση που πρέπει να γίνει είναι να πατήσει το κουμπί STOP. Σε αυτήν την περίπτωση το σύστημα πηγαίνει στην κατάσταση “STOP”.
- Αιτία δεύτερη: Υπάρχει σφάλμα κατά το οποίο το ρεύμα έχει υπερβεί το ανώτερο όριο που του έχουμε ορίσει . Το σύστημα είναι στην κατάσταση “SYSTEM OFF BY IRMS” και η οθόνη έχει το αντίστοιχο μήνυμα .

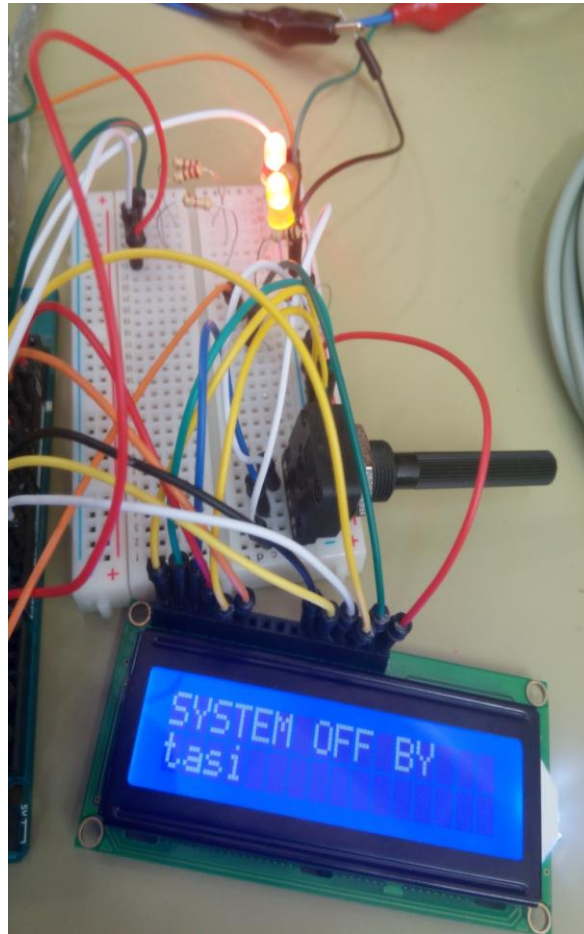


*Εικόνα 65: Περίπτωση σφάλματος από υπέρβαση ρεύματος*

- Αιτία τρίτη: Υπάρχει σφάλμα κατά το οποίο η θερμοκρασία έχει υπερβεί το ανώτερο όριο που της έχουμε ορίσει . Το σύστημα είναι στην κατάσταση “SYSTEM OFF BY



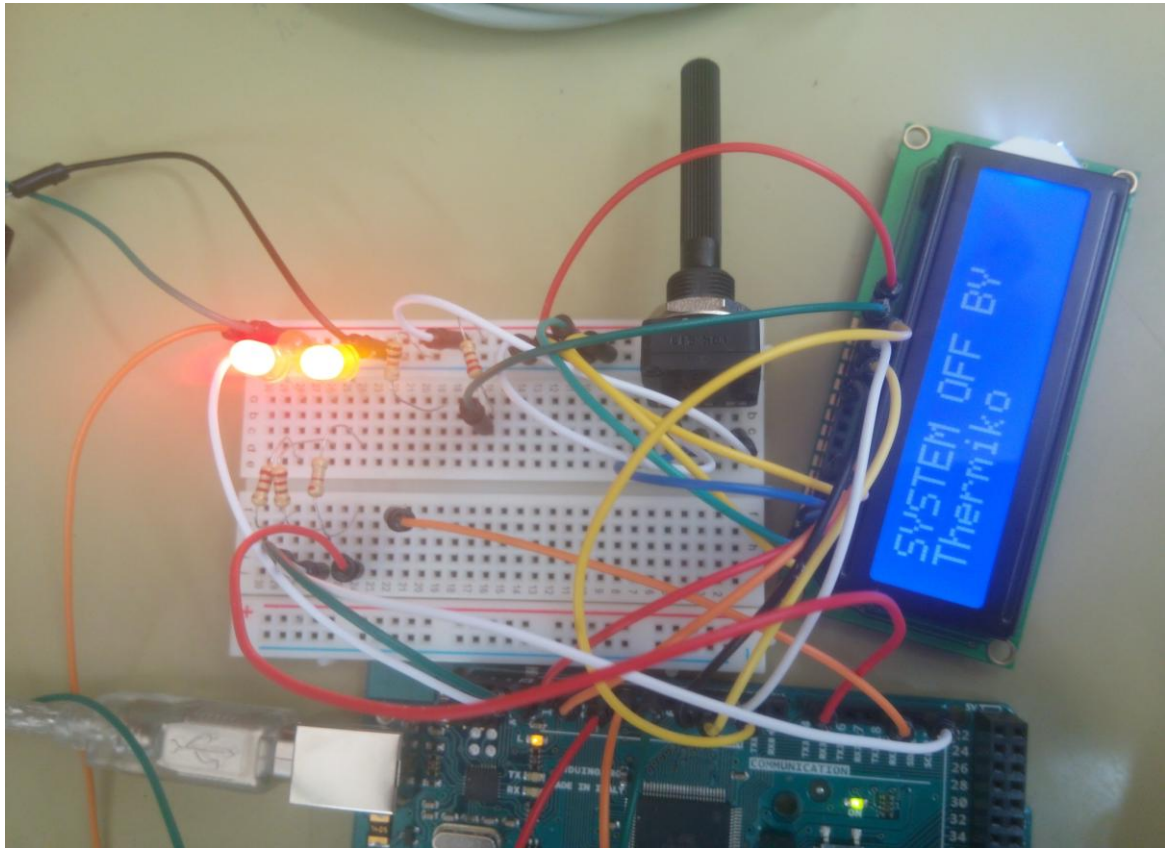
*Εικόνα 66: Περίπτωση σφάλματος από υπέρβαση θερμοκρασίας temp” και η οθόνη έχει το αντίστοιχο μήνυμα .*



Εικόνα 67:

*Περίπτωση σφάλματος από υπέρβαση ορίων τάσης*

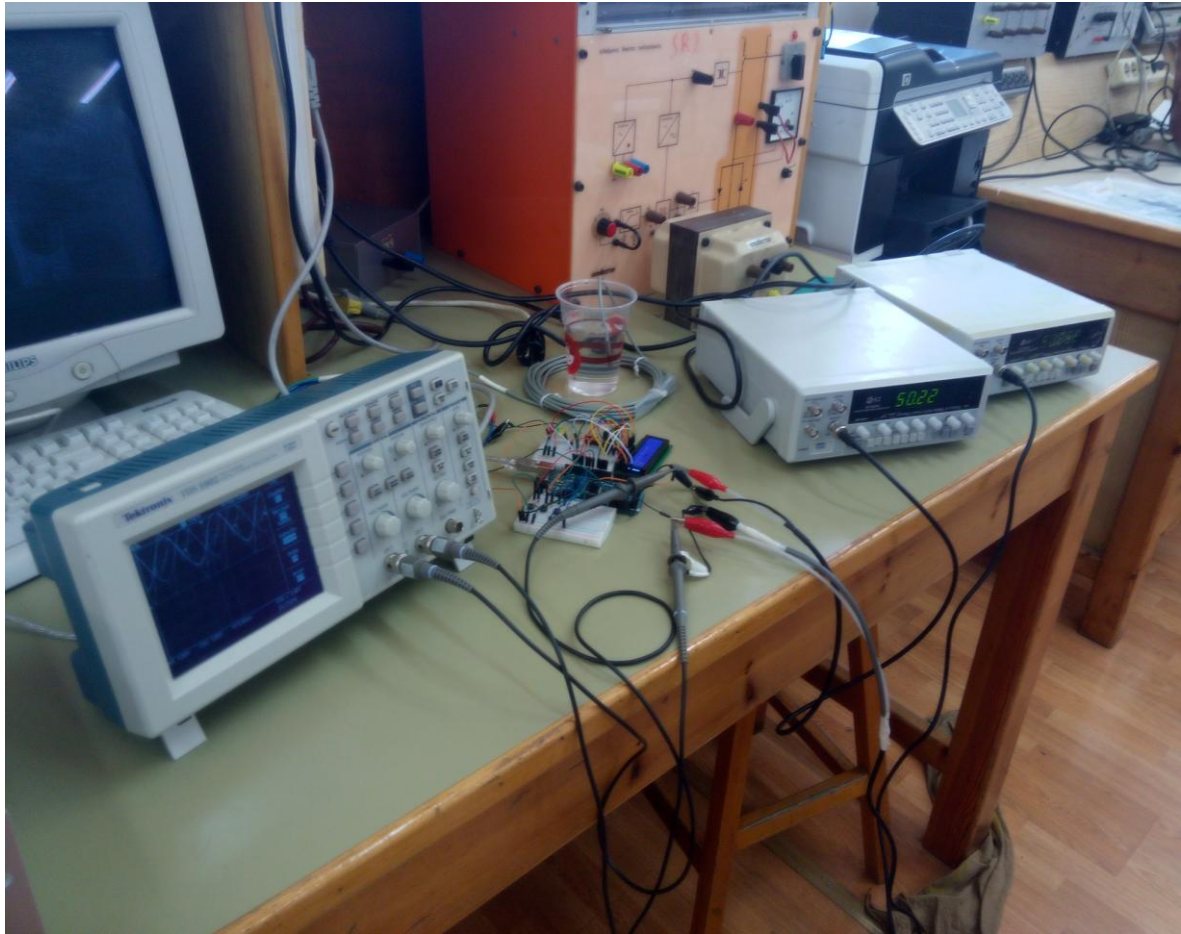
- Αιτία τέταρτη: Υπάρχει σφάλμα κατά το οποίο η τάση είτε έχει υπερβεί το ανώτερο όριο είτε το κατώτατο όριο που της έχουμε ορίσει . Το σύστημα είναι στην κατάσταση “SYSTEM OFF BY tasi” και η οθόνη έχει το αντίστοιχο μήνυμα .
- Αιτία πέμπτη: Υπάρχει σφάλμα κατά το οποίο έχουμε ένδειξη από το θερμικό . Το σύστημα είναι στην κατάσταση “SYSTEM OFF BY Thermiko” και η οθόνη έχει το αντίστοιχο μήνυμα .



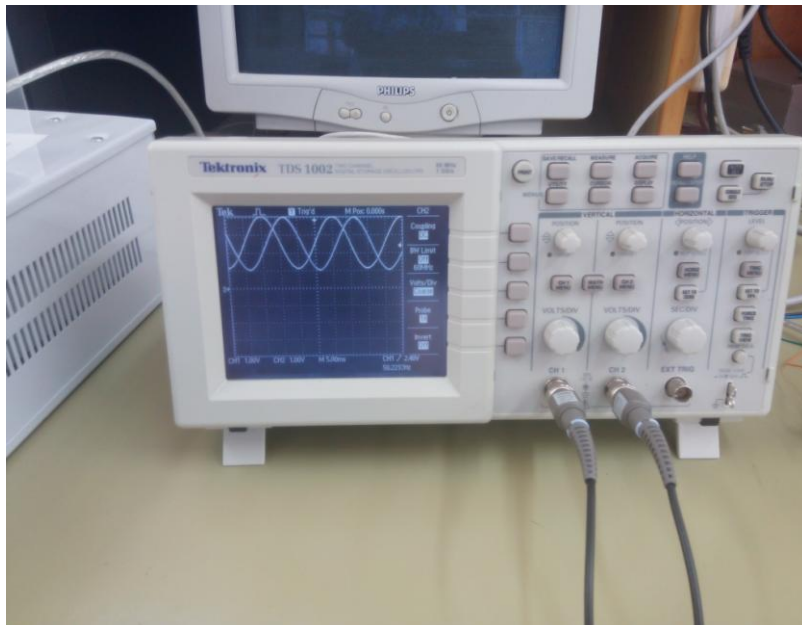
*Εικόνα 68: Περίπτωση σφάλματος από ένδειξη θερμικού*

Σημείωση : στις περιπτώσεις όπου το σύστημα από κατάσταση είτε λειτουργίας είτε εκκίνησης εισέρχεται στην κατάσταση αδράνειας λόγω σφάλματος , πρέπει για να έρθει σε κατάσταση “STOP” και από εκεί να επέλθει σε κατάσταση εκκίνησης. Αλλιώς το σύστημα δεν μπορεί να ξεκινήσει. Για να πάει σε αυτή την κατάσταση πρέπει να πατήσουμε συνεχώς το κουμπί του “STOP”

Σημείωση : Στην κατάσταση στην οποία υπάρχει σφάλμα από ένδειξη θερμικού και ταυτόχρονα έχει οριστεί η αυτόματη επανεκκίνηση τότε με την εκκαθάριση του σφάλματος του θερμικού θα υπάρξει αυτόματη επανεκκίνηση του συστήματος. Στην περίπτωση που ο χρήστης έχει θέσει μηδέν στην μεταβλητή της επανεκκίνησης από θερμικό, το σύστημα μπαίνει σε κατάσταση αδράνειας.



Εικόνα 69: Ολόκληρη η πειραματική διάταξη



Εικόνα 70: Ο παλμογράφος με τα σήματα του ρεύματος και της τάσης

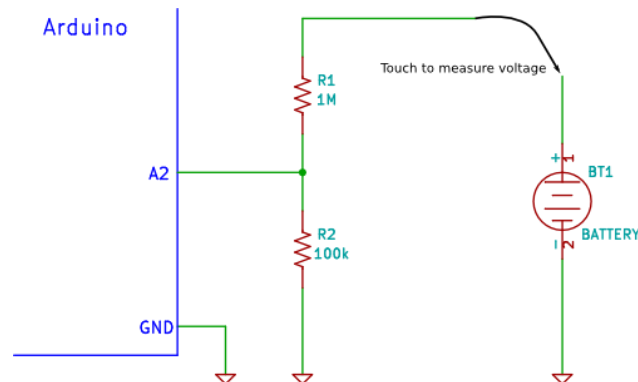
#### 4.2.2 Μεθοδολογία μέτρησης από το δίκτυο

Σε πραγματικές συνθήκες η δειγματοληψία θα πρέπει να γίνεται από αισθητήρες ρεύματος και τάσης. Δεν μπορούμε να κάνουμε δειγματοληψία απευθείας από το δίκτυο λόγω του γεγονότος ότι το Arduino δέχεται μόνο θετικές τιμές τις τάσεως έως 5volτ.

Για να ξεπεράσουμε αυτόν τον περιορισμό υπάρχουν οι παρακάτω λύσεις

##### 4.2.2.1 Μέτρηση τάσης με διαιρέτη τάσης

Ο διαιρέτης τάσης ή διαιρέτης δυναμικού είναι μία απλή κυκλωματική διάταξη η οποία αποτελείται από δύο αντιστάτες συνδεδεμένους εν σειρά, στα άκρα των οποίων εφαρμόζεται η τάση εισόδου. Ως τάση εξόδου λαμβάνεται η διαφορά δυναμικού ανάμεσα στους ακροδέκτες της μίας εκ των δύο αντιστάσεων. Οι τιμές που μπορεί να πάρει η τάση εξόδου κυμαίνονται από το 0 έως την τάση εισόδου. Επειδή εμείς θέλουμε συνεχές τάση , στην εν σειρά αντίσταση θα τοποθετήσουμε μια πυγή συνεχούς ρεύματος για την ανόρθωση του σήματος.

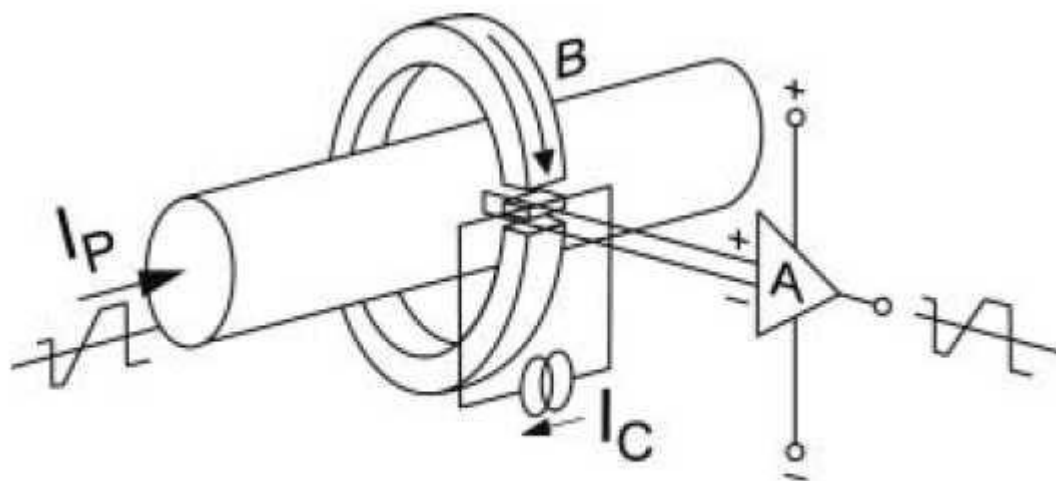


Εικόνα 71: Απλό σχηματικό διάγραμμα για την μέτρηση της τάσης από το δίκτυο

##### 4.2.2.2 Μέτρηση με αισθητήρα Hall

Η δομή ενός μετατροπέα μέτρησης ρεύματος με γαλβανική απομόνωση, μέσω ενός αισθητηρίου Hall σε διάταξη ανοικτού βρόχου εικονίζεται στην παρακάτω εικόνα. Το μετρούμενο ρεύμα κατά τη διέλευση του δημιουργεί μαγνητικό πεδίο, το οποίο συγκεντρώνεται σε ένα πυρήνα. Στο διάκενο αέρα του πυρήνα τοποθετείται ένα γραμμικό αισθητήριο Hall, το οποίο τροφοδοτείται με ένα 10 σταθερό ρεύμα  $I_{ref}$  και ανιχνεύει το μαγνητικό πεδίο, που είναι ανάλογο του μετρούμενου ρεύματος. Η τάση Hall που παράγει στη συνέχεια το αισθητήριο είναι ανάλογη του μαγνητικού πεδίου και επομένως ανάλογη του μετρούμενου ρεύματος. Αυτή η τάση, αφού υποστεί φιλτράρισμα και ενισχυθεί , αποτελεί την τάση εξόδου του μετατροπέα ( $U_o$ ) και είναι πιστό αντίγραφο του μετρούμενου ρεύματος. Η διαφορική ενίσχυση της τάσης Hall και το σταθερό ρεύμα  $I_{ref}$  παρέχονται από ηλεκτρονικά κυκλώματα που βρίσκονται στο εσωτερικό του μετατροπέα.





Εικόνα 72: Δομή του μετατροπέα μέτρησης ρεύματος με γαλβανική απομόνωση σε διάταξη ανοικτού βρόχου.

## **Κεφάλαιο 5ο**

### **Απομακρυσμένος έλεγχος του Arduino**

Ο απομακρυσμένος έλεγχος του Arduino μας επιτρέπει να έχουμε τον πλήρη έλεγχο και επίβλεψη του συστήματός μας μέσω Internet, από όπου κι αν βρισκόμαστε. Στον οδηγό αυτό θα δούμε τους σημαντικότερους τρόπους που θα μας επιτρέψουν την πρόσβαση στον Arduino μας εξ' αποστάσεως.

#### 5.1 Arduino και LabView

Το Arduino σαν πλακέτα ανοικτού κώδικα όπου είναι υπάρχουν πολλοί και διαφορετικοί τρόποι όπου μπορείς να την προγραμματίσεις αλλά και να την λειτουργήσεις . Το πρόγραμμα LabView της National Instruments έχει δύο διαφορετικούς τρόπους συνεργασίας .Ο ένας είναι να χρησιμοποιεί το Arduino ως καθαρά ένα μικροελεγκτή για την συλλογή δεδομένων από τους διάφορους αισθητήρες του έχουμε προσαρτήσει πάνω στην πλακέτα μας (μέθοδος LINX) και ο δεύτερος τρόπος είναι να προγραμματίσουμε αρχικά το Arduino με τον κωδικά μας και μετέπειτα να δημιουργήσουμε στο LabView το τι θα κάνουμε τα δεδομένα όπου δεχόμαστε από το Arduino .Η επικοινωνία αυτή γίνεται μέσω της σειριακής θύρας. Ο λόγος όπου θα θέλαμε την χρήση του Arduino με το LabView είναι ότι μέσω του LabView μπορούμε να έχουμε ένα ωραίο γραφικό περιβάλλον ελέγχου μέσα από τον υπολογιστή μας αλλά και μέσω του διαδικτύου .

Το αρνητικό στην χρήση του Arduino μέσω του προγράμματος LabView είναι ότι στην μέθοδο του LINX αν για κάποιο λόγο χαθεί η επικοινωνία ή κλείσει το πρόγραμμα ή ο υπολογιστής, το Arduino θα πάψει να δουλεύει ενώ στην δεύτερη περίπτωση το Arduino θα συνεχίσει να δουλεύει αλλά δεν θα έχουμε πλέον τον έλεγχο μέσω του LabView και η λειτουργίες του να είναι ανάλογες με το πρόγραμμα όπου τρέχει στο Arduino. Ακολουθεί ένας μικρός οδηγός για την εγκατάσταση του LabView και τον χειρισμό του Arduino μέσω της πλατφόρμας του LabView και του LINX.

##### 5.1.1 Εγκατάσταση του LabView

Για την εγκατάσταση του LabView αρκεί να μπούμε στην επίσημη ιστοσελίδα του LabView και να το κατεβάσουμε (<http://www.ni.com/download-labview/>) .Μετά το επιτυχημένο κατέβασμα από την ιστοσελίδα αρκεί να ακολουθήσουμε τις οδηγίες και να το εγκαταστήσουμε στον υπολογιστή μας.

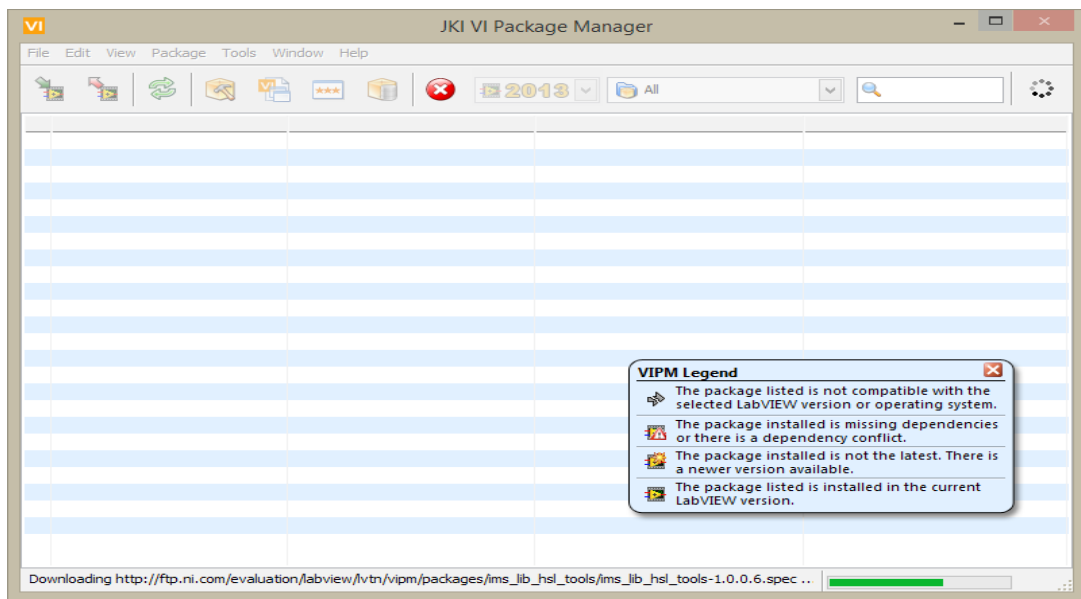
##### Εγκατάσταση του Visual Package Manager

Όταν το εγκαταστήσουμε στον υπολογιστή μας , ξεκινάμε την λειτουργία του Visual Package Manager (για ευκολία όταν θα αναφερόμαστε σε αυτό το πρόγραμμα θα αναφερόμαστε ως πρόγραμμα Συντονιστής ).Ο Συντονιστής είναι ενα δωρεάν πρόγραμμα που συνεργάζεται άριστα με το LabView και μας επιτρέπει να κατεβάζουμε δωρεάν νέα πακέτα (βιβλιοθήκες) για το LabView. Τον Συντονιστή μπορούμε εύκολα να το κατεβάσουμε από τον σύνδεσμο : <http://jki.net/vipm/download>



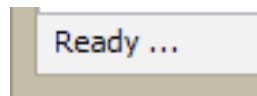
Εικόνα 73:  
Το λογότυπο του Visual  
Package Manager

Όταν ξεκινήσει ο Συντονιστής θα δούμε την παρακάτω εικόνα.



Εικόνα 74:

Ο Συντονιστής θα ξεκινήσει τη λήψη των ενημερώσεων από την αποθήκη βιβλιοθηκών . Η μπάρα κατάστασης βρίσκεται στο κάτω μέρος της εφαρμογής όταν το πρόγραμμα ενημερωθεί , η μπάρα κατάστασης θα αλλάξει σε Ready.



Εικόνα 75:

## 5.1.2 Κατεβάζουμε το Arduino IDE



Εικόνα 76: Το λογότυπο του Arduino IDE

Στην συνέχεια κατεβάζουμε και κάνουμε εγκατάσταση το πρόγραμμα για τον προγραμματισμό του Arduino όπως είδαμε σε προηγούμενο κεφάλαιο

### 5.1.3 Εγκατάσταση του LINX

Μετά την εγκατάσταση του Συντονιστή θα κατεβάσουμε και αυτόματα θα γίνει η εγκατάσταση του πακέτου LINX το οποίο έρχεται να αντικαταστήσει το παλιό **LabVIEW Interface for Arduino (LIFA)**. Ως εναλλακτικός τρόπος εγκατάστασης είναι επόμενος σύνδεσμος : <http://sine.ni.com/nips/cds/view/p/lang/en/nid/212478> Σε αυτό τον σύνδεσμο θα βρούμε τον σύνδεσμο για να κατεβάσουμε το πακέτου που χρειαζόμαστε.

## LINX - LVH

### Interface With Common Embedded Platforms

 [E-mail this Page](#) |  [Print](#) |  [PDF](#) |  [Rich Text](#)



[\[+\] Enlarge Picture](#)

- Interface with chipKIT, Arduino, and other embedded platforms
- Access peripherals such as DIO, AIO, PWM, SPI, and I2C from LabVIEW
- Take advantage of support for many common sensors
- Communicate over USB, serial, Ethernet, and wireless
- Easily deploy LINX code to NI myRIO and access I/O
- Quickly add a GUI to an embedded project

[Download](#)

Εικόνα 77: Η εικόνα όπως θα την βρούμε στην ιστοσελίδα της εταιρείας

Ακολουθώντας τον σύνδεσμο θα βρεθούμε σε μία άλλη σελίδα με τον ακριβές σύνδεσμο για να το κατεβάσουμε μέσω του Συντονιστή. Πατήστε πάνω στο κουμπί “**Download Toolkit**” για να ξεκινήσετε την διαδικασία εγκατάστασης :

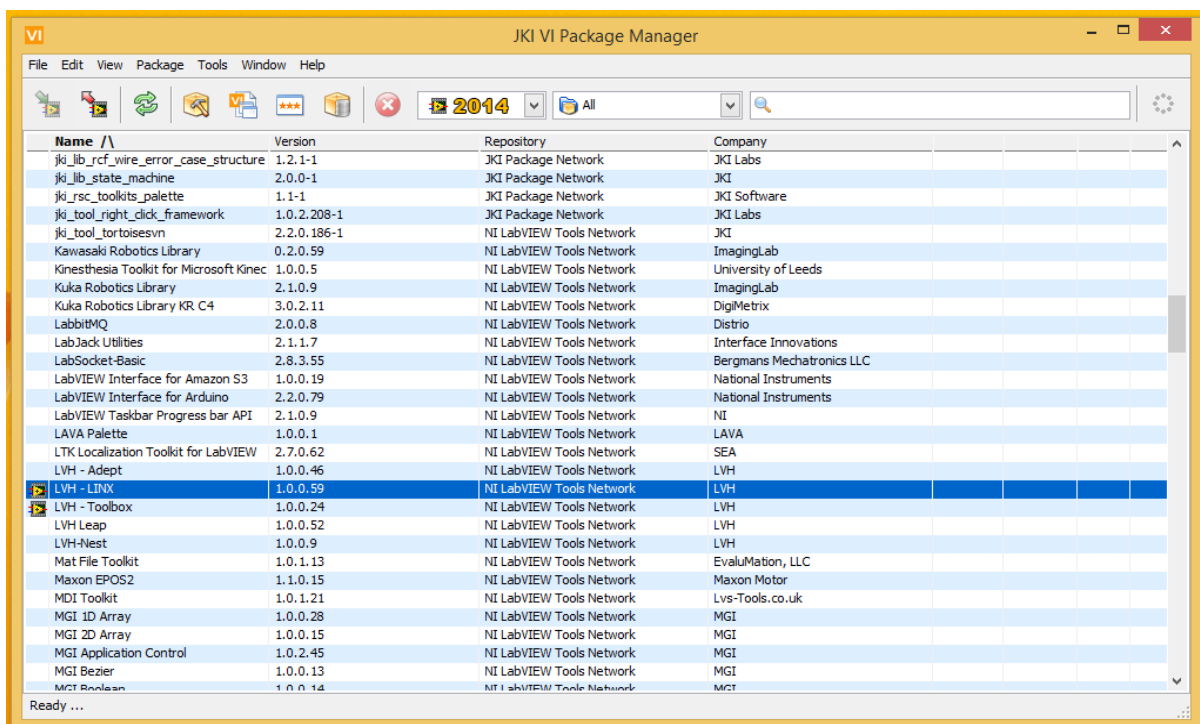
## 2 Step Two: Download LINX - LVH

Download and install the toolkit after installing VIPM.

### Download Toolkit

Εικόνα 78: Προσοχή στην ιστοσελίδα μας και υπάρχουν αρκετές παραπλανητικές διαφημίσεις

Ο Συντονιστής θα ανοίξει αυτόματα και θα ξεκινήσει το κατέβασμα και μετά την εγκατάσταση του πακέτου LINX



Εικόνα 79:

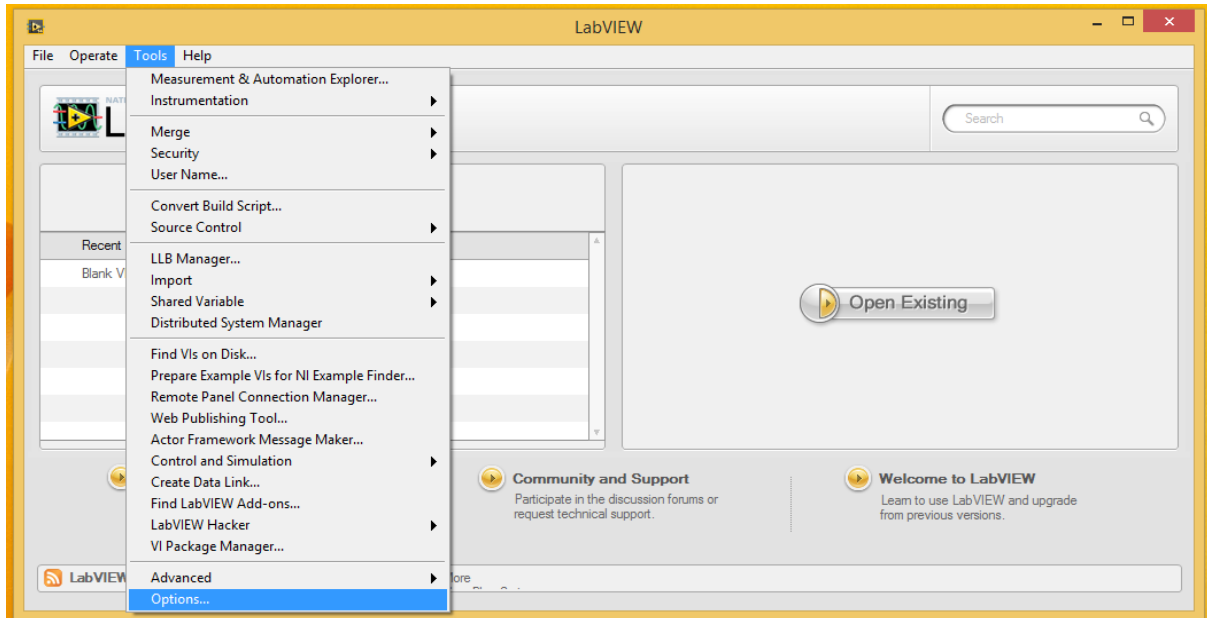
Αν ακολουθήσετε την διαδικασία και δεν την τελειώσετε επιτυχώς ή λάβετε σφάλμα κατά την λήψη ή εγκατάσταση του πακέτου , απλά ξεκινήστε την διαδικασία από το σημείο του κατεβάσματος και λογικά δεν θα αντιμετωπίσετε κάποιο πρόβλημα και θα δουλέψει

σωστά .

### Συνεργασία LabView με το LINX

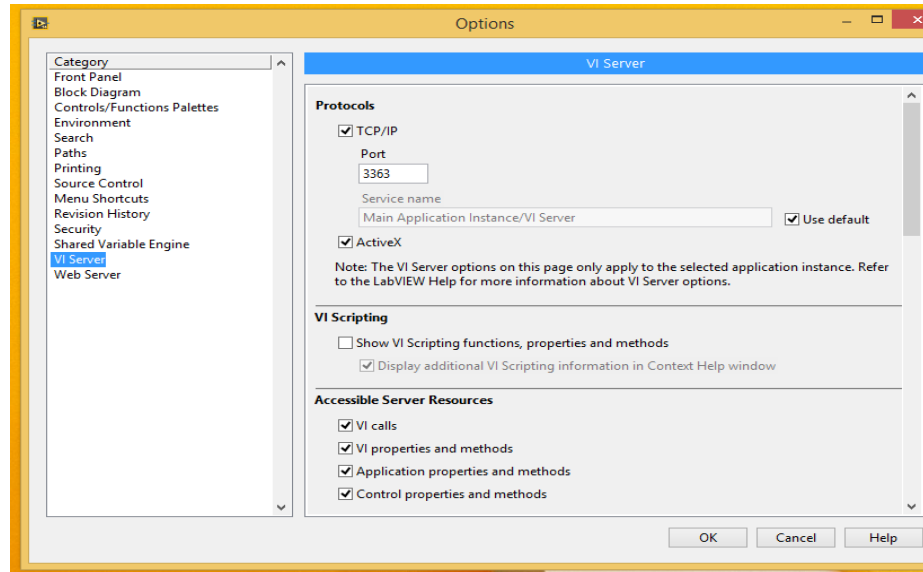
Τώρα αυτό που θα κάνουμε είναι στήσουμε το LabView έτσι ώστε να δουλέψει άριστα το παράδειγμά μας .

- Πρώτα ανοίξτε το LabView . Μην δημιουργήσετε κάποιο project , αλλά στο μενού , πατήστε Tools και μετά Options.



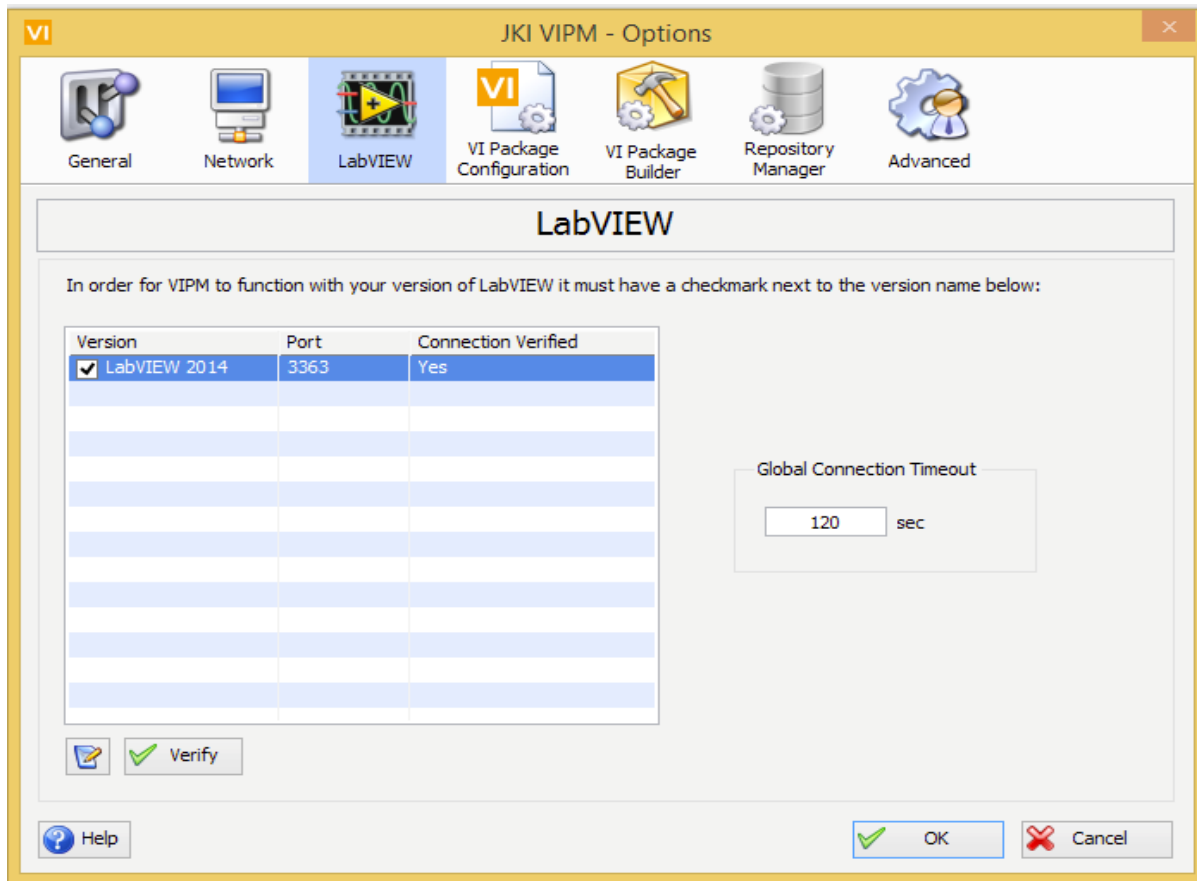
Εικόνα 80:

- Τώρα θα ανοίξει και θα σας μεταφέρει από το LabView στο παράθυρο Options όπου εδώ θα ρυθμίσουμε όλες τις επιλογές που θέλουμε . Επιλέγουμε την καρτέλα VI Server από το μενού.



Εικόνα 81:

- Εδώ βλέπουμε ότι μπορούμε να αλλάξουμε κάποιες από τις επιλογές όπου έχει . Αλλάξτε όλες τις επιλογές ώστε να μοιάζουν με την παραπάνω εικόνα .
- Μετά από αυτό πρέπει να κάνουμε τις ίδιες αλλαγές και στο πρόγραμμα του Συντονιστή ώστε και το LabView και ο Συντονιστής να μπορούν να μιλούν την ίδια γλώσσα με αποτέλεσμα να επικοινωνούν μεταξύ τους . Σε συστήματα όπως τα Windows αυτή η διαδικασία γίνεται αυτόματα σε αντίθεση με συστήματα όπως το OS X. Για να το κάνετε αυτό αρκεί να ανοίξετε το πρόγραμμα του Συντονιστή , από το μενού να επιλέξετε την επιλογή Tools και από εκεί το μενού Options . Στην καρτέλα που θα ανοίξει πατάμε το κουμπί του LabView



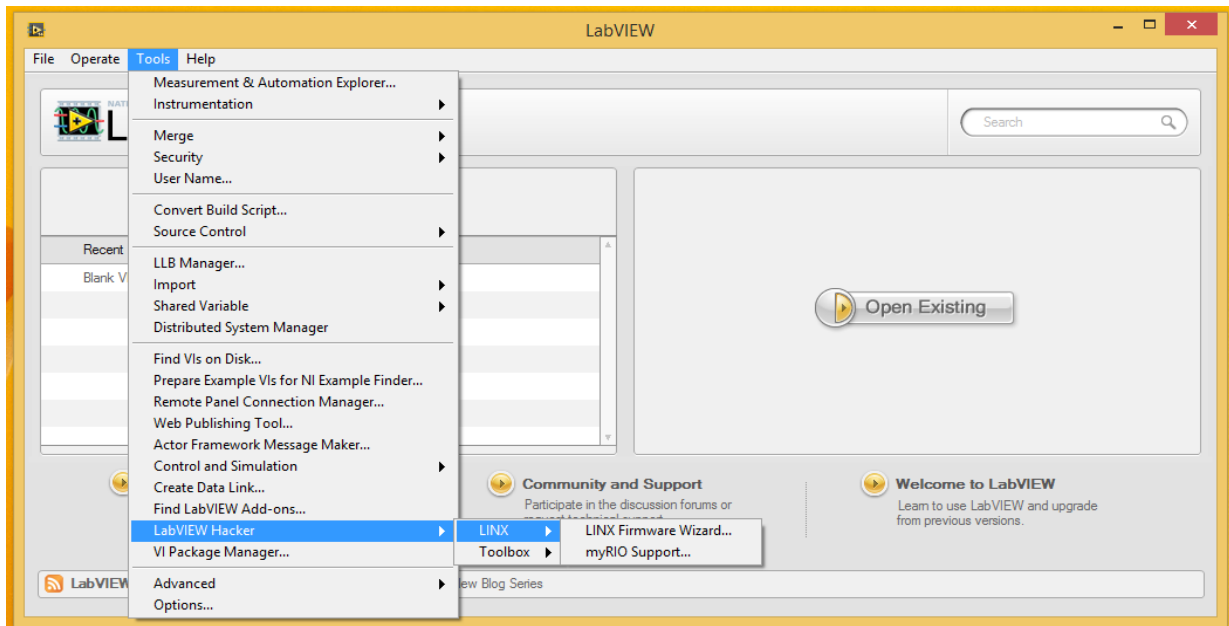
Εικόνα 82:

- Σε αυτή την καρτέλα , σιγουρευτείτε πως η τιμή της επιλογής Port είναι ίδια με την τιμή όπου ρυθμίζετε και μέσα στο LabView. Διορθώστε το εδώ αν βρίσκεστε σε αυτήν την περίπτωση και πατήστε confirm.

Δοκιμάζοντας την εγκατάσταση

Τώρα είμαστε έτοιμοι να δοκιμάσουμε τις εγκαταστάσεις των LabView και LINX και να δοκιμάσουμε τον προγραμματισμό του Arduino μέσω LabView .





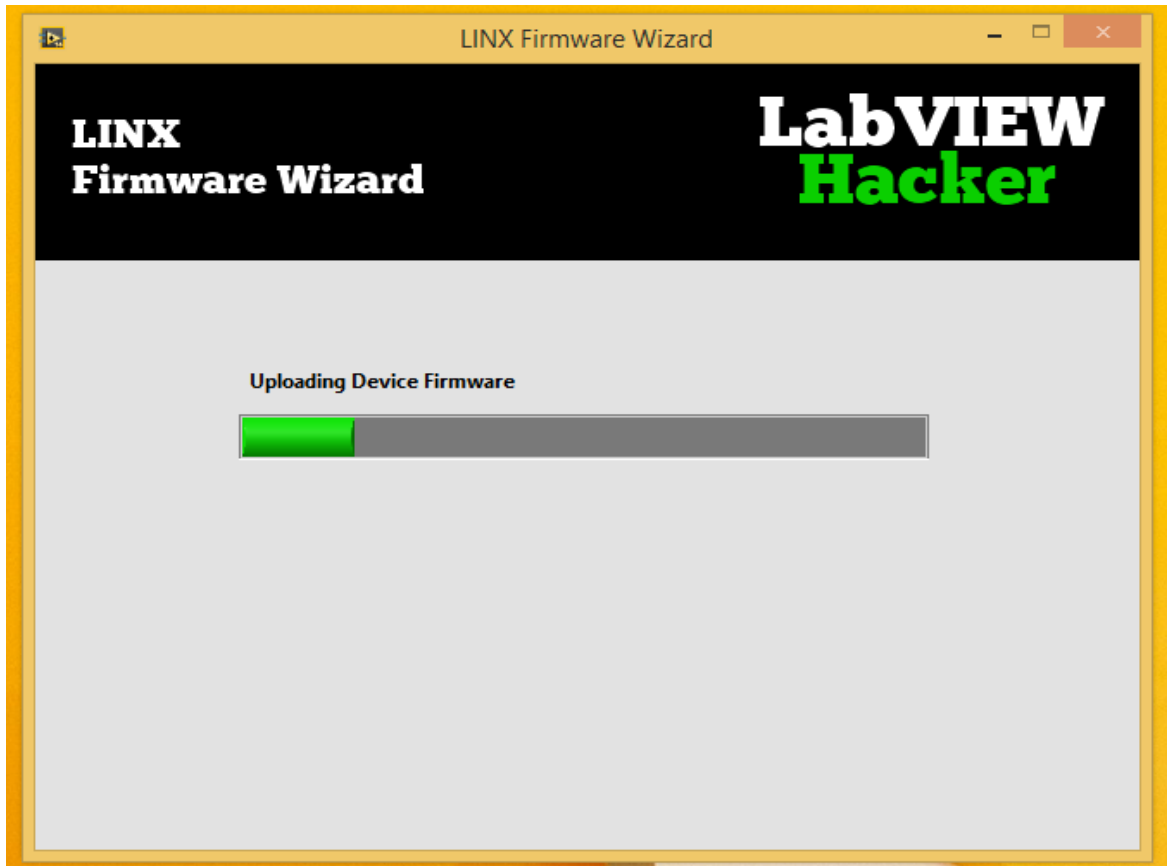
Εικόνα 83:

- Πρώτα θα ανοίξουμε το LabView και την κεντρική σελίδα του προγράμματος θα πατήσουμε Tools>LabView Hacker>LINX>LINX Firmware Wizard.
- Κάνοντας αυτό , θα μεταφερθούμε στο γραφικό περιβάλλον του LINX όπου θα πρέπει να εισάγουμε πληροφορίες όπως το μοντέλο του Arduino που έχουμε για το παράδειγμα μας. Σημείωση : Αυτό το βήμα χρειάζεται να γίνει μόνο 1 φορά . Αφού φορτωθεί το σωστό λογισμικό στο Arduino δεν χρειάζεται να του αλλάξουμε κάτι .
- Ο Οδηγός εγκατάστασης θα ξεκινήσει την εγκατάσταση ρωτώντας μας την πλακέτα όπου θα χρησιμοποιήσουμε . Ρυθμίστε την πρώτη σελίδα με τις ρυθμίσεις όπως φαίνονται στην παρακάτω εικόνα .



Εικόνα 84:

- Μετά από αυτό θα μας ζητήσει να βάλουμε την σειριακή θύρα όπου ο υπολογιστής μας “βλέπει” την πλακέτα Arduino. Τέλος αφού επιβεβαιώσουμε την σειριακή θύρα και ξεκινήσουμε το ανέβασμα του λογισμικού στο Arduino .



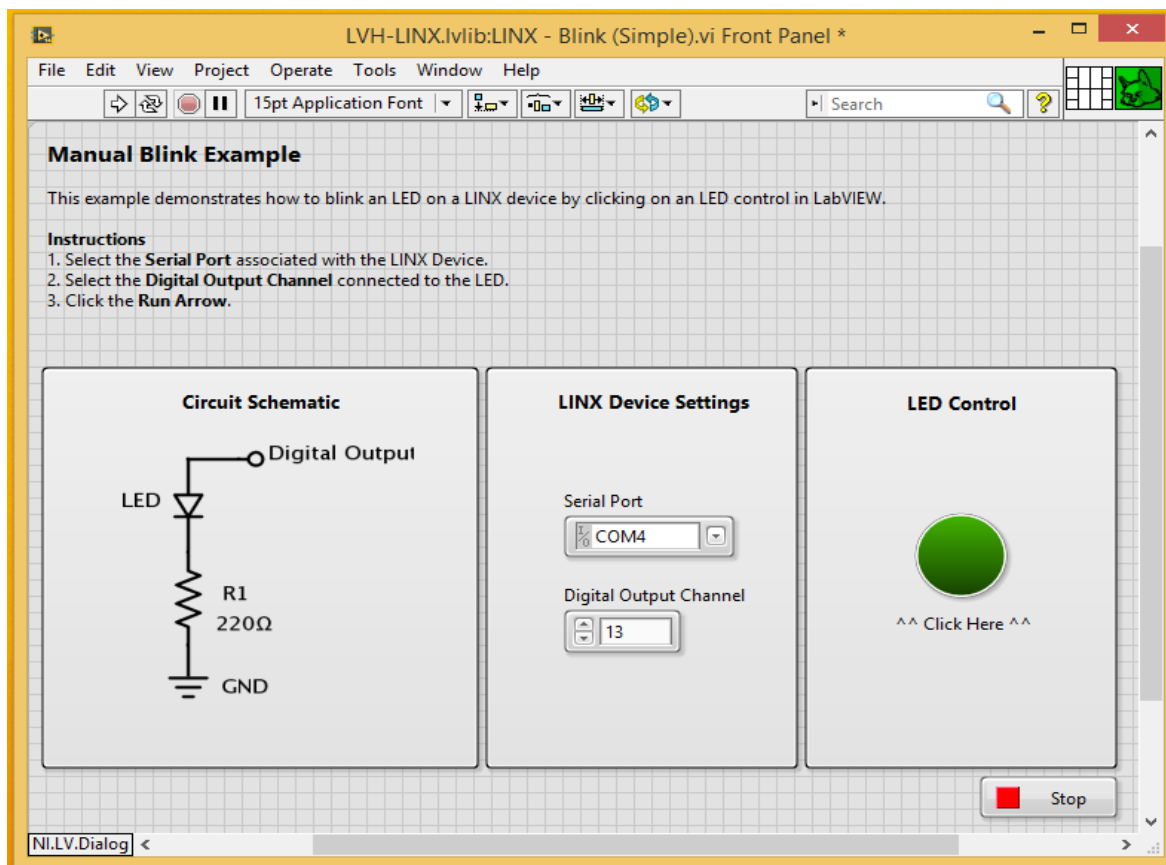
Εικόνα 85:

- Μετά το πέρας της διαδικασίας είστε έτοιμοι να χρησιμοποιήσετε το LINX για να προγραμματίσετε το Arduino .

Αν είχατε κάποιο θέμα και δεν δουλεύει κάτι σωστά , πολύ πιθανόν να χρειαστεί να κατεβάσουμε και να εγκαταστήσουμε το πακέτο NI-VISA από τον σύνδεσμο :

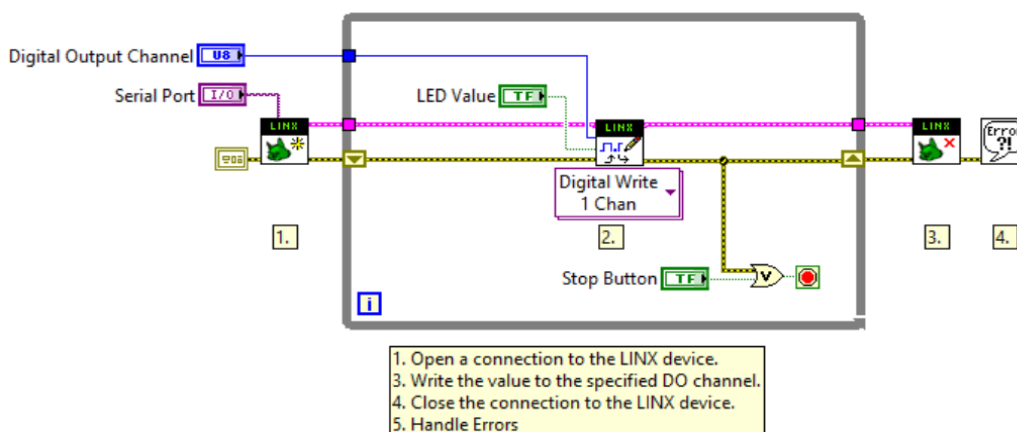
<http://www.ni.com/download/ni-visa-4.3/988/en/>

Στο τέλος της εγκατάστασης θα μας προσφερθεί να τρέξει ένα έτοιμο παράδειγμα . Δεχτείτε για να δοκιμάσετε το νέο γραφικό περιβάλλον .



Εικόνα 86:

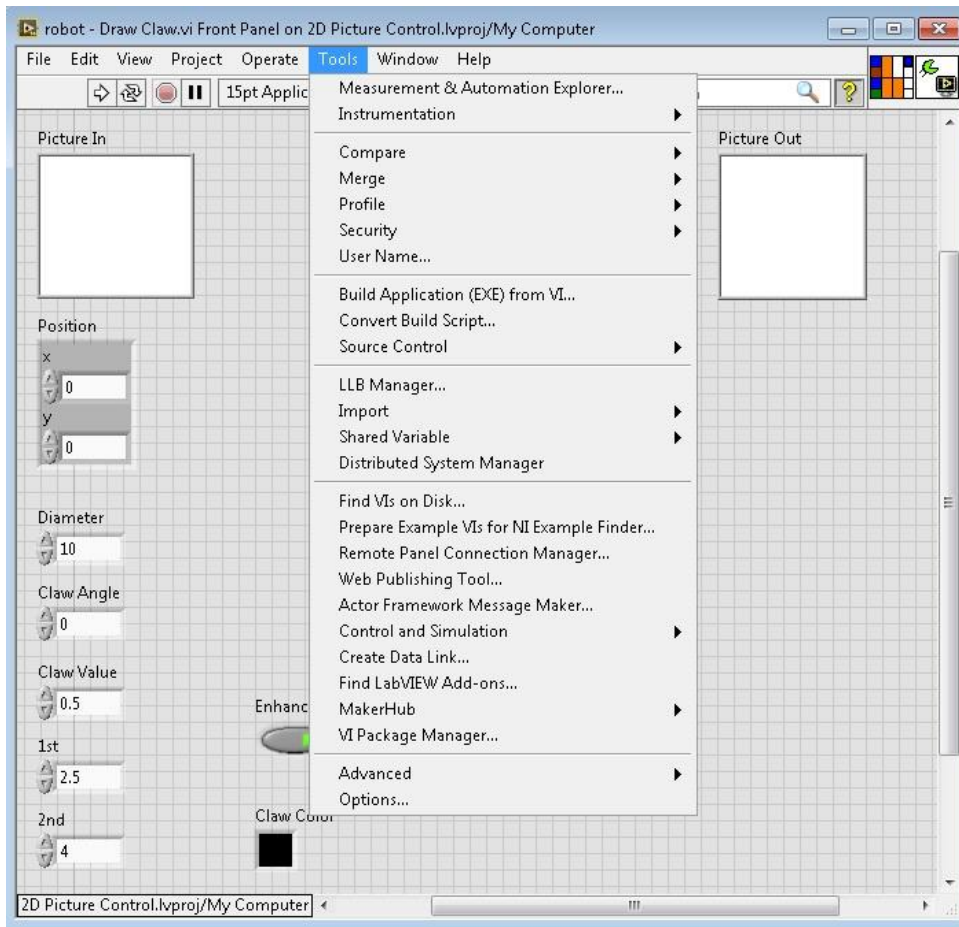
Το παράθυρο που άνοιξε λέγεται Front Panel και είναι το μέρος όπου ελέγχουμε το project μας. Στην περίπτωση όπου πατήσουμε το Ctrl+E ανοίγει το λεγόμενο Block Diagram το οποίο είναι το παράθυρο όπου προγραμματίζουμε το project μας



Εικόνα 87:

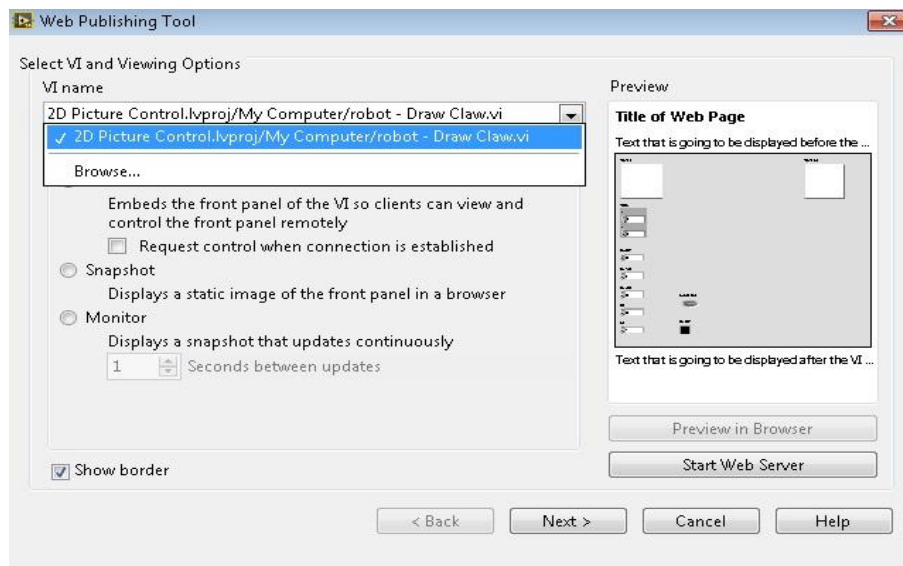
#### 5.1.4 Έξοδο στο διαδίκτυο.

Αφού έχουμε φτιάξει το πρόγραμμα μας και έχουμε διαπιστώσει ότι δουλεύει σωστά, για να το εμφανίσουμε στο διαδίκτυο αρκεί να κάνουμε τα εξής βήματα: Tools>Web Publishing Tool



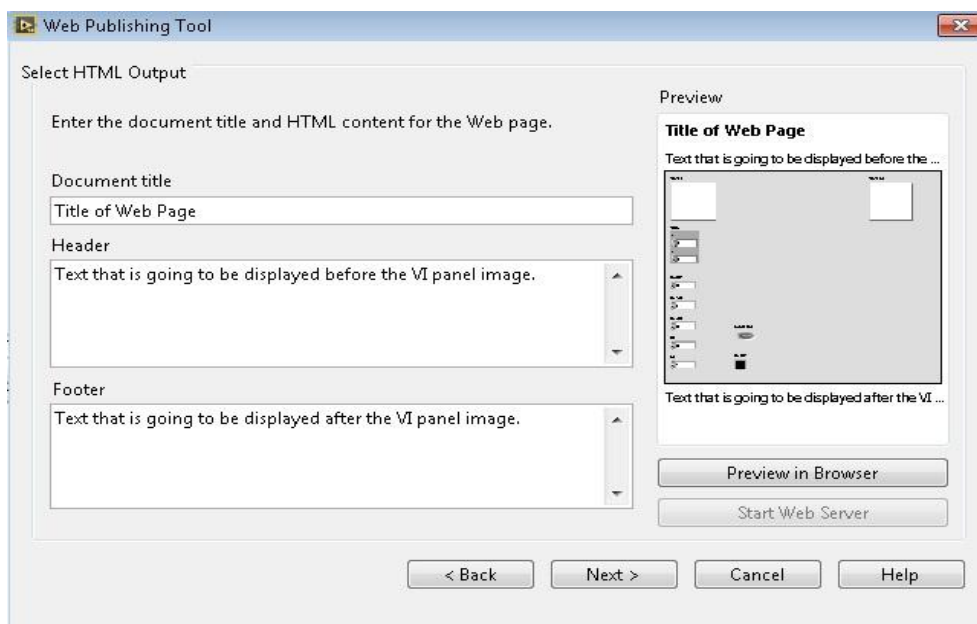
Εικόνα 88:

Στο επόμενο παράθυρο επιλέγουμε το project μας



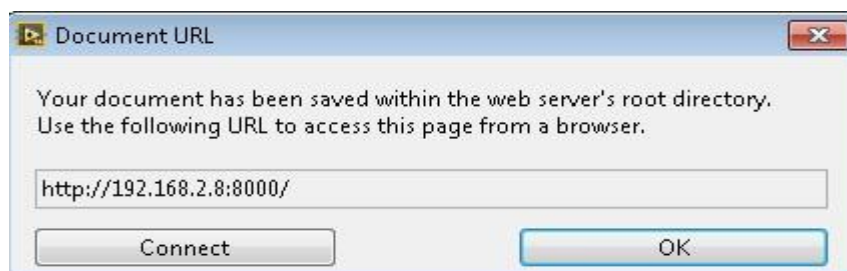
Εικόνα 89:

Μετά πατάμε το κουμπί Next> μέχρι να μας εμφανιστεί αυτή η επόμενη σελίδα.



Εικόνα 90

Σε αυτό το σημείο μπορούμε να αλλάξουμε ρυθμίσεις που αφορούν την ιστοσελίδα που θα απεικονίσουμε .Τέλος πατάμε αλλή μια φορά το Next>.



Εικόνα 91:

Στο παράθυρο αυτό βλέπουμε την διεύθυνση της ιστοσελίδας και σε ποία θύρα θα μας απαντήσει .Πατώντας Connect στο παράθυρο που θα μας εμφάνισε θα μας ανοίξει τον περιηγητή σε μία ιστοσελίδα που απεικονίζει ότι βλέπουμε στο πάνελ ελέγχου του LabView .

## 5.2 Έξοδο στο διαδίκτυο με Ethernet Shield

Όπως είδαμε είδαμε και σε προηγούμενο κεφάλαιο, υπάρχουν ήδη έτοιμες πλακέτες (Ethernet Shield) με τις οποίες ο χρήστης αφού τις συνδέσει στην πλακέτα του Arduino και φτιάχνοντας το κατάλληλο πρόγραμμα μπορεί να προβάλει τις πληροφορίες που θέλει στο διαδίκτυο. Τα θετικά στοιχεία με αυτήν την μέθοδο είναι το γεγονός ότι δεν χρειαζόμαστε κάποιο άλλο πρόγραμμα για να το προγραμματίσουμε (σε αντίθεση με το LabView ) και επίσης από την στιγμή που τροφοδοτηθεί το Arduino, φορτωθεί το πρόγραμμα, εφαρμοστεί το Ethernet Shield και τέλος συνδεθεί με το καλώδιο δικτύου, μπαίνοντας στην ιστοσελίδα που του έχουμε ορίσει , έχουμε πρόσβαση στο σύστημα μας χωρίς την ανάγκη κάποιου υπολογιστή υποστήριξης.

Το Arduino έχει στις βιβλιοθήκες του έτοιμα παραδείγματα για την χρήση της πλατφόρμας σε εφαρμογές στο διαδίκτυο, αλλά για κάτι εξεζητημένο ο χρήστης θα πρέπει να έχει και κάποιες γνώσεις δημιουργίας ιστοσελίδων .

## **Κεφάλαιο 6ο**

### **Συμπεράσματα πτυχιακής εργασίας**

Η παρούσα πτυχιακή εργασία ασχολήθηκε με τον έλεγχο ηλεκτρονόμου, που τροφοδοτεί μονοφασικό φορτίο, μέσω της πλατφόρμας Arduino με σκοπό την βελτίωση της προστασίας των ενεργητικών ή των παθητικών φορτίων του ηλεκτρονόμου σε περιπτώσεις υπερφόρτισης.

Σε περιπτώσεις βραχυκυκλώματος το κύριο μέσω προστασίας του κυκλώματος είναι η ασφάλεια ή ο μικροαυτόματος που βρίσκεται στην αρχή του κυκλώματος ισχύος.

Το Arduino λαμβάνοντας δεδομένα από τους αισθητήρες του (τάσης, έντασης και θερμοκρασίας) και την κατάσταση της βοηθητικής επαφής του θερμικού του ηλεκτρονόμου προστατεύει με μεγαλύτερη επάρκεια το κύκλωμα από υπερφορτίσεις σε σχέση με τον κλασικό αυτοματισμό.

Πιο αναλυτικά το Arduino προστατεύει το κύκλωμα από υπερτάσεις και υποτάσεις, υπερθέρμανση του φορτίου και υπερφορτίσεις μέσω του μετρητή ρεύματος και της βοηθητικής επαφής του θερμικού.

Στο βοηθητικό κύκλωμα του ηλεκτρονόμου συνδέθηκε σε σειρά με το ρελέ του Arduino η βοηθητική κανονικά κλειστή επαφή του θερμικού του ηλεκτρονόμου ως επιπρόσθετο μέρος προστασίας για την περίπτωση αστοχίας του arduino.

Το πρόγραμμα του Arduino έχει έτσι υλοποιηθεί ώστε με την εισαγωγή κατάλληλων παραμέτρων ο ηλεκτρονόμος να μπορεί να χρησιμοποιηθεί είτε για ενεργητικά είτε για παθητικά φορτία. Με τον τρόπο αυτό λαμβάνεται ή όχι υπόψη η εκκίνηση των ενεργητικών φορτίων. Το πρόγραμμα επίσης είναι έτσι υλοποιημένο έτσι ώστε αρχικά να εισαχθούν οι εξειδικευμένες (κρίσιμες) παράμετροι οι οποίες θέτουν τα όρια της τάσης, της έντασης, της θερμοκρασίας και στην συνέχεια ο οποιοσδήποτε μη εξειδικευμένος χρήστης να μην μπορεί να αλλάξει τα όρια πέραν των ορισμένων τιμών. Με τον τρόπο αυτό διασφαλίζεται η ορθή λειτουργία του συστήματος.

Το κόστος που θα δαπανηθεί για την αγορά του μικροεπεξεργαστή Arduino και των περιφερειακών του είναι σχετικά χαμηλό και αντισταθμίζεται για την την προστασία που παρέχει έναντι υπερφορτίσεων και κακών χειρισμό του χρήστη.

Τέλος η πειραματική διαδικασία επιβεβαίωσε την ορθή λειτουργία του συστήματος.

## Βιβλιογραφία

- (1) . [https:// www.arduino.cc](https://www.arduino.cc) (Ανάκτηση την 20/06/15)
- (2) Πτυχιακή εργασία Υλοποίηση ενός τρίτροχου κινούμενου ρομπότ χρησιμοποιώντας πλατφόρμα Arduino των Φελλόπουλος Αναστάσιος και Σπύρου Μαρία Τεχνολογικό Εκπαιδευτικό Ίδρυμα Σερρών Σχολή Τεχνολογικών Εφαρμογών Τμήμα Πληροφορικής και Επικοινωνιών ΣΕΡΡΕΣ, ΜΑΪΟΣ 2012
- (3) Εμμανουήλ Πουλάκης, Προγραμματίζοντας με τον μικροελεγκτή Arduino, Έκδοση 1η, Ηράκλειο Ιανουάριος 2015
- (4) Εργαστηριακές σημειώσεις μαθήματος Εσωτερικών ηλεκτρικών εγκαταστάσεων και αυτοματισμού, ΤΕΙ Δυτικής Ελλάδας ,Τμήμα Ηλεκτρολόγων Μηχανικών ΤΕ, ΠΑΤΡΑ 2014
- (5) Εργαστηριακές σημειώσεις μαθήματος Εγκαταστάσεων Ισχύος, ΤΕΙ Δυτικής Ελλάδας ,Τμήμα Ηλεκτρολόγων Μηχανικών ΤΕ, ΠΑΤΡΑ 2015
- (6) Marco Schwartz Oliver Manickum, Programming Arduino with LabVIEW, Packt Publishing Ltd. 2015

## Παράρτημα

```
#include <LiquidCrystal.h> // Συμπεριλαμβάνει την βιβλιοθήκη για την LCD οθόνη
//μας
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Ορίζει τους ακροδέκτες της LCD οθόνης μας
#include "EmonLib.h" //Συμπεριλαμβάνει την βιβλιοθήκη για τον υπολογισμό των
//rms τιμών
EnergyMonitor emon1; // Απαραίτητη συνάρτηση για τον υπολογισμό rms τιμών
int therm=0; //Ορίζουμε μεταβλητή για το θερμικό
int count=0; //Ορίζουμε μεταβλητή για το πόσες φορές επανέρχεται από
//θερμικό ο κινητήρας μας
int test=0; //Ορίζουμε βοηθητική μεταβλητή για τον έλεγχο στην επαναφορά
//από θερμικό
int i1=0; //Ορίζουμε την μεταβλητή του ρεύματος ελέγχου
int t1=0; //Ορίζουμε την μεταβλητή του ρεύματος του θερμικού
int i1min=0; //Ορίζουμε την μεταβλητή του ελάχιστου ρεύματος ελέγχου
int i1max=0; //Ορίζουμε την μεταβλητή του μέγιστου ρεύματος ελέγχου
int t1min=0; //Ορίζουμε την μεταβλητή του ελάχιστου ρεύματος θερμικού
int t1max=0; //Ορίζουμε την μεταβλητή του μέγιστου ρεύματος θερμικού
int testmillis=0; //Ορίζουμε την μεταβλητή του ελέγχου του χρόνου
int time=0; //Ορίζουμε την μεταβλητή της ώρας
int firstmillis=0; //Ορίζουμε την μεταβλητή του ελέγχου του χρόνου
int mult=1; //Ορίζουμε την μεταβλητή πολλαπλασιαστή του ρεύματος κατά
//την εκκίνηση
int mult2=1; //Ορίζουμε την μεταβλητή πολλαπλασιαστή του ρεύματος κατά
//την λειτουργία
int Tstr=0; //Ορίζουμε την μεταβλητή πολλαπλασιαστή του χρόνου εκκίνησης
//(μονάδα σε δευτερόλεπτα)
int b =0; //Ορίζουμε την βοηθητική μεταβλητή για δομή επανάληψης
int c=0; //Ορίζουμε την μεταβλητή για τον έλεγχο τον πολλαπλασιαστή
//ρεύματος κατά την λειτουργία
int d=0; //Ορίζουμε την μεταβλητή για τον έλεγχο του πολλαπλασιαστή του
//ρεύματος κατα την εκκίνηση
int temp=0; //Ορίζουμε την μεταβλητή για τον έλεγχο της θερμοκρασίας
int temp1=0; //Ορίζουμε την μεταβλητή για την δειγματοληψία της
//θερμοκρασίας
const int buttonUP = 9; //Ορίζουμε το ακροδέκτη 9 ότι θα αντιστοιχεί στο κουμπί UP
const int buttonDOWN = 8; //Ορίζουμε το ακροδέκτη 8 ότι θα αντιστοιχεί στο κουμπί
//DOWN
const int buttonMODE = 7; //Ορίζουμε το ακροδέκτη 7 ότι θα αντιστοιχεί στο κουμπί
//MODE
const int buttonTH = 18; //Ορίζουμε το ακροδέκτη 18 ότι θα αντιστοιχεί στον έλεγχο
//του θερμικού
int buttonstateUP = 0; //Ορίζουμε την κατάσταση που βρίσκεται το κουμπί UP
int buttonstateDOWN = 0; //Ορίζουμε την κατάσταση που βρίσκεται το κουμπί
//DOWN
int buttonstateMODE = 0 ; //Ορίζουμε την κατάσταση που βρίσκεται το κουμπί
//MODE
int buttonstateTH = 0; //Ορίζουμε την κατάσταση που βρίσκεται ο έλεγχος του
```



```

//θερμικού
String message; //Ορίζουμε βοηθητική μεταβλητή για την μεταφορά μηνυμάτων
float supplyVoltage = 0; //Ορίζουμε την μεταβλητή για την τοποθέτηση της τάσης rms
int Irms =0; //Ορίζουμε την μεταβλητή για την τοποθέτηση του ρεύματος rms
int Vmin=0; //Ορίζουμε την μεταβλητή για τον έλεγχο της κατώτερης τιμής της τάσης
int Vmax=0; //Ορίζουμε την μεταβλητή για τον έλεγχο της ανώτερης τιμής της τάσης
int assist=0; //Ορίζουμε βοηθητική μεταβλητή για την διαφορά αρχικής εκκίνησης και
//εκκίνησης από θερμικό
int assist1=0; //Ορίζουμε βοηθητική μεταβλητή για την διαφορά σφάλματος από θερμικό με
//τα άλλα είδη σφαλμάτων
float t = 0 ; //Ορίζουμε βοηθητική μεταβλητή
float i2=0; //Ορίζουμε την μεταβλητή για την αποθύκευση της τιμής του θερμομέτρου
int sensorPin = A3; //Ορίζουμε τον ακροδέκτη A3 ότι θα αντιστοιχεί στο όνομα sensorPin
int StopLed = 13; //Ορίζουμε το ακροδέκτη 13 ότι θα αντιστοιχεί στο StopLed
int StartLed = 20; //Ορίζουμε το ακροδέκτη 20 ότι θα αντιστοιχεί στο StartLed
int ThLed = 14; //Ορίζουμε το ακροδέκτη 14 ότι θα αντιστοιχεί στο ThLed
int i=0; //Ορίζουμε βοηθητική μεταβλητή για όταν το σύστημα είναι σε
//κατάσταση ON
int j=0; //Ορίζουμε βοηθητική μεταβλητή για όταν έχουμε ένδειξη θερμικού

void setup() { //Υποχρεωτική εντολή για ορισμό μεταβλητών που θέλουν μόνο
//μία φορά κατά την έναρξη του προγράμματος

    lcd.begin(16, 2); //Υποχρεωτική εντολή για την ρύθμιση της LCD οθόνης
    pinMode(buttonUP, INPUT_PULLUP); //Ορίζουμε ακροδέκτη ως είσοδο
    pinMode(buttonDOWN, INPUT_PULLUP); //Ορίζουμε ακροδέκτη ως είσοδο
    pinMode(buttonMODE, INPUT_PULLUP); //Ορίζουμε ακροδέκτη ως είσοδο

    pinMode(StopLed, OUTPUT); //Ορίζουμε ακροδέκτη ως έξοδο
    pinMode(StartLed, OUTPUT); //Ορίζουμε ακροδέκτη ως έξοδο
    pinMode(ThLed, OUTPUT); //Ορίζουμε ακροδέκτη ως έξοδο

    emon1.voltage(2, 230, 0); //Ορίζουμε τον υπολογισμό της τάσης .Οι τιμές είναι με την
//σειρά : ακροδέκτης εισόδου , βαθμονόμηση, διαφορά φάσης
    emon1.current(1, 10); //Ορίζουμε τον υπολογισμό του ρεύματος. Οι τιμές είναι με την
//σειρά: ακροδέκτης εισόδου, βαθμονόμηση.
} //Τέλος αρχικοποιήσεων

void loop() { //Υποχρεωτική εντολή έναρξης επανάληψης βασικού
//προγράμματος

    bailout: //εντολή δείκτης συνδιασμένη με την εντολή "goto"

    for (int thisChar = 0; thisChar < 10; thisChar++) { //Έναρξη δομής επανάληψης
        lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD
//οθόνη
        lcd.print("START IN"); //Εγγραφή μηνύματος "START IN" στην LCD οθόνη
    }
}

```

```

lcd.setCursor(0, 1);           //Ορισμός σε ποιά τοποθεσία θα ξεκινήσει η εγγραφή στην LCD
                               //οθόνη
lcd.print(10- thisChar);      //Με βάση την δομή επανάληψης ορίζεται ο αντίστροφος
                               //μετρητής

delay(500);                   //Καθυστέρηση 500 millisecond
lcd.clear();                  //Καθαρίζουμε την LCD οθόνη
}                               //Τέλος δομής επανάληψης

for (int thisChar = 0; thisChar < 11; thisChar) { //Εναρξη δομής επανάληψης
  if(digitalRead(buttonUP)== HIGH){                //Εξήγηση λειτουργίας " Αν δεις ένδειξη
                                                    //από τον ακροδέκτη ButtonUP

  temp++;                                          //Αύξησε την μεταβλητή της μέγιστης
                                                    //θερμοκρασίας"
  }                                               //Τέλος δομής επιλογής
  if(digitalRead(buttonDOWN)== HIGH && temp > 0){ //Εξήγηση λειτουργίας "Αν δεις
                                                    //ένδειξη από τον ακροδέκτη ButtonDOWN

  temp--;                                         //Μείωσε την μεταβλητή της μέγιστης θερμοκρασίας"
  }                                               //Τέλος δομής επιλογής
  lcd.clear();                                   //Καθάρισε την LCD οθόνη
  lcd.setCursor(0, 0);                           //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή
                                                    //στην LCD οθόνη
  lcd.print("Temperature=");                     //Εγγραφή μηνύματος Temperature= στην LCD
                                                    //οθόνη
  lcd.setCursor(0, 1);                           //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή
                                                    //στην LCD οθόνη
  lcd.print(temp);                               //Εγγραφή του περιεχομένου της μεταβλητής temp1
  delay(200);                                    //Καθυστέρηση 200 millisecond

  if (digitalRead(buttonMODE)== HIGH){           //Εξήγηση λειτουργίας "αν δεις ένδειξη
                                                    //από τον ακροδέκτη ButtonMODE

  lcd.clear();                                   //Καθάρισε την LCD οθόνη
  lcd.setCursor(0, 0);                           //Ορισμός σε ποιά τοποθεσία θα ξεκινήσει η εγγραφή
                                                    //στην LCD οθόνη
  lcd.print("WAIT.....");                       //Εγγραφή μηνύματος "WAIT....." στην LCD οθόνη
  delay(1000);                                   //Καθυστέρηση 1000 millisecond
  break;                                         //Σπάσε την δομή επανάληψης
  }                                               //Τέλος δομής επιλογής
  }                                               //Τέλος δομής επανάληψης

  for (int thisChar = 0; thisChar < 11; thisChar) { //Αρχή δομής επανάληψης
    if(digitalRead(buttonUP)== HIGH){            //Εξήγηση λειτουργίας " αν δεις ένδειξη
                                                    //από τον ακροδέκτη ButtonUP

    Vmin++;                                       //Αύξησε την μεταβλητή πολλαπλασιαστή του
                                                    // χρόνου εκκίνησης"
    }                                             //Τέλος δομής επιλογής
    if(digitalRead(buttonDOWN)== HIGH && Vmin > 0){ //Εξήγηση λειτουργίας "αν δεις
                                                    //ένδειξη από τον ακροδέκτη ButtonDOWN

    Vmin--;                                       //Μείωσε την μεταβλητή πολλαπλασιαστή του
                                                    //χρόνου εκκίνησης"

```

```

} //Τέλος δομής επιλογής
lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή
//στην LCD οθόνη
lcd.print("VoltageMin="); //Εγγραφή μηνύματος VoltageMin= στην LCD
//οθόνη
lcd.setCursor(0, 1); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή
//στην LCD οθόνη
lcd.print(Vmin); //Εγγραφή του περιεχομένου της μεταβλητής mult
delay(200); //Καθυστέρηση 200 millisecond

if (digitalRead(buttonMODE)== HIGH){ //Εξήγηση λειτουργίας "Αν δεις ένδειξη
//από τον ακροδέκτη ButtonMODE

lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή
//στην LCD οθόνη
lcd.print("WAIT....."); //Εγγραφή μηνύματος "WAIT....." στην LCD οθόνη
delay(1000); //Καθυστέρηση 1000 millisecond
break; //Σπάσε την δομή επανάληψης
} //Τέλος δομής επιλογής
} //Τέλος δομής επανάληψης

Vmax=Vmin; //Ορίζουμε την μέγιστη τιμή της τάσης ίσης με την ελάχιστη ώστε ο
//χρήστης να μην μπορεί να ορίσει ως μέγιστη τιμή μικρότερη από την ελάχιστη
for (int thisChar = 0; thisChar < 11; thisChar) { //Αρχή δομής επανάληψης
if(digitalRead(buttonUP)== HIGH){ //Εξήγηση λειτουργίας " Αν δεις ένδειξη
//από τον ακροδέκτη ButtonUP

Vmax++; //Αύξησε την μεταβλητή πολλαπλασιαστή του χρόνου εκκίνησης"
} //Τέλος δομής επιλογής
if(digitalRead(buttonDOWN)== HIGH && Vmax > Vmin){ //Εξήγηση λειτουργίας "Αν
//δεις ένδειξη από τον ακροδέκτη ButtonDOWN

Vmax--; //Μείωσε την μεταβλητή πολλαπλασιαστή του χρόνου
εκκίνησης"
} //Τέλος δομής επιλογής

lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή
//στην LCD οθόνη
lcd.print("VoltageMax="); //Εγγραφή μηνύματος VoltageMax= στην LCD
οθόνη
lcd.setCursor(0, 1); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή
//στην LCD οθόνη
lcd.print(Vmax); //Εγγραφή του περιεχομένου της μεταβλητής mult
delay(200); //Καθυστέρηση 200 millisecond

if (digitalRead(buttonMODE)== HIGH){ //Εξήγηση λειτουργίας "Αν δεις ένδειξη
//από τον ακροδέκτη ButtonMODE

lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή
//στην LCD οθόνη

```

```

lcd.print("WAIT.....");           //Εγγραφή μηνύματος "WAIT....." στην LCD οθόνη
delay(1000);                       //Καθυστέρηση 1000 millisecond
break;                              //Σπάσε την δομή επανάληψης
}                                    //Τέλος δομής επιλογής
}                                    //Τέλος δομής επανάληψης

for (int thisChar = 0; thisChar < 11; thisChar) { //Αρχή δομής επανάληψης
  if(digitalRead(buttonUP)== HIGH){           //Εξήγηση λειτουργίας " Αν δεις ένδειξη
                                              //από τον ακροδέκτη ButtonUP
    mult++;                                  //Αύξησε την μεταβλητή πολλαπλασιαστή του χρόνου
                                              //εκκίνησης"
  }                                           //Τέλος δομής επιλογής
  if(digitalRead(buttonDOWN)== HIGH && mult > 1){ //Εξήγηση λειτουργίας "Αν δεις
                                              //ένδειξη από τον ακροδέκτη ButtonDOWN
    mult--;                                  //Μείωσε την μεταβλητή πολλαπλασιαστή του χρόνου
                                              //εκκίνησης"
  }                                           //Τέλος δομής επιλογής
  lcd.clear();                               //Καθάρισε την LCD οθόνη
  lcd.setCursor(0, 0);                      //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή
                                              //στην LCD οθόνη
  lcd.print("MultStr=");                   //Εγγραφή μηνύματος MultStr= στην LCD οθόνη
  lcd.setCursor(0, 1);                     //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή
                                              //στην LCD οθόνη
  lcd.print(mult);                          //Εγγραφή του περιεχομένου της μεταβλητής mult
  delay(200);                               //Καθυστέρηση 200 millisecond

  if (digitalRead(buttonMODE)== HIGH){      //Εξήγηση λειτουργίας "Αν δεις ένδειξη
                                              //από τον ακροδέκτη ButtonMODE
    lcd.clear();                            //Καθάρισε την LCD οθόνη
    lcd.setCursor(0, 0);                    //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή
                                              //στην LCD οθόνη
    lcd.print("WAIT.....");                 //Εγγραφή μηνύματος "WAIT....." στην LCD οθόνη
    delay(1000);                           //Καθυστέρηση 1000 millisecond
    break;                                  //Σπάσε την δομή επανάληψης
  }                                          //Τέλος δομής επιλογής
}                                          //Τέλος δομής επανάληψης

for (int thisChar = 0; thisChar < 11; thisChar) { //Εναρξη δομής επανάληψης
  if(digitalRead(buttonUP)== HIGH){           //Εξήγηση λειτουργίας " Αν δεις ένδειξη από
                                              //τον ακροδέκτη ButtonUP
    mult2++;                                 //Αύξησε την μεταβλητή πολλαπλασιαστή του χρόνου
                                              //λειτουργίας"
  }                                           //Τέλος δομής επιλογής
  if(digitalRead(buttonDOWN)== HIGH && mult2 > 1){ //Εξήγηση λειτουργίας "Αν δεις
                                              //ένδειξη από τον ακροδέκτη ButtonDOWN
    mult2--;                                 //Μείωσε την μεταβλητή πολλαπλασιαστή του χρόνου
                                              //λειτουργίας
  }                                           //Τέλος δομής επιλογής
}

```

```

lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή
//στην LCD οθόνη
lcd.print("MultCont="); //Εγγραφή μηνύματος MultCont= στην LCD οθόνη
lcd.setCursor(0, 1); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή
στην LCD οθόνη
lcd.print(mult2); //Εγγραφή του περιεχομένου της μεταβλητής mult2
delay(200); //Καθυστέρηση 200 millisecond

if (digitalRead(buttonMODE)== HIGH){ //Εξήγηση λειτουργίας "Αν δεις ένδειξη
από τον ακροδέκτη ButtonMODE
lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή
//στην LCD οθόνη
lcd.print("WAIT....."); //Εγγραφή μηνύματος "WAIT....." στην LCD οθόνη
delay(1000); //Καθυστέρηση 1000 millisecond
break; //Σπάσε την δομή επανάληψης
} //Τέλος δομής επιλογής
} //Τέλος δομής επανάληψης

for (int thisChar = 0; thisChar < 11; thisChar) { //Εναρξη δομής επανάληψης
if(digitalRead(buttonUP)== HIGH){ //Εξήγηση λειτουργίας " Αν δεις ένδειξη από
//τον ακροδέκτη ButtonUP
++i1min; //Αύξησε την μεταβλητή του ελάχιστου ρεύματος
//ελέγχου"
} //Τέλος δομής επιλογής
if(digitalRead(buttonDOWN)== HIGH && i1min > 0){ //Εξήγηση λειτουργίας "Αν δεις
//ένδειξη από τον ακροδέκτη ButtonDOWN
--i1min; //Μείωσε την μεταβλητή του ελάχιστου ρεύματος
//ελέγχου"
} //Τέλος δομής επιλογής

lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή
//στην LCD οθόνη
lcd.print("Ithmin="); //Εγγραφή μηνύματος Ithmin= στην LCD οθόνη
lcd.setCursor(0, 1); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή
//στην LCD οθόνη
lcd.print(i1min); //Εγγραφή του περιεχομένου της μεταβλητής i1min
delay(200); //Καθυστέρηση 200 millisecond

if (digitalRead(buttonMODE)== HIGH){ //Εξήγηση λειτουργίας "Αν δεις ένδειξη
//από τον ακροδέκτη ButtonMODE
lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή
//στην LCD οθόνη
lcd.print("WAIT....."); //Εγγραφή μηνύματος WAIT..... στην LCD οθόνη
delay(1000); //Καθυστέρηση 1000 millisecond
break; //Σπάσε την δομή επανάληψης
} //Τέλος δομής επιλογής

```

```

} //Τέλος δομής επανάληψης
i1max=i1min; //Ορίζουμε την μέγιστη τιμή του ρεύματος ίση με την ελάχιστη ώστε ο
//χρήστης να μην μπορεί να ορίσει ως μέγιστη τιμή μικρότερη από την ελάχιστη

for (int thisChar = 0; thisChar < 11; thisChar) { //Εναρξη δομής επανάληψης
if(digitalRead(buttonUP)== HIGH){ //Εξήγηση λειτουργίας " Αν δεις ένδειξη από τον
//ακροδέκτη ButtonUP
++i1max; //Αύξησε την μεταβλητή του μέγιστου ρεύματος ελέγχου κατά ένα"
} //Τέλος δομής επιλογής
if(digitalRead(buttonDOWN)== HIGH && i1max > i1min){ //Εξήγηση λειτουργίας "Αν
//δεις ένδειξη από τον ακροδέκτη ButtonDOWN Και η μεταβλητή i1min είναι
//μεγαλύτερη από μηδέν
--i1max; //Μείωσε την μεταβλητή του μέγιστου ρεύματος ελέγχου κατά ένα"
} //Τέλος δομής επιλογής

lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
lcd.print("Ithmax="); //Εγγραφή μηνύματος Ithmax= στην LCD οθόνη
lcd.setCursor(0, 1); //Ορισμός σε ποιά τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
lcd.print(i1max); //Εγγραφή του περιεχομένου της μεταβλητής i1max
delay(200); //Καθυστέρηση 200 millisecond

if (digitalRead(buttonMODE)== HIGH){ //Εξήγηση λειτουργίας "Αν δεις ένδειξη από τον
//ακροδέκτη ButtonMODE
lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.setCursor(0, 0); //Ορισμός σε ποιά τοποθεσία θα ξεκινήσει η εγγραφή στην LCD
//οθόνη
lcd.print("WAIT....."); //Εγγραφή μηνύματος WAIT..... στην LCD οθόνη
delay(1000); //Καθυστέρηση 1000 millisecond
break; //Σπάσε την δομή επανάληψης
} //Τέλος δομής επιλογής
} //Τέλος δομής επανάληψης

for (int thisChar = 0; thisChar < 11; thisChar) { //Εναρξη δομής επανάληψης
if(digitalRead(buttonUP)== HIGH){ //Εξήγηση λειτουργίας " Αν δεις ένδειξη από τον
//ακροδέκτη ButtonUP
++t1min; //Αύξησε την μεταβλητή του ελάχιστου χρόνου εκκίνησης κατά ένα"
} //Τέλος δομής επιλογής
if(digitalRead(buttonDOWN)== HIGH && t1min > 0){ //Εξήγηση λειτουργίας "Αν δεις
//ένδειξη από τον ακροδέκτη ButtonDOWN Και η μεταβλητή
//t1min είναι μεγαλύτερη από μηδέν
--t1min; //Μείωσε την μεταβλητή του ελάχιστου χρόνου εκκίνησης κατά ένα"
} //Τέλος δομής επιλογής

lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
lcd.print("Tmin="); //Εγγραφή μηνύματος Tmin= στην LCD οθόνη
lcd.setCursor(0, 1); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
lcd.print(t1min); //Εγγραφή του περιεχομένου της μεταβλητής t1min

```

```

delay(200);           //Καθυστέρηση 200 millisecond

if (digitalRead(buttonMODE)== HIGH){ //Εξήγηση λειτουργίας "Αν δεις ένδειξη από τον
                                     //ακροδέκτη ButtonMODE
    lcd.clear();           //Καθάρισε την LCD οθόνη
    lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
    lcd.print("WAIT....."); //Εγγραφή μηνύματος WAIT..... στην LCD οθόνη
    delay(1000);         //Καθυστέρηση 1000 millisecond
    break;               //Σπάσε την δομή επανάληψης
}                         //Τέλος δομής επιλογής
}                         //Τέλος δομής επανάληψης

t1max=t1min; //Ορίζουμε την μέγιστη τιμή του χρόνου εκκίνησης ίση με την ελάχιστη ώστε
             //ο χρήστης να μην μπορεί να ορίσει ως μέγιστη τιμή μικρότερη από την ελάχιστη

for (int thisChar = 0; thisChar < 11; thisChar) { //Εναρξη δομής επανάληψης
    if(digitalRead(buttonUP)== HIGH){ //Εξήγηση λειτουργίας " Αν δεις ένδειξη από τον
                                       //ακροδέκτη ButtonUP
        ++t1max; //Αύξησε την μεταβλητή του μέγιστου χρόνου εκκίνησης κατά ένα"
    }           //Τέλος δομής επιλογής
    if(digitalRead(buttonDOWN)== HIGH && t1max > t1min){ //Εξήγηση λειτουργίας "Αν
                                                         //δεις ένδειξη από τον ακροδέκτη ButtonDOWN Και η μεταβλητή
                                                         //t1max είναι μεγαλύτερη από t1min
        --t1max; //Μείωσε την μεταβλητή του μέγιστου χρόνου εκκίνησης κατά ένα"
    }           //Τέλος δομής επιλογής
    lcd.clear(); //Καθάρισε την LCD οθόνη
    lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
    lcd.print("Tmax="); //Εγγραφή μηνύματος Tmax= στην LCD οθόνη
    lcd.setCursor(0, 1); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
    lcd.print(t1max); //Εγγραφή του περιεχομένου της μεταβλητής t1max
    delay(200); //Καθυστέρηση 200 millisecond

if (digitalRead(buttonMODE)== HIGH){ //Εξήγηση λειτουργίας "Αν δεις ένδειξη από τον
                                       //ακροδέκτη ButtonMODE
    lcd.clear();           //Καθάρισε την LCD οθόνη
    lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
    lcd.print("WAIT....."); //Εγγραφή μηνύματος WAIT..... στην LCD οθόνη
    delay(1000);         //Καθυστέρηση 1000 millisecond
    break;               //Σπάσε την δομή επανάληψης
}                         //Τέλος δομής επιλογής
}                         //Τέλος δομής επανάληψης

i1=i1min; //Ορίζουμε την τιμή του ρεύματος ίση με την ελάχιστη ώστε ο χρήστης να μην
          //μπορεί να ορίσει ως τιμή ρεύματος μικρότερη από την ελάχιστη
t1=t1min; //Ορίζουμε την τιμή χρόνου ίση με την ελάχιστη ώστε ο χρήστης να μην μπορεί
          //να ορίσει ως τιμή χρόνου εκκίνησης μικρότερη από την ελάχιστη
for (int thisChar1 = 0; thisChar1 < 11; thisChar1) { //Εναρξη δομής επανάληψης
for (int thisChar = 0; thisChar < 11; thisChar) { //Εναρξη δομής επανάληψης
    if(digitalRead(buttonUP)== HIGH && i1max>i1){ //Εξήγηση λειτουργίας " Αν δεις ένδειξη
                                                  //από τον ακροδέκτη ButtonUP και η μεταβλητή i1max είναι μεγαλύτερη από την i1

```

```

    ++i1;          //Αύξησε την μεταβλητή του ρεύματος έλεγχου κατά ένα"
  }              //Τέλος δομής επιλογής
  if(digitalRead(buttonDOWN)== HIGH && i1min<i1){ //Εξήγηση λειτουργίας "Αν δεις
                //ένδειξη από τον ακροδέκτη ButtonDOWN Και η μεταβλητή i1 είναι
                //μεγαλύτερη από i1min
    --i1;        //Μείωσε την μεταβλητή του ρεύματος ελέγχου κατά ένα"
  }              //Τέλος δομής επιλογής
  lcd.clear();   //Καθάρισε την LCD οθόνη
  lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
  lcd.print("Ith="); //Εγγραφή μηνύματος Ith= στην LCD οθόνη
  lcd.setCursor(0, 1); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
  lcd.print(i1);   //Εγγραφή του περιεχομένου της μεταβλητής i1
  delay(200);     //Καθυστέρηση 200 millisecond

  if (digitalRead(buttonMODE)== HIGH){ //Εξήγηση λειτουργίας "Αν δεις ένδειξη από
                                        //τον ακροδέκτη ButtonMODE
    lcd.clear(); //Καθάρισε την LCD οθόνη
    lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
    lcd.print("WAIT....."); //Εγγραφή μηνύματος WAIT..... στην LCD οθόνη
    delay(1000); //Καθυστέρηση 1000 millisecond
    break;      //Σπάσε την δομή επανάληψης
  }             //Τέλος δομής επιλογής
}              //Τέλος δομής επανάληψης

for (int thisChar = 0; thisChar < 11; thisChar) { //Εναρξη δομής επανάληψης
  if(digitalRead(buttonUP)== HIGH && t1max>t1) { //Εξήγηση λειτουργίας " Αν δεις
//ένδειξη από τον ακροδέκτη ButtonUP Και η μεταβλητή μέγιστου χρόνου εκκίνησης είναι
//μεγαλύτερη από τον χρόνο εκκίνησης
    ++t1;          //Αύξησε την μεταβλητή του χρόνο εκκίνησης κατά ένα"
  }              //Τέλος δομής επιλογής
  if(digitalRead(buttonDOWN)== HIGH && t1min<t1){ //Εξήγηση λειτουργίας "Αν
//δεις ένδειξη από τον ακροδέκτη ButtonDOWN Και η μεταβλητή ελάχιστου χρόνου είναι
//μικρότερη από τον χρόνο εκκίνησης
    --t1;         //Μείωσε την μεταβλητή του χρόνο εκκίνησης κατά ένα"
  }              //Τέλος δομής επιλογής
  lcd.clear();    //Καθάρισε την LCD οθόνη
  lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
  lcd.print("T1="); //Εγγραφή μηνύματος T1= στην LCD οθόνη
  lcd.setCursor(0, 1); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
  lcd.print(t1);   //Εγγραφή του περιεχομένου της μεταβλητής t1
  delay(200);     //Καθυστέρηση 200 millisecond

  if (digitalRead(buttonMODE)== HIGH){ //Εξήγηση λειτουργίας "Αν δεις ένδειξη από
//τον ακροδέκτη ButtonMODE
    lcd.clear(); //Καθάρισε την LCD οθόνη
    lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
    lcd.print("WAIT....."); //Εγγραφή μηνύματος WAIT..... στην LCD οθόνη
    delay(1000); //Καθυστέρηση 1000 millisecond
    break;      //Σπάσε την δομή επανάληψης
  }             //Τέλος δομής επιλογής
}

```



```

} //Τέλος δομής επανάληψης
for (int thisChar = 0; thisChar < 11; thisChar) { //Εναρξη δομής επανάληψης
  if(digitalRead(buttonUP)== HIGH && therm==0){ //Εξήγηση λειτουργίας " Αν δεις
//ένδειξη από τον ακροδέκτη ButtonUP Και η μεταβλητή therm είναι ίση με το μηδέν
    therm=1; //Η μεταβλητή therm γίνεται ίση με 1"
  } //Τέλος δομής επιλογής
  if(digitalRead(buttonDOWN)== HIGH && therm==1){ //Εξήγηση λειτουργίας "Αν δεις
//ένδειξη από τον ακροδέκτη ButtonDOWN Και η μεταβλητή therm είναι ίση με το ένα
    therm=0; //Η μεταβλητή therm γίνεται ίση με 0"
  } //Τέλος δομής επιλογής
  lcd.clear(); //Καθάρισε την LCD οθόνη
  lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
  lcd.print("RESTART FROM Ith"); //Εγγραφή μηνύματος RESTART FROM Ith στην
//LCD οθόνη
  lcd.setCursor(0, 1); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
  lcd.print(therm); //Εγγραφή του περιεχομένου της μεταβλητής therm
  delay(200); //Καθυστέρηση 200 millisecond

  if (digitalRead(buttonUP)== HIGH && digitalRead(buttonDOWN)== HIGH){
//Εξήγηση λειτουργίας "Αν δεις ένδειξη από τον ακροδέκτη ButtonMODE Και αν δεις
//ένδειξη από τον ακροδέκτη ButtonDOWN
  lcd.clear(); //Καθάρισε την LCD οθόνη
  lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
  lcd.print("RESET COUNT Ith");//Εγγραφή μηνύματος RESET COUNT Ith στην LCD
//οθόνη
  delay(1000); //Καθυστέρηση 1000 millisecond
  count=0; //Η μεταβλητή count γίνεται ίση με μηδέν
  } //Τέλος δομής επιλογής

  if (digitalRead(buttonMODE)== HIGH){ //Εξήγηση λειτουργίας "Αν δεις ένδειξη από τον
//ακροδέκτη ButtonMODE
  lcd.clear(); //Καθάρισε την LCD οθόνη
  lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
  lcd.print("WAIT....."); //Εγγραφή μηνύματος WAIT..... στην LCD οθόνη
  delay(1000); //Καθυστέρηση 1000 millisecond
  break; //Σπάσε την δομή επανάληψης
  } //Τέλος δομής επιλογής
} //Τέλος δομής επανάληψης

for (int thisChar = 0; thisChar < 11; thisChar) { //Εναρξη δομής επανάληψης
  bailout2: //Δείκτης συνδυασμένος με την εντολή "goto"
  if (digitalRead(buttonUP) == HIGH) { //Εξήγηση λειτουργίας " Αν δεις ένδειξη από τον
//ακροδέκτη ButtonUP
    i=1; //Η μεταβλητή i γίνεται ίση με ένα"

  } //Τέλος δομής επιλογής
  if (digitalRead(buttonDOWN) == HIGH) { //Εξήγηση λειτουργίας "Αν δεις ένδειξη από τον
//ακροδέκτη ButtonDOWN

```

```

i=0; //Η μεταβλητή i γίνεται ίση με μηδέν"
} //Τέλος δομής επιλογής
if (digitalRead(buttonTH) == HIGH || temp1>temp || Irms >c){ //Εξήγηση λειτουργίας "Αν
//δεις ένδειξη από τον ακροδέκτη ButtonTH ή η θερμοκρασία είναι μεγαλύτερη από την τιμή
//ελέγχου θερμοκρασίας ή η τιμή του ρεύματος rms είναι μεγαλύτερη από την μεταβλητή c
j=1; //Η μεταβλητή j γίνεται ίση με ένα"
if(digitalRead(buttonTH) == HIGH) { //Εξήγηση λειτουργίας "Αν δεις ένδειξη από τον
//ακροδέκτη ButtonTH
message="Thermiko"; //Η μεταβλητή message γίνεται ίση με το μήνυμα Thermiko "
} //Τέλος δομής επιλογής
if(temp1>temp) { //Εξήγηση λειτουργίας "Αν η θερμοκρασία είναι μεγαλύτερη από την
//τιμή ελέγχου θερμοκρασίας
message="Thermokrasia";//Η μεταβλητή message γίνεται ίση με το μήνυμα Thermokrasia "
} //Τέλος δομής επιλογής
if(Irms >c) { //Εξήγηση λειτουργίας "Αν η τιμή του ρεύματος rms είναι μεγαλύτερη από
//την μεταβλητή c
message="Reuma"; //Η μεταβλητή message γίνεται ίση με το μήνυμα Reuma "
} //Τέλος δομής επιλογής
} //Τέλος δομής επιλογής
bailout4: //Δείκτης συνδυασμένος με την εντολή "goto"
temp1=0; //Η μεταβλητή temp1 γίνεται ίση με μηδέν "
Irms=0; //Η μεταβλητή Irms γίνεται ίση με μηδέν "
if( j==0 ){ //Εξήγηση λειτουργίας " Αν η μεταβλητή j είναι ίση με 0
test=0; //Η μεταβλητή test γίνεται ίση με μηδέν "
if( i==0 ){ //Εξήγηση λειτουργίας " Αν η μεταβλητή i είναι ίση με μηδέν
digitalWrite(StopLed, HIGH); //Αναψε το Led που αντιστοιχεί στο Stop
digitalWrite(26, LOW); //Κλείσε το ρελέ
digitalWrite(StartLed, LOW); //Σβήσε το Led που αντιστοιχεί στο On
digitalWrite(ThLed, LOW); //Σβήσε το Led που αντιστοιχεί στην ένδειξη του θερμικού

lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
lcd.print("SYSTEM OFF"); //Εγγραφή μηνύματος SYSTEM OFF στην LCD οθόνη
lcd.setCursor(0,1); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
assist=0; //Η μεταβλητή assist γίνεται ίση με μηδέν "
} //Τέλος δομής επιλογής
if( i==1 ){ //Εξήγηση λειτουργίας " Αν η μεταβλητή i είναι ίση με ένα
digitalWrite(StopLed, LOW); //Σβήσε το Led που αντιστοιχεί στο Stop
digitalWrite(26, HIGH); //Ανοιξε το ρελέ
digitalWrite(StartLed, HIGH); //Αναψε το Led που αντιστοιχεί στο On
digitalWrite(ThLed, LOW); //Αναψε το Led που αντιστοιχεί στην ένδειξη του θερμικού
lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD
//οθόνη"
if(assist==1){ //Εξήγηση λειτουργίας " Αν η μεταβλητή assist είναι ίση με ένα
goto bailout5; //Πήγαινε στον δείκτη bailout5
} //Τέλος δομής επιλογής
lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.print("Starting...."); //Εγγραφή μηνύματος Starting.... στην LCD οθόνη

```

```

for (b=0; b <100; b ){           //Εναρξη δομής επανάληψης
  firstmillis=millis(); //Η μεταβλητή firstmillis λαμβάνει την τιμή της μεταβλητής millis
  firstmillis=firstmillis/1000; //Η μεταβλητή firstmillis λαμβάνει την τιμή της μεταβλητής
//firstmillis δια χίλια
  bailout3:                       //Δείκτης συνδυασμένος με την εντολή "goto"
  lcd.clear();                     //Καθάρισε την LCD οθόνη
  lcd.setCursor(0,0);              //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD
//οθόνη
  lcd.print("Starting..Irms="); //Εγγραφή μηνύματος Starting..Irms= στην LCD οθόνη
  lcd.print(Irms);                 //Εγγραφή του περιεχομένου της μεταβλητής Irms
  lcd.setCursor(0,1);              //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD
//οθόνη
  lcd.print("Volt=");             //Εγγραφή μηνύματος Volt= στην LCD οθόνη
  lcd.print(supplyVoltage); //Εγγραφή του περιεχομένου της μεταβλητής supplyVoltage
  lcd.print(" T=");               //Εγγραφή μηνύματος T= στην LCD οθόνη
  lcd.print(temp1);               //Εγγραφή του περιεχομένου της μεταβλητής temp1
  d= mult *i1; //Η μεταβλητή d γίνεται ίση με τον πολλαπλασιασμό των μεταβλητών mult και
//i1
  emon1.calcVI(20,2000); //Υπολόγισε τις τιμές rms τάσης και έντασης
  supplyVoltage = emon1.Vrms; //Κάνε εξαγωγή της Vrms στη μεταβλητή supplyVoltage
  Irms = emon1.Irms; //Κάνε εξαγωγή της Irms στη μεταβλητή Irms
  i2 = analogRead(sensorPin); //Διάβασε την τιμή στον ακροδέκτη sensorPin (όπου είναι
//το θερμόμετρό μας) και τοποθέτησε την τιμή στην μεταβλητή i2

temp1=0.000000000529387*i2*i2*i2-0.00000118191*i2*i2*i2+0.000979104*i2*i2-
0.4556*i2+123.9809; // Υπολογισμός της θερμοκρασίας

if (supplyVoltage >Vmax || supplyVoltage < Vmin){//Εξήγηση λειτουργίας "Αν η μεταβλητή
//supplyVoltage είναι μεγαλύτερη της Vmax Ή η μεταβλητή supplyVoltage είναι μικρότερη
//της Vmin
  message="tasi"; //Η μεταβλητή message γίνεται ίση με tasi
  i=0; //Η μεταβλητή i γίνεται ίση με μηδέν
  j=1; //Η μεταβλητή j γίνεται ίση με ένα
  assist1=1; //Η μεταβλητή assist1 γίνεται ίση με ένα
  goto bailout4; //Πήγαινε στον δείκτη bailout4
} //Τέλος δομής επιλογής

if (digitalRead(buttonTH) == HIGH){//Εξήγηση λειτουργίας "Αν δεις ένδειξη από τον
//ακροδέκτη ButtonTH
  message="Thermiko"; //Η μεταβλητή message γίνεται ίση με Thermiko
  i=0; //Η μεταβλητή i γίνεται ίση με μηδέν
  j=1; //Η μεταβλητή j γίνεται ίση με ένα
  assist=0; //Η μεταβλητή assist γίνεται ίση με μηδέν
  goto bailout4; //Πήγαινε στον δείκτη bailout4
} //Τέλος δομής επιλογής

if (Irms> d){ //Εξήγηση λειτουργίας "Αν δεις τη μεταβλητή Irms να είναι μεγαλύτερη από τη
//μεταβλητή d

```

```

message="REUMA"; //Η μεταβλητή message γίνεται ίση με REUMA
i=0;           //Η μεταβλητή i γίνεται ίση με μηδέν
j=1;           //Η μεταβλητή j γίνεται ίση με ένα
assist1=1;     //Η μεταβλητή assist1 γίνεται ίση με ένα
goto bailout4; //Πήγαινε στον δείκτη bailout4
}              //Τέλος δομής επιλογής

if (temp1>=temp){ //Εξήγηση λειτουργίας "Αν δεις τη μεταβλητή temp1 να είναι
//μεγαλύτερη ή ίση από τη μεταβλητή temp
i=0;           //Η μεταβλητή i γίνεται ίση με μηδέν
j=1;           //Η μεταβλητή j γίνεται ίση με ένα
assist1=1;     //Η μεταβλητή assist1 γίνεται ίση με ένα
message="thermokrasia"; //Η μεταβλητή message γίνεται ίση με thermokrasia
goto bailout4; //Πήγαινε στον δείκτη bailout4
}              //Τέλος δομής επιλογής

testmillis=millis(); //Η μεταβλητή testmillis λαμβάνει την τιμή της μεταβλητής millis
testmillis=testmillis/1000; //Η μεταβλητή testmillis λαμβάνει την τιμή της μεταβλητής
//testmillis δια χίλια

time= testmillis - firstmillis ; //Η μεταβλητή testmillis λαμβάνει την τιμή της μεταβλητής
//testmillis μείον την τιμή της μεταβλητής firstmillis
if (time < t1){ //Εξήγηση λειτουργίας "Αν δεις τη μεταβλητή time να είναι μικρότερη από
//τη μεταβλητή t1

goto bailout3; //Πήγαινε στον δείκτη bailout3
assist1=1;     //Η μεταβλητή assist1 γίνεται ίση με ένα
}              //Τέλος δομής επιλογής

break;         //Σπάσε την δομή επανάληψης
}              //Τέλος δομής επανάληψης
bailout5:      //Δείκτης συνδυασμένος με την εντολή "goto"
emon1.calcVI(20,2000); //Υπολόγισε τις τιμές rms τάσης και έντασης
supplyVoltage = emon1.Vrms; //Κάνε εξαγωγή της Vrms στη μεταβλητή supplyVoltage
Irms = emon1.Irms; //Κάνε εξαγωγή της Irms στη μεταβλητή Irms

if (supplyVoltage >Vmax|| supplyVoltage < Vmin){ //Εξήγηση λειτουργίας "Αν η
//μεταβλητή supplyVoltage είναι μεγαλύτερη της Vmax Ή η μεταβλητή supplyVoltage είναι
//μικρότερη της Vmin
i=0;           //Η μεταβλητή i γίνεται ίση με μηδέν
j=1;           //Η μεταβλητή j γίνεται ίση με ένα
assist1=1;     //Η μεταβλητή assist1 γίνεται ίση με ένα
message="tasi"; //Η μεταβλητή message γίνεται ίση με tasi
goto bailout4; //Πήγαινε στον δείκτη bailout4
}              //Τέλος δομής επιλογής

i2= analogRead(sensorPin); //Διάβασε την τιμή στον ακροδέκτη sensorPin (όπου είναι το
//θερμόμετρό μας)και τοποθέτησε την τιμή στην μεταβλητή i2

```

```
temp1=0.000000000529387*i2*i2*i2*i2-0.00000118191*i2*i2*i2+0.000979104*i2*i2-0.4556*i2+123.9809; // Υπολογισμός της θερμοκρασίας
```

```
lcd.clear(); //Καθάρισε την LCD οθόνη  
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη  
lcd.print("STM ON"); //Εγγραφή μηνύματος STM ON στην LCD οθόνη  
lcd.print(" Irms="); //Εγγραφή μηνύματος Irms= στην LCD οθόνη  
lcd.print(Irms); //Εγγραφή του περιεχομένου της μεταβλητής Irms  
lcd.setCursor(0,2); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη  
lcd.print("Counter="); //Εγγραφή μηνύματος Counter= στην LCD οθόνη  
lcd.print(count); //Εγγραφή του περιεχομένου της μεταβλητής count  
lcd.print("t="); //Εγγραφή μηνύματος t= στην LCD οθόνη  
lcd.print(temp1); //Εγγραφή του περιεχομένου της μεταβλητής temp1  
lcd.print("oC"); //Εγγραφή μηνύματος oC στην LCD οθόνη
```

```
c=mult2 *i1; //Η μεταβλητή c λαμβάνει τιμή ίση με το γινόμενο των μεταβλητών mult2 και //i1
```

```
if (Irms >c ) { //Εξήγηση λειτουργίας "Αν δεις τη μεταβλητή Irms να είναι μεγαλύτερη //από τη μεταβλητή d  
i=0; //Η μεταβλητή i γίνεται ίση με μηδέν  
j=1; //Η μεταβλητή j γίνεται ίση με ένα  
assist=0; //Η μεταβλητή assist γίνεται ίση με μηδέν  
assist1=1; //Η μεταβλητή assist1 γίνεται ίση με ένα  
message="Irms"; //Η μεταβλητή message γίνεται ίση με Irms  
goto bailout4; //Πήγαινε στον δείκτη bailout4  
} //Τέλος δομής επιλογής
```

```
if ( temp1>temp){ //Εξήγηση λειτουργίας "Αν δεις τη μεταβλητή temp1 να είναι //μεγαλύτερη ή ίση από τη μεταβλητή temp
```

```
i=0; //Η μεταβλητή i γίνεται ίση με μηδέν  
j=1; //Η μεταβλητή j γίνεται ίση με ένα  
assist1=1; //Η μεταβλητή assist1 γίνεται ίση με ένα  
message="temp"; //Η μεταβλητή message γίνεται ίση με temp  
assist=0; //Η μεταβλητή assist γίνεται ίση με μηδέν  
goto bailout4; //Πήγαινε στον δείκτη bailout4  
} //Τέλος δομής επιλογής
```

```
assist=1; //Η μεταβλητή assist γίνεται ίση με ένα  
} //Τέλος δομής επιλογής  
} //Τέλος δομής επιλογής
```

```
if( j==1 ) { //Εξήγηση λειτουργίας "Αν δεις τη μεταβλητή j να είναι ίση με ένα  
if (test==0){ //Εξήγηση λειτουργίας "Αν δεις τη μεταβλητή test να είναι ίση με μηδέν  
++count; //Αύξησε την μεταβλητή του μετρητή κατά ένα  
test=1; //Η μεταβλητή test γίνεται ίση με ένα  
} //Τέλος δομής επιλογής
```

```
digitalWrite(26, LOW); //Κλείσε το ρελέ
```

```
digitalWrite(StopLed, HIGH); //Άναψε το Led που αντιστοιχεί στο Stop
```

```
digitalWrite(StartLed, LOW); //Σβήσε το Led που αντιστοιχεί στο On
```

```
digitalWrite(ThLed, HIGH); //Άναψε το Led που αντιστοιχεί στην ένδηξη του θερμικού
```

```
lcd.clear(); //Καθάρισε την LCD οθόνη
```

```
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD
```

```

//οθόνη
  lcd.print("SYSTEM OFF BY "); //Εγγραφή μηνύματος SYSTEM OFF BY στην LCD
//οθόνη

  lcd.setCursor(0,1); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD
//οθόνη
  lcd.print(message); //Εγγραφή του περιεχομένου της μεταβλητής message
  if (therm==0){ //Εξήγηση λειτουργίας "Αν δεις τη μεταβλητή therm να είναι ίση με
//μηδέν
  i=0; //Η μεταβλητή i γίνεται ίση με μηδέν
  j=1; //Η μεταβλητή j γίνεται ίση με ένα
  assist=0; //Η μεταβλητή assist γίνεται ίση με μηδέν
  } //Τέλος δομής επιλογής
  else if (digitalRead(buttonTH) == LOW && assist1==0){ //Εξήγηση λειτουργίας "Αν δεις
//ένδειξη από τον ακροδέκτη ButtonTH Και η μεταβλητή assist1 είναι ίση με το μηδέν
  i=1; //Η μεταβλητή i γίνεται ίση με ένα
  j=0; //Η μεταβλητή j γίνεται ίση με μηδέν
  assist=0; //Η μεταβλητή assist γίνεται ίση με μηδέν
  } //Τέλος δομής επιλογής
  if (digitalRead(buttonDOWN)== HIGH){ //Εξήγηση λειτουργίας "Αν δεις ένδειξη από
//τον ακροδέκτη ButtonDOWN
  delay(1000); //Καθυστέρηση 1000 millisecond
  if (digitalRead(buttonDOWN)== HIGH){ //Εξήγηση λειτουργίας "Αν δεις ένδειξη από
//τον ακροδέκτη ButtonDOWN
  i=0; //Η μεταβλητή i γίνεται ίση με μηδέν
  j=0; //Η μεταβλητή j γίνεται ίση με μηδέν
  assist=0; //Η μεταβλητή assist γίνεται ίση με μηδέν
  assist1=0; //Η μεταβλητή assist1 γίνεται ίση με μηδέν
  goto bailout4; //Πήγαινε στον δείκτη bailout4
  } //Τέλος δομής επιλογής
  } //Τέλος δομής επιλογής
  } //Τέλος δομής επιλογής
  delay(200); //Καθυστέρηση 200 millisecond

while (i==1){ //Εξήγηση λειτουργίας "Όσο η μεταβλητή i είναι ίση με ένα
  goto bailout2; //Πήγαινε στον δείκτη bailout2"
} //Τέλος δομής επανάληψης

```

```

if (digitalRead(buttonMODE)== HIGH){ //Εξήγηση λειτουργίας "Αν δεις ένδειξη από
//τον ακροδέκτη ButtonMODE
  delay(1000); //Καθυστέρηση 1000 millisecond
  if (digitalRead(buttonMODE)== HIGH){ //Εξήγηση λειτουργίας "Αν δεις ένδειξη
//από τον ακροδέκτη ButtonMODE
  lcd.clear(); //Καθάρισε την LCD οθόνη
  lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD
//οθόνη
  i=0; //Η μεταβλητή i γίνεται ίση με μηδέν

```

```

lcd.print("WAIT....."); //Εγγραφή μηνύματος WAIT..... στην LCD οθόνη
i=0; //Κλείσε το ρελέ
digitalWrite(26, LOW); //Σβήσε το Led που αντιστοιχεί στο Stop
digitalWrite(StopLed, LOW); //Σβήσε το Led που αντιστοιχεί στο On
digitalWrite(StartLed, LOW); //Σβήσε το Led που αντιστοιχεί στο On
digitalWrite(ThLed, LOW); //Σβήσε το Led που αντιστοιχεί στην ένδειξη του θερμικού
delay(1000); //Καθυστέρηση 1000 millisecond
lcd.clear(); //Καθάρισε την LCD οθόνη
break; //Σπάσε την δομή επανάληψης
} //Τέλος δομής επιλογής
if (digitalRead(buttonMODE)== LOW){ //Εξήγηση λειτουργίας "Αν δεις ένδειξη από
//τον ακροδέκτη ButtonMODE
lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
lcd.print("Ith="); //Εγγραφή μηνύματος Ith= στην LCD οθόνη
lcd.setCursor(0, 1); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
lcd.print(i1); //Εγγραφή του περιεχομένου της μεταβλητής i1
delay(500); //Καθυστέρηση 500 millisecond
lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
lcd.print("T1="); //Εγγραφή μηνύματος T1= στην LCD οθόνη
lcd.setCursor(0, 1); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
lcd.print(t1); //Εγγραφή του περιεχομένου της μεταβλητής t1
delay(500); //Καθυστέρηση 500 millisecond
lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
lcd.print("Count Ith="); //Εγγραφή μηνύματος Count Ith= στην LCD οθόνη
lcd.setCursor(0, 1); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
lcd.print(count); //Εγγραφή του περιεχομένου της μεταβλητής count
delay(500); //Καθυστέρηση 500 millisecond
lcd.clear(); //Καθάρισε την LCD οθόνη
} //Τέλος δομής επιλογής
} //Τέλος δομής επιλογής
if (digitalRead(buttonUP)== HIGH){ //Εξήγηση λειτουργίας "Αν δεις ένδειξη από τον
//ακροδέκτη ButtonUP
if (digitalRead(buttonDOWN)== HIGH){ //Εξήγηση λειτουργίας "Αν δεις ένδειξη από τον
//ακροδέκτη ButtonDOWN
j=0; //Η μεταβλητή j γίνεται ίση με μηδέν
i=0; //Η μεταβλητή i γίνεται ίση με μηδέν
digitalWrite(26, HIGH); //Αναψε το ρελέ
digitalWrite(StopLed, HIGH); //Αναψε το Led που αντιστοιχεί στο Stop
digitalWrite(StartLed, HIGH); //Σβήσε το Led που αντιστοιχεί στο On
digitalWrite(ThLed, HIGH); //Σβήσε το Led που αντιστοιχεί στην ένδειξη του θερμικού

lcd.clear(); //Καθάρισε την LCD οθόνη
lcd.setCursor(0, 0); //Ορισμός σε ποια τοποθεσία θα ξεκινήσει η εγγραφή στην LCD οθόνη
lcd.print("RESET...."); //Εγγραφή μηνύματος RESET.... στην LCD οθόνη
delay(1000); //Καθυστέρηση 1000 millisecond
digitalWrite(26, LOW); //Κλείσε το ρελέ
digitalWrite(StopLed, LOW); //Σβήσε το Led που αντιστοιχεί στο Stop

```

```
digitalWrite(StartLed, LOW); //Σβήσε το Led που αντιστοιχει στο On
digitalWrite(ThLed, LOW); //Σβήσε το Led που αντιστοιχει στην ένδηξη του θερμικού
goto bailout; //Πήγαινε στον δείκτη bailout
} //Τέλος δομής επιλογής
} //Τέλος δομής επιλογής
} //Τέλος δομής επανάληψης
} //Τέλος δομής επανάληψης
} //Τέλος δομής κώδικα
```