



ΤΕΧΝΟΛΟΓΙΚΟ  
ΕΚΠΑΙΔΕΥΤΙΚΟ  
ΙΔΡΥΜΑ  
ΤΕΙ ΗΠΕΙΡΟΥ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

# Σχεδίαση και κατασκευή συστήματος ελέγχου ροής ύδατος

ΣΤΕΡΓΙΟΥ ΒΑΙΟΣ  
ΤΡΙΑΝΤΑΦΥΛΛΟΣ ΓΙΑΝΝΗΣ

ΑΜ: 9905  
ΑΜ: 10682

Άρτα 2014

## ΔΗΛΩΣΗ ΠΕΡΙ ΛΟΓΟΚΛΟΠΗΣ

Όλες οι προτάσεις οι οποίες παρουσιάζονται σ' αυτό το κείμενο και οι οποίες ανήκουν σε άλλους αναγνωρίζονται από τα εισαγωγικά και υπάρχει η σαφής δήλωση του συγγραφέα. Τα υπόλοιπα γραφόμενα είναι επινόηση του γράφοντος ο οποίος φέρει και την καθολική ευθύνη γι' αυτό το κείμενο και δηλώνουμε υπεύθυνα ότι δεν υπάρχει λογοκλοπή γι' αυτό το κείμενο.

Όνοματεπώνυμο: Στεργίου Βάιος

Υπογραφή.....

Όνοματεπώνυμο: Τριαντάφυλλος Γιάννης

Υπογραφή.....

Ημερομηνία:.....

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Θα θέλαμε να ευχαριστήσουμε τον καθηγητή κ. Αγγέλη Κωσταντίνο, ο οποίος είναι πέραν των άλλων ένας εξαιρετικός άνθρωπος, για την ανάθεση της πτυχιακής εργασίας, την βοήθεια που μας προσέφερε με τις γνώσεις και την εμπειρία του, καθώς και την απαραίτητη τεχνογνωσία για την εκπόνηση της πτυχιακής μας εργασίας.

## **ΠΕΡΙΕΧΟΜΕΝΑ**

Εισαγωγή.....	6
---------------	---

### **ΚΕΦΑΛΑΙΟ 1 – ΕΙΣΑΓΩΓΗ**

1.1 Αυτοματισμός .....	7
1.2 Σκοπός του αυτοματισμού στη ζωή μας .....	7
1.3 Αυτοματισμοί διαχείρισης νερού .....	8
1.4 Τομείς στους οποίους βρίσκουν εφαρμογή εγκαταστάσεις παρόμοιες με τη παρούσα .....	9

### **ΚΕΦΑΛΑΙΟ 2 – ΜΙΚΡΟΕΛΕΓΚΤΕΣ**

2.1 Γενικά .....	10
2.2 Διαφορές από τον μικροεπεξεργαστή .....	10
2.3 Συνήθη υποσυστήματα .....	11
2.4 Πρόσθετες λειτουργίες .....	13
2.5 Διαδεδομένες κατηγορίες μικροελεγκτών .....	14

### **ΚΕΦΑΛΑΙΟ 3 - Arduino**

3.1 Γενικά .....	16
3.2 Πλατφόρμα .....	17
3.3 Λογισμικό .....	17
3.4 Πλεονεκτήματα σε σχέση με άλλους μικροελεγκτές .....	18
3.5 Το υλικό (hardware) του Arduino .....	19
3.6 .....	Χαρακτηριστικά
.....	20

### **ΚΕΦΑΛΑΙΟ 4 - ΗΛΕΚΤΡΟΒΑΛΒΙΔΕΣ**

4.1 Γενικά .....	23
4.2 Αρχή λειτουργίας .....	24

## **ΚΕΦΑΛΑΙΟ 5 - ΑΙΣΘΗΤΗΡΕΣ**

5.1 Γενικά .....	26
5.2	Χαρακτηριστικά
αισθητήρων.....	26
5.3 Κατηγορίες	
αισθητήρων .....	27

## **ΚΕΦΑΛΑΙΟ 6 : ΠΕΡΙΓΡΑΦΗ ΚΑΤΑΣΚΕΥΗΣ**

6.1 Κυκλωματικό σχέδιο κατασκευής .....	28
6.2 Κατάλογος Εξαρτημάτων .....	29
6.3 Εγκατάσταση λογισμικού Arduino .....	35
6.4 Περιγραφή λειτουργίας .....	39
6.5 Υλοποίηση	
κατασκευής.....	39
6.6 Κώδικας .....	47

## **ΚΕΦΑΛΑΙΟ 7 : ΒΙΒΛΙΟΓΡΑΦΙΑ – ΠΑΡΑΓΓΕΛΙΕΣ ΕΞΑΡΤΗΜΑΤΩΝ .....**

Αναφορές.....	67
---------------	----

## Εισαγωγή

Σε μία εποχή όπου η τεχνολογία εξελίσσεται με γοργούς ρυθμούς και κάθε τι γύρω μας αλλάζει με την παρέμβαση της τεχνολογίας προσπαθούμε να μπούμε στην λογική του αυτοματισμού και να κατασκευάσουμε ένα ολοκληρωμένο σύστημα ποτίσματος το οποίο μπορεί να λειτουργεί αυτόνομα χωρίς ανθρώπινη παρεμβολή. Σ αυτή την εργασία θα δούμε την χρησιμότητα του αυτοματισμού και το πόσο καθοριστικό ρόλο παίζει στην ζωή μας, θα αναλύσουμε τους μικροελεγκτές και πιο συγκεκριμένα τον Arduino που χρησιμοποιήσαμε εμείς κι αφού γνωρίσουμε όλα τα απαραίτητα θεωρητικά κομμάτια και την ορολογία γύρω από αυτό το χώρο θα κατασκευάσουμε βήμα-βήμα ένα ολοκληρωμένο σύστημα ελέγχου ροής υδάτων.

## ΚΕΦΑΛΑΙΟ 1<sup>ο</sup> - Αυτοματισμός

Ο αυτοματισμός (ή *αυτοματοποίηση*) αφορά δύο έννοιες σχετιζόμενες μεταξύ τους. Αρχικά, σημαίνει την τυποποίηση μίας διαδικασίας μέσω της εύρεσης καλώς ορισμένων βημάτων τα οποία πρέπει να ακολουθηθούν για να παραχθεί κάποιο επιθυμητό αποτέλεσμα. Έτσι ο αυτοματισμός δεν είναι τίποτα άλλο παρά η εύρεση ενός αλγορίθμου για την επίλυση ενός προβλήματος, ή η κατασκευή ενός αυτόνομου μηχανισμού που εκτελεί αυτόν τον αλγόριθμο για κάποια είσοδο χωρίς ανθρώπινη παρέμβαση.

Ο αυτοματισμός στηρίζεται εννοιολογικά στη θεωρία ελέγχου και στους μηχανισμούς ανάδρασης, επομένως αποτέλεσε μία κύρια αφετηρία της κυβερνητικής. Σε αντίθεση με την τελευταία όμως, ο αυτοματισμός έχει έναν αυστηρά εφαρμοσμένο χαρακτήρα και στην πράξη αξιοποιεί ποικιλία εξειδικευμένων προϊόντων ηλεκτρονικής και τεχνολογίας πληροφοριών (π.χ. μικροελεγκτές, συστήματα πραγματικού χρόνου, τεχνολογίες CAx). Η σημασία του αυτοματισμού είναι μεγάλη στη βιομηχανία, όπου μειώνει σημαντικά την ανάγκη για ανθρώπινη παρέμβαση (π.χ. σε τηλεμετρίες, αυτόματο έλεγχο γραμμών παραγωγής κλπ).

Εξειδικευμένοι υπολογιστές υψηλής αντοχής, οι προγραμματιζόμενοι λογικοί ελεγκτές (PLC), χρησιμοποιούνται για να συγχρονίσουν τη ροή εισόδων από φυσικούς αισθητήρες με τη ροή εντολών προς συσκευές εξόδου (π.χ. βραχίονες).

### 1.2 Σκοπός του αυτοματισμού στη ζωή μας

Μια απλή παρατήρηση των συσκευών που βρίσκονται γύρω μας θα μας πείσει ότι όλες λειτουργούν εν μέρει ή ολικά αυτόματα, δηλαδή χωρίς την παρέμβαση του ανθρώπου. Τέτοιες συσκευές είναι το ηλεκτρικό ψυγείο, το πλυντήριο ρούχων, το θερμοσίφωνο, κ.ά. Ο αυτοματισμός στην καθημερινή ζωή έχει σκοπό να κάνει την

ζωή των ανθρώπων πιο εύκολη. Συγκρίνετε π.χ. τον τρόπο πλυσίματος των ρούχων "στο χέρι" με το πλύσιμο στο ηλεκτρικό πλυντήριο. Αυτοματισμούς μπορεί να συναντήσουμε τόσο στην καθημερινή μας ζωή, όσο και στη βιομηχανική παραγωγή, στις επιστημονικές δραστηριότητες και αλλού.

Τα συστήματα αυτοματισμού ή συστήματα αυτομάτου ελέγχου λειτουργούν χωρίς να χρειάζονται ανθρώπινη επιτήρηση ή παρέμβαση και δρουν κατά προκαθορισμένο τρόπο, ώστε να επιτευχθεί κάποιο επιθυμητό αποτέλεσμα με προκαθορισμένη ακρίβεια. Για παράδειγμα, το σύστημα ελέγχου του κλιματιστικού μας είναι σε θέση, μετά την αρχική του ρύθμιση, να ελέγχει τη λειτουργία του κλιματιστικού μας, ώστε το άμεσο περιβάλλον μας (π.χ. η αίθουσα διδασκαλίας) να έχει σταθερή θερμοκρασία, ανεξάρτητα από τις μεταβολές της εξωτερικής θερμοκρασίας, τον αριθμό των ατόμων κ.ο.κ. και χωρίς να απαιτείται να ξαναρυθμίσουμε το σύστημα, αν δεν αλλάξει η επιθυμητή τιμή της θερμοκρασίας του άμεσου περιβάλλοντος.

Ιδιαίτερη σημασία έχουν οι αυτοματισμοί στην βιομηχανική παραγωγή. Με την αυτοματοποίηση της παραγωγής, οι επιχειρήσεις προσπαθούν να βελτιώσουν την παραγωγικότητά τους μειώνοντας ταυτόχρονα το κόστος παραγωγής. Αυτό επιτυγχάνεται μέσω εξοικονόμησης ενέργειας, καλύτερης διαχείρισης των πρώτων υλών, βελτίωσης της ποιότητας των προϊόντων και μείωσης του κόστους εργασίας. Η ιδέα εφαρμογής συστημάτων αυτοματισμού απορρέει από την ανάγκη να απαλλαγεί ο άνθρωπος από εργασίες επικίνδυνες ή ιδιαίτερα κοπιαστικές. Με τον καιρό όμως επεκτάθηκε σε όλες σχεδόν τις δραστηριότητες της παραγωγικής διαδικασίας, αφού με τα συστήματα αυτοματισμού επιτυγχάνεται μεγαλύτερη ακρίβεια, λιγότερα σφάλματα και μικρότερο κόστος παραγωγής. Έτσι, τα συστήματα αυτοματισμού υποκατέστησαν τον άνθρωπο σε μεγάλο βαθμό στην παραγωγική διαδικασία. Σήμερα λειτουργούν βιομηχανίες, όπου στην παραγωγική διαδικασία η αυτοματοποίηση ξεπερνά το 95%. Στις διάφορες επιστημονικές και ερευνητικές δραστηριότητες (διαστημικά ταξίδια, έρευνα σε μεγάλα θαλάσσια βάθη) απαιτείται ιδιαίτερα μεγάλη ακρίβεια ενεργειών και αποφυγή σφαλμάτων, κάτι που ο άνθρωπος δύσκολα μπορεί να επιτύχει. Και εδώ οι διαδικασίες αυτοματισμού έδωσαν λύσεις προωθώντας σημαντικά την επιστήμη και την τεχνολογία.



### 1.3 Αυτοματισμοί διαχείρισης νερού

Η σωστή διαχείριση των υδάτινων πόρων αποτελεί στη σημερινή εποχή επιτακτική ανάγκη, διότι τα προβλήματα από την έλλειψη του νερού αλλά και από την υποβαθμισμένη ποιότητα του εντείνονται συνεχώς. Ο ρόλος των αυτοματισμών στον τομέα αυτό μεγαλώνει, αφού η εφαρμογή του στα εξωτερικά και εσωτερικά δίκτυα των χωριών και των πόλεων έχει πολλά ευεργετικά αποτελέσματα, όπως:

- Εξοικονόμηση νερού μέσω μείωσης των διαρροών
- Συγκέντρωση και καταγραφή πολλών στατιστικών δεδομένων
- Εξοικονόμηση ηλεκτρικής ενέργειας
- Διατήρηση ή επαναφορά της ποιότητας του νερού στα επιθυμητά επίπεδα

Η εφαρμογή των σύγχρονων αυτοματισμών επιφέρει εξάλλου:

- Αποφυγή πληγμάτων πίεσης
- Αποφυγή βλαβών στους σωλήνες και στις αντλίες
- Αποφυγή της επισφαλούς ενσύρματης επικοινωνίας αντλιοστασίων – Δεξαμενών
- Υλοποίηση σύνθετων αυτοματισμών
- Απλοποίηση ηλεκτρικών πινάκων
- Ευκολία χειρισμών
- Άμεση ενημέρωση σε περιπτώσεις σφαλμάτων

### 1.4 Τομείς στους οποίους βρίσκουν εφαρμογή εγκαταστάσεις παρόμοιες με τη παρούσα.

- Θερμοκήπια
- Διάφορες καλλιέργειες

- Οικιακή χρήση σε φυτά
- Κεντρικά συστήματα ποτίσματος σε δήμους
- Πηγές – υδρομαστεύσεις
- Σταθμούς κεντρικού ελέγχου

## ΚΕΦΑΛΑΙΟ 2<sup>ο</sup> – Μικροελεγκτές

### 2.1 Γενικά

Ο Μικροελεγκτής (microcontroller) είναι ένας τύπος επεξεργαστή, ουσιαστικά μια παραλλαγή μικροεπεξεργαστή, ο οποίος μπορεί να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα, λόγω των πολλών ενσωματωμένων υποσυστημάτων που διαθέτει. Χρησιμοποιείται ευρύτερα σε όλα τα ενσωματωμένα συστήματα (embedded systems) ελέγχου χαμηλού και μεσαίου κόστους, όπως αυτά που χρησιμοποιούνται σε αυτοματισμούς, ηλεκτρονικά καταναλωτικά προϊόντα (από ψηφιακές φωτογραφικές μηχανές έως παιχνίδια), ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα.

### 2.2 Διαφορές από τον Μικροεπεξεργαστή

Στους σύγχρονους μικροεπεξεργαστές για μη ενσωματωμένα συστήματα (όπως τους μικροεπεξεργαστές των προσωπικών υπολογιστών), δίνεται έμφαση στην υπολογιστική ισχύ. Η ευελιξία ανάπτυξης διαφορετικών εφαρμογών είναι μεγάλη, καθώς η λειτουργικότητα του τελικού συστήματος καθορίζεται από τα εξωτερικά περιφερειακά τα οποία διασυνδέονται με την κεντρική μονάδα (μικροεπεξεργαστή), η οποία δεν είναι εξειδικευμένη. Αντίθετα, στους μικροεπεξεργαστές για ενσωματωμένα συστήματα (μικροελεγκτές), οι οποίοι έχουν μικρότερες ή και μηδαμινές δυνατότητες συνεργασίας με εξωτερικά περιφερειακά, αυτού του είδους, η ευελιξία είναι περιορισμένη, καθώς και η υπολογιστική ισχύς. Οι μικροελεγκτές δίνουν έμφαση στο μικρό αριθμό ολοκληρωμένων κυκλωμάτων που απαιτείται για τη λειτουργία μιας συσκευής, το χαμηλό κόστος και την εξειδίκευση.

Αναλυτικά, τα πλεονεκτήματα των μικροελεγκτών είναι :

- Αυτονομία, μέσω της ενσωμάτωσης σύνθετων περιφερειακών υποσυστημάτων όπως μνήμες και θύρες επικοινωνίας. Έτσι πολλοί

μικροελεγκτές δεν χρειάζονται κανένα άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν.

- Η ενσωμάτωση περιφερειακών σημαίνει ευκολότερη υλοποίηση εφαρμογών λόγω των απλούστερων διασυνδέσεων. Επίσης, οδηγεί σε χαμηλότερη κατανάλωση ισχύος, μεγιστοποιώντας τη φορητότητα και ελαχιστοποιεί το κόστος της συσκευής στην οποία ενσωματώνεται ο μικροελεγκτής.
- Χαμηλό κόστος.
- Μεγαλύτερη αξιοπιστία, και πάλι λόγω των λιγότερων διασυνδέσεων.
- Μειωμένες εκπομπές ηλεκτρομαγνητικών παρεμβολών και μειωμένη ευαισθησία σε αντίστοιχες παρεμβολές από άλλες ηλεκτρικές και ηλεκτρονικές συσκευές. Το πλεονέκτημα αυτό προκύπτει από το μικρότερο αριθμό και μήκος εξωτερικών διασυνδέσεων καθώς και τις χαμηλότερες ταχύτητες λειτουργίας.
- Περισσότεροι διαθέσιμοι ακροδέκτες για ψηφιακές εισόδους – εξόδους (για δεδομένο μέγεθος ολοκληρωμένου κυκλώματος) λόγω της μη δέσμευσης τους για τη σύνδεση εξωτερικών περιφερειακών.
- Μικρό μέγεθος συνολικού υπολογιστικού συστήματος.

Η βασική αρχιτεκτονική των μικροελεγκτών δεν διαφέρει από αυτή των κοινών μικροεπεξεργαστών, αν και στους πρώτους είναι απαντάται συχνά η αρχιτεκτονική μνήμης τύπου Harvard, η οποία χρησιμοποιεί διαφορετικές αρτηρίες σύνδεσης της μνήμης προγράμματος και της μνήμης δεδομένης (πχ οι σειρές AVR από την Atmel και PIC από την Microchip). Στους κοινούς μικροεπεξεργαστές συνηθίζεται η ενιαία διάταξη μνήμης τύπου John Von Neumann.

### 2.3 Συνήθη υποσυστήματα

Στον μικροεπεξεργαστή, το ολοκληρωμένο κύκλωμα που τον αποτελεί περιέχει μόνο την Λογική και Αριθμητική Μονάδα (ALU), στοιχειώδεις καταχωρητές (registers), προσωρινή μνήμη RAM πολύ υψηλής ταχύτητας (cache memory) και κάποιες φορές τον ελεγκτή μνήμης (memory controller). Όμως για την λειτουργία ενός πλήρους ενσωματωμένου υπολογιστικού συστήματος, απαιτούνται πολλά εξωτερικά υποσυστήματα και περιφερειακά. Τέτοια είναι :

- Κύκλωμα συνδετικής λογικής (glue logic) για την σύνδεση των εξωτερικών μνημών και άλλων περιφερειακών παράλληλης σύνδεσης στην αρτηρία δεδομένων (bus) του επεξεργαστή.
- Μνήμη προγράμματος ( τύπου ROM, FLASH, EPROM κλπ) η οποία περιέχει το λογισμικό του συστήματος. Σε κάποια μοντέλα, είναι δυνατό το κλείδωμα

αυτής της μνήμης, μετά την εγγραφή της, ώστε να προστατευτεί το περιεχόμενο της από αντιγραφή.

- Μεγάλο μέγεθος μνήμης RAM.
- Μόνιμη μνήμη αποθήκευσης παραμέτρων λειτουργίας (τύπου EEPROM ή NVRAM) η οποία να μπορεί να γράφεται στον πυρήνα του μικροελεγκτή. Αυτή η μνήμη έχει, έναντι της FLASH, το πλεονέκτημα της δυνατότητας διαγραφής και εγγραφής οποιουδήποτε μεμονωμένου byte.
- Κύκλωμα αρχικοποίησης (reset).
- Διαχειριστή αιτήσεων διακόπτης (interrupt request controller) από τα περιφερειακά.
- Κύκλωμα επιτήρησης τροφοδοσίας (brown-out detection) το οποίο παρακολουθεί την τροφοδοσία και αρχικοποιεί ολόκληρο το σύστημα όταν αυτή πέσει κάτω από τα ανεκτά όρια, προλαμβάνοντας έτσι την αλλοίωση των δεδομένων.
- Κύκλωμα επιτήρησης λειτουργίας (watchdog timer) το οποίο αρχικοποιεί το σύστημα, αν αυτό εμφανίσει σημάδια δυσλειτουργίας λόγω κολλήματος (hang).
- Τοπικό ταλαντωτή για την παροχή παλμών χρονισμού (clock).
- Έναν ή περισσότερους χρονιστές – απαριθμητές υψηλής ταχύτητας (hardware timer – counter) για τη δημιουργία καθυστερήσεων, μέτρηση διάρκειας γεγονότων, απαρίθμηση γεγονότων και άλλων λειτουργιών ακριβούς χρονισμού.
- Ρολόι πραγματικού χρόνου (Real Time Clock, RTC) το οποίο τροφοδοτείται από ανεξάρτητη μπαταρία και γι αυτό πρέπει να έχει πολύ χαμηλή κατανάλωση ρεύματος.
- Σειρά ανεξάρτητων ψηφιακών εισόδων και εξόδων (Parallel Input – Output, PIO).

Γενικά, όλες οι οικογένειες μικροελεγκτών ενσωματώνουν τα περισσότερα από τα παραπάνω περιφερειακά, με διαφοροποιήσεις κυρίως στην ύπαρξη ή μη εσωτερικής μνήμης προγράμματος και στο είδος της. Έτσι υπάρχουν :

- Μικροελεγκτές χωρίς μνήμη προγράμματος, οι οποίοι χαρακτηρίζονται ως ROM-less. Αυτοί παρέχουν πάντοτε μια παράλληλη αρτηρία (bus) δεδομένων, πάνω στην οποία συνδέονται εξωτερικές μνήμες προγράμματος και RAM. Τέτοιοι τύποι μικροελεγκτών προορίζονται για πιο ισχυρά υπολογιστικά συστήματα ελέγχου, με μεγαλύτερες απαιτήσεις μνήμης.
- Μικροελεγκτές με μνήμη ROM, η οποία κατασκευάζεται με το λογισμικό της (mask ROM) ή γράφεται μόνο μία φορά (One Time Programmable,

OTP). Παρέχουν τη δυνατότητα πολύ χαμηλού κόστους, όταν αγοράζονται σε πολύ μεγάλες ποσότητες.

- Μικροελεγκτές με μνήμη FLASH, η οποία μπορεί συνήθως να προγραμματιστεί πολλές φορές. Αυτή είναι η πιο δεδομένη κατηγορία. Συχνά ο προγραμματισμός της μνήμης μπορεί να γίνει και πάνω στο κύκλωμα της ίδιας της ενσωματωμένης (embedded) εφαρμογής (δυνατότητα In Circuit Programming, ISP). Αυτοί οι μικροελεγκτές έχουν ουσιαστικά αντικαταστήσει τους παλαιότερους τύπου EPROM που έσβηναν με υπεριώδη ακτινοβολία (από το ειδικό τζαμάκι).

## 2.4 Πρόσθετες Λειτουργίες

Ανάλογα με την εφαρμογή για την οποία προορίζεται ένας μικροελεγκτής, μπορεί να περιέχει και :

- Μία ή περισσότερες ασύγχρονες σειριακές θύρες επικοινωνίας (Universal Asynchronous Receiver Transmitter, UART).
- Σύγχρονες σειριακές θύρες επικοινωνίας.
- Ολόκληρα υποσυστήματα για την άμεση υποστήριξη από υλικολογισμικό (firmware) των πιο σύνθετων πρωτοκόλλων επικοινωνίας όπως CAN, HDLS, ISDN, ADSL.
- Μονάδα άμεσης εκτέλεσης πράξεων κινητής υποδιαστολής (Floating Point Processing Unit, FPU), η οποία είναι πάντοτε πιο γρήγορη από την ALU του επεξεργαστή. Τέτοιες μονάδες χαρακτηρίζουν τους μικροελεγκτές με δυνατότητες ψηφιακής επεξεργασίας σήματος (Digital Signal Processing, DSP). Τα τελευταία χρόνια, με την ευρύτατη διάδοση φορητών συσκευών ήχου και εικόνας, παρατηρείται μία τάση σύγκλισης των μικροελεγκτών με τους DSP.
- Περισσότερες από μία εισόδους για μετατροπή αναλογικού σήματος σε ψηφιακό (Analog to Digital converter, ADC).
- Μετατροπέα ψηφιακού σε αναλογικό σήμα (Digital to Analog converter, DAC).
- Ελεγκτή οθόνης υγρών κρυστάλλων (Liquid Crystal Display, LCD).
- Υποσύστημα προγραμματισμού πάνω στο κύκλωμα (τύπου ISP). Χάρη σε αυτό το κύκλωμα, είναι δυνατός ο επαναπρογραμματισμός (αναβάθμιση λογισμικού) της εφαρμογής, συνδέοντας στην συσκευή μία εξωτερική συσκευή προγραμματισμού (συνήθως σε θύρα UART RS -232) ή ακόμη και από το διαδίκτυο. Αυτή η δυνατότητα απαιτεί την προϋπαρξη λογισμικού υποδοχής (bootstrap) μέσα στη μνήμη προγράμματος και επομένως δεν μπορεί να γίνει σε τελείως άδεια μνήμη προγράμματος.
- Υποσύστημα προγραμματισμού (τύπου ISP) και διάγνωσης (συνήθως είναι το καθιερωμένο πρότυπο JTAG). Χάρη σε αυτό, είναι δυνατός ο

προγραμματισμός της μνήμης, προγραμματισμός χωρίς να απαιτείται κάποιο πρόγραμμα υποδοχής. Γι αυτό το λόγο, είναι ιδιαίτερα χρήσιμο στον αρχικό προγραμματισμό (τύπου ISP) πχ κατά την συναρμολόγηση ή σε περίπτωση σφάλματος (bug) στο λογισμικό υποδοχής το οποίο να καθιστά αδύνατη την κανονική αναβάθμιση.

## 2.5 Διαδεδομένες κατηγορίες μικροελεγκτών

Λόγω του ισχυρότατου ανταγωνισμού αλλά και της τάσης ενσωμάτωσης των μικροελεγκτών σε κάθε ηλεκτρική και ηλεκτρονική συσκευή, η βιομηχανία μικροελεγκτών έχει καταλήξει στην παραγωγή ανταγωνιστικών μοντέλων μαζικής παραγωγής καθώς και μικροελεγκτών για ποιο εξειδικευμένες εφαρμογές. Έτσι διακρίνονται οι εξής κυρίως κατηγορίες :

- Μικροελεγκτές (καμία φορά 4-bit αλλά συνήθως 8-bit) πολύ χαμηλού κόστους, γενικής χρήσης, με πολύ μικρό αριθμό ακροδεκτών (ακόμη και λιγότερους από 8). Σχεδιάζονται με έμφαση στην χαμηλή κατανάλωση ισχύος και την αυτάρκεια, ώστε να χρειάζονται ελάχιστα ή και καθόλου εξωτερικά εξαρτήματα και να μην μπορεί να αντιγραφεί εύκολα το εσωτερικό λογισμικό τους . Απουσιάζει η δυνατότητα επέκτασης της μνήμης τους. Μερικά μοντέλα είναι ευρέως γνωστά στους ερασιτέχνες ηλεκτρονικούς, όπως πχ οι περισσότεροι μικροελεγκτές των σειρών PIC (Microchip), AVR (Atmel) και 8051 (Intel, Atmel, Dallas κα).
- Μικροελεγκτές (συνήθως 8-bit αλλά και 16 ή 32-bit) χαμηλού κόστους, γενικής χρήσης, με μέτριο έως σχετικά μεγάλο αριθμό ακροδεκτών. Διαθέτουν μεγάλο αριθμό κοινών περιφερειακών, όπως θύρες UART, I<sup>2</sup>C, SPI ή CAN, μετατροπείς αναλογικού σε ψηφιακό και ψηφιακού σε αναλογικό. Στους κατασκευαστές της Άνω Ανατολής (Ιαπωνία, Κορέα), συνηθίζεται η ενσωμάτωση ελεγκτών οθόνης υγρών κρυστάλλων και πληκτρολογίου. Μερικές φορές παρέχουν δυνατότητα εξωτερικής επέκτασης της μνήμης τους.
- Μικροελεγκτές (κυρίως 32-bit) μέσου κόστους, γενικής χρήσης, με μεγάλο αριθμό ακροδεκτών. Χαρακτηρίζονται από έμφαση στην ταχύτητα εκτέλεσης εντολών, υψηλή αυτάρκεια περιφερειακών και μεγάλες δυνατότητες εσωτερικής ή εξωτερικής μνήμης προγράμματος (FLASH) και RAM. Στο χώρο αυτό έχουν ισχυρή παρουσία οι αρχιτεκτονικές με υψηλή μεταφερσιμότητα λογισμικού (portability) από τον ένα στον άλλο κατασκευαστή. Πχ μεταξύ

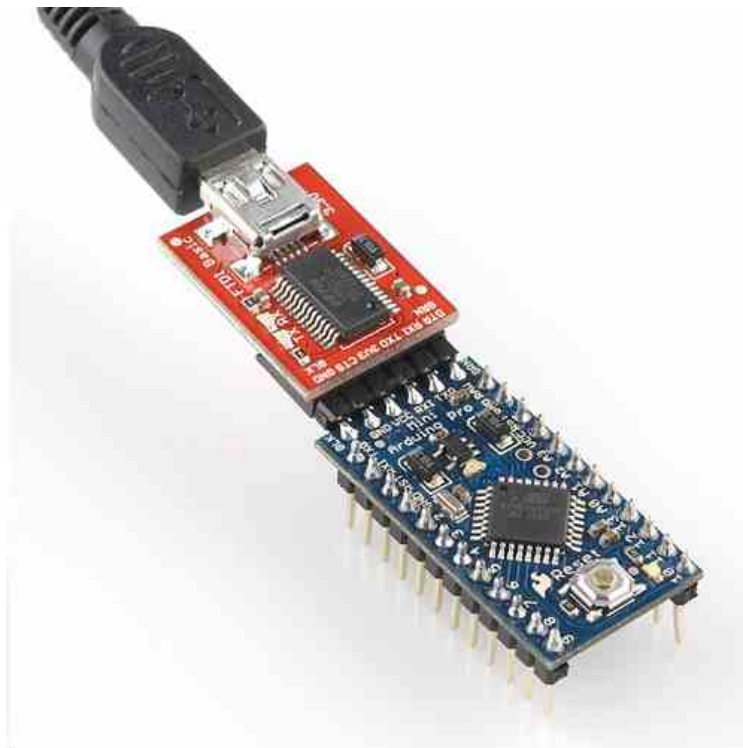
των μικροελεγκτών τύπου ARM ή MIPS, το σύνολο των βασικών εντολών που αναγνωρίζει η ALU είναι ακριβώς το ίδιο, μειώνοντας έτσι τις μεγάλες αλλαγές στο λογισμικό, όταν στο μέλλον ο πελάτης υιοθετήσει ένα μικροελεγκτή άλλου κατασκευαστή (αρκεί, φυσικά, να υποστηρίζει κι αυτός το σύνολο εντολών ARM ή MIPS, αντίστοιχα).

- Μικροελεγκτές εξειδικευμένων εφαρμογών, οι οποίοι ενσωματώνουν συνήθως κάποιο εξειδικευμένο πρωτόκολλο επικοινωνίας το οποίο υλοποιείται πάντοτε σε Hardware. Τέτοιοι μικροελεγκτές χρησιμοποιούνται σε τηλεπικοινωνιακές συσκευές όπως Modem.

Η μεγάλη μερίδα πωλήσεων των μικροελεγκτών εξακολουθεί να αφορά αυτούς των 8-bit, καθώς είναι η κατηγορία με το χαμηλότερο κόστος και το μικρότερο μέγεθος λογισμικού για το ίδιο αποτέλεσμα, ιδίως επειδή οι σύγχρονες οικογένειες μικροελεγκτών 8-bit έχουν πολύ βελτιωμένες επιδόσεις σε σχέση με το παρελθόν.

## ΚΕΦΑΛΑΙΟ 3<sup>ο</sup> – Arduino.

Το Arduino Pro Mini που χρησιμοποιήθηκε στην εργασία μας είναι ένας μικροελεγκτής όπου βασίζεται στην τεχνολογία ATmega168. Διαθέτει 14 ψηφιακές εισόδους / εξόδους (εκ των οποίων 6 μπορεί να χρησιμοποιηθούν ως PWM εξόδους), 8 αναλογικές εισόδους, 4 UARTs (σειριακές θύρες υλικού), έναν ταλαντωτή κρυστάλλου 16MHz, μια σύνδεση USB, υποδοχή τροφοδοσίας, μια επικεφαλίδα ICSP, και ένα κουμπί επαναφοράς.





### 3.1 Γενικά

Το Arduino είναι μια υπολογιστική πλατφόρμα βασισμένη σε μια απλή μητρική πλακέτα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη C++ με κάποιες μετατροπές). 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, Supercollider. Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προ-συναρμολογημένες. Το σχηματικό διάγραμμα και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους.

### 3.2 Πλατφόρμα.

Μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (AT mega328 και AT mega168 στις νεότερες εκδόσεις, AT mega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωση του σε άλλα κυκλώματα. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz (ή κεραμικό αντηχητή σε κάποιες παραλλαγές). Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα bootloader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής. Γενικά όλες οι πλακέτες είναι προγραμματισμένες μέσω μιας σειριακής σύνδεσης RS-232, αλλά ο τρόπος με τον οποίο αυτό υλοποιείται ποικίλλει ανάλογα με την έκδοση. Οι σειριακές πλακέτες Arduino περιέχουν ένα απλό κύκλωμα αντιστροφής για την μετατροπή ανάμεσα στα σήματα των επιπέδων RS-232 και TTL.

Οι πλακέτες Arduino που κυκλοφορούν σήμερα στην αγορά, συμπεριλαμβανόμενης και της Diecimila, προγραμματίζονται μέσω USB, εφαρμόζοντας ένα τσιπ προσαρμογέα USB-to-serial όπως το FTDI FT232. Κάποιες παραλλαγές, όπως το Arduino mini και το ανεπίσημο Boarduino, χρησιμοποιούν προσαρμογέα USB-to-serial σε μορφή πλακέτας ή καλωδίου. Η πλακέτα του Arduino έχει εκτεθειμένες τις περισσότερες επαφές εισόδου/εξόδου για χρήση με άλλα κυκλώματα. Το Diecimila, για παράδειγμα, παρέχει 14 ψηφιακές επαφές εισόδου/εξόδου, από τις οποίες οι 6 μπορούν να παράξουν σήματα PWM, και 6 αναλογικές εισόδους. Αυτές οι επαφές είναι διαθέσιμες στην κορυφή της πλακέτας μέσω θηλυκών συνδέσεων μεγέθους 0,1 ιντσών.

### 3.3 Λογισμικό.

Το IDE του Arduino είναι γραμμένο σε Java και μπορεί να τρέξει σε πολλαπλές πλατφόρμες. Περιλαμβάνει επεξεργαστή κώδικα (επεξεργαστή κειμένου με διάφορα εύχρηστα εργαλεία) και μεταγλωττιστή, και έχει την ικανότητα να φορτώνει εύκολα το πρόγραμμα μέσω σειριακής θύρας από τον υπολογιστή στην πλακέτα.

Το περιβάλλον ανάπτυξης είναι βασισμένο στην Processing, ένα περιβάλλον ανάπτυξης σχεδιασμένο να εισαγάγει στον προγραμματισμό νέους χρήστες μη εξοικειωμένους με την ανάπτυξη λογισμικού. Η συγκεκριμένη γλώσσα προγραμματισμού προέρχεται από την Wiring, μια γλώσσα που μοιάζει με την C η οποία παρέχει παρόμοια λειτουργικότητα για μια πιο περιορισμένης σχεδίασης πλακέτα, της οποίας το περιβάλλον ανάπτυξης βασίζεται επίσης στην Processing.

### 3.4 Πλεονεκτήματα σε σχέση με άλλους μικροελεγκτές.

Υπάρχει πληθώρα άλλων μικροελεγκτών και αναπτυξιακών στο εμπόριο για να ασχοληθεί κάποιος. Ο Basic Stamp της Parallax, ο BX-24 της Net media, το Handgboard του MIT κλπ. Όλα αυτά τα εργαλεία που προαναφέραμε είναι απλά και για τον αρχάριο χρήστη καθώς "κρύβουν" τις δύσκολες λεπτομέρειες της αρχιτεκτονικής και επιτρέπουν τον άμεσο προγραμματισμό του μικροελεγκτή, προσφέροντας τα πάντα σε ένα και μόνο "πακέτο" έτοιμο για χρήση. Ο Arduino διαφέρει από τους προηγούμενους γιατί απλοποιεί την διαδικασία να δουλεύει κάποιος με μικροελεγκτές, αλλά κάποια πλεονεκτήματα που προσφέρει σε σχέση με άλλους μικροελεγκτές για χρήση από δασκάλους, μαθητές και άλλους hobbίστες είναι τα παρακάτω:

- Φθηνός - Οι πλακέτες του Arduino είναι εξαιρετικά φθηνές σε σχέση με άλλες πλατφόρμες μικροελεγκτών. Ειδικά δε μπορεί με τα σχηματικά που κυκλοφορούν στο Internet να κατασκευάσει κάποιος την φθηνότερη εκδοχή ενός Arduino. Ωστόσο ακόμα και αν προμηθευτεί την έτοιμη (μονταρισμένη πλακέτα) αυτή θα κοστίζει το μέγιστο 50€.
- Τρέχει σε διάφορα λειτουργικά Συστήματα. Οι μηχανικοί λογισμικού, ανέπτυξαν το περιβάλλον προγραμματισμού του Arduino για Windows, Macintosh OSX και για λειτουργικά συστήματα Linux. Τα περισσότερα συστήματα ανάπτυξης μικροελεγκτών περιορίζονται στα Windows.

- Απλό, ξεκάθαρο προγραμματιστικό περιβάλλον. Το περιβάλλον προγραμματισμού ενός Arduino ενδείκνυται για αρχάριους, αλλά είναι ταυτόχρονα και ευέλικτο και για πιο προχωρημένους χρήστες.
- Ανοιχτού λογισμικού και λογισμικού που επεκτείνεται και παραμετροποιείται. Το software του Arduino διανέμεται με την μορφή εργαλείων ανοιχτού λογισμικού και είναι διαθέσιμο προς επέκταση για έμπειρους προγραμματιστές. Η γλώσσα προγραμματισμού του μπορεί να επεκταθεί διαμέσου των βιβλιοθηκών την C++ και οι άνθρωποι που θέλουν να ασχοληθούν περισσότερο με τους μικροελεγκτές μπορούν να μεταβούν από τον Arduino στην AVR C που είναι για προγραμματισμό των Atmel μικροελεγκτών και η γλώσσα στην οποία βασίστηκε το λογισμικό του Arduino. Ομοίως μπορεί κάποιος να προσθέσει κώδικα της AVR-C στο πρόγραμμα που έχει γράψει για τον Arduino του.
- Ανοιχτού Υλικού το οποίο μπορεί να επεκταθεί. Ο Arduino βασίζεται στους μικροελεγκτές της Atmel ATMEGA8 και ATMEGA168. Τα σχηματικά για τα αναπτυξιακά είναι κάτω από την άδεια της Creative Commons, επιτρέποντας σε έμπειρους σχεδιαστές να κατασκευάσουν το δικό τους αναπτυξιακό, εξελίσσοντας το ήδη υπάρχον χωρίς να έχουν νομικά προβλήματα. Η ακόμη καλύτερα όχι τόσο έμπειροι χρήστες μπορούν να επιδιώξουν την αντιγραφή και κατασκευή της πλακέτας σε ράστερ για να καταλάβουν την λειτουργία ενός Arduino.

### 3.5 Το υλικό (Hardware) του Arduino.

Ο μικροελεγκτής Arduino είναι στην ουσία μια αναπτυξιακή πλακέτα που ενσωματώνει έναν ATMEGA168. Υπάρχουν πολλές εκδόσεις του μικροελεγκτή (όπως η έκδοση NG και η έκδοση Duemilanove). Η τελευταία έκδοση (2009) είναι αυτή του Arduino Duemilanove. Οι κατασκευαστές του Arduino έχουν τοποθετήσει στις πλακέτες όλα τα απαραίτητα εξαρτήματα για την τροφοδοσία και την διασύνδεση των μικροελεγκτών με τον Ηλεκτρονικό Υπολογιστή. Η πλακέτα μπορεί να τροφοδοτηθεί είτε με τροφοδοτικό των 9Volt, είτε απευθείας από την USB θύρα του Υπολογιστή. Ο Arduino έχει 14 ψηφιακούς ακροδέκτες Εισόδου/Εξόδου. Αυτοί μπορούν να τεθούν ως είσοδοι ή ως έξοδοι με τις εντολές- συναρτήσεις pinMode(), digitalWrite(), and digitalRead(). Λειτουργούν στα 5 Volts και έχουν την δυνατότητα να παρέχουν ή να καταβυθίζουν ένταση της τάξεως των 40mA. Σε κάθε Pin υπάρχει εσωτερικά ένας Pull-up αντιστάτης στα 20-50KΩ. Επιπλέον έχει 6 Αναλογικούς ακροδέκτες Εισόδου. Αυτοί μπορούν να διαβάσουν αναλογικές τιμές όπως η τάση μιας μπαταρίας κτλ και να τις μετατρέψουν σε έναν αριθμό από 0-1023. Η μέτρηση της τάσης γίνεται από προκαθορισμένα από 0 έως 5

volts. Εκτός αυτού 6 εκ των 14 ψηφιακών ακροδεκτών οι P3, P5, P6, P9, P10 και P11 έχουν την δυνατότητα να προγραμματιστούν ώστε να λειτουργούν ως Αναλογικές Έξοδοι.

Κάποιοι ακροδέκτες έχουν συγκεκριμένες λειτουργίες :

- Σειριακή Λειτουργία: 0 (RX) and 1 (TX). Χρησιμοποιούνται για λήψη (RX) και εκπομπή (TX) TTL σειριακών δεδομένων. Αυτοί οι ακροδέκτες είναι συνδεδεμένοι με τους αντίστοιχους του ολοκληρωμένου FTDI USB-to- TTL Serial.
- Εξωτερικές Διακοπές: 2 και 3. Αυτοί οι ακροδέκτες μπορούν να ενεργοποιούν διακοπές αν ανιχνευθεί παλμός χαμηλής τάσης. Με την συνάρτηση `attachInterrupt()`.
- PWM: 3, 5, 6, 9, 10, and 11. Παρέχουν Έξοδο 8-bit PWM με την συνάρτηση `analogWrite()`.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Αυτοί οι ακροδέκτες επιτρέπουν επικοινωνία SPI, η οποία αν και παρέχεται από το hardware δεν είναι ακόμα διαθέσιμη στην γλώσσα προγραμματισμού του Arduino.
- LED: 13. Στον ακροδέκτη 13 υπάρχει ένα ενσωματωμένο LED. Όταν ο ακροδέκτης έχει τιμή HIGH, το LED φωτοβολεί.

Επιπλέον υπάρχουν ακροδέκτες για ειδικές λειτουργίες όπως :

- TWI: 4 (SDA) and 5 (SCL). Υποστηρίζει το πρωτόκολλο I2C (TWI) χρησιμοποιώντας βιβλιοθήκες τις Γλώσσας προγραμματισμού Wiring.

Υπάρχουν και κάποιοι άλλοι ακροδέκτες :

– AREF. Reference voltage for the analog inputs. Χρησιμοποιείται με την συνάρτηση `analogReference()`.

– Reset. Αν τεθεί σε κατάσταση LOW τότε επανεκκινεί τον Μικροελεγκτή. Σε αυτή τη γραμμή τοποθετείται ένας διακόπτης.

### 3.6 Χαρακτηριστικά.

Μικροελεγκτής Atmega	168
Τάση Λειτουργίας	5V
Τάση Εισόδου	7-12V
Όρια Τάσης	6-20V
Ψηφιακοί Ακροδέκτες I/O 1	14 (of which 15 provide PWM output)
Αναλογικοί Ακροδέκτες Εισόδου	8
DC ρεύμα ανά I/O Ακροδέκτη	40 mA
DC ρεύμα για 3.3V Ακροδέκτη	50 mA
Μνήμη Flash	16 KB
SRAM	1 KB
EEPROM	512 bytes
Ταχύτητα Ρολογιού	8 MHz

- **Μνήμη**

Το ολοκληρωμένο ATmega168 έχει 16KB μνήμης flash για την αποθήκευση κώδικα (2 KB εκ των οποίων χρησιμοποιούνται από τον bootloader). Έχει επίσης 1 KB SRAM και 512bytes μνήμες EEPROM (τα οποία μπορούν να διαβαστούν και να γραφτούν με την βιβλιοθήκη EEPROM).

- **Τροφοδοσία**

Το αναπτυξιακό Arduino Pro Mini τροφοδοτείται είτε από εξωτερική τροφοδοσία είτε απευθείας από την θύρα USB. Η επιλογή της πηγής γίνεται αυτόματα από το αναπτυξιακό. Ως εξωτερική τροφοδοσία ορίζεται είτε μια μπαταρία, είτε μετασχηματιστής των 9Volt από 220V. Η μπαταρία μπορεί να συνδεθεί στις υποδοχές του Arduino Vin και GND όπου τοποθετούνται ο θετικός πόλος και ο αρνητικός αντίστοιχα. Από την άλλη αν τροφοδοτήσουμε με μετασχηματιστή απλά τοποθετούμε το βύσμα στην υποδοχή που υπάρχει με τον θετικό πόλο στο κέντρο. Η πλακέτα μπορεί να λειτουργήσει με εξωτερική πηγή από 6 έως 20 Volts. Αν ωστόσο τροφοδοτηθεί με λιγότερα από 7 Volt τα pin εξόδου 5Volt δεν θα καταφέρουν να εξάγουν τάση 5 Volts. Αν από την άλλη δώσουμε πάνω από 12 Volts θα υπερθερμανθεί ο σταθεροποιητής τάσης στην πλακέτα και ενδεχόμενος να καταστραφεί. Συνεπώς μια ιδανική τάση είναι τα 9 Volts.

Οι ακροδέκτες τροφοδοσίας είναι οι εξής:

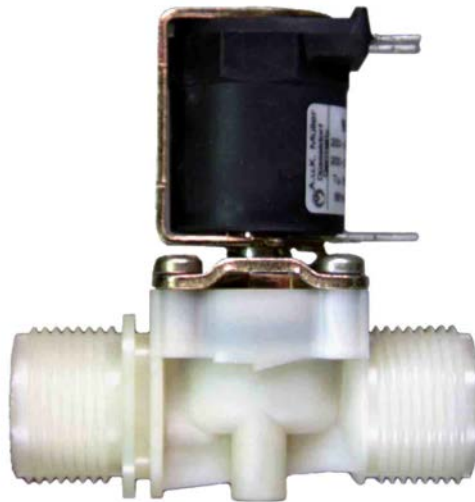
- VIN. Ακροδέκτης για μη σταθεροποιημένη τάση. Συνήθως εδώ συνδέεται μια εξωτερική πηγή τροφοδοσίας.
- 5V. Ακροδέκτης σταθεροποιημένης τάσης 5Volt. Χρησιμοποιείται για την τροφοδοσία του μικροελεγκτή ή άλλων ηλεκτρονικών στοιχείων.
- 3V3. Το ολοκληρωμένο FTDI που βρίσκεται στην πλακέτα του Arduino παράγει τάση των 3.3V με μέγιστο ρεύμα 50mA.
- GND. Ακροδέκτες Γείωσης.

- **Επικοινωνία**

Ο Arduino Pro Mini έχει την δυνατότητα να επικοινωνεί με τον Ηλεκτρονικό Υπολογιστή, έναν άλλον Arduino ή άλλους μικροελεγκτές. Το ολοκληρωμένο ATmega 168 παρέχει σειριακή επικοινωνία TTL 5Volt UART, η οποία είναι διαθέσιμη από τους ακροδέκτες (λήψη RX) 0 και (εκπομπή TX) 1 του ολοκληρωμένου. Επιπλέον στην αναπτυξιακή πλακέτα του Arduino είναι ενσωματωμένο ένα ολοκληρωμένο το FTDI FT232RL το οποίο παρέχει σειριακή επικοινωνία με τον Ηλεκτρονικό Υπολογιστή για προγραμματισμό, πάνω από την θύρα USB με την βοήθεια των ανάλογων FTDI drivers. Οι drivers αυτοί περιλαμβάνονται στο software για τον Arduino και παρέχουν μια ιδεατή θύρα επικοινωνίας στον Ηλεκτρονικό Υπολογιστή για τους σκοπούς της επικοινωνίας.

## ΚΕΦΑΛΑΙΟ 4<sup>ο</sup> – Ηλεκτροβάνες.

### 4.1 Γενικά.



Μία ηλεκτροβάνα (ηλεκτροβαλβίδα) είναι μία βαλβίδα που λειτουργεί ηλεκτρομηχανικά. Η βαλβίδα αυτή ελέγχεται από ένα ηλεκτρικό ρεύμα μέσα από ένα σωληνοειδές πηνίο.

Ηλεκτρομαγνητικές βαλβίδες μπορούν να έχουν δύο ή περισσότερους εισόδους-εξόδους. Στην περίπτωση των δύο εισόδων-εξόδων η μια είναι είσοδος και η άλλη έξοδος. Στην περίπτωση των τριών εισόδων-εξόδων της βαλβίδας η εισροή γίνεται από μία είσοδο και η εκροή από τις δύο εξόδους. Πολλαπλές ηλεκτρομαγνητικές βαλβίδες μπορούν να τοποθετηθούν μαζί σε ένα συλλέκτη.

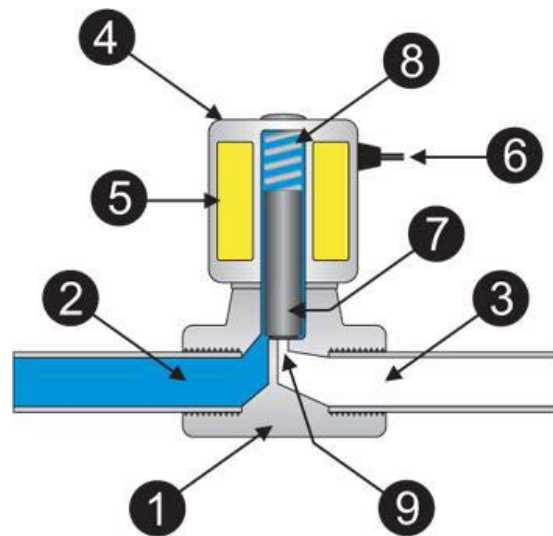
Ηλεκτρομαγνητικές βαλβίδες είναι τα πιο συχνά χρησιμοποιούμενα στοιχεία ελέγχου ρευστών και χρησιμοποιούνται σε πολλές εφαρμογές. Τα καθήκοντά τους είναι να σταματούν ή να επιτρέπουν τη ροή υγρών. Τα σωληνοειδή προσφέρουν γρήγορη και ασφαλή μεταγωγή, υψηλή αξιοπιστία, μεγάλη διάρκεια ζωής, καλό μέσο συμβατότητας των υλικών που χρησιμοποιούνται, έχουν χαμηλή ισχύς ελέγχου και συμπαγή σχεδιασμό.

#### 4.2 Αρχή Λειτουργίας.

Μια ηλεκτρομαγνητική βαλβίδα έχει δύο κύρια μέρη: το σωληνοειδές και τη βαλβίδα. Το σωληνοειδές μετατρέπει ηλεκτρική ενέργεια σε μηχανική ενέργεια το οποίο, με τη σειρά του, ανοίγει ή κλείνει τη βαλβίδα μηχανικά. Η παρακάτω εικόνα απεικονίζει τα βασικά συστατικά μιας ηλεκτρομαγνητικής βαλβίδας. Η βαλβίδα που φαίνεται στην εικόνα είναι κλειστή σε ηρεμία (normally close), άμεσης ενεργοποίησης βαλβίδα. Αυτός ο τύπος ηλεκτρομαγνητικής βαλβίδας έχει τον πιο απλό και κατανοητό τρόπο λειτουργίας.

Η ηλεκτρομαγνητική βαλβίδα αποτελείται από τα μέρη :

1. Σώμα βαλβίδας
2. Στόμιο εισόδου
3. Στόμιο εξόδου
4. Σωληνοειδές(πηνίο)
5. Τυλίγματα
6. Καλώδια τροφοδοσίας
7. Έμβολο
8. Ελατήριο
9. Στόμιο



*Βασικά συστατικά ηλεκτροβαλβίδας*



Τα κύρια μέρη της παραπάνω ηλεκτροβαλβίδας είναι το σύστημα του πηνίου και του εμβόλου:

- Το σύστημα του πηνίου είναι αυτό που ενεργοποιείται από τον ηλεκτρονικό προγραμματιστή προκειμένου να ασφαλίσει ή να απασφαλίσει την ηλεκτροβαλβίδα. Η λειτουργία του βασίζεται σε απλούς φυσικούς νόμους του ηλεκτρισμού, αλλά δεν είναι καθόλου απλό να κατασκευαστεί ένα αξιόπιστο και ανθεκτικό πηνίο ώστε να ανταπεξέρχεται στο χρόνο και τις φθορές. Τα πηνία είναι η achilles πέτρα των ηλεκτροβαλβίδων και η ποιότητα της κατασκευής τους χαρακτηρίζει καίρια την ποιότητα αυτών των βαλβίδων. Έχει τεράστια σημασία το σύστημα του πηνίου να είναι ένα συμπαγές τμήμα με πολύ καλή υδρομόνωση και δυνατότητα φιλτραρίσματος του νερού. Τα διάφορα πηνία συνήθως λειτουργούν σε τάση είκοσι τεσσάρων ή δώδεκα Volts αναλόγως του τύπου της εγκαταστάσεως που έχουμε επιλέξει. Πρέπει να διευκρινίσουμε ότι οι περισσότερες σύγχρονες ηλεκτροβαλβίδες μπορούν να ενεργοποιηθούν και χειροκίνητα με μερική στροφή των πηνίων τους.



*Ηλεκτροβαλβίδα τύπου Solenoid 1*

- Το σύστημα του διαφράγματος βρίσκεται στην καρδιά του μηχανισμού της ηλεκτροβαλβίδας. Στην ουσία λειτουργεί σαν διαχωριστικό ανάμεσα σε δύο θαλάμους επηρεάζοντας τα χαρακτηριστικά ροής του νερού με βάση την κατάσταση λειτουργίας του πηνίου. Όλη η διαδικασία η οποία ενεργοποιεί ή διακόπτει τη ροή ύδατος οφείλεται στη δυνατότητα που παρέχεται στο νερό να ρέει μέσα από μικρής διαμέτρου οπές οι οποίες βρίσκονται στο

διάφραγμα και τα υπόλοιπα τμήματα της βαλβίδας. Για αυτό έχει μεγάλη σημασία να υπάρχουν στρατηγικά τοποθετημένα φίλτρα μέσα τους προκειμένου να αποτρέπουν φερτά υλικά (πχ πετραδάκια) από το να φράσσουν αυτές τις οπές.

## ΚΕΦΑΛΑΙΟ 5<sup>ο</sup> – Αισθητήρες.

### 5.1 Γενικά.

Ως αισθητήρα χαρακτηρίζουμε μια συσκευή η οποία ανιχνεύει ένα φυσικό μέγεθος και στην συνέχεια σύμφωνα με αυτό παράγει μια μετρήσιμη έξοδο. Ο κάθε αισθητήρας χρησιμοποιεί διαφορετικές μεθόδους για την ανίχνευση των μεγεθών όμως όλοι οι αισθητήρες έχουν κάποια κοινά χαρακτηριστικά.

### 5.2 Χαρακτηριστικά αισθητήρων.

Τα βασικά κοινά χαρακτηριστικά των αισθητήρων είναι τα εξής :

- **Εύρος** Τα όρια στα οποία η συσκευή λειτουργεί αξιόπιστα.
- **Ακρίβεια** Η εγγύτητα της τιμής εξόδου προς τη τιμή εισόδου.
- **Σφάλμα** Η διαφορά ανάμεσα στη μετρούμενη τιμή και τη πραγματική τιμή.
- **Ανοχή** Το μέγιστο σφάλμα που μπορεί να δημιουργήσει ο αισθητήρας.
- **Διακριτική ικανότητα** Η μικρότερη αλλαγή τιμής εισόδου που μπορεί να ανιχνεύσει.
- **Ευαισθησία** Η σχέση της αλλαγής εξόδου προς τη αλλαγή εισόδου, είναι ίση με τη διαφορά των τιμών της εξόδου προς τη διαφορά των αντίστοιχων τιμών εισόδου.
- **Βαθμονόμηση** Η βαθμολόγηση της κλίμακας σε μονάδες.
- **Νεκρή ζώνη** Το μέγιστο ποσό αλλαγής της εισόδου που δεν επιφέρει αλλαγή στην έξοδο.
- **Γραμμικότητα** Ο βαθμός στον οποίο η γραφική παράσταση της εξόδου προσεγγίζει ευθεία ως προς την είσοδο του αισθητήρα.
- **Απόκριση** Ο χρόνος που απαιτείται για να λάβει τη τελική τιμή η έξοδος.

- **Καθυστέρηση** Η καθυστέρηση της αλλαγής της εξόδου ως προς την είσοδο.
- **Ευστάθεια** Η μεταβολή της εξόδου σε μεγάλη χρονική περίοδο, χωρίς μεταβολή της εισόδου και των συνθηκών.
- **Υστέρηση** Η διαφορά στην έξοδο όταν η κατεύθυνση της μεταβολής της εισόδου αντιστραφεί.
- **Επαναληψιμότητα** Η παραγωγή του ίδιου αποτελέσματος, σε διαφορετικές χρονικές στιγμές, με την ίδια είσοδο.
- **Ολίσθηση** Η μεταβολή των χαρακτηριστικών του αισθητήρα με το χρόνο και το περιβάλλον.
- **Στατικό σφάλμα** Σταθερό σφάλμα σε όλο το εύρος λειτουργίας, το οποίο μπορεί να αντισταθμιστεί.
- **Χρόνος λειτουργίας** Ο εκτιμώμενος χρόνος λειτουργίας στα πλαίσια των προδιαγραφών του.

### 5.3 Κατηγορίες αισθητήρων.

Τους αισθητήρες θα μπορούσαμε να τους χωρίσουμε σε δύο κατηγορίες ανάλογα με το είδος του ηλεκτρικού σήματος που παρέχουν στην έξοδο τους. Έτσι θα έχουμε:

#### 1. Αισθητήρες αναλογικής εξόδου

Όπου το σήμα στην έξοδο είναι συνεχές και ανάλογο του πλάτους τάσης του αισθητήρα.

#### 2. Αισθητήρες ψηφιακής εξόδου

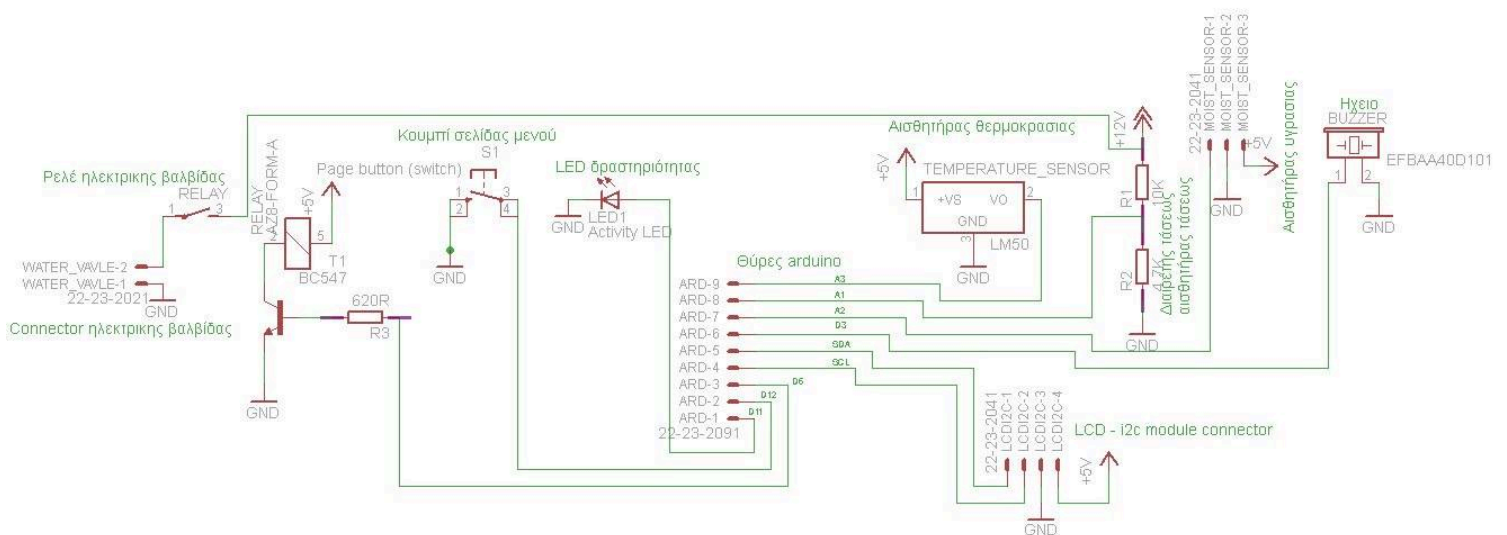
Όπου το σήμα στην έξοδο είναι ψηφιακό, παίρνει δηλαδή διακριτές τιμές υπό την μορφή δυαδικού αριθμού ή σειράς παλμών.

Η βασική διαφορά ανάμεσα σ αυτές τις δυο κατηγορίες που χωρίσαμε τους αισθητήρες είναι ότι στην πρώτη κατηγορία έχουμε μεγαλύτερο ποσοστό σφάλματος κι αυτό γιατί το ηλεκτρικό σήμα που δίνουν μπορεί να αλλοιωθεί από τον ηλεκτρικό θόρυβο.

## ΚΕΦΑΛΑΙΟ 6<sup>ο</sup> – Περιγραφή κατασκευής.

### 6.1 Κυκλωματικό σχέδιο κατασκευής.

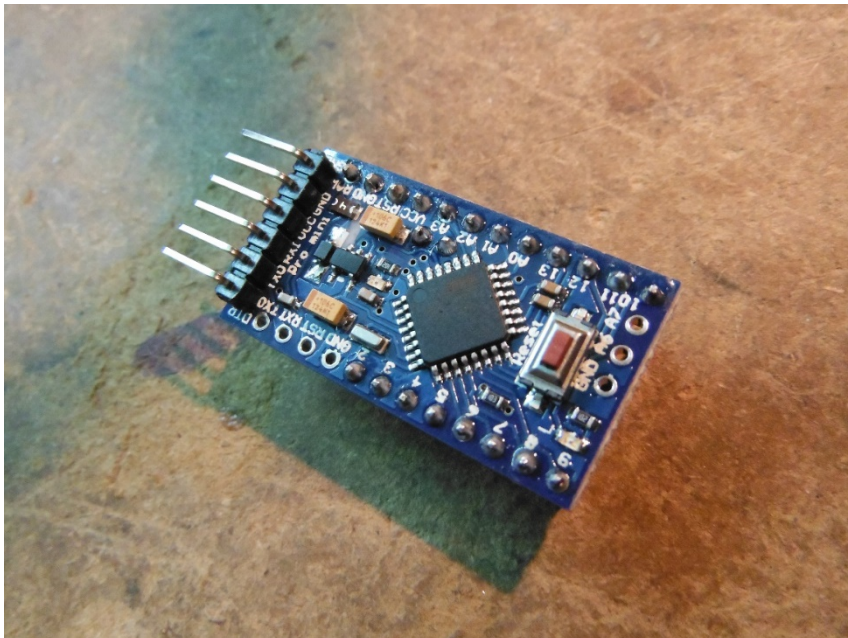
Η λειτουργία της κατασκευής μας έχει σκοπό να ελέγξει τις συνθήκες περιβάλλοντος και του χώματος και με την χρήση ηλεκτροβάννας να αναλάβει το πότισμα χωρίς ανθρώπινη παρεμβολή.



## 6.2 Κατάλογος εξαρτημάτων.

Αναλυτική περιγραφή των εξαρτημάτων που χρησιμοποιήσαμε για την κατασκευή.

- Αναπτυξιακή πλακέτα Arduino



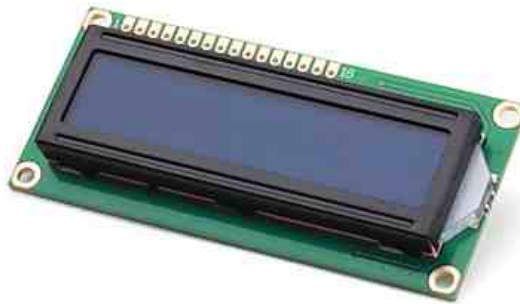
*Arduino Pro Mini 1*

Το Arduino Pro Mini που χρησιμοποιήσαμε για την κατασκευή μας είναι ένας μικροελεγκτής που βασίζεται στην τεχνολογία ATmega328 ο οποίος διαθέτει 14 ψηφιακές εισόδους/εξόδους και 8 αναλογικές εισόδους. Έχει μνήμη SRAM 1KB και EPROM 512bytes και συχνότητα ρολογιού 16MHz. Επάνω στο Pro Mini βρίσκεται ένα reset button και 6 pin headers όπου συνδέσαμε τον αντάπτορα της Sparkfun για να έχουμε διασύνδεση USB στον Arduino Mini Pro.



*Sparkfun's FTDI Basic Breakout adaptor1*

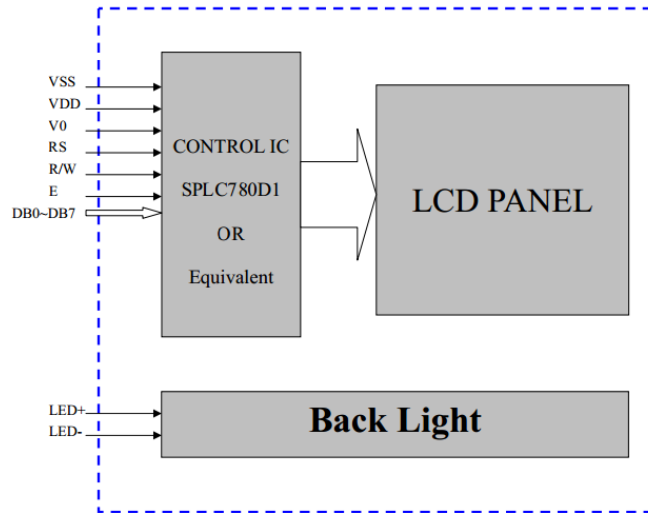
- **Οθόνη LCD 1602**



*1602 LCD display*

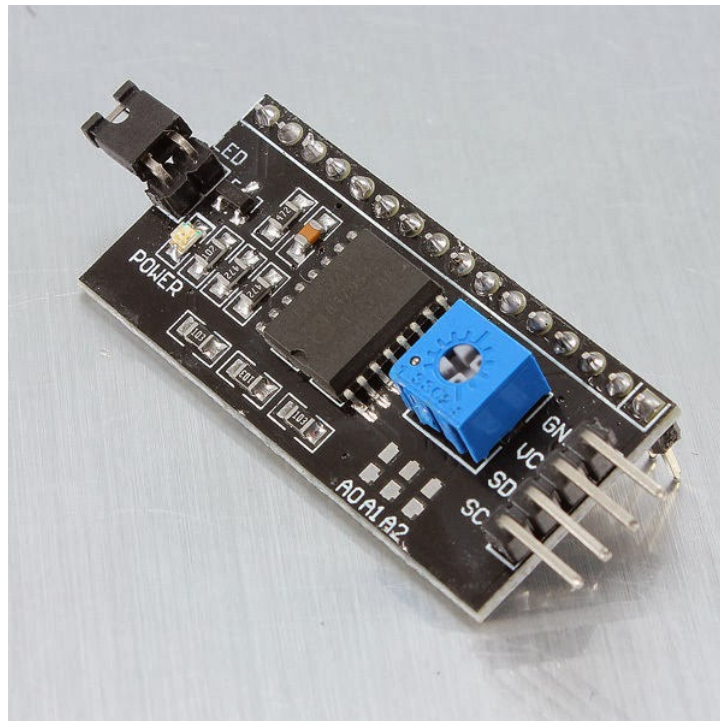
Για να μπορούμε να έχουμε μία έξοδο όπου θα παίρνουμε διαγνωστικά μηνύματα και θα μπορούμε να προβάλλουμε τιμές, μετρήσεις αλλά κι ένα μενού επιλογών χρησιμοποιήσαμε την 1602 LCD display. Το 1602 στην ονομασία της μας δείχνει ότι μπορεί να απεικονίσει 2 γραμμές των 16 χαρακτήρων η κάθε μια. Για να απεικονίσει τους χαρακτήρες χρησιμοποιεί τον HD44780U controller κουκίδων πίνακα της οθόνης υγρών κρυστάλλων.

Σχεδίαση και κατασκευή συστήματος ελέγχου ροής ύδατος.



*Σχηματικό οθόνης LCD*

Για να αποφύγουμε τον μεγάλο όγκο καλωδίωσης, να περιορίσουμε τον αριθμό των θυρών σε χρήση του Mini Pro αλλά και για να μικρύνουμε τον όγκο κατασκευής χρησιμοποιήσαμε ένα δίαυλο I2C interface module.



*I2C interface module*

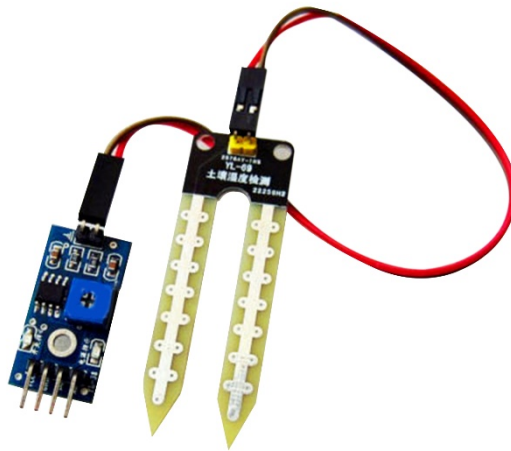






## Σχεδίαση και κατασκευή συστήματος ελέγχου ροής ύδατος.

καλός αγωγός του ηλεκτρισμού και μετράει την αντίσταση ανάμεσα στους δυο ακροδέκτες. Η ευαισθησία ρυθμίζεται από ένα ποτενσιόμετρο. Η τάση λειτουργίας είναι από 3V ως 5V.



Αισθητήρας υγρασίας

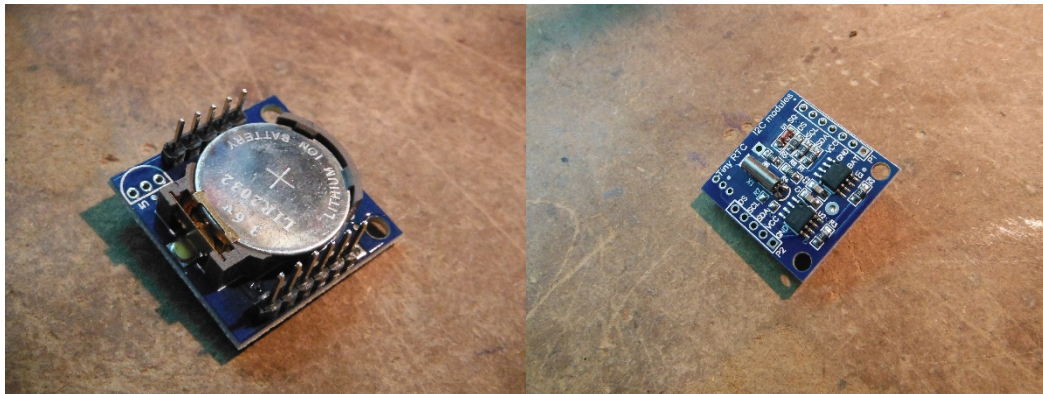


Πειραματικές μετρήσεις απ την κατασκευή

- **DS1302 Timekeeping Chip**

Δεδομένου του ότι έπρεπε να γνωρίζουμε συνέχεια την ώρα έτσι ώστε να καθορίζουμε το πότε θα γίνεται το πότισμα χρησιμοποιήσαμε ένα ρολόι πραγματικού χρόνου το DS1302.

Με μια μικρή μπαταρία μπορεί να κρατήσει στην μνήμη δεδομένα ακόμη κι όταν η κατασκευή βρίσκεται εκτός λειτουργίας.



- **Ηλεκτροβαλβίδα.**



*Σωληνοειδής ηλεκτροβαλβίδα*

Στις ηλεκτροβαλβίδες αναφερθήκαμε αναλυτικότερα σε προηγούμενο κεφάλαιο εδώ μόνο θα επισημάνουμε ότι η συγκεκριμένη δεν είναι ελεύθερης ροής χρειάζεται δηλαδή πίεση για να ανοίξει η βαλβίδα και να περάσει το νερό.

Η τάση λειτουργίας της είναι στα 12V DC με μέση κατανάλωση περίπου 250mA.

- **1 τροφοδοτικό 12V D/C , 1A**

Χρησιμοποιούμε εξωτερική τροφοδοσία για να διεγείρουμε την βαλβίδα.

- **1 ρελέ 6V D/C**

Το οποίο διεγείρεται στα 6V με τη χρήση της εξόδου 7 του μικροελεγκτή. Έχοντας μόνιμα τα 12V στις επαφές του ρελέ, όταν αυτό διεγερθεί, οι επαφές του γεφυρώνονται και η τάση των 12V πηγαίνει στα άκρα της βαλβίδας.

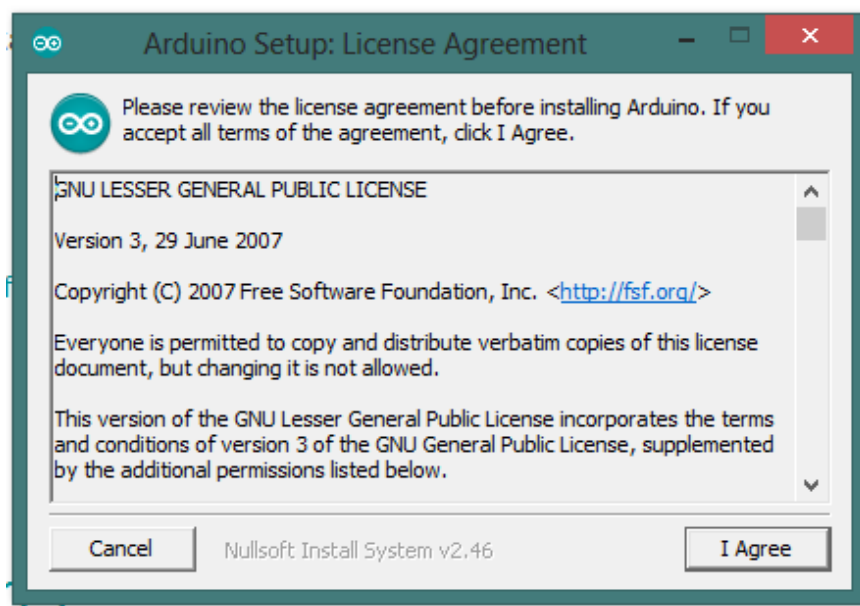
- **Διάφορα καλώδια και κουμπιά**

### 6.3 Εγκατάσταση λογισμικού Arduino

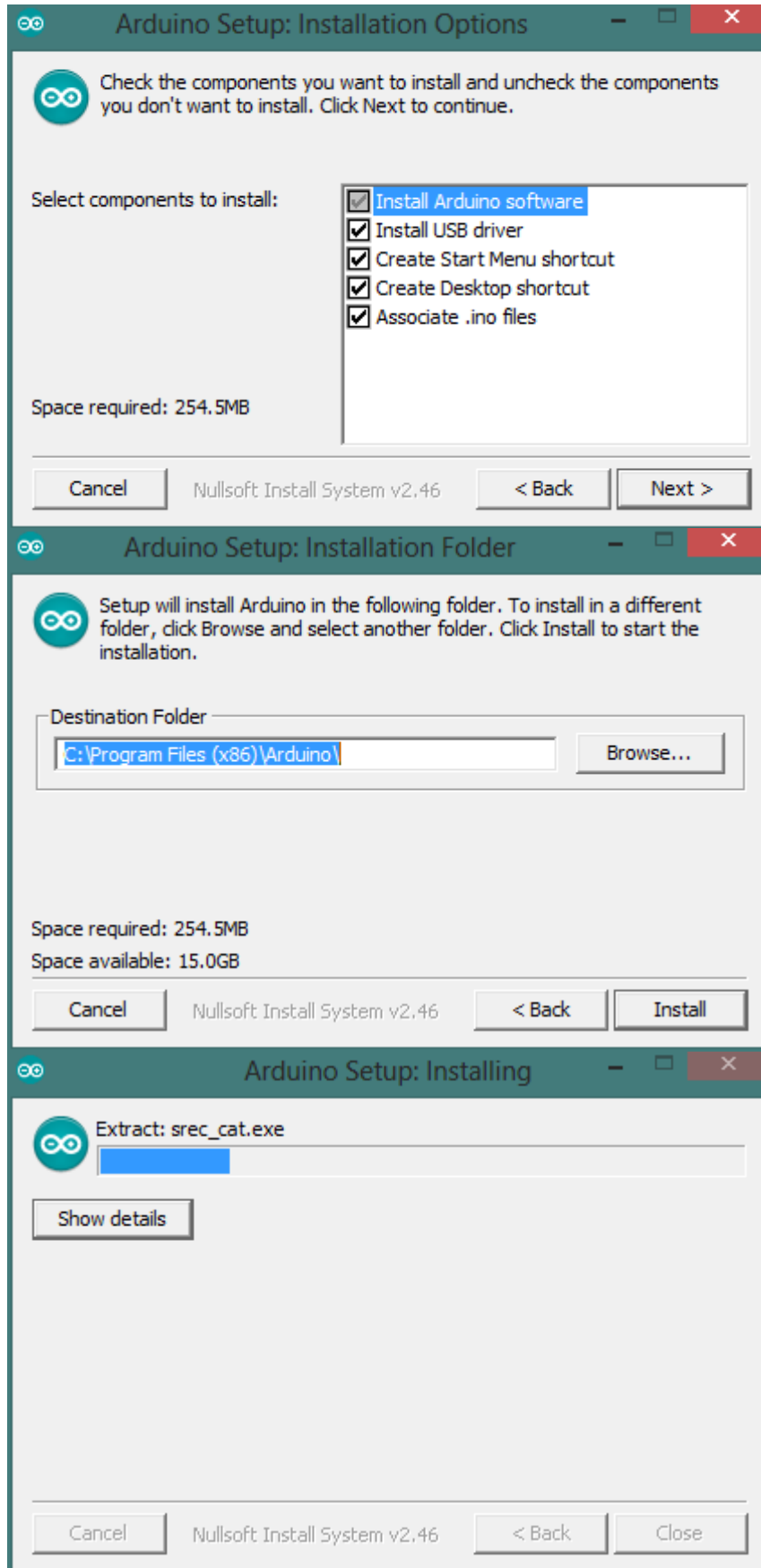
Το λογισμικό του Arduino βρίσκεται στην ιστοσελίδα <http://arduino.cc/en/Main/Software> δωρεάν για κατέβασμα με υποστήριξη σε όλες τις πλατφόρμες (Windows, Mac, Linux) για 32 και 64 bit εκδόσεις αντίστοιχα.

Πάμε να δούμε βήμα-βήμα την εγκατάσταση σε περιβάλλον windows:

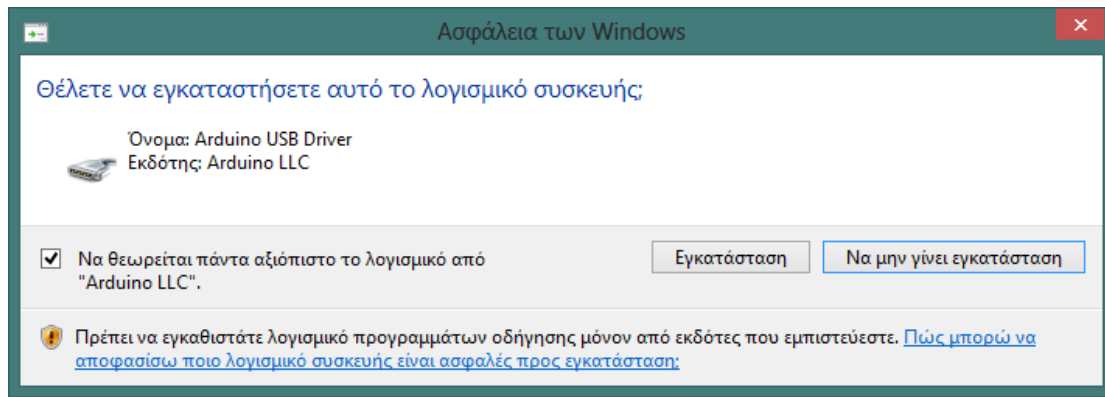
Τρέχουμε το αρχείο που έχουμε κατεβάσει με όνομα "arduino-1.0.5-r2-windows"



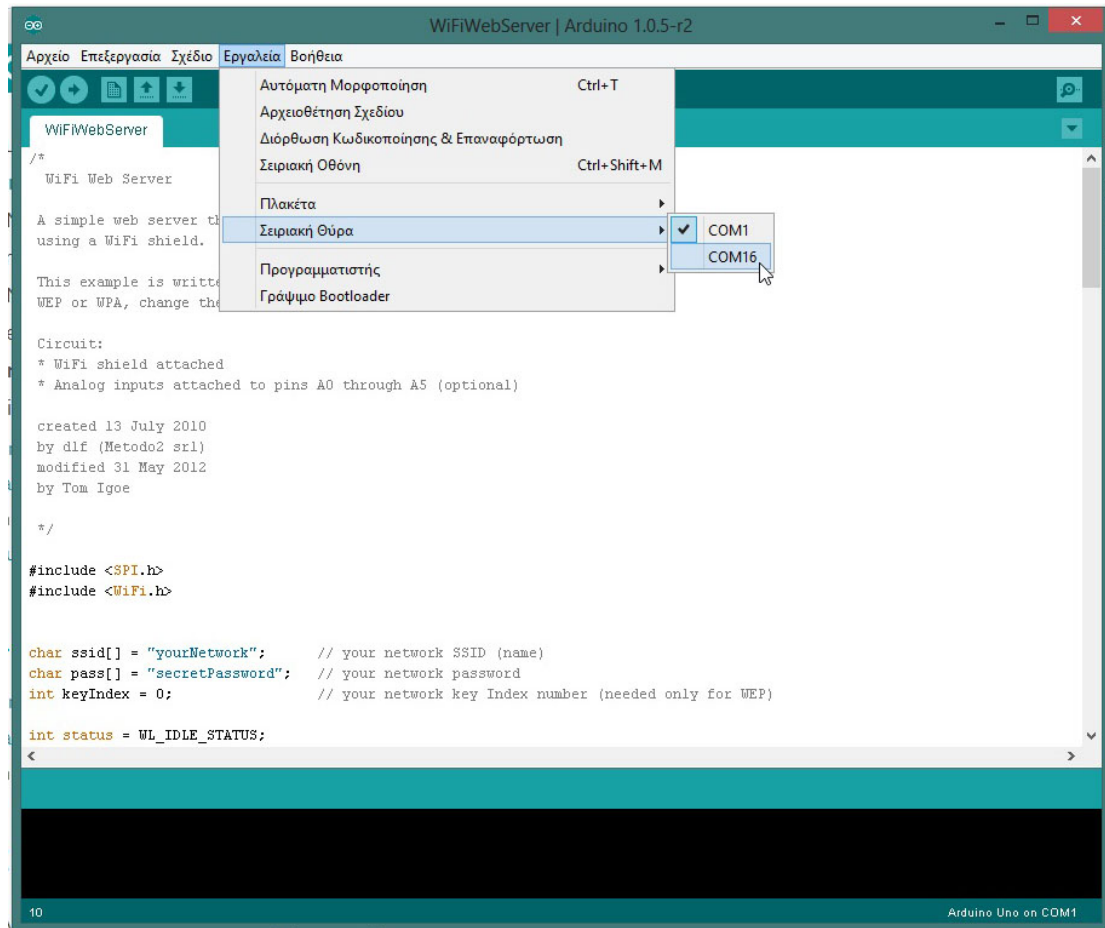
Κι ακολουθούμε βήμα-βήμα όπως και στις εικόνες:



## Σχεδίαση και κατασκευή συστήματος ελέγχου ροής ύδατος.

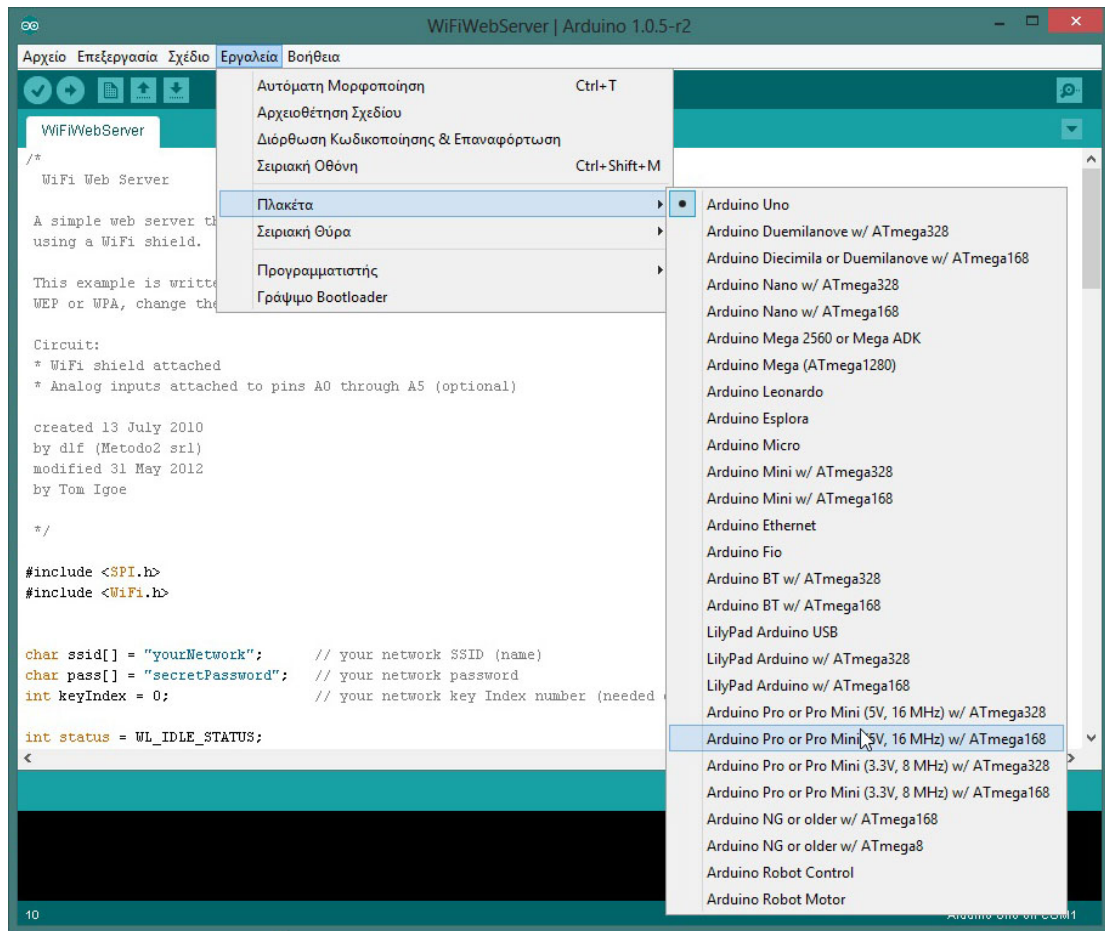


Μετά την εγκατάσταση εκτελούμε το πρόγραμμα όπου επιλέγουμε αρχικά την θύρα COM που θα συνδέσουμε το Arduino καθώς και την πλακέτα που θα χρησιμοποιήσουμε.





## Σχεδίαση και κατασκευή συστήματος ελέγχου ροής ύδατος.



Μετά την ολοκλήρωση των παραπάνω βημάτων σειρά είχε η αντιγραφή στον φάκελο προγραμμάτων των βιβλιοθηκών που ήταν απαραίτητες για να 'διαβάσει' το πρόγραμμα κάποια απ τα εξαρτήματά μας όπως ο προσαρμογέας I2C για την χρήση πολυπλεξίας στην σύνδεση της οθόνης. Όλες οι βιβλιοθήκες βρίσκονται δωρεάν στο ίντερνετ και ίσως διαφέρουν για κάθε κατασκευαστή.

## 6.4 Περιγραφή λειτουργίας

Η λειτουργία του συστήματός μας έχει ως εξής:

Ένας αισθητήρας υγρασίας κι ένα θερμόμετρο λαμβάνουν μετρήσεις τις οποίες επεξεργάζονται κι εφόσον οι μετρήσεις είναι μέσα στα όρια που εμείς έχουμε καθορίσει τότε το Arduino δίνει εντολή στην ηλεκτροβαλβίδα να ξεκινήσει το πότισμα.

## 6.5 Υλοποίηση Κατασκευής

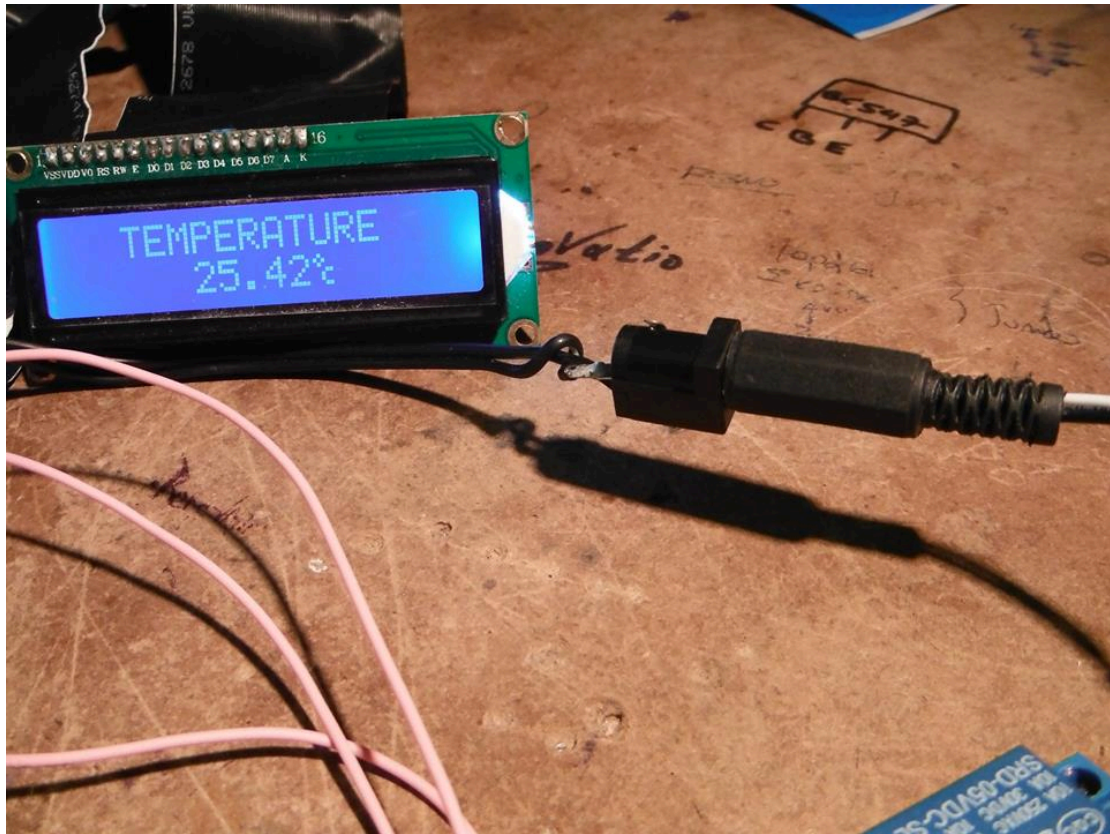


Αρχικά δοκιμάσαμε την σύνδεση της οθόνης και την αποστολή κάποιων μηνυμάτων για να βεβαιωθούμε ότι έχει επιτευχθεί η σύνδεση. Η οθόνη αρχικά συνδέθηκε σειριακά αλλά για μείωση όγκου καλωδίωσης χρησιμοποιήσαμε σύνδεση με I2C προσαρμογέα.

Σειρά είχε η σύνδεση των αισθητήρων πρώτα αυτόν της θερμοκρασίας όπου προβάλλουμε τις μετρήσεις απ' ευθείας στην οθόνη



Σχεδίαση και κατασκευή συστήματος ελέγχου ροής ύδατος.



και στην συνέχεια τον αισθητήρα υγρασίας τον οποίο τοποθετούμε ή στο νερό ή στο χώμα για να δούμε τις διαφορές ανάμεσα στις μετρήσεις.



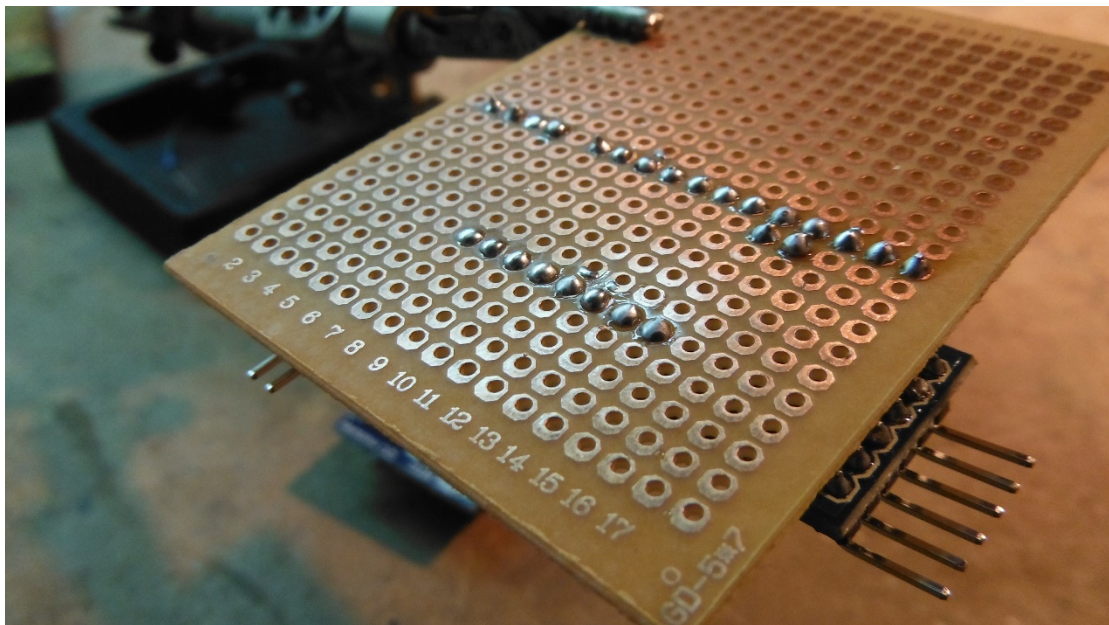


## Σχεδίαση και κατασκευή συστήματος ελέγχου ροής ύδατος.

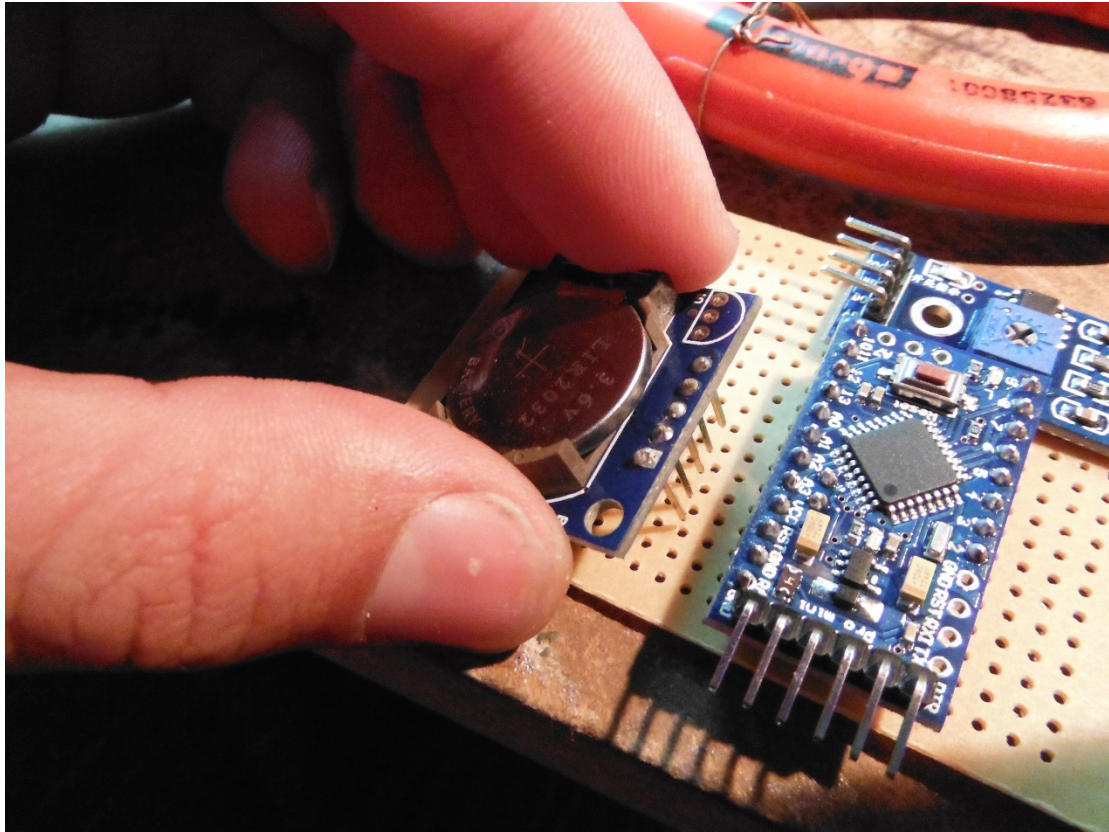
Στην μέτρηση θερμοκρασίας παρουσιάστηκαν κάποιες αποκλίσεις όταν κάναμε μεμονωμένες μετρήσεις γι αυτό και χρησιμοποιήσαμε μια τεχνική smoothing το Moving Average. Αυτό που κάνουμε στην πραγματικότητα είναι να παίρνουμε 10 μετρήσεις απ τον αισθητήρα μας στην συνέχεια τις προσθέτουμε και διαιρούμε το άθροισμα τους με το 10 για να βρούμε τον μέσο όρο τον οποίο και κρατάμε σαν τελική μέτρηση.

$$CMA_n = \frac{x_1 + \dots + x_n}{n} .$$

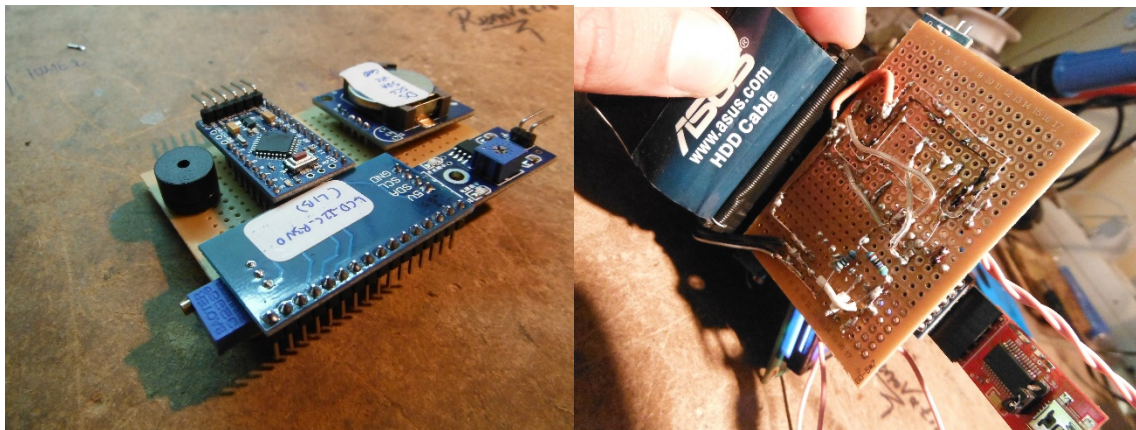
Σε μία διάτρητη πλακέτα ξεκινήσαμε να τοποθετούμε τα εξαρτήματα ένα-ένα και να κολλάμε με το κολλητήρι έτσι ώστε να σταθεροποιήσουμε την κατασκευή μας. Πρώτα ο επεξεργαστής και στη συνέχεια το ρολόι πραγματικού χρόνου που θα χρησιμοποιήσουμε για να έχουμε την σωστή ώρα ακόμη και όταν δεν τροφοδοτείται με ρεύμα το κύκλωμα



Σχεδίαση και κατασκευή συστήματος ελέγχου ροής ύδατος.

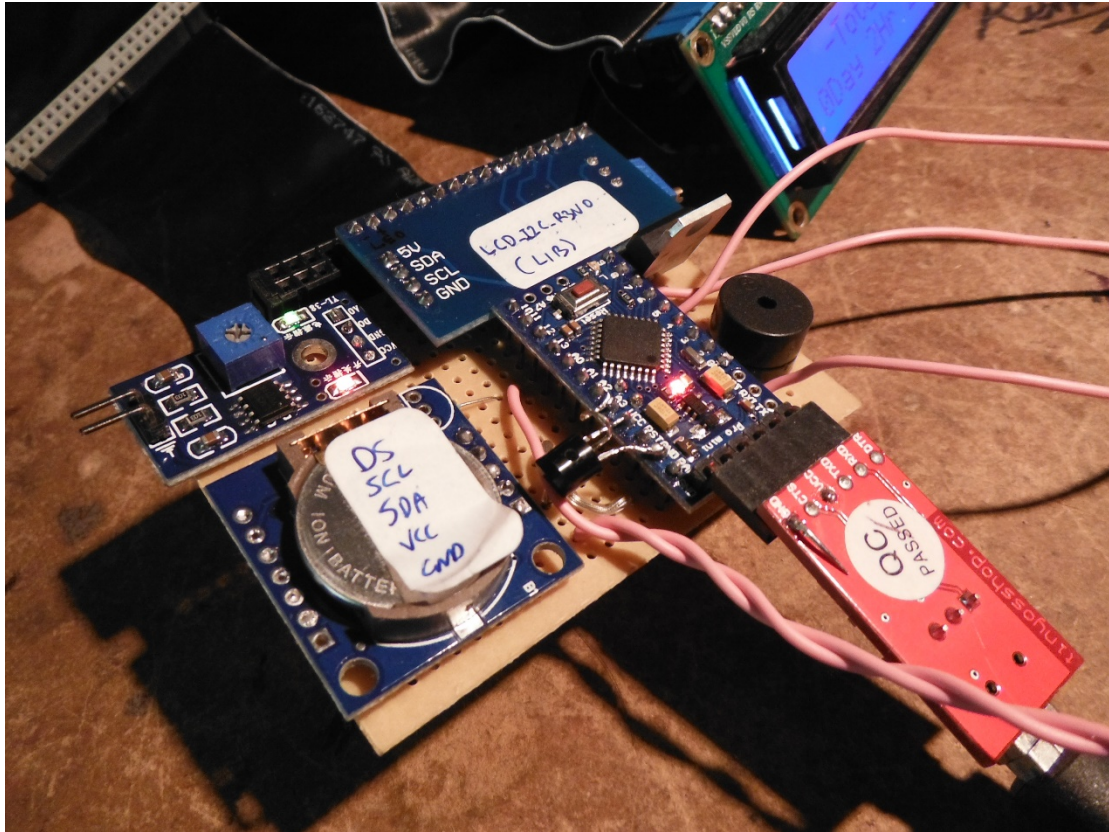


Αφού έχουμε τοποθετήσει όλα τα εξαρτήματα στην πλακέτα και κάναμε τις κολλήσεις για να σταθεροποιήσουμε την κατασκευή έπειτα κάνουμε και τις απαραίτητες συνδέσεις μεταξύ των εξαρτημάτων.

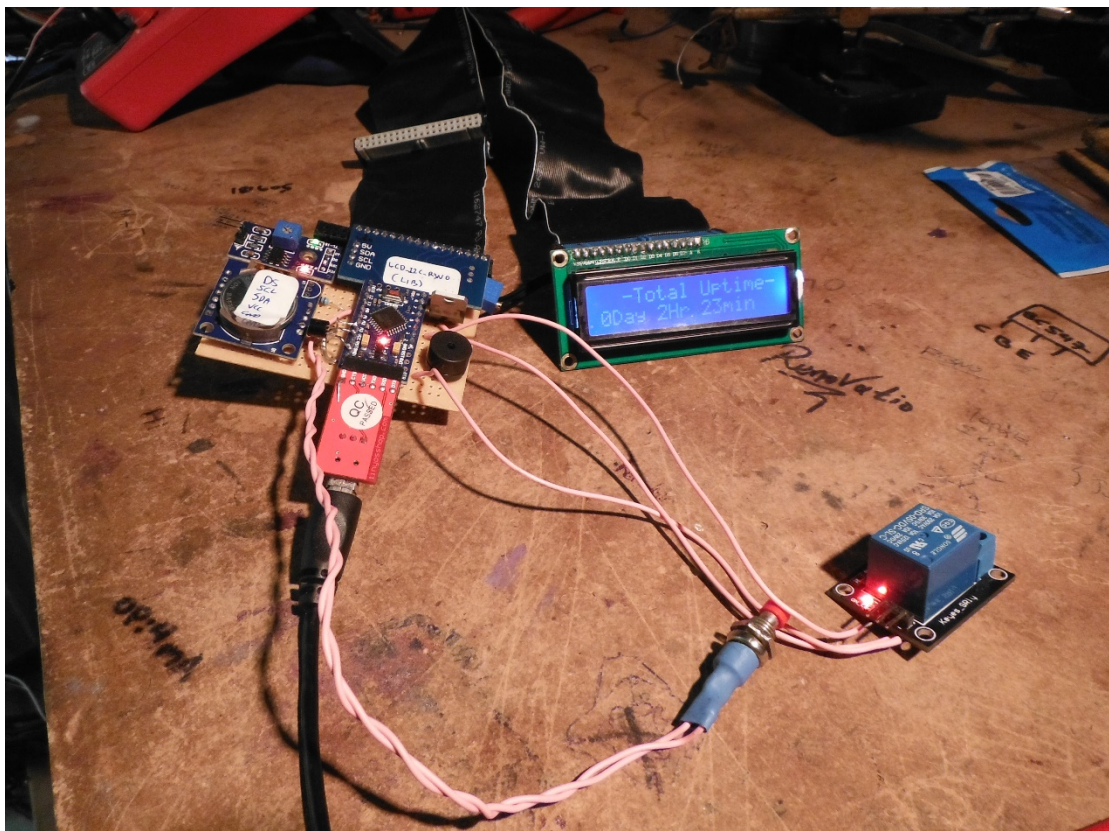




Σχεδίαση και κατασκευή συστήματος ελέγχου ροής ύδατος.



Εφόσον έχουν γίνει όλες οι συνδέσεις τοποθετούμε το ρελέ που θα οδηγεί την ηλεκτροβαλβίδα καθώς κι ένα button που θα του δώσουμε μέσα απ τον κώδικα κάποιες συγκεκριμένες λειτουργίες.



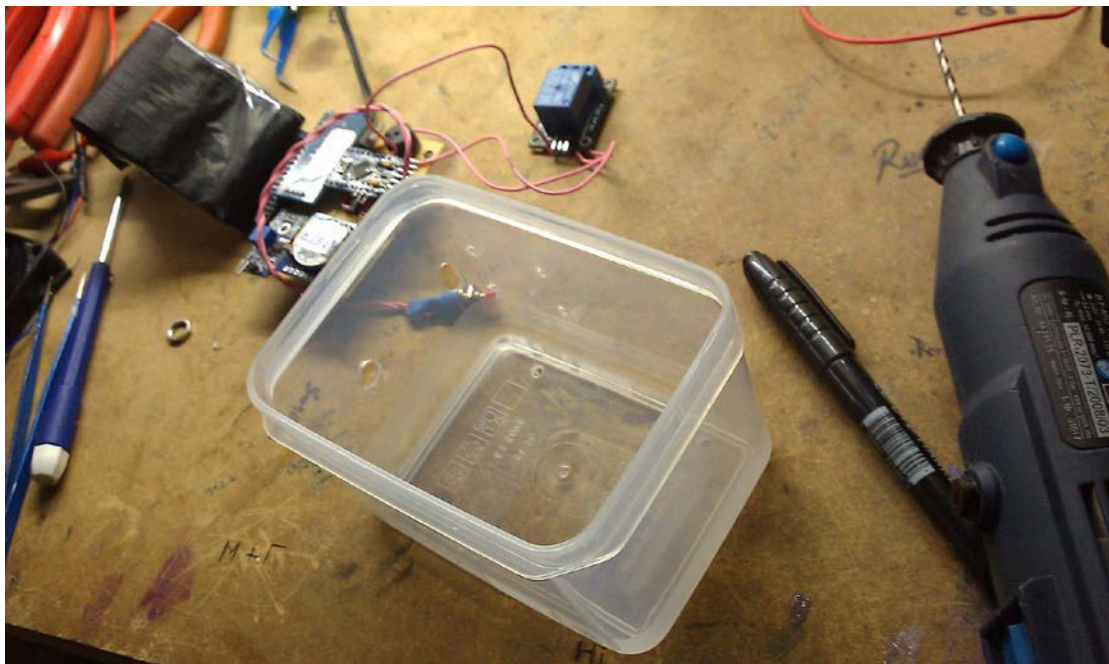
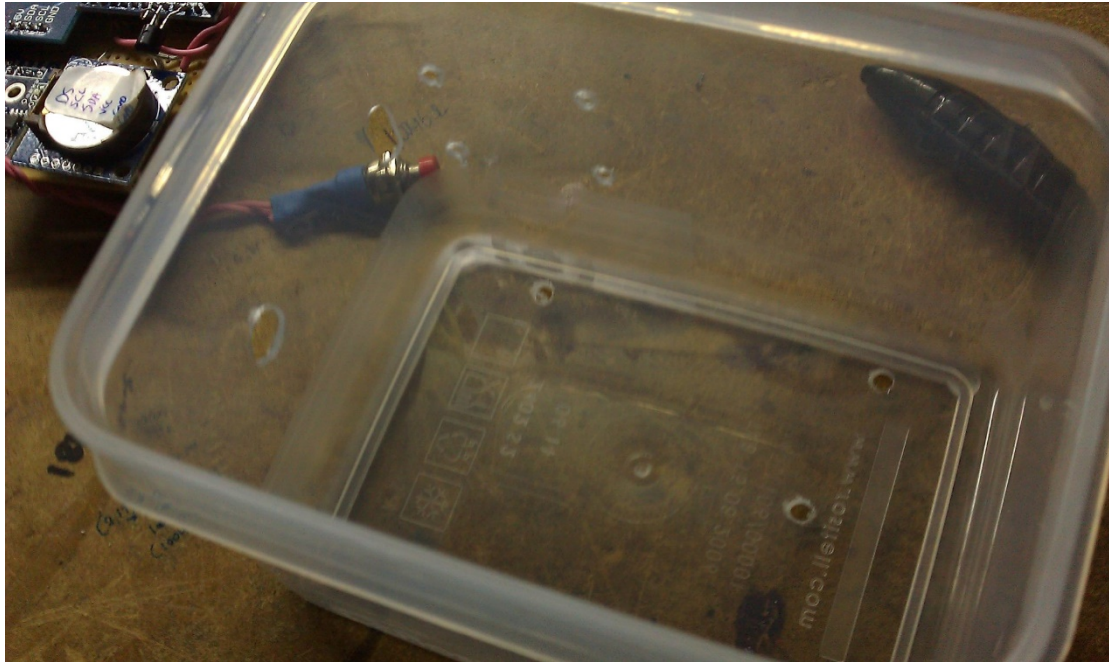
Σχεδίαση και κατασκευή συστήματος ελέγχου ροής ύδατος.

Όλη την κατασκευή μας θα φιλοξενήσει ένα ταπεράκι απ την κουζίνα της μαμάς στο καπάκι του οποίου θα σταθεροποιήσουμε την οθόνη μας και στο εσωτερικό του την πλακέτα μας.





Ξεκινάμε λοιπόν τρύπημα:



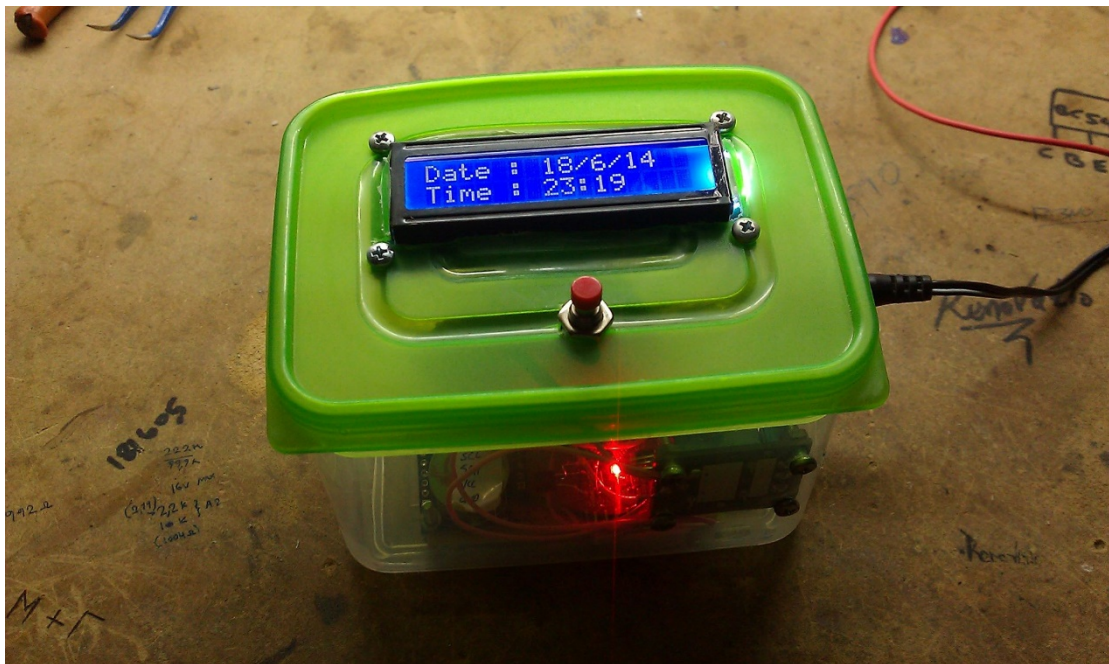
Σχεδίαση και κατασκευή συστήματος ελέγχου ροής ύδατος.

Πρώτα στερεώνουμε την οθόνη και στην συνέχεια την πλακέτα και τα βύσματα τροφοδοσίας καθώς και το ρελέ όπως φαίνεται στις εικόνες που ακολουθούν.





Σχεδίαση και κατασκευή συστήματος ελέγχου ροής ύδατος.



## 6.6 Κώδικας

/\*

Chip : ATmega328PU-TQFP

Modules : [DS1302 RTC(REAL TIME CLOCK)] [LCD I2C MODULE  
PCF8574] [1602 LCD MONITOR] [LM35 TEMPERATURE SENSOR]

Software : [Uptime Timer] [Real Time Clock i2c com.] [1602LCD i2c  
com.]

Libraries : [EEPROM] [Wire] [math] [LiquidCrystal\_I2C]

-----Selides menu-----

**1 katastash potismatos**

**2 voltage monitor**

**3 synolikos xronos leitourgias**

**4 aisthitiras thermokrasias**

**5 emfanisi hmerominias wras**

**6 aisthitiras ygrasias**

**7 DEBUG**

**8 Xeirokinith leitourgia rele - ON**



## 9 Xeirokinith leitourgia rele - OFF

-----

#####

**PIN MAP :**

**---ANALOG PORTS---**

**A1 Voltage\_Sensor**

**A2 Moisture\_Sensor**

**A3 LM35 sensor**

**---DIGITAL PORTS---**

**3 speaker (buzzer)**

**6 Relay**

**12 page button**

**13 activity led**

**\*/**

**#include <EEPROM.h>**

**#include <Wire.h>**

```
#include <math.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(32,16,2);
```

#### **/////pin set**

```
const int LM35_SENSOR=A3;

const int Voltage_Sensor=A1;

const int Moisture_Sensor=A2;

const int speakerPin = 3;

const int relayPin = 6;

const int lcd_page_btn=12;

const int activity_led=13;
```

#### **/////etc vars**

```
int manual_open=2;

int MOIST_wait=0;

int TEMP_wait=0;

int XTIME_wait=0;

int lastwait=0;

int active=0;
```

```
int Moisture_Value;
```

```
int page=7;
```

```
int pagestatus=0;
```

```
int state;
```

```
int ctx=0;
```

```
float AmbientTemperature;
```

```
//diaretis tasews - metavlites metrshs tasews
```

```
float R1 = 10010.0; //syndeetai sto +12v
```

```
float R2 = 4670.0; //syndeetai sto gnd
```

```
float DC_IN,DAC_DC_IN,DIV_DC_IN = 0.0;
```

```
//////uptime timer vars
```

```
int z_minute,z_hour,z_day,z_month,i,eprom_save=0;
```

```
////// Clock module i2c - Variables
```

```
byte decToBcd(byte val) {
```

```
    return ((val/10*16) + (val%10));
```

```
}
```

```
byte bcdToDec(byte val) {
```

```
    return ((val/16*10) + (val%16));  
  
}  
  
char buf[3];  
  
const int DS1307 = 0x68; // Address of DS1307  
  
byte second = 0;  
  
byte minute = 0;  
  
byte hour = 0;  
  
byte weekday = 0;  
  
byte monthday = 0;  
  
byte month = 0;  
  
byte year = 0;  
  
//edw diavazei to DS1307 mesw i2c  
  
void readTime() {  
  
    Wire.beginTransaction(DS1307);  
  
    Wire.write(byte(0));  
  
    Wire.endTransmission();  
  
    Wire.requestFrom(DS1307, 7);  
  
    second = bcdToDec(Wire.read());  
  
    minute = bcdToDec(Wire.read());
```

```
hour = bcdToDec(Wire.read());  
  
weekday = bcdToDec(Wire.read());  
  
monthday = bcdToDec(Wire.read());  
  
month = bcdToDec(Wire.read());  
  
year = bcdToDec(Wire.read());  
  
}
```

### **//internal voltage read function**

```
unsigned int ADCValue;  
  
double Voltage;  
  
double Vcc;  
  
long readVcc()  
  
{ long result;  
  
// 1.1V vRef vs AVcc  
  
ADMUX = _BV(REFS0) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);  
delay(2);  
  
ADCSRA |= _BV(ADSC);  
  
while (bit_is_set(ADCSRA,ADSC));  
  
result = ADCL; result |= ADCH<<8;  
  
result = 1125300L / result;
```

```
return result; }
```

```
//symvolo vathmwn kelsiou
```

```
byte kelsiou[8] = {
```

```
    B01000, B10100,
```

```
    B01000, B00011,
```

```
    B00100, B00100,
```

```
    B00011, B00000 };
```

```
void setup()
```

```
{
```

```
    pinMode(LM35_SENSOR,INPUT);
```

```
    pinMode(lcd_page_btn,INPUT);
```

```
    pinMode(Voltage_Sensor,INPUT);
```

```
    pinMode(activity_led,OUTPUT);
```

```
    pinMode(relayPin,OUTPUT);
```

```
    Wire.begin();
```

```
    lcd.init(); lcd.backlight(); lcd.createChar(1,kelsiou);
```

```
    i=    EEPROM.read(0);
```

```
    z_minute= EEPROM.read(1);
```

```
z_hour= EEPROM.read(2);
```

```
z_day= EEPROM.read(3);
```

```
z_month= EEPROM.read(4);
```

```
tone(3, 500,150);delay(100);
```

```
tone(3, 800,150);delay(100);
```

```
tone(3, 1200,150);delay(100);
```

```
tone(3, 1400,550);delay(100);
```

```
tone(3, 1550,150);delay(100);
```

```
noTone(3);
```

```
}
```

```
void loop()
```

```
{
```

```
//led drastiriotitas
```

```
digitalWrite(activity_led,LOW);
```

```
//anagnwsh tasis trofodosias elengti (eswterikis 5v)
```

```
Vcc = readVcc()/1000.0;
```

```
//anagnwsh ekswterikhs tasis trofodosias (max Vin=16v)
```

```
DAC_DC_IN=analogRead(Voltage_Sensor);
```

```
DIV_DC_IN=(DAC_DC_IN*Vcc)/1024.00;
```

```
DC_IN=DIV_DC_IN/(R2/(R1+R2));
```

```
////////// YPOLOGISMOS THERMOKRASIAS (me omalopoihsh  
mesou orou 13 metrisewn)
```

```
for (int zzx=0; zzx<12; zzx++) {ctx=analogRead(LM35_SENSOR);  
delay(10);
```

```
AmbientTemperature=AmbientTemperature+ctx;}
```

```
AmbientTemperature=AmbientTemperature / 13;
```

```
//AmbientTemperature=analogRead(A3);
```

```
AmbientTemperature=(Vcc * AmbientTemperature * 100.0)/1024.0;
```

```
//UPTIME TIMER/counter (metritis synolikou xronou leitourgias  
syskevis)
```

```
if (i<=(millis()/1000))
```

```
{ i= (millis()/1000)+60;
```

```
z_minute++;
```



```
    eprom_save++;  
  
}  
  
if (eprom_save==5) //kathe 5 lepta  
  
{ eprom_save=0;  
  
  EEPROM.write(0,i-(millis()/1000));  
  
  delay(222);  
  
  EEPROM.write(1,z_minute);  
  
  delay(222);  
  
  EEPROM.write(2,z_hour);  
  
  delay(222);  
  
  EEPROM.write(3,z_day);  
  
  delay(222);  
  
  EEPROM.write(4,z_month);  
  
  delay(2222);  
  
}  
  
if (z_minute>=60)  
  
{  
  
  z_minute=0;  
  
  z_hour++;  
  
}
```

```
if (z_hour>=24)
{
z_hour=0;
z_day++;
}
if (z_day>=30)
{
z_day=0;
z_month++;
}////////uptime timer end
```

```
/*-----ELENXOS POTISMATOS-----
```

**Tha potizei otan:**

**Ygrasia: katw apo 500 kai panw apo 300**

**Wra: ws tis 13:00 kai meta tis 18:00**

**Thermokrasia: ews 35 vathmous kelsiou**

```
-----*/
```

```
//#####
```

```
#####
```

```
//#####
```

```
#####
```

```
//#####
```

```
#####
```

```
// Υπολογισμος ygrasias
```

```
Moisture_Value=analogRead(Moisture_Sensor);
```

```
if (Moisture_Value<300) {MOIST_wait=1;}
```

```
if (Moisture_Value>500) {MOIST_wait=1;}
```

```
if (Moisture_Value>300) {if (Moisture_Value<500){MOIST_wait=0;}}
```

```
// Υπολ. θερμοκρασίας
```

```
if (AmbientTemperature>35) {TEMP_wait=1;}
```

```
if (AmbientTemperature<35) {TEMP_wait=0;}
```

```
//Ελεγχος ώρας
```

```
readTime();
```

```
if (hour>0) {if(hour<13){XTIME_wait=0;}}
```

```
if (hour>18) {XTIME_wait=0;}
```

```
if (hour>13) {if (hour<18){XTIME_wait=1;}}
```

```
//edw ginetai h istoria pou potizei 1 leptο kai perimenei 1 leptο  
  
//1 leptο= 60"  
  
if (lastwait<millis()/1000)  
  
{  
  
tone(3, 1050,150);delay(50);  
  
tone(3, 1250,150);delay(50);  
  
tone(3, 1850,150);delay(50);  
  
noTone(3);  
  
// +60 defterolepta diladi  
  
lastwait=(millis()/1000)+60; //xronos anamonis kyklou se defterolepta  
  
if (active==0) {active=1;}  
  
else {active=0;}  
  
}  
  
if (manual_open==2) //manual_open=2 shmainei oti to potisma  
ginetai aftomata  
  
{  
  
if (MOIST_wait==0 & XTIME_wait==0 & active==0 & TEMP_wait==0)  
  
{digitalWrite(relayPin,HIGH);
```

```
state=1;}

else {digitalWrite(relayPin,LOW); state=0;}

}

if (manual_open==1) {digitalWrite(relayPin,HIGH);} //parakampsh
aftomaths leitourgeias, potisma energo

if (manual_open==0) {digitalWrite(relayPin,LOW);} //parakampsh
aftomaths. apenergopoihsh potismatatos

//#####
#####

//#####          TELOS      ELENXOU      POTISMATOS
#####

//#####
#####

// -----LCD page button-----

pagestatus=digitalRead(lcd_page_btn);

if (pagestatus==HIGH)

{
```

```
tone(3, 1850,150);delay(100); noTone(3);  //ο ηχος που παίζει όταν  
πατιεται το κουμπι
```

```
page++;
```

```
if (page>9) {  //8 selides menu (9-1)
```

```
page=1;
```

```
}
```

```
}
```

```
// #####lcd page 1 #####
```

```
if (page==1) {
```

```
lcd.clear();
```

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

```
lcd.print(" KATASTASH : ");
```

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

```
if (state==0){lcd.print(" Den potizei");}
```

```
else {lcd.print(" Potizei");}
```

```
manual_open=2;
```

```
}
```

```
//#####lcd page 2 #####
```

```
if (page==2) {  
  
  lcd.clear();  
  
  lcd.setCursor(0,1);  
  
  lcd.print("Logic Vcc: "); lcd.print(Vcc); lcd.print(" V ");  
  
  lcd.setCursor(2,0);  
  
  lcd.print("Vin: "); lcd.print(DC_IN);lcd.print(" V");  
  
}
```

```
//#####lcd page 3 #####
```

```
if (page==3) { //uptime display
```

```
  lcd.clear();  
  
  lcd.setCursor(0,0);  
  
  lcd.print(" -Total Uptime-");  
  
  lcd.setCursor(0,1);  
  
  lcd.print(z_day);  
  
  lcd.print("Day ");  
  
  lcd.print(z_hour);  
  
  lcd.print("Hr ");  
  
  lcd.print(z_minute);  
  
  lcd.print("min ");
```

```
}
```

```
//#####lcd page 4 #####
```

```
if (page==4) { //version display
```

```
lcd.clear();
```

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

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

```
lcd.setCursor(5,1);
```

```
lcd.print(AmbientTemperature); lcd.write(1);
```

```
delay(50);
```

```
}
```

```
//#####lcd page 5 #####
```

```
if (page==5) { //display date/time sto lcd
```

```
lcd.clear();
```

```
readTime();
```

```
lcd.print("Date : ");
```

```
lcd.print(monthday);
```

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

```
lcd.print(month);
```



```
lcd.print("/");  
  
lcd.print(year);  
  
lcd.setCursor(0,1);  
  
lcd.print("Time : ");  
  
lcd.print(hour);  
  
lcd.print(":");  
  
sprintf(buf, "%02d", minute);  
  
lcd.print(minute);  
  
}
```

```
//#####lcd page 6 #####
```

```
if (page==6) { //aisthitiras ygrasias
```

```
lcd.clear();
```

```
lcd.print("Moisture Sensor:");
```

```
lcd.setCursor(5,1);
```

```
lcd.print(Moisture_Value);
```

```
}
```

```
//#####lcd page 7 -D E B U G - #####
```

```
if (page==7) {
```

```
lcd.clear();

lcd.setCursor(0,0);

lcd.print(Moisture_Value);

lcd.print(" TmW="); lcd.print(XTIME_wait);

lcd.print(" RL="); lcd.print(state);

lcd.setCursor(0,1);

lcd.print("delWT="); lcd.print(active);

lcd.print(" tWt="); lcd.print(TEMP_wait);

lcd.print(" M");lcd.print(MOIST_wait); //moisture

}
```

```
//#####lcd page 8 #####
```

```
if (page==8) { //RELAY === ON

lcd.clear();

lcd.print(" MANUAL RELAY ");

lcd.setCursor(3,1);

lcd.print(" OPEN");

manual_open=1;

}
```

```
//#####lcd page 9 #####  
  
if (page==9) { //RELAY === OFF  
  
  lcd.clear();  
  
  lcd.print(" MANUAL RELAY ");  
  
  lcd.setCursor(3,1);  
  
  lcd.print(" CLOSED");  
  
  manual_open=0;  
  
  }  
  
  delay(500);  
  
//loop end
```

## **ΚΕΦΑΛΑΙΟ 7<sup>ο</sup> Βιβλιογραφία – Παραγγελίες εξαρτημάτων - Αναφορές**

Οι πηγές που αναζητήθηκαν για την συγγραφή της παρούσας πτυχιακής εργασίας είναι:

Ιστοσελίδα: <http://arduino.cc>

Ιστοσελίδα: <http://wikipedia.org>

Ιστοσελίδα: <http://www.solenoid-valve-info.com/solenoid-valve-basics.html>

Ιστοσελίδα: [www.microplanet.gr](http://www.microplanet.gr)

Ιστοσελίδα: <http://arduino-info.wikispaces.com/LCD-Blue-I2C>

Οι αγορές των εξαρτημάτων που χρησιμοποιήθηκαν για τη κατασκευή έγιναν από:

Ιστοσελίδα: [www.ebay.com](http://www.ebay.com) (online αγορά) με τα περισσότερα απ αυτά να έχουν τοποθεσία προέλευσης την Κίνα.

Κάποιες επιπλέον αγορές που χρειάστηκαν για ηλεκτρολογικό και υδραυλικό εξοπλισμό έγιναν από καταστήματα της Άρτας.

Το συνολικό κόστος δεδομένου ότι κάποια εξαρτήματα ήταν ελαττωματικά ή δεν τα χρησιμοποιήσαμε σωστά με αποτέλεσμα να καταστραφούν δεν ξεπέρασε τα 50€ με 60€.

### **Αναφορές**

<http://el.wikipedia.org/wiki/%CE%91%CF%85%CF%84%CE%BF%CE%BC%CE%B1%CF%84%CE%B9%CF%83%CE%BC%CF%8C%CF%82>

Σχεδίαση και κατασκευή συστήματος ελέγχου ροής ύδατος.

<http://el.wikipedia.org/wiki/%CE%9C%CE%B9%CE%BA%CF%81%CE%BF%CE%B5%CE%BB%CE%B5%CE%B3%CE%BA%CF%84%CE%AE%CF%82>

<http://el.wikipedia.org/wiki/Arduino>

<http://www.microplanet.gr/tutorials/microcontrollers/arduino>

[https://en.wikipedia.org/wiki/Moving\\_average#Exponential\\_moving\\_average](https://en.wikipedia.org/wiki/Moving_average#Exponential_moving_average)