Τμήμα Μηχανικών Αυτοματισμού
Αλεξάνδρειο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Θεσ/νίκης

Εποπτεία για ποτιστικό καρούλι

(Sprinkler reel supervisor)

Πτυχιακή εργασία

του

Φέκα Άγγελου

Επιβλέπων καθηγητής: **Δρ. Νικόλαος Νικολαΐδης**

Θεσσαλονίκη Νοέμβριος 2016

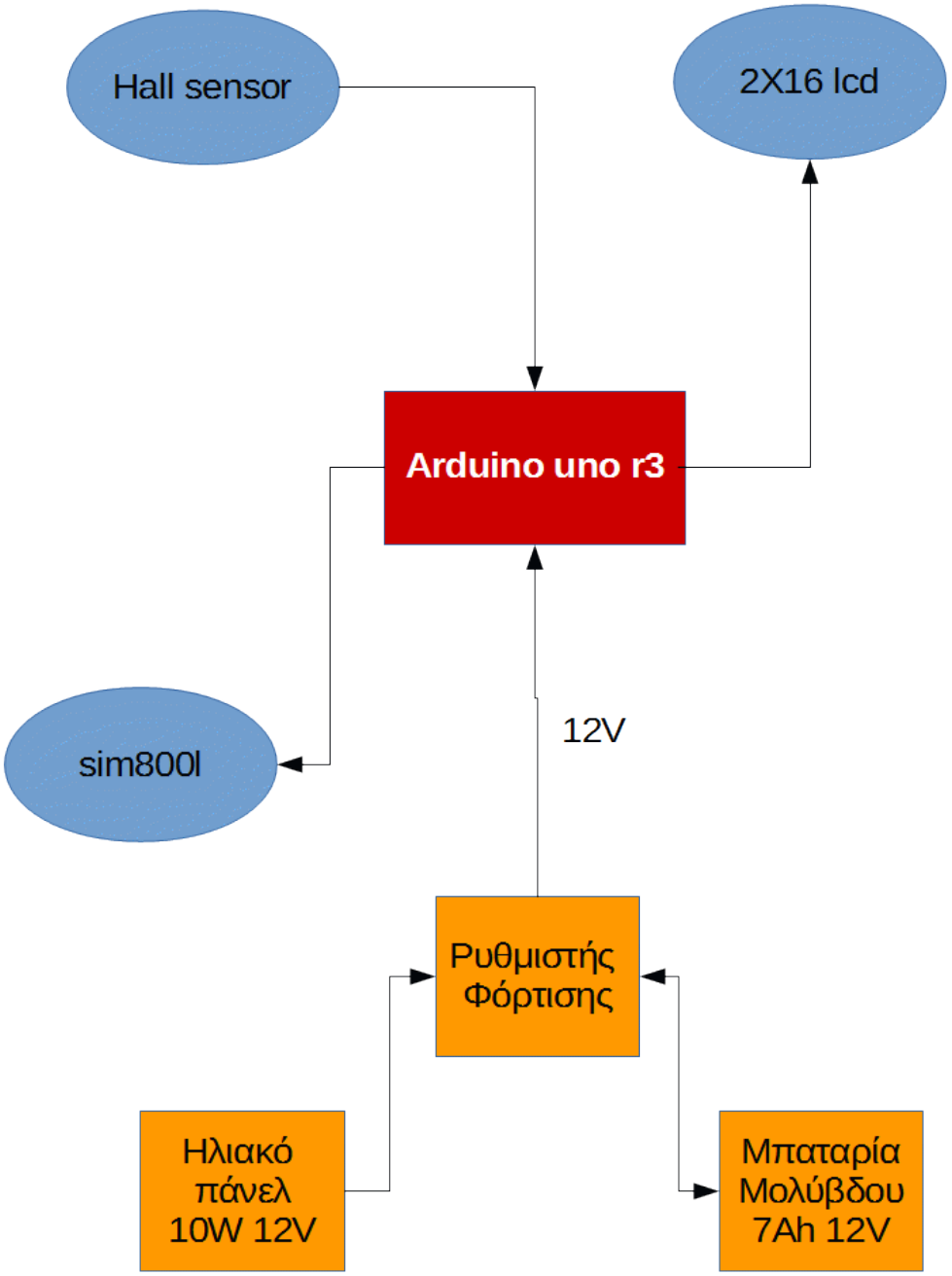
Περίληψη

Από το πότισμα των αγροτικών καλοκαιρινών καλλιεργειών εξαρτάται κατά πολύ μεγάλο βαθμό η ποιότητα και η ποσότητα της παραγωγής. Ένα μηχάνημα που μας επιτρέπει να ποτίζουμε τις καλλιέργειές μας σωστά και ευκολότερα από άλλους τρόπους είναι το ποτιστικό καρούλι. Μας δίνει την δυνατότητα να κάνουμε ομοιόμορφο πότισμα χωρίς να τραυματίζουμε τα φυτά. Ποτίζει μεγάλες εκτάσεις, είναι εύκολο στη μεταφορά και στην τοποθέτησή του.

Ο ελεγκτής – επόπτης που θα κατασκευάσουμε δίνει και αλλά πλεονεκτήματα ώστε να κάνει το πότισμα ακόμη ευκολότερο, με μεγαλύτερη ακρίβεια και χωρίς κίνδυνο ζημιάς στην καλλιέργεια. Ο χρήστης του μηχανήματος έχει την εποπτεία του καρουλιού απομακρυσμένα με ειδοποιήσεις γραπτών μηνυμάτων. Έχει τη δυνατότητα ανά πάσα στιγμή να γνωρίζει τη θέση του μηχανήματος, την ταχύτητα, τον ακριβή χρόνο ολοκλήρωσης του ποτίσματος και την απόσταση που έχει απομείνει.

Τα οφέλη έχουν να κάνουν με το οικονομικό κομμάτι, με την εξοικονόμηση χρόνου, με την ασφάλεια ομαλής λειτουργίας και με τον ευκολότερο προγραμματισμό των υπολοίπων εργασιών. Μας εξοικονομεί χρήματα από τις άσκοπες μεταφορές για να δούμε επιτόπου την ομαλή λειτουργία του καρουλιού, για τον ίδιο λόγο κερδίζουμε και σε χρόνο. Έχουμε την ασφάλεια πως αν κάτι συμβεί και σταματήσει τη κανονική λειτουργία του μηχανήματος και θα ενημερωθούμε γρήγορα ώστε να μπορέσουμε να παρέμβουμε για την αποκατάσταση της ζημίας χωρίς να προκληθεί καταστροφή στην καλλιέργεια. Γνωρίζοντας τον ακριβή χρόνο που θα ολοκληρωθεί το πότισμα μπορούμε να ρυθμίζουμε τον χρόνο μας για τις υπόλοιπες εργασίες μας.

Η υλοποίηση του ελεγκτή βασίστηκε στον arduino uno r3 πάνω στον οποίο σύνδεσα δυο μαγνητικούς αισθητήρες για τη μέτρηση των στροφών, ένα gsm sim800l το οποίο στέλνει τα μηνύματα κειμένου στο χρήστη για να τον ενημερώσει για την λειτουργία του καρουλιού και μια οθόνη 2X16 ώστε να υπάρχει απεικόνιση των τιμών πάνω στον ελεγκτή μας. Η τροφοδοσία έγινε μέσω μια μπαταρίας μόλυβδου 7Ah 12V η οποία μέσω ενός ρυθμιστή φόρτισης συνδέεται με ένα φωτοβολταϊκό πάνελ 10W.



Εικόνα 1-1 Μπλοκ διάγραμμα ελεγκτή

Abstract

The irrigation of the agricultural summer crops plays a substantial role in the quality and quantity of the crop production. One machine that allows us to irrigate our crops correctly and easier than other means is the water reel. It allows us to irrigate uniformly without damaging the plants. It irrigates larger areas, it's easy to transfer and install.

The supervisor that we will build, provides advantages in order to make the irrigation simpler, with higher accuracy and without any risk damaging our crop. The user of the machine has the supervision of the water reel remotely with the help of short messages notifications (sms). He has the ability to know at any time the location of the machine, the speed, the exact time left to complete the irrigation and the distance left.

The benefits that it provides us cover the financial part, time saving, safety of smooth operation and easier planning of the rest of our workload. It saves us time and money from the need of unnecessary trips to supervise the reel. It provides us the safety that if something happens and interrupts the normal operation of the reel we will be notified promptly, so that we can intervene and correct the damage, without destroying our crop. Knowing the exact time left to complete the irrigation, it gives us the opportunity to plan our schedule for the rest of our agricultural tasks.

The implementation of the controller is based on the Arduino Uno R3, where 2 magnetic sensors have been connected in order to count the reel turns and detect the direction, a gsm sim 800l which sends the sms to the user to notify him about the reel operation and an 2X16 LCD screen which displays the parameters of our controller. The power is provided by a 12 v 7Ah zinc battery which is connected to a 10W solar panel through a charging regulator.

Περιεχόμενα

Ευρετήριο εικόνων

1. Εισαγωγή

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τα βασικά εξαρτήματα που θα χρειαστούμε για την υλοποίηση της κατασκευής μας. Θα γίνει περιγραφή της λειτουργίας και του τρόπου χρήσης του ποτιστικού καρουλιού που είναι και το μηχάνημα το οποίο θέλουμε να εποπτεύσουμε. Στο τέλος θα αναφερθούμε στους στόχους και στον σκοπό της εργασίας μας.

1.1 Σκοπός

Θα σχεδιαστεί ένα σύστημα τηλεπληροφόρισης του αγρότη με GSM για την παρακολούθηση της πορείας του ποτίσματος ενός ποτιστικού καρουλιού.

Ο εκλεκτής έχει ως σκοπό να μετρά την απόσταση που έχει τοποθετηθεί το ποτιστικό καρούλι (1.3) στην αρχική του θέση, να ελέγχει κάθε στιγμή την ταχύτητα με την οποία κινείται και να ενημερώνει τον χρήστη για την ακριβή του θέση, τον εναπομείναντα χρόνο, την ταχύτητά του και τυχόν αστοχίες στη λειτουργία του, όπως απότομη μεταβολή ταχύτητας ή σταμάτημα. Επίσης θα ενημερώνει ότι η διαδικασία ποτίσματος πλησιάζει στο τέλος.

Όλα τα παραπάνω θα τα πετυχαίνει ανιχνεύοντας την φορά των στροφών του ποτιστικού καρουλιού και μετρώντας τον αριθμό αυτών υπολογίζοντας την ταχύτητα, την απόσταση και τον υπολειπόμενο χρόνο ολοκλήρωσης του ποτίσματος. Έτσι, θα μπορεί απομακρυσμένα να ενημερώνει τον χρήστη όταν κριθεί όταν είναι απαραίτητο, (περιπτώσεις αστοχίας ή ολοκλήρωσης του ποτίσματος) ή οποιαδήποτε στιγμή ζητήσει ο χρήστης πληροφορίες. Η ενημέρωση γίνεται μέσω γραπτών μηνυμάτων (sms) στο κινητό τηλέφωνο του χρήστη (1.4.6).

1.2 Η γεωργία στης Ελλάδα

Γεωργία είναι το σύνολο των δραστηριοτήτων που σχετίζονται με την καλλιέργεια του εδάφους της γης με σκοπό την παραγωγή φυτικών προϊόντων. Μερικές φορές ο όρος επεκτείνεται και για τη διαδικασία της καλλιέργειας φυκιών στη θάλασσα. Στη γεωργία επίσης υπάγεται και η συλλογή και πρωτογενής επεξεργασία των προϊόντων αυτών των φυτών. Η γεωργία κατατάσσεται στην ελαφρά βιομηχανία, επειδή τα περισσότερα προϊόντα που παράγονται από αυτήν είναι προϊόντα άμεσης χρήσης από τον άνθρωπο. Παράγοντες που επηρεάζουν την γεωργία είναι το κλίμα και η μορφολογία του εδάφους. Η γεωργία είναι αντικείμενο των περισσότερων κλάδων της γεωπονίας, μαζί με τη κτηνοτροφία και την αλιεία. Η γεωργική παραγωγή απασχολούσε το 2007 περίπου το ένα τρίτο των εργατών. Τα τελευταία χρόνια ο κλάδος παροχής υπηρεσιών απασχολεί τα περισσότερα άτομα.

Με τη γεωργία παράγονται προϊόντα που προορίζονται για τη διατροφή των ανθρώπων, των οικόσιτων ζώων αλλά και μερικά που προορίζονται για την παραγωγή άλλων ειδών προϊόντων ως και βιοκαυσίμων, τα τελευταία χρόνια κυρίως. Η γεωργία θεωρείται ότι ήταν το κλειδί για την αύξηση του ανθρώπινου πληθυσμού και την ανάπτυξη του πολιτισμού, αφού η γεωργία και η κτηνοτροφία δημιούργησε πλεονάσματα τροφίμων που επέτρεψαν και την αύξηση του πληθυσμού και την εξέλιξη του πολιτισμού. Κάποιου είδους γεωργικές δραστηριότητες έχουν παρατηρηθεί και σε μερικά είδη μυρμηγκιών και τερμιτών, αλλά μιλώντας για γεωργία, εννοούμε βασικά ανθρώπινες δραστηριότητες.

Η ιστορία της γεωργίας πάει πίσω αρκετές χιλιάδες χρόνια και η ανάπτυξή της οδηγήθηκε και καθορίστηκε σε μεγάλο βαθμό από τις κλιματικές διαφορές, τις κουλτούρες και την υφιστάμενη σε αυτές τεχνολογία. Ωστόσο, όλη η γεωργία βασίζεται σε τεχνικές επέκτασης και διαχείρισης εδαφών κατάλληλων για την ανάπτυξη των εξημερωμένων φυτικών ειδών. Αυτό πολλές φορές απαιτεί μορφές άρδευσης, αποστράγγισης, οριοθέτησης και προστασίας των καλλιεργούμενων εδαφών. Στον «ανεπτυγμένο» κόσμο, η βιομηχανική γεωργία που βασίστηκε σε μεγάλης κλίμακας μονοκαλλιέργειες έγινε το κυρίαρχο σύστημα σύγχρονης γεωργίας, παρόλο που υπάρχει μια ανοδική υποστήριξη για εναλλακτικές μορφές γεωργίας (π.χ. βιολογική γεωργία).

Η σύγχρονη αγρονομία, η ανάπτυξη υβριδίων, ζιζανιοκτόνων, παρασιτοκτόνων, λιπασμάτων και άλλων τεχνολογικών βελτιώσεων έχει αυξήσει ποσοτικά τις σοδιές από τη γεωργική καλλιέργεια, αλλά ταυτόχρονα προκάλεσε ευρεία οικολογική βλάβη στο περιβάλλον και είχε αρκετά αρνητικά αποτελέσματα στην ανθρώπινη υγεία. Η πολύ εκτεταμένη κτηνοτροφία αύξησε ομοίως την παραγωγή κρέατος, αλλά δημιούργησε προβλήματα σχετικά με τη σκληρότητα κατά των ζώων, βλάβες στην ανθρώπινη υγεία από την κατάχρηση αντιβιοτικών, αυξητικών ορμονών και άλλων χημικών από τη βιομηχανία παραγωγικής προϊόντων κρέατος. [1]

1.3 *Το ποτιστικό καρούλι*

Το ποτιστικό καρούλι είναι ένα γεωργικό μηχάνημα που χρησιμοποιείται σε ολόκληρη την Ελλάδα για το πότισμα διαφόρων καλλιεργειών, όπως βαμβάκι, ηλίανθος, μηδική, καλαμπόκι, φιστίκι, ζαχαρότευτλο, ελαιοκράμβη και αρκετές ακόμη.

Εικόνα 1-1 Το ποτιστικό καρούλι

Το καρούλι όπως φαίνεται και στην (Εικόνα 1-1) είναι μια μεταλλική κατασκευή μεγάλου μεγέθους. Η μεταφορά του γίνεται με την βοήθεια δυο τροχών που έχει στην βάση του, εκεί που στηρίζεται ολόκληρη η κατασκευή. Είναι εφοδιασμένο με μια έλξη η όποιοι είναι απαραίτητη για την ζεύξη με το τρακτέρ, έτσι ώστε να είναι εφικτή η μεταφορά του. Πάνω στο τρέιλερ που είναι και η βάση του, υπάρχει μια κυλινδρική κατασκευή στην οποία πάνω είναι τυλιγμένο ένα λάστιχο. Το μήκος και η διατομή του λάστιχου δεν είναι συγκεκριμένη. Συνήθως το μήκος του είναι από 200 έως 300 μέτρα και η διατομή του από 2 ίντσες έως 3,5. Η μια άκρη του λάστιχου είναι συνδεμένη με τον σωλήνα ο οποίος χρησιμοποιείται για την τροφοδοσία του συστήματος, η άλλη άκρη συνδέεται με το ποτιστικό

σύστημα. Το ποτιστικό σύστημα με την σειρά του έχει μια δική του βάση η οποία έχει 3 τροχούς σε τριγωνική διάταξη, υπάρχουν όμως και βάσεις με 2 τροχούς.

1.3.1 Ποτιστικό σύστημα

Εικόνα 1-2 Πυραυλικό σύστημα ποτίσματος

Δυο διαφορετικά συστήματα ποτίσματος συναντάμε. Το πρώτο όπως φαίνεται και στην (Εικόνα 1-2) είναι ένας πύραυλος ο οποίος περιστρέφεται δεξιά και αριστερά και είναι ικανός να ποτίσει σε διάμετρο 100 μέτρων (50 δεξιά 50 αριστερά) ίσως και παραπάνω αν επαρκεί η πίεση του δικτύου.

Πλεονεκτήματα αυτού του συστήματος είναι

- Μικρό βάρος
- Εύκολη μεταφορά
- Απλότητα συστήματος
- Ελάχιστες πιθανότητες βλάβης
- Μεγάλη διάμετρος ποτίσματος
- Μικρότερο κόστος αγοράς

Μειονεκτήματα

- Δεν είναι κατάλληλο για όλες τις καλλιέργειες
- Πειράζεται από τον αέρα σε σημείο να είναι αδύνατο το πότισμα

Ένα δεύτερο σύστημα ποτίσματος είναι μια ράμπα μήκους 50 μέτρων συνήθως η οποία έχει πολλά μπεκ ανά ίσα διαστήματα, συνήθως ανά ένα μέτρο. Για αυτό έχει πάρει και το όνομα πολυμπέκ.(Εικόνα 1-3), (Εικόνα 1-4)

Εικόνα 1-3 Ράμπα πολυμπέκ

Εικόνα 1-4 Πολυμπέκ πότισμα

Το πολυμπέκ είναι σωστότερος τρόπος ποτίσματος επειδή το νερό δεν χτυπάει με δύναμη τα φυτά, αλλά τα ψεκάζει. Είναι ιδανικό για ευαίσθητες καλλιέργειες ή για καλλιέργειες που είναι ακόμη στην ανάπτυξη τους.

Τα **πλεονεκτήματά** του είναι:

- Καλύτερος και σωστότερος τρόπος ποτίσματος
- Ικανό να ποτίζει και σε συνθήκες με αέρα
- Καλύπτει (ποτίζει) μεγάλη έκταση
- Ιδανικό για καλλιέργειες υπό ανάπτυξη
- Ίση κατανομή νερού σε όλη την ποτιζόμενη επιφάνεια

Μειονεκτήματα:

- Μεγάλο βάρος
- Δυσκολότερη μεταφορά
- Πολύπλοκη κατασκευή
- Περισσότερες βλάβες

1.3.2 Πώς ποτίζουμε

Το καρούλι μεταφέρεται στο χωράφι, όπως προαναφέρθηκε, με την χρήση του τρακτέρ και τοποθετείται σε αυτό ώστε να είναι έτοιμο προς χρήση. Ο κυλινδρικός μηχανισμός έχει πάνω του τυλιγμένο το λάστιχο.

Η άκρη του λάστιχου όπου βρίσκεται και το ποτιστικό μας σύστημα (1.3.1) έχει και αυτή με την σειρά της μια έλξη για να μπορέσει να απλωθεί το λάστιχο κατά το μήκος του χωραφιού με την χρήση και πάλι του τρακτέρ. Ολοκληρώνοντας το άπλωμα του λάστιχου, απομακρύνουμε το τρακτέρ και είμαστε έτοιμοι να τροφοδοτήσουμε το καρούλι με νερό για να ξεκινήσει το πότισμα.

Η τροφοδοσία γίνεται με αρκετούς τρόπους. Οι συνηθέστεροι τρόποι είναι μέσω γεώτρησης και αντλίας, μέσω δικτύου υδροδότησης και με πετρελαιοκίνητες αντλίες.

1.3.3 Τρόπος λειτουργίας

Μόλις το καρούλι τροφοδοτηθεί με νερό ξεκίνα και η διαδικασία του ποτίσματος. Οποιοδήποτε ποτιστικό σύστημα έχουμε επιλέξει ο τρόπος λειτουργίας παραμένει ο ίδιος. Το λάστιχο γεμίζει με νερό, ξεκινά η ροή και το πότισμα της καλλιέργειας. Ταυτόχρονα με τη ροή του νερού και το πότισμα του χωραφιού, το καρούλι τυλίγει το λάστιχο σιγά σιγά και με αυτόν τον τρόπο ποτίζει όλη την επιφάνεια του χωραφιού που έχουμε απλώσει εξ αρχής.

1.3.4 Διαδικασία τυλίγματος

Δυο είναι οι τρόποι που το καρούλι κινείται ώστε να τυλίξει το λάστιχο. Ένας υδραυλικός και ένας μηχανικός. Ο πρώτος δεν θα περιγραφεί γιατί είναι παλιός και πλέον δεν εφαρμόζεται πουθενά. Ο μηχανικός τρόπος που είναι και αυτός που εφαρμόζεται, αποτελείται από μια τουρμπίνα και από ένα σασμάν σχέσεων.

Εικόνα 1-5 Μηχανισμός κίνησης

Όπως φαίνεται και στην (Εικόνα 1-5), η είσοδος του νερού χωρίζεται σε δυο επιμέρους, η μια πηγαίνει κατευθείαν στο λάστιχο και από εκεί στο ποτιστικό σύστημα, ενώ η δεύτερη πρώτα περνά μέσα από μια τουρμπίνα περιστρέφει μια φτερωτή και στη συνέχεια καταλήγει στο λάστιχο. Με αυτόν τον τρόπο πετυχαίνουμε την κίνηση του κυλινδρικού συστήματος ώστε να τυλίξει σιγά σιγά το λάστιχο. Η ταχύτητα της περιστροφής (τυλίγματος) ρυθμίζεται με τους εξής δυο τρόπους. Μέσω των σχέσεων του σασμάν, συνήθως το σασμάν είναι εφοδιασμένο με δυο λεβιέδες, με τον έναν επιλεγούμε αργό/γρήγορο και με τον δεύτερο επιλεγούμε ανάμεσα σε τρεις σχέσεις (1/2/3). Έτσι ανάλογα με τη σχέση έχουμε και την ανάλογη ταχύτητα. Στην πράξη αυτό δεν είναι πάντα εφικτό, δηλαδή δεν μπορούμε πάντα να ρυθμίσουμε την επιθυμητή ταχύτητα.

Αυτό μπορεί να συμβεί όταν η πίεση της τροφοδοσίας δεν είναι η απαιτούμενη. Σε αυτές τις περιπτώσεις βάζουμε μικρότερη ταχύτητα για να ξεκινήσει η κίνηση του καρουλιού θυσιάζοντας την επιθυμητή ταχύτητα. Αν η πίεση της τροφοδοσίας πέσει κάτω από περίπου 1,5atm τότε το πότισμα με χρήση καρουλιού είναι μη εφικτό (σπάνια περίπτωση). Αφού επιλέξουμε την επιθυμητή σχέση στο σασμάν υπάρχει και μια δεύτερη ρύθμιση που κάνουμε ώστε να ρυθμίσουμε την ταχύτητα. Όπως φαίνεται και στην εικόνα υπάρχει μια χειροκίνητη βάνα με την οποία ρυθμίζουμε πόσο νερό θα περάσει μέσα από την τουρμπίνα ώστε να προκαλέσει κίνηση. Σε αυτή την ρύθμιση πρέπει να είμαστε προσεκτικοί διότι πρέπει να υπάρχει μια χρυσή τομή ανάμεσα στην ποσότητα νερού που πηγαίνει στο ποτιστικό μας σύστημα και της ποσότητας που περνά μέσα από την τουρμπίνα. Αυτό γιατί αν περάσει όλο το νερό από την τουρμπίνα ή μεγάλο μέρος από αυτό, το ποτιστικό σύστημα δεν θα έχει την πίεση που χρειαζόμαστε για την ομαλή του λειτουργία. Τέλος, αφού έχουμε ρυθμίσει την ταχύτητα στο επιθυμητό μπορούμε να φύγουμε από το χωράφι και να επιστρέψουμε όταν το πότισμα θα έχει ολοκληρωθεί, δηλαδή όταν το καρούλι θα έχει μαζευτεί μέχρι το τέρμα. Όταν θα ολοκληρωθεί η διαδικασία, μετακινούμε δίπλα το μηχάνημα για να ποτίσουμε το υπόλοιπο χωράφι ή το μετακινούμε σε άλλο χωράφι.

1.4 Ανάλυση κατασκευής

Σε αυτό το κεφάλαιο θα γίνει μια παρουσίαση των υλικών που θα χρειαστούμε για να υλοποιήσουμε τον ελεγκτή μας, καθώς και η περιγραφή της λειτουργίας τους.

1.4.1 Arduino

Αυτό το κεφάλαιο είναι αφιερωμένο στην καρδιά του ελεγκτή ενσωματωμένος στην πλακέτα του Arduino όπως φαίνεται και στην παρακάτω εικόνα [2]

Εικόνα 1-6 Η πλακέτα Arduino Uno

Ιστορική Αναδρομή

Το Arduino ξεκίνησε σε ένα μικρό εργοστάσιο στην πόλη της Ιβρέα (Interaction Design Institute Ivrea γνωστό και ως Interaction Ivrea, IDII ή Ivrea), η οποία είναι κωμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας, στην ίδια περιοχή στην οποία στεγαζόταν η εταιρία υπολογιστών Olivetti. Στόχος ήταν να φτιαχτεί μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο φθηνή από τα άλλα πρωτότυπα συστήματα που ήταν διαθέσιμα εκείνη την περίοδο. Οι ιδρυτές Massimo Banzi και David Cueartielles ονόμασαν το σχέδιο Arduino από τον Arduin της Ιβρέα, βασιλιά της Ιταλίας. Το πρόγραμμα Arduino έλαβε τιμητική μνεία στην κατηγορία Digital Communities στο Prix Ars Electronica το 2006.

Τι είναι το Arduino

Το Arduino είναι μία υπολογιστική πλατφόρμα ανοιχτού κώδικα βασισμένη σε μία απλή μητρική πλακέτα με ενσωματωμένο μικροελεγκτή η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring, μια παραλλαγή της C/C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως το ATmega, και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++. Θα λέγαμε ότι είναι ένα εργαλείο για να κατασκευάσουμε ένα υπολογιστικό σύστημα με την έννοια ότι αυτό θα ελέγχει συσκευές του φυσικού κόσμου, σε αντίθεση με τον κοινό μας Ηλεκτρονικό Υπολογιστή. Μπορεί να χρησιμοποιηθεί στην ανάπτυξη διαδραστικών αντικειμένων, μπορεί να πάρει είσοδο από διάφορους διακόπτες ή αισθητήρες και να ελέγξει φώτα, κινητήρες και άλλες φυσικές εξόδους. Τα project μπορεί να είναι είτε stand-alone (αυτόνομα), είτε να επικοινωνούν με λογισμικό που τρέχει στον υπολογιστή όπως Flash, Processing και MaxMSP. Οι πλακέτες

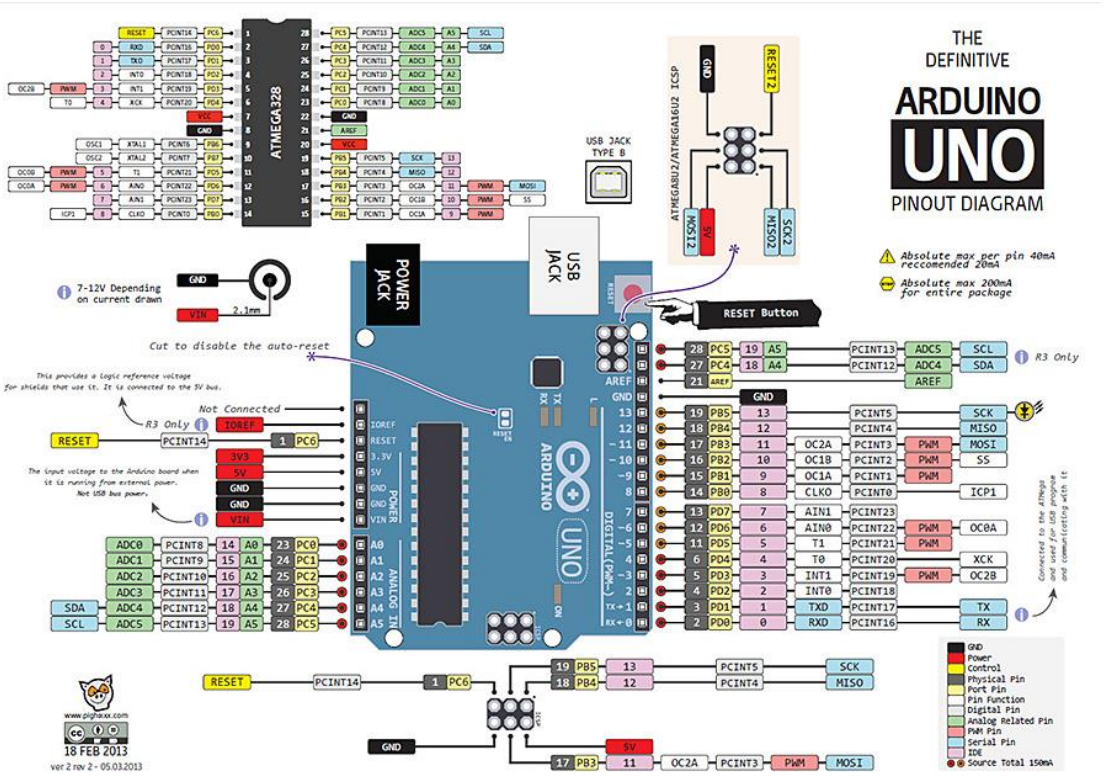
μπορούν να συναρμολογηθούν στο χέρι ή να αγοραστούν έτοιμες. Το περιβάλλον ανάπτυξης λογισμικού, το οποίο είναι ανοιχτού κώδικα, μπορεί κάποιος να το κατεβάσει δωρεάν από την επίσημη ιστοσελίδα του Arduino http://arduino.cc/en/Main/Software.

Γιατί το Προτιμούμε

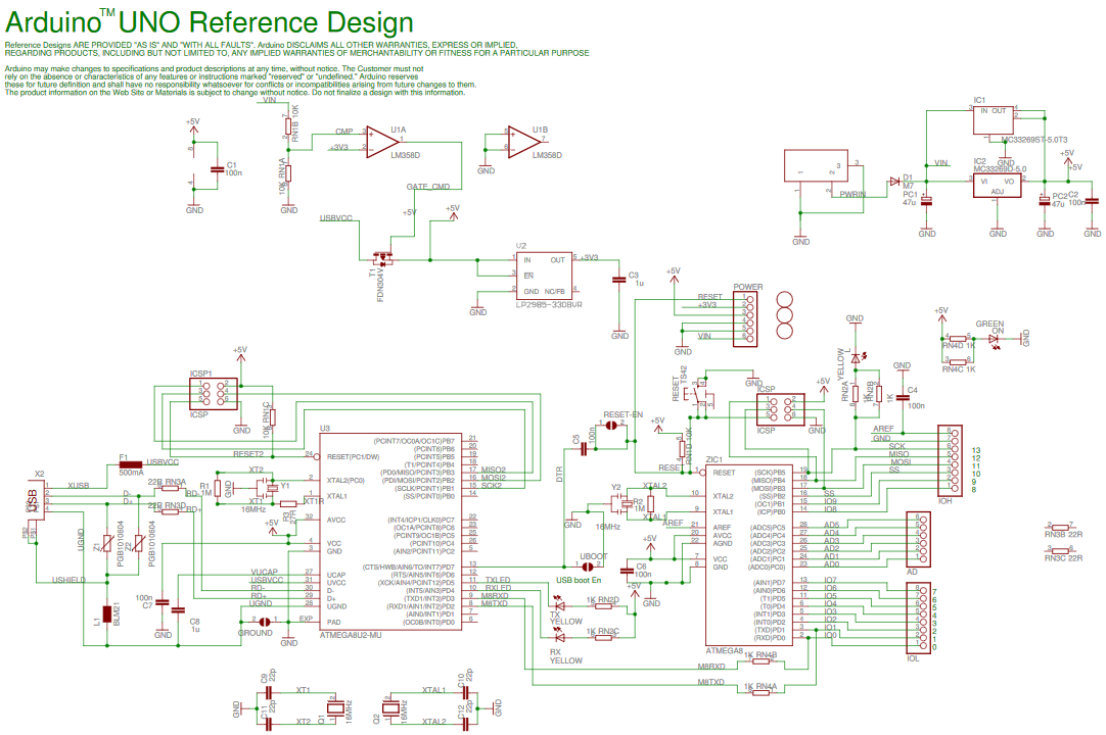
Υπάρχουν αρκετοί μικροελεγκτές διαθέσιμοι στην αγορά για κάποιον που θέλει να ασχοληθεί. Κάποιοι από αυτούς είναι ο Basic Stamp της Parallax, ο BX-24 της NetMedia, το Handyboard του MIT, και πολλοί άλλοι οι οποίοι προσφέρουν παρόμοιες δυνατότητες. Όλα αυτά τα εργαλεία που προαναφέραμε είναι απλά και 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. Το Arduino διαφέρει από τους προηγούμενους γιατί απλοποιεί την διαδικασία να δουλεύει κάποιος με μικροελεγκτές. Κάποια πλεονεκτήματα που προσφέρει σε σχέση με άλλους μικροελεγκτές για χρήση από δασκάλους, μαθητές και άλλους χομπίστες είναι τα παρακάτω:

- Είναι φθηνό. Οι πλακέτες Arduino είναι φθηνές σε σχέση με άλλους μικροελεγκτές. Η φθηνότερη εκδοχή του Arduino μπορεί να κατασκευαστεί στο χέρι αν και η έτοιμη πλακέτα δεν θα κοστίσει πάνω από τριάντα ευρώ.
- 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 άλλα συστήματα ανάπτυξης μικροελεγκτών περιορίζονται στα Windows.
- Απλό, ξεκάθαρο προγραμματιστικό περιβάλλον. Το περιβάλλον προγραμματισμού ενός Arduino ενδείκνυται για αρχάριους, αλλά είναι ταυτόχρονα 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 της C++ και οι άνθρωποι που θέλουν να ασχοληθούν περισσότερο με τους μικροελεγκτές μπορούν να μεταβούν από το Arduino στην AVR C που είναι για προγραμματισμό των Μικροελεγκτών Atmel 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 το Arduino.
- Ανοιχτού υλικού το οποίο μπορεί να επεκταθεί. Το Arduino βασίζεται στους μικροελεγκτές της Atmel ATMEGA8, ATMEGA168 και πλέον στο ATMEGA328. 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. [2]



Εικόνα 1-7 Χάρτης του Arduino Uno



Εικόνα 1-8 Σχηματικό διάγραμμα Arduino

1.4.2 Αισθητήρας Hall

Όταν μεταλλικό έλασμα εισέρχεται μέσα σε μαγνητικό πεδίο με μια συγκεκριμένη γωνία, τότε δημιουργείται στα άκρα του ελάσματος διαφορά δυναμικού (δηλαδή τάση). [3]

Οι αισθητήρες Hall έχουν τρία καλώδια ή ακροδέκτες: ένα για τη γείωση, ένα για την τάση μπαταρίας ή αναφοράς και ένα για το σήμα εξόδου. Για να παράγει ένα σήμα εξόδου, ο αισθητήρας Hall πρέπει να τροφοδοτηθεί με μια τάση αναφοράς (που μπορεί να είναι 5 έως 12V ανάλογα με την εφαρμογή). Η αρχή λειτουργίας αυτή ανακαλύφθηκε το 1879 από τον Αμερικανό επιστήμονα Edwin H. Hall. Η ανακάλυψη δεν είχε ευρεία πρακτική χρήση τότε, αργότερα αποδείχτηκε πολύ χρήσιμη ώστε οι μηχανικοί να δημιουργήσουν μια συσκευή που να παράγει σήμα τετραγωνικής κυματομορφής (on-off).

Όταν ένα μεταλλικό έλασμα περνά μέσω του κενού αέρα μεταξύ του μαγνητικού πεδίου και του τσιπ πυριτίου, διακόπτει το μαγνητικό πεδίο και αναγκάζει την τάση παραγωγής του τσιπ να μειωθεί ξαφνικά στο μηδέν.

1.4.3 Φωτοβολταϊκό πάνελ

Με τον γενικό όρο «φωτοβολταϊκό» ονομάζεται η βιομηχανική διάταξη πολλών φωτοβολταϊκών κυττάρων σε μία σειρά. Στην ουσία πρόκειται για τεχνητούς ημιαγωγούς (συνήθως από πυρίτιο) οι οποίοι ενώνονται με σκοπό να δημιουργήσουν ένα ηλεκτρικό κύκλωμα σε σειρά. Οι ημιαγωγοί αυτοί απορροφούν φωτόνια από την ηλιακή ακτινοβολία και παράγουν μια Ηλεκτρική τάση. Αυτή η διαδικασία ονομάζεται «φωτοβολταϊκό φαινόμενο».

Τα φωτοβολταϊκά ανήκουν στη κατηγορία των Ανανεώσιμων Πηγών Ενέργειας (ΑΠΕ)

Η ώρα που χρειάστηκε για να φορτίσει τη μπαταρία στον ελεγκτή ήταν ελάχιστη (λιγότερη από μία), καθώς η απόδοσή του είναι 10W ενώ η κατανάλωση του ελεγκτή μετρήθηκε στα 0,72W. Στη διάρκεια της νύχτας (6-7 ώρες) η ενέργεια που χρειάζεται είναι περίπου 5W. Η απόδοση του πάνελ είναι σχεδόν ίδια καθ' όλη την διάρκεια της ημέρας, ακόμη και τις μέρες που δεν επικρατεί ηλιοφάνεια η απόδοση του πάνελ πέφτει ελάχιστα από τις προδιάγραφες του. [4]

Εικόνα 1-9 Φωτοβολταϊκό πάνελ

Γενικοί όροι

Φωτοβολταϊκό Φαινόμενο

Το φωτοβολταϊκό (Φ/Β) φαινόμενο αφορά τη μετατροπή της ηλιακής ενέργειας σε ηλεκτρική. Ανακαλύφθηκε το 1839 από τον Εντμόντ Μπεκερέλ (Alexandre-Edmond Becquerel). Περιληπτικά πρόκειται για την απορρόφηση της ενέργειας του φωτός από τα ηλεκτρόνια των ατόμων του Φ/Β στοιχείου και την απόδραση των ηλεκτρονίων αυτών από τις κανονικές τους θέσεις με αποτέλεσμα την δημιουργία ρεύματος. Όταν το ηλιακό φως προσπίπτει στο φωτοβολταϊκό κύτταρο, μέρος της ακτινοβολίας διεγείρει ηλεκτρόνια τα οποία μπορούν να κινούνται σχετικά ελεύθερα μέσα στον ημιαγωγό. Η εφαρμογή ηλεκτρικού πεδίου υποχρεώνει τα ελεύθερα ηλεκτρόνια να κινηθούν προς συγκεκριμένη κατεύθυνση, παράγοντας ηλεκτρικό ρεύμα του οποίου η ένταση καθορίζεται από τη ροή των ηλεκτρονίων και την εφαρμοζόμενη τάση στο φωτοβολταϊκό κύτταρο.

Φωτοβολταϊκή Διάταξη

Τα Φ/Β πλαίσια έχουν ως βασικό μέρος το ηλιακό στοιχείο (solar cell) που είναι ένας κατάλληλα επεξεργασμένος ημιαγωγός μικρού πάχους σε επίπεδη επιφάνεια. Η πρόσπτωση ηλιακής ακτινοβολίας δημιουργεί ηλεκτρική τάση και με την κατάλληλη σύνδεση σε φορτίο παράγεται ηλεκτρικό ρεύμα.

Τα Φ/Β στοιχεία ομαδοποιούνται κατάλληλα και συγκροτούν τα φωτοβολταϊκά πλαίσια ή γεννήτριες (module), τυπικής ισχύος από 20W έως 300W. Οι Φ/Β γεννήτριες συνδέονται ηλεκτρολογικά μεταξύ τους και δημιουργούνται οι φωτοβολταϊκές συστοιχίες (arrays).

Τεχνολογίες Φ/Β Στοιχείων

Τα φωτοβολταϊκά στοιχεία χωρίζονται σε δυο βασικές κατηγορίες

1. Κρυσταλλικού Πυριτίου

Μονοκρυσταλλικού πυριτίου, με ονομαστικές αποδόσεις πλαισίων 14,5% έως 21%,

Πολυκρυσταλλικού πυριτίου, με ονομαστικές αποδόσεις πλαισίων 13% έως 14,5%.2.

2. Λεπτών Μεμβρανών

Άμορφου Πυριτίου, ονομαστικής απόδοσης ~7%.

Χαλκοπυριτών CIS / CIGS, ονομαστικής απόδοσης από 7% έως 14%. [5]

Το πυρίτιο (Si) είναι η βάση για το 90% περίπου της παγκόσμιας παραγωγής Φ/Β. Η κυριαρχία αυτή οφείλεται αρχικά στην τεράστια παγκόσμια επιστημονική και τεχνική υποδομή για το υλικό αυτό από τη δεκαετία του '60. Μεγάλες κυβερνητικές και βιομηχανικές επενδύσεις έγιναν σε προγράμματα για τις χημικές και ηλεκτρονικές ιδιότητες του Si, ώστε να δημιουργηθεί ο εξοπλισμός που απαιτείται στα βήματα της επεξεργασίας για την απόκτηση της απαραίτητης καθαρότητας και της κρυσταλλικής δομής του υλικού.

Η γνώση που προέκυψε έτσι για το πυρίτιο, τα χαρακτηριστικά του και η αφθονία του στη γη, το κατέστησαν ικανό και συμφέρον μέσο για την εκμετάλλευση της ηλιακής ενέργειας. Εντούτοις, λόγω του ότι είναι εύθραυστο, το πυρίτιο απαιτεί τον σχηματισμό στοιχείων σχετικά μεγάλου πάχους. Αυτό σημαίνει ότι μερικά από τα ηλεκτρόνια που απελευθερώνονται μετά την απορρόφηση της ηλιακής ενέργειας πρέπει να ταξιδέψουν μεγάλες αποστάσεις για να ενταχθούν στην ροή του ρεύματος και να συνεισφέρουν στο ηλεκτρικό κύκλωμα. Συνεπώς, το υλικό θα πρέπει να έχει υψηλή καθαρότητα και δομική τελειότητα, ώστε να αποτρέψει την επιστροφή των ηλεκτρονίων στις φυσικές τους θέσεις. Οι ατέλειες πρέπει να αποφευχθούν ώστε η ενέργεια του ηλεκτρονίου να μην μετατραπεί σε θερμότητα. Η παραγωγή θερμότητας, η οποία είναι επιθυμητή στα ηλιακά θερμικά πλαίσια, όπου αυτή η θερμότητα μεταφέρεται σε ένα ρευστό, είναι ανεπιθύμητη στα Φ/Β πλαίσια, όπου η ηλιακή ενέργεια θα πρέπει να μετατραπεί σε ηλεκτρική.

Το πυρίτιο, ανάλογα με την επεξεργασία του, δίνει μονοκρυσταλλικά, πολυκρυσταλλικά ή άμορφα υλικά, από τα οποία παράγονται τα Φ/Β στοιχεία. Τα λεπτά υλικά είναι ένας τρόπος να μειωθεί το κόστος των Φ/Β πλαισίων και να αυξηθεί η απόδοσή τους. Εκτός από τη χρήση μικρότερης ποσότητας υλικού, ένα άλλο πλεονέκτημα είναι ότι ολόκληρα πλαίσια μπορούν να κατασκευαστούν παράλληλα με τη διαδικασία απόθεσης. Αυτό είναι συμφέρον οικονομικά, αλλά επίσης πολύ απαιτητικό τεχνικά, επειδή η επεξεργασία χωρίς ατέλειες αφορά μεγαλύτερη επιφάνεια.

Στα πλεονεκτήματα των πλαισίων λεπτού υμενίου τα οποία αναφέρθηκαν παραπάνω, θα πρέπει να αντιπαρατεθεί η ελαφρώς χαμηλότερη απόδοσή τους, που φτάνει μέχρι 14% στα τεχνολογίας CIS / CISG. Οι άλλες τεχνολογίες λεπτού υμενίου φτάνουν περίπου μέχρι 10%, ανάλογα με το υλικό. Πάντως η τεχνολογία λεπτού στρώματος (thin film) είναι σε φάση ανάπτυξης, αφού με διάφορες μεθόδους επεξεργασίας και χρήση διαφορετικών υλικών αναμένεται αύξηση της απόδοσης, σταθεροποίηση των χαρακτηριστικών τους και αύξηση της διείσδυσης στην αγορά. Σήμερα πάντως αποτελούν την πιο φθηνή επιλογή Φ/Β πλαισίων. [5]

Αρχή λειτουργίας φωτοβολταϊκών

Η λειτουργία του φωτοβολταϊκού συστήματος στηρίζεται στις βασικές ιδιότητες των ημιαγωγών υλικών σε ατομικό επίπεδο.

Εάν φέρουμε σε επαφή δύο κομμάτια πυριτίου τύπου n και τύπου p, δημιουργείται μια δίοδος η αλλιώς ένα ηλεκτρικό πεδίο στην επαφή των δύο υλικών το οποίο επιτρέπει την κίνηση ηλεκτρονίων προς μια κατεύθυνση μόνο. Τα επιπλέον ηλεκτρόνια της επαφής n έλκονται από τις οπές τις επαφής p. Αυτό το ζευγάρι των δύο υλικών είναι το δομικό στοιχείο του φωτοβολταϊκού κελιού και η βάση της φωτοβολταϊκής τεχνολογίας.

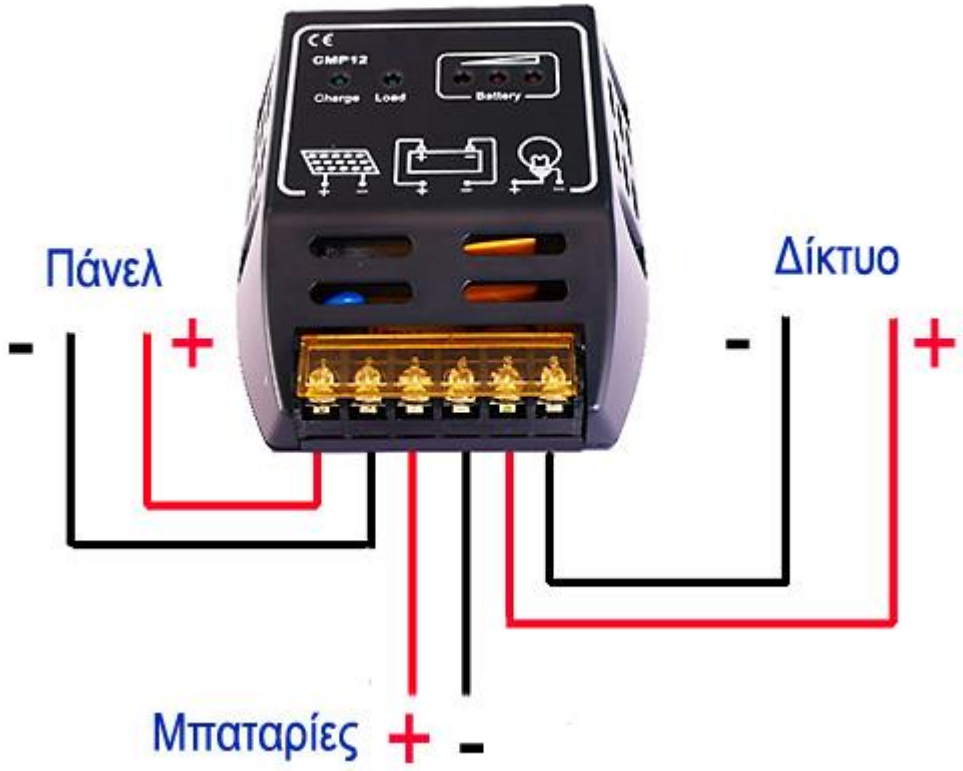
Η ηλιακή ακτινοβολία έρχεται με την μορφή πακέτων ενέργειας ή φωτονίων. Τα φωτόνια όταν προσπίπτουν σε μια διάταξη φβ κελιού περνούν αδιατάραχτα την επαφή τύπου n και χτυπούν τα άτομα της περιοχής τύπου p.

Τα ηλεκτρόνια της περιοχής τύπου p αρχίζουν και κινούνται μεταξύ των οπών ώσπου τελικά φτάνουν στην περιοχή της διόδου όπου και έλκονται πλέον από το θετικό πεδίο της εκεί περιοχής. Αφού ξεπεράσουν το ενεργειακό χάσμα αυτής της περιοχής μετά είναι αδύνατον να επιστρέψουν. Στο κομμάτι της επαφής n πλέον έχουμε μια περίσσεια ηλεκτρονίων που μπορούμε να εκμεταλλευτούμε. Αυτή η περίσσεια των ηλεκτρονίων μπορεί και παράγει τελικά το ηλεκτρικό ρεύμα, εάν τοποθετήσουμε μια διάταξη όπως έναν μεταλλικό αγωγό στο πάνω μέρος της επαφής n και στο κάτω της επαφής p και το φορτίο ενδιάμεσα έτσι ώστε να ολοκληρωθεί ο αγώγιμος δρόμος για το ηλεκτρικό ρεύμα που παράγεται. Αυτή είναι απλοποιημένα η γενική αρχή λειτουργίας του φωτοβολταϊκού φαινομένου.

1.4.4 Ελεγκτής φόρτισης

Ο ελεγκτής, ή αλλιώς ρυθμιστής φόρτισης, αποτελεί μέρος κάθε αυτόνομου φωτοβολταϊκού συστήματος. Ακόμη και πολύ μικρής κλίμακας να είναι ο ελεγκτής φόρτισης είναι απαραίτητος και δεν πρέπει να λείπει από καμία υλοποίηση, ακόμη και από αυτές που φαίνεται να μην είναι απαραίτητος. Το κόστος του μπορεί στις μέρες μας να είναι πολύ χαμηλό, ειδικά αν βρούμε κάποιο μοντέλο

που απευθύνεται σε ερασιτέχνες. Έτσι και αλλιώς οι μπαταρίες που χρησιμοποιούμε στα φωτοβολταϊκά (lead acid) θα μπορούσαν εύκολα να υποστούν ζημιές αν απουσίαζε ο ελεγκτής φόρτισης όπως για παράδειγμα όταν μια πλήρως φορτισμένη μπαταρία συνεχίσει να δέχεται υψηλό φορτίο από τα ηλιακά πάνελ ή τις ανεμογεννήτριες. Ο ελεγκτής φόρτισης αναλαμβάνει όχι μόνο να ελέγξει την φόρτιση των μπαταριών αλλά και να τις προστατέψει. Όταν λοιπόν φορτίζεται μια μπαταρία συνδεμένη στον ελεγκτή, θα φτάσει μέχρι μια τάση ασφαλείας και όχι παραπάνω. Παρόλο που μπορεί να συνεχίσει ο ήλιος να παρέχει ενέργεια και υψηλή τάση, ο ελεγκτής αποφασίζει πως δεν χρειάζεται περεταίρω φόρτιση και την διακόπτει. Ο ελεγκτής όμως φροντίζει και όταν η μπαταρία κοντεύει να τελειώσει. Δεν πρέπει ποτέ οι μπαταρίες να ξεφορτίζονται πλήρως. Ο ρυθμιστής φόρτισης δεν αφήνει τις μπαταρίες να τελειώσουν και κλείνει «τον γενικό» με σκοπό να τις προστατέψει.[6]



Εικόνα 1-10 Ελεγκτής φόρτισης

Τρόπος λειτουργίας ενός ελεγκτή φόρτισης

Ένας τυπικός ρυθμιστής φόρτισης, ακόμη και ο πιο απλός, έχει 3 ζεύγη για υποδοχή ακροδεκτών. Το ένα δέχεται το ρεύμα από τα ηλιακά πάνελ ή τις ανεμογεννήτριες. Το άλλο στέλνει αυτό το ρεύμα στις μπαταρίες έτσι ώστε να μπορεί να αποφασιστεί αν πρέπει να συνεχιστεί η φόρτιση ή όχι. Τέλος το τρίτο ζεύγος δίνει το ρεύμα χρήσης. Δηλαδή είναι το σημείο από το οποίο στην ουσία ξεκινά η νησίδα παροχής ηλεκτρικής ενέργειας. Υπάρχουν πολύ πιο εξελιγμένοι

ελεγκτές φόρτισης από αυτό που περιγράφηκε παραπάνω. Στο διαδίκτυο μπορεί να βρει κάποιος πολύ φτηνούς ελεγκτές φόρτισης. Στις υλοποιήσεις που απασχολούν αυτά τα φτηνά εξαρτήματα αρκούν. Το κύριο χαρακτηριστικό για την επιλογή τους είναι το ρεύμα που μπορούν να δώσουν. Ένας ελεγκτής φόρτισης των 5A, 10A ή 20A είναι ένα τυπικό παράδειγμα για μικρές υλοποιήσεις. Επίσης η τάση λειτουργίας είναι σημαντική. Αν το σύστημά μας είναι στα 12V τότε τέτοιας τάσης πρέπει να είναι και ο ελεγκτής για την φόρτιση.

Εικόνα 1-11 Τυπικός ρυθμιστής φόρτισης

Οι ρυθμιστές φόρτισης πέραν της προφανούς λειτουργίας που επιτελούν, παρέχουν και κάποιους αυτοματισμούς που είναι ίσως αρκετοί για τις ερασιτεχνικές εφαρμογές. Επειδή η πιο συχνή αιτία που κάποιος υλοποιεί το δικό του φωτοβολταϊκό σύστημα είναι ο φωτισμός, οι πιο πολλοί κατασκευαστές ελεγκτών φόρτισης φροντίζουν ακόμη και τα πιο απλά προϊόντα τους να παρέχουν αυτοματισμούς για τον φωτισμό. Ο λόγος είναι απλός. Πολύ συχνά θέλουμε φωτισμό ασφαλείας σε επαγγελματικές στέγες για το βράδυ, να φωτίσουμε ένα μνημείο ή απλώς να υπάρχει ένα φως στο σπίτι που θα μένει όλο το βράδυ ανοιχτό ανεξάρτητα αν έχει κοπεί το κανονικό ρεύμα ή όχι. Οι ρυθμιστές φόρτισης λύνουν αυτό το πρόβλημα πολύ εύκολα και ανέξοδα.

Έχουν την δυνατότητα :

- Να δίνουν ρεύμα όσο υπάρχει απουσία φωτός (όλο το βράδυ δηλαδή)
- Να δίνουν ρεύμα για Χ ώρες μετά την δύση του ηλίου
- Να δίνουν ρεύμα Χ ώρες πριν την ανατολή του ηλίου

- Να δίνουν ρεύμα όποτε θέλει ο χρήστης, σαν γενικός διακόπτης.

Με τις παραπάνω πληροφορίες μπορεί ο καθένας να φανταστεί ότι καλύπτονται οι περισσότερες των περιπτώσεων για ένα μικρό αυτόνομο σύστημα. Τέλος μια σημαντική λειτουργία που κανείς δεν πρέπει να ξεχνά, είναι ότι οι ελεγκτές έχουν την δυνατότητα να διακόψουν κάθε παροχή ρεύματος σε περίπτωση βραχυκυκλώματος. Αυτό μπορεί να γλιτώσει πολλά ατυχήματα. Η τάση σε αυτά τα δίκτυα μπορεί να είναι μικρή αλλά σπινθήρες μπορούν να προκληθούν πολύ εύκολα στα 12 και 24 V. Αν αυτοί συνεχιστούν και κοντά υπάρχει κάποιο εύφλεκτο υλικό το κακό δεν αργεί να γίνει. Όμως ο ελεγκτής στο πρώτο δευτερόλεπτο μετά την ρευματοδότηση κλείνει την παροχή, σαν γενικός. Ένα λαμπάκι πάνω του συνήθως σου δείχνει από μακριά καθώς αναβοσβήνει ότι κάτι πήγε στραβά.

1.4.5 Μπαταρία

Η μπαταρία ή ηλεκτρικός συσσωρευτής (ενίοτε και απλά συσσωρευτής) είναι η συσκευή η οποία αποθηκεύει χημική ενέργεια και την αποδεσμεύει με τη μορφή ηλεκτρισμού. Για το σκοπό αυτό χρησιμοποιούνται ηλεκτροχημικές διατάξεις όπως η γαλβανική στήλη. Η ανάπτυξη των μπαταριών άρχισε με την κατασκευή της Βολταϊκής στήλης από τον Αλεσάντρο Βόλτα. Εικάζεται όμως ότι κάποια αντικείμενα, που χρονολογούνται γύρω στο έτος 600 και είναι γνωστά ως μπαταρίες της Βαγδάτης. [7]

Εικόνα 1-12 Μπαταρία

Περιγραφή

Ο συσσωρευτής στην ηλεκτρολογία είναι χημική πηγή ρεύματος, ικανή να αποθηκεύσει ηλεκτρική ενέργεια (αφού τη μετατρέψει σε χημική) και όταν χρειαστεί, να την αποδώσει σε εξωτερικό κύκλωμα. Αποτελείται από δοχείο κατασκευασμένο από μονωτικό υλικό (εβονίτη, πλαστικό, γυαλί) με ηλεκτρολύτη (οξύ ή

αλκάλιο), στο οποίο βυθίζονται τα ηλεκτρόδια. Η σύνδεσή τους σε εξωτερικό κύκλωμα προκαλεί σε αυτό διέλευση ρεύματος (εκ φόρτιση του ηλεκτρικού συσσωρευτή). Έτσι, στον ηλεκτρικό συσσωρευτή γίνονται χημικές διεργασίες, που έχουν σχέση με τη μετατροπή της χημικής ενέργειας σε ηλεκτρική.

Ο εκφορτισμένος ηλεκτρικός συσσωρευτής φορτίζεται όταν περάσει από αυτόν συνεχές ρεύμα από άλλη πηγή, ενώ ταυτόχρονα στον ηλεκτρικό συσσωρευτή γίνονται αντίστροφες χημικές διεργασίες, με τις οποίες η ηλεκτρική ενέργεια μετατρέπεται σε χημική. Ο ηλεκτρικός συσσωρευτής χαρακτηρίζεται από τη χωρητικότητα, δηλ. την ποσότητα του ηλεκτρισμού σε αμπερώρια, που μπορεί ο συσσωρευτής να δώσει στο κύκλωμα που τροφοδοτεί, από τη μέση τάση σε V κατά το χρόνο της φόρτισης και εκφόρτισης, από την ειδική ενέργεια κατά βάρος και όγκο, δηλ. την ενέργεια σε βατώρια που παρέχεται κατά την εκφόρτιση από 1 kgr βάρους ή 1 δεκατόμετρο του όγκου του ηλεκτρικού συσσωρευτή, από την απόδοση κατά χωρητικότητα, δηλ. τον λόγο της ποσότητας των αμπερωρίων που αποδίδεται κατά την εκφόρτιση προς την ποσότητα των αμπερωρίων που απορροφάται κατά τη φόρτιση, από την απόδοση κατά ενέργεια (ή βαθμό απόδοσης), δηλ. το λόγο της ενέργειας που αποδίδεται κατά την εκφόρτιση προς την ενέργεια που απορροφάται κατά τη φόρτιση. Υπάρχουν ηλεκτρικοί συσσωρευτές σε μόνιμη εγκατάσταση (για τις ανάγκες των ηλεκτρικών σταθμών, των τηλεφωνικών και τηλεγραφικών σταθμών, των ραδιοσταθμών κ.ά.) και φορητοί (για τροφοδότηση κινητών ραδιοσυσκευών και συσκευών ενσύρματης επικοινωνίας, αυτοκινήτων, αεροπλάνων κ.ά.).

Στις εφαρμογές που γίνεται η χρήση φωτοβολταϊκού πάνελ για την φόρτιση της μπαταρίας επιλέγεται μπαταρία βαθειάς εκφόρτισης.

Χαρακτηριστικά μπαταρίας βαθειάς φόρτισης.

- Δεν παρέχουν ρεύμα εκκίνησης μεγάλης έντασης
- Πολλές φορτίσεις – εκφορτίσεις
- Εκφόρτιση μέχρι και 40%
- Διάφορα μεγέθη και σχήματα για εύκολη τοποθέτηση

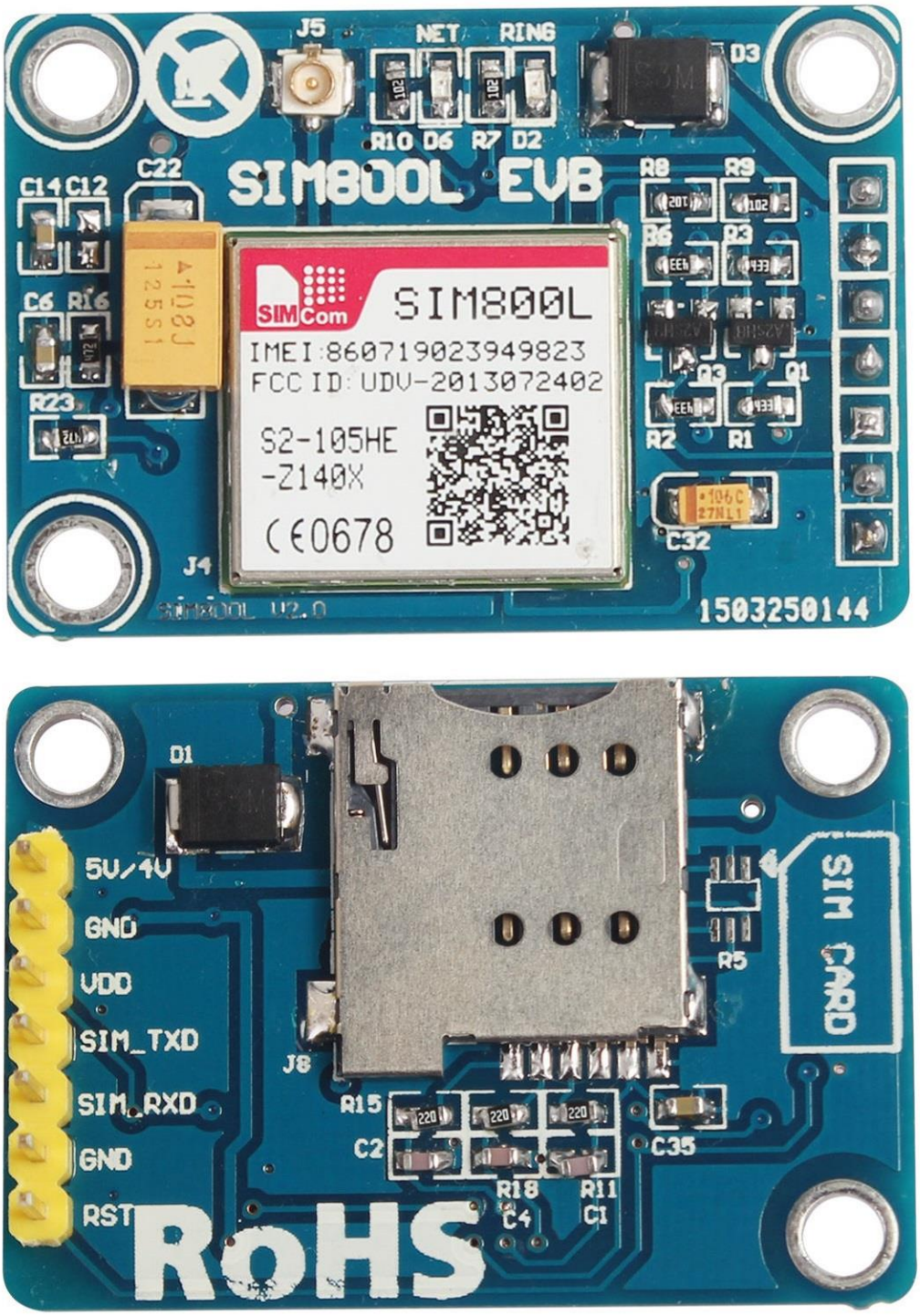
Η διάρκεια μπαταρίας εξαρτάται από το φορτίο που τροφοδοτεί και από την χωρητικότητά της. Ο ελεγκτής του ποτιστικού καρουλιού έχει κατανάλωση 0,06A και η μπαταρία που χρησιμοποιείται είναι 12V 7Ah δηλαδή το φορτίο μας είναι 0,72W. Θεωρητικά, η διάρκεια της μπαταρίας ξεπερνά τις 4 ημέρες. Το φωτοβολταϊκό πάνελ έχει ονομαστική απόδοση 10W όποτε υπερκαλύπτει το μέγεθος του φορτίου και διατηρεί την μπαταρία σχεδόν συνέχεια σε πλήρη φόρτιση.

1.4.6 SIM – 800 gsm shield

Το GPRS Shield είναι βασισμένο στο module SIM800 της εταιρίας SIMCOM και είναι συμβατό με τον Arduino Uno. Αυτό το Shield μας παρέχει την ικανότητα

να επικοινωνούμε με το δίκτυο κινητής τηλεφωνίας GSM. Αυτή η δυνατότητα μας επιτρέπει να επιτύχουμε συνδυασμό SMS (Short Message Service), MMS, GPRS και ήχο μέσω UART στέλνοντας εντολές AT (AT commands). Αρκεί να τοποθετήσουμε μια κάρτα sim και με τις κατάλληλες εντολές έχουμε συνδέσει ένα κινητό με τον ελεγκτή μας. Τον sim800 στην αγορά το συναντάμε σε διάφορες εκδόσεις ανάλογα με την πλακέτα υλοποίησης. Στην δικιά μας κατασκευή θα χρησιμοποιηθεί, όπως φαίνεται και στην (Εικόνα 1-14) η έκδοση L.

Εικόνα 1-13 Διάγραμμα ακροδεκτών του SIM800



Εικόνα 1-14 SIM800L

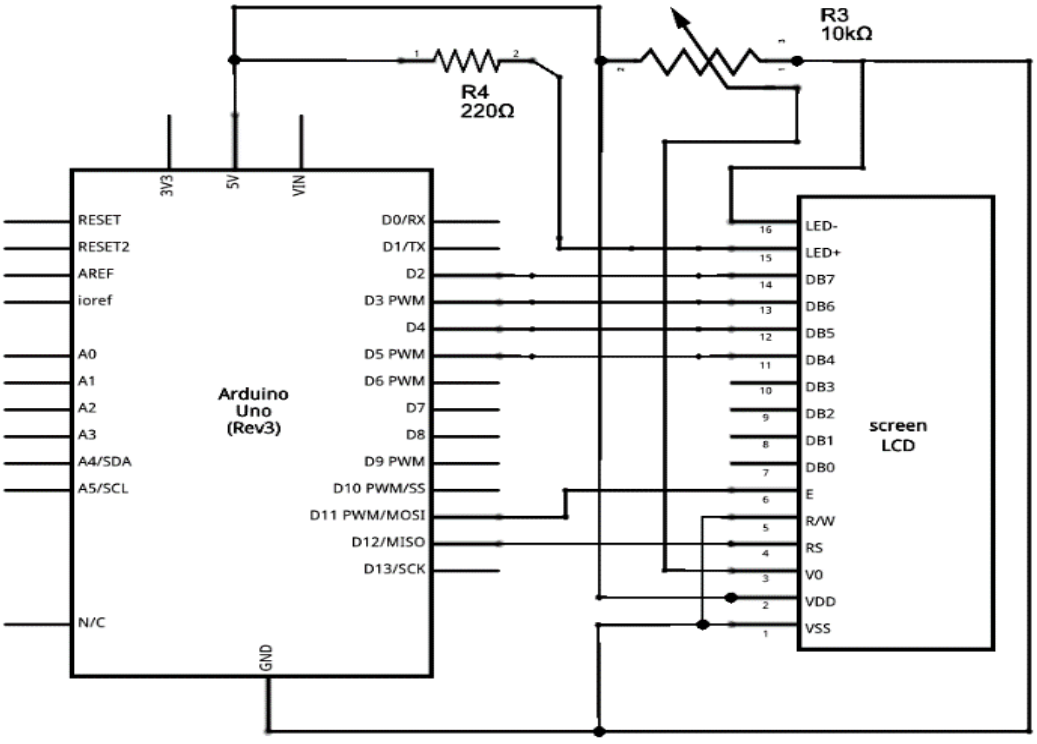
1.4.7 Οθόνη LCD 1602

Πήγες: [8]

Η οθόνη μας είναι 2 γραμμών και 16 χαρακτήρων σε κάθε γραμμή.

Εικόνα 1-15 Οθόνη 16X2

Συνήθως έχει κίτρινο φόντο και μαύρους χαρακτήρες ή μαύρο φόντο και λευκούς χαρακτήρες. Υπάρχει οπίσθιος φωτισμός που κατά επιλογή μας ενεργοποιείται. Η οθόνη της κατασκευής οδηγείται από το Hitachi HD44780 το οποίο είναι και το πιο διαδεδομένο.



Εικόνα 1-16 Σύνδεση με Arduino Uno

Στην (Εικόνα 1-16) φαίνεται η σύνδεσή της με τον Arduino Uno. Όπως βλέπουμε υπάρχουν 2 αντιστάσεις, η R4 που προστατεύει τον φωτισμό από με-

γάλο ρεύμα και η R3, μια μεταβλητή αντίσταση των 10ΚΩ η οποία είναι απαραίτητη για να ρυθμίσουμε την αντίθεση της οθόνης. Η τάση με την οποία πρέπει να το τροφοδοτήσουμε είναι 5V.

1.4.8 Χαρακτηριστικά

Ο χειριστής του καρουλιού, και κατά επέκταση και ο χειριστής του ελεγκτή, θα έχει αρκετά και σημαντικά οφέλη. Θα γίνει μια επιγραμματική αναφορά σε αυτά και παρακάτω θα γίνει εκτενέστερη ανάλυση.

Πλεονεκτήματα για τον χειριστή – αγρότη

- Εξοικονόμηση χρήματων
- Οικονομία χρόνου
- Μη άσκοπη μετακίνηση
- Ακρίβεια στο χρονοδιάγραμμα των εργασιών του
- Απομακρυσμένη πληροφόρηση για την πορεία του ποτίσματος
- Εξάλειψη άγχους για την ομαλή διαδικασία του ποτίσματος

Ένα από τα βασικότερα πλεονεκτήματα της κατασκευής, που είναι και ο σκοπός της κατασκευής μας, είναι η απομακρυσμένη ενημέρωση (1.4.6) του χρήστη για την εξέλιξη της διαδικασίας του ποτίσματος. Σε αντίθετη περίπτωση, θα έπρεπε ανά τακτά χρονικά διαστήματα ο χρήστης να πηγαίνει επιτόπου στο χωράφι για να δει αν όλα πηγαίνουν καλά, όμως και πάλι δε θα μπορούσε να έχει την ακρίβεια των πληροφοριών που θα του δώσει ο ελεγκτής μας. Έτσι κερδίζει χρήμα, από τις άσκοπες μετακινήσεις (πάλι χρήματα –φθορές αυτοκίνητου), όπως επίσης κερδίζει και χρόνο στον οποίο θα μπορεί να κάνει το οτιδήποτε άλλο. Γνωρίζοντας με ακρίβεια σε πόση ώρα θα ολοκληρωθεί η διαδικασία μπορεί να δρομολογήσει καλυτέρα τις υπόλοιπες εργασίες του, και να εκμεταλλευτεί καλυτέρα το χρόνο του. Ένα άλλο επίσης βασικό πρόβλημα που λύνει ο εκλεκτής είναι το άγχος που υπάρχει για την διαδικασία ποτίσματος. Ποτέ θα ολοκληρωθεί; Πόση ώρα άραγε θέλει; Πηγαίνουν άραγε όλα καλά; Υπάρχει κάποιο πρόβλημα, κάποια δυσλειτουργία; Όλα αυτά είναι ερωτήματα που χωρίς τον ελεγκτή δε μπορούν απαντηθούν. Ο χρήστης έχει την ελευθερία και την ηρεμία χωρίς να χρειάζεται να επισκέπτεται το χωράφι και το καρούλι συνέχεια.

Πλεονεκτήματα και οφέλη για την καλλιέργεια

- Πρόληψη τυχόν αστοχίας – καταστροφή παραγωγής
- Ακρίβεια στην ταχύτητας ποτίσματος – ομοιόμορφο πότισμα
- Αποφυγή άσκοπης κατανάλωσης νερού

Δεν είναι λίγες οι φορές που η διαδικασία του ποτίσματος και η λειτουργία του καρουλιού διακόπτεται. Οι λόγοι είναι αρκετοί, θα αναφέρουμε αυτούς που συναντάμε συχνότερα.

Ο συνηθέστερος λόγος που μπορεί να σταματήσει το μάζεμά του (1.3.4), το καρούλι είναι η βλάβη στην γραμμή υδροδότησής του και σταματήσει να τροφοδοτείται με νερό . Με την σειρά του και αυτό έχει μια λίστα από λόγους για τους οποίους θα μπορούσε να συμβεί. Η διακοπή λόγω βλάβης του δικτύου ύδρευσης είναι μια περίπτωση που συναντάμε συχνά. Για τις περιοχές όπου η άντληση νερού γίνεται με γεωτρήσεις, κάποιο πρόβλημα στην αντλία ή στον ηλεκτρολογικό της έλεγχο. Σε τέτοια περίπτωση δεν υπάρχει σοβαρός κίνδυνος ζημίας της παραγωγής. Ο ελεγκτής μας ενημερώνει για την διακοπή της διαδικασίας άλλα και για την επανέναρξή της.

Μια περίπτωση να πλημυρίσει το χωράφι υπάρχει, αν το πρόβλημα της πηγής δεν είναι η διακοπή παροχής του νερού, αλλά η μεγάλη πτώση της πίεσης. Το καρούλι τότε, σταματάει και ποτίζει συνέχεια στο ίδιο μέρος μέχρι εμείς να το αντιληφθούμε. Επίσης, μπορεί να μη κινδυνεύει το χωράφι, αλλά να υπάρχει σοβαρό πρόβλημα στην γεώτρηση.

Μια σοβαρή περίπτωση να κινδυνεύσει η καλλιέργεια, είναι η περίπτωση που το πρόβλημα δεν είναι της πηγής νερού, αλλά της ίδιας της γραμμής τροφοδοσίας. Κάποιο λάστιχο έχει τρυπήσει ή έχει σκιστεί ή κάποιος σωλήνας έχει αποσυνδεθεί. Σε τέτοιες περιπτώσεις και με τις πιέσεις να φτάνουν τα 5-7 bar η ανεξέλεγκτη ροή του νερού μπορεί να καταστρέψει την καλλιέργεια. Ακόμη χειροτέρα μπορεί να καταστρέψει το χωράφι του γείτονα ή τους αγροτικούς δρόμους. Με την έγκυρη ενημέρωση από τον ελεγκτή, θα μπορέσουμε να είμαστε όσο το δυνατόν γρηγορότερα στο χωράφι ώστε να περιορίσουμε ή να προλάβουμε την ζημία.

Σε δίκτυα υδροδότησης κοινής χρήσης και ειδικά σε περιόδους όταν η ζήτηση νερού είναι μεγάλη, παρατηρούνται μεγάλες αλλαγές στην πίεση. Ένα χαρακτηριστικό παράδειγμα είναι να βάλουμε το καρούλι να ποτίσει και μετά από εμάς να κάνουν το ίδιο 4-5 γείτονες, ή το αντίθετο, να ξεκινήσουμε το πότισμα και να σταματήσουν αρκετοί γύρω μας. Αυτό έχει ως αποτέλεσμα την μεγάλη αλλαγή στην ταχύτητα (1.3.4) του καρουλιού και στην ποσότητα νερού που πέφτει στο χωράφι μας. Για τέτοιες αλλαγές ενημερωνόμαστε από τον ελεγκτή ώστε να κάνουμε τις απαραίτητες ενέργειες.

Έχοντας σταθερή ροή εισόδου (σε δίκτυα με γεωτρήσεις ή σε περιόδους που το δίκτυο καλύπτει επαρκώς τις ανάγκες του) ο ελεγκτής δίνει την δυνατότητα να ρυθμίσουμε με ακρίβεια την ταχύτητα που θα ποτίζουμε, έτσι ώστε σε κάθε μετακίνηση του να έχουμε ομοιόμορφο πότισμα. Αυτή η επιλογή δίνεται γνωρίζοντας την ακριβή ταχύτητα η οποία απεικονίζεται στην οθόνη του ελεγκτή. Η επιλογή της ταχύτητας γίνεται από το κιβώτιο ταχυτήτων (χειροκίνητα από τον αγρότη) που διαθέτει το καρούλι. (1.3.4)

Ένα πολύ συνηθισμένο πρόβλημα είναι η λανθασμένη εκτίμηση της ώρας που θα ολοκληρωθεί το πότισμα, έτσι το καρούλι έχει φτάσει στο τέλος του και ρίχνει

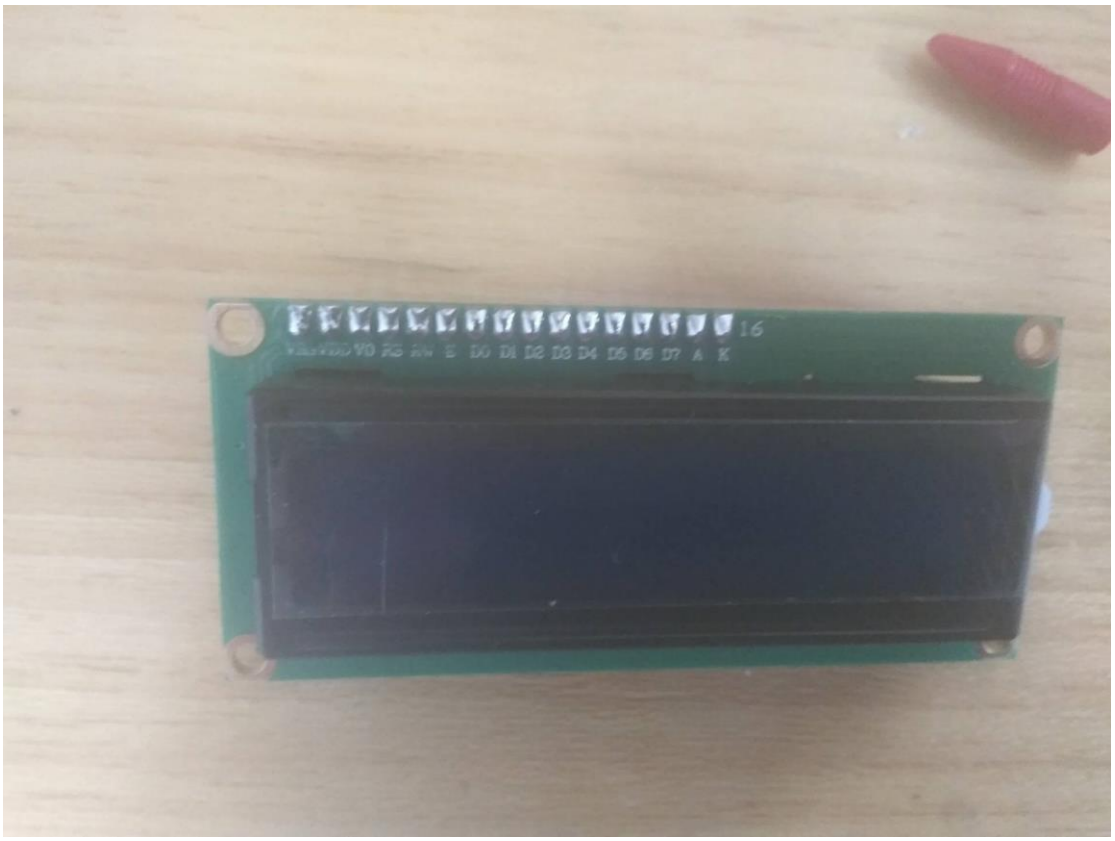
στο ίδιο μέρος νερό μέχρι εμείς να φτάσουμε εκεί. Αν ο χρόνος αυτός είναι μεγάλος (για μερικές καλλιέργειες και η μισή ώρα θα μπορούσε να είναι καταστροφική) προκαλούμε μερική ζημιά στην καλλιέργεια και κάνουμε άσκοπη χρήση νερού. Σε κάποιες περιόδους αυτό δημιουργεί πρόβλημα στο δίκτυο και κατά επέκταση σε όλους τους αγρότες.

Όλα τα παραπάνω προβλήματα, είναι προβλήματα που παρουσιάζονται με αρκετά μεγάλη συχνότητα. Αυτά αποσκοπεί να λύσει ο ελεγκτής ή να τα περιορίσει όσο το δυνατόν περισσότερο.

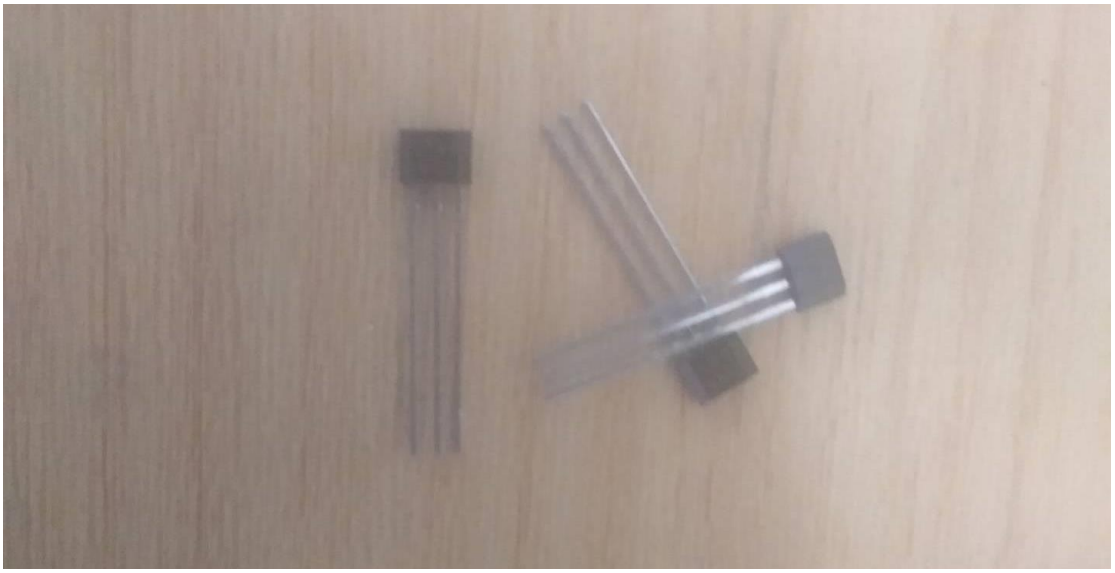
2. Υλοποίηση κατασκευής ελεγκτή

Σε αυτό το κεφάλαιο θα δείξουμε την υλοποίηση της κατασκευής μας. Θα υπάρχει αντίστοιχο φωτογραφικό υλικό. Θα υπάρχει το σχηματικό του κυκλώματος καθώς και το μυαλό της κατασκευής μας που είναι ο κώδικάς μας. Ο κώδικας θα συνοδεύεται από λογικά διαγράμματα και από κάποια σχόλια για την επεξήγησή του. Στο τέλος, θα δείξουμε και κάποιες φωτογραφίες, που θα παρουσιάζουν τον ελεγκτή μας τοποθετημένο πάνω στο ποτιστικό καρούλι.

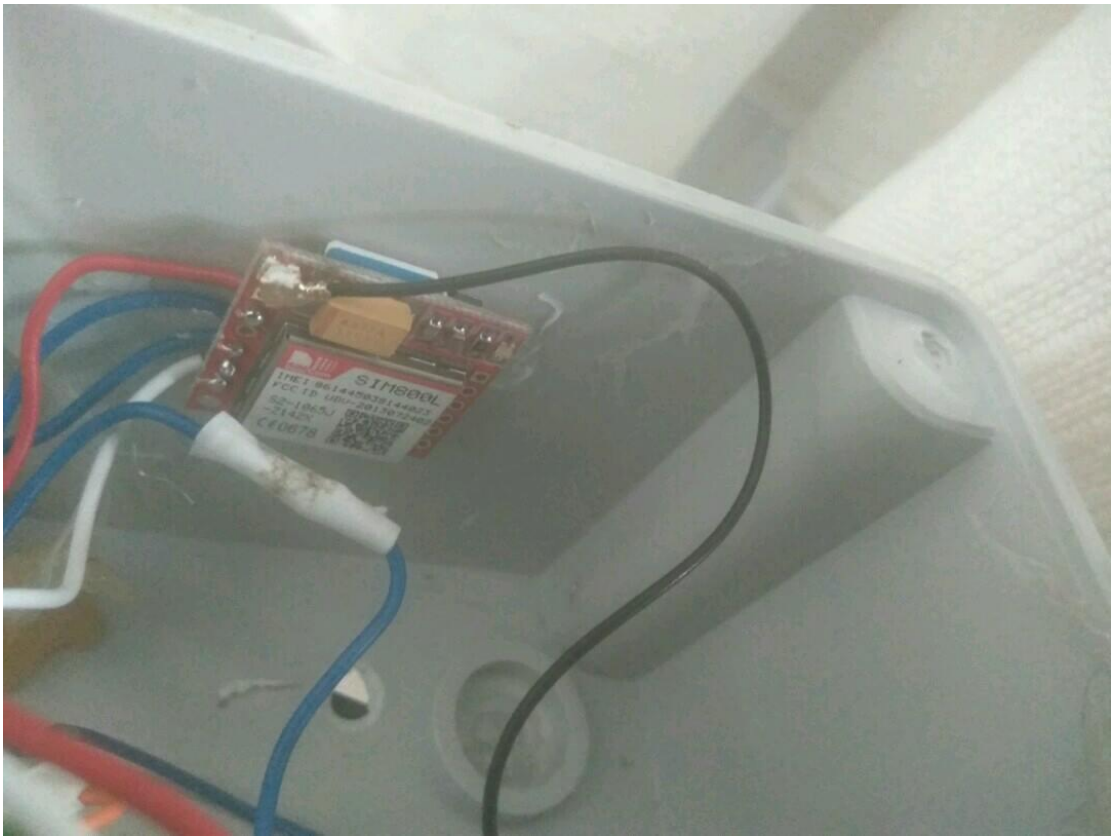
1.5 Υλικό – φωτογραφίες



Εικόνα 2-1 Οθόνη 2X16



Εικόνα 2-2 Αισθητήρες hall

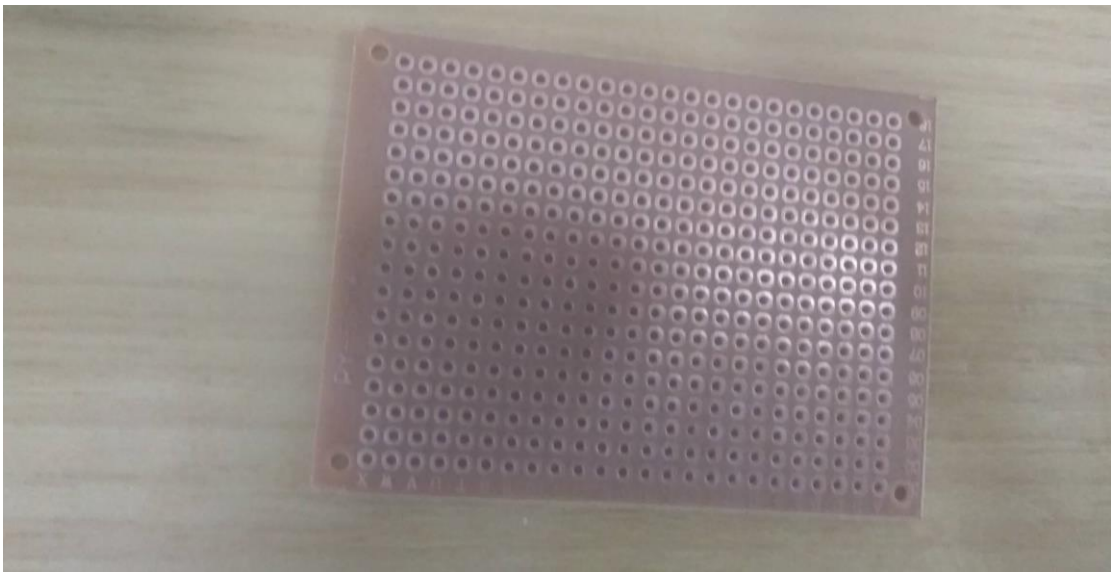


Εικόνα 2-3 GSM shield sim800l

Εικόνα 2-4 Ρυθμιστής φόρτισης

Εικόνα 2-5 Ηλιακό πάνελ

Εικόνα 2-6 Μπαταρία



Εικόνα 2-7 Πλακέτα PCB

Εικόνα 2-8 Κουτί κατασκευής IP56

Εικόνα 2-9 Ηλιακό πάνελ, κουτί, μπαταρία

1.6 Λογισμικό – κώδικας

Πριν προχωρήσουμε στην ανάλυση και την περιγραφή του λογισμικού που θα τρέχει ο ελεγκτής μας θα κάνουμε μια μικρή αναφορά στο εργαλείο (IDE) με το οποίο θα γράψουμε το πρόγραμμα και θα το μεταφορτώσουμε στον ελεγκτή μας. Θα χρησιμοποιήσουμε τον επίσημο IDE του Arduino (1.6.1)

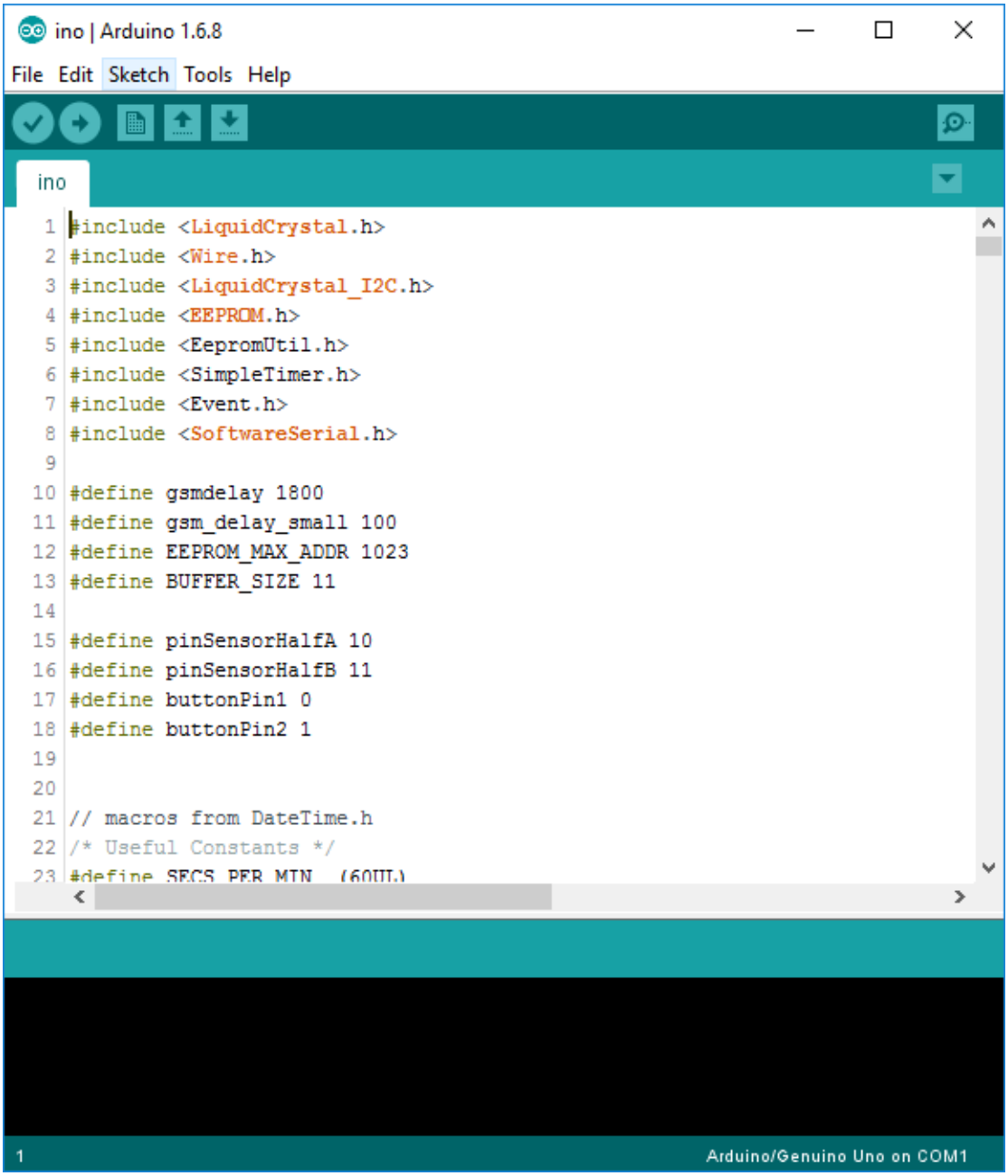
1.6.1 Arduino IDE

Το περιβάλλον ανάπτυξης λογισμικού (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε JAVA. Διατίθεται για τις πλατφόρμες Windows, Mac και Linux. Μας βοηθά στη συγγραφή των προγραμμάτων (σκίτσα ή στα αγγλικά sketch, στην ορολογία του Arduino). Διαθέτει αρκετά έτοιμα παραδείγματα και μερικές έτοιμες βιβλιοθήκες για προέκταση της γλώσσας και για τον εύκολο χειρισμό των εξαρτημάτων που είναι συνδεδεμένα με το Arduino. Διαθέτει ένα serial monitor με το οποίο στέλνουμε αλφαριθμητικά στην οθόνη του υπολογιστή. Αυτό είναι πολύ χρήσιμο στην επιδιόρθωση σφαλμάτων (debugging).Επίσης, είναι σε θέση να φορτώνει (Upload) το πρόγραμμά μας στο Arduino με ένα μόνο κλικ.

Τα σκίτσα αυτά γράφονται στον επεξεργαστή κειμένου που παρέχει το Arduino IDE και αποθηκεύονται με την επέκταση αρχείου «.ino». Επίσης, το IDE του Arduino μας παρέχει τα χαρακτηριστικά γνωρίσματα της αντιγραφής/επικόλλησης και της αναζήτησης/αντικατάστασης κειμένου. Στην περιοχή μηνύματος εμφανίζονται μηνύματα που αφορούν την διαχείριση του Arduino, όπως μηνύματα επιβεβαίωσης μετά από αποθήκευση ή φόρτωση του προγράμματος. Ακόμα, η κονσόλα απεικονίζει μηνύματα κειμένου από το περιβάλλον του Arduino, συμπεριλαμβανομένων και των μηνυμάτων λάθους που υπάρχουν στο πρόγραμμα μετά την μεταγλώττιση. Στην κάτω δεξιά γωνία του παραθύρου εμφανίζεται το όνομα της πλατφόρμας που έχουμε συνδεδεμένη στον υπολογιστή μας (Arduino UNO στην δική μας περίπτωση) και η θύρα που είναι συνδεδεμένη (πχ.COM3).

Τα κουμπιά που υπάρχουν στη γραμμή εργαλείων του Arduino IDE μας επιτρέπουν να δημιουργήσουμε, να αποθηκεύσουμε και να ανοίξουμε σκίτσα, να ε-

λέγξουμε και να ανεβάσουμε προγράμματα στο arduino, καθώς και να ανοίξουμε το παράθυρο της σειριακής οθόνης (serial monitor). [2]



Εικόνα 2-10 Arduino IDE

1.6.2 Εντολές AT

Οι εντολές αυτές στέλνονται από έναν ελεγκτή ή ένα πληκτρολόγιο στο modem ενός κινητού τηλεφώνου, ή στη δική μας περίπτωση στο chip SIM800 που θεωρείται το modem. Χρησιμοποιούνται για λειτουργίες όπως η κλήση ενός τηλεφωνικού αριθμού και η αποστολή, η ανάγνωση ή η διαγραφή του SMS. Ακόμη, χρησιμοποιούνται για την ανάγνωση ή τη διαγραφή μιας επαφής, την ανάγνωση της έντασης λήψης (σήμα δικτύου) κλπ. [9]

Οι εντολές AT που χρησιμοποιήθηκαν στην εργασία είναι οι εξής:

Store the current configuration in nonvolatile memory. When the board is reset, the configuration changes from the last session are loaded.	ATE0&W AT [Reset the board] AT	OK [No echo] OK [No echo] OK
Set SMS system into text mode, as opposed to PDU mode.	AT+CMGF=1	OK

AT+CLIP Calling Line Identification Presentation	
Test Command **AT+CLIP=?**	Response **+CLIP: (**list of supported **<n>**s**)** **OK**
	Parameters See Write Command

AT+CRC Set Cellular Result Codes for Incoming Call Indication	
Test Command **AT+CRC=?**	Response **+CRC: (**list of supported **<mode>**s**)** **OK**
	Parameters See Write Command
Read Command **AT+CRC?**	Response **+CRC: <mode>** **OK**

AT+CREG Network Registration	
Test Command **AT+CREG=?**	Response **+CREG: (**list of supported **<n>**s**)** **OK**
	Parameters See Write Command

1.6.3 *Ανάλυση – περιγραφή κώδικα*

Βήμα βήμα θα δούμε και θα περιγράψουμε τον κώδικα του ελεγκτή μας. Μεγαλύτερη προσοχή θα δοθεί στα βασικά κομμάτια της λειτουργίας του ελεγκτή, αυτά που θα χαρακτηρίζουν την κατασκευή μας. Στα σημεία που συναντάμε σχεδόν σε όλα τα προγράμματα θα γίνει μια πιο περιληπτική περιγραφή. Θα υπάρχουν λογικά διαγράμματα στις βασικές συναρτήσεις.

1.6.4 *Εισαγωγή βιβλιοθηκών*

```
1 #include <LiquidCrystal.h>
2 #include <Wire.h>
3 #include <LiquidCrystal_I2C.h>
4 #include <EEPROM.h>
5 #include <EepromUtil.h>
6 #include <SimpleTimer.h>
7 #include <Event.h>
8 #include <SoftwareSerial.h>
```

Εικόνα 2-11 Εισαγωγή βιβλιοθηκών

Στην αρχή του προγράμματος όπως φαίνεται στην (Εικόνα 2-11), εισάγουμε κάποιες βιβλιοθήκες που είναι απαραίτητες για τις λειτουργίες του ελεγκτή. Θα τις περιγράψουμε παρακάτω, όταν θα τις χρησιμοποιήσουμε.

1.6.5 *Εντολή #define*

```
10 #define gsmdelay 1800
11 #define gsm_delay_small 100
12 #define EEPROM_MAX_ADDR 1023
13 #define BUFFER_SIZE 11
14
15 #define pinSensorHalfA 10
16 #define pinSensorHalfB 11
17 #define buttonPin1 0
18 #define buttonPin2 1
19
20 // macros from DateTime.h
21 /* Useful Constants */
22 #define SECS_PER_MIN  (60UL)
23 #define SECS_PER_HOUR (3600UL)
24 #define SECS_PER_DAY  (SECS_PER_HOUR * 24L)
25
26 /* Useful Macros for getting elapsed time */
27 #define numberOfSeconds(_time_) (_time_ % SECS_PER_MIN)
28 #define numberOfMinutes(_time_) ((_time_ / SECS_PER_MIN) % SECS_PER_MIN)
29 #define numberOfHours(_time_) (( _time_% SECS_PER_DAY) / SECS_PER_HOUR)
30 #define elapsedDays(_time_) ( _time_ / SECS_PER_DAY)
```

Εικόνα 2-12 Αντικείμενα define

1.6.6 *Αρχικοποίηση βιβλιοθηκών*

```
32 SoftwareSerial gsmSerial(8, 9); //rx-4 tx-5
33 LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
34
35 SimpleTimer timer;
36 EepromUtil eepromUtil;
37
```

Εικόνα 2-13 Αρχικοποίηση βιβλιοθηκών

Οι παραπάνω εντολές (Εικόνα 2-13) είναι απαραίτητες για να αρχικοποιήσουμε κάποιες τιμές ή να δηλώσουμε κάποιους ακροδέκτες για την ορθή λειτουργία των βιβλιοθηκών που έχουμε εισάγει. Στη γραμμή (32) η συνάρτηση SoftwareSerial δημιουργεί έναν δίαυλο επικοινωνίας rx-tx στους ακροδέκτες 8 και 9 του arduino. Η εντολή LiquidCrystal λέει στον arduino σε ποιους ακροδέκτες είναι συνδεμένη η οθόνη του ελεγκτή (23). Οι εντολές στις γραμμές 35-36 δημιουργούν δυο μεταβλητές, την timer και την eepromUtil ώστε να μπορέσουμε να χειριστούμε τις δυο αντίστοιχες βιβλιοθήκες. Στη συνέχεια του κώδικα θα δούμε πώς γίνεται αυτό.

1.6.7 *Μεταβλητές του προγράμματος (καθολικές - global)*

```
38 int stopTimer=0;
39 int setStepTimer = 0;
40 int SMS_location_number;
41 int adminNumberSize;
42 int counder2Rpm=2;
43 int smsMetra=0;
44 int varButton1=0, oldVarButton1=0;
45 int varButton2=0, oldVarButton2=0;
46 int meter=0;
47 int signalRequestTime = 30000;
48 int gsmInitSize;
49 int gsmSignalLevel;
50 int rpm=-1;
51 int countCloseRpm=0;
52 unsigned int input_pos = 0;
53 unsigned int stepMinMax = 20; // pososto %
54 unsigned long elapsedTimeAll=0;
55 unsigned long stepPerHour=0;
56 unsigned long trueStepValue=0;
57 float h=0,m=0,s=0;
58 float stepValue = 0;
59 const char* adminNumbers[] = {"6937043744"};
60 const char* gsmInit[] = {"ATE0", "AT+CMGF=1", "AT+CLIP=1", "AT+CRC=1", "AT+CREG=1"};
61
62 boolean isSMS = false;
63 boolean isSendingSms = false;
64 boolean isDialing = false;
65 boolean isUnder30Min = false;
66 boolean isMinMaxCalc = false;
67 boolean isLockStepManual = false;
68 boolean isOnTimerMinMax = false;
69 boolean isSignalOk = false;
70 boolean isStop = false;
71 boolean isRunCode = false;
72 boolean isAuthorized = false;
73 byte modeReadButtons =1;
74 String inputString = "";
75 String smsSender;
76 String textMessage;
77 String authorizedNumbers[3];
```

Εικόνα 2-14 Μεταβλητές

Στην εικόνα (Εικόνα 2-14) υπάρχουν οι καθολικές μεταβλητές του προγράμματος μας, δηλαδή οι μεταβλητές που κάθε συνάρτηση μπορεί να έχει πρόσβαση. Η χρησιμότητά τους θα αναφερθεί κατά την χρήση τους παρακάτω. Παρακάτω επίσης θα δούμε ότι υπάρχουν και άλλες μεταβλητές μέσα στις συναρτήσεις οι οποίες είναι τοπικές και μπορεί μόνο η κάθε συνάρτηση να τις προσπελάσει και να τις μεταβάλει.

1.6.8 *Συναρτηση Setup*

```
81 void setup() {
82   Serial.begin(9600);
83   delay(10);
84   gsmSerial.begin(9600);
85   delay(10);
86   lcd.init();                      // initialize the lcd
87   lcd.noBacklight();
88   adminNumberSize = sizeof(adminNumbers) / sizeof(char *);
89   gsmInitSize = sizeof(gsmInit) / sizeof(char *);
90   printLcd(" ----START----",0,0,0);
91   delay(1500);
92   lcd.clear();
93   printLcd("r0",0,0,0);
94   pinMode(buttonPin1, INPUT_PULLUP);
95   pinMode(buttonPin2, INPUT_PULLUP);
96   delay(10);
97   deleteALLSMS();
98   Serial.println("Reading eeprom");
99   readEeprom();
100  delay(300);
101 }
```

Εικόνα 2-15 Συνάρτηση setup

Κάθε πρόγραμμα που γράφεται για τον arduino πρέπει να έχει απαραίτητα δυο βασικές συναρτήσεις, την *setup* και την *loop*. Στην πρώτη γίνονται δηλώσεις αρχικοποίησης, ορίζεται ο τρόπος λειτουργίας των ακροδεκτών (pins) και ενεργοποιούνται κάποιοι μηχανισμοί, όπως το serial monitor. Η συνάρτηση αυτή είναι η πρώτη που εκτελείται από τον επεξεργαστή και εκτελείται μια και μοναδική φορά, εκτός αν υπάρξει κάποιο reset από δική μας επιλογή, μέσα από το κώδικα, ή υπάρξει διακοπή στην παροχή ρεύματος του ελεγκτή. Θα ακολουθήσει μια επιγραμματική περιγραφή των εντολών που βρίσκονται μέσα στην setup ώστε να γίνει κατανοητή ευκολότερα η λειτουργία της.

Πίνακας 2-1 Περιγραφή της συνάρτησης Setup

81	Αρχική εντολή που δηλώνει την συνάρτηση setup
82	Εκκίνηση του Serial monitor
83	Μια μικρή καθυστέρηση (delay), δεν είναι απαραίτητη
84	Εκκίνηση του gsmSerial
85	Delay
86	Προετοιμασία της LCD
87	Απενεργοποίηση του οπίσθιου φωτισμού της LCD
88	Μέγεθος του πίνακα adminNumbers
89	Μέγεθος του πίνακα gsmInit
90	Εμφάνιση μνήματος «start» στην οθόνη
91	Delay
92	Καθαρισμός της οθόνης

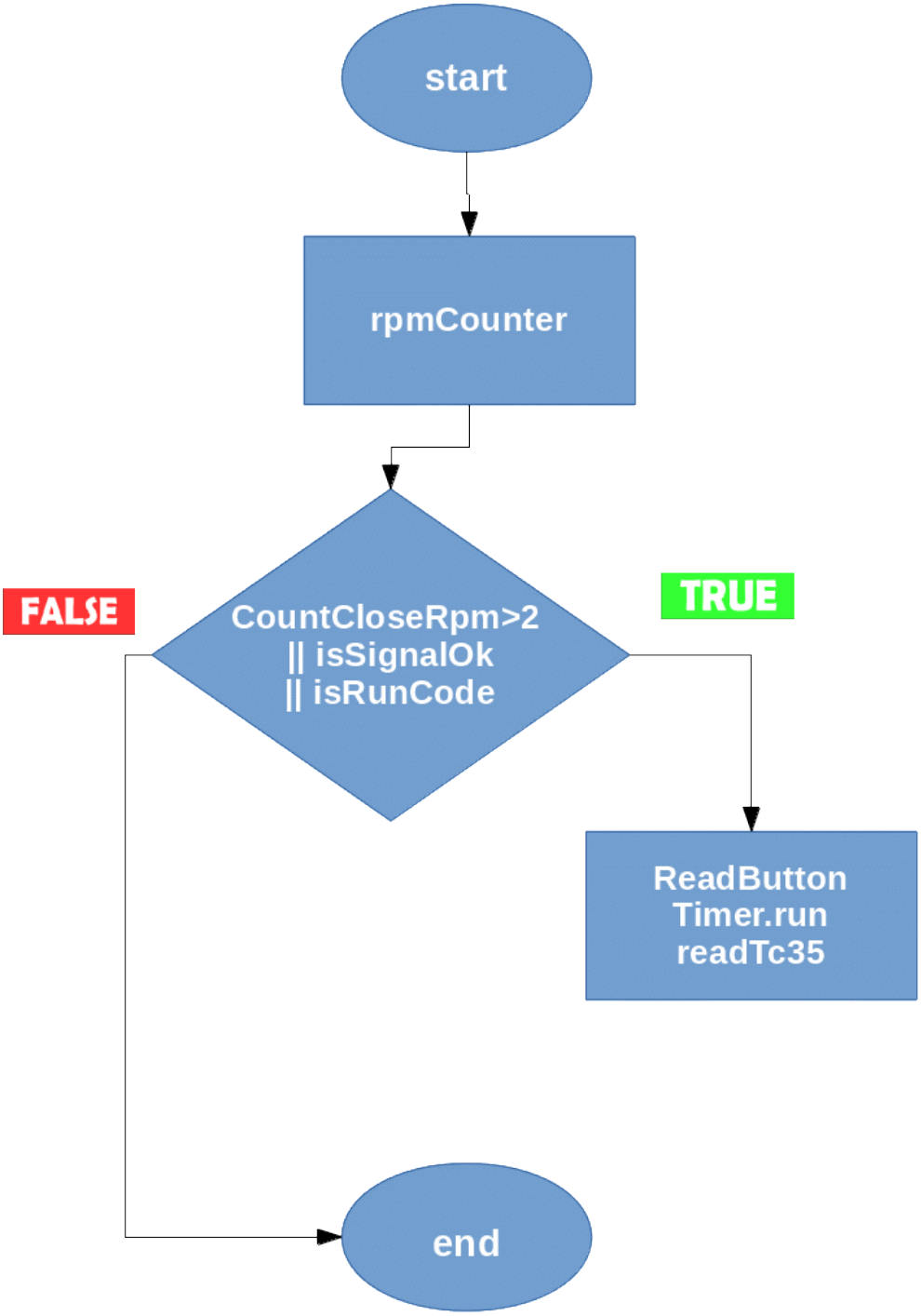
```
93      Εμφανίζεται  στην οθόνη το «r0»
94      Ορισμό του μπουτόν 1 ως είσοδος και ενεργοποίηση της pull-up
95      Ορισμό του μπουτόν 2 ως έξοδος και ενεργοποίηση της pull-up
96      Delay
97      Διεγραφή των sms από τη sim
98      Εμφάνιση μνήματος στο Serial monitor
99      Διάβασμα της eeprom
100     Delay
```

1.6.9 Συνάρτηση loop

```
158 void loop()
159   {
160
161     rpmCounter();
162
163     if(countCloseRpm >= 2 || isSignalOk == false || isRunCode == true)
164     {
165     read_Button(modeReadButtons);
166     timer.run();
167     readTC35();
168     }
169   }
```

Εικόνα 2-16 Συνάρτηση loop

Αρχικά,όπως φαίνεται και στην γραμμή (161) της παραπάνω εικόνας, καλούμε την συνάρτηση rpmCounder που θα εξηγηθεί σε παρακάτω κεφάλαιο. Ο παρακάτω έλεγχος γίνεται για να απομονώσουμε τις υπόλοιπες λειτουργίες του ελεγκτή, έτσι ώστε στην αρχή που το ποτιστικό μας καρούλι θα «ανοίγει», να μην απασχολείται ο επεξεργαστής μας από άλλα πράγματα πάρα μόνο με την μέτρηση των στροφών. Άλλωστε οι υπόλοιπες λειτουργίες είναι χρήσιμες αφού το καρούλι θα αρχίζει την διαδικασία του ποτίσματος (1.3.2). Όταν λοιπόν το καρούλι ξεκινήσει το πότισμα, τίθενται σε λειτουργία και οι υπόλοιπες λειτουργίες του ελεγκτή. Αυτό το πετυχαίνουμε καλώντας τις τρεις συναρτήσεις που βλέπουμε στις γραμμές 165-167. παρακάτω υπάρχει και το λογικό διάγραμμά της συνάρτησης.



Εικόνα 2-17 Λ.Δ loop

1.6.10 Συνάρτηση readEeprom

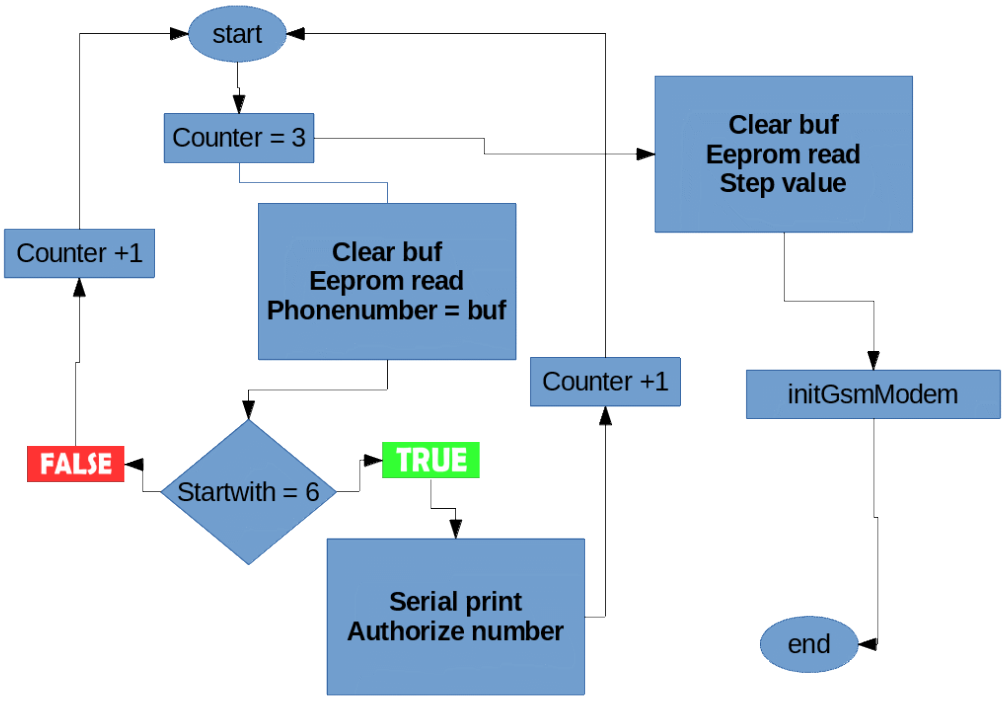
```
103 void readEeprom()
104 {
105  char buf[BUFFER_SIZE];
106 
107  for(int i=0; i<3; i++){
108      // erase buffer
109      memset(buf, 0, sizeof(buf));
110 
111      eepromUtil.eeprom_read_string(i*BUFFER_SIZE, buf, BUFFER_SIZE);
112 
113      String phoneNumber = (String)buf;
114 
115      if(phoneNumber.startsWith("6")){
116          Serial.println("\nAdding Admin Number: ");
117          Serial.print(phoneNumber);
118          Serial.print("\n");
119          authorizedNumbers[i] = phoneNumber;
120      }
121  }
122 
123  //erease buffer
124  memset(buf, 0, sizeof(buf));
125  //read step
126  eepromUtil.eeprom_read_string(100, buf, BUFFER_SIZE);
127 
128  stepValue = atof(buf);
129  String stepDataString = (String)buf;
130  Serial.println("\nReading step: ");
131  Serial.print(stepValue);
132  Serial.println("\nInitializing GSM Modem");
133  timer.setTimer(gsmdelay, initGsmModem, gsmInitSize + 1);
134 
135 }
```

Εικόνα 2-18 Συναρτηση readEeprom

Η συνάρτηση αυτή καλείται μέσα από το setup(1.6.8) άρα θα τρέξει μια φορά στην αρχή. Διαβάζουμε κάποιους αριθμούς κινητών τους οποίους έχουμε αποθηκεύσει (θα δούμε παρακάτω πώς), οι οποίοι είναι η αριθμοί που θέλουμε να αναγνωρίζει ο ελεγκτής και να δίνει πληροφορίες. Αυτό γίνεται ώστε να μη μπορεί ο οποιοσδήποτε γνωρίζει το αριθμό κλήσης του ελεγκτή να παίρνει πληροφορίες. Όπως επίσης δεν θα ανταποκρίνεται σε κλήσεις διάφορων εταιρειών που πιθανόν να καλέσουν.

Δηλώνουμε μια τοπική μεταβλητή χαρακτήρων (Char), δηλαδή έναν πίνακα μεταβλητών. Με την βοήθεια του πίνακα αυτού θα διαβάσουμε 3 θέσεις στην μνήμη eeprom, θα διαβάζουμε 10 bits την κάθε φορά. Με τον βρόγχο for που εκτελούμε τρεις φορές και τη βιβλιοθήκη eeprom διαβάζουμε τους αριθμούς που είναι αποθηκευμένοι στις θέσεις μνήμης της eeprom. Οι θέσεις μνήμης που είναι γραμμένοι οι αριθμοί είναι οι εξής 0-10, 11-21, 22-33. Στη συνέχεια κάνουμε έναν έλεγχο αν ο αριθμός ξεκινάει από 6 που αυτό είναι το επιθυμητό και τους αποθηκεύουμε σε έναν πίνακα (authorizedNumbers).

Τρέχουμε άλλη μια φορά την εντολή eeprom.read αυτή την φορά διαβάζουμε την μνήμη 100 στην οποία έχουμε αποθηκεύσει μια τιμή την οποία έχουμε μετρήσει και μας δίνει την απόσταση που διανύει το καρούλι κάνοντας μια στροφή ο άξονας της τουρμπίνας που δίνει την κίνηση. Με την σειρά της την αποθηκεύουμε σε μια μεταβλητή (stepvalue). Την εκτυπώνουμε στην σειριακή θύρα του arduino. Στο τέλος καλούμε έναν timer ο όποιος στέλνει μια σειρά από εντολές ("ATE0", "AT+CMGF=1", "AT+CLIP=1", "AT+CRC=1", "AT+CREG=1") στο sim800 ώστε να ξεκινήσει η λειτουργία του



Εικόνα 2-19 Λ.Δ eeprom

1.6.11 Συνάρτηση initGsmModem

```
137 void initGsmModem() {
138   static int gsmInitStep = 0;
139
140   if (gsmInitStep < gsmInitSize) {
141       Serial.println("\nSending command:");
142       Serial.print(gsmInit[gsmInitStep]);
143       gsmSerial.println(gsmInit[gsmInitStep]);
144       gsmInitStep++;
145     }
146   else {
147       Serial.println("\nGSM Modem Initialization Completed");
148      timer.setInterval(signalRequestTime, checkGSMSignal);
149   }
150
151 }
```

Εικόνα 2-20 initGsmModem

Με αυτή τη συνάρτηση στέλνουμε ένα πλήθος από εντολές AT στο modem της κατασκευής μας (1.4.6 sim800). Αυτές είναι απαραίτητες για να ξεκινήσει η λειτουργία του modem, να ορίσουμε σε τι δίκτυο θα εγγραφεί, σε ποια γεωγραφική περιοχή βρισκόμαστε, όπως και κάποιες άλλες λειτουργίες του sim800l. Τις εντολές αυτές τις έχουμε δηλώσει στην περιοχή που έχουμε δηλώσει όλες μας τις μεταβλητές στη θέση gsmInit (Εικόνα 2-20 initGsmModem)

Συνάρτηση checkGSMSignal

```
153 void checkGSMSignal() {
154   if (!isSendingSms && !isDialing)
155     gsmSerial.println("AT+CSQ");
156 }
```

Εικόνα 2-21 checkGsmSignal

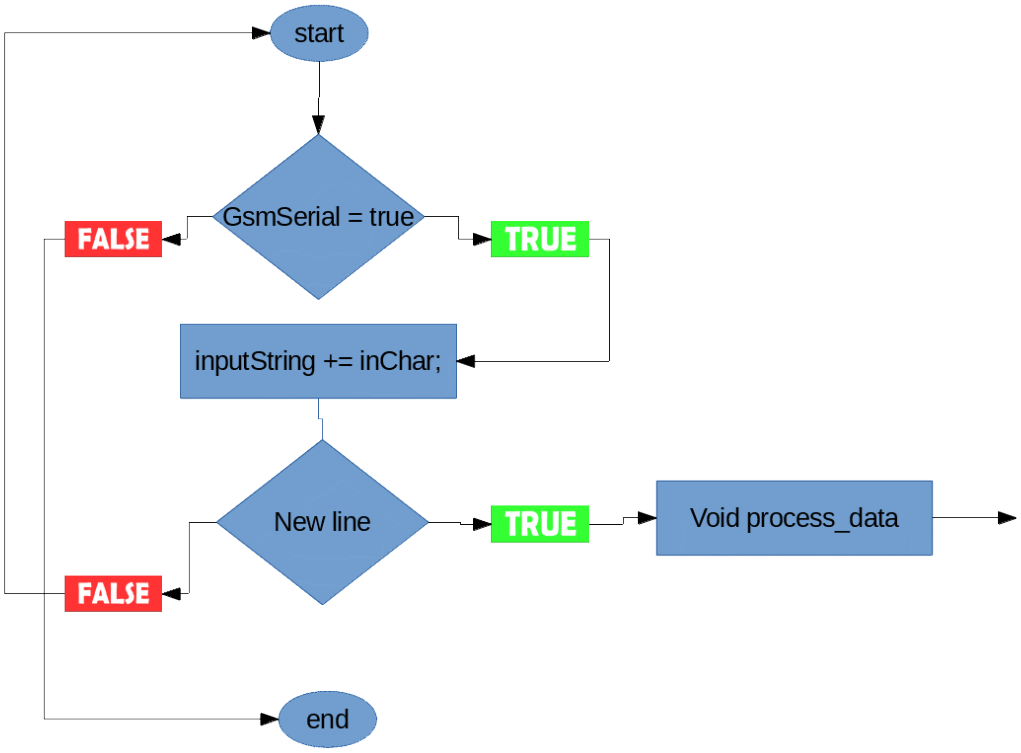
Στέλνουμε την εντολή AT+CSQ (γραμμή 155) και μας επιστρέφει την ποιότητα του σήματος.

1.6.12 Συνάρτηση readSim800

```
171 void readSim800() {
172     while (gsmSerial.available()) {
173     // get the new byte:
174     char inChar = (char)gsmSerial.read();
175     // add it to the inputString:
176     inputString += inChar;
177     // if the incoming character is a newline, set a flag
178     // so the main loop can do something about it:
179     if (inChar == '\n') {
180       inputString.toUpperCase();
181       process_data (inputString);
182       inputString = "";
183     }
184   }
185 }
```

Εικόνα 2-22 readSim800

Οι χαρακτήρες που θα έρθουν στο gsmSerial θα αποθηκευτούν σε ένα string και θα σταλθούνε στην συνάρτηση process_data η οποία θα τα επεξεργαστεί και θα πάρει ανάλογες αποφάσεις.



Εικόνα 2-23 Λ.Δ readSim800

1.6.13 Συνάρτηση ProcessData

```
void FN_processData (String data) {
  Serial.println (data);
  if (data.indexOf("+CSQ:") >= 0) {
    FN_parseSignal(data);
  }
  else if (data.indexOf("+CMTI:") >= 0) {
    SMS_location_number = data.substring(data.indexOf(",")+1).toInt();
    gsmSerial.print("AT+CMGR=");
    gsmSerial.println(SMS_location_number);
  }
  else if (data.indexOf("+CLIP:") >= 0) {
    int commaIndex = data.indexOf(",");
    String smsNumber = data.substring(data.indexOf(":")+3,commaIndex-1);

    Serial.println("Calling Number: ");
    Serial.println(smsNumber);
    isAuthorized = false;

    for (int i=0; i<3; i++) {
      if(smsNumber.indexOf(authorizedNumbers[i])>=0){
          isAuthorized = true;
          smsSender = String(authorizedNumbers[i]);
          break;
      }
    }

    //hang up
    timer.setTimeout(gsmDelay,FN_hangUp);


   if (isAuthorized)
   {
      timer.setTimeout(gsmDelay*2,FN_smsInfoCall);
   }
  }

  else if (data.indexOf("+CMGR:") >= 0) {
    int commaIndex = data.indexOf(",");
    int secondCommaIndex = data.indexOf(",", commaIndex+1);
    String smsNumber = data.substring(commaIndex+2,secondCommaIndex-1);
    isAuthorized = false;
```

2-24 ProcessData

```
253
254       for (int i=0; i<adminNumberSize; i++) {
255         if(smsNumber.indexOf(adminNumbers[i])>=0){
256             isAuthorized = true;
257             smsSender = String(adminNumbers[i]);
258             break;
259         }
260       }
261
262       if(!isAuthorized){
263           //parse notificationNumbers array
264           for (int i=0; i<3; i++) {
265             if(smsNumber.indexOf(authorizedNumbers[i])>=0){
266                 isAuthorized = true;
267                 smsSender = String(authorizedNumbers[i]);
268                 break;
269             }
270           }
271
272       }
273
274       Serial.println ("authorized ?");
275       Serial.println (isAuthorized);
276       isSMS = isAuthorized;
277     }
278     else if(isSMS) {
279       FN_parseSms(data);
280       isSMS = false;
281       //delete last SMS
282       //timer.setTimeout(gsmDelay,deleteLastSMS);
283     }
284   }
```

Σε αυτήν την συνάρτηση στέλνουμε και λαμβάνουμε δεδομένα από το gsm (sim800l). Τα δεδομένα που στέλνει το gsm στον arduino τα επεξεργαζόμαστε και ανάλογα με το μήνυμα που δέχεται ο arduino εκτελεί μια διεργασία ή αγνοεί το μήνυμα. Ας δούμε πιο αναλυτικά πώς γίνεται αυτό. Ξεκινώντας από την γραμμή 213 που απλά εμφανίζουμε τα δεδομένα στην σειριακή πόρτα για να τα δούμε στην οθόνη του υπολογιστή ώστε να μπορέσουμε να κάνουμε την ανάλογη επεξεργασία. Ένας άλλος σημαντικός λόγος που θέλουμε να βλέπουμε τα δεδομένα στην οθόνη είναι η αποσφαλμάτωση του κώδικα. Στην γραμμή 214 ελέγχουμε αν το gsm modem στείλει το μήνυμα "+CSQ:" που θα μας ενημερώνει για την ποιότητα του σήματος.

Στην γραμμή 217 περιμένουμε να διαβάσουμε το μήνυμα "+CMTI" το οποίο σημαίνει πως κάποιο μήνυμα κειμένου (sms) έχει έρθει και με την εντολή "AT+CMGR" ζητάμε από το gsm να μας εμφανίσει το μήνυμα.

Με τον ίδιο τρόπο στην γραμμή 222 ελέγχουμε αν υπάρχει κάποια εισερχόμενη κλήση η οποία γίνεται αντιληπτή από το μήνυμα "+CLIP". Με την εντολή στην γραμμή 224 κρατάμε τον αριθμό του καλούντος σε μια μεταβλητή

(smsNumber). Την εμφανίζουμε στην οθόνη του υπολογιστή, εμφανίζουμε δηλαδή το νούμερο του τηλέφωνου που κάλεσε στο gsm modem. Στην γραμμή 230 τρέχουμε έναν βρόχο για να ελέγξουμε αν ο αριθμός που κάλεσε είναι ένας από τους 3 αριθμούς που έχουμε αποθηκεύσει στην eeprom και μόνο σε αυτούς δίνει απάντηση ο ελεγκτής.

Στην γραμμή 239 μετά από χρόνο gsmDelay καλούμε την συνάρτηση (FN_hangUp) η οποία απορρίπτει την κλήση που δέχεται το gsm modem.

Παρακάτω με μια εντολή if (242) ελέγχει αν ο αριθμός είναι ένας από τους 3 έγκυρους και στη συνέχεια μετά από χρόνο gsmDelay καλεί την συνάρτηση FN_smsInfoCall με την οποία στέλνεται ένα sms με τις πληροφορίες για το καρούλι (μέτρα για να φτάσει, ταχύτητα μέτρα/ώρα, χρόνος για να ολοκληρώσει το πότισμα).

Τέλος, στη γραμμή 248 διαβάζουμε το μήνυμα κειμένου (απάντηση από την γραμμή 219) και αποθηκεύουμε τον αριθμό του αποστολέα.

1.6.14 Συνάρτηση parseSms

```
void FN_parseSms(String data) {
  if (data.startsWith("PROGNUMBER")) {
    //programming
    /*boolean isSmsValid = FN_checkSms(data, true, true);

    if(!isSmsValid) {

      return;
    }*/

    Serial.println ("programming....");
    int commaIndex = data.indexOf("#");
    int secondCommaIndex = data.indexOf("#", commaIndex+1);
    int endCommaIndex = data.indexOf("#",secondCommaIndex+1);
    String cellNumber = data.substring(secondCommaIndex+1,endCommaIndex);
    cellNumber.trim();
    int memPos=data.substring(commaIndex+1,secondCommaIndex).toInt();

    char buf[BUFFER_SIZE];
    cellNumber.toCharArray(buf,BUFFER_SIZE);
    eepromUtil.eeprom_write_string((memPos-1)*BUFFER_SIZE,buf);
    delay(1000);
    authorizedNumbers[memPos-1] = cellNumber;
    //String message = "Programmed Auth Number: "+String(cellNumber)+" in position
    //FN_smsInfo(message,true);
    FN_printLcd("prog ok",0,0,0);

  }
   else if(data.startsWith("PROGSTEP")) //progstep
  {
    /*boolean isSmsValid = FN_checkSms(data, false, false);

    if(!isSmsValid)
      return;*/

    //programming step
    int commaIndex = data.indexOf("#");
    String stepDataString = data.substring(commaIndex+1);

    char buf[stepDataString.length()];
    stepDataString.toCharArray(buf,stepDataString.length());
    stepValue = atof(buf);

    eepromUtil.eeprom_write_string(100,buf);
    delay(100);
    Serial.println ("Programmed step: ");
    Serial.print (stepValue);
    //String message = "Programmed step: "+String(stepDataString);
    //FN_smsInfo(message,true);
    FN_printLcd("prog step ok",0,0,0);
  }
  else if(data.startsWith("MTR"))
  {

    int commaIndex = data.indexOf("#");
    int secondCommaIndex = data.indexOf("#",commaIndex+1);
    smsMetra = data.substring(commaIndex+1, secondCommaIndex).toInt();
    rpm = smsMetra / stepValue;
    FN_printLcd("&",rpm,0,0);
    Serial.println(smsMetra);
    Serial.println(rpm);
  }
}
```

2-25 Parse Sms

Από το κινητό μας στο gsm modem μπορούμε να στείλουμε 3 εντολές τις οποίες αυτή η συνάρτηση θα διαβάσει και θα αποθηκεύσει τα δεδομένα τους. Η αποθήκευση γίνεται είτε σε μεταβλητές του ελεγκτή, είτε στη μνήμη eeprom που διαθέτει ο arduino.

Οι εντολές που μπορούμε να στείλουμε στο gsm είναι οι εξής:

- (287) PROGNUMBER#X#693*****20#
- (314) PROGSTEP#Y.YY
- (337) MTR#ZZ#

Οι δυο πρώτες εντολές (287, 314) είναι κρίσιμης σημασίας και χωρίς αυτές δεν μπορεί να λειτουργήσει ο ελεγκτής του ποτιστικού καρουλιού. Αυτές οι εντολές δίνονται μόνο την πρώτη φορά που τοποθετείται ο ελεγκτής. Αυτό συμβαίνει γιατί τα δεδομένα αποθηκεύονται στην eeprom του arduino και μένουν μόνιμα εκεί. Αν χρειαστεί να αλλάξουμε κάποιο από αυτά τα δεδομένα ξαναστέλνουμε την εντολή με τα καινούρια. Η τρίτη εντολή (337) είναι περισσότερο για να μπορεί να γίνουν ευκολότερα κάποιες δοκιμές.

Ας εξηγήσουμε τώρα πώς διαβάζονται από το gsm οι παρακάτω εντολές και πώς τις χειριζόμαστε με τον arduino. Για να μπορέσει η εντολή να γίνει κατανοητή από τον arduino πρέπει η σύνταξή της να είναι συγκεκριμένη.

Στην πρώτη εντολή (287) PROGNUMBER#X#693*****20# με την λέξη "PROGNUMBER" λέμε ότι πρόκειται να αποθηκεύσουμε έναν από τους 3 αριθμούς στους οποίους θα δίνει απάντηση με sms ο ελεγκτής. Στη θέση του X δίνουμε την θέση (1,2,3) μνήμης της eeprom που θα αποθηκεύσει τον αριθμό. Εδώ πρέπει να πούμε ότι μια θέση αποτελείται από 11 bytes μνήμης eeprom. Στην θέση 693*****20 γράφουμε τον αριθμό του τηλέφωνου που θέλουμε να αναγνωρίζει ο arduino.

Στη δεύτερη εντολή (314) PROGSTEP#Y.YY και μετά την λέξη PROGSTEP δίνουμε την απόσταση (Y.YY) του λάστιχου σε μέτρα που απλώνει η μαζεύει το καρούλι μας σε μια στροφή. Η μέτρηση της στροφής γίνεται πάνω στην τουρμπίνα (Εικόνα 3-3) που δίνει κίνηση στο καρούλι. Στα καρούλια που έγινε η τοποθέτηση του ελεγκτή η μέτρηση αυτή ήταν από 0,10 έως 0,13 μέτρα.

Στην τρίτη εντολή, η οποία όπως ανέφερα υπάρχει για λόγους δοκιμών του ελεγκτή, μετά την λέξη MTR δίνουμε μια απόσταση σε μέτρα (ZZ) και ο ελεγκτής την μετατρέπει σε στροφές.

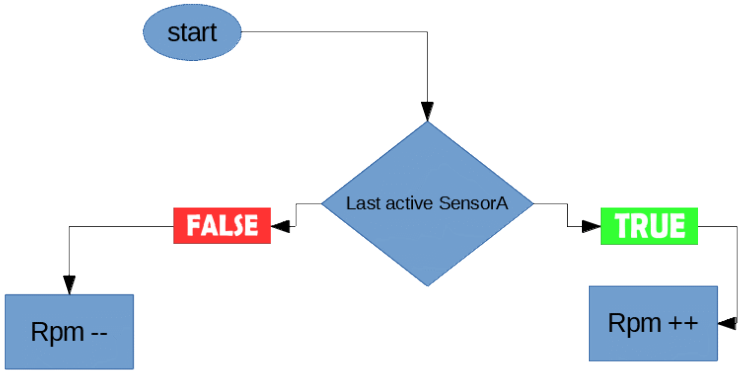
1.6.15 Συνάρτηση RpmCounter

```
void FN_rpmCounter(){

  static boolean isClockWise = false;
  static boolean readCounter = false;

    byte sensorHalfA = digitalRead(pinSensorHalfA);
    byte sensorHalfB = digitalRead(pinSensorHalfB);


    if(sensorHalfA == LOW && sensorHalfB == HIGH) //anoigma karoylioy
    {
      isClockWise = true;
      readCounter = true;
    }
    else if(sensorHalfA == HIGH && sensorHalfB == LOW) //kleisimo karoulioy
    {
      isClockWise = false;
      readCounter = true;
    }

      if(sensorHalfA == LOW && sensorHalfB == LOW && readCounter)
      {
        readCounter = false;
        if(isClockWise)
        {
          rpm++;
          FN_printLcd("r",rpm,0,0);


        }

                else if (rpm>0)
                {
                    rpm--;
                    counder2Rpm--; // flag metrhth
                    stopTimer = millis();

                       //mhdenismos voithitikwn flag
                    if(rpm < 5 && countCloseRpm > 15)
                    {
                      FN_resetFlags();
                    }

                    countCloseRpm++; // metrhths strofes kleisimatos
                    isStartAgainRpmCount++;

                    FN_printLcd("r",rpm,0,0);
                    FN_calcTime();

                    if(isStartAgain && isStartAgainRpmCount > 4)
                        {
                            FN_startAgain();
                        }
                }
```

2-26 RpmCounter

Σε αυτήν την συνάρτηση ανιχνεύεται η δεξιόστροφη και αριστερόστροφη φορά, δηλαδή το πότε απλώνει και το πότε μαζεύει το ποτιστικό σύστημα. Ενδιάμεσα υπάρχουν και κάποιες άλλες λειτουργίες στις οποίες θα γίνει αναφορά παρακάτω.

Ανάλογα με τον ποιον αισθητήρα δει τελευταίο ενεργοποιημένο, ανιχνεύει αν η φορά είναι δεξιόστροφη η αριστερόστροφη, δηλαδή αν το ποτιστικό μας σύστημα απλώνει ή μαζεύει. Όταν δει και τους δυο αισθητήρες ενεργοποιημένους, γνωρίζοντας από πριν την φορά, μετράει θετικά ή αρνητικά αντίστοιχα. Ενδιάμεσα υπάρχει ένας χρονικός μετρητής για να ανιχνεύσει ενδεχόμενη διακοπή του ποτίσματος.



Εικόνα 2-27 ΛΔ rpmCounter

Μια από τις βασικότερες λειτουργίες είναι ο έλεγχος της διακοπής της κίνησης του ποτιστικού καρουλιού. Στην γραμμή 442 κάθε φορά που ο ελεγκτής μας μετρήσει έναν παλμό αρνητικής περιστροφής (440) δίνει στην μεταβλητή stopTimer την τιμή millis. Εάν το καρούλι σταματήσει να κινείται, δηλαδή δεν μετρήσουμε άλλη στροφή για τα επόμενα 75 δευτερόλεπτα, εκτελείται η συνάρτηση FN_stopRpm, έτσι ο αγρότης – χρήστης δέχεται μήνυμα στο κινητό του με την χαρακτηριστική λέξη STOP. Στη γραμμή 456 αν το καρούλι είχε σταματήσει και ξεκίνησε πάλι ο χρήστης ενημερώνεται και πάλι με γραπτό μήνυμα.

1.6.16 Συναρτηση stopRpm

```
465 void FN_stopRpm()
466 {
467     smsSender = authorizedNumbers[1];
468 
469       String message =("***Stop***");
470       FN_printLcd("//",0,12,1);
471       Serial.println("stop");
472       //trueStepValue = 0;
473       stepPerHour = 0;
474       h=0;
475       m=0;
476       isStop = true;
477       //isLockStepManual = false;
478       isStartAgain = true;
479       isStartAgainRpmCount = 0;
480 
481       FN_smsInfo(message,true);
482 
483 }
```

2-28 StopRpm

Αυτή η συνάρτηση καλείται όταν περάσουν 75 δευτερόλεπτα (Εικόνα 2-29)και δεν γίνει μέτρηση παλμού περιστροφής. Αυτό σημαίνει ότι το καρούλι έχει σταματήσει και ειδοποιεί τον χρήση με ένα sms (STOP). Μηδενίζει ορισμένες τιμές τις οποίες υπολογίζει πάλι από την αρχή όταν το καρούλι ξεκινήσει πάλι να ποτίζει. Οι τιμές που μηδενίζονται είναι ο χρόνος που απομένει για την ολοκλήρωση του ποτίσματος, ο οποίος χωρίς κίνηση δεν μπορεί να υπολογιστεί. Η άλλη τιμή που μηδενίζει είναι η ταχύτητα κίνησης, γίνεται μηδέν όταν το καρούλι σταματήσει. Οι τιμές αυτές θα υπολογιστούν εκ νέου όταν ξεκινήσει το καρούλι να κινείται πάλι.

```
836 
837       void Fn_Timers()
838   {
839     if((millis()-stopTimer > 75000) && (rpm > 10 && countCloseRpm > 4 && isStop == false)),
840       {
841         Serial.println("Stop enable");
842         FN_stopRpm();
843       }
844 
845   }
```

Εικόνα 2-29 stopTimer

1.6.17 Συνάρτηση startAgain

```
485 void FN_startAgain()
486 {
487     smsSender = authorizedNumbers[1];
488 
489     isStartAgain = false;
490     isStop = false;
491     FN_printLcd("||",0,12,1);
492     Serial.println("startAgain");
493     isStartAgainRpmCount = 0;
494     FN_smsInfoCall();
495 
496 }
```

2-30 StartAgain

Αν υπάρξει κάποιο πρόβλημα και το καρούλι σταματήσει, όπως είδαμε παραπάνω, ο ελεγκτής, μας ενημερώνει. Αντίστοιχα, αν ξεκινήσει και πάλι, μας ειδοποιεί με ένα sms.

```
498 void FN_smsInfoCall()
499 {
500     String message =("info");
501     isStartAgainRpmCount = 0;
502     FN_smsInfo(message,false);
503 }
504
505
506
507 void sendSms(String messageText) {
508   isSendingSms = true;
509   textMessage = messageText;
510   gsmSerial.println("AT+CMGF=1\r");
511   timer.setTimeout(gsmDelay,sendSmsAtCommand);
512 }
513
514 void sendSmsAtCommand() {
515     gsmSerial.println("AT+CMGS=\""+ smsSender +"\",145\r");
516     timer.setTimeout(gsmDelay,sendSmsMessage);
517 }
518
519 void sendSmsMessage() {
520     gsmSerial.print(textMessage);
521     gsmSerial.print("\x1A");
522     isSendingSms = false;
523 }
526  void FN_smsInfo(String txtInfo, boolean onlyText)
527 {
528     if(onlyText)
529     {
530      sendSms(txtInfo);
531     }
532     else
533     {
534       String message = "m";
535       message += meter;
536       message += "\ns";
537       message += stepPerHour;
538       message += "/";
539       message += trueStepValue;
540       message += "\n";
541       message += int(h);
542       if(m <10)
543           {
544             message+=":0";
545           }
546       else
547           {
548             message += ":";
549           }
550       message += int(m);
551       Serial.print(message);
552       sendSms(message);
553     }
554 }
```

2-31 SmsInfoCall

Οι 5 παραπάνω συναρτήσεις αναλαβαίνουν την αποστολή των μηνυμάτων κειμένου στο χρήστη. Η Fn_smsInfoCall (498) προσθέτει στο sms την λέξη “info” (500) και μηδενίζει έναν μετρητή (501). Η δουλειά του μετρητή είναι να εμποδίζει την ταυτόχρονη αποστολή 2 μηνυμάτων, ώστε να αποφύγουμε τα σφάλματα.

Στην επομένη γραμμή (502) καλούμε την συνάρτηση Fn_smsInfo. Η συνάρτηση αυτή συντάσσει το κείμενο του γραπτού μηνύματος που θα σταλθεί στο χρήστη. Υπάρχουν ορισμένες συμβολοσειρές και ορισμένες τιμές μεταβλητών. Οι λέξεις μας βοηθούν στην κατανόηση των μεταβλητών που ακολουθούν. Οι μεταβλητές είναι οι ενδείξεις που μας ενδιαφέρουν και είναι οι έξης : (535,537,541)

- Μέτρα που απομένουν για την ολοκλήρωση
- Ταχύτητα με την οποία κινείται (βήμα) σε μέτρα ανά ώρα
- Χρόνος που υπολείπεται για να ολοκληρωθεί το πότισμα

Στην γραμμή 552 καλούμε την συνάρτηση sendSms. Η βασική εντολή (AT) αυτής της συνάρτησης (510) ενημερώνει το gsm modem ότι πρόκειται να στείλει ένα sms. Στην επομένη γραμμή (511) υπάρχει ένας χρονικός μετρητής ο οποίος μετά από χρόνο gsmdelay καλεί την συνάρτηση sendSmsAtCommand. Ο χρονικός μετρητής όπως και στην επομένη συνάρτηση (516) έχει τοποθετηθεί για την σωστότερη απόκριση του gsm modem. Χωρίς τους χρονικούς μετρητές υπήρχαν κάποια σφάλματα επικοινωνίας μεταξύ του arduino και του sim800l.

Στην γραμμή 515 λέμε στο gsm modem τον αριθμό του παραλήπτη. Τέλος με την συνάρτηση sendSmsMessage στέλνουμε τον χαρακτήρα “x1A” στο sum800l ώστε να σταλεί το sms στο χρήστη (521).

1.6.18 Συνάρτηση printLcd

```
557 void FN_printLcd(String LcdText,unsigned long lcdVal, int column, int line)
558
559 {
560   lcd.setCursor(column,line);
561   lcd.print(LcdText);
562
563
564     if(lcdVal !=0)
565     {
566        lcd.print(lcdVal);
567        if((lcdVal < 10 ) || (lcdVal < 100 && lcdVal > 90));
568        {
569         lcd.print(" ");
570        }
571     }
572
573 }
```

2-32 PrintLcd

Για να εμφανίσουμε κάτι στην οθόνη του ελεγκτή καλούμε την printLcd δίνοντας τις παραμέτρους για την θέση εμφάνισης και τα δεδομένα.

1.6.19 Συνάρτηση calcTime

```
582  void FN_calcTime()
583  {
584    static unsigned long startTimer = 0;
585    static unsigned long rpmTime = 0;
586
587
588    if(counder2Rpm == 1)
589    { startTimer = millis();
590      FN_printLcd("R",rpm,0,0);
591    }
592    else if(counder2Rpm == 0)
593    {
594
595      rpmTime = millis() - startTimer;
596      elapsedTimeAll = rpmTime * rpm; //xronos gia na fta
597      stepPerHour = (3600000 * stepValue) / rpmTime; //st
598      FN_printLcd("v",stepPerHour,9,0);
599
600      counder2Rpm = 2;
601    }
602
603      meter = rpm * stepValue; //metra gia na ftasei (glc
604      FN_printLcd("m",meter, 5, 0); //lcdprint
605
606
607      if(isUnder30Min == false && m < 29 && h == 0
608      && isStartAgainRpmCount > 1 && countCloseRpm > 30)
609      {
610        isUnder30Min = true;
611
612        smsSender = authorizedNumbers[1];
613
614        String message = ("**30 min**");
615        FN_smsInfo(message,true);
616        FN_printLcd("<3",0,12,1);
617        isStartAgainRpmCount = 0;
```

```
617     }
618
619     if(countCloseRpm == 35 && isLockStepManual == false) //
620       {
621         FN_changeTrueStep();
622         FN_printLcd("k",trueStepValue,7,1);
623       }
624
625
626
627       // metatrepei se wres kai lepta toy timer
628       unsigned long elapsedTime1000 = elapsedTimeAll / 1000;
629       h = numberOfHours(elapsedTime1000);
630       m = numberOfMinutes(elapsedTime1000);
631       s = numberOfSeconds(elapsedTime1000);
632
633       FN_printLcd("t",h,0,1);
634       FN_printLcd(":",m,3,1);
635
636
637       if(h<1)
638       {
639         FN_printLcd(" 0",0,1,1);
640       }
641
642        // elegxos +- % step
643        if(isMinMaxCalc && isStartAgainRpmCount > 10)
644        {
645           FN_calcTrueStep();
646        }
647   }
```

2-33 CalcTime

Άλλη μια βασική συνάρτηση του ελεγκτή. Εδώ γίνονται οι περισσότεροι χρονικοί υπολογισμοί που χρειάζονται, ώστε ο χρήστης να είναι ενημερωμένος. Όλοι οι υπολογισμοί στηρίζονται σε έναν hall sensor ο οποίος μετρά στροφές. Ας δούμε λίγο πώς ακριβώς γίνονται αυτοί οι υπολογισμοί. Έχοντας μετρήσει ο ελεγκτής τις θετικές στροφές κατά το άνοιγμα του καρουλιού γνωρίζει και την απόσταση που έχει ανοιχτεί, πολλαπλασιάζοντας τον αριθμό των στροφών με το μήκος της μια στροφής που εμείς έχουμε μετρήσει και του έχουμε ορίσει (εγγραφή eeprom 1.6.14)

Ξεκινώντας το πότισμα το καρούλι σιγά σιγά μαζεύεται, ξεκινά η αντίστροφη μέτρηση των στροφών. Περνώντας ο hall sensor μπροστά από το μαγνήτη ένας μετρητής ξεκινά να μετρά χρόνο, την επομένη φορά που θα περάσει ο αισθητήρας μπροστά από το μαγνήτη σταματά η μέτρηση, έτσι ο ελεγκτής ξέρει τον χρόνο μιας στροφής. Γνωρίζοντας το πλήθος όλων των στροφών μπορεί με ακρίβεια να υπολογίσει το χρόνο που απομένει για να ολοκληρωθεί το πότισμα. Η μέτρηση αυτή γίνεται κάθε δεύτερη στροφή, την πρώτη ξεκινά ένας μετρητής την δεύτερη σταματά και κάνει τους υπολογισμούς.

Ένας ακόμη υπολογισμός που γίνεται σε αυτή την συνάρτηση είναι το βήμα με το όποιο κινείται το καρούλι, δηλαδή ποια είναι η ταχύτητα με την όποια προχωράει, και είναι εκφρασμένη σε μέτρα ανά ώρα. Ο υπολογισμός αυτός εύκολα μπορεί να γίνει, γνωρίζοντας την απόσταση που κάνει σε μια στροφή και μετρώντας το χρόνο που κάνει για μια στροφή.

Η μονάδα μέτρησης του χρόνου στον ελεγκτή μας είναι τα millisecond, όποτε με τις ανάλογες μετατροπές μπορούμε να ξέρουμε τις ώρες και τα λεπτά που απομένουν, έτσι ώστε ο χρήστης εύκολα να μπορεί να δεχτεί την πληροφορία.

Σε αυτήν την συνάρτηση επίσης γίνεται ένας έλεγχος ώστε αν απομένει λιγότερο από μισή ώρα, ενημερώνει το χρήση με ένα sms.

1.6.20 Συνάρτηση calcTrueStep

Σε αυτήν την συνάρτηση ανιχνεύονται οι αλλαγές στην ταχύτητα με την όποια κινείται το ποτιστικό μας σύστημα και αν ξεπεράσουν το ποσοστό του 15% ενημερώνουν τον χρήστη.

```
793     void FN_calcTrueStep()
794     {
795
796         unsigned long divStepMinMax = trueStepValueVariable * (stepMinMax * 0.01);
797         unsigned long divStepMinMaxABS = abs(trueStepValueVariable - stepPerHour);
798
799         if(divStepMinMaxABS > divStepMinMax )
800         {
801           smsSender = authorizedNumbers[1];
802
803           Serial.println("+-");
804           FN_printLcd("+-",0,12,1);
805           isStartAgainRpmCount = 0;
806           trueStepValueVariable = stepPerHour;
807           FN_smsInfoCall();
808         }
809     }
```

2-34 CalcTrueStep

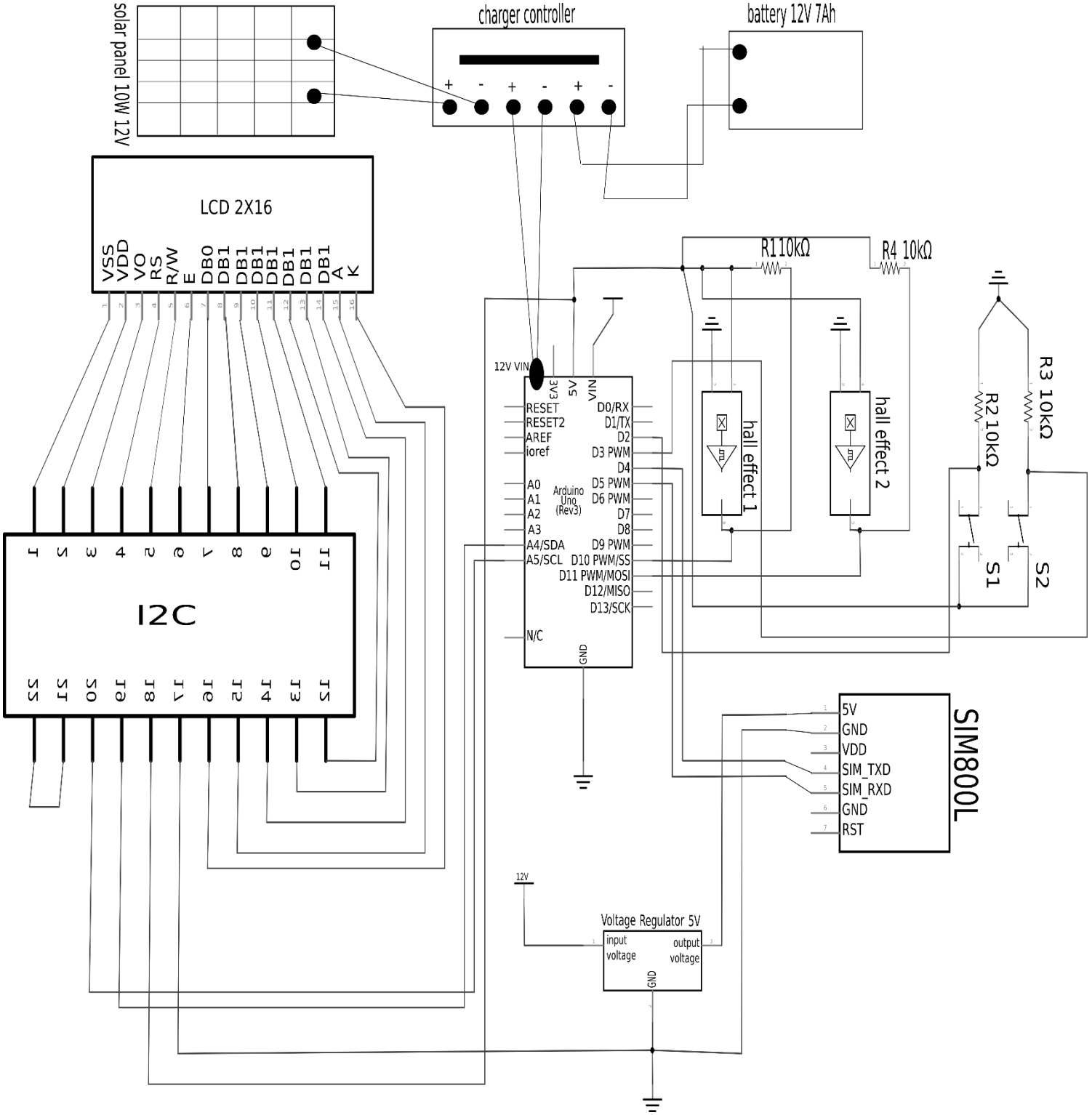
1.6.21 Συνάρτηση resetFlags

Στην συνάρτηση αυτή, που είναι και η τελευταία συνάρτηση ο οποία θα αναλύσουμε, μηδενίζονται όλες οι τιμές των μεταβλητών, όπως στροφές ανοίγματος, στροφές κλεισίματος κάποια βοηθητικά flags και ενημερώνεται ο χρήστης με ένα sms ότι το πότισμα έχει ολοκληρωθεί και έτσι ο ελεγκτής μας είναι έτοιμος για το επόμενο πότισμα.

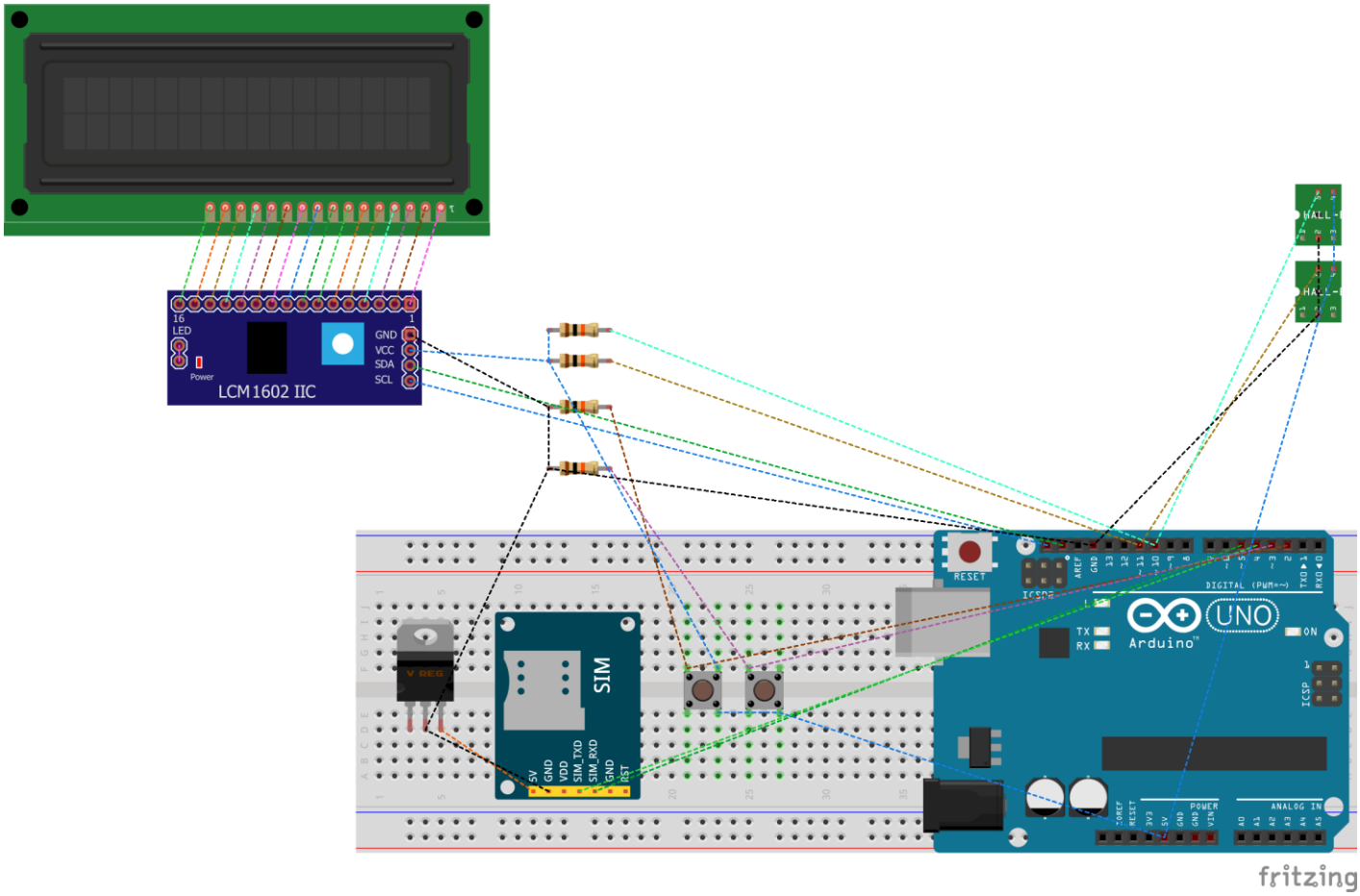
```
812     void FN_resetFlags()
813     {
814         smsSender = authorizedNumbers[1];
815 
816         isUnder30Min = false;
817         countCloseRpm = 0;
818         isStartAgainRpmCount = 0;
819         isMinMaxCalc = false;
820         isLockStepManual = false;
821         trueStepValue = 0;
822         isSignalOk = false;
823         isRunCode = false;
824 
825 
826         FN_smsInfo("-termatismos-",true);
827 
828         FN_deleteAllSms(25);
829 
830         rpm=-1;
831         delay(100);
832         lcd.clear();
833         delay(100);
834         FN_printLcd("RS0",0,0,0);
835     }
```

2-35 ResetFlag

1.6.22 Σχηματικό διάγραμμα



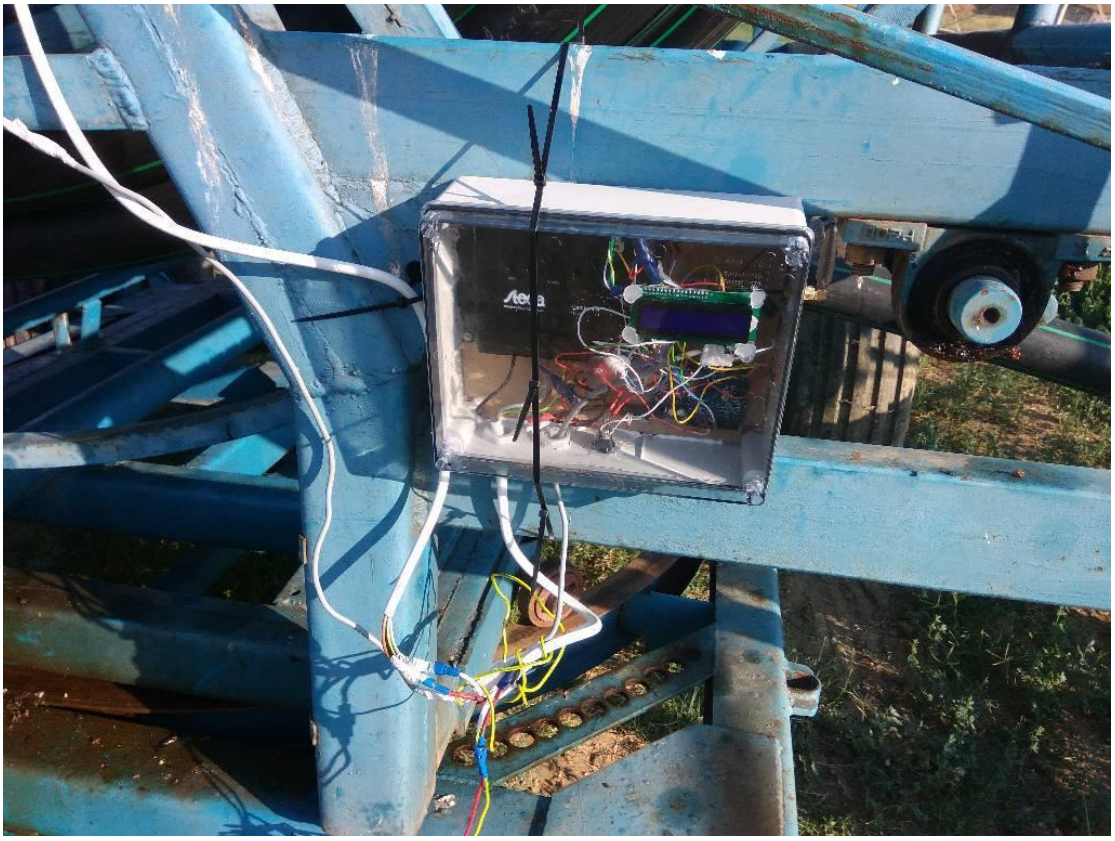
1.6.23 Σχέδιο Fritzing



3. Λειτουργία – Συμπεράσματα

1.7 Τοποθέτηση – εγκατάσταση

Η τοποθέτηση του ελεγκτή πάνω στο ποτιστικό καρούλι δεν είχε κάποια δυσκολία. Το πλαστικό κουτί του ελεγκτή τοποθετήθηκε και στερεώθηκε με την χρήση πλαστικών δεματικών. Το ηλιακό πάνελ στερεώθηκε σε ψηλό σημείο. Ο αισθητήρας στροφών με την βοήθεια δύσκαμπτου χοντρού καλωδίου τοποθετήθηκε αρκετά εύκολα. Ο μαγνήτης κόλλησε πάνω στο μεταλλικό κινητό μέρος από το οποίο γίνεται και η μέτρηση των στροφών. Τέλος η μπαταρία η οποία μπήκε σε ένα στεγανό πλαστικό κουτί και αυτή στερεώθηκε στη μεταλλική βάση του καρουλιού κοντά στο ελεγκτή.



3-1 Κουτί κατασκευής

3-2 Ηλιακό πανελ

3-3 Αισθητηρας στροφων

1.8 Λειτουργία – δοκιμές

Κατά την διάρκεια των δοκιμών δεν παρουσιάστηκαν σημαντικά προβλήματα και η λειτουργία του ελεγκτή ήταν ομαλή, ακριβώς όπως περιμέναμε να είναι. Μικρές βελτιώσεις στο κώδικα του και στην τοποθέτηση βελτίωσαν τη λειτουργία του. Οι χρήστες με ιδιαίτερη ευκολία έμαθαν τη λειτουργία του και κατανόησαν τις απεικονίσεις που γίνονται στην οθόνη του ελεγκτή και στα μηνύματα κειμένου που τους ειδοποιούσαν στο κινητό τους τηλέφωνο.

1.9 Συμπεράσματα

Ο ελεγκτής μας μετά από πολλές δοκιμές κατάφερε να καλύψει στο μέγιστο τους αρχικούς μας στόχους. Η μέτρηση της απόστασης έγινε με μεγάλη ακρίβεια, χωρίς ποτέ να γίνει κάποια λάθος εκτίμηση. Έτσι ο χρήστης γνώριζε ανά πάσα στιγμή την ακριβή θέση του ποτιστικού συστήματος. Η απόσταση, όπως έχω αναφέρει και νωρίτερα, είναι μόνο για πληροφοριακούς λόγους. Οι κρίσιμες πληροφορίες που μας δίνει ο ελεγκτής είναι ο χρόνος που χρειάζεται για να ολοκληρωθεί το πότισμα και η ταχύτητα με την όποια κινείται.

Η ένδειξη της ταχύτητας έδωσε στο χρήστη τη δυνατότητα να ρυθμίζει με ακρίβεια το πότισμα της καλλιέργειάς του και να την κρατάει σταθερή σε κάθε νέο πότισμα. Χωρίς την χρήση του ελεγκτή η ταχύτητα ρυθμίζονταν κατά προσέγγιση. Επίσης μεταβολές στο βήμα της τάξεως 20-30% δεν θα μπορούσαν εύκολα να γίνουν αντιληπτές. Όλα αυτά χρειαζόταν συνεχή παρακολούθηση με την παρουσία του χρήστη στο χωράφι που πραγματοποιείται το πότισμα ανά τακτά χρονικά διαστήματα. Με τη χρήση του ελεγκτή δεν υπήρχε κανένας λόγος να βρίσκεται εκεί παρά μόνο όταν ο ελεγκτής τον ενημέρωνε.

Ο υπολογισμός του χρόνου ήταν η πιο χρήσιμη πληροφορία. Έδωσε στο χρήση την πολυτέλεια να γνωρίζει την ώρα που θα τελειώσει το πότισμα του χωραφιού του. Έτσι, μπορούσε να οργανώσει τις υπόλοιπες εργασίες του, να μην χρειάζεται να πηγαίνει συνέχεια στο χωράφι του ώστε να υπολογίσει το πότε θα ολοκληρωθεί το πότισμα. Ο υπολογισμός του χρόνου ολοκλήρωσης του ποτίσματος εξοικονόμησε πολύ χρόνο και πολλά έξοδα μετακίνησης. Στην περίπτωση που ο χρήστης είχε σε λειτουργία περισσότερα από ένα ποτιστικά καρούλια η πληροφορία αυτή έγινε ακόμη πιο χρήσιμη.

Τα μηνύματα συναγερμού απέτρεψαν αρκετές φορές βλάβες του ποτιστικού συστήματος και διάφορες δυσάρεστες καταστάσεις, όπως πλημμύρες χωραφιού, ζημία της καλλιέργειας, ζημίες σε καλλιέργειες τρίτων και ζημιές των δρόμων. Η λειτουργία που ενημέρωνε το χρήστη ότι το καρούλι έχει σταματήσει έδωσε την δυνατότητα για γρήγορη και άμεση παρέμβαση. Σημαντική αποδείχτηκε και η πληροφορία της επανέναρξης του ποτίσματος. Τα μηνύματα τα οποία ενημέρωναν το χρήστη για τη μεταβολή στο βήμα (ταχύτητα) του ποτί-

σματος συνήθως είχαν πληροφοριακό χαρακτήρα χωρίς να χρειάζεται η παρέμβαση και η εκ νέου ρύθμιση. Υπήρχαν φυσικά και περιπτώσεις που οι αλλαγές ήταν μεγάλες και χρειάστηκε να επισκεφτούν το χωράφι τους ώστε να ρυθμίσουν την ταχύτητα του ποτίσματος. Η ενημέρωση ότι το πότισμα θα ολοκληρωθεί σε 30 λεπτά και η πληροφορία ότι το πότισμα ολοκληρώθηκε λειτούργησαν σαν μια πολύ χρήσιμη υπενθύμιση την όποια οι χρήστες θεώρησαν παρά πολύ εξυπηρετική.

Το σύστημα αυτό εγκαταστάθηκε ήδη σε 4 ποτιστικά καρούλια και λειτούργησε άψογα.

4. Προτάσεις – βελτιώσεις

1.10 Αυτοπροσαρμογή βήματος (ταχύτητα)

Οι αυξομειώσεις στην πίεση του δικτύου έχουν ως αποτέλεσμα και την αλλαγή στο βήμα με το οποίο κινείται το ποτιστικό σύστημα. Στις μέχρι τώρα δοκιμές δεν παρατηρήθηκε ιδιαίτερο πρόβλημα καθώς ο μηχανισμός που δίνει την κίνηση κρατεί αρκετά σταθερό το βήμα. Υπάρχουν φυσικά και οι περιπτώσεις στις οποίες οι μεγάλες αλλαγές στην πίεση του νερού επιφέρουν και αρκετά μεγάλες αλλαγές στο βήμα του. Η τοποθέτηση μιας ηλεκτροβάνας σε συνδυασμό με κατάλληλο λογισμικό θα επιτρέπουν στο μηχανισμό κίνησης να αυτοδιορθώνει (ανάδραση) το βήμα του, αυξάνοντας ή μειώνοντας την ποσότητα του νερού που θα εισέρχεται στην τουρμπίνα η οποία κινεί το ποτιστικό σύστημα.

1.11 Επιπρόσθετοι αισθητήρες

Δυο ακόμη χρήσιμες πληροφορίες που θα μπορούσε να γνωρίζει ο χρήστης είναι η ένταση του αέρα που επικρατεί στο χωράφι του και η πίεση του αρδευτικού δικτύου. Το πρώτο είναι χρήσιμο, ειδικά αν το ποτιστικό μας σύστημα είναι εφοδιασμένο με κάποιο πυραυλικό σύστημα. Αυτό επηρεάζεται κατά πολύ από την ένταση του ανέμου, σε πολλές περιπτώσεις κάνοντας το πότισμα απαγορευτικό. Το δεύτερο, η πίεση του νερού με την οποία τροφοδοτείται το καρούλι, είναι μια πληροφορία από την οποία μπορούμε να υπολογίσουμε την ποσότητα του νερού που πέφτει στην καλλιέργειά μας.

5. Αναφορές – βιβλιογραφία

[1] Βικιπαίδεια, *Γεωργία (δραστηριότητα) --- Βικιπαίδεια, Η Ελεύθερη Εγκυκλοπαίδεια,* 2016.

[2] Π. Ζαχαρίας, ΠΤΥΧΙΑΚΗ Ανάπτυξη εφαρμογής τηλεχειριζόμενου οχήματος, με πλατφόρμα Arduino, Κρήτη: Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης, 2014.

[3] Βικιπαίδεια, *Αισθητήρας Hall --- Βικιπαίδεια, Η Ελεύθερη Εγκυκλοπαίδεια,* 2013.

[4] Βικιπαίδεια, *Φωτοβολταϊκά --- Βικιπαίδεια, Η Ελεύθερη Εγκυκλοπαίδεια,* 2016.

[5] Δ. ΡΙΖΟΣ, «rizosdimitris,» 28 ΑΠΡΙΛΙΟΣ 2012. [Ηλεκτρονικό]. Available: http://rizosdimitris.blogspot.gr/2012/04/blog-post_27.html.

[6] T. Karaferis, «http://www.karaferis.gr/,» [Ηλεκτρονικό].

[7] Βικιπαίδεια, *Μπαταρία --- Βικιπαίδεια, Η Ελεύθερη Εγκυκλοπαίδεια,* 2014.

[8] https://www.arduino.cc/, «https://www.arduino.cc/,» [Ηλεκτρονικό].

[9] Νικολαΐδης Μάρκος, Παρπάς Μάριος και Κονδύλης Στέλιος, " ΈΛΕΓΧΟΣ ΧΩΡΟΥ ΜΕΣΟ ΔΙΚΤΥΟΥ GSM ", Αιγάλεω, 2013.

[10] Μ. ΣΤΑΜΑΤΙΟΣ, ΠΤΥΧΙΑΚΗ: ΕΛΕΓΧΟΣ ΚΑΙ ΚΑΤΑΓΡΑΦΗ ΚΑΙΡΙΚΩΝ ΣΥΝΘΗΚΩΝ ΜΕ ONLINE ΕΝΗΜΕΡΩΣΗ ΣΤΟ ΔΙΑΔΙΚΤΥΟ ΜΕΣΩ ARDUINO, Τ. Η. Μ. Τ.Ε., Επιμ., ΚΑΒΑΛΑ: ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΑΝΑΤΟΛΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ ΚΑΙ ΘΡΑΚΗΣ, 2014.