

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ 1348

Έλεγχος αντλίας ποτίσματος με μικροελεγκτή και τη βοήθεια αισθητήρων, με δυνατότητα απομακρυσμένου ελέγχου μέσω δικτύου GSM

Κωνσταντίνος Τζούμας (5682)

ΕΠΟΠΤΕΥΩΝ ΚΑΘΗΓΗΤΗΣ: Πέτρος Βλαχόπουλος

ΠΑΤΡΑ 2021

©2018, Copyright από Τζούμα Κωνσταντίνο

Πρόλογος

Η παρούσα πτυχιακή εργασία με τίτλο «Έλεγχος αντλίας ποτίσματος με μικροελεγκτή και τη βοήθεια αισθητήρων, με δυνατότητα απομακρυσμένου ελέγχου μέσω δικτύου GSM» εκπονήθηκε στα πλαίσια του προγράμματος σπουδών του Τμήματος Ηλεκτρολόγων Μηχανικών που ανήκει στο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Δυτικής Ελλάδας.

Η πρώτη βασική ανάγκη του ανθρώπου είναι η τροφή, για αυτό το λόγο σε αυτή την πτυχιακή θα ασχοληθούμε με την διευκόλυνση που προσφέρουν τα τεχνολογικά μέσα στην παραγωγική διαδικασία. Στην παρούσα εργασία θα υλοποιηθεί ένα σύστημα έξυπνου ποτίσματος και αντιπαγετικής προστασίας με χρήση της αναπτυξιακής πλατφόρμας Arduino.

Περίληψη

Λαμβάνοντας υπόψιν τις ολοένα και αυξανόμενες ανάγκες του ανθρώπου για μεγαλύτερη και ποιοτικότερη παραγωγή τροφίμων, τον περιορισμό σπατάλης φυσικών και ανθρωπίνων πόρων έγινε σχεδιασμός και κατασκευή ενός έξυπνου συστήματος, το οποίο θα ελέγχει και θα προστατεύει μια γεωργική καλλιέργεια συγκεκριμένα ενός πορτοκαλεώνα. Ζητούμενο είναι ο αγρότης να βρίσκεται πάντα ενημερωμένος για την κατάσταση του αρδευτικού αλλά και αντιπαγετικού συστήματος έτσι ώστε να προστατευτεί από το παγετό αλλά και να γίνεται ορθή άρδευση της καλλιέργειας του.

Η επίτευξη των παραπάνω μπορεί να γίνει με την χρήση της αναπτυξιακής πλακέτας Arduino και αισθητήρων. Λόγω των φυσικών μεγεθών που θέλαμε να μετρήσουμε επιλέξαμε τα αισθητήρια LM35 για την μέτρηση της θερμοκρασίας του περιβάλλοντος, την φωτοδίοδο (LDR) για την μέτρηση του φωτός και το αισθητήριο μέτρησης υγρασίας του εδάφους. Επίσης για τον απομακρυσμένο έλεγχο και πληροφόρηση χρησιμοποιείται ένα GSM Module. Ο μικροελεγκτής Arduino ελέγχει τα φυσικά μεγέθη και μας πληροφορεί όταν υπάρχει χαμηλή θερμοκρασία ή χαμηλή υγρασία εδάφους. Σε κάθε μια από τις περιπτώσεις αυτές ο μικροελεγκτής ενεργεί διαφορετικά. Στην περίπτωση του παγετού θέτει σε λειτουργία το αντλητικό συγκρότημα για την προστασία του πορτοκαλεώνα και μας ειδοποιεί με SMS για την ενέργεια αυτή. Στην περίπτωση της χαμηλής υγρασίας εδάφους μας πληροφορεί με SMS και περιμένει εντολή μας για την ενεργοποίηση του αντλητικού συγκροτήματος για την άρδευση εφόσον είναι βράδυ.

Abstract

Taking into account the growing needs of man for greater and better food production, the reduction of waste of natural and human resources, a smart system was designed and built, which will control and protect an agricultural crop specifically of an orange grove. It is requested that the farmer is always informed about the condition of the irrigation and antifreeze system so that it is protected from frost and that its crop is properly irrigated.

The above can be achieved by using the Arduino development board and sensors. Due to the physical quantities we wanted to measure, we chose the LM35 sensors to measure the ambient temperature, the light emitting diode (LDR) to measure the light and the ground humidity sensor. A GSM Module is also used for remote control and information. The Arduino microcontroller checks the physical sizes and informs us when there is a low temperature or low soil moisture. In each of these cases the microcontroller acts differently. In case of frost, it activates the pumping unit for the protection of the orange grove and notifies us by SMS about this action. In case of low soil moisture, he informs us by SMS and waits for our order to activate the pumping unit for irrigation if it is night.

Πίνακας περιεχομένων

Πρόλογος.....	3
Περίληψη.....	4
Abstract.....	5
Πίνακας περιεχομένων.....	6
Εισαγωγή.....	7
1. Γνωριμία με την αναπτυξιακή πλακέτα Arduino.....	8
2. Τεχνικά χαρακτηριστικά Arduino Mega.....	11
2.2.1 Ακροδέκτες σύνδεσης (Pins) Arduino Mega 2560.....	15
2.2.2 Επιμέρους λειτουργίες των ακροδεκτών σύνδεσης (Pins) εισόδου - εξόδου.....	16
I2C ήTWI (Two-Wire Interface).....	18
2.2.3 Τα pin τροφοδοσίας.....	22
3.1 GPRS shield.....	23
3.2 Εντολές AT για το SIM900.....	25
3.2.1 AT+CMGD.....	26
3.2.2 AT+CMGF.....	29
3.2.3 AT+CMGR.....	30
3.3 Soil moisture sensor.....	31
3.4 Relay module.....	32
3.5 Αισθητήριο θερμοκρασίας LM35.....	33
3.6 LDR (Light dependent resistance) Φωτοαντίσταση.....	34
4. Προγραμματισμός Arduino Mega 2560.....	36
4.1 Γλώσσα προγραμματισμού Wiring.....	36
4.2 Δομή κώδικα.....	44
5. Πειραματική διαδικασία.....	45
5.1 Βαθμονόμηση αισθητήρα θερμοκρασίας.....	45
5.2 Βαθμονόμηση αισθητήρα φωτός.....	46
5.3 Βαθμονόμηση αισθητήρα υγρασίας εδάφους.....	46
6. Πίνακας αυτοματισμού αντλίας.....	47
6.1 Επιτηρητής τάσης.....	49
6.2 Θερμικό προστασίας.....	50
6.3 Βοηθητικά ρελέ.....	51
6.4 Ψηφιακός θερμοστάτης.....	52
6.5 Απαιτούμενες αλλαγές στο πίνακα ελέγχου.....	53
7. Λειτουργία συστήματος - διαδικασία αυτοματισμού.....	54
8. Κώδικας προγράμματος.....	56
Βιβλιογραφία.....	68

Εισαγωγή

Μεγάλες αλλαγές πρόκειται να δημιουργήσει στη διαδικασία της γεωργικής παραγωγής περίπου έως το 2050 η αυξημένη ζήτηση τροφίμων σε παγκόσμιο επίπεδο που υπολογίζεται να αγγίξει 9,1 δις και περισσότερο από 10 δισεκατομμύρια έως το τέλος του αιώνα. Για την επιθυμητή αύξηση των καλλιεργειών προϋπόθεση αποτελεί η καλύτερη διαχείριση των καλλιεργήσιμων εκτάσεων.

Η Ελλάδα έως το τέλος του 1960 αποτελούσε κατά κύρια βάση αγροτική χώρα και οι γεωργικές δραστηριότητες εκτελούνταν παραδοσιακά κυρίως με όργανα με ζώα, αλέτρι και εγχώριους σπόρους. Στη σημερινή γεωργική παραγωγή τα πράγματα γίνονται πιο εύκολα για τον αγρότη καθώς οι γεωργικές δραστηριότητες γίνονται με τρακτέρ, θεριζοαλωνιστικές μηχανές κ.α. Η εξέλιξη στο γεωργικό τομέα θεωρείται απαραίτητη καθώς με τη διευκόλυνση των γεωργικών μέσων και την πιο ξεκούραστη διαδικασία επιτυγχάνεται καλλιέργεια πολύ μεγαλύτερων εκτάσεων σε πολύ λιγότερο χρόνο και με καλύτερα αποτελέσματα.

Εκτός από τα γεωργικά μέσα στη γεωργική διαδικασία ,σήμερα, σημαντική είναι και η βοήθεια που παρέχει στον αγρότη η τεχνολογία. Μέσα από την χρήση των αισθητήριων αναπτύσσονται μοντέρνοι μέθοδοι έλεγχου, παροχή πληροφοριών κ.α. που συμβάλλουν σημαντικά στη συνολική προσπάθεια για την επίτευξη των γεωργικών στόχων. Στην παρούσα πτυχιακή ο μικροελεγκτής (Arduino) που θα χρησιμοποιηθεί θα λαμβάνει τις μετρήσεις από τους αισθητήρες θερμοκρασίας περιβάλλοντος, υγρασίας εδάφους, φωτός και θα μας πληροφορεί με SMS για την κατάσταση της αντλίας ποτίσματος, την υγρασία του εδάφους, την εξωτερική θερμοκρασία καθώς και για απρόσμενα γεγονότα όπως π.χ. διακοπή ρεύματος.

Στην εργασία θα υλοποιηθεί ένα σύστημα έξυπνου ποτίσματος με χρήση της αναπτυξιακής πλατφόρμας Arduino. Ο χρήστης θα μπορεί να στέλνει SMS έτσι ώστε να ξεκινά η αντλία να ποτίζει για ένα χρονικό διάστημα, καθοριζόμενο από το κείμενο του SMS. Επίσης θα μπορεί ανά πάσα στιγμή ο χρήστης να ζητήσει να λάβει μήνυμα από το σύστημα για την κατάσταση της αντλίας ή για απρόσμενα γεγονότα (πχ. διακοπή ρεύματος). Ακόμα με τη χρήση αισθητήρα θερμοκρασίας μπορεί ο μικροελεγκτής να ξεκινά το πότισμα πχ. όταν η θερμοκρασία πέσει κάτω από 2 βαθμούς έτσι ώστε να μην παγώσουν τα κηπευτικά, ενώ όταν η θερμοκρασία είναι πολύ υψηλή (πχ. πάνω από 35 βαθμούς) να μην ξεκινάει πότισμα αν είναι μέρα (κάτι που θα ελέγχεται από αισθητήρα φωτός). Τέλος με έναν αισθητήρα υγρασίας χώματος το σύστημα θα μας ενημερώνει ανά πάσα στιγμή αν αυτή πέσει κάτω από κάποιο όριο.

1. Γνωριμία με την αναπτυξιακή πλακέτα Arduino

Το Arduino είναι μια αναπτυξιακή πλακέτα βασισμένη σε έναν μικροελεγκτή ο οποίος προγραμματίζεται με τη γλώσσα wiring. Το Arduino μπορεί να χρησιμοποιηθεί για την κατασκευή διαφόρων πρότζεκτ σε εκπαιδευτικό επίπεδο και μη. Οι περισσότεροι τύποι του Arduino είναι προκατασκευασμένοι, άρα δεν χρειάζονται ιδιαίτερες γνώσεις επάνω στα ηλεκτρονικά κυκλώματα.

Η πλακέτα του Arduino αποτελείται από τον μικροελεγκτή, συνήθως είναι ένας Atmel AVR, καθώς και κάποια άλλα ολοκληρωμένα κυκλώματα που βοηθούν στο προγραμματισμό μέσω κοινής θύρας USB. Σημαντικός είναι και ο ενσωματωμένος ρυθμιστής τάσης (*Εικόνα 2*) επάνω στη πλακέτα για την σωστή τάση τροφοδοσίας του μικροελεγκτή. Έναν κρυσταλλικό ταλαντωτή (*Εικόνα 1*) με συχνότητα τα 16 Mhz που σκοπό έχει τη δημιουργία παλμών για τη λειτουργία του μικροελεγκτή. Από την εργοστασιακή μορφή έχει εγκατεστημένο έναν bootloader, έτσι είναι απλούστερη η χρήση του γιατί δεν απαιτείται εξωτερικός προγραμματιστής (*Εικόνα 3*) για τον προγραμματισμό του. Ένα chip USB to serial αναλαμβάνει όλη τη δουλειά να περάσει το κώδικα μας στον μικροελεγκτή. Οι διαφορές στις εκδόσεις του Arduino έχουν να κάνουν ως προς τις εισόδους εξόδους και το μικροελεγκτή.



Εικόνα 1. (Κρυσταλλικός ταλαντωτής 16 MHz)



Εικόνα 2. (Ρυθμιστής τάσης AMS1117)



Εικόνα 3. (Εξωτερικός προγραμματιστής)

Χαρακτηριστικά παραδείγματα Arduino αποτελούν οι παρακάτω εκδοχές του.

- Arduino Uno
- Arduino Leonardo
- Arduino Mega 2560
- Arduino LilyPad
- Arduino Mega ADK
- Arduino Fio

- Arduino Ethernet
- Arduino Pro
- Arduino BT
- Arduino Nano
- Arduino Mini
- Arduino Pro Mini
- Arduino Uno WiFi

Το κάθε ένα είναι σχεδιασμένο έτσι ώστε να καλύπτει τις ανάγκες της κάθε πιθανής εφαρμογής που θα θέλει κάποιος να κατασκευάσει.



Εικόνα 4. (Διάφορες αναπτυξιακές πλακέτες Arduino)



Εικόνα 5. (Arduino Uno WiFi)

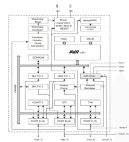
2. Τεχνικά χαρακτηριστικά Arduino Mega

Παρακάτω βλέπουμε κάποια από τα τεχνικά χαρακτηριστικά όπως μας τα δίνει ο κατασκευαστής.

Microcontroller	ATmega2560
Τάση τροφοδοσίας	5V
Τάση εισόδου (προτεινόμενη)	7-12V
Τάση εισόδου (Όρια)	6-20V
Ψηφιακές είσοδοι/έξοδοι	54 από τις οποίες 6 είναι Pulse Width Modulation (ψευδοαναλογικές)
Αναλογικές είσοδοι	6

DC ρεύμα σε κάθε είσοδο/έξοδο	20 mA
DC ρεύμα σε στην έξοδο 3.3V	50 mA
Flash Memory	256 KB από τα οποία τα 8 KB χρησιμοποιούνται από τον bootloader
SRAM	8 KB
EEPROM	4KB
Clock Speed	16 Mhz
Μήκος	101,52 mm
Πλάτος	53,3 mm
Βάρος	37gr

Στην επόμενη εικόνα βλέπουμε το σχηματικό Block διάγραμμα λειτουργίας του μικροελεγκτή Atmega 2560.



Εικόνα 6. (Σχηματικό διάγραμμα Atmega 2560)

2.2 Εξαρτήματα (Hardware)

Στην εικόνα που ακολουθεί βλέπουμε το Arduino Mega 2560 Ver.3 το οποίο θα χρησιμοποιήσουμε και εξηγούνται τα στοιχεία που το συνθέτουν.



Εικόνα 7. (Arduino Mega 2560 Ver.3)

Αναλυτικότερα διακρίνουμε πάνω στη πλακέτα του Arduino Mega τα εξής :

- Το βύσμα των 2.1mm εξωτερικής τροφοδοσίας από το οποίο μπορούμε να τροφοδοτήσουμε το arduino. Για παράδειγμα όταν το εγκαταστήσουμε κάπου μόνιμα ή αν δεν θέλουμε μεσώ του usb.
- Ο ρυθμιστής τάσης ο οποίος αναλαμβάνει την σωστή τροφοδοσία του κυκλώματος του arduino. Μας δίνει 5 volts ενώ εμείς μπορούμε να δώσουμε μέσω του βύσματος από 7-12 volts από μια μπαταριά ή από οποιαδήποτε άλλη πηγή DC.
- Η θήρα usb μεσώ της οποίας μπορούμε να τροφοδοτήσουμε το Arduino μεσώ του υπολογιστή μας άλλα το κυριότερο να το προγραμματιστεί . Για τη σύνδεση απαιτείται ένα καλώδιο usb type A σε type B.

- Ο ελεγκτής serial over usb είναι αυτός που μετατρέπει τη σειριακή σε usb σύνδεση έτσι μπορούμε να επιβλέπομαι στον υπολογιστή μας δεδομένα από το Arduino αλλά και να έχουμε εύκολα τη δυνατότητα να περάσουμε το κωδικά μας χωρίς την ανάγκη επιπλέον προγραμματιστή.
- Το pin led 13 το οποίο είναι συνδεδεμένο με την έξοδο 13, με το οποίο μπορούμε να υλοποιήσουμε το απλό blink χωρίς να χρειαστούμε τίποτα άλλο πέραν του Arduino board.
- Τα serial RX/TX leds τα οποία είναι ενδεικτικά και βλέπουμε εάν υπάρχει επικοινωνία μέσω του μετατροπέα usb to serial
- Το power led είναι ενδεικτικό και μας δείχνει την κατάσταση λειτουργίας της πλακέτας.
- το reset button που όταν το πιέσουμε επανεκκινεί το Arduino.
- Ο μικροελεγκτής Atmega2560 ο οποίος είναι η καρδιά του συστήματος μας.

2.2.1 Ακροδέκτες σύνδεσης (Pins) Arduino Mega 2560

Το ArduinoMega 2560 απαρτίζεται από ένα μεγάλο πλήθος εισόδων - εξόδων

1. 16 αναλογικές εισόδους A0 έως A15 που χρησιμοποιούν έναν ADC (analog to digital converter) ο οποίος είναι ενσωματωμένος στο μικροελεγκτή. Για παράδειγμα εάν τροφοδοτήσουμε μια από αυτές τις εισόδους με τάση 0-5 Volts ο μικροελεγκτής λόγω της 10-bit ανάλυσης θα διαβάσει μια τιμή από 0 (για τάση 0 Volts) - 1023 (για τάση 5 Volts).
2. 44 pin τα 0 έως 53 τα οποία μπορούν να λειτουργούν στα 5 Volts και να μας δώσουν ή να δεχθούν έως 40 mA. Αυτά τα ψηφιακά pin μπορούν να προγραμματιστούν είτε σαν εισοδοι αλλά και έξοδοι. Κάποια από αυτά εκτελούν και κάποιες άλλες λειτουργίες οι οποίες αναλύονται παρακάτω.
3. Τα pin τροφοδοσίας των οποίων η λειτουργίες αναλύονται παρακάτω.

2.2.2 Επιμέρους λειτουργίες των ακροδεκτών σύνδεσης (Pins) εισόδου - εξόδου

➤ Σειριακές θύρες επικοινωνίας

Οι θύρες επικοινωνίας του Arduino Mega 2560 είναι αρκετές, δίνοντας έτσι τη δυνατότητα ευελιξία στις εφαρμογές. Αναλυτικότερα έχουμε:

- Σειριακή θύρα: Pin 0(RX) και Pin1(TX),
- Σειριακή θύρα 1: Pin 19(RX) και Pin18(TX),
- Σειριακή θύρα 2: Pin 17(RX) και Pin16(TX),
- Σειριακή θύρα 3: Pin 15(RX) και Pin14(TX)

Τα Pins 0,19,17,15 (RX) λαμβάνουν σειριακά δεδομένα και τα Pins 1,18,16,14 (TX) στέλνουν σειριακά δεδομένα υπό το πρότυπο TTL

➤ Εξωτερικό Interrupt

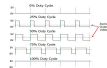
Τα pin18,19,20,21 όπως και τα 2 και 3 χρησιμοποιούνται για το εξωτερικό interrupt. Αναλόγως τον κώδικα μας δίνετε η δυνατότητα όταν δοθεί σήμα σε κάποιο από αυτά η ροή του προγράμματος που τρέχει σταμάτα άμεσα και εκτελείτε μια ορισμένη εντολή που έχει οριστεί.

➤ (PWM) Pulse Width Modulation

PWM (Pulse Width Modulation) Pins: τα Pins 2 έως 13 και τα 44 έως46 έχουν τη δυνατότητα να λειτουργήσουν και με διαμόρφωση εύρους παλμών. Τα pin 3,5,6,9,10 και 11 μπορούν να λειτουργήσουν και ως αναλογικές εξόδοι χρησιμοποιώντας το σύστημα PWM (pulse width modulation). Με αυτό τον τρόπο μπορούμε να έχουμε αναλογική έξοδο 0-5 volt.

Στη παρακάτω εικόνα βλέπουμε τη μορφή εξόδου της αναλογικής εξόδου PWM. Αναλόγως τον χρόνο ενεργοποίησης της εξόδου(Duty Cycle) μεταβάλλεται η μέση τιμή τάσης αλλά και ρεύματος που τροφοδοτεί ένα φορτίο. Όπως βλέπουμε και στην εικόνα για:

- Duty Cycle 0% η μέση τιμή τάσης θα είναι 0 Volts
- Duty Cycle 25% η μέση τιμή τάσης θα είναι 1.25 Volts
- Duty Cycle 50% η μέση τιμή τάσης θα είναι 2.5 Volts
- Duty Cycle 75% η μέση τιμή τάσης θα είναι 3.75 Volts
- Duty Cycle 100% η μέση τιμή τάσης θα είναι 5 Volt



Εικόνα 8. (PWM έξοδος)

➤ SPI (Serial Peripheral Interface bus)

Ο δίαυλος σειριακής περιφερειακής διασύνδεσης (SPI) είναι μια συγχρονισμένη σειριακή επικοινωνία που χρησιμοποιείται για επικοινωνία σε μικρές αποστάσεις. Η διασύνδεση αναπτύχθηκε από τη Motorola στα μέσα της δεκαετίας του '80 και έχει γίνει de facto πρότυπο. Οι συσκευές με SPI επικοινωνούν σε full duplex δηλαδή ο Master να λαμβάνει δεδομένα από τον Slave και να του μεταδίδει ταυτόχρονα. Η αρχιτεκτονική που εφαρμόζεται είναι η master-slave με ένα μόνο master σε κάθε δίκτυο. Η κύρια συσκευή παράγει το πλαίσιο για το πότε θα γίνει αποστολή και λήψη δεδομένων. Υπάρχει η δυνατότητα πολλαπλών σκλάβων συσκευών (slave) μέσω επιλογής με μεμονωμένες γραμμές επιλεκτικής σάρωσης SS (Slave Select).

Το Arduino Mega 2560 μέσω των ακροδεκτών (pins) 50(MISO), 51(MOSI), 52(SCK), 53 (SS) μπορεί να επικοινωνήσει με άλλες συσκευές που διαθέτουν SPI. Ο ρόλος του Arduino Mega 2560 μπορεί να είναι Master είτε Slave.

Ο δίαυλος SPI αποτελείται από τέσσερα λογικά σήματα:

- SCLK: Σειριακό ρολόι (έξοδος από το Master).
- MOSI: Master Out Slave In (έξοδος δεδομένων από τον Master).
- MISO: Master In Slave Out (έξοδος δεδομένων από τον Slave).
- SS: Επιλογή Slave (έξοδος από τον Master συνήθως ανενεργή).



Εικόνα 9. (SPI Επικοινωνία)

I2C ή TWI (Two-Wire Interface)

Τα Pins 20 (SDA) και 21 (CSL) υποστηρίζουν τη λειτουργία Two Wire (I2C)

Ο δίαυλος επικοινωνίας **I²C** είναι ένας σειριακός δίαυλος που δημιουργήθηκε από τη Philips ο οποίος χρησιμοποιείται για την σύνδεση περιφερειακών μικρής ταχύτητας σε ηλεκτρονικές συσκευές. Ο δίαυλος επικοινωνίας I2C δεν χρησιμοποιείται μόνο για συσκευές που βρίσκονται πάνω σε ένα τυπωμένο κύκλωμα, αλλά και για την επικοινωνία συσκευών που συνδέονται με αυτή μέσω καλωδίου.

Στο δίαυλο I2C για τη μεταφορά δεδομένων απαραίτητα είναι μόνο δυο καλώδια τα οποία είναι ημιαμφίδρομης κατεύθυνσης. Ο ένας αγωγός χρησιμοποιείται για το SCL, δηλαδή για το συγχρονισμό με το ρολόι. Ο δεύτερος αγωγός το SDA χρησιμοποιείται για τη μεταφορά δεδομένων.

Για τη λειτουργία επίσης απαιτείται και ένας αγωγός γείωσης GND ή 0V. Οι τρεις αυτοί αγωγοί GND, SDA και SCL πρέπει να είναι συνδεδεμένοι σε κάθε συσκευή του δικτύου I2C.

Ανάλογα τη συσκευή μπορεί να είναι απαραίτητη η τροφοδοσία της, για αυτό χρησιμοποιείται ένας τέταρτος αγωγός το (VCC). Οι τυπικές τάσεις που συναντώνται σε ένα σύστημα I2C είναι τα +5V & +3.3V αν και είναι δυνατή η χρήση από +1,2V έως +5,5V.

Το πλήθος των συσκευών που μπορούν να συνδεθούν στο δίαυλο I2C περιορίζεται από τις διαθέσιμες διευθύνσεις αλλά και το μήκος της γραμμής επικοινωνίας το οποίο δεν πρέπει να υπερβαίνει τα 2 μέτρα ώστε η χωρητικότητα της γραμμής να είναι μικρότερη από 400pF.



Εικόνα 10. (I²C διάγραμμα συνδεσμολογίας Master και Slave Συσκευών)

Οι ακροδέκτες SCL και SDA μπορούν να είναι *ανοικτού απαγωγού* (open drain) ή *ανοικτού συλλέκτη* (open collector). Κάθε μία από τις δύο γραμμές, πρέπει να συνδέεται με μία αντίσταση με την γραμμή τροφοδοσίας, την *αντίσταση τερματισμού*. Η τιμή των αντιστάσεων δεν είναι κρίσιμη (1KΩ έως 10KΩ), αλλά μαζί με την χωρητικότητα του διαύλου, επηρεάζει την μέγιστη ταχύτητα λειτουργίας του διαύλου. Μεγάλες χωρητικότητες του διαύλου, μπορούν να αντισταθμιστούν με μικρές αντιστάσεις τερματισμού. Η χρήση αντιστάσεων είναι απαραίτητη, χωρίς αυτές οι γραμμές SCL και SDA θα είναι μονίμως σε κατάσταση λογικού 0 με αποτέλεσμα ο δίαυλος να μην δουλεύει.

Οι συσκευές στον δίαυλο επικοινωνίας I²C είναι είτε (*Masters*) είτε (*Slave*). Η Master συσκευή είναι αυτή που ελέγχει τη γραμμή ρολογιού SCL (παράγει τους παλμούς ρολογιού). Οι Slave συσκευές ανταποκρίνονται στις συσκευές Master. Μία συσκευή Slave δεν μπορεί να

ξεκινήσει να μεταφέρει δεδομένα πάνω στο δίαυλο. Σε έναν δίαυλο είναι δυνατόν να έχουμε συνδεδεμένες πολλές Master και Slaves συσκευές, αλλά μόνο οι Master συσκευές ελέγχουν την μεταφορά.

Τα δεδομένα μεταφέρονται σε ακολουθίες των 8 bit. Τα bit αποστέλλονται μέσω της γραμμής SDA, ξεκινώντας πάντα από το (MSB) περισσότερο σημαντικό bit . Η γραμμή SCL μεταβαίνει στη κατάσταση high και μετά low. Κάθε 8 bit δεδομένων που μεταφέρονται, η συσκευή που τα λαμβάνει στέλνει πίσω ένα bit επιβεβαίωσης (ACK). Έτσι πραγματικά απαιτούνται 9 παλμοί ρολογιού, για την μεταφορά κάθε byte δεδομένων. Όταν η συσκευή που λαμβάνει, στείλει πίσω ένα low *bit επιβεβαίωσης (ACK)*, τότε έχει λάβει τα δεδομένα και είναι έτοιμη να λάβει το επόμενο byte δεδομένων. Εάν όμως στείλει πίσω ένα high bit επιβεβαίωσης (NACK δηλαδή Not Acknowledged), δείχνει ότι η συσκευή που λαμβάνει, δεν μπορεί να λάβει επιπλέον δεδομένα και πρέπει η Master συσκευή να τερματίσει την αποστολή δεδομένων.

Ο δίαυλος επικοινωνίας I²C είναι κατάλληλος για περιφερειακά, όπου η απλότητα και το χαμηλό κόστος κατασκευής είναι σημαντικότερα σε σχέση με την ταχύτητα. Συνηθισμένες εφαρμογές είναι:

- Ανάγνωση-εγγραφή *σειριακών* μνημών EEPROM
- Πρόσβαση σε χαμηλής ταχύτητας *μετατροπείς αναλογικού σήματος σε ψηφιακό (ADC) ή ψηφιακού σε αναλογικό (DAC)*.
- Ανάγνωση αισθητήρων με σύνδεση I²C
- Ανάγνωση *ρολογιών πραγματικού χρόνου (Real Time Clocks-RTCs)*
- Ανάγνωση επιτηρητών Hardware και διαγνωστικών αισθητήρων

Στην αγορά υπάρχουν μικροελεγκτές με ενσωματωμένες θύρες I²C, αλλά ένα ιδιαίτερα ισχυρό χαρακτηριστικό είναι ότι ένας μικροελεγκτής μπορεί να εξομοιώσει τις θύρες I²C μόνο με γενικής χρήσης ακροδέκτες (general purpose I/O) και λογισμικό, δεν χρειάζεται εξειδικευμένο hardware.

Για παράδειγμα τα ESP 8266 και ESP 32 που είναι και αυτά αναπτυξιακές πλακέτες χρησιμοποιούν τα GPIO για παραπάνω από μια λειτουργίες. Ανάλογα το GPIO που θα χρησιμοποιήσουμε αυτό μπορεί να λειτουργήσει σαν απλή ψηφιακή είσοδος ή έξοδος, αναλογική είσοδος ή έξοδος αλλά και δίαυλος επικοινωνίας (SPI, I²C). Στην παρακάτω εικόνας βλέπουμε τις δυο αυτές αναπτυξιακές πλακέτες με το pinout για την καθεμία.



Εικόνα 11. (Λειτουργίες Pin αναπτυξιακή πλακέτα ESP 32)



Εικόνα 12. (Λειτουργίες Pin αναπτυξιακή πλακέτα ESP 8266)

2.2.3 Τα pin τροφοδοσίας

Οι ακροδέκτες (Pins) τροφοδοσίας αποτελούνται από:

- Reset

Το Reset το οποίο εάν γειωθεί σε ένα GND κάνει επανεκκίνηση το Arduino όπως και το ενσωματωμένο στη πλακέτα μπουτόν (Button) RESET.

- 3.3V

Το 3.3 V το οποίο μας δίνει τάση 3.3V το μέγιστο ρεύμα που μπορεί να δώσει είναι 50 μ A.

- 5V

Το 5V το οποίο δίνει τάση 5V κατευθείαν από το usb εάν τροφοδοτηθεί το σύστημα μέσω αυτού ή κάνοντας χρήση του ρυθμιστή τάσης αν τροφοδοτείτε μέσω του βύσματος 2.1 mm.

- GND

Δυο GND τα οποία είναι οι γειώσεις (κοινό σημείο τάση αναφοράς 0 Volts).

- Vin

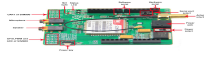
Το Vin μέσω του οποίου μπορεί να τροφοδοτηθεί το Arduino αντί του βύσματος 2.1 mm. Αν υπάρχει εξωτερική τροφοδοσία συνδεδεμένη στο βύσμα 2.1mm τότε και το Vin θα έχει την ίδια τάση είναι δηλαδή άμεσα συνδεδεμένα χωρίς να μεσολαβεί κάποιος ρυθμιστής τάσης.

3.1 GPRS shield

Το GPRS shield το οποίο χρησιμοποιούμε είναι βασισμένο στο chip sim 900 της SIMCOM. Είναι σχεδιασμένο να λειτουργεί στο GSM δίκτυο της κινητής τηλεφωνίας πράγμα που μας προσφέρει μεγάλη ευελιξία και δυνατότητες στον απομακρυσμένο έλεγχο. Το shield αυτό μας προσφέρει δυνατότητα να στέλνουμε SMS, MMS καθώς και να πραγματοποιούμε φωνητικές κλήσεις. Όλα αυτά είναι δυνατά χρησιμοποιώντας τις εντολές AT.

Μερικά από τα τεχνικά χαρακτηριστικά του φαίνονται παρακάτω όπως μας δίνονται στο site του κατασκευαστή. (http://www.geeetech.com/wiki/index.php/Arduino_GPRS_Shield)

- *Quad-Band 850 / 900/ 1800 / 1900 MHz*
- *GPRS multi-slot class 10/8*
- *GPRS mobile station class B*
- *Compliant to GSM phase 2/2+*
- *Κλάση 4 (2 W @ 850 / 900 Mhz)*
- *Κλάση 1 (1 W @ 1800 / 1900MHz)*
- *Έλεγχος μέσω εντολών AT*
- *Υπηρεσία γραπτών μηνυμάτων SMS (Short Message Service)*
- *Embedded TCP/UDP stack*
- *Υποστηριζόμενη λειτουργία RTC (Real Time Clock).*
- *Δυνατότητα επιλογής σειριακής θύρας*
- *Βύσματα 3.5 mm ακουστικών και μικρόφωνου*
- *Χαμηλή κατανάλωση ενέργειας - 1.5mA (sleep mode)*
- *Εύρος θερμοκρασίας λειτουργίας -40°C έως +85 °C*



Εικόνα 13. (Sim 900 Module)

Στη παραπάνω *Εικόνα 13* βλέπουμε το GPRS Module το οποίο έχει ενσωματωμένο πάνω του το τσιπ SIM900 καθώς και:

- Power key: Μπουτόν από το οποίο ενεργοποιούμε/απενεργοποιούμε το shield
- Net light: Ενδεικτική λυχνία led όπου βλέπουμε την κατάσταση δικτύου, ανάλογα της κατάστασης βρίσκεται σε σύνδεση ή αναζητά δίκτυο.
- Status led: Ενδεικτική λυχνία led ή οποία μας δείχνει εάν είναι σε λειτουργία το GPRS shield.
- Software serial: Μέσω αυτών των pin μπορεί να επικοινωνήσει το GPRS shield με το Arduino χρησιμοποιώντας τα Digital pins 7-8
- Hardware serial: Μέσω αυτών των pin συνδέεται στη σειριακή θύρα του Arduino χρησιμοποιώντας τα Digital pins 0,1
- Serial port select: Είναι τα pins στα οποία αναλόγως πού τοποθετήσαμε τα jumper (βραχυκυκλωτήρες) επιλέγουμε τον τρόπο επικοινωνίας hardware ή software serial.
- Power select: Με το διακόπτη αυτόν προσδιορίζουμε την τροφοδοσία του GPRS shield, η οποία μπορεί να επιτευχθεί διαμέσου των pins του Arduino (εάν το GPRS shield βρίσκεται κουμπωμένο πάνω στο arduino) ή από εξωτερική πηγή τροφοδοσίας

δια μέσω του Power jack.

- Power jack: Βύσμα των 2,1 mm διαμέσου του οποίου μπορούμε να τροφοδοτήσουμε το GPRS shield από μια πηγή τάσης (π.χ. μπαταρία).
- Antenna interface: Σε αυτό το ειδικό βύσμα (sma female) συνδέεται η κεραία του δικτύου GSM (sma male)
- Microphone: Βύσμα για την σύνδεση μικρόφωνου (σε περίπτωση φωνητικής κλήσης).
- Speaker: Βύσμα για τη σύνδεση ακουστικών ή ηχείων (σε περίπτωση φωνητικής κλήσης).
- UART of SIM900: Τα pins αυτά μπορούν να χρησιμοποιηθούν για τη σύνδεση του GPRS shield μέσω UART(Universal Asynchronous Receiver / Transmitter).
- GPIO, PWM and ADC of SIM900: Ακροδέκτες για συνδέσεις GPIO (General Purpose Input / Output) και ADC (Analog to Digital Converter).

3.2 Εντολές AT για το SIM900

Οι εντολές του SIM900 για τον έλεγχο μηνυμάτων SMS σύμφωνα με τον κατασκευαστή είναι οι παρακάτω:

Εντολή	Περιγραφή λειτουργίας εντολής
AT+CMGD	Διαγραφή μηνύματος SMS
AT+CMGF	Επιλογή μορφοποίησης μηνύματος SMS
AT+CMGL	Δημιουργία λίστας μηνυμάτων SMS από την προτιμώμενη μνήμη
AT+CMGR	Ανάγνωση μηνύματος SMS
AT+CMGS	Αποστολή μηνύματος SMS
AT+CMGW	Εγγραφή μηνύματος SMS στη μνήμη

AT+CMSS	Αποστολή μηνύματος SMS από τη μνήμη
AT+CNMI	Ειδοποίηση για νέο μήνυμα SMS
AT+CPMS	Προτιμώμενη μνήμη αποθήκευσης μηνύματος SMS
AT+CRES	Επαναφορά ρυθμίσεων μηνυμάτων SMS
AT+CSAS	Αποθήκευση ρυθμίσεων μηνυμάτων SMS
AT+CSCA	Κέντρο διευθύνσεων μηνυμάτων SMS
AT+CSCB	Επιλογή κυψέλης κινητής τηλεφωνίας αποστολής μηνυμάτων SMS
AT+CSDH	Εμφάνιση παραμέτρων μορφοποίησης μηνυμάτων SMS
AT+CSMP	Αποθήκευση παραμέτρων μορφοποίησης μηνυμάτων SMS
AT+CSMS	Επιλογή λειτουργίας μηνύματος

Η σύνταξη των εντολών που θα χρησιμοποιήσουμε στην εφαρμογή μας έχει ως εξής:

3.2.1 AT+CMGD

Εντολή για την διαγραφή μηνυμάτων από τη μνήμη όπως βλέπουμε στην *εικόνα 14* και *εικόνα 15* από το manual του κατασκευαστή εάν επιλέξουμε το:

- “0” Διαγραφή του μηνύματος που είναι στη θέση μνήμης <Index>.
- “1” Διαγραφή όλων των αναγνωσμένων μηνυμάτων από την προτιμώμενη θέση μνήμης, αφήνοντας τα μη αναγνωσμένα και αποθηκευμένα μηνύματα που έχουν σταλεί ή όχι άθικτα.
- “2” Διαγραφή όλων των αναγνωσμένων μηνυμάτων από την προτιμώμενη θέση αποθήκευσης μηνυμάτων και αποστολή των μηνυμάτων που προέρχονται από κινητά, αφήνοντας τα μη αναγνωσμένα μηνύματα και τα μηνύματα που προέρχονται από κινητά άθικτα.
- “3” Διαγραφή όλων των αναγνωσμένων μηνυμάτων από την προτιμώμενη θέση αποθήκευσης μηνυμάτων και αποστολή των μη απεσταλμένων μηνυμάτων που προέρχονται από κινητά, αφήνοντας τα μη αναγνωσμένα μηνύματα άθικτα.
- “4” Διαγραφή όλων των αναγνωσμένων μηνυμάτων από την προτιμώμενη θέση αποθήκευσης μηνυμάτων συμπεριλαμβανομένων και των μη αναγνωσμένων .

Στη περίπτωση μας θα χρησιμοποιήσουμε το “1” και το “4”, η εντολή θα έχει τη μορφή :
AT+CMGD =1,4



Εικόνα 14

UNIT 15: THE HISTORY OF THE UNITED STATES	
UNIT OBJECTIVES	
Knowledge	<ul style="list-style-type: none"> 1. Know the major events of the American Revolution. 2. Know the major events of the American Civil War. 3. Know the major events of the American Industrial Revolution. 4. Know the major events of the American Westward Expansion. 5. Know the major events of the American Immigration. 6. Know the major events of the American Civil Rights Movement. 7. Know the major events of the American Space Program. 8. Know the major events of the American Environmental Movement. 9. Know the major events of the American Cultural Movement. 10. Know the major events of the American Economic Crisis. 11. Know the major events of the American Globalization. 12. Know the major events of the American Technology. 13. Know the major events of the American Healthcare. 14. Know the major events of the American Education. 15. Know the major events of the American Social Security. 16. Know the major events of the American Medicare. 17. Know the major events of the American Medicaid. 18. Know the major events of the American Social Security Administration. 19. Know the major events of the American Medicare Administration. 20. Know the major events of the American Medicaid Administration.
Skills	<ul style="list-style-type: none"> 1. Analyze the causes and effects of the American Revolution. 2. Analyze the causes and effects of the American Civil War. 3. Analyze the causes and effects of the American Industrial Revolution. 4. Analyze the causes and effects of the American Westward Expansion. 5. Analyze the causes and effects of the American Immigration. 6. Analyze the causes and effects of the American Civil Rights Movement. 7. Analyze the causes and effects of the American Space Program. 8. Analyze the causes and effects of the American Environmental Movement. 9. Analyze the causes and effects of the American Cultural Movement. 10. Analyze the causes and effects of the American Economic Crisis. 11. Analyze the causes and effects of the American Globalization. 12. Analyze the causes and effects of the American Technology. 13. Analyze the causes and effects of the American Healthcare. 14. Analyze the causes and effects of the American Education. 15. Analyze the causes and effects of the American Social Security. 16. Analyze the causes and effects of the American Medicare. 17. Analyze the causes and effects of the American Medicaid. 18. Analyze the causes and effects of the American Social Security Administration. 19. Analyze the causes and effects of the American Medicare Administration. 20. Analyze the causes and effects of the American Medicaid Administration.
Resources	<ul style="list-style-type: none"> 1. Textbook: "The History of the United States". 2. Textbook: "The American Revolution". 3. Textbook: "The American Civil War". 4. Textbook: "The American Industrial Revolution". 5. Textbook: "The American Westward Expansion". 6. Textbook: "The American Immigration". 7. Textbook: "The American Civil Rights Movement". 8. Textbook: "The American Space Program". 9. Textbook: "The American Environmental Movement". 10. Textbook: "The American Cultural Movement". 11. Textbook: "The American Economic Crisis". 12. Textbook: "The American Globalization". 13. Textbook: "The American Technology". 14. Textbook: "The American Healthcare". 15. Textbook: "The American Education". 16. Textbook: "The American Social Security". 17. Textbook: "The American Medicare". 18. Textbook: "The American Medicaid". 19. Textbook: "The American Social Security Administration". 20. Textbook: "The American Medicare Administration". 21. Textbook: "The American Medicaid Administration".

Εικόνα 15

3.2.2 AT+CMGF

Εντολή για την επιλογή του τύπου μηνύματος όπως βλέπουμε στην *εικόνα16* από το manual του κατασκευαστή εάν επιλέξουμε το “0” το μήνυμα θα έχει μορφή PDU, εάν επιλέξουμε το “1” θα έχει μορφή Text δηλαδή απλού κειμένου. Στη περίπτωση μας θα χρησιμοποιήσουμε το “1” (Text) και η εντολή θα έχει τη μορφή : AT+CMGF =1



Εικόνα 16

3.3 Soil moisture sensor

Το αισθητήριο μέτρησης της υγρασίας του χώματος είναι μια απλή ηλεκτρονική διάταξη με την οποία μετρούμε την υγρασία του χώματος. Η αρχή λειτουργίας είναι απλή, έχει δυο ακίδες οι οποίες τοποθετούνται στο χώμα. Το ηλεκτρικό κύκλωμα που δημιουργείται είναι ίδιο με αυτό ενός διαιρέτη τάσης. Έχουμε μια σταθερή αντίσταση στο κύκλωμα και μια μεταβλητή το χώμα. Όταν η υγρασία στο χώμα είναι χαμηλή η περιεκτικότητα του σε νερό είναι μικρή άρα παρουσιάζει μεγάλη ωμική αντίσταση. Ακριβώς το αντίστροφο ισχύει όταν η υγρασία είναι υψηλή, αυτό έχει σαν αποτέλεσμα η ωμική αντίσταση που παρουσιάζει το χώμα να είναι μικρή. Αυτό έχει σαν αποτέλεσμα οι τιμές της τάσης στην έξοδο A0 του αισθητηρίου να είναι αντιστρόφως ανάλογες του ποσοστού υγρασίας στο χώμα.



Εικόνα 18. (Soil Moisture Sensor)

Στην *εικόνα 18* βλέπουμε το κύκλωμα του αισθητηρίου αποτελούμενο από τις ακίδες και την αντίσταση 10K, στη πλακέτα διακρίνουμε ένα μικρό ηλεκτρονικό ολοκληρωμένο κύκλωμα με και τρίμμερ για την ρύθμιση της ψηφιακής εξόδου που έχει ενσωματωμένη. τους ακροδέκτες +5V, A0 (αναλογική έξοδος) και GND. Δεν θα το χρησιμοποιήσουμε με ψηφιακή έξοδο καθώς θέλουμε να καθορίσουμε εμείς τα όρια της υγρασίας. Παρακάτω βλέπουμε τα τεχνικά χαρακτηριστικά που μας δίνει ο κατασκευαστής.

- Τάση τροφοδοσίας: 3.3V έως 5V
- Ρεύμα λειτουργίας: 0 έως 35 μ A.
- Τιμή A0 σε ξερό χώμα: 0~300
- Τιμή A0 σε υγρό χώμα: 300~700
- Τιμή A0 σε νερό: 700~950

3.4 Relay module

Το Relay module που θα χρησιμοποιήσουμε είναι μια πλακέτα η οποία ενσωματώνει ρελέ ισχύος. Το κάθε ρελέ έχει κλέμα σύνδεσης με τρεις ακροδέκτες, τον κοινό κόμβο (common), και δυο επαφές οι οποίες σε κατάσταση ηρεμίας είναι μια κλειστή επαφή (N.C.) και μια ανοιχτή (N.O.). Για τη λειτουργία του όπως παρατηρούμε απαιτείται η τροφοδοσία 5V μεταξύ Vcc-Gnd καθώς και να δώσουμε σήμα (τάση) σε όποιο pin (IN1-8) θέλουμε για να ενεργοποιήσουμε το αντίστοιχο ρελέ. Κάθε είσοδος είναι οπτικά αλλά και ηλεκτρικά απομονωμένο με το κύκλωμα ισχύος. Αυτό σημαίνει ότι μπορούμε να ελέγξουμε απροβλημάτιστα οποιαδήποτε τάση στην επαφή του π.χ. 12-24-48-110-230 Volt συνεχές (DC) ή εναλλασσόμενο (AC).



Εικόνα 19. (Relay Module)

3.5 Αισθητήριο θερμοκρασίας LM35

Για τη μέτρηση της θερμοκρασίας θα χρησιμοποιήσουμε το αισθητήριο LM35 *Εικόνα 20*, το οποίο είναι εύκολο στη χρήση λόγω των χαρακτηριστικών του που δίνονται παρακάτω:

- Βαθμονομημένο κατευθείαν σε βαθμούς Κελσίου ($^{\circ}\text{C}$)
- Γραμμικό $+10\text{mV}/^{\circ}\text{C}$
- Εγγυημένη ακρίβεια 0.5°C στους 25°C
- Θερμοκρασία λειτουργίας -55°C έως 150°C
- Κατάλληλο για απομακρυσμένες εφαρμογές
- Χαμηλό κόστος
- Τάση λειτουργίας 4 έως 30 V
- Μικρή κατανάλωση ενέργειας λιγότερο από 60 μA
- Μικρή αυτοθέρμανση 0.08°C σε λειτουργία στον αέρα
- Μικρή διακύμανση γραμμικότητας $\pm 1/4^{\circ}\text{C}$
- Μικρή αντίσταση εξόδου 0.1 Ω για κάθε 1 mA φορτίου

Οι LM35 είναι αισθητήρες θερμοκρασίας ακριβείας σε μορφή ολοκληρωμένου κυκλώματος, με γραμμική τάση εξόδου ανάλογη με τη θερμοκρασία του αισθητήρα. Το LM35 έχει πλεονέκτημα έναντι των γραμμικών αισθητήρων θερμοκρασίας τα οποία είναι βαθμονομημένα στη κλίμακα Kelvin, καθώς δεν χρειάζεται να κάνουμε περίπλοκες μαθηματικές πράξεις ώστε να κάνουμε μετατροπή της κλίμακας Kelvin σε Celsius. Το LM35 δεν απαιτεί καμία εξωτερική βαθμονόμηση καθώς έρχεται βαθμονομημένο από το εργοστάσιο και οι τυπικές ακρίβειες που θα έχουμε είναι: $\pm 1/4^{\circ}\text{C}$ σε θερμοκρασία δωματίου και $\pm 3/4^{\circ}\text{C}$ για θερμοκρασίες -55°C έως 150°C .

Η μικρή σύνθετη αντίσταση εξόδου που παρουσιάζει, η γραμμική του έξοδος, και η ακριβής εγγενής βαθμονόμηση του LM35 το κάνει ιδιαίτερα εύκολο στη διασύνδεση με ένα κύκλωμα ανάγνωσης ή ελέγχου. Λόγω του ότι η ηλεκτρική κατανάλωση για την λειτουργία του LM35 είναι μόνο 60 μA , έχει πολύ χαμηλή αυτοθέρμια μικρότερη από $0,1^{\circ}\text{C}$ σε λειτουργία στον αέρα. Οι αισθητήρες LM35 λειτουργούν σε περιοχή θερμοκρασιών από -55°C έως 150°C , ενώ το LM35C έχει ονομαστική τιμή από -40°C έως 110°C (-10° με βελτιωμένη ακρίβεια). Το στοιχείο μέτρησης της θερμοκρασίας αποτελείται από μια αρχιτεκτονική δέλτα-V BE. Το στοιχείο ανίχνευσης θερμοκρασίας στη συνέχεια ρυθμίζεται από έναν ενισχυτή και παρέχεται στην έξοδο VOUT. Ο ενισχυτής έχει μια απλή βαθμίδα εξόδου κατηγορίας A με τυπική αντίσταση εξόδου 0,5 Ω .



Εικόνα 20. (LM 35 και σύνδεση ακροδεκτών του)

3.6 LDR (Light dependent resistance) Φωτοαντίσταση

Η φωτοαντίσταση (Light Dependent Resistance) είναι μια μεταβλητή αντίσταση. Η ωμική τιμή αντίστασης (Ω) της φωτοαντίστασης μειώνεται με την αύξηση της έντασης του φωτός και αυξάνεται με τη μείωση του φωτός που πέφτει επάνω της.



Εικόνα 21. (Light Dependent Resistance)

Οι φωτοαντίστασεις βρίσκουν χρήση σε κυκλώματα ανίχνευσης της φωτεινότητας παραδείγματος ένα κύκλωμα ενεργοποίησης νυχτερινών φώτων. Ο ημιαγωγός που χρησιμοποιείται για την κατασκευή της φωτοαντίστασης είναι υψηλής αντοχής. Στο σκοτάδι η τιμή της αντίστασης είναι της τάξεως αρκετών Megohms (MΩ), ενώ κάτω από έντονο φως (π.χ. ηλίου) μπορεί να έχει τιμή μερικών εκατοντάδων Ω.

Στην εικόνα παρακάτω βλέπουμε μια τυπική καμπύλη πώς μεταβάλλεται η τιμή μίας φωτοαντίστασης σε σχέση με τη φωτεινότητα.



Εικόνα 22. (Καμπύλη μεταξύ Ωμικής αντίστασης και φωτεινότητας)

Όταν η φωτεινότητα φως που πέφτει σε μια φωτοαντίσταση ξεπεράσει ένα ορισμένο όριο, τα φωτόνια που θα απορροφηθούν από τον ημιαγωγό θα δώσουν αρκετή ενέργεια στα ηλεκτρόνια που είναι δεσμευμένα ώστε αυτά να μεταβούν στη ζώνη αγωγιμότητας.

Τα ηλεκτρόνια που είναι στη ζώνη αγωγιμότητας, δηλαδή ελεύθερα, είναι αυτά που άγουν το ηλεκτρικό ρεύμα, μειώνοντας έτσι την ωμική αντίσταση. Η τιμές αντίστασης και η ευαισθησία μίας φωτοαντίστασης μπορεί να διαφέρουν αρκετά μεταξύ διαφορετικών αισθητήρων. Αυτό οφείλεται αφενός στα τεχνικά χαρακτηριστικά της εκάστοτε φωτοαντίστασης αλλά και στα υλικά με τα οποία θα κατασκευαστεί. Σημαντικό ρόλο παίζει η καθαρότητα του πυριτίου το οποίο είναι ο το υλικό του ημιαγωγού.

4. Προγραμματισμός Arduino Mega 2560

Ο προγραμματισμός του Arduino γίνεται εύκολα σε σχέση με άλλες αναπτυξιακές πλατφόρμες. Ο λόγος είναι ότι δεν απαιτείται ειδικός προγραμματιστής παρά μόνο ένα απλό καλώδιο USB και η εφαρμογή ArduinoIDE.

Το ArduinoIDE (integrated development environment) . Είναι μια cross platform εφαρμογή και μπορεί να εγκατασταθεί στα τρία δημοφιλέστερα λειτουργικά συστήματα Mac, Linux, Windows.

Το ArduinoIDE είναι βασισμένο σε JAVA και περιέχει τα εξής:

- Ένα απλό και πρακτικό περιβάλλον εργασίας στο οποίο μπορούμε να δημιουργήσουμε τα sketch (sketch στην ορολογία του Arduino είναι το πρόγραμμα ή κώδικας) μας. Έχει μάλιστα κάποιες χρωματικές λεπτομέρειες όταν γράφουμε κώδικα ώστε να είναι ποιο εύκολο το να ξεχωρίσουμε τις εντολές και να βρούμε πιθανά λάθη.
- αρκετά έτοιμα παραδείγματα από τα οποία μπορούμε να παραδειγματιστούμε ή να χρησιμοποιήσουμε κάποια κομμάτια τους ώστε να συνθέσουμε έναν δικό μας κώδικα.
- Έτοιμες βιβλιοθήκες που σκοπό έχουν τον εύκολο χειρισμό κάποιων εξωτερικών εξαρτημάτων. Αλλά και προέκταση της γλώσσας προγραμματισμού.
- Τον compiler για την μεταγλώττιση του sketch από τη γλώσσα wiring για να περαστεί στο Arduino sketch αλλά και τον έλεγχο του.
- Μια σειριακή κονσόλα από την οποία μπορούμε να παρατηρούμε τα δεδομένα που μας στέλνει μέσω του usb το arduino. Αλλά και να εισαγάγουμε κάποιες εντολές η μεταβλητές εάν θέλουμε. Όλα αυτά υπό την προϋπόθεση ότι ο κώδικας μας δίνει αυτή τη δυνατότητα, και φυσικά να έχουμε το Arduino συνδεδεμένο μέσω του usb.

4.1 Γλώσσα προγραμματισμού Wiring

Η γλώσσα προγραμματισμού για το Arduino είναι η wiring. Η wiring είναι μια παραλλαγή της C και της C++ σχεδιασμένη για την αρχιτεκτονική μικροελεγκτών όπως ο AVR της Atmega. Η wiring υποστηρίζει απόλυτα την γλώσσα C αλλά ενσωματώνει και κάποια χαρακτηριστικά της C++. Ως compiler χρησιμοποιείται ο AVR gcc και βασική βιβλιοθήκη η AVR libc. Λόγω του ότι η wiring είναι σχεδόν ίδια με τη C μπορούμε να χρησιμοποιήσουμε τις ίδιες βασικές εντολές,

συναρτήσεις, ίδια σύνταξη και τους ίδιους τύπους δεδομένων. Εκτός από αυτές όμως υπάρχουν κάποιες άλλες επιπλέον και ειδικά εντολές, σταθερές και συναρτήσεις, που χρησιμεύουν για τη σωστή λειτουργία του hardware του arduino. Παρακάτω εξηγούνται οι πιο σημαντικές.

Όρισμα	Είδος	Τύπος	Παράμετροι	Περιγραφή
LOW	Σταθερά	int	-	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
HIGH	Σταθερά	int	-	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
INPUT	Σταθερά	int	-	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
OUTPUT	Σταθερά	int	-	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
pinMode	Εντολή	-	(<i>pin, mode</i>)	Καθορίζει αν το συγκεκριμένο ψηφιακό <i>pin</i> θα είναι <i>pin</i> εισόδου ή <i>pin</i> εξόδου ανάλογα με την τιμή που δίνεται στην παράμετρο <i>mode</i> (INPUT ή OUTPUT αντίστοιχα).
digitalWrite	Εντολή	-	(<i>pin, pinstatus</i>)	Θέτει την κατάσταση <i>pinstatus</i> (HIGH ή LOW) στο συγκεκριμένο ψηφιακό <i>pin</i> .
digitalRead	Συνάρτηση	int	(<i>pin</i>)	Επιστρέφει την κατάσταση του συγκεκριμένου ψηφιακού <i>pin</i> (0 για LOW και 1 για HIGH) εφόσον αυτό είναι <i>pin</i> εισόδου.
analogReference	Εντολή	-	(<i>type</i>)	Δέχεται τις τιμές DEFAULT, INTERNAL ή EXTERNAL στην παράμετρο <i>type</i> για να καθορίσει την τάση αναφοράς (<i>Vref</i>) των αναλογικών εισόδων (5V, 1.1V ή η εξωτερική τάση με την οποία τροφοδοτείται το <i>pin</i> AREF αντίστοιχα)
analogRead	Συνάρτηση	int	(<i>pin</i>)	Επιστρέφει έναν ακέραιο από 0 έως 1023, ανάλογα με την τάση που τροφοδοτείται το συγκεκριμένο <i>pin</i> αναλογικής εισόδου στην κλίμακα 0 ως <i>Vref</i> .
analogWrite	Εντολή	-	(<i>pin, value</i>)	Θέτει το συγκεκριμένο ψηφιακό <i>pin</i> σε κατάσταση ψευδοαναλογικής εξόδου (PWM). Η παράμετρος <i>value</i> καθορίζει το πλάτος του παλμού σε σχέση με την περίοδο του παραγόμενου σήματος στην κλίμακα από 0 ως 255 (π.χ. με <i>value</i> 127, το πλάτος του παλμού είναι ίσο με μισή περίοδο).
millis	Συνάρτηση	unsigned long	()	Μετρητής που επιστρέφει το χρονικό διάστημα σε ms από την στιγμή που άρχισε η εκτέλεση του προγράμματος. Λάβετε υπόψη ότι λόγω του τύπου μεταβλητής (unsigned long δηλ. 32bit) θα γίνει overflow σε 2 [^] 32ms δηλαδή περίπου σε 50 μέρες, οπότε ο μετρητής θα ξεκινήσει πάλι από το μηδέν.

delay	Εντολή	-	(<i>time</i>)	Σταματά προσωρινά τη ροή του προγράμματος για <i>time</i> ms. Η παράμετρος <i>time</i> είναι unsigned long (από 0 ως 2 ³²). Σημειώστε ότι παρά την προσωρινή παύση, συναρτήσεις των οποίων η εκτέλεση ενεργοποιείται από interrupt θα εκτελεστούν κανονικά κατά την διάρκεια μιας delay.
attachInterrupt	Εντολή	-	(<i>interrupt,function,triggermode</i>)	Θέτει σε λειτουργία το συγκεκριμένο <i>interrupt</i> , ώστε να ενεργοποιεί την συνάρτηση <i>function</i> , κάθε φορά που ικανοποιείται η συνθήκη που ορίζεται από την παράμετρο <i>triggermode</i> : LOW (ενεργοποίηση όταν η κατάσταση του pin που αντιστοιχεί στο συγκεκριμένο interrupt γίνει LOW) RISING (όταν από LOW γίνει HIGH) FALLING (όταν από HIGH γίνει LOW) CHANGE (όταν αλλάξει κατάσταση γενικά)
detachInterrupt	Εντολή	-	(<i>interrupt</i>)	Απενεργοποιεί το συγκεκριμένο <i>interrupt</i> .
noInterrupts	Εντολή	-	()	Σταματά προσωρινά την λειτουργία όλων των interrupt
interrupts	Εντολή	-	()	Επαναφέρει την λειτουργία των interrupt που διακόπηκε προσωρινά από μια εντολή noInterrupts.
Serial.begin	Μέθοδος κλάσης	-	(<i>datarate</i>)	Θέτει τον ρυθμό μεταφοράς δεδομένων του σειριακού interface (σε baud)
Serial.println	Μέθοδος κλάσης	-	(<i>data</i>)	Διοχετεύει τα δεδομένα <i>data</i> για αποστολή μέσω του σειριακού interface. Η παράμετρος <i>data</i> μπορεί να είναι είτε αριθμός είτε αλφαριθμητικό.

Δομές ελέγχου ροής

1. *if* (δομή ελέγχου μίας συνθήκης)
2. *if ... else* (δομή ελέγχου πολλαπλών συνθηκών)
3. *or* (δομή επαναληπτικού ελέγχου συνθήκης)
4. *while* (δομή επαναληπτικού ελέγχου συνθήκης)

5. *do ... while* (δομή επαναληπτικού ελέγχου συνθήκης)
6. *switch ... case* (δομή ελέγχου περιπτώσεων)
7. *break* (εντολή διακοπής μιας επαναληπτικής δομής)
8. *continue* (εντολή παράλειψης της τρέχουσας επανάληψης)
9. *return* (εντολή επιστροφής από μία συνάρτηση)
10. *goto* (εντολή μετάβασης σε κάποιο σημείο του κώδικα)

Αριθμητικοί τελεστές

1. = (τελεστής εκχώρησης)
2. + (τελεστής πρόσθεσης)
3. - (τελεστής αφαίρεσης)
4. * (τελεστής πολλαπλασιασμού)
5. / (τελεστής διαίρεσης)
6. % (τελεστής υπόλοιπου ακεραίας διαίρεσης)

Λογικοί τελεστές

1. && (λογική σύζευξη)
2. || (λογική διάζευξη)
3. ! (λογική άρνηση)

Διαδικοί τελεστές

1. & (δυναμική σύζευξη)
2. | (δυναμική διάζευξη)
3. ^ (δυναμική αποκλειστική διάζευξη)
4. ~ (δυναμική άρνηση)

5. << (δυναδική αριστερή ολίσθηση)
6. >> (δυναδική δεξιά ολίσθηση)

Τελεστές αύξησης και μείωσης

1. ++ (αύξηση κατά μία ακέραιη μονάδα)
2. -- (μείωση κατά μία ακέραιη μονάδα)

Σύνθετοι τελεστές

1. +=, -=, *=, /=, %= (σύνθετοι αριθμητικοί τελεστές)
2. &=, |=, ^=, ~=, <<=, >>= (σύνθετοι δυναδικοί τελεστές)

Τελεστές σύγκρισης

1. == (ισότητα)
2. != (ανισότητα)
3. < (μικρότερο)
4. > (μεγαλύτερο)
5. <= (μικρότερο ή ίσο)
6. >= (μεγαλύτερο ή ίσο)

Τελεστές δεικτών

1. * (τελεστής απόκτησης περιεχομένου)
2. & (τελεστής απόκτησης διεύθυνσης)

Σταθερές

1. HIGH (τιμή υψηλής στάθμης για μία επαφή εισόδου ή εξόδου)
2. LOW (τιμή χαμηλής στάθμης για μία επαφή εισόδου ή εξόδου)
3. false (λογικό επίπεδο ψεύδους σε μία συνθήκη)
4. true (λογικό επίπεδο αλήθειας σε μία συνθήκη)
5. INPUT (χρησιμοποιείται για τον ορισμό μίας επαφής ως είσοδο)
6. OUTPUT (χρησιμοποιείται για τον ορισμό μίας επαφής ως έξοδο)
7. A0, ..., A5 (συμβολοσταθερές για τις αναλογικές επαφές εισόδου)

Τύποι δεδομένων

1. boolean (λογική δυαδική τιμή)
2. char (προσημασμένος χαρακτήρας 8 ψηφίων)
3. unsigned char (μη προσημασμένος χαρακτήρας 8 ψηφίων)
4. byte (μη προσημασμένος χαρακτήρας 8 ψηφίων)
5. int (προσημασμένος ακέραιος αριθμός 16 ψηφίων)
6. unsigned int (μη προσημασμένος ακέραιος αριθμός 16 ψηφίων)
7. word (μη προσημασμένος ακέραιος αριθμός 16 ψηφίων)
8. long (προσημασμένος ακέραιος αριθμός 32 ψηφίων)
9. unsigned long (μη προσημασμένος ακέραιος αριθμός 32 ψηφίων)
10. double float, (αριθμός κινητής υποδιαστολής)
11. String (αντικείμενο αλφαριθμητικού με χρήσιμες μεθόδους)

Συναρτήσεις μετατροπής τύπων

1. char(), byte()
2. int(), word(), long()
3. float(), double()

Συναρτήσεις εισόδου και εξόδου

1. pinMode() (ορίζει μια επαφή ως είσοδο ή έξοδο)

Συναρτήσεις ψηφιακής εισόδου και εξόδου

1. digitalWrite() (γράφει σε μία ψηφιακή επαφή εξόδου)
2. digitalRead() (διαβάζει από μία ψηφιακή επαφή εισόδου)

Συναρτήσεις αναλογικής εισόδου και εξόδου

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

Προηγμένες συναρτήσεις εισόδου και εξόδου

1. tone() (παράγει ένα τετραγωνικό σήμα ορισμένης συχνότητας)
2. noTone() (διακόπτει την παραγωγή τετραγωνικών σημάτων)
3. shiftOut() (ολισθαίνει τα ψηφία μιας τιμής σε μία επαφή εξόδου)
4. pulseIn() (επιστρέφει τη διάρκεια σε μs ενός παλμού HIGH ή LOW)
- 5.

Συναρτήσεις χρόνου

1. millis() (διάρκεια εκτέλεσης του προγράμματος σε ms)
2. micros() (διάρκεια εκτέλεσης του προγράμματος σε μs)
3. delay() (παύση προγράμματος - η διάρκεια δίδεται σε ms)
4. delayMicroseconds() (παύση προγράμματος - η διάρκεια δίδεται σε μs)

Μαθηματικές και Τριγωνομετρικές συναρτήσεις

1. `max()` (βρίσκει τον μεγαλύτερο ανάμεσα σε δύο αριθμούς)
2. `min()` (βρίσκει τον μικρότερο ανάμεσα σε δύο αριθμούς)
3. `abs()` (επιστρέφει την απόλυτη τιμή ενός αριθμού)
4. `constrain()` (ελέγχει για υπερχείλιση ή υποχείλιση ορίων)
5. `map()` (πραγματοποιεί γραμμικό μετασχηματισμό ορίων)
6. `pow()` (επιστρέφει το αποτέλεσμα μίας δύναμης)
7. `sqrt()` (επιστρέφει τη ρίζα ενός αριθμού)
8. `sin()` (υπολογίζει το ημίτονο ενός αριθμού)
9. `cos()` (υπολογίζει το συνημίτονο ενός αριθμού)
10. `tan()` (υπολογίζει την εφαπτομένη ενός αριθμού)

Συναρτήσεις γεννήτριας ψευδοτυχαίων αριθμών

1. `random()` (δίδεται ένας νέος αριθμός από την γεννήτρια)
2. `randomSeed()` (θέτει τον σπόρο της γεννήτριας παραγωγής)

Συναρτήσεις επεξεργασίας δυαδικών αριθμών

1. `lowByte()` (επιστρέφει το δεξιότερο byte μίας μεταβλητής)
2. `highByte()` (επιστρέφει το αριστερότερο byte μίας μεταβλητής)
3. `bitRead()` (διαβάζει ένα συγκεκριμένο ψηφίο μίας μεταβλητής)
4. `bitWrite()` (γράφει σε ένα συγκεκριμένο ψηφίο μιας μεταβλητής)
5. `bitSet()` (γράφει την τιμή 1 σε κάποιο ψηφίο μίας μεταβλητής)
6. `bitClear()` (γράφει την τιμή 0 σε κάποιο ψηφίο μιας μεταβλητής)
7. `bit()` (υπολογίζει μία συγκεκριμένη δύναμη με βάση το 2)

Συναρτήσεις χρήσης ρουτινών εξυπηρέτησης διακοπών

1. `attachInterrupt()` (ενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής)
2. `detachInterrupt()` (απενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής)

Συναρτήσεις ενεργοποίησης και απενεργοποίησης διακοπών

1. `interrupts()` (ενεργοποιεί τα σήματα διακοπής)
2. `noInterrupts()` (απενεργοποιεί τα σήματα διακοπής)

Υποστήριξη σειριακής επικοινωνίας

1. `Serial` (Χρησιμοποιείται για την λειτουργία των σειριακών επικοινωνιών)

4.2 Δομή κώδικα

Ο κώδικας χωρίζεται σε τρία βασικά κομμάτια:

- Αρχικοποίηση μεταβλητών. Εδώ γίνεται η δήλωση όλων των μεταβλητών του προγράμματος μας π.χ. `const int buttonPin1 = 2` εδώ δηλώνουμε το αναλογικό pin 2 ότι είναι συνδεδεμένο το button. `int buttonState = 0;` εδώ δηλώνουμε ότι η αρχική τιμή θα είναι μηδέν.
- `Void setup();` είναι ρουτίνα η οποία τρέχει μόνο μια φορά κατά την εκκίνηση του Arduino και αρχικοποιεί κάποιες τιμές όπως π.χ. `pinMode (ledPin, OUTPUT);` εδώ δηλώνουμε πως στο pin που έχουμε συνδέσει το led θα είναι έξοδος.
- `Void loop();` είναι η ρουτίνα η οποία τρέχει συνεχόμενα και ασταμάτητα για όσο είναι εν λειτουργία το arduino. Μέσα σε αυτή τη ρουτίνα γράφουμε όλο το κώδικα που σύμφωνα με αυτό θα αλληλεπιδρούν οι είσοδοι με τις εξόδους. Για παράδειγμα εάν πατήσουμε το button θα ανάψει το led στο pin13.



Εικόνα 23. (Γραφικό περιβάλλον Arduino IDE)

5. Πειραματική διαδικασία

Κατά την πειραματική διαδικασία αρχικά δοκιμάσαμε το κάθε υλικό ξεχωριστά ώστε να είμαστε σίγουροι για την σωστή λειτουργία τους.

5.1 Βαθμονόμηση αισθητήρα θερμοκρασίας

Αρχικά ξεκινήσαμε από το αισθητήριο θερμοκρασίας με το οποίο πήραμε κάποιες

δοκιμαστικές μετρήσεις σε διάφορες θερμοκρασίες και συγκρίνοντας τες με το θερμόμετρό μας, διαπιστώσαμε ότι δεν παρουσίαζε απόκλιση.

5.2 Βαθμονόμηση αισθητήρα φωτός

Στη συνέχεια δοκιμάσαμε τον αισθητήρα φωτός LDR, για τον οποίο δεν κάναμε βαθμονόμηση καθώς δεν χρειαζόμαστε την ακριβή τιμή της φωτεινότητας. Δηλαδή μας είναι αρκετό να γνωρίσουμε ότι έχει νυχτώσει. Ο τρόπος που προσδιορίστηκε η τιμή αυτή ήταν να πάρουμε μετρήσεις φωτεινότητας την ημέρα και ιδιαίτερα το απόγευμα και το βράδυ.

5.3 Βαθμονόμηση αισθητήρα υγρασίας εδάφους

Η βαθμονόμηση του αισθητήρα υγρασίας του εδάφους ήταν η πιο περίπλοκη διαδικασία από όλες καθώς έπρεπε να γίνει υπό πραγματικές συνθήκες για να έχουμε τα καλύτερα αποτελέσματα. Για αυτό το λόγο έγινε σε χωράφι πριν ποτιστεί, κατά τη διάρκεια ως την ολοκλήρωση του. Εδώ πρέπει να τονίσουμε ότι έχει μεγάλη σημασία το σημείο τοποθέτησής του αισθητήρα. Ο πορτοκαλεώνας όπου έγιναν οι μετρήσεις αρδεύεται με σύστημα υδρονέφωσης ακριβώς δίπλα από το κορμό του δέντρου. Έτσι ο αισθητήρας δεν μπορεί να τοποθετηθεί κοντά στο δέντρο άλλα ούτε επιφανειακά. Για να έχουμε σωστό αποτέλεσμα τοποθετήσαμε τον αισθητήρα περίπου 2 μέτρα μακριά από το κορμό και το μπέκ υδρονέφωσης. Καθώς το επιφανειακό χώμα αποκτά γρήγορα υγρασία σκάψαμε και τοποθετήσαμε τον αισθητήρα στα 20 με 25 εκατοστά του μέτρου κάτω από την επιφάνεια. Η τιμή που πήραμε από τον αισθητήρα άρχισε να μεταβάλετε λίγο πριν το χρονικό όριο που μας υπέδειξε ο αγρότης ότι χρειάζεται για την άρδευση του πορτοκαλεώνα. Ο τύπος του εδάφους διαδραματίζει σημαντικό ρόλο καθώς η ηλεκτρική αγωγιμότητα διαφέρει από περιοχή σε περιοχή.

Το έδαφος αποτελείται από τρία βασικά μέρη. Τα στερεά σωματίδια, το νερό που περιβάλλει τα στερεά σωματίδια και τον αέρα που γεμίζει τους μεγάλους πόρους ανάμεσα τους. Κάθε ένα υλικό από τα τρία έχει διαφορετικές ηλεκτρικές ιδιότητες. Ο αέρας είναι μονωτής, δηλαδή κακός αγωγός του ηλεκτρισμού, το νερό, που είναι καλός αγωγός του ηλεκτρισμού, και το στερεό μέρος που είναι μέτριος αγωγός του ηλεκτρισμού, δηλαδή έχει ενδιάμεσες ιδιότητες από τον αέρα και το νερό.

Η σύσταση των καλλιεργούμενων εδαφών είναι ένα μείγμα στερεών σωματιδίων σε διάφορες αναλογίες. Τα σωματίδια διακρίνονται σε τρεις κατηγορίες ανάλογα με το μέγεθός τους. Τον άργιλο ο οποίος είναι σωματίδια πολύ μικρού μεγέθους (κάποια χιλιοστά του χιλιοστού). Την άμμο η οποία είναι σωματίδια της τάξεως των χιλιοστών καθώς και τα σωματίδια μεγέθους ενδιάμεσα από την άμμο και τον άργιλο που ονομάζεται ιλύς.

Τα καλλιεργούμενα εδάφη περιέχουν και τα τρία υλικά σε διαφορετικές αναλογίες. Βαριά χαρακτηρίζονται τα εδάφη που επικρατούν τα λεπτόκοκκα υλικά (άργιλος κυρίως αλλά και ιλύς), ελαφρά χαρακτηρίζουμε αυτά που επικρατεί άμμος. Οι πόροι μεταξύ των στερεών σωματιδίων στα λεπτόκοκκα εδάφη είναι πολύ μικρού μεγέθους, αυτό έχει ως αποτέλεσμα να είναι γεμάτοι με νερό. Οι δε μεγαλύτεροι πόροι έχουν μέσα τους αέρα αλλά είναι λιγότεροι σε ποσοστό. Έτσι τα λεπτόκοκκα εδάφη έχουμε μεγαλύτερη ηλεκτρική αγωγιμότητα σε αντίθεση με τα αδρόκοκκα τα οποία έχουν μικρότερη.

Η ηλεκτρική αγωγιμότητα του εδάφους επηρεάζεται και από άλλους παράγοντες όπως την περιεκτικότητα αυτών και του νερού σε ιόντα (άλατα). Τα αλατούχα εδάφη έχουν μεγάλη ηλεκτρική αγωγιμότητα. Οι οργανικές ουσίες που περιέχουν οι καλλιεργήσιμες εκτάσεις αλλά ακόμα και η θερμοκρασία του εδάφους επηρεάζουν και αυτά με τη σειρά τους την ηλεκτρική αγωγιμότητα.

6. Πίνακας αυτοματισμού αντλίας

Ο πίνακας αυτοματισμού της αντλίας δεν θα μας απασχολήσει ιδιαίτερα καθώς θα είναι κάποιος που προϋπάρχει και εμείς τον αναβαθμίζουμε με το σύστημα μας. Καθώς το σύστημα μας αλληλεπιδρά με κάποια μέρη του πίνακα θα πρέπει να γνωρίζουμε ποια είναι αυτά και ποια η λειτουργία τους.

Σε ένα πίνακα εκκίνησης με διάταξη Αστέρα -Τριγώνου (Υ-Δ) κινητήρα βραχυκυκλωμένου δρομέα (αντλίας) θα βρούμε τα εξής:

- Ερμάριο πίνακα το οποίο μπορεί να είναι μεταλλικό ή από συνθετικό υλικό (Πολυεστερικό, PVC κ.α.)
- Γενικός διακόπτης τροφοδοσίας
- Γενικές ασφάλειες (μικροαυτόματοι ή τήξεως)
- Επιμέρους ασφάλειες για τροφοδοσία κυκλωμάτων φωτισμού, πριζών κ.α.
- Επιτηρητή τάσης
- Επιτηρητής ξηράς λειτουργίας
- Ρελέ ισχύος που συνθέτουν το κύκλωμα Υ-Δ
- Θερμικό προστασίας ηλεκτροκινητήρα
- Χρονικό μεταγωγής Υ-Δ
- Κλέμενες σύνδεσης αγωγών στο πίνακα (αγωγοί τροφοδοσίας πίνακα, ηλεκτρικού κινητήρα, κυκλώματα φωτισμού και πριζών κ.α.)
- Ενδεικτικές λυχνίες για παρουσία τάσης, σφάλματος θερμικού κ.α.
- Επιλογικός διακόπτης 1-0-2 (Χειροκίνητο - 0 – Αυτόματο)
- Ψηφιακός θερμοστάτης μέτρησης θερμοκρασίας
- Βοηθητικά ρελέ
- Πλαστικά κανάλια για την όδευση και τακτοποίηση των καλωδίων
- Ράγες στήριξης ραγούλικού (ρελέ, μικροαυτόματοι, Κλέμενες σύνδεσης κ.α.)
- Όργανο μέτρησης τάσης
- Όργανο μέτρησης έντασης ρεύματος



Εικόνα 24. (Ενδεικτικός πίνακας εκκίνησης κινητήρα Υ-Δ)

Στο σύστημα μας θα συμπεριλάβουμε τις εντολές από τον επιτηρητή τάσης και το θερμικό προστασίας του ηλεκτροκινητήρα. Αυτό μας επιτρέπει να ενεργήσουμε κατάλληλα σε αυτές τις δύο περιπτώσεις.

6.1 Επιτηρητής τάσης

Ο επιτηρητής τάσης είναι μια διάταξη η οποία ελέγχει τη τιμή τάσης των τριών φάσεων του δικτύου, την ορθή διαδοχή των φάσεων, την απώλεια κάποιας φάσης και την υπέρταση και υπόταση στα όρια που εμείς έχουμε ρυθμίσει. Στην εικόνα 24 βλέπουμε τον επιτηρητή τάσης PT 431 και την εικόνα 25 βλέπουμε το τρόπο συνδεσμολογίας του.

Ο επιτηρητής τάσης PT 431 διαθέτει μία μεταγωγική ξηρή επαφή δηλαδή χωρίς τάση, η οποία έχει το κοινό άκρο μια Normally Closed και μια Normally Open επαφή που αλλάζει κατάσταση. Οι ξηρές επαφές χρησιμοποιούνται γιατί στα κυκλώματα αυτοματισμού μπορεί να έχουμε διαφορετικές τάσεις όπως 230VAC, 48VAC, 24VAC, 24VDC, 12VAC, 12VDC κ.α.



Εικόνα 24. (Επιτηρητής τάσης PT431)

Εικόνα 25. (Συνδεσμολογία PT431)

6.2 Θερμικό προστασίας

Το θερμικό είναι διάταξη που προστατεύει έναν ηλεκτροκινητήρα από υπερφόρτιση. Η υπερφόρτιση αυτή μπορεί να δημιουργηθεί όταν η ισχύς του φορτίου του κινητήρα είναι μεγαλύτερη από την ονομαστική του ισχύ , όταν ο κινητήρας τροφοδοτείται μονό από τις δυο φάσεις , όταν είναι τριφασικός , όταν η τάση του δικτύου τροφοδοσίας είναι μικρότερη από την ονομαστική ή όταν ο άξονας του κινητήρα είναι μπλοκαρισμένος. Τότε το ρεύμα αυξάνεται και προκαλεί υπερθέρμανση των τυλιγμάτων των μονώσεων και γενικά στο σώμα του κινητήρα για αυτό η λειτουργία του σε τέτοιες καταστάσεις πρέπει να σταματήσει πριν υποστεί βλάβη ο κινητήρας.

Στο θερμικό ρυθμίζουμε εμείς το μέγιστο επιτρεπτό ρεύμα που θέλουμε να περάσει ανάλογα με τον ονομαστικό ρεύμα λειτουργίας του κινητήρα.

Οι βοηθητικές επαφές του θερμικού 95-96 (Normally Closed) και 97-98 (Normally Open) είναι σε κατάσταση ηρεμίας .Η επαφή 95-96 συνδέεται σε σειρά στο βοηθητικό κύκλωμα. Μόλις το θερμικό ενεργοποιηθεί η επαφή 95-96 ανοίγει και τότε σταματάει η λειτουργία του κινητήρα, η επαφή 97-98 κλείνει (η επαφή αυτή χρησιμοποιείται συνήθως σαν επαφή σήμανσης). Σε παλαιότερης κατασκευής θερμικά συναντάμε μια μεταγωγική επαφή 95-96/98 η οποία σε θέση ηρεμίας είναι στην θέση 95-96 και όταν το θερμικό ενεργοποιηθεί μετακινείται στην θέση 95-98.

•

Εικόνα 26. (Θερμικό προστασίας Schneider Electric)

6.3 Βοηθητικά ρελέ

Τα βοηθητικά ρελέ είναι ρελέ τα οποία οι επαφές τους δεν μπορούν να διαχειριστούν μεγάλα ρεύματα όπως τα ρελέ ισχύος που χρησιμοποιούνται στο κύκλωμα Υ-Δ. Η χρήση τους είναι σε κυκλώματα ελέγχου. Η κατασκευή τους είναι παρόμοια με των ρελέ ισχύος δηλαδή αποτελούνται από το πηνίο και τις επαφές Normally Open και Normally Closed. Το κάθε ένα μπορεί να έχει 1,2, 4 ή και παραπάνω επαφές. Το ρεύμα που μπορούν να διαχειριστούν οι επαφές τους είναι συνήθως μέχρι 6 ή 10Α για ωμικά φορτία και η τάση τροφοδοσίας του πηνίου τους μπορεί να είναι AC είτε DC.

Στην εικόνα 27 βλέπουμε ένα βοηθητικό ρελέ της Carlo Cavazzi το οποίο έχει δύο μεταγωγικές ξηρές επαφές των 10Α η κάθε μια, η τάση λειτουργίας του πηνίου είναι στα 24VDC και ενσωματωμένη ενδεικτική LED λυχνία. Στα πολλά βοηθητικά ρελέ βρίσκουμε και ένα μηχανισμό για τον χειροκίνητο έλεγχο των επαφών, στο συγκεκριμένο είναι το μπλε εξάρτημα επάνω αριστερά το οποίο εάν το ανασηκώσουμε αλλάζουν κατάσταση οι επαφές όπως ακριβώς και με την τροφοδοσία του πηνίου του. Αυτή η λειτουργία υπάρχει για απασφαλίση και εύρεση βλαβών σε κυκλώματα αυτοματισμού.



Εικόνα 27. (Βοηθητικό ρελέ Carlo Cavazzi RMI 2 10 24VDC)

Συνήθης τάσης τροφοδοσίας βοηθητικών ρελέ :

- 5,9,12,24,48 VDC
- 12,24,48,230 VAC

6.4 Ψηφιακός θερμοστάτης

Ο ψηφιακός θερμοστάτης είναι όργανο που μετρά την θερμοκρασία μέσω ενός PTC thermistor που τοποθετείτε στο χώρο που θέλουμε να ελέγξουμε την θερμοκρασία. Ο θερμοστάτης αυτός διαθέτει μια μεταγωγική ξηρή επαφή και αναλόγως τη θερμοκρασία που το έχουμε ρυθμίσει και τη θερμοκρασία που μετρά το αισθητήριο του, αλλάζει κατάσταση στην επαφή. Μπορεί να διαθέτει πάνω από μια επαφές, επαφή για Alarm κ.α. Η χρήση του μπορεί να είναι για θέρμανση αλλά και για ψύξη.

Στην εικόνα 28 βλέπουμε τον LT102N της GEMO ο οποίος μπορεί να λειτουργήσει σε θερμοκρασίες -19.9 έως 99.9 βαθμούς Κελσίου. Διαθέτει μια μεταγωγική επαφή καθώς και μια επαφή για Alarm. Η τάση τροφοδοσίας του ψηφιακού θερμοστάτη για την λειτουργία του είναι 230VAC.

—

Εικόνα 28. (Ψηφιακός θερμοστάτης GEMO LT102N)

6.5 Απαιτούμενες αλλαγές στο πίνακα ελέγχου

Για να λειτουργήσει το σύστημα μας στον ήδη υπάρχων πίνακα θα πρέπει να γίνουν κάποιες αλλαγές. Αρχικά θα πρέπει να πάρουμε εισόδους από το θερμικό προστασίας και τον επιτηρητή τάσης. Επίσης για τον αυτόματο έλεγχο από το σύστημα μας θα πρέπει να δώσουμε με κάποιο τρόπο εντολή για εκκίνηση της αντλίας.

Για αυτό το σκοπό θα χρησιμοποιήσουμε βοηθητικά ρελέ. Λόγο του ότι συνήθως το κύκλωμα αυτοματισμού σε τέτοιες εφαρμογές είναι 230VAC και το σύστημα μας λειτουργεί με 5 VDC δεν μπορούμε να δώσουμε εντολή από το θερμικό κατευθείαν στο Arduino.

Για τον επιτηρητή τάσης θα τοποθετήσουμε ένα βοηθητικό ρελέ με πηνίο 230VAC τροφοδοτώντας το από την επαφή 3 (Normaly Open) η οποία δίνει και στο κύκλωμα ισχύος. Από τις βοηθητικές επαφές του ρελέ θα χρησιμοποιήσουμε μια Normaly Closed που θα την συνδέσουμε στην είσοδο του Arduino και στο σταθερό της Common θα τροφοδοτήσουμε με +5VDC.

Όπως για τον επιτηρητή τάσης έτσι και για το θερμικό προστασίας του κινητήρα θα τοποθετήσουμε ένα βοηθητικό ρελέ. Η σύνδεση θα γίνει στην επαφή 98 από την οποία παίρνει τροφοδοσία και η ενδεικτική λυχνία πτώσης θερμικού. Από τις βοηθητικές επαφές του ρελέ θα χρησιμοποιήσουμε μια Normaly Closed που θα την συνδέσουμε στην είσοδο του Arduino και στο σταθερό της Common θα τροφοδοτήσουμε με +5VDC.

Για την εκκίνηση του κινητήρα θα χρησιμοποιήσουμε μια επαφή από το Relay Module του Arduino. Χωρίς να κάνουμε εκτεταμένες αλλαγές στο κύκλωμα αυτοματισμού του πίνακα μπορούμε να το συνδέσουμε στη θέση του ψηφιακού θερμοστάτη. Καθώς στον κλασικό αυτοματισμό αυτός έλεγχε την λειτουργία του όταν η θερμοκρασία ήταν κάτω από το όριο. Τώρα αυτό θα γίνεται από το σύστημα μας και για την αντιπαγετική προστασία αλλά και για την άρδευση. Εδώ πρέπει να τονίσουμε ότι ο πίνακας έχει επιλογικό διακόπτη (Χειροκίνητο - 0 - Αυτόματο). Κατά την χειροκίνητη λειτουργία ο κινητήρας θα δουλεύει ασταμάτητα και κατά την αυτόματη σύμφωνα με το σύστημά μας.

7. Λειτουργία συστήματος - διαδικασία αυτοματισμού

Ο τρόπος λειτουργίας του συστήματος χωρίζεται σε δύο μέρη, την αντιπαγετική προστασία και τον έλεγχο για άρδευση. Επιπλέον υπάρχει έλεγχος για την πτώση θερμοκού στην αντλία αλλά και ύπαρξη τάσης στο πίνακα ελέγχου της αντλίας.

Διαδικασία 1. (Αντιπαγετική προστασία)

Το Arduino ελέγχει συνεχώς τη θερμοκρασία του περιβάλλοντος μέσω του LM35 εάν η θερμοκρασία είναι κάτω από 1.5 βαθμό Κελσίου. Εάν η θερμοκρασία είναι χαμηλότερη τότε θα μας ειδοποιήσει με γραπτό μήνυμα **"XAMHLH THERMOKRASIA ENARKSH ANTIPAGETIKOU"** και θα θέσει σε λειτουργία την αντλία έως ότου η θερμοκρασία ξεπεράσει τους 2.5 βαθμούς Κελσίου. Τότε θα ειδοποιηθούμε ξανά με μήνυμα **"THERMOKRASIA SE KANONIKA EPIPEDA STOP ANTIPAGETIKOU"** ότι η διαδικασία έχει σταματήσει.

Διαδικασία 2. (Έλεγχος άρδευσης)

Το Arduino ελέγχει συνεχώς την υγρασία στο χώμα μέσω του Moisture Sensor. Εάν η υγρασία πέσει κάτω από το επιτρεπτό όριο τότε θα μας ειδοποιήσει με γραπτό μήνυμα **"XSERO XWMA ANAGKH GIA POTISMA"** ότι η υγρασία βρίσκεται σε χαμηλό επίπεδο και απαιτείται άρδευση. Η αντλία δεν θα εκκινήσει χωρίς να δώσουμε εντολή μέσω γραπτού μηνύματος.

Η λειτουργία της αντλίας θα ξεκινήσει εφόσον λάβει μήνυμα επιβεβαίωσης **"START"** και η ένταση του φωτός πολύ χαμηλή δηλαδή νύχτα. Η ένταση του φωτός για το εάν είναι ημέρα η νύχτα ελέγχεται από την LDR.

Όταν η υγρασία φτάσει στο επίπεδο που έχουμε ορίσει ως ποτισμένο τότε θα σταματήσει η λειτουργία της αντλίας και θα μας ειδοποιήσει με το μήνυμα **"KINHTHRAS STOP XWRAFI POTISMENO"**.

Ο λόγος που γίνεται αυτό είναι για οικονομία στο νερό που απαιτείται κατά τη διαδικασία της άρδευσης καθώς και προστασία των δέντρων στην απότομη μεταβολή θερμοκρασίας.

Εάν η άρδευση πραγματοποιείται κατά την διάρκεια της ημέρας υπάρχουν απώλειες λόγω εξάτμισης του νερού λόγω των υψηλότερων θερμοκρασιών που επικρατούν τους καλοκαιρινούς μήνες.

Διαδικασία 3. (Έλεγχος πτώσης θερμοκού αλλά και παροχής πίνακα αυτοματισμού αντλίας)

Εάν υπάρξει πτώση θερμοκού θα μας αποσταλεί το μήνυμα **"THERMIKO SFALMA"** για να μας ειδοποιήσει. Αυτό είναι σημαντικό ειδικά κατά την αντιπαγετική λειτουργία του συστήματος. Τότε είναι απαραίτητο να προβούμε στον έλεγχο του γιατί υπάρχει θερμικό σφάλμα. Σε αυτή την περίπτωση δεν υπάρχει ανάγκη για ενημέρωση με μήνυμα καθώς θα πρέπει να πάμε οι ίδιοι στην εγκατάσταση για έλεγχο.

Στη περίπτωση που υπάρχει σφάλμα στο δίκτυο τροφοδοσίας και ο επιτηρητής τάσης θα διακόψει την τροφοδοσία του κυκλώματος αυτοματισμού του πίνακα θα μας αποσταλεί το μήνυμα "DIAKOPH REYMATOS" . Εάν η τάση του δικτύου επανέλθει στα κανονικά επίπεδα και ο επιτηρητής τάσης τροφοδοτήσει το κύκλωμα αυτοματισμού θα μας έρθει μήνυμα "EPANAFORA REYMATOS TASH" το οποίο θα μας πληροφορεί για την επαναφορά του δικτύου τροφοδοσίας.

8. Κώδικας προγράμματος

```
#include <SPI.h>

// Elegxos 3 relay mesw sms + temperature sensor + LDR +soil humidity sensor

String msg = String("");
byte i;
byte present = 0;
byte data[12];
float celsius;
byte addr[8]; // [0]...[7]
int IN1_State = 0;    // HIGH h LOW
int IN2_State = 0;    // HIGH h LOW
int IN3_State = 0;
int IN4_State =0;
int sensorPin1 = A0; //pin gia to ldr
int sensorValue1 = 0; //arxikopoihsh timis sto mhden.
int LM35_celsius=0;
int sum=0;
int t1=0;
int t2=0;
int sensorPin2 = A1; //pin gia to moisture sensor
int sensorValue2 = 0; //arxikopoihsh timis sto mhden
int potizwgiapago = 0;
int LM35_sensor = 0;
int thelopotisma = 0;
int K=0;           // 0 h 1
int L=0;
```



```
int F=0; // EPAFH THERMIKOU
```

```
int SmsContentFlag = 0;
```

```
//ELEGXOS PINS TON RELAY
```

```
int relay_1=48;
```

```
int relay_2=50;
```

```
int relay_3=52;
```

```
void setup()
```

```
{
```

```
  Serial.begin(19200); // the GPRS baud rate
```

```
  pinMode( 48, OUTPUT ); // Relay_1_k1_kurios
```

```
  pinMode( 50, OUTPUT ); // Relay_2_k2_D
```

```
  pinMode(52,OUTPUT); // Relay_3_k3_Y
```

```
  digitalWrite( 48, LOW );
```

```
  digitalWrite( 50, LOW );
```

```
  digitalWrite(52,LOW);
```

```
//-----POWER UP-----
```

```
pinMode(9, OUTPUT);
```

```
digitalWrite(9,LOW);
```

```
delay(1000);
```

```
digitalWrite(9,HIGH);
```

```
delay(2000);
```

```
digitalWrite(9,LOW);
```

```

delay(3000);
//.....
Serial.println( "AT+CMGF=1" );
delay(200);
}

```

```

void loop() {

{
    sum=0;
    for (i=0;i<20;i++){
    t1 = analogRead(2); // diavasma tashs eksodou LM35 (10mV gia kathe vathmo
    delay(200);
    t2 = analogRead(2); // diavasma tashs eksodou LM35 (10mV gia kathe vathmo
    delay(200);
    sum+=t2;}
    LM35_sensor=sum/20;
    LM35_celsius = 0.48828125 * LM35_sensor;
    delay(1000);    }
    LDR ();
    moisturesensor ();

Serial.println("METRISEIS");
Serial.println(LM35_celsius);
Serial.println(sensorValue1);

char SerialInByte;

```

```

if(Serial.available())
{
  SerialInByte = (unsigned char)Serial.read();
  delay(5);
}
// -----
// EN: If the message ends with <CR> then process the message
if( SerialInByte == 13 ){
  // EN: Store the char into the message buffer
  ProcessGprsMsg();
}
if( SerialInByte == 10 ){

}
else {

  msg += String(SerialInByte);
}

```

```

if ((thelopotisma=1) && (sensorValue1<150 )) { //An steiloume START xsekina kata
//apaithsh mas kai doulevei gia oso

```

```

thelopotisma=1;
pumproutine();

```

```

delay(200);
K=1;
Serial.print( "AT+CMGF=1\r" );

```

```

delay(100);
Serial.println( "AT+CMGS="+3069XXXXXXXXX\ "" );
delay(100);
Serial.println( "KINHTHRAS START" );
delay(100);
Serial.println( char(26) );
Serial.println();
}

```

if (sensorValue2>650){ //stamatima gt to xwrafi potistike kai eidopoihsh me SMS.

```

stoproutine();
thelopotisma=0;
delay(200);
Serial.print( "AT+CMGF=1\r" );
    delay(100);
    Serial.println( "AT+CMGS="+3069XXXXXXXXX\ "" );
    delay(100);
    Serial.println( "KINHTHRAS STOP XWRAFI POTISMENO" );
    delay(100);
    Serial.println( char(26) );
    Serial.println();
}

```

//.....ANAFORA-ELEGXO IN2..GIA THERMIKO.....

```

IN2_State=digitalRead(3);
if ( (IN2_State==HIGH) && (F!=1) ) {

    delay(200);

```

```

F=1;
Serial.print( "AT+CMGF=1\r" );
delay(100);
Serial.println( "AT+CMGS=\"+3069XXXXXXXXX\"" );
delay(100);
Serial.println( "THERMIKO SFALMA" );
delay(100);
Serial.println( char(26) );
Serial.println();
}

```

```
//.....ANAFORA-ELEGXO IN3..GIA..DIAKOPH..REUMATOS.....
```

```

IN3_State=digitalRead(10);
if ( (IN3_State==HIGH) && (L!=1) ) {

    delay(200);
    L=1;
    Serial.print( "AT+CMGF=1\r" );
    delay(100);
    Serial.println( "AT+CMGS=\"+3069XXXXXXXXX\"" );
    delay(100);
    Serial.println( "DIAKOPH REYMATOS" );
    delay(100);
    Serial.println( char(26) );
    Serial.println();
}

if (( IN3_State==LOW )&& L==1 ){

```

```

delay(200);
L=0;
Serial.print( "AT+CMGF=1\r" );
delay(100);
Serial.println( "AT+CMGS=\"+3069XXXXXXXXX\"" );
delay(100);
Serial.println( "EPANAFORA REYMATOS TASH" );
delay(100);
Serial.println( char(26) );
Serial.println();

}
}

//.....Epeksergasia SMS k ekinisi +elegxos ugrasias.....

void ProcessSms( String sms ){
  if ( sms.indexOf("START"))
  thelopotisma=1;
  //-----den peirazw tpt-----
  // Delete all message storage
  Serial.println( "AT+CMGD=1,4" );
  delay(500);
}

```

```

void stoproutine(){                                // rotina STOP kinitira(afoplizei ola ta rele!)
    digitalWrite(relay_1,LOW );
    delay(2000);
    digitalWrite(relay_2,LOW);
    delay(1000);
    digitalWrite(relay_3,LOW);
}

```

// EN: Request Text Mode for SMS messaging

```

void GprsTextModeSMS(){
    Serial.println( "AT+CMGF=1" );
}

```

```

void GprsReadSmsStore( String SmsStorePos ){
    Serial.print( "AT+CMGR=" );
    Serial.println( SmsStorePos );
}

```

// EN: Clear the GPRS shield message buffer

```

void ClearGprsMsg(){
    msg = "";
}

```

// EN: interpret the GPRS shield message and act appropriately

```

void ProcessGprsMsg() {
    if( msg.indexOf( "Call Ready" ) >= 0 ){

```

```

// Serial.println( "*** GPRS Shield registered on Mobile Network ***" );
  GprsTextModeSMS();
}

// EN: unsolicited message received when getting a SMS message
if( msg.indexOf( "+CMTI" ) >= 0 ){
  // EN: Look for the coma in the full message (+CMTI: "SM",6)
  //   In the sample, the SMS is stored at position 6
  int iPos = msg.indexOf( "," );
  String SmsStorePos = msg.substring( iPos+1 );
  // EN: Ask to read the SMS store
  GprsReadSmsStore( SmsStorePos );
}

// EN: SMS store readed through UART (result of GprsReadSmsStore request)
if(( msg.indexOf( "+CMGR:" ) >= 0 )&& ( msg.indexOf( "6989220575" ) >= 0 )){
  // EN: Next message will contains the BODY of SMS
  SmsContentFlag = 1;
  // EN: Following lines are essential to not clear the flag!
  ClearGprsMsg();
  return;
}

// EN: +CMGR message just before indicate that the following GRPS Shield message
//   (this message) will contains the SMS body
if( SmsContentFlag == 1 ){

  ProcessSms( msg );
}

```



```

ClearGprsMsg();
// EN: Always clear the flag
SmsContentFlag = 0;

}

//-----

//-----routina kinitira-----
void pumproutine () {

digitalWrite(relay_3,HIGH);
delay(100);

digitalWrite(relay_1,HIGH);
delay(100);

digitalWrite(relay_3,LOW);
delay(100);

digitalWrite(relay_2,HIGH);
delay(10);

}

//-----elegxos thermokrasias gia to antipagetiko-----
void elegxospagetou(){
if (LM35_celsius<1.5) { //otan h thermokrasia pesei katw apo 1.5 vathmous kelsiou amesa

```

```

                //enarxsh ths antlias kai enhmerwsh me SMS.
pumprountine ();
potizwgiapago=1;
Serial.print( "AT+CMGF=1\r" );
    delay(100);
    Serial.println( "AT+CMGS=\"+3069XXXXXXXXX\" );
    delay(100);
Serial.println( "XAMHLH THERMOKRASIA ENARKSH ANTIPAGETIKOU" );
    delay(100);
    Serial.println( char(26) );
    Serial.println();}
else if ((LM35_celsius>2.5) && (potizwgiapago=1)){ //otan h thermokrasia megalhteri apo
2.5 vathmous stamatima tis
                //antlias kai enhmerwsh me SMS.
stoprountine();
potizwgiapago=0;
Serial.print( "AT+CMGF=1\r" );
    delay(100);
    Serial.println( "AT+CMGS=\"+3069XXXXXXXXX\" );
    delay(100);
Serial.println( "THERMOKRASIA SE KANONIKA EPIPEDA STOP ANTIPAGETIKOU" );
    delay(100);
    Serial.println( char(26) );
    Serial.println();
}
}

//-----periptwsh ksirasias(elegxos ugrasias-mono eidopoihsh)-----
void moisturesensor() {

    sensorValue2 = analogRead(sensorPin1);

```

```

//Serial.println(sensorValue2);
delay(5000);

if (sensorValue2<350) { //as upothesoume oti sto 350 to xwrafi xreiazete nero. Eidopohsh
me SMS

                //otan ftasei auth th timi. Stelnoume emeis entolh na xsekinisei.
delay(10000);
Serial.print( "AT+CMGF=1\r" );
    delay(100);
    Serial.println( "AT+CMGS=\"+3069XXXXXXXXX\"" );
    delay(100);
Serial.println( "XSERO XWMA ANAGKH GIA POTISMA" );
    delay(100);
    Serial.println( char(26) );
    Serial.println();
}
}
//-----Elegxos fwthnohtas(gia potisma mono vradu)-----

void LDR () { //elegxos tis fwthnohtas me xrisi LDR sundesmologia diareths tashs

    sensorValue1 = analogRead(sensorPin1);
    //Serial.println(sensorValue1);
    delay(5000);
}

```

Βιβλιογραφία

1. https://www.espruino.com/datasheets/SIM900_AT.pdf
2. <https://www.arduino.cc/>
3. <https://howtomechatronics.com/tutorials/arduino/control-high-voltage-devices-arduino-relay-tutorial/>
4. http://wiki.sunfounder.cc/index.php?title=8_Channel_5V_Relay_Module
5. <https://www.instructables.com/id/Arduino-Soil-Moisture-Sensor/>
6. <https://create.arduino.cc/projecthub/electropeak/complete-guide-to-use-soil-moisture-sensor-w-examples-756b1f>
7. https://www.ti.com/product/LM35?utm_source=google&utm_medium=cpc&utm_campaign=asc-null-null-GPN_EN-cpc-pf-google-ww&utm_content=LM35&ds_k=LM35&DCM=yes&gclid=EAlaIQobChMI56bq0aK26AIVhrTtCh0ycg4wEAAAYASAAEgJ-zfD_BwE&gclid=aw.ds
8. <https://www.instructables.com/id/ARDUINO-TEMPERATURE-SENSOR-LM35/>
9. <https://en.wikipedia.org/wiki/Photoresistor>
10. www.instructables.com/id/How-to-use-a-photoresistor-or-photocell-Arduino-Tu/
11. <https://www.espruino.com/arduino-sim900>
12. https://www.espruino.com/datasheets/SIM900_AT.pdf
13. https://science.fandom.com/el/wiki/%CE%93%CE%B5%CF%89%CF%81%CE%B3%CE%B9%CE%BA%CE%AE_%CE%A4%CE%B5%CF%87%CE%BD%CE%BF%CE%BB%CE%BF%CE%B3%CE%AF%CE%B1
14. [https://el.wikipedia.org/wiki/%CE%93%CE%B5%CF%89%CF%81%CE%B3%CE%AF%CE%B1_\(%CE%B4%CF%81%CE%B1%CF%83%CF%84%CE%B7%CF%81%CE%B9%CF%8C%CF%84%CE%B7%CF%84%CE%B1\)](https://el.wikipedia.org/wiki/%CE%93%CE%B5%CF%89%CF%81%CE%B3%CE%AF%CE%B1_(%CE%B4%CF%81%CE%B1%CF%83%CF%84%CE%B7%CF%81%CE%B9%CF%8C%CF%84%CE%B7%CF%84%CE%B1))
15. [https://en.wikipedia.org/wiki/Wiring_\(development_platform\)](https://en.wikipedia.org/wiki/Wiring_(development_platform))
16. <http://wiring.org.co/>
17. <https://en.wikipedia.org/wiki/I%C2%B2C>
18. https://en.wikipedia.org/wiki/AVR_microcontrollers
19. <https://fritzing.org/home/>
20. https://en.wikipedia.org/wiki/Serial_Peripheral_Interface

21. <https://www.best-microcontroller-projects.com/digital-i-o.html>
22. <https://www.best-microcontroller-projects.com/i2c-tutorial.html>
23. <https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-mega-2560.html>
24. <https://www.espressif.com/en/products/modules>
25. <https://grobotronics.com/esp8266-wifi-module.html>
26. <https://randomnerdtutorials.com/esp8266-pinout-reference-gpios/>
27. <https://circuits4you.com/wp-content/uploads/2018/12/ESP32-Pinout.jpg>
28. <https://www.kth.se/social/files/54ef17dbf27654753f437c56/GL5537.pdf>
29. <https://www.datasheetarchive.com/photoresistor%20LDR%205mm-datasheet.html>
30. <https://el.wikipedia.org/wiki/Lux>
31. <https://www.pelc.gr/el/voltage-control/33-pt431.html>
32. <https://el.wikipedia.org/wiki/%CE%97%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%BF%CE%BD%CF%8C%CE%BC%CE%BF%CF%82>
33. <https://dimitrisefthimiou.weebly.com/thetaepsilonrhoiotaakappa940.html>
34. Arduino Cookbook, Michael Margolis and Nicholas Welling Εκδόσεις O'REILY Έτος 2011

