ee

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη εφαρμογής ελέγχου αισθητήρων με Arduino.

Κωνσταντίνος Μουρτζανός

Επιβλέπων καθηγητής: Ευριπίδης Γλαβάς

Κοσμήτορας σχολής πληροφορικής και τηλεπικοινωνιών Πανεπιστημίου Ιωαννίνων.

Αθήνα ,Σεπτέμβριος 2022

Development of sensor control application with Arduino.

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Αρτα, 28/9/2022

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπων καθηγητής

   Ευριπίδης Γλαβάς,

2. Μέλος επιτροπής

   Νικόλαος Γιαννακέας,

3. Μέλος επιτροπής

   Αλέξανδρος Τζάλας,

© Μουρτζανός, Κωνσταντίνος, 2022.
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα μεταπτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Μουρτζανός, Κωνσταντίνος

Υπογραφή

ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας πτυχιακής εργασίας είναι η δημιουργία ενός ενσωματωμένου συστήματος θερμοστάτη με την δυνατότητα τοποθέτησης σε εσωτερικό ενός κλειστού εσωτερικού χώρου με τη ικανότητα αυτόματου ή χειροκίνητου ελέγχου της ψύξης και της θέρμανσης ανάλογα με την εξωτερική θερμοκρασία που επικρατεί εμφανίζοντας την θερμοκρασία σε μια LCD οθόνη ,επίσης δημιουργήθηκε σύστημα με δυνατότητα εντοπισμού φωτιάς με πρόβλεψη για την αντιμετώπισης της. Η εργασία αυτή αναπτύσσεται μέσα σε 4 κεφάλαια. Στο πρώτο κεφάλαιο γίνεται μια ιστορική αναδρομή στους μικροελεγκτές σε συνδυασμό με ανάλυση της βασικής αρχιτεκτονικής σχεδίασης διαφόρων μοντέλων επεξεργαστών και παρουσίαση της πλακέτας Arduino uno R3 που χρησιμοποιήθηκε σε συνδυασμό με διάφορα εξαρτήματα . Στο επόμενο κεφάλαιο γίνεται περιγραφή της γλώσσας προγραμματισμού μαζί με το περιβάλλον που χρησιμοποιήθηκε με αναφορά στις βασικές εντολές και τις βιβλιοθήκες .Στο τρίτο κεφάλαιο δημιουργήθηκε κώδικας στο περιβάλλον του Visual Code Studio που ενσωματώθηκε στην πλακέτα του Arduino και στην συνέχεια γίνεται παρουσίαση του κώδικα για την λειτουργία του θερμοστάτη με δυνατότητα εμφάνισης των θερμοκρασιών που επικρατούν σε μια οθόνη σε συνδυασμό με την λειτουργία πυρόσβεσης, επίσης παρουσιάστηκε ο τρόπος με τον οποίον ενσωματώθηκαν στο σύστημα τα components με παρουσίαση της συνδεσμολογίας τους και επεξήγηση του κώδικα με ξεχωριστή ανάλυση των διαφορετικών functions που δημιουργήθηκαν .Τέλος στο τέταρτο και τελευταίο κεφάλαιο παρουσιάζονται τα συμπεράσματα μας από την ολοκλήρωση της πτυχιακής καθώς και κάποιες σκέψεις για την μελλοντική επέκταση της παρούσας εφαρμογής.

Λέξεις-κλειδιά: Arduino,μικροελεγκτής , θερμοστάτης ,αισθητήρας θερμοκρασίας.

ABSTRACT

The purpose of this thesis is to create a integrated thermostat system with the ability to placed inside to indoor space with the possibility of automatic or manual control of the cold and heating according to the prevailing external temperature by displaying the temperature on LCD screen, also created system which detecting of fire and ability to deal with it. The work is developed in 4 chapters. In the first chapter there is a historical review of microcontrollers combined with an analysis of the basic architectural design of various processor models and a presentation of the Arduino uno R3 board which was used in combination with various accessories. The next chapter describes the programming language together with environment used with reference to the basic commands and libraries. In the third chapter , code was created in the Visual Code Studio environment that was integrated into the Arduino board ,then code for the operation of the thermostat with the ability to display the prevailing temperature on LCD screen in combination with the fire extinguishing function , in this way also presented the components were integrated into the system with a presentation of their wiring and an explanation of the code with a separate analysis of the different function that were created .Finally the fourth and last chapter presents our conclusion from the completion of the thesis as well some thoughts for the future expansion of this application.

Keywords: Arduino, microprocessor, thermostat, temperature sensor.

Περιεχόμενα

1 Μικροελεγκτές.

Οι επαναπρογραμματιζόμενες πλακέτες είναι υπολογιστική πλατφόρμα που βασίζεται σε μια μητρική πλακέτα με έναν ενσωματωμένο μικροελεγκτή και περιέχει εισόδους/εξόδους .Χρησιμοποιείται κυρίως για εκπαιδευτικούς σκοπούς για την ανάπτυξη κυκλωμάτων. Η συσκευές τέτοιου τύπου συνδέονται με έναν Η/Υ μεσώ ενός απλού καλωδίου usb με την χρήση των κατάλληλων drivers και την δημιουργία κώδικα σε περιβάλλον κατάλληλο για αυτήν την χρήση. Η πλακέτα είναι πλήρως προγραμματιζόμενη και το λογισμικό εκτελείται από τον υπολογιστή οπού γίνεται η ανάπτυξη του κώδικα και η φόρτωση του στην πλακέτα. Ανάλογα με την χρήση υπάρχει ποικιλία σε προγραμματιζόμενες πλακέτες από διάφορες εταιρείες (Texas instruments, board τύπου Arduino, keyestudio, adafruit)που συνδυάζουν διάφορα χαρακτηρίστηκα. Ανεξάρτητα από την πλακέτα ο προγραμματισμός και η λογική παραμένει η ίδια προσαρμόζοντας τις θέσεις pin ή χρησιμοποιώντας την ανάλογη βιβλιοθήκη . Υπάρχει πλήθος δυνατοτήτων σε αυτές τις πλακέτες καθώς υπάρχουν άπειρα modules που μπορεί να χρησιμοποιήσει ο χρήστης ανάλογα με τις ανάγκες του.

1.1 Εισαγωγή.

Ο πρώτος επαναπρογραμματιζόμενος ηλεκτρονικός υπολογιστής ονομάστηκε ENIAC (Electronic Numerical Integrator and Computer) σχεδιάστηκε και κατασκευάστηκε υπό την εποπτεία του καθηγητή φυσικής John Mauchly και του μεταπτυχιακού φοιτητή John Presper Eckert στο πανεπιστήμιο της πενσυλβάνια , με την ικανότητα να λύσει ένα πλήρες εύρος υπολογιστικών προβλημάτων. Ο αρχικός σκοπός της κατασκευής του ENIAC ήταν η ανάπτυξη ενός συστήματος που να έχει την ικανότητα άμεσης αναγνώρισης εμβέλειας και τροχιάς για να γίνει συνδυασμός με υπερόπλα της εποχής κατασκευασμένα από τον αμερικάνικο στρατό κατά τον Β΄ παγκόσμιο πόλεμο . Η επίσημη παρουσίαση έγινε στις 15 Φεβρουαρίου του 1946 πράγμα που σημαίνει ότι δεν χρησιμοποιήθηκε για τον αρχικό σκοπό του , με την κατασκευή να έχει κοστίσει σχεδόν 500.000$.

1.1.1 Ιστορική αναδρομή μικροεπεξεργαστών.

Σχεδόν τρείς δεκαετίες μετά την δημιουργία του πρώτου υπολογιστή δημιουργήθηκε ο πρώτος μικροεπεξεργαστής το 1972. Οι σχεδιαστές αυτών των συσκευών μετέφεραν τις ιδέες από τη σχεδίαση μεγάλων υπολογιστικών συστημάτων με αποτέλεσμα να γίνει χρήση πολλών στοιχείων της οργάνωσης και της αρχιτεκτονικής των μεσαίων και μεγάλων συστημάτων. Οι μικροελεγκτές είναι ένα προγραμματιζόμενο ολοκληρωμένο κύκλωμα το οποίο διαθέτει επεξεργαστή μνήμη, διάφορα περιφερειακά κυκλώματα και θύρες εισόδου / εξόδου για επικοινωνία με εξωτερικές συσκευές. Αυτά τα κυκλώματα έχουν σχεδιαστεί για χρήση με ενσωματωμένα συστήματα . Ενσωματωμένο σύστημα είναι ένα σύστημα ηλεκτρονικού υπολογιστή με μια συγκεκριμένη λειτουργία ή πιο απλά είναι ένας υπολογιστής μέσα σε μια πλακέτα. Οι μικροελεγκτές διαθέτουν συγκεκριμένα

χαρακτηρίστηκα και συγκεκριμένες αποδόσεις. Ένας μικροελεγκτής έχει ως σκοπό τον μικρό αριθμό ολοκληρωμένων κυκλωμάτων έτσι ώστε να επιτύχει μια συγκεκριμένη λειτουργία με χαμηλό κόστος. Ο μικροελεγκτής είναι παρόμοιος με έναν επεξεργαστή , η χρήση τους γίνεται σε συστήματα που έχουν ως σκοπό την ελαχιστοποίηση του κόστους (πχ. Συστήματα αυτοματισμού ,ηλεκτρονικές/ηλεκτρικές συσκευές , μονάδες ABS , συστήματα συλλογής δεδομένων),τον τελευταίο καιρό ο άνθρωπος χρησιμοποιεί στην καθημερινότητα του πληθώρα αριθμό συσκευών που χρησιμοποιούν συστήματα μικροεπεξεργαστών.

1.1.2 Ιστορία του Arduino.

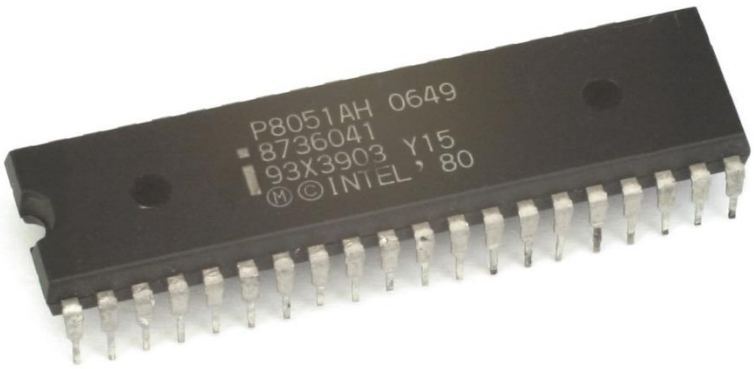
Η ιδέα του Arduino ξεκίνησε το 2005 στην βόρεια Ιταλία στο Interaction Design institute of Ivrea, οι σχεδιαστές χρησιμοποίησαν την γλώσσα προγραμματισμού processing σε συνδυασμό με ένα ολοκληρωμένο περιβάλλον ανάπτυξης που ονομάζεται wiring.Το processing είναι μια γλώσσα προγραμματισμού που δημιουργήθηκε από τον Casey Reas και τον Benjamin Fry το 2001. Η ανάγκη τους για χειρισμό διαδραστικών εφαρμογών multimedia τους ώθησε στην δημιουργία δικιάς τους γλώσσας προγραμματισμού που βασίζεται στην JAVA αλλά με απλοποιημένη σύνταξη κώδικα και πλήρες IDE με σκοπό την άμεση δοκιμή των εφαρμογών τους. Η συγκεκριμένη γλώσσα χρησιμοποιήθηκε για διδακτικούς λόγους σε αρχάριους χρήστες που δεν έχουν καμία γνώση προγραμματισμού αλλά και σε έμπειρους χρήστες που θέλανε να ασχοληθούνε με την ανάπτυξη εφαρμογών με multimedia. Το 2003 στην βόρεια Ιταλία στο ίδιο ινστιτούτο ο Hernando Barragan αναλάβε ως μεταπτυχιακή εργασία (wiring) με επιβλέποντες καθηγητές τον Massimo Banzi και Casey Reas (Δημιουργός γλώσσας Processing). Η Wiring είναι ανοιχτού κώδικα εφαρμογή με πλήρες IDE παρόμοιας λογικής με του Processing . Η αρχική ομάδα σχεδίασης του Arduino αποτελούνταν από τον David mellis, Gianluca Martino, Tom Igoe, David Cuartielles και Massimo Banzi.Το Arduino είναι μια αναβάθμιση της Wiring(μικροελεγκτής wiring: ATmega168) καθώς οι σχεδιαστές του αύξησαν την καλωδίωση με την υποστήριξη του πιο οικονομικού μικροελεγκτή ATmega8 ,σκοπός τους ήταν να δημιουργήσουν μια οικονομική συσκευή με δυνατότητα ελέγχου και αλληλεπίδρασης ανάλογα με το περιβάλλον.

1.2 Αρχιτεκτονική.

Στο εμπόριο υπάρχουν άπειρες επιλογές πλακετών με διαφορετικά χαρακτηρίστηκα .Η επιλογή ενός μικροελεγκτή εξαρτώνται από την χρήση και τις απαιτήσεις του χρήστη . Αν χρειαζόμαστε μικρή κατανάλωση μπορούμε να επιλέξουμε ανάμεσα σε MSP430 της Texas Instruments , τους PIC της Microchip, ή τους AVR της Atmel. Αντίθετα αν θέλουμε υψηλότερη απόδοση αλλά σε μικρό μέγεθος μπορούμε να διαλέξουμε πλακέτες όπως το RaspberryPi ή το Beagleboard. Αν κάποιος χρειάζεται τον μέγιστο βαθμό αξιοπιστίας και ελάχιστα σφάλματα μπορεί να χρησιμοποιήσει μικροελεγκτές με ενσωματωμένες στο hardware δυνατότητες διόρθωσης λαθών, όπως στη σειρά Hercules της TexasInstruments ή στους Qorivva της Freescale.

1.2.1 Microcontroller.

Ένας μικροελεγκτής (microcontroller) είναι μια μορφή επεξεργαστή με την ικανότητα σύνδεσης με εξωτερικά εξαρτήματα .Η χρήση του είναι διαδεδομένη σε ενσωματωμένα συστήματα (embedded systems) με την δυνατότητα ελέγχου εφαρμογών χαμηλού κόστους. Είναι εξοπλισμένο σε αυτοματοποιημένα συστήματα όπως φορητές παιχνιδοκονσόλες ,ηλεκτρικές συσκευές και κάθε είδους συσκευή που διαθέτει αυτοματοποιημένες λειτουργίες .



Εικόνα 2.2.1 Intel 8051 φωτογραφία από Wikipedia.

Οι Μικροελεγκτές χρησιμοποιούνται σε ενσωματωμένα συστήματα για την δημιουργία εφαρμογών ελέγχου σε πραγματικό χρόνο με απευθείας φόρτωση του κατάλληλου κώδικα στην μνήμη του με σκοπό την εκμετάλλευσή των περιφερειακών του. Τα τελευταία χρόνια τέτοιου τύπου συσκευές έχουν μπει στην ζωή του ανθρώπου με διάφορους τρόπους. Η χρήση τους γίνεται στα μηχανοκίνητα οχήματα(ταμπλέτες, εγκέφαλοι οχήματος), μηχανές γραφείου(εκτυπωτές ,FAX , σαρωτές).Ιατρικές συσκευές που βοηθούν ως προς την διάγνωση μιας αρρώστιας, ως προς την παρακολούθηση της υγείας του ασθενή και ως προς την τεχνική υποστήριξη (τεχνητά μέλη). Χρησιμοποιούνται σε μηχανήματα αυτόματης πώλησης και τέλος σ ε οικιακές συσκευές (έξυπνες λάμπες ,φούρνοι, οποιαδήποτε συσκευή έχει την ικανότητα να συνδεθεί απομακρυσμένα με κάποια συσκευή).

Μερικά πλεονεκτήματα των μικροελεγκτών:

- Αυτονομία.
- Αξιοπιστία.
- Μικρό μέγεθος υπολογιστικού συστήματος.
- Απλές διασυνδέσεις με ευκολότερη υλοποίηση.
- Οικονομικό.
- Επεκτασιμότητα.

- Μεγάλη κοινότητα υποστήριξης.
- Open-source hardware.

Μερικά μοντέλα μικροελεγκτή και εταιρείες :

- MSP430F1x (Texas Instruments)
- PIC16F84A (Microchip)
- S3F80P5 (Zilog)
- HC08EY (Freescale)
- AVR ATMEGA328 (Atmel)
- BasicStamp (Parallax)
- MAXQ611 (Maxim)
- 8051 (Intel)
- MEGA (Arduino)

Οι μικροελεγκτές ανεξάρτητα από την εταιρεία προσφέρουν πολλές επιλογές εφαρμογών και δυνατοτήτων , ο κάθε χρήστης έχει την επιλογή απο έναν μικροεπεξεργαστή ανάλογα με την χρήση για την οποία τον θέλει .

- MSP430 είναι ιδανικός για ενσωματωμένα συστήματα με μικρή κατανάλωση ενέργειας αλλά δύσκολο στη μετακίνηση δεδομένων καθώς δεν είναι εξοπλισμένο με DMA output strobe.
- Οι μικροεπεξεργαστές της ARM είναι αρχιτεκτονικής RISC , χρησιμοποιούνται κυρίως σε κινητά τηλέφωνα. Σκοπός της εταιρείας είναι οι κατασκευάστριες εταιρείες να χρησιμοποιούν τα προϊόντα της ως βασικό σχεδιασμό.
- Η atmel χρησιμοποιεί επαναπρογραμματιζόμενους μικροελεγκτές και έχει ως σκοπό να δίνει λύσεις σε αυτά τα συστήματα , παρέχει συσκευές (ASICs ,ASSPs)με συγκεκριμένα χαρακτηρίστηκα ανάλογα με τις απαιτήσεις των πελατών της .
- Τα προϊόντα της microchip technologies περιέχουν συσκευές EEPROM ,συσκευές σειριακών SRAM ,μικροελεγκτές (PICmicro, dsPIC / PIC24, PIC32),αναλογικές συσκευές διαχείρισης θερμότητας-ισχύς και μπαταριάς. Οι συσκευές αυτές είναι κατάλληλες για αλληλεπίδραση με κάποιο πληροφοριακό σύστημα λόγο της μικρής κατανάλωσης ισχύος και της λειτουργικότητας είναι ιδανικοί για εφαρμογές που απαιτούν μικρή κατανάλωση ενέργειας .
- Η intel είναι ο κύριος προμηθευτής μικροεπεξεργαστών σε υπολογιστές και σε συνεργασία με την Microsoft βρίσκεται σε συνεχή προσπάθεια ανάπτυξης των προϊόντων της
- Η freescale έχει προϊόντα ολοκληρωμένων κυκλωμάτων στην αγορά της αυτοκίνησης και ενσωματωμένων συστημάτων τηλεπικοινωνιών .

Διαφορές μικροεπεξεργαστή με μικροελεγκτή:

Οι μικροελεγκτές καταναλώνουν λιγότερη ενέργεια με μικρότερη κατανάλωση χώρου στην μνήμη σε αντίθεση με τους μικροεπεξεργαστές που απαιτούν μεγαλύτερη ενέργεια-μνήμη για να λειτουργήσουν . Ένας

μικροεπεξεργαστής μπορεί να εκτελέσει πολλαπλές λειτουργίες συγχρόνως (multitasking) ενώ η φιλοσοφία επικεντρώνεται σε μια πολύ συγκεκριμένη λειτουργία. Σε έναν μικροεπεξεργαστή τα τεχνικά χαρακτηρίστηκα μπορούν να αναβαθμιστούν αφήνοντας τον χρήστη να διαλέξει μνήμη RAM ,ROM,θύρες εισόδου/εξόδου, αντιθέτως με τον μικροελεγκτή που όλα αυτά περιέχονται σε μια πλακέτα με συγκεκριμένα τεχνικά χαρακτηρίστηκα. Ο χρήστης θα επιλέξει έναν μικροελεγκτή για να δημιουργήσει ένα σύστημα με πολύ συγκεκριμένη λειτουργία με λιγότερο κόστος και ευκολότερη μετακίνηση.

Μοντέλα μικροελεγκτών:

	MSP430	**PIC**	**AVR**	**8051**
Αρχιτεκτονική	16bit	16/32bit	8/32bit	8bit
Μνήμη Flash	Έως 512KB	512KB	Έως 256KB	Έως 64 KB
Μνήμη RAM	Έως128 KB	Έως 512 KB	Έως 16 KB	256 bytes
Επεξεργαστής (μέγιστα MHz)	25MHz	252MHz	66MHz	12MHz
Ενδεικτική τιμή	Χαμηλή(<5$)	Χαμηλή(<3$)	Χαμηλή(<10$)	Χαμηλή(<2$)
Υποστηριζόμενα Λειτουργικά	RTOS	RTOS	RTOS	RTOS

Πίνακας2.2.1 Σύγκριση μοντέλων μικροελεγκτών.

Στον παραπάνω πίνακα γίνεται σύγκριση ανάμεσα σε 4 ανταγωνιστικούς μικροελεγκτές τους MSP430 της Texas Instruments , τους PIC της Microchip, τους AVR της Atmel (πλέον Microchip) και τον 8051 της Intel (και άλλων εταιρειών με συμβατές αρχιτεκτονικές). Η πλειοψηφία των πλακετών Arduino αποτελούνται από μικροελεγκτές AVR των 8 bit. Οι AVR των 32 bit (AVR32) ωστόσο είναι υπό εξαφάνιση, αφού η Atmel στράφηκε προς τους Cortex-M της ARM. Οι δύο αρχιτεκτονικές παρότι μοιράζονται το ίδιο όνομα έχουν σημαντικές σχεδιαστικές διαφορές. Οι αναφερθέντες Cortex -M χρίζουν αναφοράς λόγω της όλο και αυξανόμενης χρήσης τους, ωστόσο ποικίλλουν στα χαρακτηριστικά τους ανάμεσα στις διάφορες γενιές που δεν είναι εύκολο να συνοψιστούν σε έναν πίνακα .Ο 8051 αποτελεί μία παλιά σχεδίαση, του1980 ,και δεν παράγεται πλέον από την Intel. Ωστόσο υπάρχουν πολλοί συμβατοί κλώνοι(αρκετές φορές με καλύτερη απόδοση από το πρωτότυπο)και ευρεία συμβατότητα και χρήση , χάρη στο μεγάλο διάστημα που ήταν στην αγορά. Οι τιμές είναι ενδεικτικές αφορούν μόνο τον μικροελεγκτή και όχι ολόκληρη την πλακέτα , ποικίλουν ανάλογα με την ιστοσελίδα-αποθήκη και την γνησιότητα τους.

1.2.2 Τροφοδοσία.

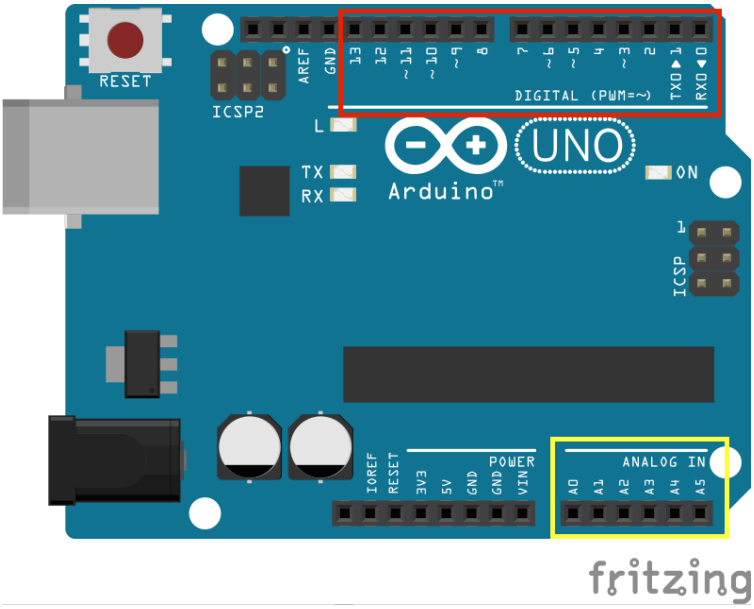
Ένα οποιοδήποτε board αποτελείτε από usb connector οπού γίνεται η σύνδεση του υπολογιστή για ανέβασμα του κώδικα και η τροφοδοσία της πλακέτας που είναι μια υποδοχή φις των 2.1mm με έναν θετικό πόλο . Εφόσον δεν υπάρχει διαθέσιμη θύρα usb και θύρα τροφοδοσίας ρεύματος που είναι σημαντική για την αλληλεπίδρασής συσκευής και του χρήστη μπορεί να τοποθετηθεί το MB102 Breadboard power supply module που είναι μια διεπαφή USB που μπορεί συγχρόνως να δώσει τάση 3.3/5V. Ο κατασκευαστής στο datasheet της εκάστοτε πλακέτας ορίζει το μέγιστο αριθμό volt που μπορεί να υποστηρίξει η πλακέτα (συνήθως είναι 20 V) πράγμα που σημαίνει ότι η εξωτερική τροφοδοσία μπορεί να αποτελείτε από μπαταρίες ή έναν μετασχηματιστή με την προτεινόμενη τιμή να είναι από 7V μέχρι 12V.

Εικόνα 2.2.2α Arduino power DC.

Εικόνα 1.2.2β Arduino mini.

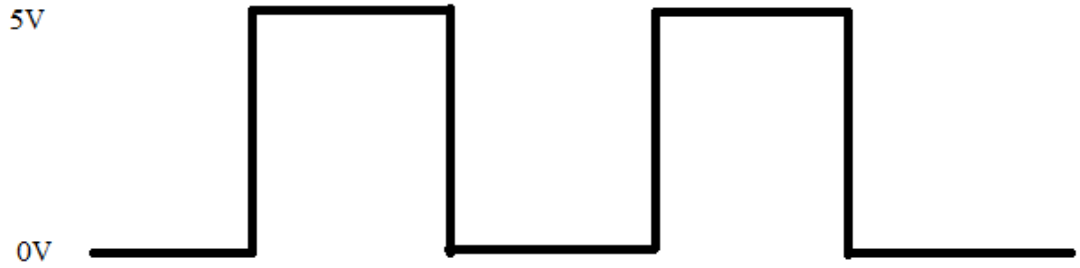
1.2.3 Είσοδοι/ Έξοδοι(I/O).

Οι περισσότερες πλακέτες που βρίσκονται στο εμπόριο χρησιμοποιούν τάση από 0V μέχρι 5v. Είναι εξοπλισμένοι με analog και digital pins εισόδου/εξόδου οπού εκεί μπορεί να συνδεθούν άπειρα εξαρτήματα (led ,potensiometers, αισθητήρες θερμοκρασίας, αισθητήρας φωτός, κουμπιά κτλ) επίσης είναι εξοπλισμένο με κουμπί επανεκκίνησης (RESET) στην περίπτωση που υπάρξει bug στη πλακέτα. Τα pins χωρίζονται σε δύο κατηγορίες τα αναλογικά (analog) και τα ψηφιακά (digital).



Εικόνα 2.2.3 Analog and digital pins.

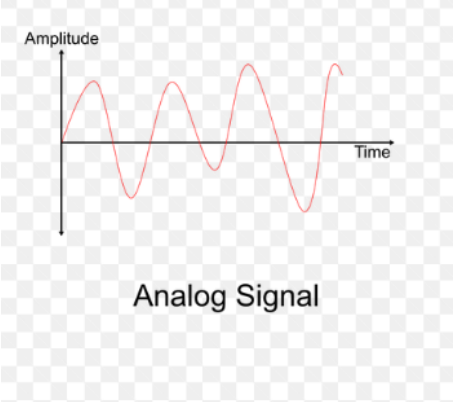
Digital: Στα digital Pins ενός μικροελεγκτή μπορούμε να στείλουμε μόνο 0V ή 5V δηλαδή LOW ή HIGH αντίστοιχα. Δηλαδή μπορεί να σταλθεί όλη η δύναμη της πλακέτας ή καθόλου δύναμη. Όταν δίνουμε την εντολή digitalWrite(LED,HIGH) τότε 5V από την πλακέτα περνάνε στο λαμπάκι με αποτέλεσμα το λαμπάκι να ανάβει.



Εικόνα 2.2.3a DIGITAL 1 OR 0.

Analog: Στα ANALOG IN Pins ενός μικροελεγκτή τοποθετούνται διάφορα ηλεκτρονικά εξαρτήματα που μπορούμε να ορίσουμε τιμές και να τις αλλάζουμε συνεχώς σε πραγματικό χρόνο (πχ ποτενσιόμετρο ,οποιοδήποτε σένσορας) αυτό γίνεται επειδή χρησιμοποιούμε τα volt της πλακέτας σαν μεταβλητή τιμή δηλαδή ο χρήστης ορίζει οποιαδήποτε τιμή ανάμεσα από 0V και 5V (πχ 2.5 Volt).Η κάθε θύρα (analog in) λειτουργεί ως αναλογική είσοδος χρησιμοποιώντας ADC (Analog to Digital Converter) που είναι ενσωματωμένο στον μικροελεγκτή έτσι αν γίνει τροφοδοσία ενός συγκεκριμένου Pin μπορεί να λάβει τιμές από το 0 (0V)μέχρι το

1023 (5V) . Η έξοδος του αναλογικού σήματος γίνεται μέσω του συστήματος διαμόρφωσης εύρους παλμού (PWM-pulse width modulation) από συγκεκριμένα Pins της πλακέτας που είναι σημειωμένα με την κατάλληλη ένδειξη. Η λειτουργία του PWM χρησιμοποιείται στις μητρικές των υπολογιστών με σκοπό τον έλεγχο της ταχύτητας που γυρνάνε τα ανεμιστηράκια για την ψύξη του συστήματος ανάλογα με την θερμοκρασία του συστήματος.



Εικόνα 2.2.3β Αναλογικό σήμα απο pixabay.

```
#include <Arduino.h>


int ledPin = 9;
int potenPin = A0;
int myValue = 0;
void setup()
{
pinMode(ledPin,OUTPUT);
pinMode(potenPin,INPUT);
}


void loop()
{
  myValue=analogRead(potenPin);

  myValue = map (myValue , 0 , 1023 , 0 , 255 );

 analogWrite(ledPin,myValue);
  delay (0);
}
```

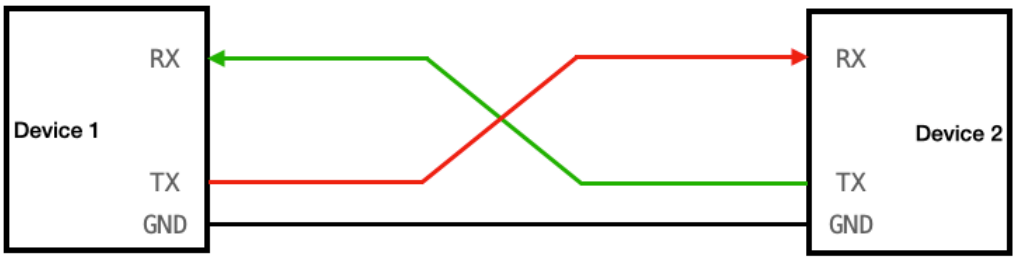
Εικόνα 2.2.3γ Potentiometer with led.

Ο παραπάνω κώδικας της εικόνας 2.2.3γ είναι μια απλή εφαρμογή με LED και ποτενσιόμετρο που περιέχει PWM . Μπορούμε να ελέγξουμε πλήρως την φωτεινότητα του LED από το ποτενσιόμετρο. Οι τιμές που παίρνει το LED είναι από 0(LOW) μέχρι 255(HIGH) με οποιαδήποτε τιμή στην μέση να μεταφράζεται σε duty cycle.Δίνοντας την τιμή 127 δεν σημαίνει ότι στο λαμπάκι δόθηκαν 2.5 V αλλά ότι δίδεται ένας εναλλασσόμενος παλμός με μεγάλη συχνότητα και ίσα χρονικά διαστήματα μεταξύ των τιμών 0V και 5V με την μέση τιμή να ισούται με 2.5V.

1.2.4 Universal Asynchronous Receiver Transmitter (UART).

Οι μικροελεγκτές έχουν την δυνατότητα να επικοινωνούνε με τον ηλεκτρονικό υπολογιστή καθώς και μεταξύ άλλων πλακετών. Στο εσωτερικό ενός

μικροελεγκτή υπάρχει πρωτόκολλο σειριακής επικοινωνίας TTL 5V UART (Universal Asynchronous Receiver Transmitter), η επικοινωνία μεταξύ συσκευών απαιτεί την ύπαρξη αυτού του πρωτοκόλλου. Η σειριακή επικοινωνία χωρίζεται σε δυο κομμάτια στην λήψη (RX) όπου ο χρήστης λαμβάνει δεδομένα και στην εκπομπή (TX) όπου ο χρήστης στέλνει δεδομένα.



Εικόνα 2.2.4 Συνδεσμολογία σειριακής επικοινωνίας (φωτο:vanhunteradams.com).

Το συγκεκριμένο πρωτόκολλο λαμβάνει δεδομένα από μια πλακέτα σε μορφή bytes και τα μετατρέπει σε bits σειριακής μορφής για την αποστολή τους . Διαθέτει έναν εσωτερικό καταχωρητή ολίσθησης (shift register) αποθηκεύοντας όλα τα δεδομένα που λήφθηκαν ή αποστάλθηκαν με κύρια λειτουργία την μετατροπή τους από σειριακή επικοινωνία σε παράλληλη. Για την χρήση UART είναι απαραίτητο οι συσκευές που θα συνδεθούν μεταξύ τους να διαθέτουν το συγκεκριμένο πρωτόκολλο καθώς η αποστολή και η λήψη δεδομένων γίνεται αποκλειστικά με τους εσωτερικούς καταχωρητές του .

1.2.5 Inter-Integrated Circuit (I²C).

Το πρωτόκολλο I2c ή I^2C σχεδιάστηκε το 1982 από την Philips για την επικοινωνία περιφερειακών με την χρήση καλωδίων, χρησιμοποιείται από πολλές ηλεκτρονικές συσκευές που απαιτούν επικοινωνία μεταξύ τους (κινητά, ενσωματωμένα συστήματα , μητρικές) καθώς είναι απλό στην λειτουργία του , οικονομικό και παρέχει αρκετά μεγάλο εύρος με υψηλές τιμές συχνότητας που ήταν το κύριο χαρακτηριστικό του που είχε ως αποτέλεσμα την σταθερή χρήση του τόσα χρόνια, επίσης είναι γνωστό σαν TWI (Two Wire Interface). Το Arduino κάνει χρήση αυτού του πρωτοκόλλου με σκοπό την χρήση λιγότερων Pins στην πλακέτα, με το συγκεκριμένο πρωτόκολλο μπορούν όλες οι συνδεμένες συσκευές να έχουν την δικιά τους μοναδική διεύθυνση με αποτέλεσμα να μπορεί να γίνει σύνδεση δύο ή περισσότερων πλακετών Arduino μεταξύ τους με την μορφή master-Slave. Το master είναι υπεύθυνο για την αποστολή και την λήψη δεδομένων από την δεύτερη συσκευή που χρησιμοποιείται σαν slave.Το Slave συνδέεται με το master με την χρήση 7 bit διευθύνσεων οπού ανάλογα με την επιλογή του χρήστη στέλνει ή λαμβάνει δεδομένα .Πέρα από την σύνδεση πλακετών μεταξύ τους υπάρχουν άπειρα I2C modules με δυνατότητα σύνδεσης στο Arduino που βοηθούν τον χρήστη στην επεκτασιμότητα και την διεύρυνση των δυνατοτήτων του project που θα δημιουργήσει .Το πρωτόκολλο I2C υποστηρίζει δυο τρόπους λειτουργίας ,την μονόδρομη μετάδοση δεδομένων που επιτρέπει ταχύτητες μετάδοσης μέχρι 5MHz με εφαρμογή σε

συσκευές που δεν απαιτείται ανατροφοδότηση για την λειτουργία τους (Πχ LED) και την αμφίδρομη μετάδοση δεδομένων που παρέχει όλες τις δυνατότητες χρήσης στον προγραμματιστή . Για να γίνει η σύνδεση στην πλακέτα του Arduino χρησιμοποιούνται δύο αναλογικά Pins το A4 που αντιστοιχεί στο SDA (Serial Data) που γίνονται όλες οι μεταφορές δεδομένων και το A5 που αντιστοιχεί στο SCL (Serial Clock) που βρίσκεται ο παλμός ρολογιού , οι θύρες A4-A5 βρίσκονται στα αναλογικά Pins ενώ τα SDA-SCL στα ψηφιακά Pins μετά το 13^{ο} και την γείωση.

Για να γίνει χρήση του συγκεκριμένου πρωτοκόλλου από την πλακέτα θα πρέπει ο χρήστης να γνωρίζει την διεύθυνση που είναι συνδεμένη η I2C συσκευή που χρησιμοποιεί. Ο συγκεκριμένος κώδικας εκτελέστηκε σε Arduino uno R3 υπάρχουν τροποποιήσεις ανάλογα με την πλακέτα.

```
#include <Arduino.h>
#include <Wire.h>

void setup() {
 Serial.begin (9600);

 // Leonardo: wait for serial port to connect
 while (!Serial)
   {
   }

 Serial.println ();
 Serial.println ("I2C scanner. Scanning ...");
 byte count = 0;

 Wire.begin();
 for (byte i = 8; i < 120; i++)
 {
   Wire.beginTransmission (i);
   if (Wire.endTransmission () == 0)
     {
     Serial.print ("Found address: ");
     Serial.print (i, DEC);
     Serial.print (" (0x");
     Serial.print (i, HEX);
     Serial.println (")");
     count++;
     delay (1);  // maybe unneeded?
     } // end of good response
 } // end of for loop
 Serial.println ("Done.");
 Serial.print ("Found ");
 Serial.print (count, DEC);
 Serial.println (" device(s).");
}  // end of setup
```

```
void loop() {}
```

Με την εκτέλεση του παραπάνω κώδικα βλέπουμε στο serial monitor σε ποια διεύθυνση είναι συνδεμένο το I2C module μας. Ο έλεγχος βρήκε μια συσκευή συνδεδεμένη στην διεύθυνση 0x27.

```
I2C scanner. Scanning ...
Found address: 39 (0x27)
Done.
Found 1 device(s).
```

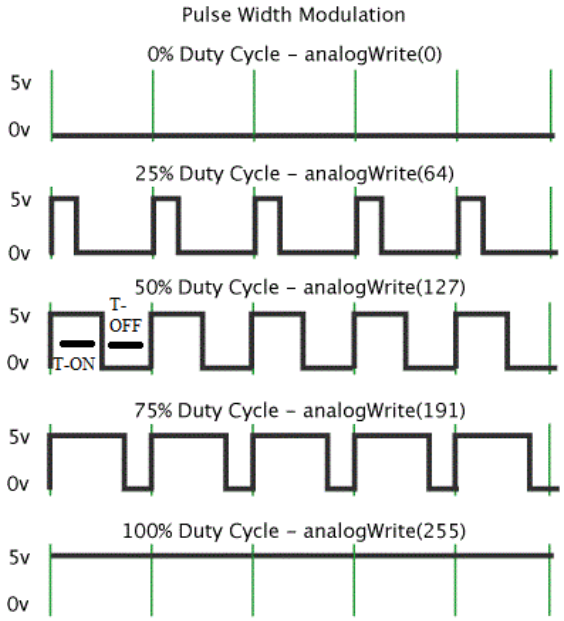
Εικόνα 2.2.5 2I2C scanner.

1.2.6 Crystal Oscillator.

Ένας κρυσταλλικός ταλαντωτής (Crystal Oscillator) είναι ένα ηλεκτρονικό κύκλωμα που χρησιμοποιεί έναν μηχανικό ταλαντωτή κρυστάλλου πιεζοηλεκτρικού υλικού που παράγει ηλεκτρικό σήμα με σταθερή συχνότητα για την παρακολούθηση του χρόνου. Οι μικροελεγκτές είναι εξοπλισμένοι με Crystal Oscillator για τον χρονισμό και τον έλεγχο. Όλες οι χρονικές λειτουργίες του μικροελεγκτή εξαρτούνται από χρονικά σήματα που για αυτά είναι υπεύθυνος ο συγκεκριμένος ταλαντωτής. Οι ταλαντωτές αυτοί έχουν χαμηλό κόστος , ελάχιστη κατανάλωση ρεύματος, μεγάλη ακρίβεια και μεγάλη παραγωγή συχνότητας.

1.2.7 Pulse Width Modulation (PWM).

Η διαμόρφωση εύρους παλμών ή Pulse Width Modulation (PWM) χρησιμοποιείτε σε πλήθος εφαρμογών για τον ακριβή έλεγχο ισχύος σερβομηχανισμών , διαφόρων ειδών DC motors , Step motors και τον έλεγχο της φωτεινότητας των LED . Το PWM είναι μια τεχνική για τον έλεγχο ενός αναλογικού σήματος με τις ψηφιακές εξόδους ενός μικροελεγκτή , ο παλμός που παράγεται μετά την σύγκριση ενός σήματος υψηλής συχνότητας με ενός χαμηλής είναι τετραγωνικός και έχει ως αποτέλεσμα την δημιουργία σταθερών ή μεταβλητών παλμών. Το σήμα υψηλής συχνότητας ονομάζεται ON-time οπού το σήμα έχει την μέγιστη τιμή του , το σήμα χαμηλής συχνότητας ονομάζεται OFF-time οπού το σήμα έχει την χαμηλότερή τιμή του, επίσης υπάρχει το Period και είναι το άθροισμα του χρόνου Ontime-OFFtime ενός σήματος PWM . Με την παραπάνω σύγκριση το PWM χρησιμοποιεί την τάση και την γείωση (οπού διαφέρει ανάλογα με την πλακέτα) έτσι ώστε να δημιουργήσει μια κυματομορφή εξόδου κατάλληλη για τον ακριβή έλεγχο ενός μικροηλεκτρονικού συστήματος.



Εικόνα 2.2.7 PWM SIGNAL DUTY CYCLE πηγή apo arduino.cc.

Κατά την περίοδο του σήματος PWM το αναμμένο τμήμα της συχνότητας ονομάζεται Duty Cycle και μετριέται σε ποσοστό (%) επί της περιόδου. Το Duty Cycle ισούται T-ON δια του T-Off + T-On. Θεωρούμε πώς γίνεται χρήση της πλακέτας Arduino uno R3 όπου η τάση ρεύματος είναι 5V και θέλουμε να ανάψουμε με PWM ένα LED, Δίνουμε την εντολή analogWrite(led,X) με σκοπό να φωτίσει το Led , στην παραπάνω εικόνα οι πράσινες γραμμές είναι μια κανονική χρονική περίοδος με την κλίμακα να είναι από το 0(0V) μέχρι το 255(5V).

- duty cycle (0%): Έχουμε 0V ρεύματος στο Led άρα το φωτάκι θα είναι πλήρως σβηστό.
- duty cycle (50%): Έχουμε 2.5V ρεύματος στο LED με αποτέλεσμα να είναι το μισό αναμμένο.
- duty cycle (100%): Έχουμε 5v ρεύματος στο LED με αποτέλεσμα να είναι πλήρως αναμμένο.

1.3 Arduino uno.

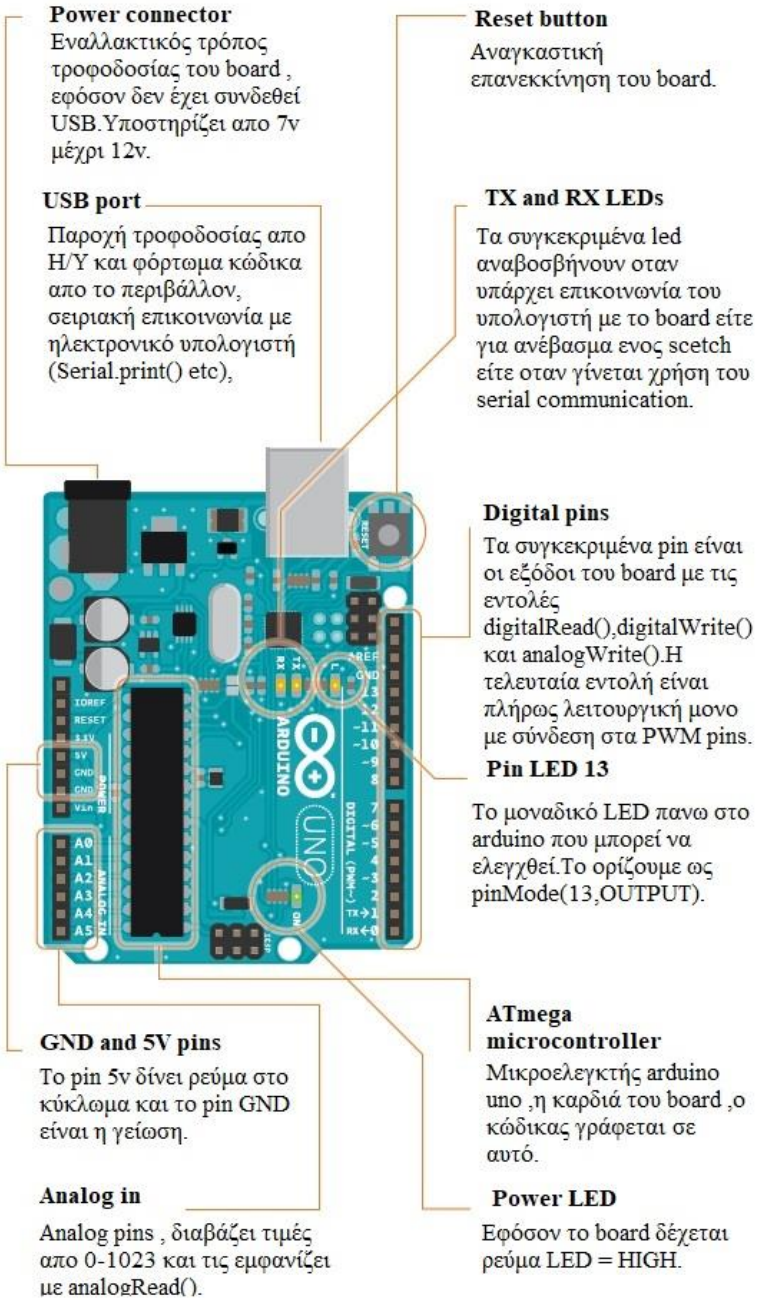
Στην παρούσα εργασία έγινε χρήση του Arduino Uno R3 , η συγκεκριμένη πλακέτα προσφέρει με μικρή υπολογιστική ισχύ απεριόριστες δυνατότητες για δημιουργία και κατασκευή ηλεκτρονικών συσκευών . Το Arduino γενικά χαρακτηρίζεται από την ευκολία προγραμματισμού σε αρχάριους χρήστες και μεγάλη ευελιξία σε πιο έμπειρους.

Εικόνα 2.3 Arduino Uno R3.

Το Arduino uno R3 είναι η πιο γνωστή πλακέτα της εταιρείας αποτελείται από μια απλή μητρική πλακέτα με ενσωματωμένο τον μικροελεγκτή , βασίζεται σε έναν 8-bit RISC μικροελεγκτή ο οποίος είναι συγχρονισμένος στα 16 Mhz τον ATMEGA328. Περιέχει 14 ψηφιακούς ακροδέκτες εισόδου-εξόδου I/O, 6 αναλογικές εισόδους και 6 ψηφιακές εξόδους .Η σύνδεση γίνεται με usb καλώδιο, ενώ υπάρχει και υποδοχή σύνδεσης με ρεύμα, καθώς και δυνατότητα εντός του κυκλώματος, σειριακού προγραμματισμού-ICSP («In-Circuit Serial Programming») και ένα κουμπί επανεκκίνησης σε περίπτωση που βραχυκυκλώσει η πλατφόρμα. Η συγκεκριμένη πλακέτα είναι ικανή να διαχειριστεί μεγάλο πλήθος εφαρμογών με πολλές επιλογές στον χρήστη ως προς την αξιοποίηση των δυνατοτήτων της .

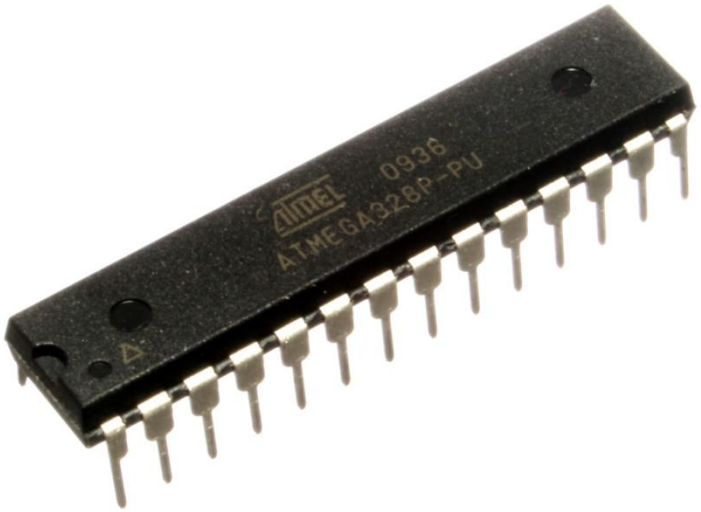
Μικροεπεξεργαστής	**ATmega328**
Τάση Λειτουργίας	5V
Συνιστάμενη Τάση Εισαγωγής	7-12V
Τάση Εισαγωγής(όρια)	6-20V
Ψηφιακές I/O Είσοδοι	14(εκ των οποίων οι 6 μπορούν να παράγουν P.W.M).PWM=11,10,9,6,5,3.
Αναλογικές Είσοδοι	6
Συνεχές Ρεύμα(ανά είσοδο)	40 mA
Μνήμη	32 KB (ATmega328 εκ των οποίων 0.5 KB χρησιμοποιεί ο bootloader)
SRAM	2 KB(ATmega328)
EEPROM	1 KB(ATmega328)
Ταχύτητα Ρολογιών	16 MHz
Βάρος	25 γραμμάρια
Length	68.6 mm
Width	53.4 mm

Πίνακας 2.3 Στοιχεία Arduino.



Εικόνα 2.3b Arduino

1.3.1 Μικροελεγκτής.



Εικόνα 2.3.1 Atmega 328p φωτογραφια απο wikipedia.

Η καρδιά του Arduino είναι ένας μικροελεγκτή της ATMEL AVR ο ATmega328p, έχει ενσωματωμένο bootloader δίνοντας μας την ικανότητα να ανεβάζουμε γρήγορα και εύκολα στην πλακέτα τα sketch χωρίς την χρήση εξωτερικού προγραμματιστή. Η επικοινωνία του γίνεται με το πρωτόκολλο STK500 με μνήμη flash 32KB στην οποία τοποθετείτε κάθε φόρα το sketch που δημιουργεί ο προγραμματιστής παραμένοντας και μετά την απενεργοποίηση της πλακέτας , από τα οποία τα 0.5KB είναι η μνήμη του bootloader .Διαθέτει 2KB στατικής μνήμης (SRAM) η οποία χρησιμοποιείτε για την προσωρινή αποθήκευση στατικών-μεταβλητών δεδομένων του προγράμματος που εκτελείτε και 1KB EEPROM όπου αποθηκεύονται οι τιμές των μεταβλητών όταν η πλακέτα δεν βρίσκετε σε λειτουργία ή έχει γίνει επανεκκίνηση (RESET). Είναι εξοπλισμένο με έναν κρυσταλλικό ταλαντωτή(Crystal Oscillator) που είναι χρονισμένος στα 16MHz ο οποίος βοηθάει στην λειτουργία του κυκλώματος . Στην περίπτωση που γραφεί κώδικας άνω του 70% της μνήμης το Arduino θα μας ειδοποιήσει ότι μπορεί να υπάρχει αστάθεια στο κύκλωμα.

1.3.2 Είσοδος/έξοδος.

Η συγκεκριμένη πλακέτα στα ψηφιακά pin έχει τοποθετημένα 14 θηλυκά pin αριθμημένα από το 0 μέχρι το 13, αυτά τα pin λειτουργούν σαν ψηφιακές είσοδοι/έξοδοι με ρεύμα έντασης 20mA(μπορεί να φτάσει και 40mA υπό προϋποθέσεις) με ασφάλεια και τάση λειτουργίας τα 5V,συμπεριλαμβάνονται pin(11,10,9,6,5) που λειτουργούν ως έξοδος του αναλογικού σήματος μέσω PWM οπού είναι σημειωμένες με το σύμβολο «~» .Επίσης υπάρχουν 6 Analog pins εισόδου τα A0,A1,A2,A3,A4,A5. Το καθένα από αυτά λειτουργεί σαν αναλογική είσοδος με χρήση ADC που είναι ενσωματωμένο στον μικροελεγκτή .

1.3.3 Τροφοδοσία .

Η τάση λειτουργίας είναι τα 5V DC τα οποία μπορούνε να προέρχονται από καλώδιο USB συνδεμένο σε ηλεκτρονικό υπολογιστή ή εξωτερική τροφοδοσία (μπαταρίες, μετασχηματιστή). Χάρης του γραμμικού ρυθμιστή τάσης 5v που διαθέτει μπορεί να συνδεθεί τάση εισόδου από 6-20V αλλά οι προτεινόμενες τιμές είναι από 7-12V DC. Έχει ενσωματωμένο σύστημα πολυασφάλειας έτσι ώστε να προστατεύει τις θύρες USB του υπολογιστή προστατεύοντας τον από βραχυκύκλωμα και υπέρταση ,η ασφάλεια αυτή ενεργοποιείται όταν περάσουν περισσότερα από 500mAh οπού διακόπτετε αμέσως η σύνδεση.

Η πλακέτα διαθέτει μια ομάδα από Pin με την ονομασία POWER που αφορούν αποκλειστικά την τροφοδοσία:

- Στο 1ο Pin με την ένδειξη Vin έχουμε τάση ίση με την τάση τροφοδοσίας δηλαδή αν η σύνδεση γίνει με το καλώδιο USB θα έχουμε 5V,σε συνδυασμό με μια γείωση μπορεί να χρησιμοποιηθεί σαν κύρια παροχή ρεύματος, εφόσον δεν έχει συνδεθεί καμία άλλη παροχή ρεύματος.
- Το 2ο και 3ο Pin με την ένδειξη GND είναι γειώσεις του κυκλώματος.
- Το 4ο Pin με την ένδειξη 5V, τροφοδοτεί τα εξαρτήματα που θα συνδεθούν με 5V. Στην περίπτωση που συνδεθεί απευθείας από την θύρα USB η τροφοδοσία είναι άμεση και σταθεροποιημένη στην αντίθετη περίπτωση που η τροφοδοσία γίνει με εξωτερική τροφοδοσία το ρεύμα θα περάσει από τον ρυθμιστή τάσης μετατρέποντας το ρεύμα σε 5v.
- Το 5ο με την ένδειξη 3.3V, τροφοδοτεί το κύκλωμα με 3.3V με μέγιστη ένταση τα 50mA.
- Το 6ο Pin με την ένδειξη reset όταν γειωθεί με οποιοδήποτε από τους 3 ακροδέκτες με την ένδειξη GND που βρίσκονται στη πλακέτα του Arduino έχει ως αποτέλεσμα την επανεκκίνηση της πλακέτας.
- Το 7ο Pin με την ένδειξη μας δίνει την τάση λειτουργίας του μικροελεγκτή 5V DC ή 3.3V DV.

1.3.4 Σειριακή επικοινωνία.

Τα δυο αρχικά pin του DIGITAL 0(RX) και 1(TX) αφορούν την σειριακή επικοινωνία με λειτουργία την αποστολή και την λήψη δεδομένων. Στην πλακέτα υπάρχουν και δυο αντίστοιχα LED αυτά μας ειδοποιούν ανάλογα με την λήψη η την αποστολή δεδομένων. Αυτές οι θύρες μπορούν να χρησιμοποιηθούν και σαν digital pins με δήλωση τιμής GPIO0 και GPIO1 απλά ο χρήστης θα πρέπει να βεβαιωθεί ότι δεν είναι συνδεμένα κατά την διάρκεια φόρτωσης του κώδικα στην πλακέτα.Μέσω των Pins SDA και SCL το Arduino μπορεί να υποστηρίξει I2C επικοινωνία με άλλες συσκευές που υποστηρίζουν το ίδιο πρωτόκολλο, η σύνδεση μπορεί να γίνει επίσης από τα Pin A4-A5 στο ANALOG IN.

1.3.5 Button and LEDs.

Η συγκεκριμένη πλακέτα έχει ένα κουμπί που εκτελεί επανεκκίνηση της πλακέτας, ένα LED με τη σήμανση POWER για την παροχή της τροφοδοσίας. Περιλαμβάνει και ένα ενσωματωμένο LED που είναι συνδεμένο με το digital Pin 13 ,

με την εντολή digitalWrite(13,HIGH) το LED θα πάρει τιμή HIGH άρα θα ανάψει. Υπάρχει το LED με την σήμανση L που αναβοσβήνει συνήθως για δοκιμαστικό σκοπό και τα RX-TX που αφορούν την σειριακή επικοινωνία .

1.4 Λειτουργικά μέρη Arduino.

Κάθε είδους μικροεπεξεργαστής διαθέτει μονάδες εισόδους που σε αυτές ο χρήστης μπορεί να δεχτεί κάποια δεδομένα να τα επεξεργαστεί και να τα προωθήσει σε εξόδους με τις κατάλληλες εντολές. Τα κυκλώματα με Arduino μπορούν να γίνουν αρκετά πολύπλοκα καθώς μπορούμε να χρησιμοποιήσουμε διάφορες επεκτάσεις(Arduino shields) σε συνδυασμό με διάφορα μικροηλεκτρονικά εξαρτήματα που έχουν ως λειτουργία την συλλογή πληροφορίων με σκοπό την εκμετάλλευση τους από τον χρήστη. Στο εμπόριο υπάρχουν άπειρες επεκτάσεις για το Arduino και πακέτα starter kits που περιλαμβάνουν όλα τα απαραίτητα για έναν αρχάριο χρήστη με μικρό κόστος , ο αγοραστής μπορεί να τα προμηθευτεί από παντού με πολύ μεγάλη γκάμα σε διάφορες τιμές ανάλογα με την ποιότητά τους. Ένα ολοκληρωμένο starter kit περιλαμβάνει jumper wires , breadboard ,αντιστάσεις , κουμπιά , LEDS. Ανάλογα με το ποσό που έχει στην διάθεσή του ο χρήστης υπάρχουν και πακέτα που περιέχουν LCD οθόνες , servo motor ,steper motor και διάφορους επιπλέον αισθητήρες για πιο εξειδικευμένα projects .

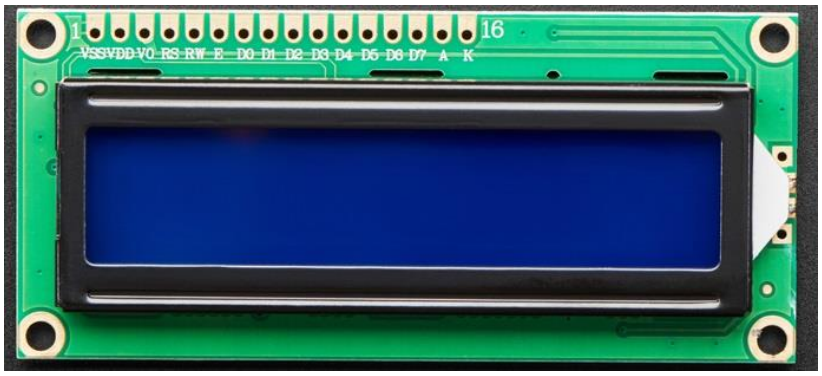
1.4.1 Breadboard.

Το breadboard αποτελείται από Pins χωρισμένα σε ομάδες που συνδέουν την πλακέτα του arduino με τα διάφορα εξαρτήματα που μπορούμε να χρησιμοποιήσουμε, η σύνδεση γίνεται μέσω jumper wires. Τα μεγάλα breadboard έχουν την ιδιαιτερότητα πως χωρίζονται στην μέση με μόνο τρόπο επικοινωνίας αυτών των πλευρών να είναι η γεφύρωση τους με καλώδια. επίσης στις άκρες του υπάρχουν δυο στήλες (+,-) για την τροφοδοσία ρεύματος , συνδέουμε κατάλληλα την τάση-γείωση σε ολόκληρο το κύκλωμα με την ιδιαιτερότητα πως η γείωση θα πρέπει να είναι ίδια σε όλα τα εξαρτήματα και τους τρόπους τροφοδοσίας του κυκλώματος. Όλες οι γραμμές από την ίδια πλευρά επικοινωνούν μεταξύ τους .

1.4.2 Ποτενσιόμετρα.

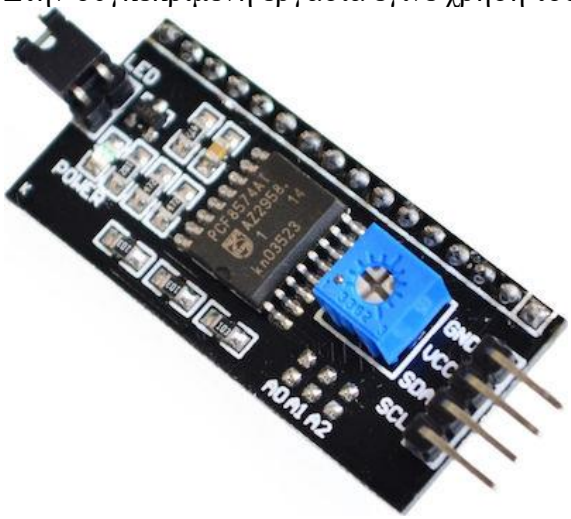
Το ποτενσιόμετρο είναι μια μεταβαλλόμενη αντίσταση της τάξης των 10.000 Ω με μέγιστη τάση τα 200V . Η μεταβολή αυτή ελέγχεται από τον χρήστη γυρνώντας την περιστρεφόμενη ροδέλα που διαθέτει ελέγχοντας έτσι με αυτή την τάση που δέχεται . Γενικά τα ποτενσιόμετρα χρησιμοποιούνται σε συσκευές για τον έλεγχο της έντασης του ήχου ,στον κόσμο του Arduino μπορεί να παραχθεί PWM παλμός δίνοντας την δυνατότητα στον χρήστη του άμεσου ελέγχου της τάσης από την ροδέλα .

1.4.3 Liquid Crystal Displays (LCD).



Εικόνα 2.4.3α 16x2 πηγή : https://www.adafruit.com/product/181

Η οθόνη υγρών κρυστάλλων LED LCD 16x02 μπορεί να εμφανίσει 16 χαρακτήρες σε 2 γραμμές ,για την εμφάνιση των δεδομένων χρησιμοποιεί οπίσθιο φωτισμό που πρέπει να δηλωθεί στο setup() του προγράμματος για να λειτουργήσει αλλιώς θα εμφανίζει κενό .Η οθόνη περιέχει μια σειρά από 16 θύρες προς κόλληση στα οποία μπορεί να τοποθετηθεί μια σειρά από pins με ειδική θερμοκόλληση (κολάι) έτσι ώστε να είναι σταθερό και να μπορεί να γίνει η σύνδεση με την πλακέτα. Στην συγκεκριμένη εργασία έγινε χρήση του I2C serial module για LCD.



Εικόνα 2.4.3β i2c module φωτογραφία απο επίσημο σαιτ εμπόρου.

Το Arduino uno R3 διαθέτει 13 pins προς χρήση ,η παραπάνω συνδεσμολογία απαιτεί την σύνδεση 6 καλωδίων από την οθόνη στην πλακέτα καταλαμβάνοντας Pins που θα μπορούσαν να συνδεθούνε για διαφορετικά ηλεκτρονικά components για την επέκταση του προγράμματος και συνολικά θα χρειαστούν 12 καλώδια .Για την αποφυγή του παραπάνω προβλήματος έγινε χρήση ενός I2C module. Συνολικά διαθέτει 20 αρσενικά pins ,16 από αυτά που τα pins του κολλήθηκαν στις θύρες οθόνης με ειδική κόλληση (κολάι) και άλλες 4 θύρες καλωδίων για την σύνδεση του με την πλακέτα.. Στην πίσω πλευρά έχει ένα μικρό ποτενσιόμετρο που γίνεται ο έλεγχος της αντίθεσης της οθόνης ,διπλά περιέχει ένα ολοκληρωμένο κύκλωμα 8-bit I/O PCF8574 που μετατρέπει τα σειριακά δεδομένα του πρωτοκόλλου I2C σε παράλληλα δεδομένα για την επικοινωνία και την αλληλεπίδραση του χρήστη με την

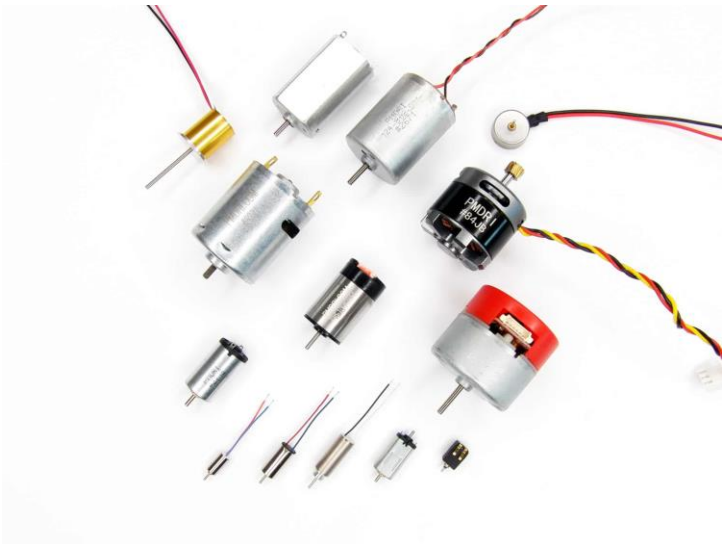
οθόνη .Τέλος έχει 2 αρσενικά pin βραχυκυκλωμένα μεταξύ τους για τον οπίσθιο φωτισμό της οθόνης .

1.4.4 Servo.

Εικόνα 2.4.4 Servo sg90 micro tower πηγή : amazon.com

Τα servo χρησιμοποιούνται για έλεγχο ενός αντικειμένου που απαιτεί τον ακριβή χειρισμό των λειτουργιών, ο συγκεκριμένος έλεγχος μπορεί να ελεγχθεί με ποτενσιόμετρο. Αποτελείτε από έναν σερβομηχανισμό που τροφοδοτείται από το ένα σήμα PWM δίνοντας συγκεκριμένη κίνηση στους εσωτερικούς μηχανισμούς που είναι συνδεμένοι μεταξύ τους με τελικό αποδέκτη έναν άξονα , ανάλογα με το μοντέλο έχει διαφορά ως προς την ταχύτητα ,την ροπή και την ποιότητα κατασκευής του μπορεί να έχει μεταλλικό άξονα. Η ταχύτητα αφορά τον χρόνο που απαιτεί για κίνηση του άξονα σε 60 μοίρες πχ. το μοντέλο sg90 micro tower έχει την ικανότητα να διανύσει τις 60 μοίρες σε ¼ του δευτερολέπτου , ως προς την ροπή εξαρτάται από την ποσότητα της δύναμης του σερβομηχανισμού που μπορεί να εφαρμόσει σε έναν μοχλό . Για την σύνδεση του στην πλακέτα περιλαμβάνει 3 Pin , ανάλογα με το μοντέλο τα καλώδια έχουν διαφορετικό χρωματισμό οπότε θα πρέπει να συμβουλευτούμε το datasheet του αντίστοιχου μοντέλου. Στο συγκεκριμένο μοντέλο η γείωση είναι καφέ , η τάση κόκκινη και το Pin που συνδέεται πάντα σε PWM της πλακέτας είναι πορτοκαλί.

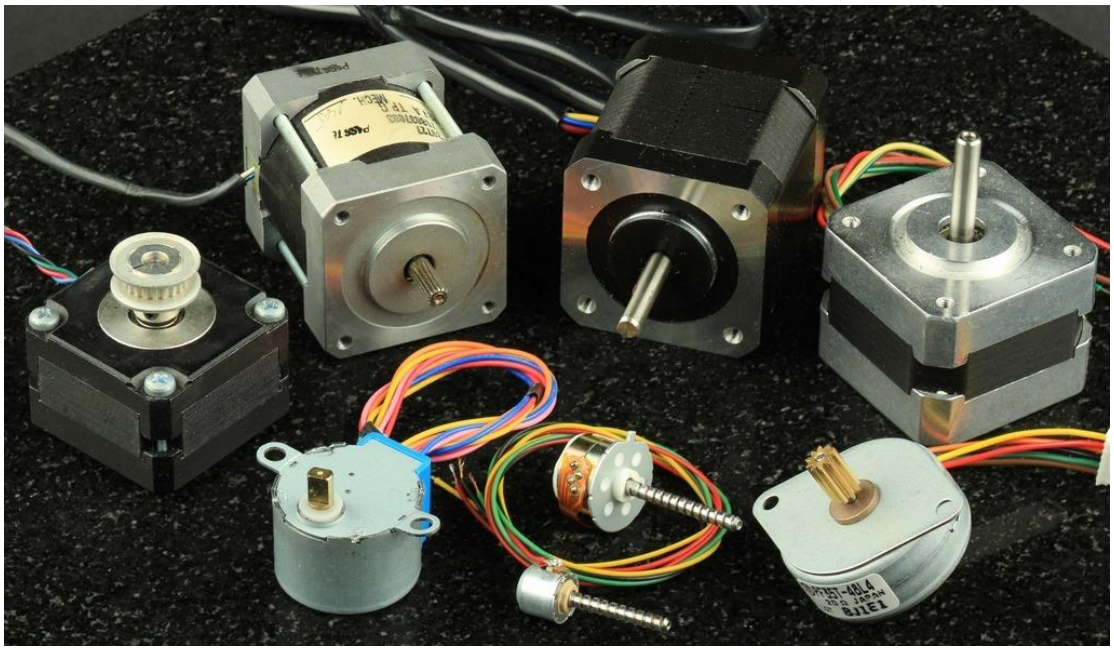
1.4.5 DC motors.



Εικόνα 2.4.5 DC motors πηγήhttps://www.precisionmicrodrives.com/motors/dc-motors

Οι κινητήρες DC motor είναι μια συσκευή συνεχούς ρεύματος που βρίσκει χρήση σε robot, αεροπλάνα ,βάρκες . Στο εσωτερικό ενός τέτοιου κινητήρα βρίσκεται ένα πηνίο που είναι περικυκλωμένο από σταθερούς μαγνήτες , το μέγεθός αυτών εξαρτάται από την δύναμη που θα έχει .Εξαιτίας του σταθερού μαγνήτη υπάρχει αντίστροφη μαγνητική πολικότητα με αποτέλεσμα τη δημιουργία ενός μαγνητικού πεδίου . Όταν γίνει τροφοδοσία με παλμό από θύρα PWM του μικροελεγκτή στο πηνίο τότε βρίσκεται σε μια σταθερή αλληλεπίδραση με τους μόνιμους μαγνήτες δημιουργώντας κίνηση ανάλογη του σήματος. Για την σύνδεση τους έχουν ένα καλώδιο ρεύματος και ένα της γείωσης . Μπορούν να χρησιμοποιηθούν τρανζίστορ για να δημιουργηθεί σήμα που θα συνδεθεί στην πλακέτα , είναι απαραίτητο να δημιουργηθεί μια αντίστροφη πολικότητα με μια δίοδο ZENER σε συνδυασμό με το τρανζίστορ για να μην υπάρξει βλάβη στην πλακέτα . Σε κυκλώματα με μεγάλες τάσεις ρεύματος είναι ιδανικό να χρησιμοποιηθεί κάποιού είδους driver για την σύνδεσή του ανάλογα με τα volt που απαιτεί ο μηχανισμός για την κίνησή του .

1.4.6 Stepper motors.



Εικόνα 2.4.6 steppers motors πηγή : adafruit.com

Τα steppers motors είναι μηχανισμοί βηματικού ελέγχου με επαναληπτική λειτουργία , μετά την εκτέλεση ενός βήματος ο άξονας γυρνάει σε συγκεκριμένη θέση . Η μετακίνηση του άξονα γίνεται με σταθερό βήμα από ηλεκτρομαγνήτες που ελέγχονται με την τροφοδοσία ρεύματος ενός σήματος PWM μέσω μιας αντίστασης. Υπάρχουν δύο κατηγορίες stepper motors που διαφέρουν ανάλογα με τον τύπο της αντίστασης που χρησιμοποιούνται . Τα bipolar steppers απαιτούν 4 καλώδια για την σύνδεση με πλακέτα, στο εσωτερικό τους παράγουν κίνηση στον άξονα μέσω της αντίστροφης πολικότητας και σε συνδυασμό με την πλήρη εκμετάλλευση της αντίστασης έχουν υψηλή ροπή με χαμηλή ταχύτητα. Τα unipolar απαιτούν 5ή 6 καλώδια για την σύνδεση τους με πλακέτα, έχουν χαμηλή ροπή καθώς χρησιμοποιείται ένα μέρος της αντίστασης με αποτέλεσμα να έχουν υψηλότερη ταχύτητα και απλούστερο χειρισμό .Για την σύνδεση τους απαιτείται driver , για μικρά stepper motor μπορεί να γίνει χρήση ενός UN2003a.

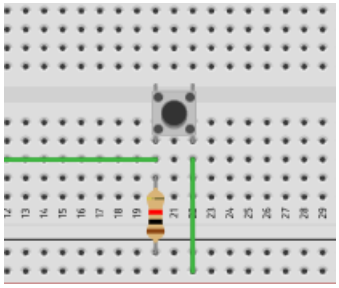
1.4.7 Buttons και Led.

Τα λαμπάκια led έχουν την ικανότητα να ανάβουν όταν τροφοδοτούνται με ρεύμα από το κύκλωμα, μπορούν να χρησιμοποιηθούν και με PWM παλμό ελέγχοντας έτσι την φωτεινότητα τους .Τα button είναι κουμπιά που μπορούν να χρησιμοποιηθούν για συγκεκριμένες λειτουργίες που τις καθορίζει ο προγραμματιστής ανάλογα με τις ανάγκες του προγράμματος .Τα buttons χρησιμοποιήθηκαν σε όλες τις δοκιμές που έγιναν για τον έλεγχο της σωστής λειτουργίας του προγράμματος . Η σύνδεση γινόταν πάντα σε συνδυασμό με μια αντίσταση 10κΩ. Στην χρήση του button με σκοπό την τροφοδοσία ενός led χωρίς αντίσταση παρατηρείτε ότι όταν ασκείται πίεση στο κουμπί γίνεται μεταφορά 5V, το πρόγραμμα αναγνωρίζει ένα HIGH σήμα με αποτέλεσμα να τροφοδοτήσει με ρεύμα

το led , με το που σταματήσει η πίεση στο κουμπί χάνεται το ρεύμα αλλά υπάρχει διαρροή ρεύματος οπότε το led αναβοσβήνει. Υπάρχουν δύο επιλογές για την λύση αυτού του προβλήματος η σύνδεσης των κουμπιών με pullup αντίσταση ή pulldown ανάλογα με το πώς γίνεται η σύνδεση της αντίστασης.

Pulldown:

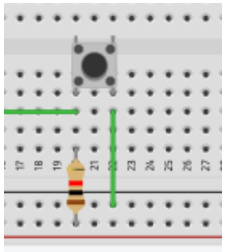
Με την χρήση αντίστασης στην γείωση όταν ασκείτε πίεση στο κουμπί το ρεύμα διοχετεύεται από τα 5V στην αντίσταση και μετά στην γείωση του arduino αυτό έχει ως αποτέλεσμα το λαμπάκι να ανάβει μόνο όταν πιέζεται το κουμπί.



Εικόνα 2.4.7α PullDown.

Pullup:

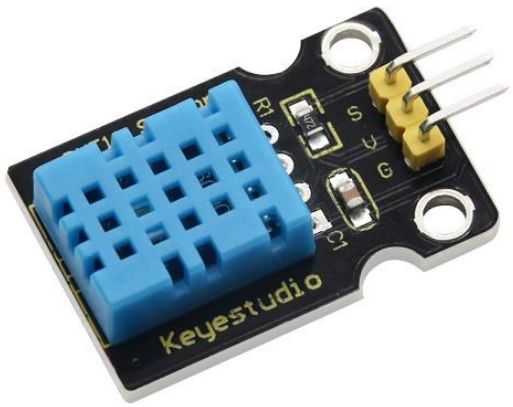
Με την χρήση αντίστασης στην τάση όταν δεν ασκείτε πίεση στο κουμπί διαπερνάει την αντίσταση HIGH ρεύμα και όταν πιέζεται το κουμπί το ρεύμα περνάει προς την γείωση. Το λαμπάκι δηλαδή είναι σε λειτουργία φωτισμού μέχρι να πατηθεί το κουμπί που σβήνει.



Εικόνα 2.4.7α PullUp.

1.4.8 Αισθητήρας DHT11.

Ο DHT11 είναι αισθητήρας θερμοκρασίας και υγρασίας . Ο αισθητήρας μπορεί να εμφανίσει από 0 μέχρι 50 βαθμούς κελσίου με απώλεια 2 βαθμών και από 20 μέχρι 90% υγρασία με απώλεια 5%. . Ο συγκεκριμένος αισθητήρας αποτελείτε από ένα θερμίστορ που είναι ένας τύπος αντίσταση που η τιμή της επηρεάζεται από την θερμοκρασία και ένα στρώμα ενωμένο με δύο αντιστάσεις που συγκρατεί την υγρασία και μας εμφανίζει το ποσοστό της υγρασίας.



Εικόνα 2.4.8,DHT11 πηγή : wiki.keyestudio.com.

1.4.9 Transistors και δίοδοι zener.

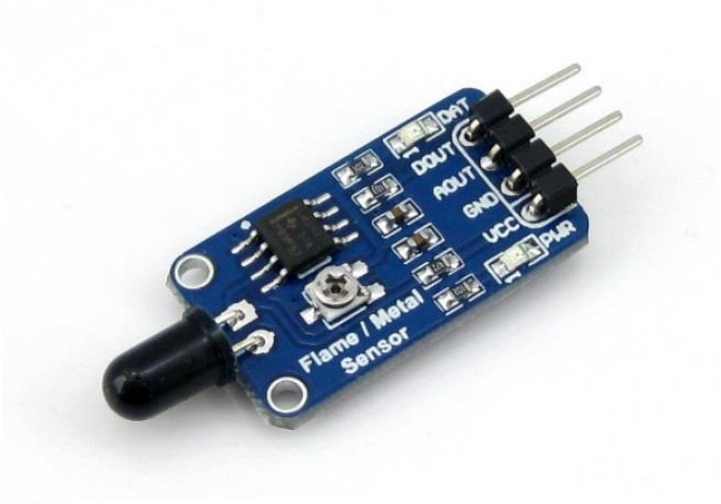
Τα τρανζίστορ είναι ηλεκτρονικά εξαρτήματα που χρησιμοποιούνται σε ένα κύκλωμα για την ενίσχυση-σταθεροποίηση της τάσης και την διαμόρφωση της συχνότητας . Διαθέτουν 3 pins το ένα της τάσης ,το άλλο της γείωσης και το άλλο είναι η βάση για την σύνδεση με την πλακέτα. Χωρίζονται σε δύο κατηγορίες τα BJT (Bipolar Junction Transistors) για χρήση σε μικρά φορτία ρεύματος (μικρότερα του 1AMP) με συχνότητα ελέγχου έως 100 KHz ιδανικά για χρήση με stepper motors και τα MOSFET (Metal-Oxide Semiconductor Field Effect Transistors) για μεγαλύτερα φορτία ρεύματος με την συχνότητα του να φτάνει τα 500KHz ιδανικά για Dc motors. Η χρήση τρανζίστορ έγινε για την εκκίνηση ενός DC motor ελέγχοντας την ταχύτητά του . Μαζί με τα τρανζίστορ συνδέθηκε μια δίοδος ζένερ η ,συγκεκριμένη δίοδος έχει σχεδιαστεί για την χρήση της σε ένα κύκλωμα έτσι ώστε να δημιουργεί αντίστροφη πόλωση στην περιοχή ης κατάρρευσης. Όταν η διοδος έχει συνδεθεί με αντίστροφη πόλωση έχει την ιδιαιτερότητα να συμπεριφέρεται σαν απλή δίοδος μέχρι να φτάσει στο σημείο που το ανάστροφο ρεύμα έχει γρήγορη αύξηση τάσης σε αντίθεση με την τάση στα άκρα της που είναι σταθερή ,είναι σχεδιασμένη έτσι ώστε να λειτουργεί συνεχώς στην συγκεκριμένη περιοχή ,επίσης σε ορθή σύνδεση λειτουργεί σαν μια απλή δίοδος.

1.4.10 Shields .

Τα shields είναι επεκτάσεις που συνδέονται πάνω στα Pins της πλακέτας , είναι απαραίτητα εξαρτήματα για εξειδικευμένα κυκλώματα που προσφέρουν απεριόριστες δυνατότητες ανάλογα με της ανάγκες του project .Τα shields είναι ολοκληρωμένες πλακέτες που σχεδιάστηκαν για να προεκτείνουν την λειτουργία και την επεκτασιμότητα της πλακέτας . Υπάρχουν shields που μπορούν να συνδέσουν την πλακέτα με το διαδίκτυο μέσω ethernet ή Wifi (ethernet shield , WIFI shield), να

προσθέσουν οθόνη υψηλής ανάλυσης , να συνδεθούν με GPS για λειτουργία εντοπισμού θέσεις (GPS shield), να ελέγξουμε διάφορους ηλεκτροκινητήρες ως προς την κατεύθυνση και την ταχύτητα τους με μια πλακέτα (DV motors ,steppers , servos).Υπάρχει μεγάλη γκάμα από επεκτάσεις για το Arduino ο χρήστης ανάλογα με τις απαιτήσεις που έχει μπορεί να επιλέξει τις κατάλληλες για αυτόν .

1.4.11 Flame sensor.



Εικόνα 2.4.11 Αισθητήρας φλόγας πηγή :waveshare.com

Flame sensor ή αισθητήρας φλόγας είναι ένας αισθητήρας που σχεδιάστηκε για ανιχνεύει φλόγα όταν βρίσκεται στο πεδίο των IR υπέρυθρων. Οι υπέρυθρες έχουν ευαισθησία της τάξης των 750 nm έως 1100nm και όταν εκπέμπει η φλόγα σε κοντινή απόσταση των 100cm το αντιλαμβάνεται εντός της γωνίας των 60 μοιρών .

1.4.12 Buzzer.

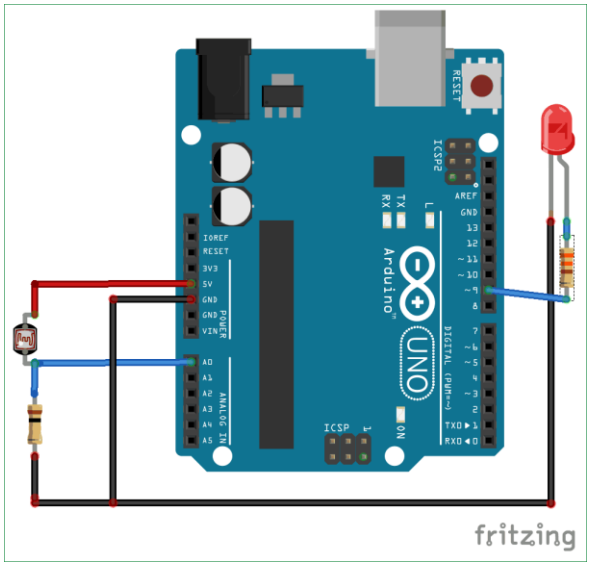
Εικόνα 2.4.12 buzzer πηγή skroutz.

Το buzzer είναι γνωστό και σαν piezo buzzer είναι μια μικρή συσκευή ηχείου που μπορεί να συνδεθεί με το Arduino .

1.4.13 Light Dependent Resistor

Light Dependent Resistor (LDR) ή Photoresistor σημαίνει φωτοαντίσταση, το συγκεκριμένο εξάρτημα είναι μια αντίσταση με την ικανότητα να μεταβάλει την τιμή της ανάλογα με το φως που την διαπερνάει, δηλαδή λειτουργεί σαν είσοδος οπότε την

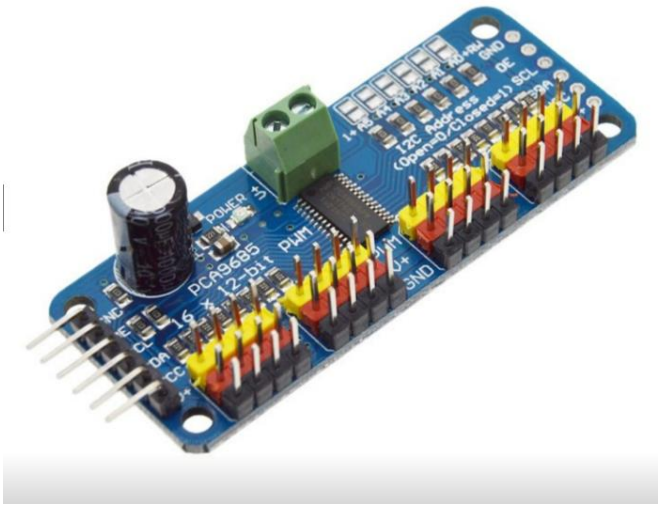
συνδέουμε στο ANALOG IN καθώς λαμβάνει συνεχώς τιμές. Όσο περισσότερο φως διοχετεύεται σε αυτόν τον αισθητήρα τόσο μικραίνει η αντίστασή του ανάλογα με την φωτεινότητα που δέχεται μεταβάλλονται και τα ohms της αντίστασής του .Η κύρια χρήση αυτού του εξαρτήματος είναι η δημιουργία προγράμματος με σκοπό να διαβάζει αν είναι μέρα ή νύχτα. Είναι απαραίτητο να συνδεθεί αντίσταση , στο σκοτάδι χρησιμοποιείται χαμηλή τιμή αντίστασης 10K ενώ στο φως πιο υψηλή τιμή αντίστασης.



Εικόνα 2.4.13 LDR με LED φωτογραφία από FRITZING

1.4.14 PC9682 Pwm/Servo driver

Όταν θέλουμε να ελέγξουμε πόσα volts θα διαπεράσουν μια έξοδο (πχ led) για να γίνει έλεγχος της κατάστασης τότε χρησιμοποιούμε έναν παλμό , αυτό στο Arduino γίνεται μεσώ των θυρών της πλακέτας που έχουν την ένδειξη PWM ,Αυτές οι θύρες είναι περιορισμένες ανάλογα με το μοντέλο της πλακέτας , στο uno R3 έχει 6 Pin PWM με αποτέλεσμα να υπάρχει περιορισμός στις 6 εξόδους. Η σύνδεση servo στην πλακέτα απαιτεί την χρήση της αντίστοιχης βιβλιοθήκης <servo.h> , όταν συνδεθούν πολλές συσκευές servo και συνδυάζονται με άλλους αισθητήρες δημιουργείται πρόβλημα μεταξύ τους καθώς η βιβλιοθήκη του servo χρησιμοποιεί τους χρόνους από την πλακέτα.



Εικόνα 2.4.14PCA9685 πηγή : amazon.com

Το πρόβλημα με τη διαθεσιμότητα των Pins το λύνει το PCA9685 που είναι ένα module χρήσης I2C πρωτοκόλλου που αποτελείται από 16 PWM κανάλια ,12 bits κωδικοποίησης με 4096 θέσεις σήματος για έλεγχο του παλμού. Το συγκεκριμένο module είναι κατάλληλο να διαχειριστεί όλες τις εξόδους που έχουν ανάγκη για PWM παλμό (servo,led,steppers κτλ) και λειτουργεί σαν drivers για τα servos με την πλήρη διαχείριση αυτών. Η πλακέτα του Arduino δίνει παλμό με 50Hz συχνότητας ενώ το module 60Hz συχνότητας με πιο τετραγωνισμένο παλμό. Είναι λογικής I2C οπότε θα συνδεθεί στις αντίστοιχές θύρες της πλακέτας SDA και SCL, το VCC θα συνδεθεί στα 5V και το GND στην γείωση της πλακέτας , η θύρα OE στην περίπτωση που γίνει HIGH αποσυνδέει τα Outputs ,είναι δηλωμένη LOW με αποτέλεσμα να μην χρειάζεται αλλαγή. Διαθέτει αυτόνομη τροφοδοσία 5v με δυο διαθέσιμες θύρες , η μία θύρα είναι η V+ και η άλλη βρίσκεται στην μέση της πλακέτας η σύνδεση γίνεται με δυο καλώδια που ασφαλίζουν με βίδες. Προτείνετε η χρήση της δεύτερης θύρας καθώς είναι προστατευμένη με αντίστροφη πολικότητα (revers polarity) και στην περίπτωση που γίνει βραχυκύκλωμα κόβεται το ρεύμα αυτόματα . Η συγκεκριμένη πλακέτα είναι εξοπλισμένη με solder pads που έχουν την ικανότητα να μεταβάλουν την διεύθυνση I2C με αποτέλεσμα να μπορούν να συνδεθούν και άλλα modules διευρύνοντας έτσι τις θύρες PWM, υπάρχει η δυνατότητα επέκτασης μέχρι και 62 module που σημαίνει πως δίνεται η δυνατότητα διαχειρίσεις πάνω από 900 servos.

2 Ολοκληρωμένο περιβάλλον ανάπτυξης (IDE).

Πρώτο βήμα μετά την απόκτηση του Arduino είναι να εγκαταστήσουμε οποιοδήποτε πρόγραμμα επεξεργασίας κειμένου(text editor) και έναν μεταγλωττιστή(compiler) .Compiler είναι ένα πρόγραμμα που διαχειρίζεται τον κώδικα που του δόθηκε και τον μετατρέπει σε γλώσσα μηχανής (ο υπολογιστής

καταλαβαίνει μόνο 0 ή1). Ανεξάρτητα από τον text editor το IDE προσφέρει ένα περιβάλλον προσαρμοσμένο για τον έλεγχο του μικροελεγκτή με όλα τα απαραίτητα εργαλεία συγκεντρωμένα .Ένα περιβάλλον IDE που δημιουργήθηκε με σκοπό τον έλεγχο κάποιου μικροελεγκτή έχει ενσωματωμένο πρόγραμμα text editor με κάποιες διευκολύνσεις στον χρήστη και compiler για έλεγχο για τυχών λαθών στον κώδικα .Επιπλέον περιέχει ένα φορτωτή (loader) που κάνει την μεταφορά του προγράμματος (sketch) από τον υπολογιστή στην πλακέτα. Το συγκεκριμένο μοντέλο μικροελεγκτή (Arduino uno) χρησιμοποιεί την γλώσσα προγραμματισμού wiring(συνδυασμός C και C++).Υπάρχει πληθώρα επιλογή σε εφαρμογές συγγραφής κώδικα , η ίδια εταιρεία που κατασκεύασε το Arduino έχει δημιουργήσει μια απλοϊκή εφαρμογή ανοιχτού κώδικα με όλα τα απαραίτητα εργαλεία που θα χρειαστεί ένας προγραμματιστής για να ξεκινήσει να προγραμματίζει την πλακέτα δίχως. Ο χρήστης έχει την δυνατότητα σύνδεσης της πλακέτας σε οποιαδήποτε περιβάλλον ανάπτυξης IDE (Integrated development environment).

2.1 Γλώσσα προγραμματισμού.

Η συγκεκριμένη γλώσσα διαθέτει όλες τις λειτουργίες από C/C++ .Η παρούσα εργασία θα επικεντρωθεί σε γλώσσα προγραμματισμού για μικροελεγκτές με αρχιτεκτονική AVR (ATmega).Οι βασικές εντολές και συναρτήσεις παραμένουν με την λογική της C/C++ χρησιμοποιώντας τους ίδιους τύπους δεδομένων και τους ίδιους τελεστές . Ανάλογα με τον μικροελεγκτή διαφέρουν τα Pin που μπορούν να χρησιμοποιηθούν, καθώς ο καθένας έχει την δικιά του αρχιτεκτονική με συγκεκριμένα τεχνικά χαρακτηριστικά .

2.1.1 Βασική δομή.

Υπάρχουν δύο βασικές συναρτήσεις πού είναι μέρος του κάθε sketch(πρόγραμμα) οι οποίες είναι η setup() και η loop() .Η συνάρτηση setup() θα εκτελεστεί μία φορά στην αρχή και ακολούθως η loop() θα εκτελείτε συνεχώς μέχρι να σταματήσει η τροφοδοσία ή να πατήσουμε το πλήκτρο reset που βρίσκεται στον μικροεπεξεργαστή .Στην περίπτωση του reset γίνεται η παραπάνω διαδικασία από την αρχή και είναι σαν να τρέξαμε για πρώτη φορά το πρόγραμμα. Οι δηλώσεις μεταβλητών χρησιμοποιούνται από όλες της γλώσσες προγραμματισμού για να δηλωθούν τα ονόματα και οι τύποι των μεταβλητών υπάρχουν πολλοί τύποι μεταβλητών μερικοί και πιο συνηθισμένοι είναι οι :int, float ,Boolean,char, string.

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Πίνακας 3.1.1 Συναρτήσεις βασικής λειτουργίας

Η γλώσσα προγραμματισμού του Arduino βασίζεται στην γλώσσα Processing.

setup() : Εδώ βάζουμε τις εντολές που θα τρέξουν μια φορά ,όταν δίνουμε ρεύμα στη μονάδα μας .Μέσα στην συνάρτηση αυτή καθορίζουμε τις μεταβλητές που χρησιμοποιεί το πρόγραμμα, τις θύρες εισόδου εξόδου και τις βιβλιοθήκες που πρόκειται να χρησιμοποιήσουμε

loop(): Η συνάρτηση αυτή αποτελεί τον κύριο ατέρμονα βρόγχο του κώδικα μας στον οποίο γράφουμε το κυρίως πρόγραμμα. Όλο το μπλοκ κώδικα εκτελείτε σε βρόγχο επανάληψης μέχρι να πατήσουμε το πλήκτρο reset έτσι ώστε να φορτώσει το πρόγραμμα μας από την αρχή.

2.1.2 Βασικές δομές ελέγχου.

Οι δομές ελέγχου είναι κάποιες συναρτήσεις οι οποίες χρησιμοποιούνται σε αρκετές γλώσσες προγραμματισμού και μας βοηθούν να κάνουμε κάποιους απαραίτητους ελέγχους για το αν μια κατάσταση είναι αληθής ή ψευδής και το πρόγραμμα να λειτουργήσει ανάλογα

if: Συνάρτηση που χρησιμοποιείται σε συνδυασμό με έναν τελεστή σύγκρισης, ελέγχει αν μια συγκεκριμένη κατάσταση έχει επιτευχθεί, όπως ο έλεγχος μιας εισόδου εάν είναι πάνω από ένα ορισμένο αριθμό και ανάλογα εάν είναι ακολουθείται μια συγκεκριμένη ενέργεια, αλλιώς μια διαφορετική.

if / else: Συνάρτηση που μοιάζει με την «if» με την διαφορά ότι επιτρέπει μεγαλύτερο έλεγχο της ροής του κώδικα, επιτρέποντας πολλαπλές δοκιμές

Επανάληψης (loop):

For ():Στο πρώτο σκέλος γίνεται η δήλωση μεταβλητής ,στο δεύτερο σκέλος έλεγχος συνθήκης και στο τρίτο ορισμός επόμενης τιμής, γίνεται χρήση εφόσον γνωρίζουμε πόσες φορές θα εκτελεστεί ο συγκεκριμένος κώδικας.

While (): Βρόγχος που εκτελείται συνεχώς και απείρως, έως ότου η έκφραση μέσα στην παρένθεση γίνει ψευδής. Ιδανικά χρησιμοποιείτε όταν δεν γνωρίζουμε πόσες φορές θα εκτελεστεί.

Do …..While() : Η κατάσταση εσωτερικά του βρόγχου θα εκτελεστεί σίγουρα μια φορά και μετά γίνεται έλεγχος συνθήκης στο while().

Σταθερές:

Όλες οι σταθερές είναι τύπου integer .

LOW: Έχει την τιμή 0 δηλαδή λογικό false.

HIGH: Έχει την τιμή 1 δηλαδή λογικό true.

Input: Έχει την τιμή 0 δηλαδή λογικό false.

Output: Έχει την τιμή 1 δηλαδή λογικό true.

Arduino functions:

Η κύρια λειτουργία στους μικροελεγκτές είναι ο έλεγχος των θυρών που διαθέτει ο εκάστοτε μικροεπεξεργαστής, ο χρήστης πρέπει να γνωρίζει τις θύρες και τα pin (δίδονται στις επίσημες σελίδες του μικροεπεξεργαστή) για να μπορεί να προγραμματίσει σωστά ένα πρόγραμμα. Στην αρχικοποίηση κάθε προγράμματος (setup) θα πρέπει να δηλώσουμε τα Pins-led-button που χρησιμοποιούμαι ως είσοδο ή ως έξοδο. Πχ η συνάρτηση pinMode(RED_LED,OUTPUT); Ορίζει το κόκκινο LED ως έξοδο και η pinMode(buttonPin,INPUT_PULLUP); Ορίζει το button που είναι συνδεμένο στο 13^{ο} pin του board ως είσοδο (αφού έχουμε δηλώσει στην αρχή του προγράμματος buttonPin=13).

pinMode(pin, mode): Ορισμός ψηφιακού pin σαν είσοδο ή έξοδο , ανάλογα με την τιμή που δίνεται στο mode(INPUT / OUTPUT).

digitalWrite(pin, status): Ορίζει σε συγκεκριμένο Pin μια συγκεκριμένη κατάσταση(HIGH ή LOW).

digitalRead(pin): Επιστρέφει μια συγκεκριμένη (0 ή 1) κατάσταση από συγκεκριμένο pin

analogRead(pin): Επιστρέφει ακέραιο αριθμό από 0 εως 1023, ανάλογα με την τάση που τροφοδοτείται.

analogWrite(pin, value):Θέτει ένα Pin σε λειτουργία PWM(Pulse Width Modulation), η τιμή της value καθορίζει το σήμα που είναι από 0 μέχρι 255.

delay(time): Σταματάει προσωρινά η ροή του προγράμματος ανάλογα με τον ορισμό του time.

Millis (): Επιστροφή millisecond από την έναρξη λειτουργίας του arduino. Η συγκεκριμένη λειτουργία δεν χρησιμοποιεί παραμέτρους.

Σειριακή επικοινωνία:

Το Arduino είναι εξοπλισμένο με μια σειριακή θύρα επικοινωνίας μεταξύ της πλακέτας και του υπολογιστή, η σύνδεση αυτή γίνεται με ένα καλώδιο USB και ενεργοποίηση της σειριακής θύρας επικοινωνίας στην εκτέλεση προγράμματος δίνοντας στην διαδικασία setup() την εντολή Serial.begin(baudRate) όπου baudRate 9600. Η σειριακή θύρα χρησιμοποιείτε είτε για να στείλουμε δεδομένα είτε για να λάβουμε δεδομένα. Με την εντολή Serial.print(“ 12345”) το serial control μας εμφανίζει ‘12345’ δηλαδή μας εμφανίζει την τιμή απτον κώδικα στην οθόνη , όπου Serial.print μπορούμε να χρησιμοποιήσουμε και Serial.println οπού γίνεται εκτύπωση με αλλαγή γραμμής κάθε φορά .

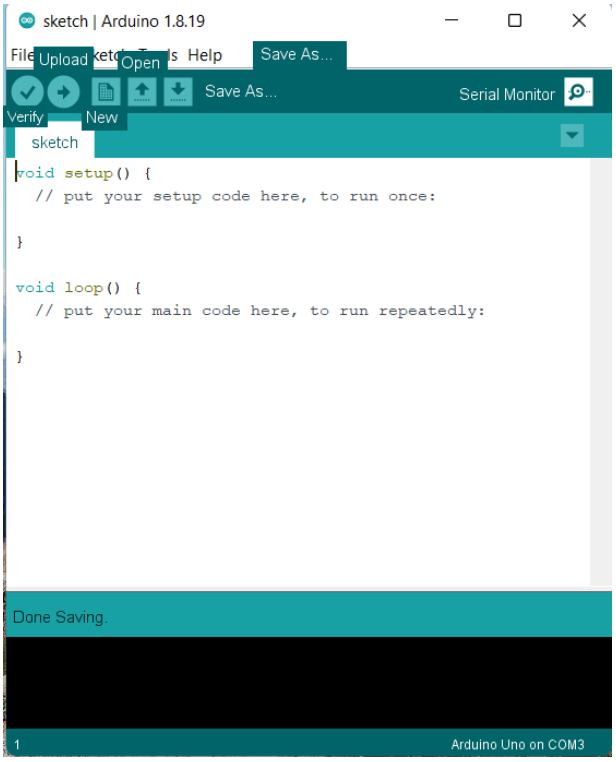
Serial.begin(baudrate): Ρυθμός μεταφοράς δεδομένων μεταξύ υπολογιστή και board.

Serial.print(data): Αποστολή δεδομένων μέσω της σειριακής επικοινωνίας του περιβάλλοντος και εκτύπωση αυτών.

2.2 Αναπτυξιακό Περιβάλλον

2.2.1 Arduino IDE:

Το Arduino IDE είναι μια εφαρμογή κατασκευασμένη από την ιδιά εταιρεία που έβγαλε τις πλακέτες Arduino ,είναι το απλούστερο περιβάλλον ανάπτυξης. Η συγκεκριμένη εφαρμογή υποστηρίζει όλες της πλακέτες τις εταιρείας αλλά και πλακέτες που είναι τύπου Arduino με την ίδιο μικροελεγκτή (Atmel AVR).Είναι ένα πρόγραμμα βασισμένο σε Java με πλήρη χαρακτηριστικά ενός IDE. Τα προγράμματα που δημιουργεί ο χρήστης ονομάζονται sketch ,υπάρχει μεγάλη επιλογή έτοιμων sketch παραδειγμάτων (examples) και έτοιμων βιβλιοθηκών (libraries). Υπάρχει και Library manager (Tools -> manage libraries) οπού βρίσκονται οι βιβλιοθήκες που θα χρειαστούν για την ορθή λειτουργία ενός Project.



Εικόνα 3.2.1 ARDUINO IDE.

Verify: Μεταγλώττιση κώδικα που βρίσκεται στον editor και μετατροπή σε μορφή κατάλληλη για την μεταφορά του στον μικροελεγκτή.

Upload: Αφού ολοκληρωθεί το verify και ο προγραμματιστής έχει επιλέξει το σωστό board (tools ->board -> κατάλληλης συσκευής) προωθεί τον κώδικα στην θύρα που έχει επιλέξει ο προγραμματιστής από το menu->port.

New: Δημιουργία νέου sketch διαγράφοντας τυχόν υπάρχοντα προγράμματα με την δυνατότητα υπενθυμίσεις αποθήκευσης του παλιού αρχείου.

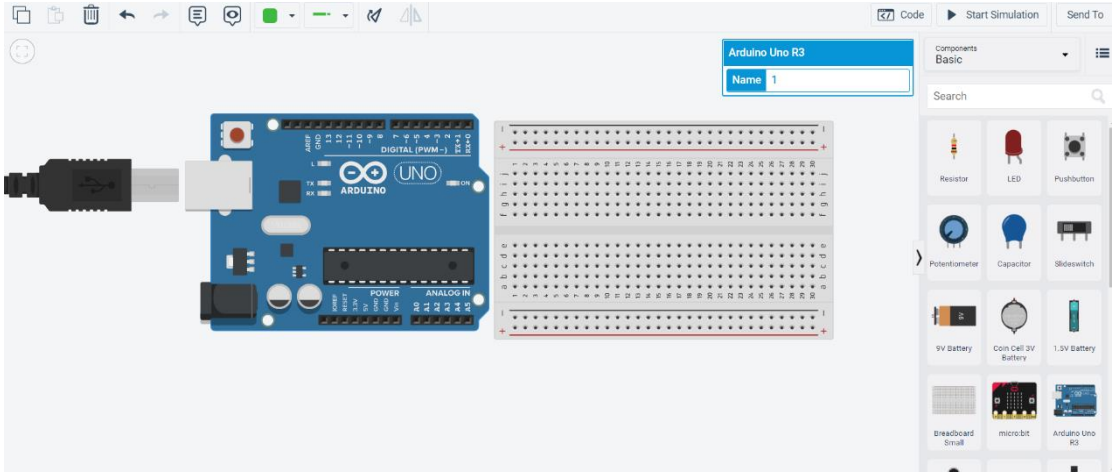
Open: Άνοιγμα sketch που είναι αποθηκευμένο στον υπολογιστή.

Serial monitor: Επικοινωνία του υπολογιστή με το Arduino μέσω σειριακής σύνδεσης. Άνοιγμα παραθύρου όπου εμφανίζονται τα δεδομένα που ο χρήστης έχει επιλέξει να εμφανιστούνε μέσω της εντολής Serial.Print καθώς και δεδομένα που μπορεί να στείλει ο χρήστης προς την πλακέτα.

2.2.2 Tinkercad.

Το tinkercad είναι δωρεάν εφαρμογή από την Autodesk ανοιχτού κώδικα μοντελοποίησης 3D με την δυνατότητα σχεδίασης κυκλωμάτων με πληθώρα επιλογών σε πλακέτες του arduino, την δημιουργία sketch και της αποσφαλμάτωσης (debugging) χωρίς να απαιτεί εγκατάσταση ή να απαιτεί από τον χρήστη να έχει κάποιου είδους πλακέτα στην κατοχή του. Απαιτεί από τον χρήστη δημιουργία

λογαριασμού .



Εικόνα 3.2.2 Σχεδίαση tinkercad.

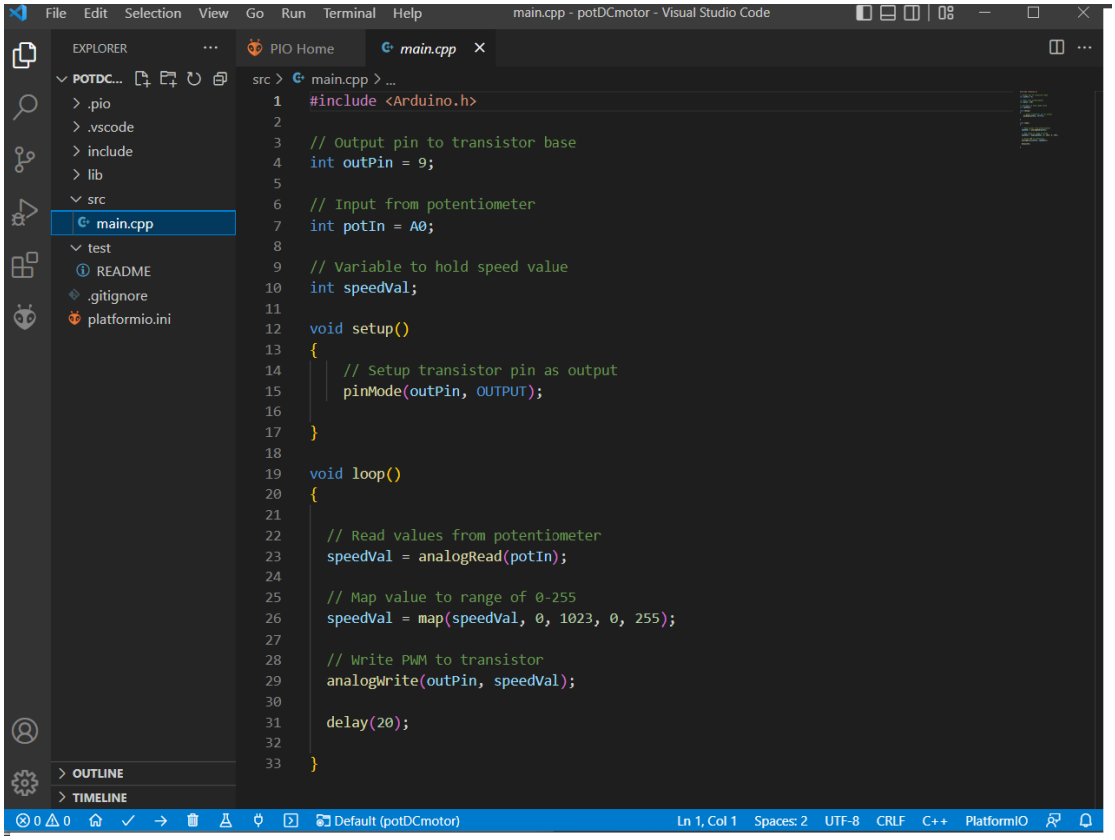
Στην δεξιά στήλη περιέχει διάφορα components που θα χρειαστούν για την δημιουργία ενός κυκλώματος , αφού γίνει η πρόσθεση τους στο breadboard και στα σωστά pins του arduino μπορεί να δημιουργηθεί κατάλληλος κώδικας .Το περιβάλλον είναι πλήρως διαδραστικό με την δυνατότητα προσομοίωσης του κώδικα που γράφτηκε .

2.2.3 Visual Code Studio.

Το visual code studio είναι ένα σύγχρονο IDE ανοιχτού κώδικα που δημιουργήθηκε το 2015 από την Microsoft ,τα τελευταία χρόνια είναι το πιο διαδεδομένο περιβάλλον ανάπτυξης λογισμικού δίνοντας την δυνατότητα στον χρήστη να σώσει τον κώδικα απευθείας στο GitHub. Είναι ικανό για συγγραφή κώδικα σέ όλες τις γλώσσες προγραμματισμού java,c,c++,JavaScript,python,Node.js αφού περιέχει βασική υποστήριξη για όλες αυτές τις γλώσσες , στο περιβάλλον μπορούν να προστεθούν πολλές επεκτάσεις προσαρμόζοντας έτσι ο χρήστης στις ανάγκες του. Το περιβάλλον είναι φιλικό προς τον χρήστη διαθέτοντας στον προγραμματιστή όλα τα απαραίτητα εργαλεία (debbuger,compiler) για την δημιουργία ενός προγράμματος ,επιπλέον διαθέτει αυτόματη συμπλήρωση κώδικα (πχ συμπληρώνει αυτόματα μια δηλωμένη μεταβλητή) και επισήμανση σύνταξης (πχ με την επιλογή μιας μεταβλητής μας εμφανίζει πόσες φορές έχει γραφτεί). Επίσης διαθέτει λειτουργία ταυτόχρονης προβολής διαφορετικών sketch με την ικανότητα σύγκρισης και επισήμανσης διαφορετικών μπλοκ κώδικα .

Η δημιουργία sketch στην παρούσα πτυχιακή έγινε αποκλειστικά με την χρήση του visual code studio επιλέχθηκε αντί για το Arduino IDE καθώς είναι πιο σύγχρονο-εξελιγμένο περιβάλλον. Για την σύνδεση του Arduino έγινε χρήση της επέκτασης platformio.io . Η συγκεκριμένη επέκταση είναι μια επαγγελματική πλατφόρμα διαχείρισης οποιοδήποτε ενσωματωμένου συστήματος διαθέτοντας εξειδικευμένο debugging με στατιστικά δεδομένα που είναι χρήσιμα για να γνωρίζουμε σε ποια κατάσταση βρίσκεται η πλακέτα μας κατά την δημιουργία

sketch, παρέχει υποστήριξη σε περισσότερες από 800 πλακέτες οπότε είναι το κατάλληλο εργαλείο για τον προγραμματισμό οποιαδήποτε πλακέτας .Διαθέτει ενσωματωμένες αρκετές βασικές βιβλιοθήκες καθώς και αναζήτηση για πρόσθεση νεών βιβλιοθηκών .



Εικόνα 3.2.3 Περιβάλλον Visual Code Studio.

Στην αριστερή στήλη explorer μπορεί να γίνει πρόσθεση οποιουδήποτε sketch με δυνατότητα ταυτόχρονης προβολής του κώδικα καθώς και ιστορικό για να μπορούμε να βρούμε οποιαδήποτε αλλαγή έγινε στον κώδικα . Η μορφή του αρχείου που δημιουργείται είναι διαφορετική(.cpp) από αυτή που χρησιμοποιείται στο Arduino (.ino) δημιουργώντας κατευθείαν αρχείο c++ .Υπάρχουν δύο τρόποι πρόσδεσης μιας βιβλιοθήκης στο συγκεκριμένο περιβάλλον ο ένας τρόπος είναι η μεταφορά των βιβλιοθηκών στον ίδιο φάκελο (scr) που βρίσκεται το main αρχείο και η αναφορά τους στην αρχή του κώδικα , ο άλλος τρόπος είναι η μετάβαση στην αναζήτηση βιβλιοθηκών που διαθέτει το Platformio.io και η συμπερίληψη τους στον κώδικα .Στην κάτω πλευρά βρίσκονται τα εργαλεία που χρησιμοποιούμε για την επικοινωνία με την πλακέτα (έλεγχος κώδικα ,φόρτωση στην πλακέτα, άνοιγμα serial communication) και ένα παράθυρο με την εμφάνιση όλων των διαθέσιμων sketch για την επιλογή του προγράμματος που θα γίνει έλεγχος .

2.3 Βιβλιοθήκες.

Η χρήση βιβλιοθηκών για το Arduino είναι αναγκαία για την διαχείριση διάφορων components που κυκλοφορούν ,έχουν βοηθητικό ρόλο για την ευκολότερη διαχείριση αυτών. Οι βιβλιοθήκες είναι έτοιμες γραμμές κώδικα που δημιουργήθηκαν από άλλους χρήστες για τον έλεγχο ενός εξαρτήματος , η χρήση από τον χρήστη γίνεται με την επισύναψη της βιβλιοθήκης στο δικό προσωπικό τους sketch δηλώνοντας τη στην αρχή του προγράμματος με την εντολή #INCLUDE <library.h>. Για έναν μέσο χρήστη θα ήταν αδύνατο να χρησιμοποιήσει κάποιου είδους κινητήρα χωρίς την χρήση βιβλιοθηκών καθώς θα απαιτούσε την πλήρη εξοικείωση και μελέτη του με το αντικείμενο για την εκτέλεση απλών λειτουργιών . Ο χρήστης αφού φορτώσει τις βιβλιοθήκες που χρειάζεται στο IDE χρησιμοποιεί συγκεκριμένες functions ή μπλοκ κώδικα για να τις χρησιμοποιήσει.

Στην παρούσα εργασία χρησιμοποιήθηκαν οι παρακάτω βιβλιοθήκες:

- <Arduino.h> : Φάκελος εγκατάστασης αρχείων του Arduino .
- <dht11.h>: Βιβλιοθήκη για την χρήση του αισθητήρα θερμοκρασίας και υγρασίας dht11.
- <wire.h>: Απαραίτητο όταν γίνεται χρήση component με σύνδεση πρωτοκόλλου I2C.
- <liquidCrystal.h>: Χρήση οθόνης με πλήρη συνδεσμολογία .
- <liquidCrystal_I2C.h>: Συγκεκριμένη βιβλιοθήκη για την χρήση της οθόνης με συνδυασμό με το I2C module.
- <servo.h>: Εκκίνηση servo motor.
- <Stepper.h>: Εκκίνηση stepper motor .
- <adafruit_PWMServoDriver.h>: Βιβλιοθήκη για την χρήση του I2c PWM/DRIVER module .

3 Υλοποίηση

Στην συγκεκριμένη ενότητα γίνεται παρουσίαση του πειραματικού μέρους της πτυχιακής, τα sketch δημιουργήθηκαν μεμονωμένα για την σωστή λειτουργία της εφαρμογής δημιουργώντας συναρτήσεις με σκοπό την άμεση διαχείριση του κώδικα . Η συνδεσμολογία όλων των components παρουσιάζεται μέσω simulation (tinkercad) και ο κώδικας μέσω visual code studio για την πλήρη κατανόηση της εφαρμογής δημιουργήθηκαν διαγράμματα ροής σε συνδυασμό με σχόλια .

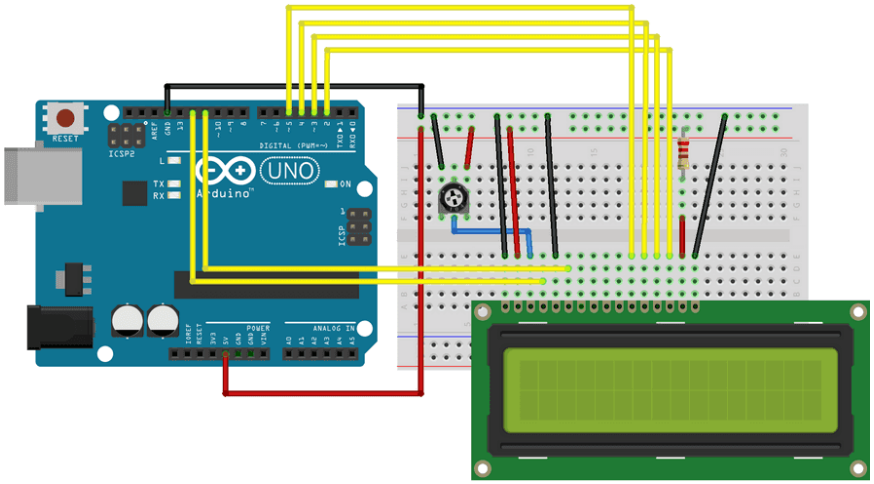
3.1 Εκτύπωση θερμοκρασίας και υγρασίας στην οθόνη.

Η παρούσα εφαρμογή μας ενημερώνει για την θερμοκρασία και την υγρασία μέσα στον χώρο που είναι συνδεμένο το σύστημα. Μέσω του αισθητήρα DHT11 λαμβάνονται οι απαραίτητες πληροφορίες που εμφανίζονται σε μια οθόνη.

3.1.1 Συνδεσμολογία οθόνης και περιγραφή επαφών.

Έγινε χρήση της οθόνης υγρών κρυστάλλων LED LCD 16x02 ,Για την σύνδεση της οθόνης υπάρχουν δύο τρόποι σύνδεσης με τα θετικά τους και τα αρνητικά τους γίνεται. Η σύνδεση της οθόνης με απευθείας σύνδεση των Pins απαιτεί την χρήση πολλών θυρών από το Arduino, αυτό μας δημιουργεί το πρόβλημα μείωση των διαθέσιμων θυρών άρα λιγότερα components γιαυτό έγινε επιλογή της σύνδεσης της οθόνης με το πρωτόκολλο I2C.Ανεξάρτητα από τον τρόπο που επιλεγεί ο τρόπος που θα συνδεθεί η οθόνη θα πρέπει να γίνει έλεγχος του datasheet της οθόνης για την επιλογή των σωστών Pins καθώς οι οθόνες που είναι διαθέσιμες προς αγορά είναι διαφορετικές μεταξύ τους, επίσης προτιμήθηκε να γίνει χρήση εξωτερικής παροχής ρεύματος με μπαταρία .Για την χρήση της οθόνης έγινε χρήση της βιβλιοθήκης «LiquidCrystal».

- GND: GND board
- VCC: Τάση 5V
- V0: Σύνδεση potensiometer για έλεγχο του backlight
- RS: Pin12
- RW: γείωση board
- E: Pin 11
- D0: Κενό
- D1: Κενό
- D2: Κενό
- D3: Κενό
- D4: Pin5 του Arduino.
- D5: Pin 4 του Arduino.
- D6: Pin 3 του Arduino.
- D7: Pin 2 του Arduino.
- A: Τάση 5V μαζί με αντίσταση 220Ωμ
- K: GND board



Εικόνα 4.1.1 Συνδεσμολογία οθόνης φωτογραφία από FRITZING.

- Εφόσον δεν θα γίνει σύνδεση του ποτενσιόμετρου, για να μπορεί να λειτουργήσει η οθόνη θα πρέπει να συνδέσουμε μια αντίσταση έτσι ώστε να δημιουργηθεί ένα μικρό κύκλωμα που θα αντικαταστήσει το ποτενσιόμετρο.

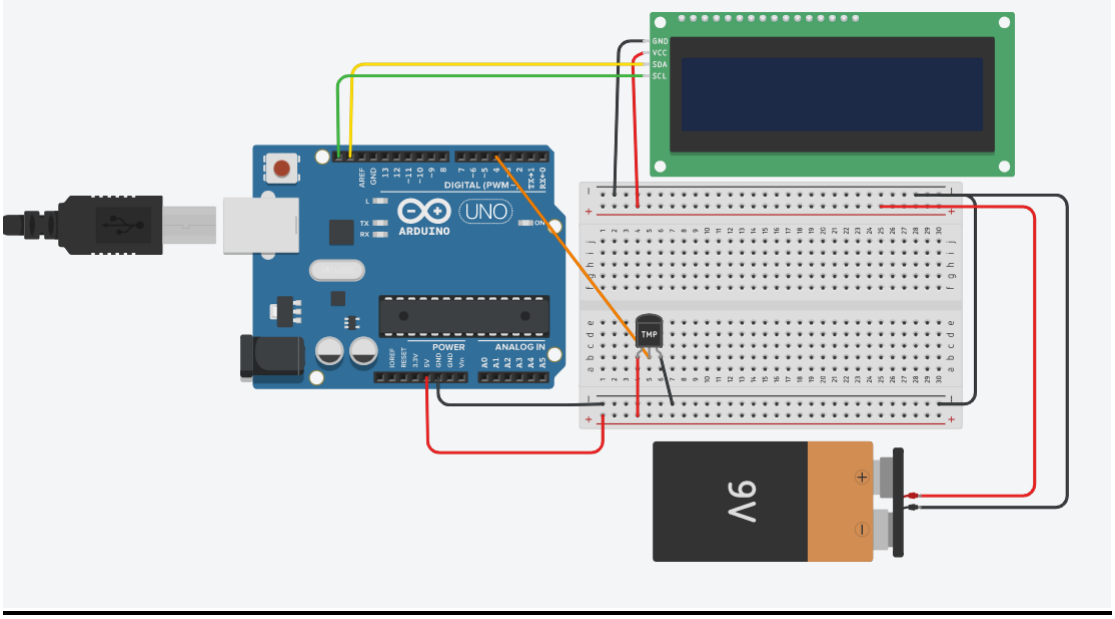
- Η επαφή RS (register select) χρησιμοποιείται για την αποστολή δεδομένων στην οθόνη . Η συγκεκριμένη θύρα της οθόνης μπορεί να συνδεθεί σε οποιαδήποτε digital pin άλλα και στην τάση-γείωση. Στην περίπτωση που συνδεθεί στην γείωση 0V μπορούν να σταλθούν σχόλια σε συγκεκριμένες θέσεις της οθόνης. Αν συνδεθεί στην τάση 5V μπορούν να στέλνονται δεδομένα και χαρακτήρες.

- Η επαφή RW (read-write) χρησιμοποιείται για την επιλογή της οθόνης γράψιμο και την αποστολή δεδομένων στην οθόνη.

- Η επαφή E (enable) χρησιμοποιείται για συγγραφή χαρακτήρων στην οθόνη.

- Οι επαφές D4,D5,D6,D7 της οθόνες χρησιμοποιούνται με σκοπό την δημιουργία 4bit συνδεσμολογίας , μπορούμε να συνδέσουμε και τις 8 επαφές για την δημιουργία 8bit.
- Η οθόνη υγρών κρυστάλλων χρησιμοποιεί οπίσθιο φωτισμό(backround) με την τροφοδοσία του να γίνεται μέσω των δύο τελευταίων επαφών (A/K) η μια είναι η άνοδος(σύνδεση στα 5V) και η άλλη είναι κάθοδος(σύνδεση στα 0V) .

3.1.2 Συνδεσμολογία I2c.



Εικόνα 4.1.2 Σχεδίασή μέσω tinkercad.

Η τροφοδοσία με 5V στην οθόνη για την αποφυγή βραχυκυκλώματος έγινε με εξωτερική μπαταρία .Από την οθόνη η θύρα GND συνδέεται στην γείωση της μπαταρίας ,το VCC που συνδέεται στην τροφοδοσία της μπαταρίας ,το SDA και το SCL συνδέεται στις αντίστοιχα pins του Arduino. Το μεσαίο pin του αισθητήρα θερμοκρασίας συνδέεται στο 4^{ο} pin της αναλογικής θύρας του arduino , το αριστερό της τάσης συνδέεται στα 5V και το δεξί στην γείωση της πλακέτας .Ο αισθητήρας που χρησιμοποιήθηκε στην παρούσα εργασία (DHT11)δεν βρισκόταν στο tinkercad οπότε έγινε προσαρμογή της καλωδίωσης για τις ανάγκες τις σχεδίασης.

3.1.3 Πρόγραμμα και κώδικας.

Αρχικά γίνεται η συμπερίληψη των βιβλιοθηκών .

```
#include <Wire.h>
#include <Arduino.h>
#include <LiquidCrystal_I2C.h>
#include <dht11.h>
```

Ορισμός αισθητήρα θερμοκρασία στο ψηφιακό Pin 4 της πλακέτας

```
#define dht11Pin 4;
```

Δημιουργία αντικείμενου κλάσης dht11 για τον αισθητήρα θερμοκρασίας .

Δημιουργία αντικειμένου κλάσης LiquidCrystal_I2C για την οθόνη που περιέχει 3 τιμές την διεύθυνση του πρωτοκόλλου I2C, την τιμή για τις στήλες και την τιμή για τις σειρές. Η διεύθυνση I2C βρέθηκε με χρήση του I2C SCANNER τα υπόλοιπα πεδία συμπληρώνονται ανάλογα με το υλικό που χρησιμοποιείται.

```
dht11 DHT;
LiquidCrystal_I2C lcd(0x27,16,2);
```

Δήλωση μεταβλητών για την θερμοκρασία (temp) και την υγρασία (humi) και της μεταβλητής που επιστρέφει της τιμές (chk)

```
int temp = 0;
int humi = 0;
int chk = 0;
```

Στην συνάρτηση setup γίνεται χρήση της συνάρτησης init() για αρχικοποίηση της οθόνης ,στην συνέχεια η εντολή clear() η οποία σβήνει τα δεδομένα της οθόνης τέλος η εντολή backlight () η οποία χρησιμοποιείται για την ενεργοποίηση του οπίσθιου φωτισμού της οθόνης.

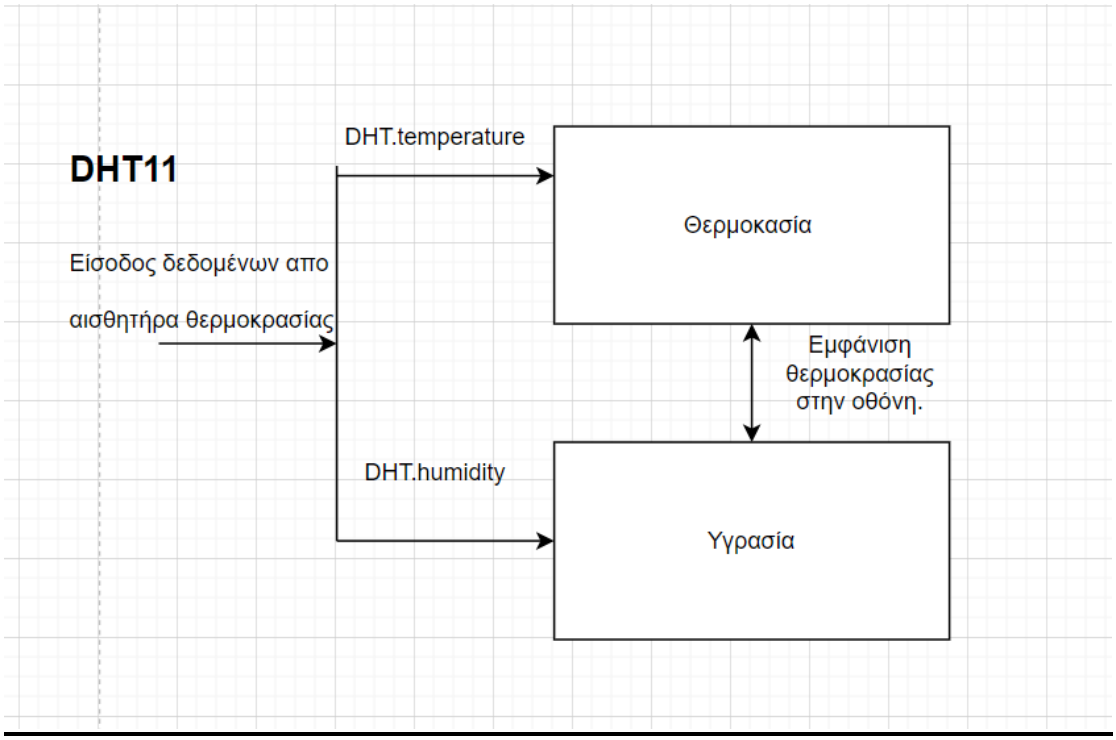
```
void setup() {
  lcd.init();
  lcd.clear();
  lcd.backlight();
}
```

Αρχικά δηλώνουμε ότι ο DHT11 θα διαβάζει τις τιμές του αισθητήρα με τη μεταβλητή chk και οριστικοποιούμε τις μεταβλητές temp-humi για να εμφανίζουν τα σωστά δεδομένα. Στην συνέχεια με την συνάρτηση lcd.setCursor(2,0) ορίζεται η θέση που θα εμφανιστεί το κείμενο και με την εντολή lcd.print(temp) εκτυπώνεται στην οθόνη η θερμοκρασία . Αντίστοιχη λειτουργία έχει και το lcd.setCursor(2,1) για την υγρασία.

```
void loop() {

   chk = DHT.read(dht11Pin);


  temp = DHT.temperature;
  humi = DHT.humidity;

  lcd.setCursor(2,0);

  lcd.print("Temp= ");
  lcd.print(temp);
  lcd.print(" C");

  lcd.setCursor(2,1);
```

```
lcd.print("Humi= ");
lcd.print(humi);
lcd.print(" %");
delay(0);
```

3.1.4 Σχεδιάγραμμα.

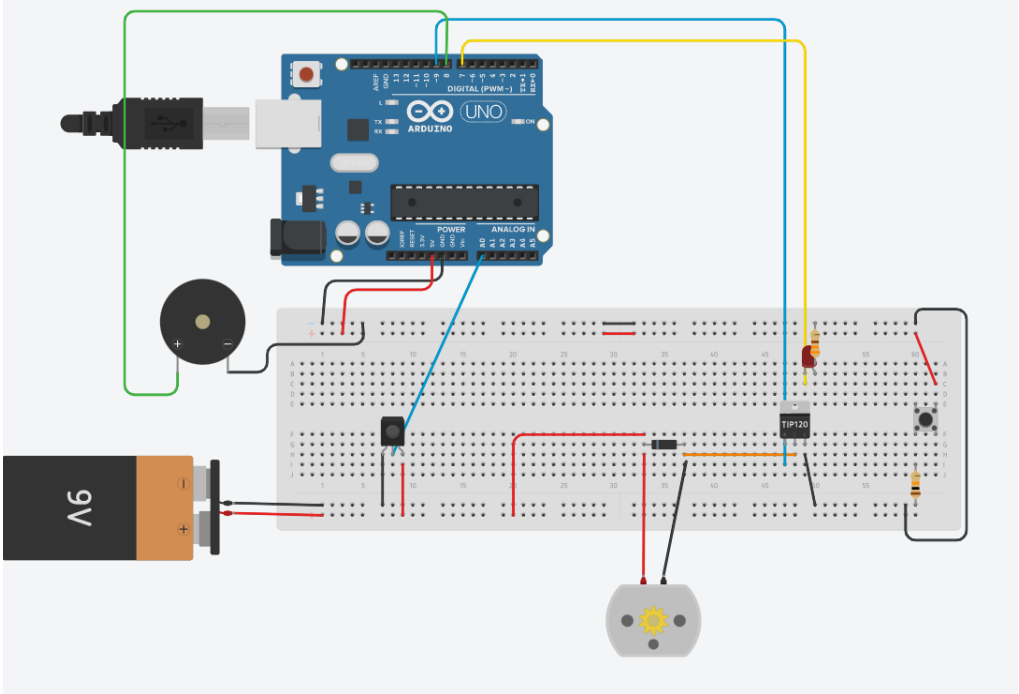


Εικόνα 4.1.4 Σχεδιάγραμμα με draw.io.

3.2 Αισθητήρας φωτιάς.

Η συγκεκριμένη εφαρμογή χρησιμοποιεί σαν είσοδο έναν αισθητήρα φωτιάς . Εφόσον εντοπιστεί φλόγα στο πεδίο δράσης του αισθητήρα τότε ανάλογα με την απόσταση θα λειτουργήσει η ανάλογη έξοδος, όταν εντοπιστεί φλόγα παράγεται ήχος από το buzzer με σκοπό της ενημέρωση για την ύπαρξη φωτιάς σε κοντινή απόσταση με άμεση επέμβαση του dc motor που χρησιμοποιήθηκε σαν μια μορφή συστήματος πυρόσβεσης ,που στην παρούσα μορφή με τον αέρα που δημιουργεί σβήνει την φωτιά. .Σαν έξοδος χρησιμοποιήθηκε και ένα led που ενεργοποιείται όταν πλησιάζει η φωτιά τον αισθητήρα για να μπορούμε να ελέγξουμε την λειτουργία της εφαρμογής. Για την προστασία του κυκλώματος από βραχυκύκλωμα από το dc motor έγινε χρήση μιας διόδου συνδεμένη με τέτοιο τρόπο ώστε να δημιουργείται αντίστροφη πολικότητα και ενός τρανζίστορ.

3.2.1 Συνδεσμολογία.



Εικόνα 4.2.1 Συνδεσμολογία.

Για την τροφοδοσία του αισθητήρα φωτιάς και dc motor χρησιμοποιείται εξωτερική 9V μπαταρία .Για την ασφαλή σύνδεση του DCmotor έγινε χρήση μια διόδου ζένερ 1A και ενός τρανζίστορ TIP120 (NPN 5A) .Στην πλευρά της διόδου που είναι η κάθοδος συνδέθηκε η τάση του DC motor για την τροφοδοσία του με σύνδεση του θετικού πόλου της μπαταρίας ,η γείωση συνδέθηκε στην άλλη πλευρά της διόδου μαζί με το μεσαίο πόδι του τρανζίστορ και την γείωση της μπαταρίας , η αριστερή πλευρά του τρανζίστορ συνδέεται στο 9ο PWM pin του Arduino .Το buzzer συνδέθηκε στο 8ο pin και στην γείωση . Το Led στο 7ο Pin μαζί με αντίσταση 330Ω .Το button στο 2ο Pin μαζί με αντίσταση 10K .Ο αισθητήρας φλόγας συνδέθηκε στο αναλογικό Pin A0 .

3.2.2 Πρόγραμμα και κώδικας.

Εισαγωγή βιβλιοθήκης

```
#include <Arduino.h>
```

Δήλωση ελαχίστων(0) και μεγίστων τιμών(1023) που θα πάρει ο αισθητήρας φωτιάς .Οι τιμές είναι const επειδή είναι σταθερές και μόνο για διάβασμα από το πρόγραμμα.

```
const int sensorMin = 0;
const int sensorMax = 1023;
```

Δήλωση pins .Έλεγχος του delay για καλύτερη διαχείριση το προγράμματος , 0 για άμεση ανταπόκριση , μεγαλύτερο του 2000 για έλεγχο λειτουργίας.

```
int buzzerPin = 8;
int ledPin = 7;
int sensorReading = 0;
int flamePin = A0
int delayTime = 0 ;
int motorControl = 9;
```

Στην Setup() γίνεται η δήλωση σαν έξοδο του buzzer,του led και του DCmotor. Σαν είσοδο τον αισθητήρα φωτιάς και δήλωση της σειριακής επικοινωνίας (9600 baud rate) που χρησιμοποιήθηκε για έλεγχο ορθότητας της εφαρμογής.

```
void setup() {
pinMode(buzzerPin,OUTPUT);
pinMode(ledPin,OUTPUT);
pinMode(flamePin,INPUT);
pinMode(motorControl, OUTPUT);

Serial.begin(9600);
}
```

Ορισμός της τιμής του αισθητήρα της φωτιάς στο sensorReading . Δημιουργία χάρτη με τις τιμές του αισθητήρα. Το sensorReading δίνει την τιμή του αισθητήρα που είναι σε πραγματικό χρόνο και επιστρέφει τα ελάχιστα-μέγιστα. Το 0,3 διαιρεί δια του 3 το 1023 που είναι η μέγιστη τιμή που μπορεί να πάρει ο αισθητήρας και δημιουργεί 3 χάρτες η θέση 0 ανήκει από το 0 μέχρι το 341 , η θέση 1 από το 342 μέχρι το 682 και η θέση 2 από το 683 μέχρι το 1023.

```
 sensorReading = analogRead(flamePin);
 int range = map(sensorReading, sensorMin, sensorMax, 0, 3);
```

Όταν η συνθήκη είναι αληθής μπαίνει σε λειτουργία το DC motor και ανάβει το LED. Το DC motor έχει επιλεχθεί για την επίδειξη ενός συστήματος πυρόσβεσης και αυτό γίνεται λίγο μετά από την προειδοποίηση για φωτιά του buzzer με ηχητικό .

```
if(sensorReading<550){


   for(int x = 0; x <= 255; x++)
   {
   analogWrite(motorControl, x);
   delay(0);
   digitalWrite(ledPin,HIGH);

   }
```

Όταν η φωτιά είναι σε πολύ κοντινή απόσταση από τον αισθητήρα τότε μπαίνει σε λειτουργία το map 0 .

```
 switch (range) {
 case 0:

 Serial.println(sensorReading);
  if(sensorReading<341)
  {    Serial.println("Close Fire");

  }
     break;
```

Όταν η φωτιά είναι σε μέτρια απόσταση από τον αισθητήρα τότε το map είναι 1 και ενεργοποιείται το buzzer με σκοπό την προειδοποίηση για φωτιά. Εδώ το Delay είναι 500 καθώς αν μπει 0 δεν θα βγάζει ήχο.

```
case 1:

   Serial.println(sensorReading);


 if(sensorReading<=682 || sensorReading >=341)
 {
 digitalWrite(buzzerPin,HIGH);
 delay(500);
 digitalWrite(buzzerPin,LOW);
 Serial.println(" Distant Fire");
 delay(delayTime);
 }

delay(1);
   break;
```

Η τελευταία περίπτωση είναι να μην υπάρχει φωτιά οπότε έχουμε το map 2 και δεν εκτελείται καμιά ενέργεια.

```
case 2:    .
   Serial.println("No Fire");
   Serial.println(sensorReading);
   break;
 }
```

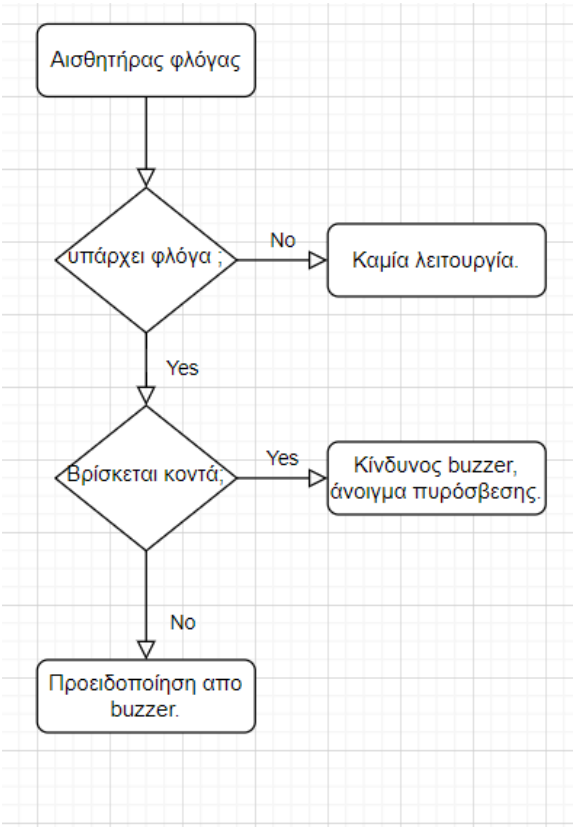
Όταν η συνθήκη είναι ψευδής σταματάει η λειτουργία του DC motor και σβήνει το LED.

```
if(sensorReading >400){


   for(int x = 255; x >= 0; x--)
   {
     analogWrite(motorControl, x);
     delay(0);
   }
     digitalWrite(ledPin,LOW);
  }
  delay(delayTime);
}
```

Στα μπλοκ κώδικα που γράφτηκαν για την λειτουργία του DC motor τα delay είναι 0 για να έχουμε άμεση ανταπόκριση .

Εφόσον υπάρχει φωτιά ανάλογα με την απόσταση υπάρχουν διαφορετικές καταστάσεις. Οι καταστάσεις αυτές εναλλάσσονται ανάλογα με την τιμή που έχει ο αισθητήρας , οι αλλαγές αυτές βρίσκονται στο range αλλάζουν με την εντολή switch και τερματίζουν (εφόσον αλλάξει το range).

3.2.3 Σχεδιάγραμμα

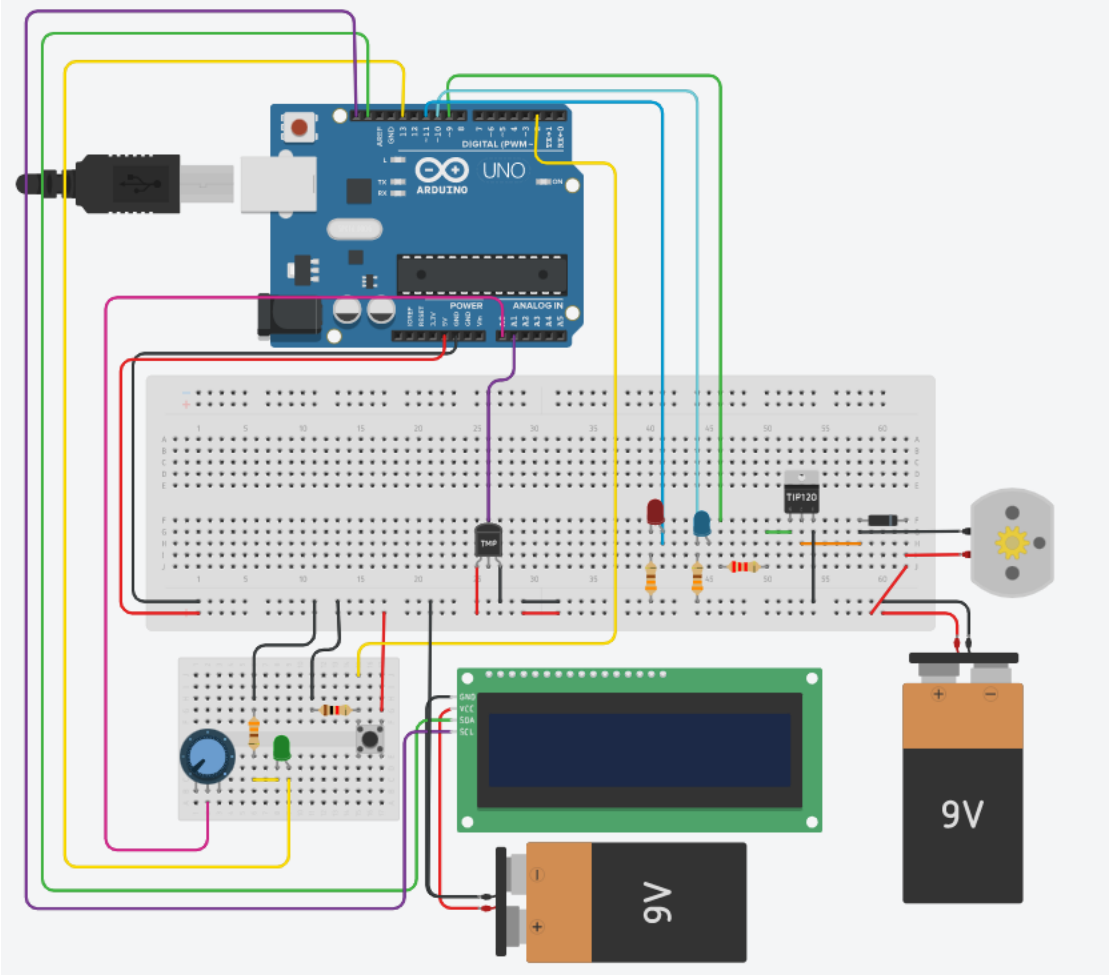


Εικόνα 4.2.3 Σχεδιάγραμμα με draw.io

3.3 Θερμοστάτης.

Στην παρούσα εφαρμογή χρησιμοποιήθηκε ως κύρια είσοδος ένας αισθητήρας θερμοκρασίας-υγρασίας DHT11 ,ένα ποτενσιόμετρο για ρύθμιση τιμών και ένα κουμπί για την επιλογή ανάμεσα σε λειτουργία χειροκίνητης ρύθμισης με το ποτενσιόμετρο και αυτόματης ρύθμισης ανάλογα με την θερμοκρασία που επικρατεί. Σαν έξοδοι χρησιμοποιήθηκαν 3 LED διαφορετικού χρώματος και ένα DC motor .Το πράσινο led όταν είναι αναμμένο μας ενημερώνει για την λειτουργία της αυτόματης κατάστασης και όταν είναι σβηστό της χειροκίνητης. Το κόκκινο led βρίσκεται σε λειτουργία PWM όταν οι τιμές τις θερμοκρασίας ή του ποτενσιόμετρου είναι κάτω από 20 βαθμούς κελσίου και όσο πλησιάζει προς το 0 τόσο αυξάνεται η φωτεινότητα του. Το μπλε led λειτουργεί μετά τους 25 βαθμούς κελσίου σε συνδυασμό με το DC motor πού χρησιμοποιείται σαν ανεμιστήρας ,όσο μεγαλώνει η θερμοκρασία τόσο αυξάνεται ο παλμός με αποτέλεσμα της αύξησης της φωτεινότητας και της ταχύτητας του κινητήρα. Από 20 μέχρι 25 βαθμούς υπάρχει αδράνεια και δεν υπάρχει καμία λειτουργία από το κύκλωμα.

3.3.1 Συνδεσμολογία.



Εικόνα 4.3.1 Συνδεσμολογία με tinkercad.

Χρησιμοποιήθηκαν 2 μπαταρίες των 9Volt η μία συνδέθηκε στην οθόνη και άλλη στο DC motor.Το ποτενσιόμετρο συνδέθηκε στην αναλογική θύρα A0 και ο αισθητήρας DHT11 στην αναλογική θύρα A1, ο συγκεκριμένος αισθητήρας έχει την ικανότητα να συνδέεται και στους δυο τύπους ψηφιακών-αναλογικών pin που διαθέτει η πλακέτα. Για τα led χρησιμοποιήθηκαν αντιστάσεις 330KΩμ και για το button η σύνδεση της αντίστασης (10κΩ) έγινε με την μορφή pulldown . Έγινε χρήση μιας οθόνης 16x2 με το πρωτόκολλο I2C με τις θύρες εξόδου SDA-SCL να συνδέονται στις αντίστοιχες του Arduino .Η σύνδεση του DC motor αρχικά έγινε με την χρήση ενός TIP120 NPN BJT ,παρατηρήθηκε άμεση αύξηση της θερμοκρασίας του και μη επιθυμητή απόδοσης ,έγινε αντικατάσταση του με ένα MOSFET IRF540 τρανζίστορ με τα επιθυμητά αποτελέσματα χωρίς υπερθερμάνσεις με τις μέγιστες επιδόσεις του κινητήρα. Το MOSFET περιέχει 3 pins το δεξί συνδέεται με την γείωση του κυκλώματος-μπαταρίας, το μεσαίο με μια δίοδο ζένερ που με αυτή δημιουργήθηκε αντίστροφη πολικότητα για την προστασία του κυκλώματος και το αριστερό σε μια αντίσταση 2.2K που αυτό τελικά συνδέθηκε στο 9^{ο} pin του Arduino για τον έλεγχο της ταχύτητας του κινητήρα με PWM παλμό. Στην δίοδο ζένερ ο θετικός πόλος του κινητήρα συνδέθηκε στη κάθοδο και ο αρνητικός στην άλλη

πλευρά. Με την σύνδεση της οθόνης και την πλήρη λειτουργία του κινητήρα παρατηρήθηκαν παρεμβολές με την μορφή κυματισμού στην οθόνη.

3.3.2 Κώδικας και function.

Έγινε χρήση των παρακάτω βιβλιοθηκών.

```
#include <Arduino.h>
#include <dht11.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

Δήλωση διεύθυνσης και θέσεων οθόνης.

```
LiquidCrystal_I2C lcd(0x27,16,2);
```

Δήλωση εισόδων και εξόδων προγράμματος ο κινητήρας συνδέεται στο pin9 , το μπλε led στο $10^{ο}$,το κόκκινο στο $11^{ο}$ και το πράσινο στην $13^{η}$.

```
int motorPin = 9;
int ledPinBlue = 10;
int ledPinRed = 11;
int ledPin = 13;
int ledState = HIGH;        // the current state of the output pin
```

Οι παρακάτω δηλώσεις αφορούν το μέρος του προγράμματος που ελέγχεται με το ποτενσιόμετρο ,γίνεται δήλωση μεταβλητών που χρησιμοποιήθηκαν για την δημιουργία χαρτών που προσαρμόστηκαν έτσι ώστε να παραχθεί παλμός PWM με ακρίβεια .

```
int potIn = A0
int potDelayTime = 0;
int potSpeedVal;
int potMotorSpeed;
int potMotorpotSpeedVal = 0 ;
int potRedLedVal =0 ;
int potBlueLedVal = 0;
int potTempVal=0;
```

Οι παρακάτω δηλώσεις αφορούν το μέρος του προγράμματος που ελέγχεται με τον αισθητήρα DHT11 ,γίνεται δήλωση μεταβλητών που χρησιμοποιήθηκαν για την δημιουργία χαρτών που προσαρμόστηκαν έτσι ώστε να δημιουργηθεί παλμός PWM ανάλογα με τις τιμές της θερμοκρασίας στις εξόδους .

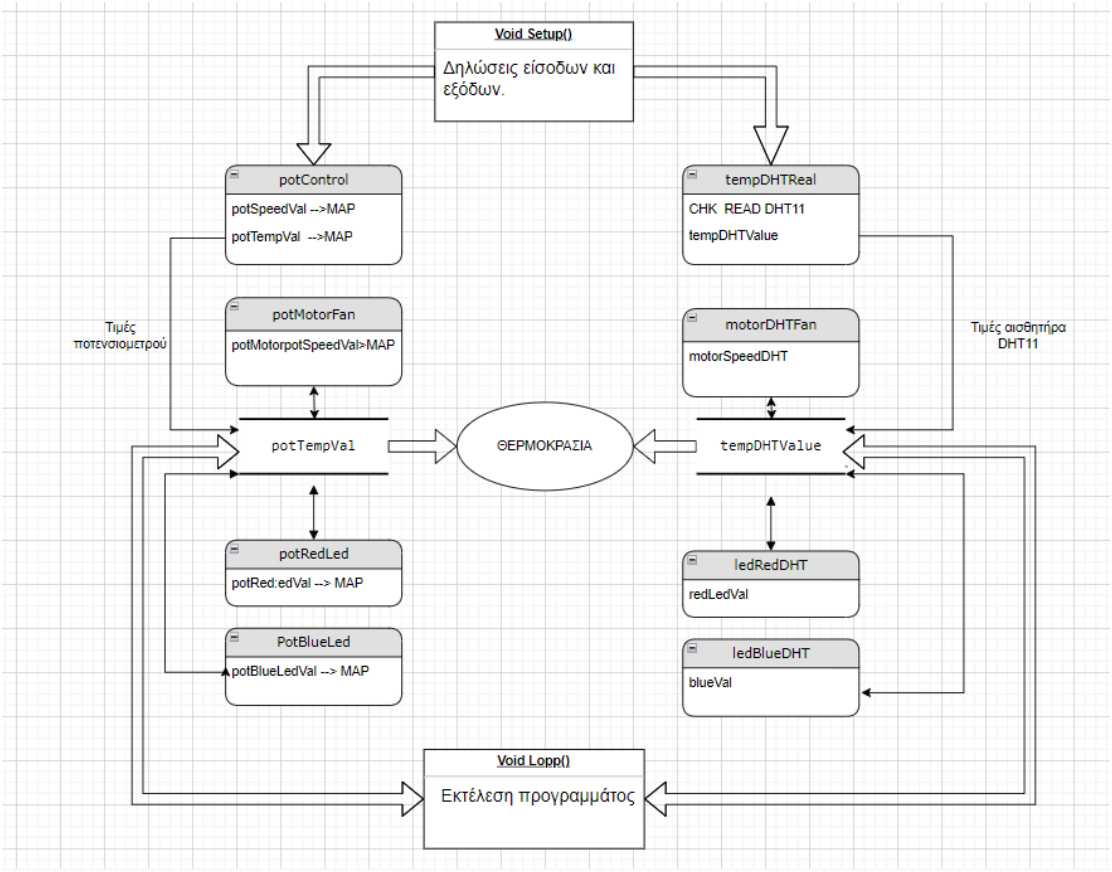
```
//variables DHT 11.
int delayTimeDHT = 0;

int tempDHTValue;
int chk = 0;
//variables for create the DHT MAPS
int motorSpeedDHT = 0 ;
int redLedVal=0;
int blueVal=0;
#define DHT11_PIN A1
dht11 DHT;
```

Στο παρακάτω μπλοκ δηλώνονται οι μεταβλητές οι οποίες χρησιμοποιήθηκαν για τις καταστάσεις του κουμπιού . Το buttonState αποθηκεύει την τιμή όταν ασκείται πίεση στο κουμπί, το lastButtonState είναι η προηγούμενη τιμή που είχε , το currentState αποθηκεύει μια κατάσταση (1ή2) για την εκκίνηση διαφορετικού μπλοκ κώδικα στην Loop με την πίεση του κουμπιού. Το modeSwitch αλλάζει την τιμή του currentState.

```

int buttonPin = 2;
int buttonState;
int lastButtonState = LOW;
int currentState;
const byte modeSwitch = 2;
```

Functions:

Για την ανάπτυξη του κώδικα δημιουργήθηκαν κάποιες function με τις λειτουργίες του προγράμματος .Αυτές βρίσκονται σε πλήρη επικοινωνία με την void setup που γίνονται οι δηλώσεις των components και την void loop που καλεί αυτές τις συναρτήσεις για την πραγματοποίηση των λειτουργιών. Οι function χωρίστηκαν σε δύο κατηγορίες ,στην μια κατηγορία χρησιμοποιείται σαν είσοδος ένα ποτενσιόμετρο που δίνει χειροκίνητα την θερμοκρασία και στην άλλη κατηγορία οι τιμές δίνονται αυτόματα από την αισθητήρα θερμοκρασίας και υγρασίας.



Εικόνα 4.3.2α Σχεδιάγραμμα με draw.io

3.3.2.1 potControl():

```
void  potControl(int potIN, int potOut)
{
  potSpeedVal = analogRead(potIn);
  potSpeedVal = map(potSpeedVal, 0, 1023, 0 ,255 );
  potTempVal = potSpeedVal;
  potTempVal = map(potTempVal,0,255,0,40);
}
```

Έλεγχος τιμών που δίνει το ποτενσιόμετρο και δημιουργία χάρτη για την προσαρμογή του αναλογικού σήματος της εισόδου που έχει τιμές από 0 μέχρι 1023 ,αυτές οι τιμές διαμορφώθηκαν σε έξοδο PWM (0-255) και στην συνέχεια προσαρμόστηκαν στο εύρος μιας φυσιολογικής θερμοκρασίας από το 0 μέχρι το 40. Η έξοδος της συγκεκριμένης συναρτήσεις είναι το potTempVal που χρησιμοποιείται για να δώσει τιμές σε όλες τις εξόδους όταν το πρόγραμμα βρίσκεται σε χειροκίνητη λειτουργία.

3.3.2.2 potMotorFan():

```
void potMotorFan(int potIn , int motorOut)
{
  potMotorpotSpeedVal = potTempVal;
  potMotorpotSpeedVal = map (potMotorpotSpeedVal,25,40,0,255);

  analogWrite(motorPin, potMotorpotSpeedVal);
  delay(20);

}
```

Η συγκεκριμένη συνάρτηση λαμβάνει σαν είσοδο την τιμή του ποτενσιόμετρου (potTempVal) και την μετατρέπει σε PWM παλμό για εκκίνηση του κινητήρα που λειτουργεί σαν ανεμιστήρας όταν ο χρήστης επιλέξει θερμοκρασία μεγαλύτερη των 25 βαθμών. Όσο αυξάνετε η τιμή του potTempVal τόσο μεγαλώνει και η ταχύτητα του ανεμιστήρα .

3.3.2.3 PotRedLed():

```
void PotRedLed(int potIn , int ledOut)
{
  potRedLedVal = potTempVal;
  potRedLedVal = map (potRedLedVal,20,0,0,255);
  analogWrite(ledPinRed,potRedLedVal);
  delay(20);

}
```

Η συγκεκριμένη συνάρτηση λαμβάνει την τιμή του ποτενσιόμετρου και την μετατρέπει σε ψηφιακό παλμό για το κόκκινο λαμπάκι που χρησιμοποιείται για την λειτουργία της θέρμανσης στις χαμηλές θερμοκρασίες. Όταν οι τιμές είναι μικρότερες του 20 το λαμπάκι ανάβει και όσο πλησιάζει η θερμοκρασία κοντά στο 0 τόσο αυξάνεται η φωτεινότητα του.

3.3.2.4 PotBlueLed():

```
void PotBlueLed(int potIn , int ledOut)
{
  potBlueLedVal = potTempVal;
  potBlueLedVal = map (potBlueLedVal,25,40,0,255);
```

```
 analogWrite(ledPinBlue,potBlueLedVal
 delay(20);

}
```

Η συγκεκριμένη function λαμβάνει την τιμή του ποτενσιόμετρου και την μετατρέπει σε ψηφιακό παλμό για το μπλε λαμπάκι που χρησιμοποιείται για την λειτουργία της ψύξης στις υψηλές θερμοκρασίες σε συνδυασμό με τον ανεμιστήρα . Με την άνοδο της θερμοκρασίας αυξάνεται και η φωτεινότητα του LED.

3.3.2.5 tempDHTReal():

```
void tempDHTReal(int chkIn , int chkOut)
{
 chk = DHT.read(DHT11_PIN);
 tempDHTValue = DHT.temperature;
 analogWrite(chk,tempDHTValue);
}
```

Η συγκεκριμένη εφαρμογή διαβάζει τις τιμές του αισθητήρα DHT11 και τις αποθηκεύει σε μια μεταβλητή tempDHTValue που η τιμή της χρησιμοποιείται για την έξοδο των components .

3.3.2.6 motorDHTFan():

```
void motorDHTFan(int motorIn , int motorOut)
{
 motorSpeedDHT = tempDHTValue;
 motorSpeedDHT = map(tempDHTValue,25,40,0,255);
 analogWrite(motorPin, motorSpeedDHT);
 delay(20);
}
```

Η συγκεκριμένη συνάρτηση προσαρμόζει τις τιμές της μεταβλητής tempDHTValue σε PWM παλμό για τον ανεμιστήρα.

3.3.2.7 ledRedDHT():

```
void ledRedDHT(int redIn , int redOut)
{
 if(tempDHTValue <20)
   {
    redLedVal = tempDHTValue;
    redLedVal = map(tempDHTValue,0,20,255,0);
    analogWrite(ledPinRed,redLedVal);
    delay(20);
   }
```

```
}
```

Η συγκεκριμένη συνάρτηση προσαρμόζει τις τιμές της μεταβλητής tempDHTValue σε PWM παλμό για το κόκκινο led όταν ή θερμοκρασία είναι μικρότερη των 20 βαθμών.

3.3.2.8 ledBlueDHT():

```
void ledBlueDHT(int blueIn, int blueOut)
{
  if(tempDHTValue> 25)
  {
    blueVal = tempDHTValue;
    blueVal = map(tempDHTValue,25,40,0,255);
    analogWrite(ledPinBlue,blueVal);
    delay(20);
  }
}
```

Η συγκεκριμένη συνάρτηση προσαρμόζει τις τιμές της μεταβλητής tempDHTValue σε PWM παλμό για το μπλε led όταν ή θερμοκρασία είναι μεγαλύτερη των 25 βαθμών.

3.3.2.9 void setup():

```
void setup(){
  pinMode(potIn,INPUT);
  //button State
  pinMode(buttonPin, INPUT);
  pinMode(modeSwitch, INPUT_PULLUP);
  // LED states
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, ledState);
  pinMode(motorPin, OUTPUT);
  pinMode(ledPinBlue,OUTPUT);
  pinMode(ledPinRed,OUTPUT);
  pinMode(DHT11_PIN,INPUT);
  //LCD
  lcd.init();
  lcd.clear();
  lcd.backlight();
  Serial.begin(9600);
}
```

Η συνάρτηση αυτή έχει σκοπό την δήλωση της επικοινωνίας των εισόδων και των εξόδων που χρησιμοποιήθηκαν . Έγινε δήλωση του ποτενσιόμετρου ,του button και του αισθητήρα θερμοκρασίας σαν είσοδοι και σαν έξοδοι δηλώθηκε η οθόνη lcd ,το dc motor και τα λαμπάκια .

3.3.2.10 void loop ():

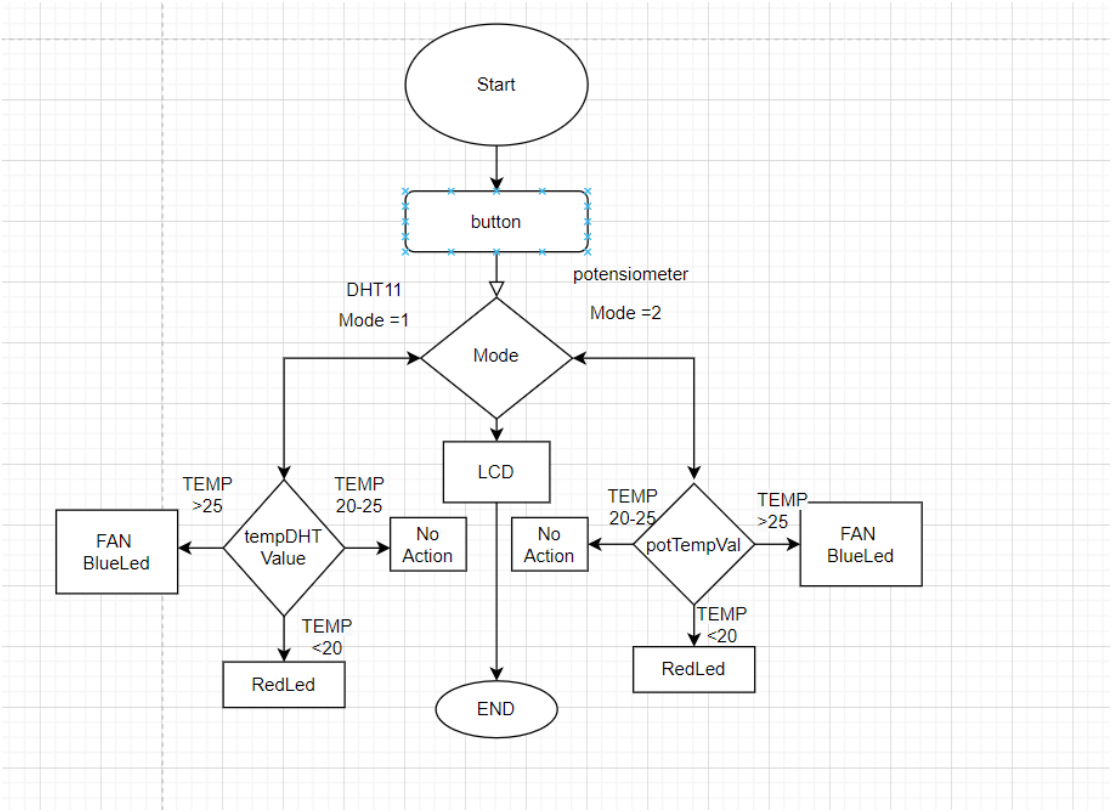
Η συγκεκριμένη συνάρτηση μας δίνει την δυνατότητα ελέγχου της λειτουργίας του προγράμματος μέσα από ένα button, σαν πρώτη λειτουργία έχει ρυθμιστεί να ανάβει το πράσινο led ενημερώνοντας μας πως η εφαρμογή είναι στην αυτόματη λειτουργία και ανάλογα με της θερμοκρασίες που επικρατούν προσαρμόζει και την λειτουργία του θερμοστάτη και όταν πιέζεται το κουμπί το σύστημά μπαίνει στην χειροκίνητη επιλογή θερμοκρασίας. Στη μεταβλητή currState βρίσκεται η τιμή του κουμπιού , στην περίπτωση που ασκηθεί πίεση θα αποθηκευτεί η τιμή στο lastState. Όταν το currState και το lastState έχουν διαφορετική τιμή τότε αλλάζει το selectMode (από 1 σε 2) .Όλες οι συναρτήσεις συνδυάζονται στο συγκεκριμένο μπλοκ κώδικα ώστε να λειτουργήσουν πλήρως.

```
void loop(){
delay(0);
  //  last state of the button
  static byte lastState = HIGH;
  // selected mode
  static byte selectedMode = 1;

  byte currState = digitalRead(modeSwitch);
  if (currState != lastState)
```

```
  {
    if (lastState == LOW)
    {
      digitalWrite(ledPinRed,LOW);
      digitalWrite(ledPinBlue,LOW);
      digitalWrite(motorPin,LOW);
      if (selectedMode == 1)
        selectedMode = 2;
      else
        selectedMode = 1;
      currentState = 1;
    }
    lastState = currState;
  }
  if (selectedMode == 1)
{
tempDHTReal(chk,tempDHTValue);

  if(tempDHTValue >25)
  {
    motorDHTFan(tempDHTValue,motorPin);
    ledBlueDHT(blueVal,ledPinRed);
    digitalWrite(ledPinRed,LOW);
  }
  else if(tempDHTValue <20)
{
    ledRedDHT(redLedVal,ledPinRed);
}else
{

    digitalWrite(motorPin,LOW);
    digitalWrite(ledPinBlue,LOW);
    digitalWrite(ledPinRed,LOW);
}
lcd.setCursor(2,0);
lcd.print("temperature :");
lcd.println(tempDHTValue);
digitalWrite(ledPin,HIGH);
}
else
{

  potControl(potIn,potTempVal);

  if(potTempVal > 25)
  {
    potMotorFan(potTempVal,motorPin);
    PotBlueLed(potTempVal,ledPinBlue);
```

```
   digitalWrite(ledPinRed,LOW);
  }
   else if(potTempVal<20)
   {
     PotRedLed(potTempVal,ledPinRed);
   }
 else
   {
     digitalWrite(motorPin,LOW);
     digitalWrite(ledPinBlue,LOW);
     digitalWrite(ledPinRed,LOW);
   }
  lcd.setCursor(2,1);
  lcd.print("Thermostat");
  lcd.print(potTempVal);
  digitalWrite(ledPin,LOW);
}
}
```

Η συνάρτηση Loop() διαχειρίζεται τις παραπάνω συναρτήσεις .Με ένα button γίνεται ο έλεγχος της λειτουργίας ,με την εκκίνηση του προγράμματος η πρώτη λειτουργία που εκτελείται είναι το mode 1 που εμφανίζει τις θερμοκρασίες περιβάλλοντος στην οθόνη και ανάλογα με την θερμοκρασία εκτελεί κάποιες λειτουργίες .Όταν η θερμοκρασία κυμαίνεται από 20 μέχρι 25 βαθμούς δεν εκτελείται καμία λειτουργία, όταν ξεπεράσει τους 25 βαθμούς ξεκινάει ο ανεμιστήρας την λειτουργία του μαζί με το μπλε led όσο ανεβαίνει η θερμοκρασία τόσο μεγαλύτερη ταχύτητα αποκτά ο κινητήρας και τόσο πιο φωτεινό είναι το λαμπάκι τέλος όταν οι θερμοκρασίες είναι κάτω των 20 βαθμών ανοίγει το κόκκινο λαμπάκι με μέγιστη φωτεινότητα όταν φτάσει 0 βαθμούς. Με την πίεση του button αλλάζει το mode που γίνεται 2 με χειροκίνητο έλεγχο θερμοκρασίας . Οι λειτουργίες παραμένουν ίδιες και δεν υπάρχει σύνδεση μεταξύ τους.



Εικόνα 4.3.2β Σχεδιάγραμμα με draw.io

4 Συμπεράσματα.

Σκοπός της παρούσας πτυχιακής ήταν ο σχεδιασμός και η υλοποίηση ενός συστήματος θερμοστάτη πού έχει την δυνατότητα αυτοματοποιημένης ρύθμισης της θερμοκρασίας ενός εσωτερικού χώρου σύμφωνα με τις εξωτερικές θερμοκρασίες που επικρατούν στο συγκεκριμένο περιβάλλον που τοποθετήθηκε ο αισθητήρας με την δυνατότητα και χειροκίνητης ρύθμισης της θερμοκρασίας. Επιπλέον δημιουργήθηκε εφαρμογή για την προστασία από πυρκαγιά την έγκαιρη ειδοποίηση και την απόσβεση της . Για την δημιουργία του συστήματος έγινε χρήση του μικροεπεξεργαστή Arduino uno R3 που σε συνδυασμό με τα απαραίτητα εξαρτήματα μπορεί να συνδυάσει λειτουργίες χρήσης παρόμοιας με άλλα πιο ακριβά συστήματα με λιγότερο κόστος, το μεγάλο πλεονέκτημα της συγκεκριμένης πλακέτας είναι ο μεγάλος αριθμός συσκευών που μπορούν να συνδεθούν με σκοπό την υλοποίηση μιας εφαρμογής . Με την περάτωση της εφαρμογής με Arduino καταλήξαμε σε μερικά συμπεράσματα όσον αφορά τις δυνατότητες και τους περιορισμούς προκύπτουν από την κατασκευή του . Οι αισθητήρες που χρησιμοποιήθηκαν έχουν απόκλιση από τις πραγματικές τιμές με αποτέλεσμα να είναι ακατάλληλη για χρήση τους σε πραγματικές συνθήκες. Ο αισθητήρας θερμοκρασίας και υγρασίας DHT11 εμφανίζει καθυστέρηση στην λήψη των τιμών καθώς και απώλεια της τάξης των 2-3^{ων} βαθμών κελσίου. Ο αισθητήρας φλόγας λειτουργεί με ακρίβεια το βράδυ σε περιορισμένη απόσταση βέβαια δεδομένο που τον κάνει ακατάλληλο για χρήση σε πραγματικές συνθήκες και την μέρα οι ακτίνες του φωτός μπερδεύουν τον αισθητήρα με αποτέλεσμα να εμφανίζει την ένδειξη της φωτιάς. Για την επιδείξει των λειτουργιών της ψύξης του χώρου όσο ανεβαίνει η θερμοκρασία έγινε χρήση ενός κινητήρα που χρησιμοποιήθηκε σαν ανεμιστήρας σε συνδυασμό με ένα μπλε λαμπάκι led και στην αντίθετη περίπτωση της πτώσης της θερμοκρασίας ένα κόκκινο led που αντιπροσωπεύει την θέρμανση . Το συγκεκριμένο σύστημα στο μέλλον μπορεί να υποστηρίξει δυνατότητες IoT(Internet of Things) με την προσθήκη των κατάλληλων επεκτάσεων (Wifi ,Bluetooth module). Με την σύνδεση της πλακέτας σέ μια πιο σταθερή πηγή ρεύματος και με την δημιουργία της κατάλληλης εφαρμογής αλληλεπίδρασης διαθέτει τις προϋποθέσεις για απομακρυσμένη σύνδεση μέσω μια ηλεκτρονικής συσκευής (κινητό, ταμπλέτα, υπολογιστής) για έλεγχο συσκευών θερμότητας και ψύξης .Επίσης θα μπορούσε να χρησιμοποιηθεί σε οποιοδήποτε περιβάλλον που απαιτείται η παρατήρηση θερμοκρασίας και η λήψη αποφάσεων για την θερμοκρασία με τις απαραίτητες τροποποιήσεις στις ανάγκες που θα παρουσιαστούν .

5 ΒΙΒΛΙΟΓΡΑΦΙΑ.

[1]. Tammy Noergaard, “The Embedded Systems Architecture”, Elsevier, 2012 ,Url Link

[2]. Μηνάς Δασύγενης, Δημήτριος Σούντρης, “Ενσωματωμένα Συστήματα – Ο αθέατος ψηφιακός κόσμος”, Ελληνικά Ακαδημαϊκά Ηλεκτρονικά Συγγράμματα και Βοηθήματα, 2015. Url Link

[3]. John Wiley, Sons Inc “Arduino Projects for Dummies”,2013, Url Link

[4]. Παναγιώτης Παπάζογλου , “Μικροεπεξεργαστές, Αρχές και Εφαρμογές” Εκδόσεις Τζιόλα ,2015

[5]. Αριστείδης Μπούρας, Γιάννης Κάππος, “Arduino, Αλγοριθμική ,Προγραμματισμός και Εφαρμογές ”, Εκδόσεις Κλειδάριθμος ,2021

Wikipedia

[6]. Eniac

[7]. Wiring

[8]. Tinkerkad

[9]. Μικροελεγκτής

[10]. Processing

[11]. I2C

[12]. Arduino

[13]. Visual Code Studio

Arduino.CC

[14]. Arduino

[15]. Language reference

[16]. Examples

[17]. Libraries

[18]. Arduino IDE

[19]. Arduino Uno R3

Youtube

[20]. Armit Aryal ,millis()fuction .

[21]. Element14 presents , LCD .

[22].Enjoy Mechatronics , DC motors using L298N Driver

[23]. Useful Electronics, SM32 DC MOTOR Speed Control Using ULN2003