



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ**
UNIVERSITY OF PATRAS

**ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΔΙΟΙΚΗΣΗΣ
ΕΠΙΧΕΙΡΗΣΕΩΝ**

ΤΜΗΜΑ ΔΙΟΙΚΗΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

**ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ ΜΕ ΤΗ ΧΡΗΣΗ
ΜΙΚΡΟΕΛΕΓΚΤΩΝ ARDUINO
ΚΑΤΑΣΚΕΥΗ SMART LAMP**

ΥΛΟΠΟΙΗΣΗ ΑΠΟ ΤΟΥΣ :

- **ΚΟΤΤΑΡΑΣ ΓΕΩΡΓΙΟΣ**
- **ΚΑΤΣΩΛΗΣ ΑΘΑΝΑΣΙΟΣ**

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ :

- **ΠΑΠΑΔΟΠΟΥΛΟΣ ΔΗΜΗΤΡΙΟΣ**

ΠΑΤΡΑ,2020

ΕΥΧΑΡΙΣΤΙΕΣ

Με την ευκαιρία της Πτυχιακής εργασίας θα θέλαμε να ευχαριστήσουμε τον κύριο Παπαδόπουλο Δημήτριο, τον υπεύθυνο της εργασίας μας, για όλη του την καθοδήγηση και βοήθεια. Επίσης θα θέλαμε να ευχαριστήσουμε την οικογένεια μας για τη στήριξη!

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία αναφέρεται στην μελέτη και υλοποίηση εφαρμογής πάνω στην πλατφόρμα arduino Uno. Στην εργασία αυτή θα κατασκευάσουμε μια λάμπα η οποία θα τροφοδοτείται με μπαταρίες και κάνοντας χρήση της πλατφόρμας μας θα προγραμματίσουμε την λειτουργία της χρησιμοποιώντας κώδικα και θα την μετατρέψουμε σε μια smart lamp με διάφορες λειτουργίες. Θα παρατηρήσουμε βήμα βήμα την διαδικασία υλοποίησης από το στάδιο κατασκευής μέχρι το στάδιο προγραμματισμού καθώς και τα αποτελέσματα μας και την μελλοντική εξέλιξη πάνω σε αυτό . Σκοπός μας είναι να δείξουμε τι μπορεί να αποφέρει η χρήση τεχνολογίας μέσα στην καθημερινότητα μας καθώς και τις ευκολίες που θα μας προσφέρει.

ABSTRACT

The present dissertation refers to the study and implementation of an application on the arduino Uno platform. In this work we will build a lamp that will be powered by batteries and using our platform we will program its operation using code and turn it into a smart lamp with various functions. We will observe step by step the implementation process from the construction stage to the planning stage as well as our results and future developments on it. Our goal is to show what the use of technology can bring in our daily lives as well as the facilities it will offer us.

Λέξεις κλειδιά

Arduino uno, led, lamp, lcd οθόνη, ακροδέκτες, γλώσσα προγραμματισμού, περιββαλον ide.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT.....	4
1.2 ΙΣΤΟΡΙΑ ARDUINO	6
1.3 ΕΙΣΑΓΩΓΗ(ΤΙ ΕΙΝΑΙ ΤΟ ARDUINO UNO)	7
1.4 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ARDUINO UNO	8
1.4 ΑΝΑΛΥΣΗ ARDUINO	9
1.5 ΑΝΑΛΥΣΗ ΑΚΡΟΔΕΚΤΩΝ	10
1.6 ΑΝΑΛΥΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ	11
1.7 ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ ΚΑΙ ΛΕΙΤΟΥΡΓΙΕΣ	12
1.8 ΨΗΦΙΑΚΕΣ ΑΚΙΔΕΣ (Digital pins)	17
1.9 ΑΝΑΛΟΓΙΚΕΣ ΑΚΙΔΕΣ ΕΙΣΟΔΟΥ (Analog input pins).....	18
1.10 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ARDUINO UNO.....	19
2.1 ΥΛΙΚΑ ΥΛΟΠΟΙΗΣΗΣ	20
2.2 BREADBOARD.....	23
2.3 ΣΥΝΔΕΣΜΟΛΟΓΙΑ	25
2.4 ΑΝΑΛΥΣΗ ΣΥΝΔΕΣΜΟΛΟΓΙΑΣ.....	26
2.5 ΥΛΟΠΟΙΗΣΗ ΣΥΝΔΕΣΜΟΛΟΓΙΑΣ ΣΤΑΔΙΟ 1ο.....	28
2.6 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ARDUINO	29
2.7 ΚΩΔΙΚΑΣ ARDUINO IDE	30
3.1 ΠΡΟΓΡΑΜΜΑ ΣΤΟ ARDUINO IDE	32
3.2 ΑΠΟΤΕΛΕΣΜΑΤΑ	33
3.3 ΕΠΙΠΛΕΟΝ ΣΤΑΔΙΑ ΥΛΟΠΟΙΗΣΗΣ	34
3.4 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ TEMP SENSOR	35
3.5 ΠΕΡΙΒΑΛΛΟΝ IDE	36
3.6 ΑΠΟΤΕΛΕΣΜΑΤΑ ΣΕΙΡΙΑΚΗΣ ΟΘΟΝΗΣ	37
3.7 LCD SCREEN.....	38
3.8 ΠΕΡΙΒΑΛΛΟΝ ΟΘΟΝΗΣ ΣΕ IDE	39
3.9 ΣΥΜΠΛΗΡΩΜΑΤΙΚΟ PROJECT	40
3.10 ΚΩΔΙΚΑΣ ΠΡΟΓΡΑΜΜΑΤΟΣ.....	41
3.11 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	42
3.12 ΜΕΛΟΝΤΙΚΕΣ ΠΡΟΤΑΣΕΙΣ	43
Βιβλιογραφία	44

1.1 ΙΣΤΟΡΙΑ ARDUINO

Το 2005 ο Massimo Banzi και ο David Cueartielles ξεκίνησαν να φτιάχνουν μια συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο φθηνή από άλλα πρωτότυπα συστήματα που υπήρχαν εκείνη την περίοδο. Οι ιδρυτές ονόμασαν την συσκευή από τον Arduino İverea και ξεκίνησαν να παράγουν πλακέτες σε ένα εργαστήριο στην Ιβρέα κωμόπολη του Τορίνο της Ιταλίας. Το σχέδιο Arduino είναι μία διακλάδωση της πλατφόρμας Wiring για λογισμικό ανοικτού κώδικα και προγραμματίζεται χρησιμοποιώντας μια γλώσσα βασισμένη στο Wiring (σύνταξη και βιβλιοθήκες) παρόμοια με την C++ με απλοποιήσεις και αλλαγές, καθώς και ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE)

- Οι βασικές πλακέτες (Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro, Arduino ADK, Arduino DUE....)
- Οι πλακέτες με πρόσβαση στο Internet (Arduino Ethernet)
- Τα shields για Arduino (Wi-Fi Shield, Motor Shield, Ethernet Shield, SD Shield...)

Οι διαφορές που έχουν μεταξύ τους οι βασικές πλακέτες Arduino είναι συνήθως στον επεξεργαστή και στο πλήθος των pins (Input / Output)

Εμείς θα ασχοληθούμε με το Arduino υπο η οποία είναι η πιο διαδεδομένη πλακέτα στις μέρες μας

	Arduino Leonardo	Arduino Uno Rev3	Arduino Mega 2560 Rev3
Microcontroller	ATmega32u4	ATmega328	ATmega2560
Operating Voltage	5V	5V	5V
Input Voltage	7-12V	7-12V	7-12V
Input Voltage (limits)	6-20V	6-20V	6-20V
Digital I/O Pins	20	14	54
PWM Channels	7	6	15
Analog Input Channels	12	6	16
DC Current per I/O Pin	40 mA	40 mA	40 mA
DC Current for 3.3V Pin	50 mA	50 mA	50 mA
Flash Memory	32 KB	32 KB	256 KB
SRAM	2.5 KB (ATmega32u4)	2 KB (ATmega328)	8 KB
EEPROM	1 KB (ATmega32u4)	1 KB (ATmega328)	4 KB
Clock Speed	16 MHz	16 MHz	16 MHz
Price	25€	26€	59€

Εικόνα 1

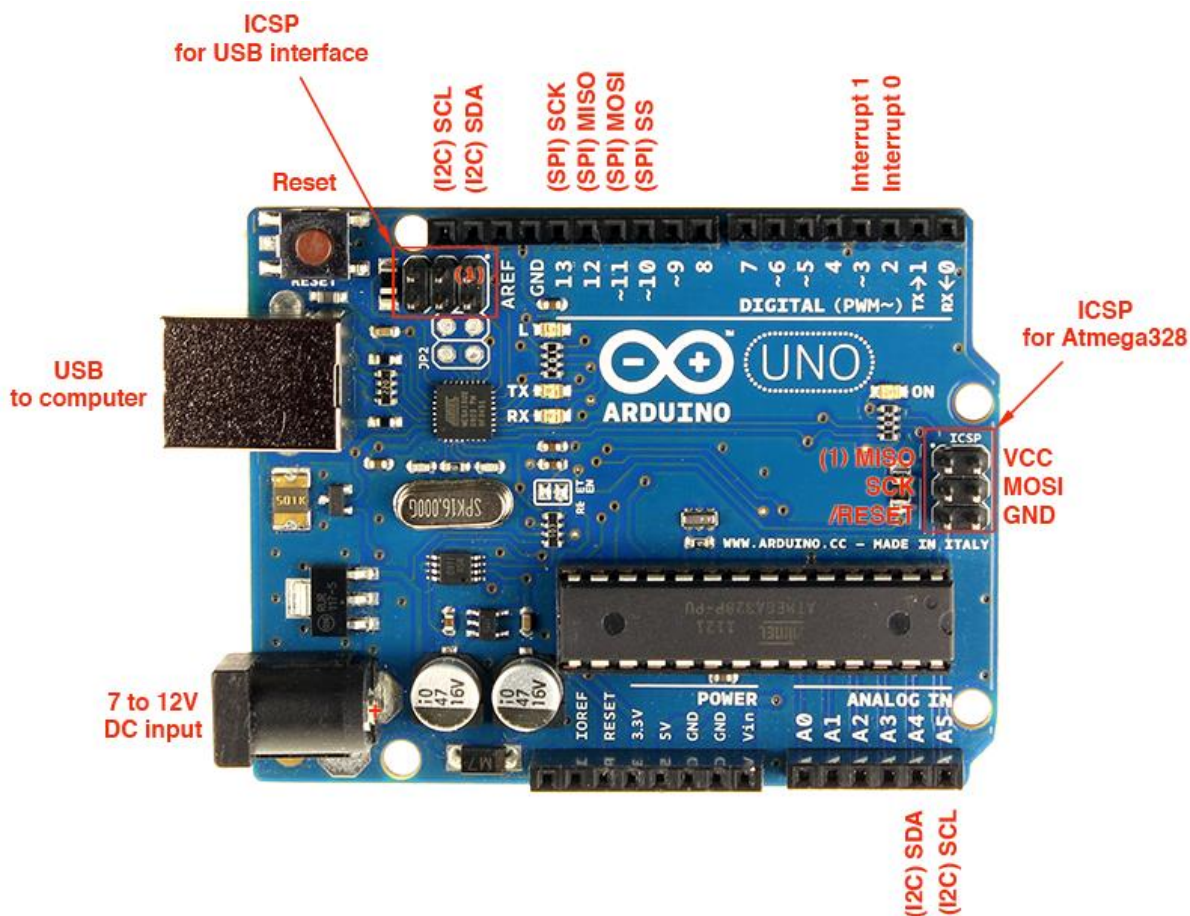
1.2 ΕΙΣΑΓΩΓΗ(ΤΙ ΕΙΝΑΙ ΤΟ ARDUINO UNO)

Το arduino uno είναι μια πλακέτα ανοικτού κώδικα “open source” η οποία μας δίνει την δυνατότητα να κατασκευάζουμε εφαρμογές και να υλοποιούμε συστήματα αυτοματισμού και ρομποτικής . Ο επεξεργαστής του είναι ο ATmega από την ATMEL . Αποτελείτε από μονάδες εισόδου και έξοδου οι οποίες είναι αναλογικές και ψηφιακές. Ονομαστικά συσκευές που συνδέουμε στο Aduino uno : αντιστάσεις , φώτα , servo-μοτοράκια , αισθητήρες πίεσης , θερμοκρασίας , υγρασίας κ.α . Ο προγραμματισμός του γίνεται συνδέοντας το μέσω της θύρας usb που έχει στον υπολογιστή . Κάνοντας το αυτό μπορούμε να προγραμματίζουμε το arduino άλλα και να δεχόμαστε πληροφορίες από αυτό για της συσκευές που έχει συνδεδεμένες πάνω του . Το πρόγραμμα που χρησιμοποιεί είναι μια παραλλαγή της c++ - java και υποστηρίζει τις ίδιες συναρτήσεις και βιβλιοθήκες . Είναι εύκολο για κάποιον που γνωρίζει βασικές έννοιες προγραμματισμού . Διατίθεται δωρεάν και ονομάζεται arduino IDE .

1.3 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ARDUINO UNO

Το Arduino Uno έχει τα εξής χαρακτηριστικά:

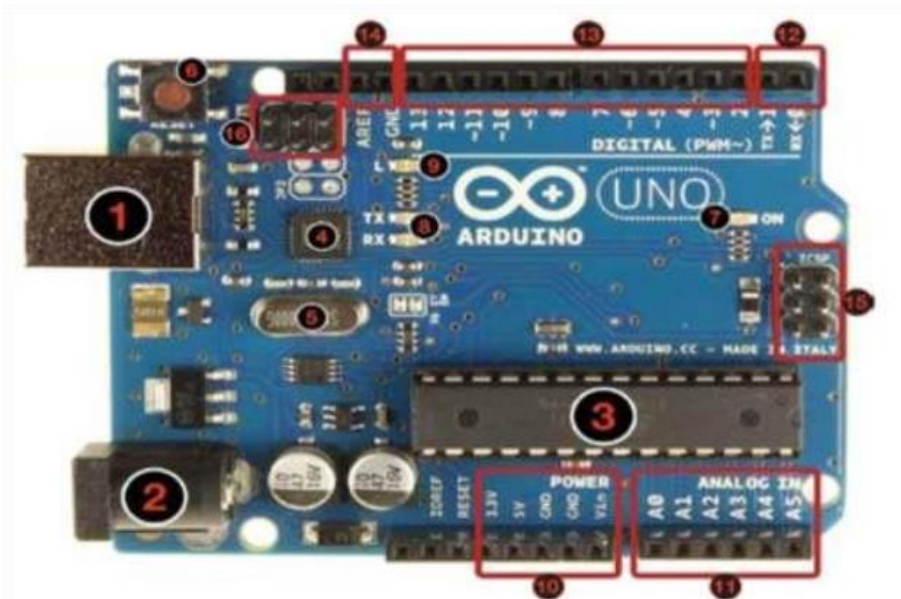
- Επεξεργαστής ATMEGA328
- Ρεύμα λειτουργίας 5V
- Ρεύμα εισόδου από 7 έως 12 volt
- 14 ψηφιακοί ακροδέκτες εκ των οποίων οι 6 χρησιμοποιούνται για PWM έξοδο
- 6 αναλογικοί ακροδέκτες εισόδου
- Ισχύς συνεχόμενου ρεύματος ανά ακροδέκτη 40mA
- Ισχύς συνεχόμενου ρεύματος για ακροδέκτη τάσης 3.3V 50mA
- Μνήμη flash 32KB
- Μνήμη SRAM 2KB
- Μνήμη EEPROM 1KB
- Ταχύτητα cpu 16MHz



Εικόνα 2

1.4 ΑΝΑΛΥΣΗ ARDUINO

- 1:υποδοχή σύνδεσης με usb
- 2:υποδοχή σύνδεσης ρεύματος
- 3:επεξεργαστής arduino uno
- 4:chip επικοινωνίας
- 5:16 MHz κρυστάλλου
- 6:reset button
- 7:led λειτουργίας
- 8:led ένδειξης επικοινωνίας arduino με υπολογιστή
- 9:led εντοπισμού σφαλμάτων
- 10:ακροδέκτες ρεύματος
- 11:Αναλογικές Είσοδοι
- 12:TX και RX ακροδέκτες
- 13:Ψηφιακοί ακροδέκτες
- 14:γείωση και AREF ακροδέκτης.
- 15:ICSP Σειριακός προγραμματισμός κυκλώματος Atmega328
- 16:ICSP Σειριακός προγραμματισμός κυκλώματος με χρήση usb



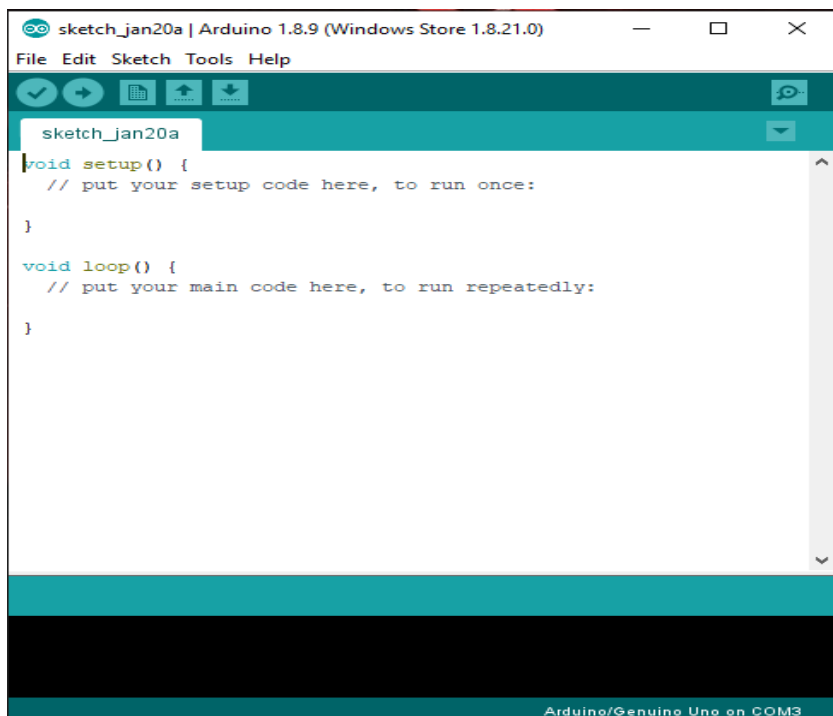
Εικόνα 3

1.4 ΑΝΑΛΥΣΗ ΑΚΡΟΔΕΚΤΩΝ

- **Οι ακροδέκτες ρεύματος** αποτελούνται από(βλέπουμε στην παραπάνω εικόνα με αριθμό 10):
 - 3.3 volt εισόδου
 - 5 volt εισόδου
 - GND: Γείωση
 - VIN: Τάση εισόδου μεγαλύτερη τον 5volt κάποιες φορές θα την δούμε και ως 9v
- **Οι αναλογικοί ακροδέκτες**(βλέπουμε στην παραπάνω εικόνα με αριθμό 11): Εκεί συνδέουμε αναλογικά συστατικά, όπως τα ποτενσιόμετρα και άλλους αισθητήρες. Μπορούν να πάρουν αναλογικές τιμές από 0 έως 1023.
- **Οι ψηφιακοί ακροδέκτες**(βλέπουμε στην παραπάνω εικόνα με αριθμό 13): Λειτουργούν ως εισροές ή εκροές και τις ορίζουμε με την εντολή pinMode(). Τα κυματάκια με σχέδιο "~" μπροστά από τα νούμερα είναι PWM (Pulse Width Modulation) έξοδοι και έχουν τη δυνατότητα να μιμηθούν αναλογικές εξόδους με την εντολή analogWrite ().

1.6 ΑΝΑΛΥΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ

Το περιβάλλον ανάπτυξης (arduino ide) περιέχει : ένα παράθυρο με επεξεργασία κειμένου, για την σύνταξη του κώδικα ένα παράθυρο το οποίο εμφανίζονται μηνύματα , μια γραμμή εργαλείων καθώς και μια μπάρα με τις βασικές ρυθμίσεις του (βλέπουμε στην εικόνα 4) . Το πρόγραμμα ονομάζει τον κώδικα σαν sketch και δίπλα βάζει την ημερομηνία υλοποίησης . Όπως αναφέραμε και στην εισαγωγή επειδή αντλεί πληροφορίες από java και c++ το ide έχει ένα πρακτικό περιβάλλον για την συγγραφή κώδικα , κάνει χρήση βιβλιοθηκών , έχει compiler για την διόρθωση άλλα για την ορθή λειτουργία του προγράμματος (μεταγλώττιση) , μια σειριακή οθόνη που μας δίνει τις πληροφορίες από ότι είναι συνδεδεμένο πάνω στο arduino καθώς και την επιλογή να φορτώσουμε τον κώδικα στο arduino . Η κώδικας του arduino στηρίζεται σε μια γλώσσα wiring όπως προαναφέραμε που θυμίζει C++ -java και χρησιμοποιείτε για μικροελεγκτες αρχιτεκτονικής AVR όπως είναι και ο ATmega. Ο compiler που χρησιμοποιεί είναι AVR gcc και σαν βασική βιβλιοθήκη την AVR libe . Επειδή αντλεί τα βασικά στοιχεία του από την c++ μπορούμε να χρησιμοποιούμε τις ίδιες εντολές και συναρτήσεις , επίσης έχουμε και τους ίδιους τύπους δεδομένου και τελεστών . Εκτός όμως από όλα αυτά υπάρχουν και κάποιες ειδικές εντολές που θα δούμε παρακάτω που βοηθούν στην διαχείριση και στην λειτουργία του arduino .



Εικόνα 4

1.7 ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ ΚΑΙ ΛΕΙΤΟΥΡΓΙΕΣ

Δομές ελέγχου ροής:

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

Αριθμητικοί τελεστές

- = (τελεστής εκχώρησης)
- + (τελεστής πρόσθεσης)
- - (τελεστής αφαίρεσης)
- * (τελεστής πολλαπλασιασμού)
- -/ (τελεστής διαίρεσης)
- % (τελεστής υπόλοιπου ακεραίας διαίρεσης)

Λογικοί τελεστές

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

Δυαδικοί τελεστές

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

Τελεστές αύξησης και μείωσης

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

Σύνθετοι τελεστές

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

Τελεστές σύγκρισης

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

Τελεστές δεικτών

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

Σταθερές

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

Τύποι δεδομένων

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

Συναρτήσεις μετατροπής τύπων

- char(), byte()
- int(), word(), long()
- float(), double()

Συναρτήσεις εισόδου και εξόδου

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

Συναρτήσεις ψηφιακής εισόδου και εξόδου

- digitalWrite() (γράφει σε μία ψηφιακή επαφή έξοδο)
- digitalRead() (διαβάζει από μία ψηφιακή επαφή είσοδο)

Προηγμένες συναρτήσεις εισόδου και εξόδου

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

Συναρτήσεις αναλογικής εισόδου και εξόδου

- analogReference() (ορίζει την τάση αναλογικής αναφοράς)
- max() (βρίσκει τον μεγαλύτερο ανάμεσα σε δύο αριθμούς)
- min() (βρίσκει τον μικρότερο ανάμεσα σε δύο αριθμούς)
- abs() (επιστρέφει την απόλυτη τιμή ενός αριθμού)
- constrain() (ελέγχει για υπερχείλιση ή υποχείλιση ορίων)
- map() (πραγματοποιεί γραμμικό μετασχηματισμό ορίων)
- analogWrite() (γράφει PWM σήματα σε μία επαφή εξόδου)
- analogRead() (διαβάζει από μία αναλογική επαφή εισόδου)

Συναρτήσεις χρόνου

- millis() (διάρκεια εκτέλεσης του προγράμματος σε ms)
- micros() (διάρκεια εκτέλεσης του προγράμματος σε μs)
- delay() (παύση προγράμματος -η διάρκεια δίδεται σε ms)
- delayMicroseconds() (παύση προγράμματος -η διάρκεια δίδεται σε μs)

Μαθηματικές και Τριγωνομετρικές συναρτήσεις

- pow() (επιστρέφει το αποτέλεσμα μίας δύναμης)
- sqrt() (επιστρέφει την ρίζα ενός αριθμού)
- sin() (υπολογίζει το ημίτονο ενός αριθμού)
- cos() (υπολογίζει το συνημίτονο ενός αριθμού)
- tan() (υπολογίζει την εφαπτομένη ενός αριθμού)

Συναρτήσεις γεννήτριας ψευδοτυχαίων αριθμών

- random() (δίδεται ένας νέος αριθμός από την γεννήτρια)
- randomSeed() (θέτει τον σπόρο της γεννήτριας παραγωγής)

Συναρτήσεις επεξεργασίας δυαδικών αριθμών

- lowByte() (επιστρέφει το δεξιότερο byte μίας μεταβλητής)
- highByte() (επιστρέφει το αριστερότερο byte μίας μεταβλητής)
- bitRead() (διαβάζει ένα συγκεκριμένο ψηφίο μίας μεταβλητής)
- bitWrite() (γράφει σε ένα συγκεκριμένο ψηφίο μίας μεταβλητής)
- bitSet() (γράφει την τιμή 1 σε κάποιο ψηφίο μίας μεταβλητής)
- bitClear() (γράφει την τιμή 0 σε κάποιο ψηφίο μίας μεταβλητής)
- bit() (υπολογίζει μία συγκεκριμένη δύναμη με βάση το 2)

Συναρτήσεις χρήσης ρουτινών εξυπηρέτησης διακοπών

- attachInterrupt() (ενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής)
- detachInterrupt() (απενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής)

Συναρτήσεις ενεργοποίησης και απενεργοποίησης διακοπών

- interrupts() (ενεργοποιεί τα σήματα διακοπής)
- noInterrupts() (απενεργοποιεί τα σήματα διακοπής)

Υποστήριξη σειριακής επικοινωνίας

- Serial (αντικείμενο σειριακής επικοινωνίας με χρήσιμες μεθόδους)

1.8 ΨΗΦΙΑΚΕΣ ΑΚΙΔΕΣ (DIGITAL PINS)

Οι ακίδες (ακροδέκτες) αυτές στο Arduino μπορούν να χρησιμοποιηθούν είτε ως είσοδοι είτε ως έξοδοι, σαν προεπιλογή(default) είναι ρυθμισμένες ως είσοδοι . Οι συναρτήσεις ψηφιακής εισόδου και έξοδου που χρησιμοποιούμε είναι οι :

pinMode (): η οποία είναι η βασική εντολή και με αυτήν επιλέγουμε την συμπεριφορά του ακροδέκτη με το αν θα είναι είσοδος η έξοδος.

- Εντολή :pinMode(pin, mode)
- pin: Είναι ο αριθμός του ακροδέκτη που θα επιλέξουμε
- mode: INPUT/OUTPUT (μονάδα εισόδου η έξοδου αντίστοιχα)

digitalWrite (): Δίνει μια υψηλή (HIGH) ή μια χαμηλή (LOW) τιμή σε μια ψηφιακή ακίδα. Σε περίπτωση που έχει ρυθμιστεί ως έξοδος τότε η τάση της θα παει την τιμή 5V για HIGH και 0V για LOW. Αν η ακίδα έχει ρυθμιστεί ως είσοδος, γράφοντας HIGH στην θα ενεργοποιήσει μια αντίσταση των 20 K ενώ γράφοντας LOW θα την απενεργοποιήσει.

- Εντολή :digitalWrite(pin,value)
- pin: Είναι ο αριθμός του ακροδέκτη που θα επιλέξουμε
- Value: INPUT/OUTPUT (μονάδα εισόδου η έξοδου αντίστοιχα)

digitalRead (): Διαβάζει την τιμή από έναν συγκεκριμένο ψηφιακό ακροδέκτη και παίρνει τιμές HIGH/LOW.

- Εντολή: digitalRead(pin)
- pin: Είναι ο αριθμός του ακροδέκτη που θα επιλέξουμε
- Τιμές που επιστρέφει: HIGH/LOW

1.9 ΑΝΑΛΟΓΙΚΕΣ ΑΚΙΔΕΣ ΕΙΣΟΔΟΥ (ANALOG INPUT PINS)

Τα Arduino υπο μέσω του Atmega περιέχουν έναν ενσωματωμένο αναλογικό-σε-ψηφιακό μετατροπέα 6 καναλιών. Ο μετατροπέας αυτός διαθέτει ανάλυση 10 bit και έχει την δυνατότητα να επιστρέφει ακέραιους με τιμές από 0 έως 1023. Ενώ η κύρια λειτουργία των αναλογικών ακροδεκτών για τους περισσότερους χρήστες πάνω στο Arduino είναι να διαβάζει αναλογικούς αισθητήρες, οι αναλογικές ακίδες διαθέτουν και την δυνατότητα να συμπεριφέρονται όπως όλοι οι γενικοί ακροδέκτες εισόδου/εξόδου. Οι συναρτήσεις αναλογικής εισόδου και εξόδου είναι οι παρακάτω:

analogWrite():Γράφει μια αναλογική τιμή (PWM κύμα) σε έναν ακροδέκτη. Μπορεί να χρησιμοποιηθεί για παράδειγμα να ανάψει ένα LED σε διάφορες φωτεινότητες .Η εντολή αυτή λειτουργεί στις ακίδες 3, 5, 6, 9, 10, 11.

- Εντολή : analogWrite(pin, value)
- pin: Ο αριθμός του ακροδέκτη που θα γράψει επάνω
- value: ο κύκλος λειτουργίας με τιμές μεταξύ 0 και 255

analogRead(): Διαβάζει την τιμή από την καθορισμένη αναλογική ακίδα.

- Εντολή :analogRead(pin)
- pin: Ο αριθμός της αναλογικής ακίδας εισόδου από όπου θα διαβάζει
- Επιστέφει: ακέραιο από 0 έως 1023

1.10 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ARDUINO UNO

Πλεονεκτήματα

- Σε σχέση με τα υπόλοιπα μοντέλα που κυκλοφορούν είναι πολύ φτηνό(μέγιστο κόστος αγοράς 25 ευρώ)
- Είναι open source hardware
- Έχει δυνατότητα μεγάλης επεκτασιμότητας αν συγκρίνουμε την τιμή του σε σχέση με αυτά που μας προσφέρει
- Υποστήριξη από πολύ μεγάλο κομμάτι της κοινότητας
- Πλήθος tutorial καθώς και βοηθημάτων σε βιβλία άλλα και στο διαδίκτυο

Μειονεκτήματα

- Όπως και τα υπόλοιπα έτσι και αυτό, για την χρήση εξειδικευμένων επεκτάσεων για πιο δύσκολα project χρειάζεται γνώσεις πάνω στις βασικές έννοιες κυκλωμάτων.
- Επίσης χρειάζονται βασικές γνώσεις πάνω στην έννοια του προγραμματισμού άσχετα με τον αν γνωρίζουμε κάποιο περιβάλλον γλώσσας (java , c, κ.α)
- Χρήση κολλητηριου σε ειδικές περιπτώσεις

Σε γενικές γραμμές ίσως η καλύτερη επιλογή για κάποιον που αρχίζει και θέλει να εξοικειωθεί με την αυτοματοποίηση-ρομποτική. Μέγιστο θετικό ,η υλοποίηση κάθε project με πολύ χαμηλό κόστος.

2.1 ΥΛΙΚΑ ΥΛΟΠΟΙΗΣΗΣ

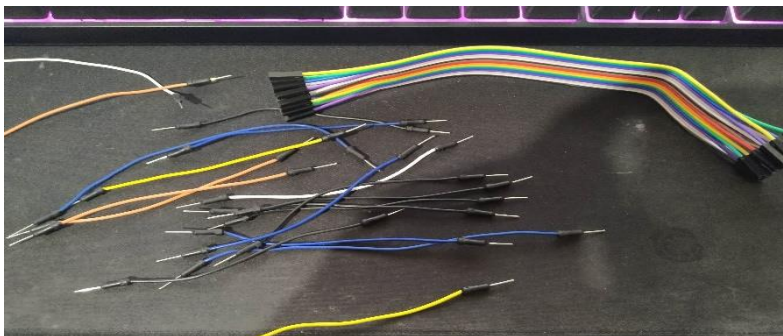
Τα υλικά που θα χρησιμοποιήσουμε αρχικά είναι :

1. Το arduino uno



Εικόνα 5

2. Διάφορα καλώδια για την συνδεσμολογία



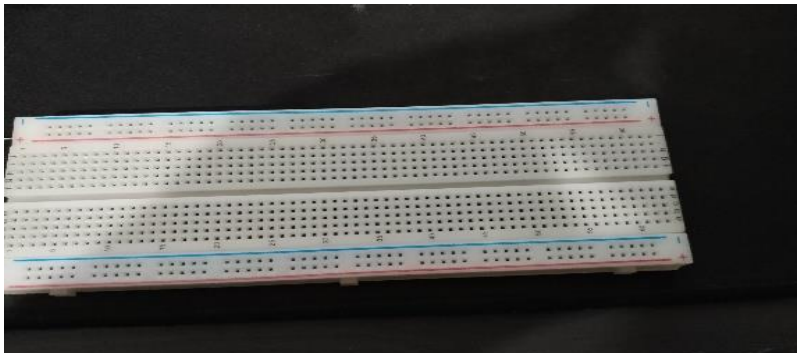
Εικόνα 6

3. Μια λάμπα με μπαταρίες



Εικόνα 7

4. Το breadboard



Εικόνα 8

5. Ένα τρανζίστορ, μια φωτοαντίσταση καθώς και αντιστάσεις(10KOhm)



Εικόνα 9

6. Έναν διακόπτη(button)



Εικόνα 10

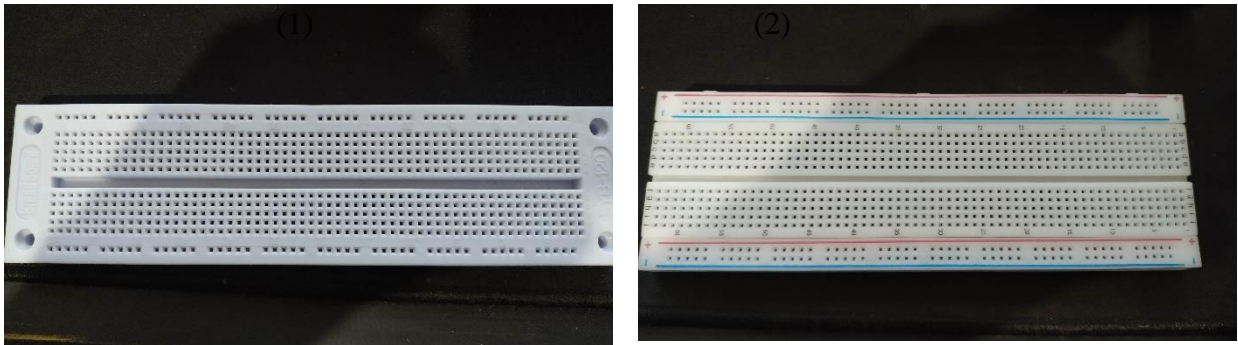
7. Ένα ποτενσιόμετρο



Εικόνα 11

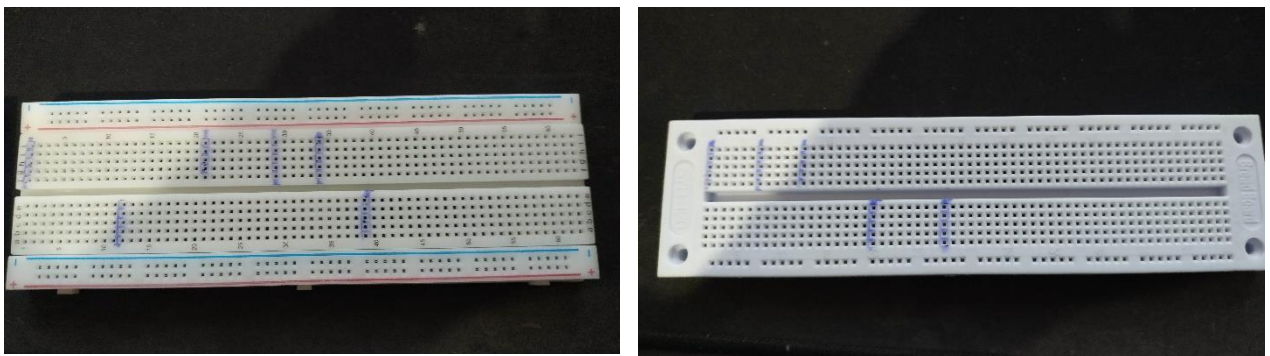
2.2 BREADBOARD

Σαν πρώτο στάδιο θα αναλύσουμε την συνδεσμολογία πάνω στο breadboard. Το breadboard όπως θα δούμε και στην παρακάτω εικόνα αποτελείται από διάφορες τρύπες οι οποίες συνδέονται μεταξύ τους από την κάτω μεριά. Εμείς θα δείξουμε δυο διαφορετικά είδη breadboard (το ένα για την πειραματική υλοποίηση(1) και το άλλο για την κανονική υλοποίηση του project(2)) τα οποία έχουν μια μικρή διαφορά μεταξύ τους.



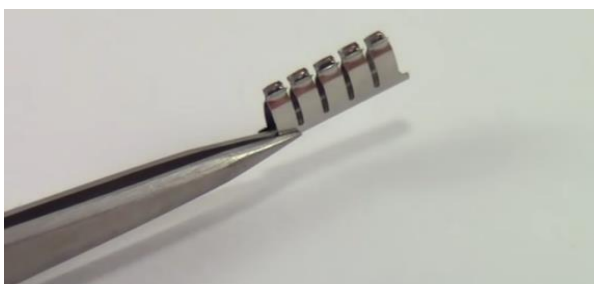
Εικόνα 12

Οι τρύπες και στα δυο δηλώνουν σημεία τα οποία επικοινωνούν μεταξύ τους. Όπως θα δούμε στην εικόνα παρακάτω και στα δυο breadboard με μπλε μαρκαδόρο είναι σημειωμένα τα σημεία αυτά.



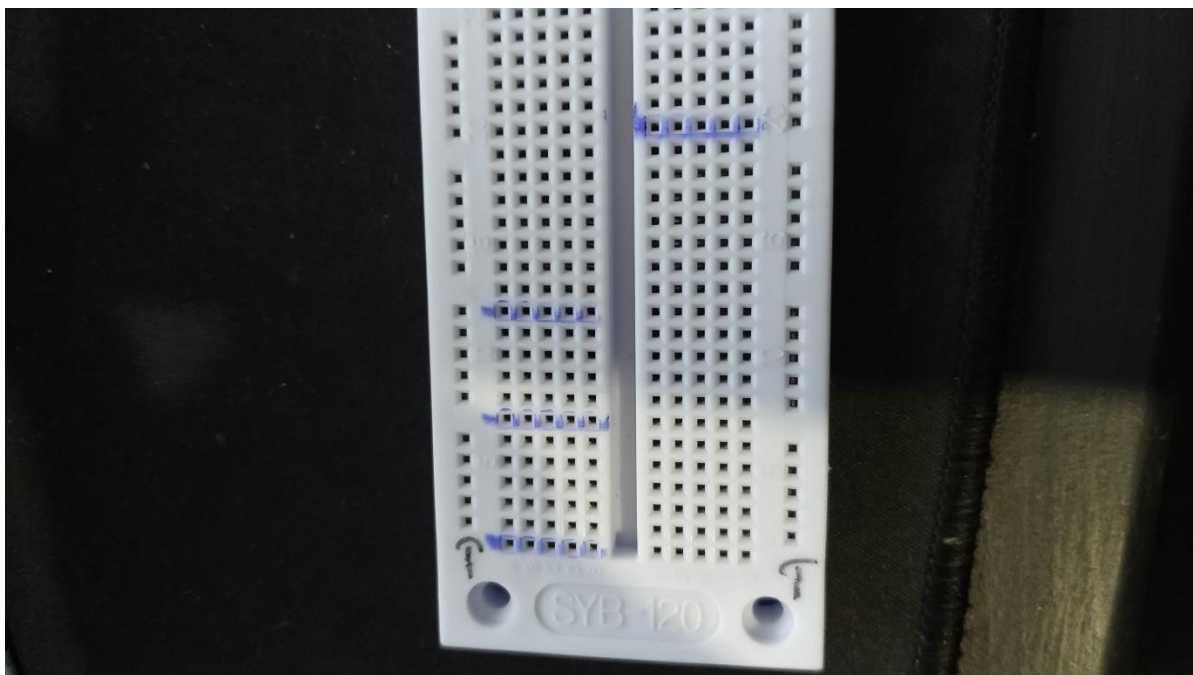
Εικόνα 13

Η επιλογή των σημείων είναι τυχαία. Αυτό που θέλουμε να δείξουμε είναι ότι τα σημεία είναι κάθετα και είναι 5 τα οποία επικοινωνούν μαζί. Μία επιπλέον φωτογραφία ίσως το δείξει καλύτερα!

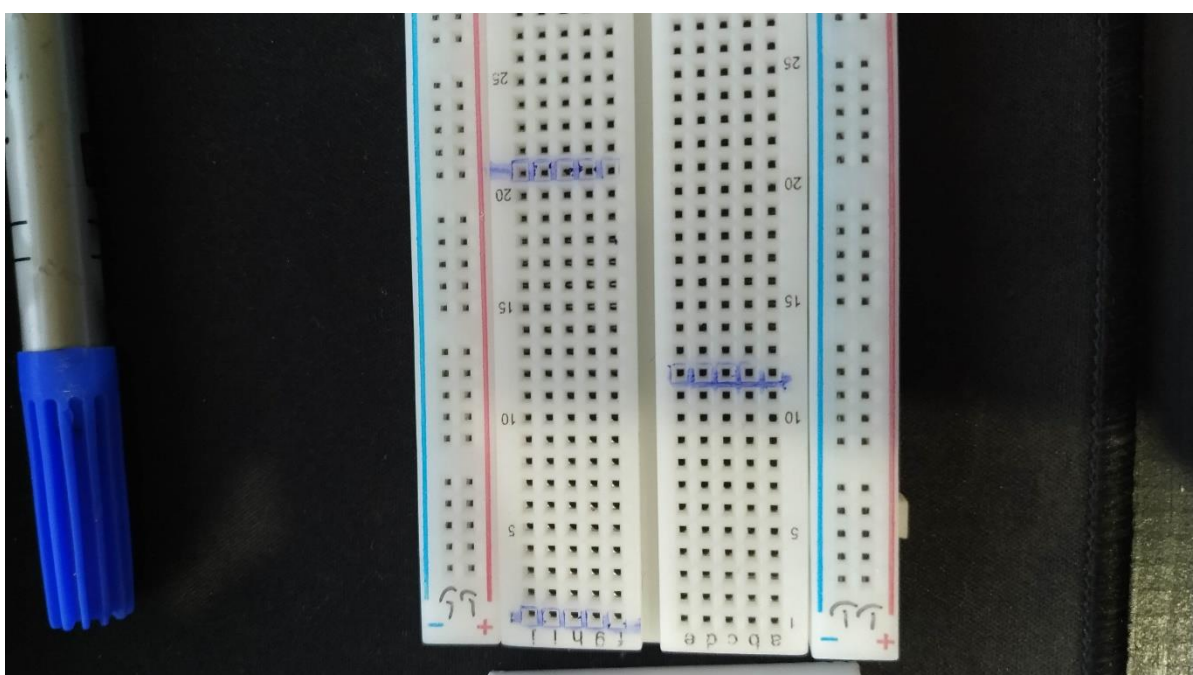


Εικόνα 14

Στη συνέχεια όπως είδαμε και στις παραπάνω εικόνες η διαφορά είναι εμφανής. Το ένα έχει δυο σημεία από κάθε μεριά ενώ το άλλο ένα. Αυτά είναι τα άκρα μας τα οποία τα χρησιμοποιούμε για θετικό και αρνητικό πόλο για την παροχή ρεύματος. Αυτό μας δηλώνει ότι στο ένα έχουμε δυο μεριές όπου μπορούμε την μια να την ορίσουμε θετική και την άλλη αρνητική και ανάποδα ανάλογα πάντα την συνδεσμολογία μας, ενώ στο άλλο μπορούμε να έχουμε και τους δυο πόλους από την μια μεριά. Στην ουσία είναι πιο εύχρηστο για μεγαλύτερα project με πολλά πράγματα και άνεση στην συνδεσμολογία.



Εικόνα 15



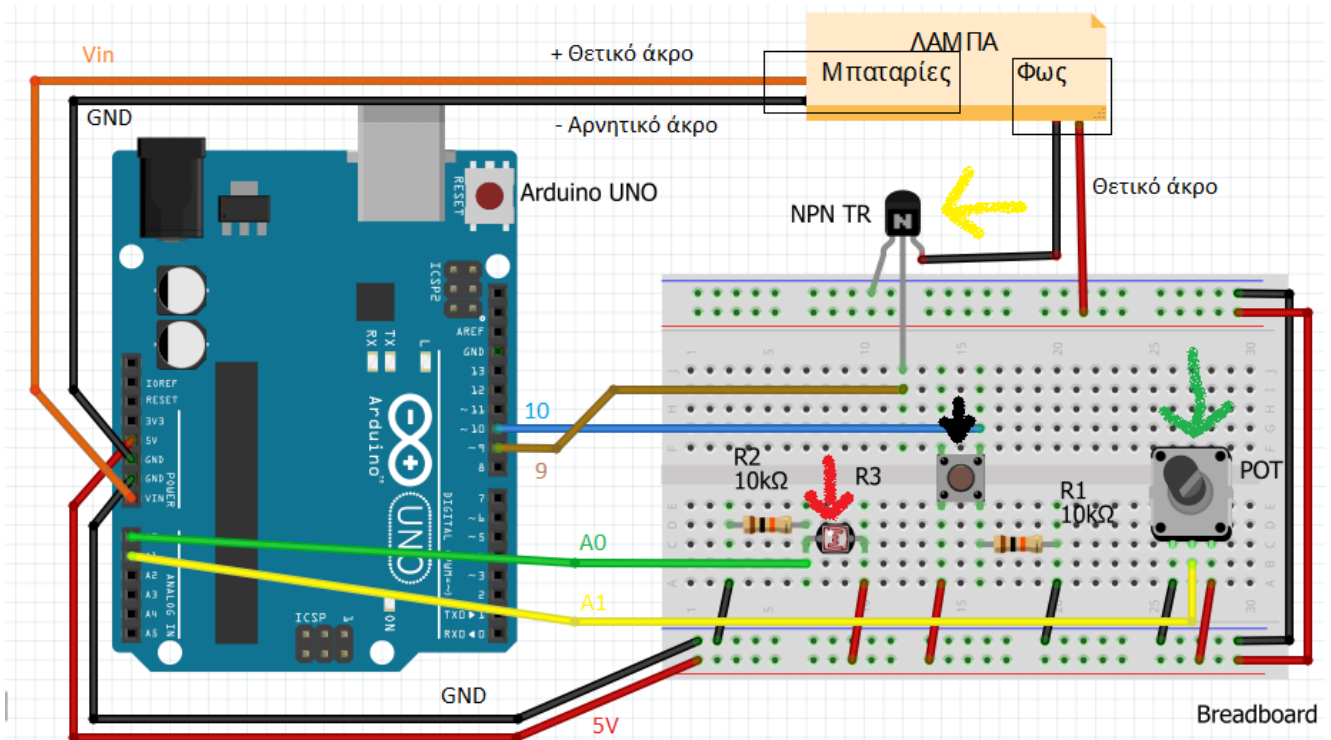
Εικόνα 16

2.3 ΣΥΝΔΕΣΜΟΛΟΓΙΑ

Αφού λοιπόν είδαμε τι είναι το breadboard θα συνεχίσουμε με την κατασκευή του κυκλώματος μας. Αναφέραμε πιο πάνω τα υλικά που θα χρησιμοποιήσουμε και τώρα θα δούμε το πλάνο υλοποίησης

Αναλυτικά τώρα πάνω στο πλάνο βλέπουμε ως εξής:

- Στην αριστερή μεριά βλέπουμε το arduino μας
- Στην δεξιά μεριά βλέπουμε το breadboard μας
- Με κόκκινο βελάκι παρατηρούμε την φωτοαντίσταση μας
- Με μαύρο βελάκι είναι ο διακόπτης μας
- Με πράσινο βελάκι το ποτενσιόμετρο μας
- Και με κριτίνο το τρανζίστορ μας το οποίο είναι αυτό που θα συνδεθεί πάνω στο breadboard μας

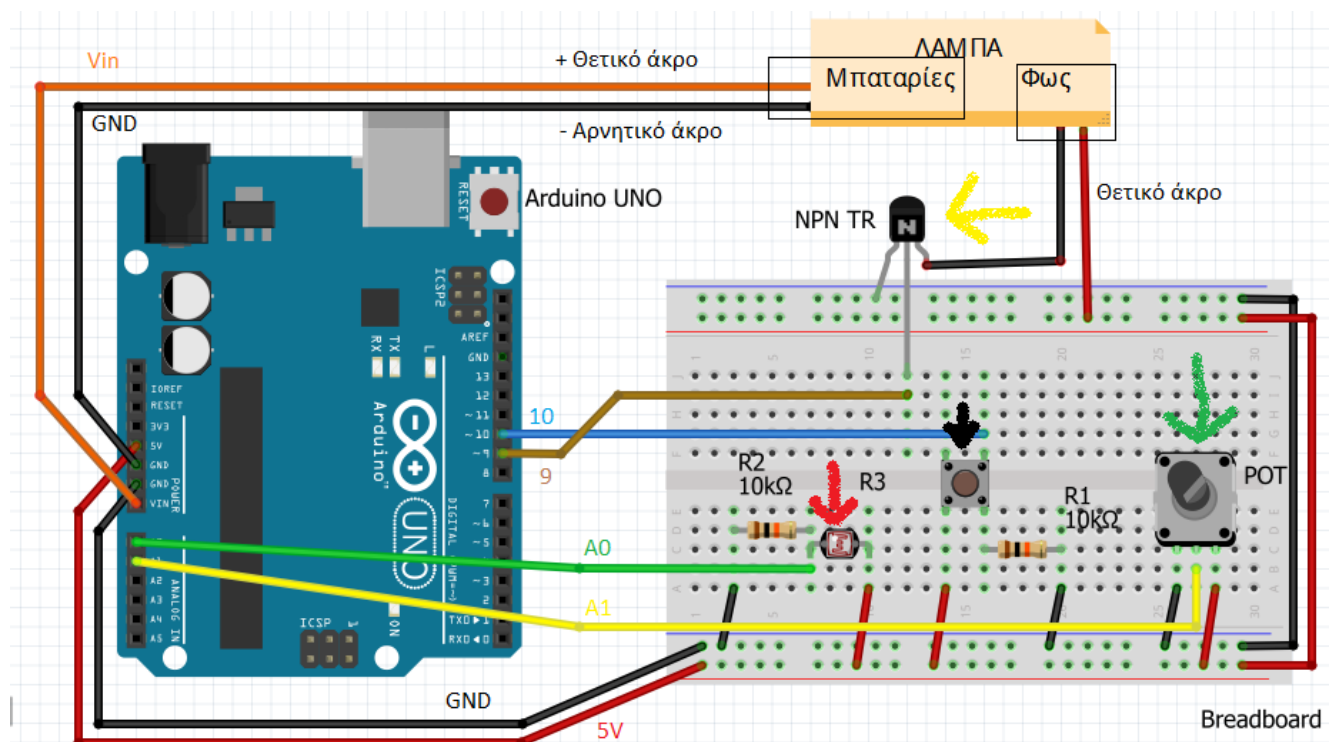


Εικόνα 17

2.4 ΑΝΑΛΥΣΗ ΣΥΝΔΕΣΜΟΛΟΓΙΑΣ

Πιο συγκεκριμένα ας ξεκινήσουμε με :

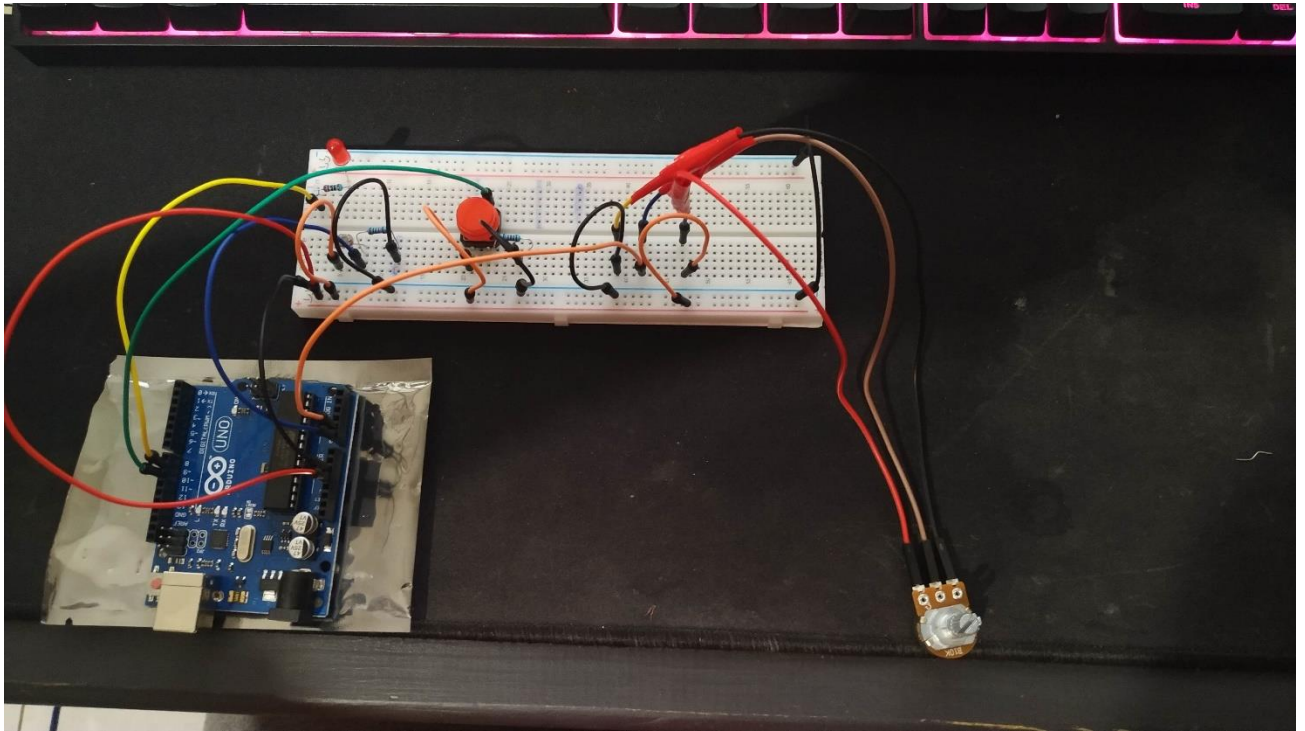
- Η φωτοαντίσταση μας συνδέεται σε δυο θέσεις πάνω στο breadboard όπου η δεξιά θέση της συνδέεται με καλώδιο στην θετική μεριά του breadboard και η αριστερή της με μια αντίσταση $10\text{k}\Omega$ όπως βλέπουμε και στην εικόνα .Η δεξιά μεριά της αντίστασης συνδέεται στην αρνητική μεριά του arduino ενώ η δεξιά μεριά της αντίστασης και η αριστερή μεριά της φωτοαντίστασης συνδέεται με καλώδιο στην θέση α0 πάνω στο arduino μας
- Το button μας συνδέεται σε τέσσερις θέσεις πάνω στο breadboard.η κάτω δεξιά μεριά του συνδέεται με μια αντίσταση $10\text{k}\Omega$ η οποία στην δεξιά της μεριά συνδέεται με την αρνητική μεριά του breadboard.η κάτω αριστερά μεριά συνδέεται με την θετική μεριά του breadboard και τέλος η πάνω δεξιά συνδέεται με καλώδιο στην θέση ~10 του arduino
- Το ποτενσιομετρο συνδέεται σε τρεις θέσεις στο breadboard.η δεξιά στην θετική μεριά του breadboard,η αριστερή στην αρνητική και τέλος η μεσαία με καλώδιο στην θέση α1 του arduino
- Το τρανζίστορ συνδέεται σε δυο θέσεις στο breadboard.η αριστερή του μεριά στην αρνητική μεριά του breadboard,η μεσαία με καλώδιο που συνδέεται στην θέση ~9 του arduino και η δεξιά με την λάμπα μας.



Εικόνα 18

- Όσον αφορά τις αντιστάσεις που χρησιμοποιούμε έχουν να κάνουν με τον νόμο του Ωμ. Ο Νόμος του Ωμ (Ohm) συνδέει την τάση, την ένταση και την αντίσταση. Όταν στα άκρα μιας αντίστασης R συνδέσουμε μια πηγή τάσης V τότε αυτή διαρρέεται από ρεύμα εντάσεως I. [$I = V/R$]. Πιο συγκεκριμένα : ένα led από αυτό που θα δούμε παρακάτω μπορεί να αντέξει ρεύμα μέχρι 20mA. Στα 20mA μας δίνει τη μέγιστη φωτεινότητά του. Από τον νόμο του Ωμ γνωρίζουμε ότι $I = V / R$. Ξέρουμε ότι σε ρεύμα περίπου 20mA ένα led έχει τάση 2.0 Volts και λειτουργεί με ασφάλεια. Άρα τα 2V από τα 5V πηγαίνουν στο led και τα 3V στην αντίσταση. Η αντίσταση για led και ρεύμα 20mA θα βγει από τον νόμο του Ohm: $R=3/0.02=150 \Omega$. Έτσι από αυτό καταλήγουμε και επιλέγουμε μια αντίσταση 220 Ω μεγαλύτερη από τα 150Ω αφενός γιατί στην αγορά κυκλοφορούν αντιστάσεις 100, 220, 330Ω αφετέρου για λόγους ασφαλείας. Στην περίπτωση μας στις αντιστάσεις για την φωτοαντίσταση και το button χρησιμοποιούμε 10kΩ για τον απλούστατο λόγο ότι και τα δυο δεν χρειάζονται ελάχιστο ρεύμα για να λειτουργήσουν.
- Όσον αφορά το τρανζίστορ μας το χρησιμοποιούμε γιατί λειτουργεί σαν ελεγχόμενος διακόπτης. Αυτό μας δίνει την δυνατότητα να χρησιμοποιήσουμε μια λάμπα που έχει παραπάνω τάση από ότι μπορεί να δώσει μια θέση στο arduino μας.
- Τέλος η σύνδεση στο vin στο arduino από την μπαταριά της λάμπας γίνεται για να έχουμε παροχή ρεύματος και χωρίς την χρήση του usb καλωδίου. ωστόσο υπάρχει και παροχή με 9volt μπαταρια με χρήση της παροχής του arduino

2.5 ΥΛΟΠΟΙΗΣΗ ΣΥΝΔΕΣΜΟΛΟΓΙΑΣ ΣΤΑΔΙΟ 1ο

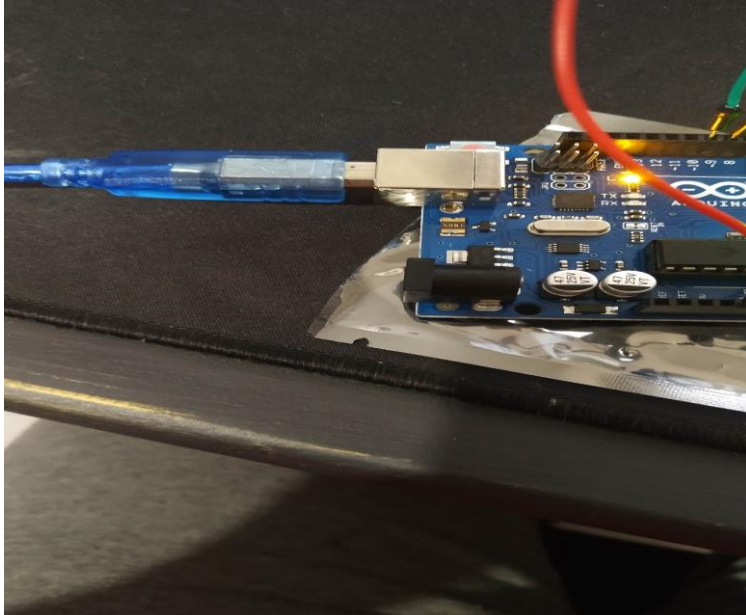


Εικόνα 19

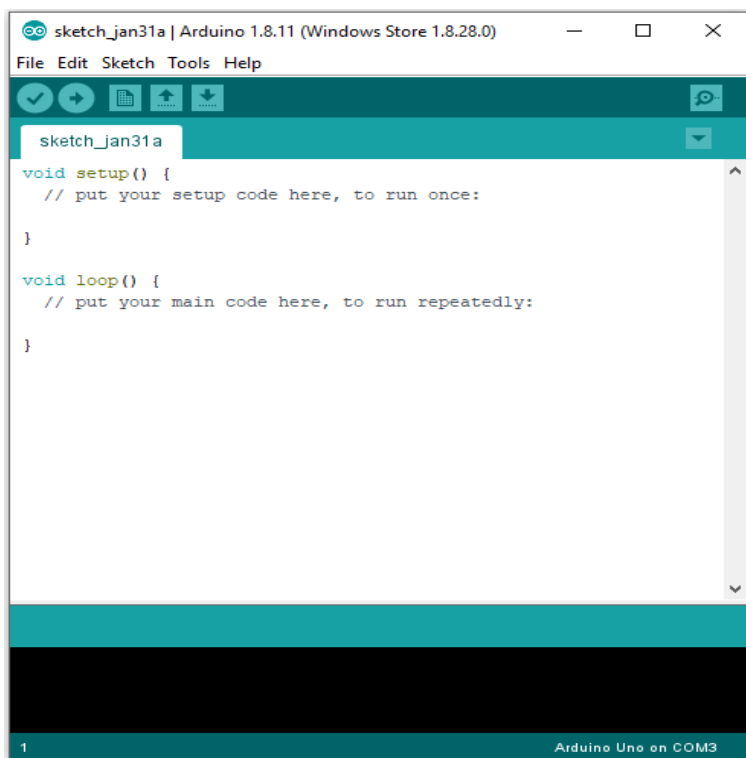
Στην παραπάνω φωτογραφία βλέπουμε την υλοποίηση του project στο στάδιο με λαμπάκι led και χρήση 220Ω αντίσταση. Η συνδεσμολογία έχει γίνει όπως έχουμε δείξει στο πλάνο υλοποίησης. Από αυτό σημείο το μόνο που μένει για την ολοκλήρωση είναι η σύνδεση του arduino στον υπολογιστή και ο προγραμματισμός του.

2.6 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ARDUINO

Συνδεουμε το arduino με usb και ανοίγουμε το περιβάλλον του arduino ide



Εικόνα 20



Εικόνα 21

2.7 ΚΩΔΙΚΑΣ ARDUINO IDE

Την υλοποίηση θα την χωρίσουμε σε στάδια:

Στάδιο 1°.

Στο στάδιο αυτό θα δηλώσουμε τις σταθερές μας

- `const int ledPin = 9;` Αυτή η εντολή αφορά την δήλωση της λάμπας μας στο pin 9 του arduino.
- `const int button = 10;` Αυτή η εντολή αφορά την δήλωση του διακόπτη/κουμπιού στο pin 10 του arduino.
- `const int potPin = A1;` Αυτή η εντολή αφορά την δήλωση του ποτενσιόμετρου στο pin A1 του arduino
- `const int photoR = A0;` Αυτή η εντολή αφορά την δήλωση της φωτοαντίστασης στο pin A0 του arduino

Στάδιο 2°.

Στο στάδιο αυτό θα δηλώσουμε τις μεταβλητές μας

- `int pot_value;` Αυτή η εντολή αφορά την δήλωση της τιμής του ποτενσιόμετρου.
- `int phR_value;` Αυτή η εντολή αφορά την δήλωση για την τιμή της φωτοαντίστασης.
- `int bt_state;` Αυτή η εντολή αφορά την δήλωση για το κουμπί/διακόπτη.
- `int i=0;` Αυτή η εντολή αφορά την ένδειξη για να βλέπουμε αν άναψε το led από τον διακόπτη

Στάδιο 3°.

Στο στάδιο αυτό μέσα στην void setup με την χρήση της συνάρτησης pin mode.Δηλώνουμε εισόδους και εξόδους στο πρόγραμμα

- `pinMode(ledPin, OUTPUT);` Αυτή η εντολή αφορά την δήλωση ότι το led μας είναι έξοδος
- `pinMode(button, INPUT);`
- `PinMode(photoR, INPUT);` Αυτή η εντολή αφορά την δήλωση ότι το κουμπί και φωτοαντίσταση είναι είσοδος .Αυτές οι εντολές είναι και οι δυο αχρείαστες γιατί ανέκαθεν οι αναλογικές θύρες του arduino είναι είσοδοι
- `Serial.begin(9600);` Αυτή η εντολή αφορά την δήλωση για να εκτυπώσουμε τις τιμές της φωτοαντίστασης στην σειριακή οθόνη του arduino ide.

Στάδιο 4^ο.

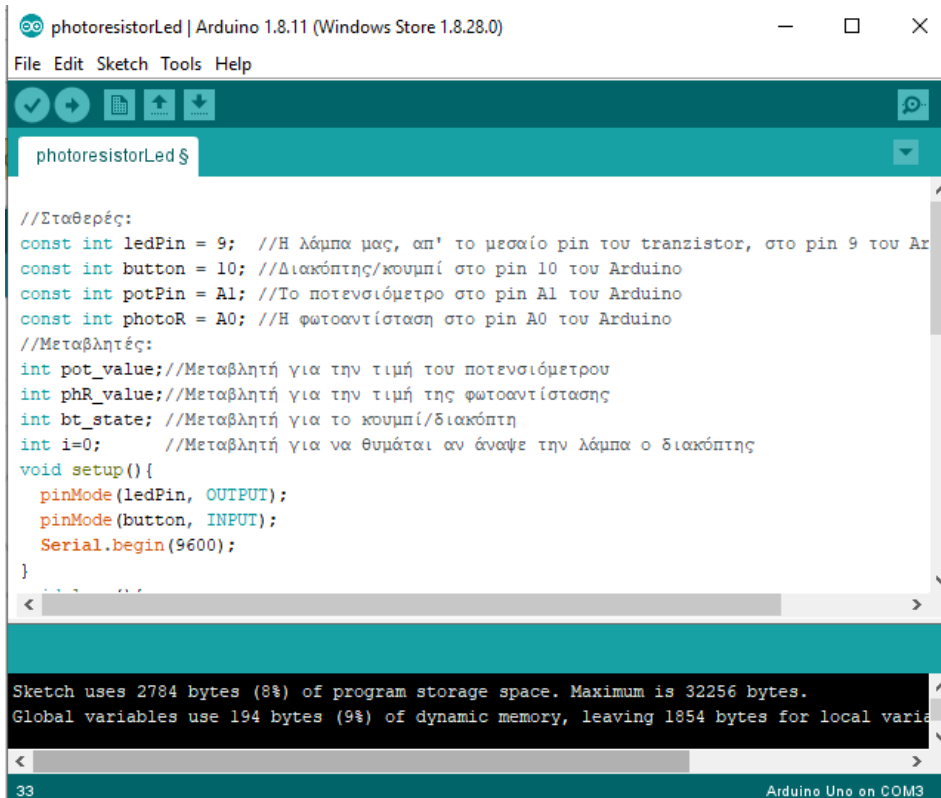
Μέσα στην void loop

- `Serial.println(phR_value);` Αυτή η εντολή μας εκτυπώνει στη σειριακή οθόνη τις τιμές της φωτοαντίστασης
- `phR_value = analogRead(photoR);` Αυτή η εντολή αποθηκεύει τις τιμές της φωτοαντίστασης στην μεταβλητή `phR_value`
- `pot_value = analogRead(potPin);` Αυτή η εντολή αποθηκεύει την τιμή του ποτενσιόμετρου στην μεταβλητή `pot_value`
- `pot_value = map(pot_value, 0, 1024, 0, 255);` // Αυτή η εντολή αντιστοιχεί τις αναλογικές τιμές του ποτενσιόμετρου 0-1024 με PWM 0-255 ;)
- `bt_state = digitalRead(button);` Αυτή η εντολή μας δείχνει με LOW ή HIGH αν πατήθηκε το κουμπί
- `if (phR_value < 50){`

`analogWrite(ledPin, 255);` Αυτή η εντολή είναι για την φωτοαντίσταση και την αυτόματη ενεργοποίηση της λάμπας σε χαμηλό φωτισμό.

- `else{`
- `if (bt_state == HIGH){`
- `if (i == 0){`
- `i=1;`
- `}`
- `else{`
- `i=0;`
- `}`
- `}` Αυτή η εντολη για τον ελεγχο του κουμπιού
- `if (i==1){`
- `analogWrite(ledPin, pot_value);` Αυτή η εντολή για να ελέγχουμε την φωτεινότητα και να ανάβει η λάμπα στην αντίστοιχη φωτεινότητα
- `}`
- `else{`
- `analogWrite(ledPin,0);` Αυτή η εντολή είναι για τον διακόπτη (αν ξαναπατιέται να σβήνει η λάμπα)
- `}`
- `}`
- `delay(200);` Τέλος η εντολή αυτή αφορά την χρονο καθυστερηση που θα έχει το arduino για να εκτελεί τις εντολές η οποία μετρείται σε ms.

3.1 ΠΡΟΓΡΑΜΜΑ ΣΤΟ ARDUINO IDE



```
photoresistorLed $
//Σταθερές:
const int ledPin = 9; //Η λάμπα μας, απ' το μεσαίο pin του tranzistor, στο pin 9 του Ar
const int button = 10; //Διακόπτης/κουμπί στο pin 10 του Arduino
const int potPin = A1; //Το ποτενσιόμετρο στο pin A1 του Arduino
const int photoR = A0; //Η φωτοαντίσταση στο pin A0 του Arduino
//Μεταβλητές:
int pot_value; //Μεταβλητή για την τιμή του ποτενσιόμετρου
int phR_value; //Μεταβλητή για την τιμή της φωτοαντίστασης
int bt_state; //Μεταβλητή για το κουμπί/διακόπτη
int i=0; //Μεταβλητή για να θυμάται αν άναψε την λάμπα ο διακόπτης
void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(button, INPUT);
  Serial.begin(9600);
}

```

Sketch uses 2784 bytes (8%) of program storage space. Maximum is 32256 bytes.
Global variables use 194 bytes (9%) of dynamic memory, leaving 1854 bytes for local variables.

33 Arduino Uno on COM3

Εικόνα 23



```
photoresistorLed $
,
void loop(){
  Serial.println(phR_value);
  phR_value = analogRead(photoR); //Αποθήκευση τιμής φωτοαντίστασης στην μεταβλητή phR_value
  pot_value = analogRead(potPin); //Αποθήκευση τιμής ποτενσιόμετρου στην μεταβλητή pot_value
  pot_value = map(pot_value, 0, 1024, 0, 255); // Αντιστοίχιση αναλογικών τιμών 0-1024 με PWM 0-255 ;)
  bt_state = digitalRead(button); //LOW ή HIGH, ανάλογα αν πατήθηκε το κουμπί
  if ( phR_value < 300 ){
    analogWrite(ledPin, 255); //Σε χαμηλό φωτισμό άναψε το φως
  }
  else{
    //Κώδικας για το κουμπί:
    if ( bt_state == HIGH){
      if ( i == 0 ){
        i=1;
      }
      else{
        i=0;
      }
    }
    //Απ έξω απ την προηγούμενη if για να ελέγχουμε την φωτεινότητα (!)
    if ( i==1){
      analogWrite(ledPin, pot_value); //Ανάβει η λάμπα στην αντίστοιχη φωτεινότητα
    }
    else{
      analogWrite(ledPin,0); //Αν ξαναπατηθεί ο διακόπτης σβήνει η λάμπα
    }
  }
  delay(200);
}

```

Sketch uses 2784 bytes (8%) of program storage space. Maximum is 32256 bytes.
Global variables use 194 bytes (9%) of dynamic memory, leaving 1854 bytes for local variables. Maximum

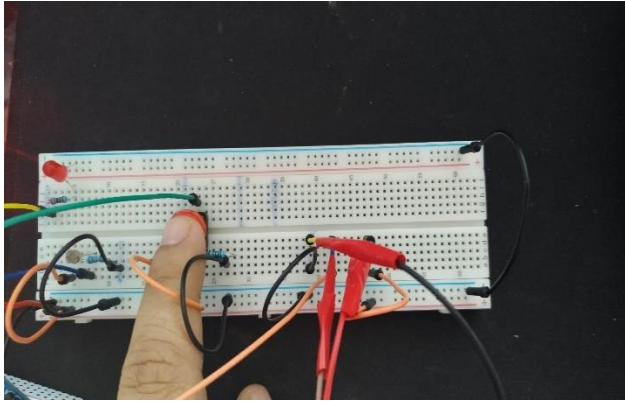
33 Arduino Uno on COM3

Εικόνα 22

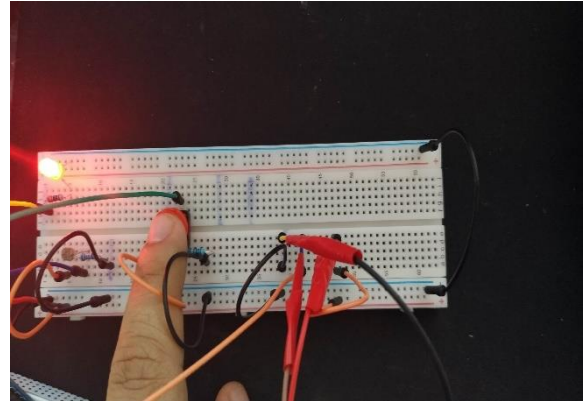
3.2 ΑΠΟΤΕΛΕΣΜΑΤΑ

Σαν αποτέλεσμα έχουμε κατασκευάσει ένα κύκλωμα που μας προσφέρει τρεις δυνατότητες:

1. Πρώτον να μπορούμε να αναβοούμε και να σβήνουμε το led μας με έναν διακόπτη

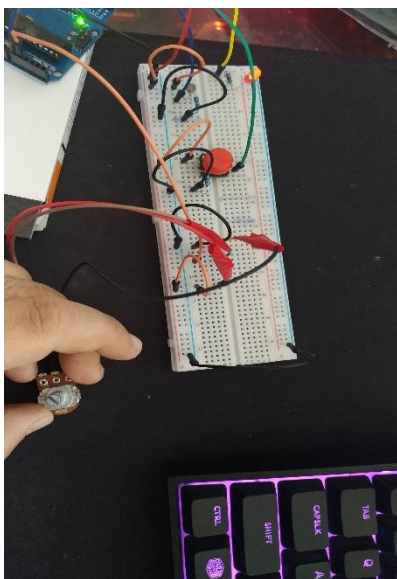


Εικόνα 24

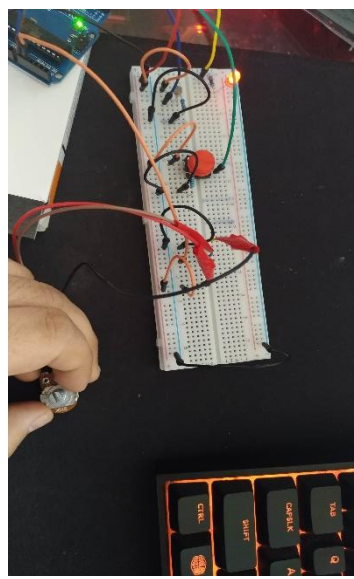


Εικόνα 25

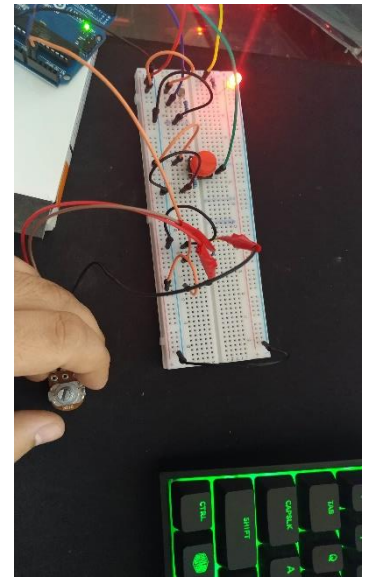
2. Να τροποποιούμε την φωτεινότητα του με το ποτενσιόμετρο.



Εικόνα 28



Εικόνα 26

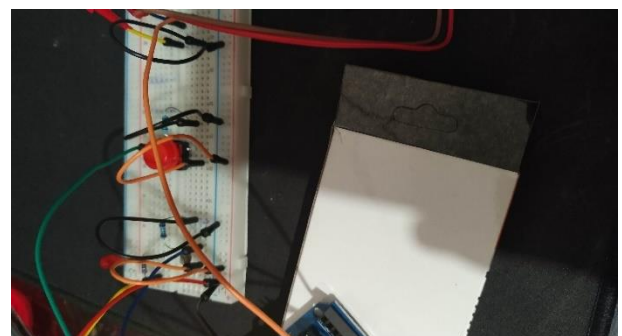


Εικόνα 27

3. Τέλος να ανάβει μόνη της σε χαμηλό φωτισμό και σε υψηλό να σβήνει



Εικόνα 29

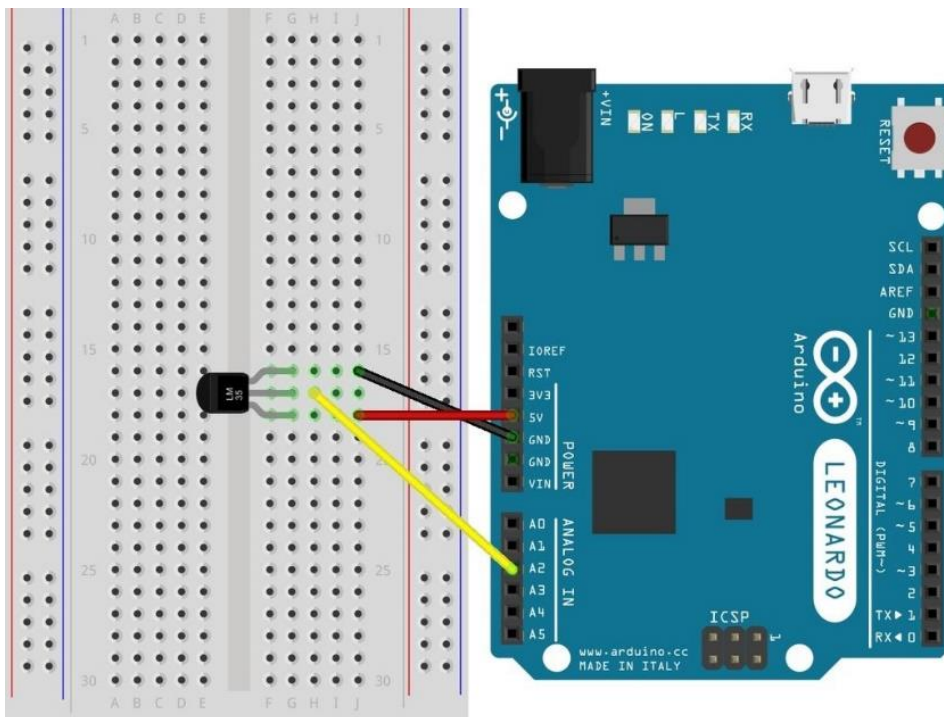


Εικόνα 30

3.3 ΕΠΙΠΛΕΟΝ ΣΤΑΔΙΑ ΥΛΟΠΟΙΗΣΗΣ

Αφού λοιπόν είδαμε την κατασκευή και στην ουσία την λειτουργία του βασικού μας project (smart lamp) θα μπορούσαμε να το εμπλουτίσουμε με παραπάνω χαρακτηριστικά. Ένα από αυτά τα χαρακτηριστικά τα οποία θα μπορούσαμε να τα ενσωματώσουμε πάνω στο arduino μας είναι ο αισθητήρας θερμοκρασίας. Με απλά λόγια ο αισθητήρας αυτός ελέγχει και μας εμφανίζει στη σειριακή οθόνη του arduino την θερμοκρασία που διαθέτει εκείνη την στιγμή.

- Το κύκλωμα μας πάνω σε breadboard.
- Τα υλικά που θα χρειαστούμε:
 1. Arduino uno
 2. LM35 temperature sensor
- Συνδεσμολογία
Ενωνουμε το μεσαίο pin του sensora στην αναλογική θύρα a2
Το δεξι στο gnd και το αριστερό στο 5v



Εικόνα 31

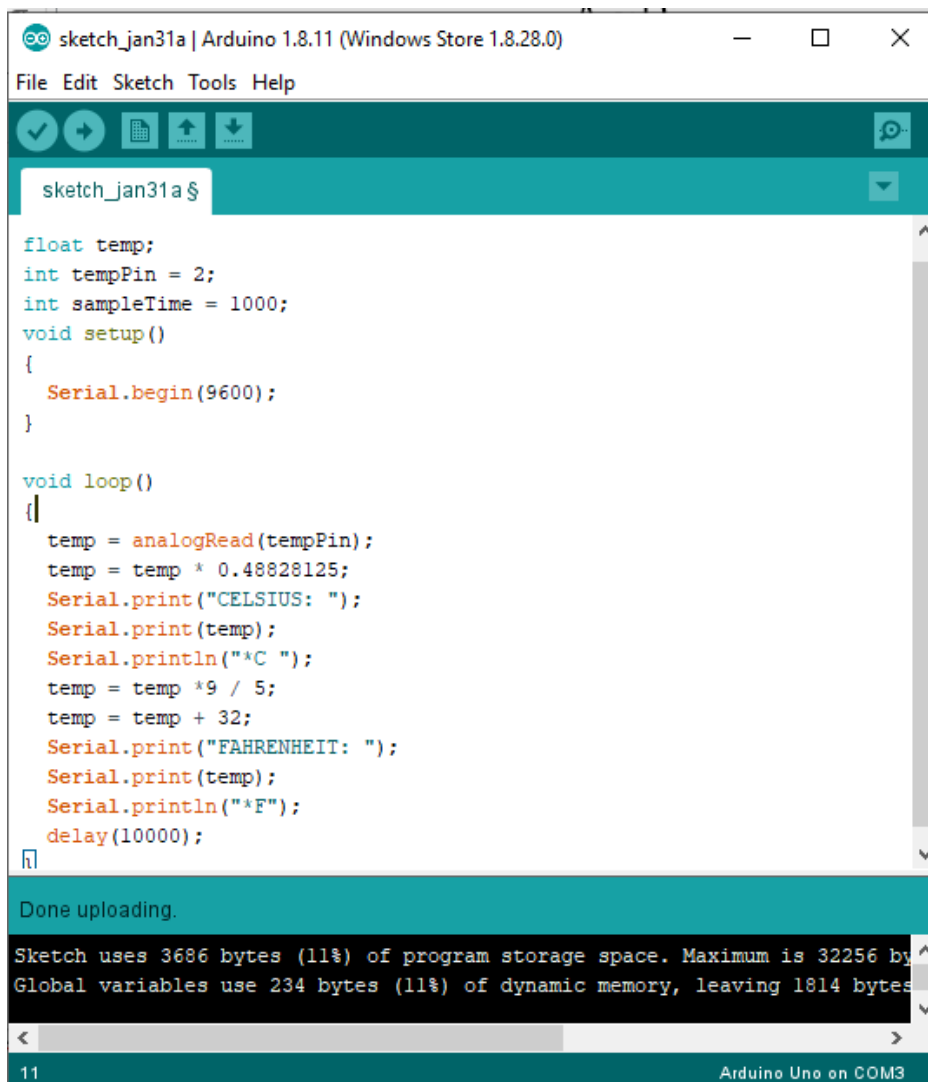
3.4 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ TEMP SENSOR

```
float temp;
int tempPin = 2;
int sampleTime = 1000;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  temp = analogRead(tempPin);
  temp = temp * 0.48828125;
  Serial.print("CELSIUS: ");
  Serial.print(temp);
  Serial.println("°C ");
  temp = temp * 9 / 5;
  temp = temp + 32;
  Serial.print("FAHRENHEIT: ");
  Serial.print(temp);
  Serial.println("°F");
  delay(10000);
}
```

3.5 ΠΕΡΙΒΑΛΛΟΝ IDE

Πιο αναλυτικά θα δούμε ότι:

1. Με την `temp = analogRead(tempPin);` καθώς και με την `temp = temp * 0.48828125;` διαβάζουμε τις τιμές του `sensora` και τις μετατρέπουμε σε βαθμούς κελσίου
2. Με την `temp = temp * 9 / 5;` Και την `temp = temp + 32;` μετατρέπουμε τις τιμές σε Φαρενάιτ
3. Τέλος βλέπουμε αποτελέσματα στην σειριακή οθόνη ανά 10 sec



```
sketch_jan31a | Arduino 1.8.11 (Windows Store 1.8.28.0)
File Edit Sketch Tools Help
sketch_jan31a $
float temp;
int tempPin = 2;
int sampleTime = 1000;
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  temp = analogRead(tempPin);
  temp = temp * 0.48828125;
  Serial.print("CELSIUS: ");
  Serial.print(temp);
  Serial.println("*C ");
  temp = temp * 9 / 5;
  temp = temp + 32;
  Serial.print("FAHRENHEIT: ");
  Serial.print(temp);
  Serial.println("*F");
  delay(10000);
}

Done uploading.
Sketch uses 3686 bytes (11%) of program storage space. Maximum is 32256 bytes.
Global variables use 234 bytes (11%) of dynamic memory, leaving 1814 bytes free.

11 Arduino Uno on COM3
```

Εικόνα 32

3.6 ΑΠΟΤΕΛΕΣΜΑΤΑ ΣΕΙΡΙΑΚΗΣ ΟΘΟΝΗΣ

Στην παρακάτω εικόνα βλέπουμε τα αποτελέσματα της σειριακής οθόνης στο arduino ide και βλέπουμε τις διακυμάνσεις που έχει η τιμή της θερμοκρασίας αρχικά με χρήση αναπτήρα κοντά στον sensora (αύξηση) και στην συνέχεια με χρήση πάγου κοντά στον sensora (μείωση).

```
COM3
Send
CELSIUS: 19.53°C
FAHRENHEIT: 67.16°F
CELSIUS: 19.53°C
FAHRENHEIT: 67.16°F
CELSIUS: 19.53°C
FAHRENHEIT: 67.16°F
CELSIUS: 20.02°C
FAHRENHEIT: 68.04°F
CELSIUS: 20.02°C
FAHRENHEIT: 68.04°F
CELSIUS: 21.48°C
FAHRENHEIT: 70.67°F
CELSIUS: 21.97°C
FAHRENHEIT: 71.55°F
CELSIUS: 25.39°C
FAHRENHEIT: 77.70°F
CELSIUS: 23.93°C
FAHRENHEIT: 75.07°F
CELSIUS: 22.46°C
FAHRENHEIT: 72.43°F
CELSIUS: 21.97°C
FAHRENHEIT: 71.55°F
CELSIUS: 21.48°C
FAHRENHEIT: 70.67°F
CELSIUS: 21.00°C
FAHRENHEIT: 69.79°F
CELSIUS: 18.55°C
FAHRENHEIT: 65.40°F
CELSIUS: 16.60°C
FAHRENHEIT: 61.88°F
CELSIUS: 15.63°C
FAHRENHEIT: 60.13°F
CELSIUS: 17.09°C
FAHRENHEIT: 62.76°F
 Autoscroll  Show timestamp
Newline 9600 baud Clear output
```

Εικόνα 33

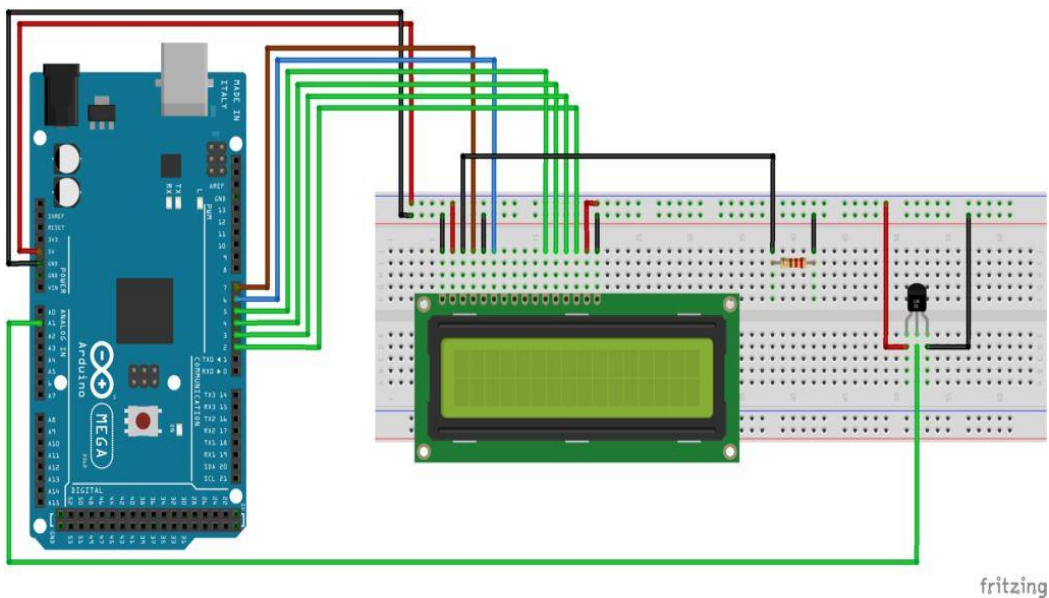
3.7 LCD SCREEN

Είναι οθόνη η οποία μπορεί να συνδεθεί στο arduino και να μας προφέρει πληροφορίες για ότι ακριβός θέλουμε σε ψηφιακή μορφή .Εμείς θα την συνδυάσουμε με τον temp sensor για να περνούμε στην οθόνη τις θερμοκρασίες που διαβάζει.

Τα υλικά που θα χρειαστούμε:

- Arduino uno
- LM35 temperature sensor
- LCD 2x16 ή 4x20
- Resistor (1 K ohm)

Το κύκλωμα μας σε breadboard:

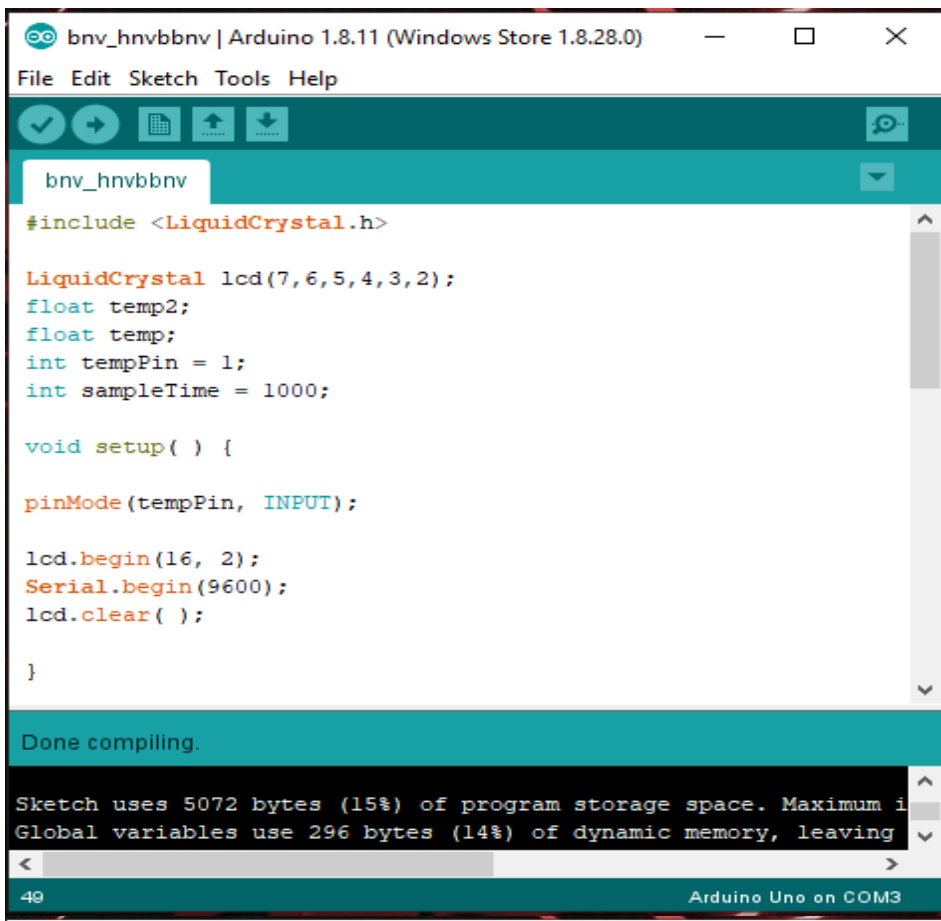


Εικόνα 34



Εικόνα 35

3.8 ΠΕΡΙΒΑΛΛΟΝ ΟΘΟΝΗΣ ΣΕ IDE



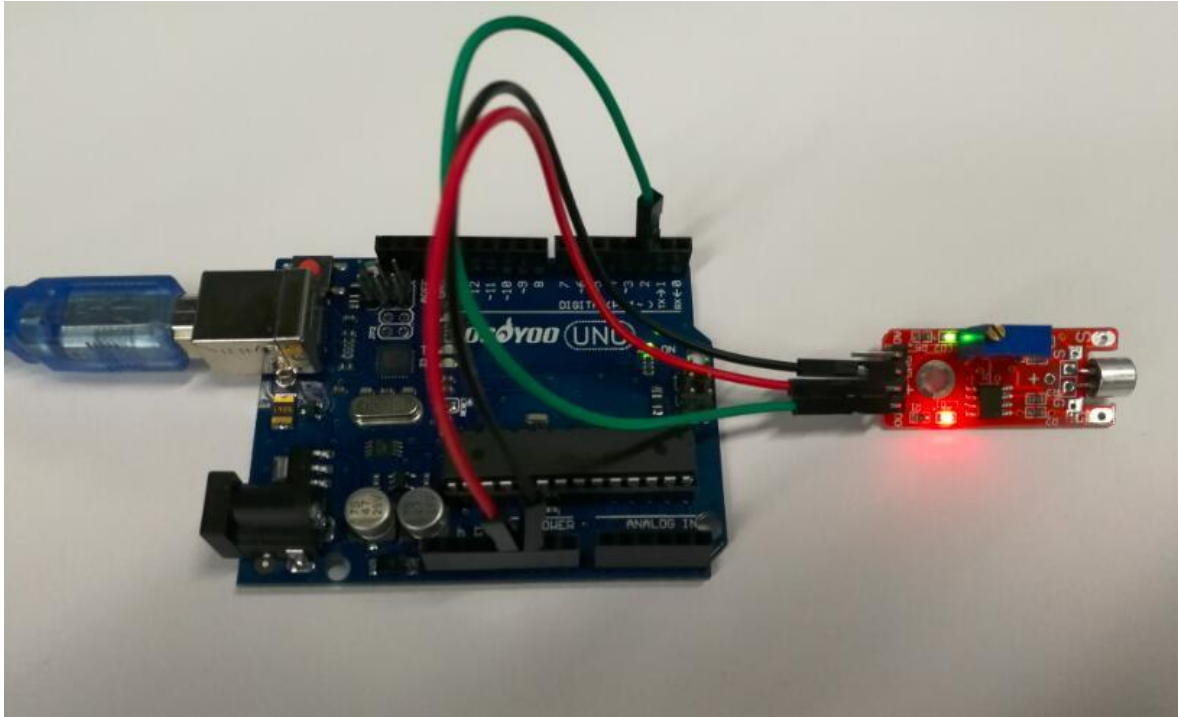
Εικόνα 36



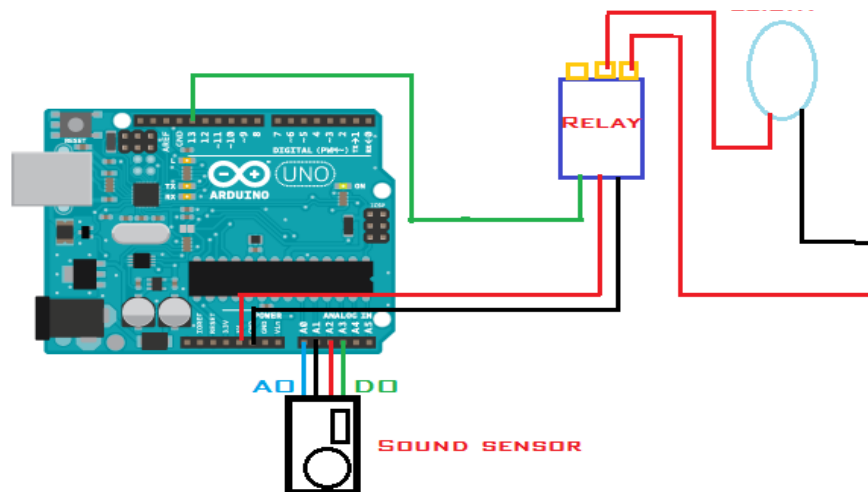
Εικόνα 37

3.9 ΣΥΜΠΛΗΡΩΜΑΤΙΚΟ PROJECT

Το παρακάτω project είναι παρόμοιο με αυτό που έχουμε δει πιο πάνω άπλα αντί για την χρήση μιας φωτοαντίστασης μπορούμε να χρησιμοποιήσουμε ένα σενσορα ήχου με τον οποίο κάθε φορά που θα κτυπάμε παλαμάκι να ανοίγει το led μας και κάθε φορά που θα ξανά κτυπάμε να σβήνει.



Εικόνα 38



Εικόνα 39

3.10 ΚΩΔΙΚΑΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

```
int micPin = A0; // pin that the mic is attached to
int gndPin = A1;
int powerPin = A2;
int micValue1 = 0;
int micValue2 = 0; // the Microphone value
int led1 = 13;
boolean lightOn = false;
void setup() {
  pinMode(led1, OUTPUT);
  pinMode(powerPin, OUTPUT);
  pinMode(gndPin, OUTPUT);
  pinMode(micPin, INPUT);
  digitalWrite(gndPin,LOW);
  delay(500);
  digitalWrite(powerPin,HIGH);
  Serial.begin(9600); //for test the input value initialize serial
}
void loop() {
  micValue1 = analogRead(micPin); // read pin value
  Serial.println(micValue1);
  delay(1);
  micValue2 = analogRead(micPin);
  Serial.println(micValue2);
  if (micValue1-micValue2 > 2||micValue2-micValue1 > 2){
    lightOn = !lightOn;
    delay(100);
    digitalWrite(led1, lightOn);
  }
}
```

3.11 ΣΥΜΠΕΡΑΣΜΑΤΑ

Η πτυχιακή εργασία που παρουσιάστηκε στις παραπάνω επιμέρους παραγράφους, είναι η υλοποίηση και η κατασκευή μιας smart lamp με διάφορες δυνατότητες μέσω του Arduino uno. Η χρήση των μικροελεγκτών και συγκεκριμένα ο ArduinoUno, μας βοήθησε να φτιάξουμε την δική μας αυτόνομη ηλεκτρονική συσκευή και να την προγραμματίσουμε να κάνει ότι ακριβώς αποσκοπούσαμε . Το Arduino δεν απευθύνετε μόνο σε προγραμματιστές και ηλεκτρονικούς αλλά και σε όποιον ενδιαφέρεται να έχει σχέση γενικότερα με την τεχνολογία και να θέλει να δει την αλληλεπίδραση της μέσα στην καθημερινότητα του . Όπως αναφέραμε και παραπάνω δεν χρειάζεται να έχεις γνώσεις πάνω στον προγραμματισμό, οι βασικές έννοιες αρκούν για οπουδήποτε υλοποίηση ενός απλού project. Όπως προαναφέραμε το κόστος αγοράς είναι χαμηλό. Το συνολικό κόστος αυτού του project δεν ξεπέρασε τα 120 ευρώ. Το χαμηλό κόστος μας βοήθησε να δημιουργήσουμε κάτι εύκολο και χειροπιαστό όπως είναι η κατασκευή μιας smart lamp και μας έδειξε τις ευκολίες και τις δυνατότητες που προσφέρει η τεχνολογία στις μέρες μας

3.12 ΜΕΛΟΝΤΙΚΕΣ ΠΡΟΤΑΣΕΙΣ

Όπως είδαμε και πιο πάνω δεν υπάρχουν όρια στο τι θες να υλοποιήσεις. τα όρια τα θέτεις εσύ με βάση την όρεξη καθώς και τις βλέψεις που έχεις για ένα project. Κλείνοντας θέλουμε να αναφέρουμε ότι αυτό είναι ένα από τα project υλοποίησης πάνω σε arduino uno. Τα σχέδια μας δεν τελειώνουν εδώ μελλοντικό και υλοποιήσιμο project είναι η κατασκευή smart κλειδαριάς με fingerprint sensor με την χρήση του arduino uno.

Βιβλιογραφία

1. Βιβλίο:Ανάπτυξη εφαρμογών με το Arduino του Παπάζουγλου Π. Και του Λιωνή ΣΠ.
2. Βιβλίο:Ενσωματωμένα Συστήματα,οι μικροελεγκτές AVR και ARDUINO του Πογαρίδη Δ
3. Book: Arduino Workshop: A Hands-On Introduction with 65 Projects from John Boxall by John Boxall
4. Book :Arduino Programming in 24 Hours, Sams Teach Yourself by Blum Richard
5. Γνώσεις πάνω σε arduino uno <http://www.ardumotive.com/>
6. Βιντεάκια συνδεσμολογίας
<https://www.youtube.com/watch?v=6uzGJilzHEM>
7. Τι είναι το arduino καθώς και γενικές πληροφορίες για αυτό
<http://learning.grobotronics.com/el/getting-started/arduino-uno/>
8. Γενικές πληροφορίες
<https://www.google.com/search?client=firefox-b-d&q=arduino+uno>
9. Ιστορία του arduino uno
https://en.wikipedia.org/wiki/Arduino_Uno
- 10.Κανάλι στο youtube με χρήσιμες πληροφορίες
<https://www.youtube.com/watch?v=6V2Br6J-pBc>
- 11.Για project με lcd <https://www.dwrean.net/2015/01/28-lcd-arduino.html>
- 12.Για project με lcd και θερμόμετρο
<https://how2electronics.com/digital-thermometer-arduino-lm35-temperature-sensor/>
- 13.Γενικές γνώσεις και μελλοντικά project
<https://how2electronics.com/digital-thermometer-arduino-lm35-temperature-sensor/>
- 14.Clap sensor <https://www.youtube.com/watch?v=hzUFnP3Xt7c>
- 15.Clap sensor <https://osoyoo.com/2017/07/26/arduino-lesson-sound-detection-sensor/>