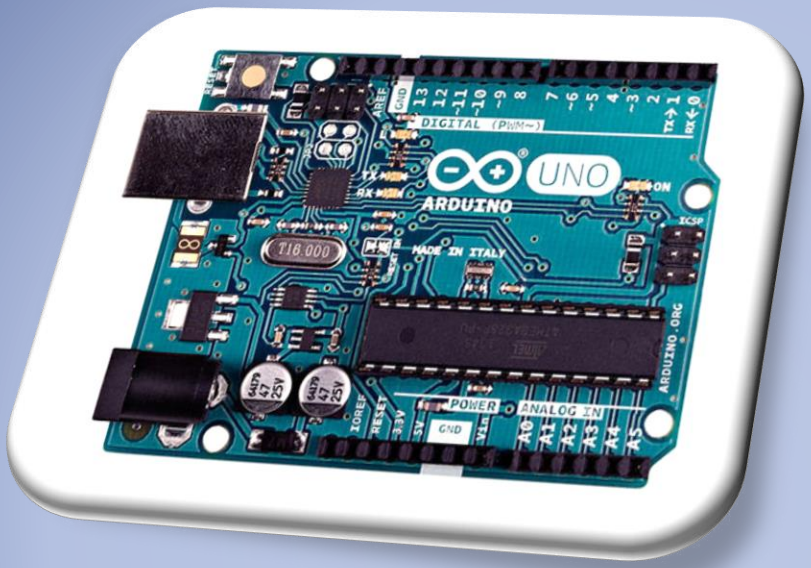
ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ (Τ.Ε.Ι.) ΔΥΤΙΚΗΣ ΕΛΛΑΔΟΣ
ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ
ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ (ΠΡΩΗΝ Ε.Π.Δ.Ο)



"ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΩΝ ΜΕ ΧΡΗΣΗ ΜΙΚΡΟΕΛΕΓΚΤΩΝ ARDUINO"

ΚΩΗΣ ΔΗΜΗΤΡΙΟΣ
Α.Μ. 12867

Επιβλέπων Καθηγητής: Παπαδόπουλος Δημήτριος

Πάτρα 2017

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον κ. Δημήτριο Παπαδόπουλο, επιβλέποντα καθηγητή, διότι με βοήθησε με την καθοδήγηση του να υλοποιήσω την ιδέα μου και να δημιουργήσω την πτυχιακή μου εργασία πετυχαίνοντας τον στόχο μου.

Ακόμη θα ήθελα να ευχαριστήσω την οικογένεια μου γα την στήριξη που μου παρείχαν για όλα αυτά τα χρόνια ώστε να ολοκληρώσω τις σπουδές μου και να έχω όλα όσα έχω μέχρι στιγμής στη ζωή μου.

Και τέλος όλους όσους στάθηκαν δίπλα μου.

ΠΕΡΙΛΗΨΗ

Στις μέρες μας έχει αρχίσει και αναπτύσσεται περισσότερο από πότε μέχρι στιγμής η χρήση της περιρρέουσας και τεχνητής νοημοσύνης όπου ο άνθρωπος μπορεί και αλληλεπιδρά στο περιβάλλον του με μηχανές προσπαθώντας να διευκολύνει την καθημερινότητα του Έξυπνο σπίτι, λάμπα, πρίζα).
Με τη χρήση ενός μικροελεγκτή (arduino) μπορούμε να δημιουργήσουμε κυκλώματα όπου προγραμματίζοντας τα να μας βοηθήσουν στην ιδέα της περιρρέουσας νοημοσύνης.
Σκοπός της συγκεκριμένης πτυχιακής εργασίας είναι η δημιουργία μιας μακέτας έξυπνου θερμοκηπίου όπου θα λειτουργεί αυτόνομα ή με την μικρή συμμετοχή του Χρήστη. Το θερμοκήπιο θα διαθέτει δυο αισθητήρες, έναν θερμοκρασίας και υγρασίας περιβάλλοντος χώρου, υγρασίας εδάφους και ένα μοτέρ για αυτόματο πότισμα. Οι τιμές που δέχεται από τους αισθητήρες θα εμφανίζονται σε μια σειριακή οθόνη και ότι πρόβλημα προκύπτει θα ενεργοποιεί ένα led ώστε να δίνει την δυνατότητα στον χρήστη να γνωρίζει ότι υπάρχει πρόβλημα το οποίο θα έχει φροντίσει να επιλυθεί αυτόματα με το κατάλληλο κομμάτι κώδικα .Η παρούσα εργασία θα μπορούσε να χρησιμοποιηθεί και ως βάση για τη δημιουργία ενός πραγματικού με φυσικές διαστάσεις θερμοκηπίου, άλλωστε αυτός είναι άλλος ένας σκοπός της.

ABSTRACT

In our days it seems to begin and develop more than ever the use of artificial and perceiving intelligence where the human can interacts on his environment with machines, trying to ease his daily routine (Intelligent home, lamp, outlet).
By using a microcontroller (arduino) we can create circuits where we program them to help us with the idea of perceiving intelligence.
The aim of this graduate thesis is to create a smart greenhouse model where it will operate autonomously or with the small involvement of the User. The greenhouse will have two sensors, a room temperature and humidity, ground moisture and an air pump sensor for the automatic watering. Sensor readings will be displayed on a serial screen and if a problem arises, it will either activate a led to give the possibility for the user to know that there is a problem which will ensure that it is automatically selected with the appropriate piece of code. This work could also be used as a basis for creating a real greenhouse with natural dimensions; after all, this is another purpose of this thesis.

ΠΕΡΙΕΧΟΜΕΝΑ

Κεφάλαιο 1 - Η Ιστορία του Arduino

Το arduino είναι μια ηλεκτρονική πλατφόρμα ανοικτού κώδικα και σχεδιασμού, που βασίζεται σε ευέλικτο και εύκολο για τον χρηστή υλικό και λογισμικό.
Το 2005, στο Ivrea Interaction Design Institute του Τωρινού " γεννήθηκε" το arduino από τους Massimo Banzi & David Cuartielles οι οποίοι ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρέα - στο Ivrea Interaction Design Institute-η οποία είναι κωμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας, στην ίδια περιοχή στην οποία στεγαζόταν η εταιρία υπολογιστών Olivetti.

Ivrea Interaction Design Institute

Σήμερα έχει κατακτήσει όλο τον κόσμο με τις δεκάδες πλακέτες ανάπτυξης που έχουν δημιουργηθεί.
Στόχος ήταν η δημιουργία ενός εύκολου εργαλείου για τη γρήγορη δημιουργία πρωτοτύπων, που απευθύνεται σε μαθητές χωρίς υπόβαθρο στα ηλεκτρονικά και τον προγραμματισμό. Από τη στιγμή μου έγινε γνωστό στο εύρη κοινό , η πλακέτα Arduino άρχισε να αλλάζει για να προσαρμοστεί στις νέες ανάγκες και προκλήσεις, διαφοροποιώντας την προσφορά της από απλές πλακέτες 8-bit σε προϊόντα για εφαρμογές IoT, 3D εκτύπωση, και τα ενσωματωμένα περιβάλλοντα. Όλες οι πλακέτες Arduino είναι εντελώς open-source, δίνοντας έτσι το πάνω χέρι στο χρήστη.
Ο λόγος που αυτή η ιδέα έγινε παγκόσμια αποδεκτή είναι γιατί οι δημιουργοί του Arduino κατάφεραν να κάνουν τον άνθρωπο να "μιλάει" με την μηχανή με πολύ απλό τρόπο. Δεδομένου ότι πρόκειται για ανοικτού κώδικα πλατφόρμα, προγραμματιστές απ' όλο τον κόσμο έχουν φτιάξει βιβλιοθήκες-οδηγούς για πολλά εξαρτήματα, μέσω των οποίων ο χρήστης απλά συνδέει το εξάρτημα με την πλακέτα Arduino που έχει και με απλές εντολές μπορεί να επιτύχει την λειτουργία που επιθυμεί. Το arduino βασίστηκε πάνω στη λογική του wiring ενός ήδη μιας ήδη υπάρχουσας πλατφόρμας ανάπτυξης ανοιχτού κώδικα, την οποία δημιούργησε ο Hernando Barragan ως τη διατριβή του για το Interaction Design Institute Ivrea. Σκοπός του ήταν να δημιουργήσει μια ηλεκτρονική έκδοση επεξεργασίας που θα χρησιμοποιούσε ένα περιβάλλον προγραμματισμού. Συμπεραίνοντας έτσι ότι το arduino δεν θα υπήρχε χωρίς το wiring και ούτε το wiring δίχως το arduino καθώς μιλάμε για δύο αλληλένδετες ιδέες.

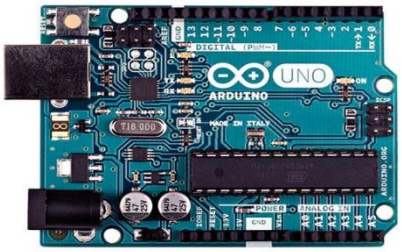
1.1 Η εξέλιξη του Arduino

Σεπτέμβριος 2006	Ανακοινώθηκε το Arduino Mini
Οκτώβριος 2008	Ανακοινώθηκε το Arduino Duemilanove. Αρχικά βασίστηκε στο Atmel Atmega168, αλλά μετά στάλθηκε με το ATmega328
Μάρτιος 2009	Ανακοινώθηκε το Arduino Mega. Είναι βασισμένο στο Atmel ATmega1280
Μάιος 2011	Πάνω από 300.000 Arduino ήταν σε χρήση σε όλο τον κόσμο
Ιούλιος 2012	Ανακοινώθηκε το Arduino Leonardo. Είναι βασισμένο στο Atmel ATmega32u4
Οκτώβριος 2012	Ανακοινώθηκε το Arduino Due. Είναι βασισμένο στο Atmel SAM3X8E, που είχε πυρήνα ARM Cortex-M3
Νοέμβριος 2012	Ανακοινώθηκε το Arduino Micro. Είναι βασισμένο στο Atmel ATmega32u4
Μάιος 2013	Ανακοινώθηκε το Arduino Robot. Είναι βασισμένο στο Atmel ATmega32u4 και ήταν το πρώτο επίσημο Arduino με ρόδες
Μάιος 2013	Ανακοινώθηκε το Arduino Yun. Είναι βασισμένο στο ATmega32u4 και στο Atheros AR9331 και ήταν το πρώτο προϊόν wifi που συνδύαζε το Arduino με το Linux.

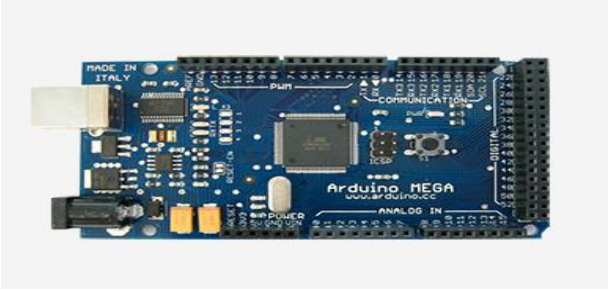
Arduino Symbol

1.2 Είδη μικροελεγκτών

Μέχρι στιγμής υπάρχουν 200 αποκλειστικοί αντιπρόσωποι προϊόντων arduino σε όλο τον κόσμο, με το 80% αυτών που αγοράζουν το arduino να είναι από την Αμερική και την Ευρώπη. Το σημαντικό είναι ότι το arduino αναπτύσσετε και γίνετε όλο και πιο γνωστό και σε άλλες αγορές όπως στη Κίνα, Ινδία και στη Νότια Αμερική. Με το πέρασμα των χρόνων δημιουργούνται όλο και περισσότερα σχέδια arduino με το αρχικό σχέδιο να ονομάζεται Arduino UNO. Μερικά από αυτά τα σχέδια είναι :

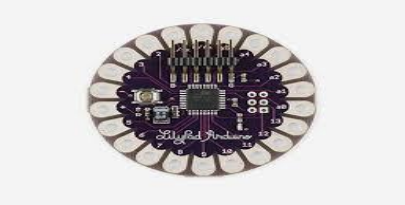


Arduino Uno



Arduino Mega

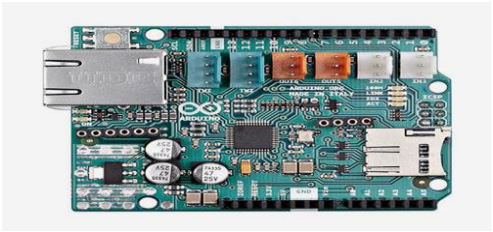
Το Arduino Mega είναι ένας μικροελεγκτής που βασίζεται στο AT mega1280. Διαθέτει 54 ψηφιακούς ακροδέκτες εισόδου / εξόδου (από τους οποίους 14 μπορούν να χρησιμοποιηθούν ως έξοδοι PWM), 16 αναλογικές εισόδους, 4 UART (σειριακές θύρες υλικού), ταλαντωτή κρυστάλλου 16 MHz, σύνδεση USB, υποδοχή τροφοδοσίας, Και ένα κουμπί για επαναφορά. Περιέχει όλα όσα χρειάζονται για να υποστηρίξουν τον μικροελεγκτή. Μπορεί να συνδεθεί με έναν υπολογιστή με καλώδιο USB ή με προσαρμογέα AC ή DC ή μπαταρία. Το Mega είναι συμβατό με τις περισσότερες ασπίδες που έχουν σχεδιαστεί για το Arduino Duemilanove ή το Diecimila.



LilyPad Arduino

Βασίζεται στην AT mega168V (έκδοση χαμηλής ισχύος του AT mega168) ή AT mega328V. Το LilyPad Arduino σχεδιάστηκε και αναπτύχθηκε από τους Leah Benchley και Spark Fun Electronics.

Microcontroller	ATmega168 or ATmega328V
Operating Voltage	2.7-5.5 V
Input Voltage	2.7-5.5 V
Digital I/O Pins	14
PWM Channels	6
Analog Input Channels	6
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (of which 2 KB used by bootloader)
SRAM	1 KB
EEPROM	512 bytes
Clock Speed	8 MHz



Arduino Ethernet

Το Arduino Ethernet είναι ένας μικροελεγκτής που βασίζεται στο AT mega328. Έχει 14 ψηφιακές ακίδες εισόδου / εξόδου, 6 αναλογικές εισόδους, ταλαντωτή κρυστάλλου 16 MHz, σύνδεση RJ45, υποδοχή τροφοδοσίας, κεφαλίδα ICSP και κουμπί επαναφοράς.

Σημείωση: Οι ακίδες 10, 11, 12 και 13 προορίζονται για διασύνδεση με τη μονάδα Ethernet και δεν πρέπει να χρησιμοποιηθούν διαφορετικά. Αυτό μειώνει τον αριθμό των διαθέσιμων ακροδεκτών σε 9, με 4 διαθέσιμες ως εξόδους PWM. Μια προαιρετική μονάδα Power over Ethernet μπορεί να προστεθεί και στην πλακέτα.

Το Ethernet διαφέρει από τις άλλες πλακέτες στο ότι δεν διαθέτει ενσωματωμένο τσιπ οδήγησης USB-to-serial, αλλά διαθέτει διασύνδεση Wiznet Ethernet. Αυτή είναι η ίδια διεπαφή που βρίσκεται στην ασπίδα Ethernet. Ένα ενσωματωμένο πρόγραμμα ανάγνωσης καρτών microSD, το οποίο μπορεί να χρησιμοποιηθεί για την αποθήκευση αρχείων για την υπηρεσία μέσω του δικτύου, είναι προσβάσιμο μέσω της βιβλιοθήκης SD. Ο ακροδέκτης 10 προορίζεται για τη διασύνδεση Wiznet, ενώ η SS για την κάρτα SD βρίσκεται στον Pin 4. Η κεφαλίδα σειριακού προγραμματισμού 6 ακίδων είναι συμβατή με τον προσαρμογέα USB Serial καθώς και με τα καλώδια USB FTDI ή με το βασικό USB στυλ Sparkfun και Adafruit FTDI -ο-σειριακόό πίνακα ξεμπλοκαρίσματος. Διαθέτει υποστήριξη για την αυτόματη επαναφορά, επιτρέποντας τη μεταφόρτωση σκίτσων χωρίς να πιέσετε το κουμπί επαναφοράς στη σανίδα. Όταν συνδέεται σε προσαρμογέα USB σε Serial, το Arduino Ethernet τροφοδοτείται από τον προσαρμογέα.

Τα τελευταία χρόνια το arduino έχει κερδίσει δημοσιότητα καθώς συνεργάζεται με την Google. Η Google έβγαλε στην αγορά το Arduino ADK (Accessory Development Kit) το οποίο βασίζεται στο arduino. Μέσω αυτής την εφαρμογής ένας χρήστης μπορεί να δημιουργήσει μια εφαρμογή android που θα χρησιμοποιεί την camera κινητού τηλεφώνου, αισθητήρες κίνησης, εικόνες αφής καθώς και συνδεσιμότητα στο διαδικτυακού όλα μπορούμε να καταλάβουμε ότι το arduino δημιουργεί ένα νέο και πιο οικονομικό είδος προγραμματισμού. Το arduino φαίνεται και ήρθε για να μείνει και γίνεται κάθε μέρα που περνάει όλο και πιο δημοφιλές.

Arduino developer team - David Cuartielles, Gianluca Martino, Tom Igoe, David Mellis, and Massimo Banzi. Photo Courtesy - Randi Klett/IEEE Spectrum

Κεφάλαιο 2 – Τα βασικά του arduino

Το arduino είναι μια ηλεκτρονική πλατφόρμα ανοικτού κώδικα και σχεδιασμού, που βασίζεται σε ευέλικτο και εύκολο για τον χρηστή υλικό και λογισμικό. Το Arduino γεννήθηκε στο Ivrea Interaction Design Institute από τους Massimo Bandi & David Cuartielles ως ένα εύκολο εργαλείο για τη γρήγορη δημιουργία πρωτοτύπων, που απευθύνεται σε μαθητές χωρίς υπόβαθρο στα ηλεκτρονικά και προγραμματισμό. Από τη στιγμή που έφτασε μια ευρύτερη κοινότητα, η πλακέτα Arduino άρχισε να αλλάζει για να προσαρμοστεί στις νέες ανάγκες και προκλήσεις, διαφοροποιώντας την προσφορά της από απλές πλακέτες 8-bit σε προϊόντα για εφαρμογές IoT, 3D εκτύπωση, και τα ενσωματωμένα περιβάλλοντα. Όλες οι πλακέτες Arduino είναι εντελώς ανοικτού κώδικα, δίνοντας έτσι το πάνω χέρι στο χρήστη.
Τεχνικά, ένα arduino είναι ένα κύκλωμα το οποίο διαθέτει έναν αριθμό πυλών εισόδου/εξόδου χρησιμοποιώντας έναν μικροελεγκτή. Αυτές οι πύλες μπορούν να χρησιμοποιηθούν γράφοντας έναν κώδικα στο κατάλληλο περιβάλλον προγραμματισμού ειδικά διαμορφωμένο για arduino βασισμένο στις γλώσσες C/C++, το ARDUINO IDE.

Με τη χρήση του arduino μπορούμε να φτιάξουμε κάτι απλό όπως ένα λαμπάκι να ανάβει ή κάτι πιο πολύπλοκο όπως να αλληλεπιδρά με ένα κινητό τηλέφωνο. Με λίγα λόγια ένας μικροελεγκτής μπορεί να ικανοποιήσει άτομα που απλά θέλουν να δοκιμάσουν εύκολα κυκλώματα για χόμπι ή διασκέδασή αλλά και άτομα που θέλουν να ασχοληθούν και να υλοποιήσουν κυκλώματα με περισσότερες απαιτήσεις.

Για να υλοποιήσουμε κυκλώματα που θέλουμε με ένα arduino από το πιο απλό μέχρι το πιο απαιτητικό θα χρειαστούμε και τον κατάλληλο εξοπλισμό/εργαλεία τα οποία μπορούμε να βρούμε στην αγορά. Μπορούμε να βρούμε το γνήσιο με την ονομασία Arduino που προέρχεται από τους δημιουργούς και επίσημους κατασκευαστές του στην Ιταλία στο επίσημο site: https://www.arduino.cc/ καθώς και ότι άλλο έχει σχέση με arduino, ενώ μπορούμε να βρούμε και άλλες πλακέτες παρόμοιες με αυτή του Arduino για υλοποίηση προγραμμάτων και κυκλωμάτων. Το μόνο που έχει απαιτηθεί από τους δημιουργούς του Arduino είναι ότι το όνομα αυτό θα το χρησιμοποιούν μόνο εκείνοι με αποτέλεσμα οι άλλες εκδόσεις μικροελεγκτών που κυκλοφορούν στην αγορά να χρησιμοποιούν άλλα ονόματα τα οποία συνήθως καταλήγουν σε ino, όπως το Funduino.

2.1 Βασικά εξαρτήματα για το arduino

Τα βασικότερα εξαρτήματα για να ξεκινήσει κάποιος με την ενασχόληση του με έναν μικροελεγκτή θα πρέπει να είναι η βασική πλακέτα arduino, ένα καλώδιο usb για σύνδεση της πλακέτας με το η/υ, ένα breadboard, μερικά καλώδια, λαμπάκια, αντιστάσεις και γενικά όλα αυτά που χρειάζονται για την δημιουργία του κυκλώματος που έχει αποφασίσει ο αγοραστής να δημιουργήσει. Όλα αυτά τα εργαλεία μπορούν να βρεθούν και σε ένα kit αρχαρίων το οποίο περιλαμβάνει όλα τα βασικά για να ξεκινήσει κάποιος την ενασχόληση του με το arduino.

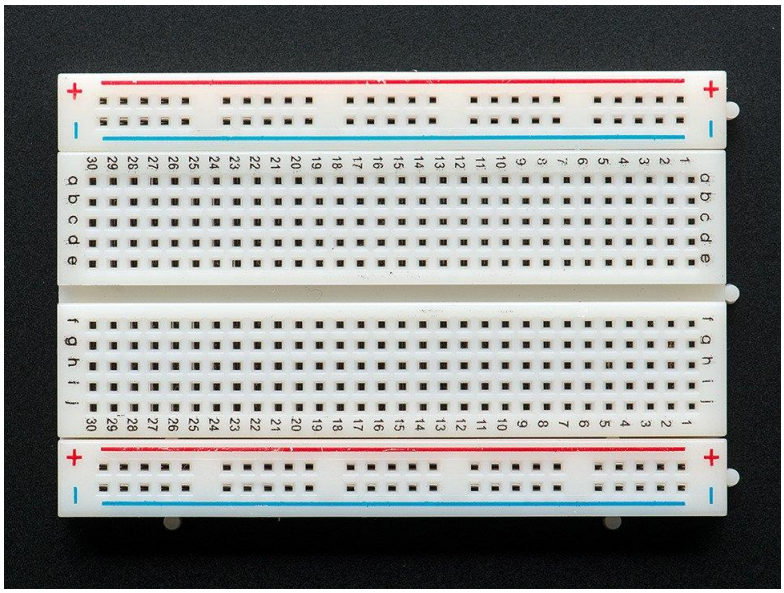
Πλακέτα arduino: είναι το μέρος το οποίο δέχεται και υλοποιεί στο κύκλωμα το πρόγραμμα που έχουμε δημιουργήσει.

Arduino Uno

- Διαθέτει
  14 ψηφιακές θύρες εισόδου/εξόδου (από το 0...13) με χαρακτηρισμό LOW&HIGHκαι
- 6 αναλογικές εισόδους ονομάζοντάς τες με το A και έναν αριθμό από το 0..5 αναλόγως τη θύρα (π.χ. A0...A5)

Όλες σε διάστημα 0-5V.

Breadboard: χρησιμοποιείται προκειμένου να μπορέσουμε να δημιουργήσουμε ένα κύκλωμα το οποίο θα συνδεθεί αφού ολοκληρωθεί με το arduino. Επίσης μπορεί να βραχυκυκλώνει μεταξύ τους καλώδια ώστε να υπάρχουν ελεύθερες είσοδοι και έξοδοι.



Breadboard

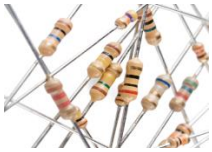
Οι οριζόντιες γραμμές + και – σε κάθε μεριά είναι βραχυκυκλωμένες μεταξύ τους, ενώ στις στήλες είναι βραχυκυκλωμένες οι πέντε κάθετες υποδοχές μεταξύ τους (συμβολισμένες με γράμματα abcde-fghij).

Μερικά από τα βασικά εξαρτήματα για τη δημιουργία κυκλωμάτων με arduino είναι τα εξής:

I. Leds, λαμπάκια τα οποία δουλεύουν μόνο αν συνδεθούν στην κατάλληλη φορά ρεύματος (πολικότητα). Διαθέτουν 2 πόδια από τα οποία το πιο μακρύ συνδέεται στη θετική κατεύθυνση και το άλλο στην αρνητική φορά

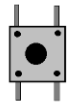
Leds

II. Αντιστάσεις, ρυθμίζουν την τάση του ρεύματος που θα ρέει στο κύκλωμα μας. Έχουν ως μονάδα μέτρησης τα ohm



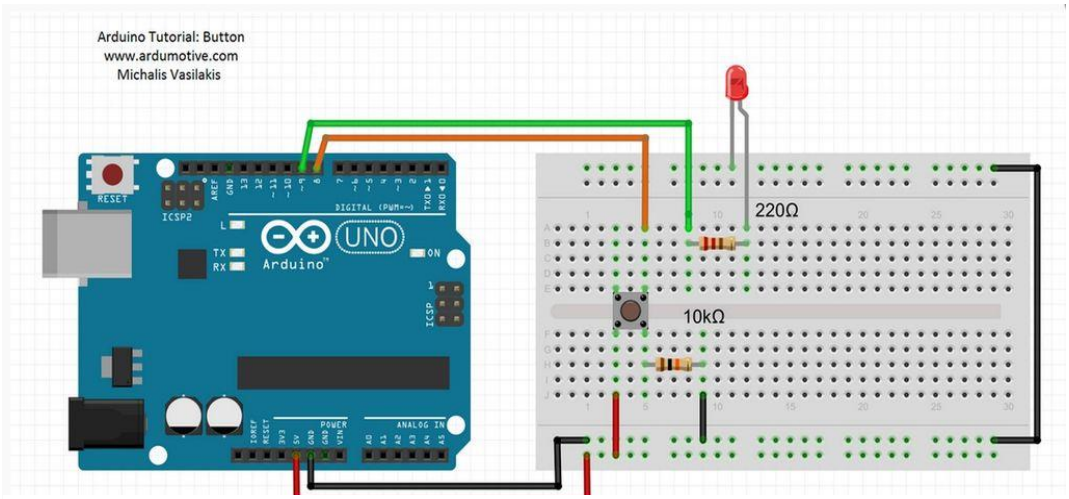
Resistors

III. Button, μας δίνει την δυνατότητα να μπορούμε να παρεμβαίνουμε στο κύκλωμά μας όποτε εμείς θέλουμε για παράδειγμα με το να δίνουμε ή όχι ρεύμα ενός τμήματος του κυκλώματος.



Button

Παρακάτω το σχήμα και ο κώδικας:



Παράδειγμα 1: Led_Button

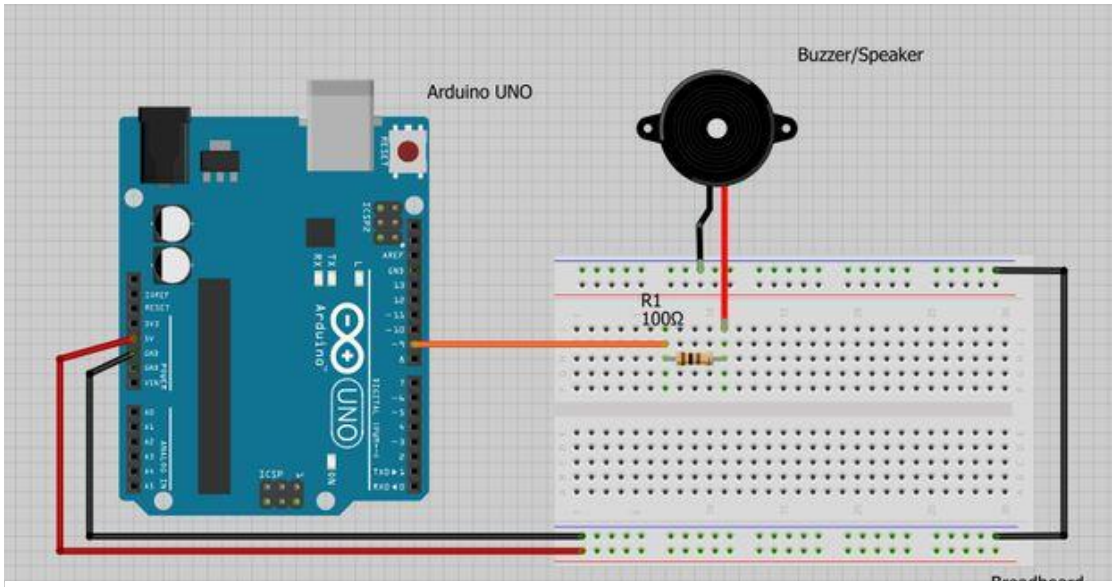
```
ConstintbuttonPin = 4;
constintledPin = 3;


IntbuttonState = 0;

intflag=0;

void setup() {
pinMode(ledPin, OUTPUT);
pinMode(buttonPin, INPUT_PULLUP);
}
void loop(){
buttonState = digitalRead(buttonPin);
if (buttonState == LOW) {
if ( flag == 0){
digitalWrite(ledPin, HIGH);
 flag=1; //change flag variable
}
else if ( flag == 1){
```

```
digitalWrite(ledPin, LOW);
flag=0;
 }
 }
delay(200); //Small delay
}
```

IV. Buzzer, δουλεύει όπως ακριβώς ένα led μόνο που αντί για φώς παράγει ήχο. Το κόκκινο καλώδιο συνδέεται με την πηγή και το μαύρο με τη γείωση.

Buzzer

Παρακάτω το σχήμα και ο κώδικας:



Παράδειγμα 2: Buzzer

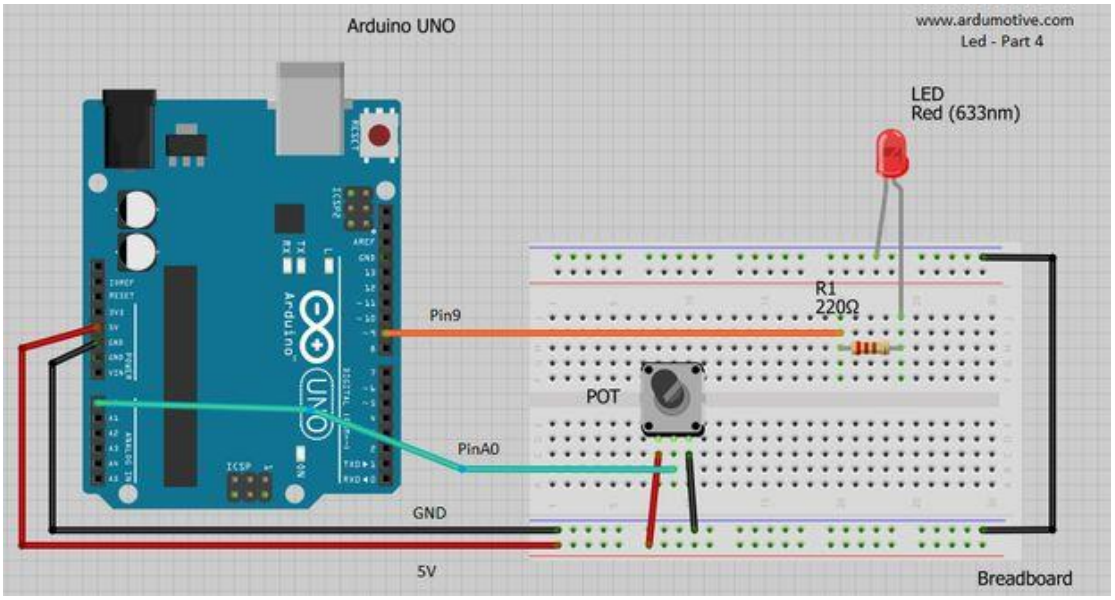
```
const int buzzer = 9;

void setup(){
pinMode (buzzer, OUTPUT);
}

void loop(){
tone(buzzer, 1000);
delay(1000);
noTone (buzzer);
delay(1000);
```

}

V. Potentiometer, γυρίζοντας τον μοχλό που διαθέτει, μπορούμε να μεταβάλλουμε την αντίσταση του, αφήνοντας έτσι να περάσει λιγότερο ή περισσότερο ρεύμα.

Potentiometer

Παρακάτω το σχήμα και ο κώδικας:



Παραδειγμα 3: Led_Potentiometer

```
const int ledPin = 9;
const int potPin = A0;

int value;

void setup(){
pinMode(ledPin, OUTPUT);
pinMode(potPin, INPUT);
}

void loop(){
```

```
value = analogRead(potPin);
value = map(value, 0, 1023, 0, 255);
analogWrite(ledPin, value);
delay(100);
}
```

Κεφάλαιο 3 – Η γλώσσα προγραμματισμού Arduino IDE

Για να λειτουργήσει και να υλοποιήσει αυτό που θέλει ένας χρήστης με ένα arduino, θα πρέπει πρώτα να το προγραμματίσει γράφοντας κάποιον κώδικα όπως έγινε παραπάνω. Αυτός ο κώδικας δημιουργείται στην γλώσσα προγραμματισμού που χρησιμοποιεί το arduino, την ArduinoIDE, μια γλώσσα προγραμματισμού βασισμένη στη ΣκαιC++.

ArduinoIDE

Είναι η γλώσσα προγραμματισμού που θα χρησιμοποιήσουμε για το Arduino προκειμένου να υλοποιήσουμε τα κυκλώματα μας. Μπορούμε να την κατεβάσουμε από το επίσημο sitehttp://arduino.cc/en/Main/Software εντελώς δωρεάν και για όποια έκδοση λειτουργικού συστήματος διαθέτουμε στον υπολογιστή μας (π.χ. Windows, Mac, Linux).
Η λογική της είναι πολύ απλή. Στην ουσία υπάρχουν δύο βασικές συναρτήσεις:

A. **setup()**:
   εδώ βάζουμε όλους τις συναρτήσεις όπου θα τρέξουν μια φορά αφού συνδέσουμε το arduinoστο ρεύμα ή όταν πατηθεί το κουμπί resetυπάρχει.
B. **loop():**
   εδώ είναι το κύριο μέρος του προγράμματος μας στο οποίο οι εντολές που έχουμε γράψει θα υλοποιηθούν και στο τέλος θα ξαναεκτελεστούν από την αρχή. Αυτό θα συνεχίζεται έως ότου πατηθεί το κουμπί resetή αποσυνδεθεί από το ρεύμα το arduino.

Εκτός από τις δομές επανάληψης θα πρέπει να δηλώσουμε και το είδος των μεταβλητών που θα χρησιμοποιήσει το πρόγραμμά μας. Τα ονόματα που θα δηλώσουμε για τις μεταβλητές μας δεν διαφέρουν σε σχέση με άλλες γλώσσες προγραμματισμού όπως είναι και οι παρακάτω:

- ✓ **Boolean**, με τιμές 0 και 1 (αφορά λογικές τιμές,True-False)
- ✓ **byte**, με τιμές 0-255
- ✓ **int**, για ακέραιες μεταβλητές με τιμές -32768 – 32768
- ✓ **long**, για ακέραιες μεταβλητές με τιμές -2147483648 έως και 2147483647
- ✓ **float**, δεκαδικοί αριθμοί
- ✓ **char**, ένας χαρακτήρας (μέγεθος ένα Byte)
- ✓ **string**, πίνακας χαρακτήρων

3.1 Βασικές συναρτήσεις

1. **pinMode(pin, OUTPUT/INPUT);**
2. **digitalWrite(pin, HIGH/LOW);**
3. **digitalRead(pin);**
4. **variable=analogRead(analogPin);**
5. **variable=digitalRead(digitalPin);**
6. **delay(ms);**
7. **map(0,1023,0,255);**
8. **tone(pin,Freq);/noTone(pin);**

Με τη χρήση της συνάρτησης **pinMode** ορίζουμε την θύρα που είναι συνδεδεμένο το led εάν θα είναι εξόδου(OUTPUT) ή εισόδου(INPUT)

Με τη χρήστης **digitalWrite** θα έχουμε ψηφιακή έξοδο(HIGH) ή είσοδο(LOW). Η αντίστοιχη θύρα θα πρέπει να ορισθεί ως εισόδου στη διαδικασία "setup()", με χρήση της pinMode
π.χ. pinMode(10,INPUT);

Με τη χρήση της συνάρτησης **variable=analogRead** δίνουμε τη δυνατότητα να διαβάσει ένα αναλογικό εξάρτημα, για παράδειγμα μια τιμή φωτοαντήστασης ή ποτενσιόμετρου.

Με τη χρήση της **variable=digitalWrite** δίνουμε τη δυνατότητα να διαβάσει ένα ψηφιακό εξάρτημα

Με τις υπόλοιπες συναρτησεις **delay**, mapκαι tone λειτουργούμε ως εξής:
στην**delay**επιλέγουμε για πόσο χρόνο (ms)θα παγώσει ο κώδικάς μας, στην**map** γίνεται αντιστοίχιση τιμών για παράδειγμα τις τιμές από ένα ποτενσιόμετρο να τις κάνει για ένα led.
Τέλος με την **tone/noTone**γίνεται προσαρμογή ενός τόνου ή συχνότητας σε ήχο.

Για όλες τις παραπάνω συναρτησεις εκτός των **map**και **delay**θα πρέπει να δηλώσουμε τις αντίστοιχες θύρες που χρησιμοποιούν στην setup().

3.1.1 Παραδείγματα με χρήση των βασικών συναρτήσεων

- **pinMode();**

pinMode**(**12,OUTPUT**);**
pinMode(8,INPUT);

- **digitalWrite();**

digitalWrite(ledPin,HIGH);

*Η αντίστοιχη θύρα θα πρέπει να έχει οριστεί ως εξόδου στην **setup()***
pinMode(10,OUTPUT);

- **variable=analogRead(analogPin);**

int i=analogReade(A1);

*Η αντίστοιχη θύρα θα πρέπει να έχει οριστεί ως εξόδου στην **setup()***
pinMode(A1,INPUT);

- **variable=digitalRead(digitalPin);**

Val=digitalRead(ledPin);

*Η αντίστοιχη θύρα θα πρέπει να έχει οριστεί ως εξόδου στην **setup()***
pinMode(10,INPUT);

- **delay(ms);**

delay(1000); σταματάει το πρόγραμμά μας για 1second

- **map(0,1023,0,255)**

val=map(val,0,1023,0,179);

- **tone(pin,Freq);/noTone(pin);**

tone(buzzer, 1000);
noTone(buzzer);

*Η αντίστοιχη θύρα θα πρέπει να έχει οριστεί ως έξοδος στην **setup()***
pinMode(buzzer, OUTPUT);

3.2 Δομές επανάληψης

Εκτός από τις παραπάνω βασικές συναρτήσεις θα χρειαστούμε πολλές φορές να ελέγξουμε μια συνθήκη για το εάν θα εκτελεστεί ένα τμήμα κώδικα ή αν θα εκτελεστεί κάποιο άλλο ή να επαναλάβουμε κάποια διαδικασία αρκετές φορές. Αυτό μπορούμε να το πράξουμε με τη χρήση δομών επιλογής, επανάληψης όπως και σε άλλες γλώσσες προγραμματισμού (java,C,C++) στις οποίες έχει βασιστεί και η ArduinoIDE.

Μια δομή επιλογής συντάσσεται ως εξής:

```
if(συνθήκη){
<εντολές1>
}
else{
<εντολές2>
}
```

- (συνθήκη) ο έλεγχος που θέλουμε να γίνει, χρησιμοποιώντας τους τελεστές (>,<,=,<>,>=,<=) καθώς και λογικούς τελεστές(||,==,&&)
- <εντολές1 και 2> οι εντολές που θα εκτελεστούν εάν ισχύει ή όχι η συνθήκη
- elseμπορεί και να μην υπάρχει

Μια δομή επανάληψης συντάσσεται ως εξής:

```
for(<αρχική τιμή>;<συνθήκη_τερματισμού>;<βήμα>){

<εντολές>

}
```

- <αρχική τιμή>δίνουμε την αρχική τιμή πχ i=0;
- <συνθήκη_τερματισμού> η συνθήκη για να τελειώσει η επανάληψη πχ i<10;
- <βήμα> αλλαγή κάθε επανάληψης πχ i++

```
do{

<εντολή>

}while(συνθήκη);
```

Εδώ χρησιμοποιούνται οι δεσμευμένες λέξεις do, while. Μετά τη λέξη do ακολουθούν οι εντολές οι οποίες τελειώνουν πριν από τη λέξη while. Και στη περίπτωση αυτή, οι συνθήκες που χρησιμοποιούνται είναι ίδιες με τις συνθήκες που χρησιμοποιούνται στην εντολή επιλογής. Η εντολή τελειώνει με το while (συνθήκη). Σε αυτή τη μορφή της εντολής επανάληψης ο έλεγχος γίνεται μετά την εκτέλεση της ομάδας εντολών. Αυτό σημαίνει ουσιαστικά ότι η ομάδα εντολών εκτελείται τουλάχιστον μία φορά. Μετά την εκτέλεση της ομάδας εντολών ελέγχεται η συνθήκη. Αν αυτή δεν ικανοποιείται, τότε εκτελείται ξανά η ομάδα εντολών, όσο η συνθήκη παραμένει ψευδής. Όταν η συνθήκη γίνει αληθής, λήγει η

εκτέλεση της επαναληπτικής δομής. Σε αυτή την δομή δεν είναι απαραίτητο να χρησιμοποιήσετε άγκιστρα στη δομή do - while, αν υπάρχει μόνο μια εντολή στο σώμα. Ωστόσο προτιμάτε να περιλαμβάνετε τα άγκιστρα για να αποφεύγετε την σύγχυση μεταξύ των δομών do - while και while.

while (συνθήκη)
{
ομάδα εντολών;
}

Εδώ χρησιμοποιείται η δεσμευμένη λέξη while. Μετά τη λέξη while ακολουθεί η συνθήκη συνέχειας. Αμέσως μετά γράφεται η εντολή ή η ομάδα εντολών. Οι συνθήκες που χρησιμοποιούνται εδώ είναι ίδιες με τις συνθήκες που χρησιμοποιούνται στην εντολή επιλογής. Στην αρχή της εκτέλεσης ελέγχεται η συνθήκη. Αν αυτή ικανοποιείται, τότε εκτελείται η εντολή ή η ομάδα εντολών. Στη συνέχεια, ελέγχεται ξανά η συνθήκη. Δεν είναι σίγουρο ότι αυτή εξακολουθεί να είναι αληθής διότι το αποτέλεσμά της μπορεί να έχει επηρεαστεί από την εκτέλεση της εντολής ή της ομάδας εντολών. Αν η συνθήκη ικανοποιείται ξανά, εκτελείται πάλι η εντολή ή η ομάδα εντολών. Η διαδικασία συνεχίζεται, όσο η συνθήκη παραμένει αληθής. Όταν η συνθήκη συνέχειας γίνει ψευδής, η εντολή ή η ομάδα εντολών δεν εκτελείται και τότε λήγει η εκτέλεση της δομής.

3.2.1 Boolean μεταβλητές

Μερικές φορές θα πρέπει να επιλέξουμε ανάμεσα σε δύο καταστάσεις για το κύκλωμα μας ωστε το κύκλωμα να εκτελέσει μια διαδρομή ανάλογη με την επιλογή μας. Για παράδειγμα όπως το οπή offή το ζεστό ή κρύο. Μια μεταβλητή Booleanπαίρνει τιμές 0,falseκαι 1,trueκαι θα πρέπει να τη δηλώσουμε για να τη χρησιμοποιήσουμε.

Πχ. boolean raining = true;
ή
raining = false;

Χρήση μεταβλητών Booleanμαζί με δομή επανάληψης if:

- if (raining == true){
  if (summer != true){
  }
  }

Μπορούμε να χρησιμοποιήσουμε διάφορες λειτουργίες για μια Booleanμεταβλητή όπως η not (!), and (&&) , or (||).

Not
Τη συμβολίζουμε με ! και τη χρησιμοποιούμε για κάτι που δεν είναι αληθές.

- if (!raining){
  (raining == false)
  }

And

Τη συμβολίζουμε με && και τη χρησιμοποιούμε για να λιγοστέψουμε τον αριθμό των testμε if.

- if ((raining == true) && (!summer)){
  (raining == true and summer == false)
  }

Or

Τη συμβολίζουμε με || και η χρήση της –η οποία είναι πολύ απλή- θα την δούμε παρακάτω με ένα παράδειγμα.

- if ((raining == true) || (summer == true)){
  }

3.2.2 Χρήση Boolean μεταβλητών σε συνδυασμό με δομή επανάληψης

Χρήση δύο ή περισσότερων λειτουργιών σε μια συνθήκη

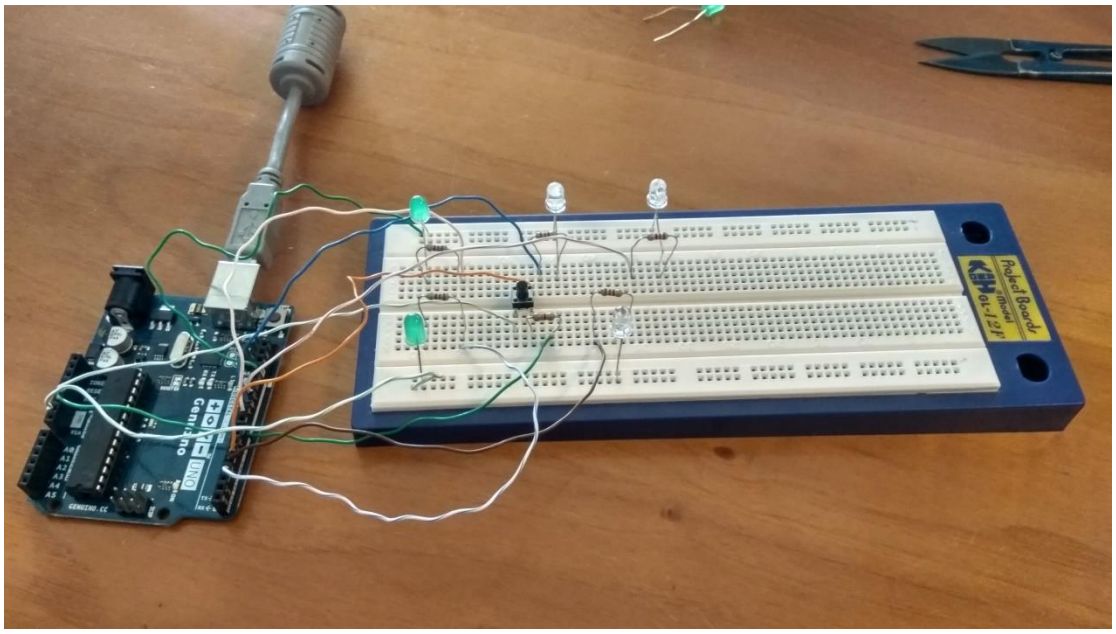
- if (snow == true&&rain == true&& !hot){
  }
- if ((snow == true || rain == true) && hot == false)){
  }
- if ((snow || rain) && !hot){
  }

Ένα καλό παράδειγμα για τη χρήση Boolean μεταβλητών στη γλώσσα προγραμματισμού που χρησιμοποιεί το arduino την arduinoIDE είναι το παρακάτω όπου αναπαριστά τον έλεγχο την κυκλοφοριακής κίνησης.

Τα υλικά που θα χρειαστούμε για τη δημιουργία του κυκλώματός μας

- ✓ (2x) led κόκκινο χρώμα
- ✓ (2x) led πράσινο χρώμα
- ✓ (1x) led πορτοκαλί χρώμα
- ✓ (1x) Genuino Uno R3
- ✓ (1x) button
- ✓ (1x) breadboard

Σχήμα:



Παράδειγμα 4: Έλεγχος κυκλοφοριακής κίνησης

Κώδικας:

```
constintledRed=11;
constintledOrange=10;
constintledGreen=9;
constintpedRed=4;
constintpedGreen=3;
constintbtn=5;
constintlastPedPass=0;
constintbtnState=0;


void setup(){

pinMode(11,OUTPUT);
pinMode(10,OUTPUT);
pinMode(9,OUTPUT);
pinMode(4,OUTPUT);
pinMode(3,OUTPUT);
pinMode(5,OUTPUT);

digitalWrite(4,HIGH);
digitalWrite(3,HIGH);
digitalWrite(11,HIGH);
digitalWrite(10,HIGH);
digitalWrite(9,HIGH);
```

```
}

void loop(){
intbtnState=digitalRead(btn);

if(btnState==HIGH&&(millis()-lastPedPass)>5000){

digitalWrite(11,LOW);
digitalWrite(10,HIGH);
digitalWrite(9,LOW);

digitalWrite(11,HIGH);
digitalWrite(10,LOW);
digitalWrite(9,LOW);
delay(2000);

digitalWrite(4,LOW);
digitalWrite(3,HIGH);
 delay(3000);

digitalWrite(4,HIGH);
digitalWrite(3,LOW);
delay(1000);

digitalWrite(11,LOW);
digitalWrite(10,LOW);
digitalWrite(9,HIGH);
intlastPedPass=millis();

 }

}
```

3.3 Αναλογική και ψηφιακή είσοδος/έξοδος

Ψηφιακή είσοδο/έξοδο

Και τα 14 pinsπου έχει ένα arduinoμπορούν να δουλεύουν ως ψηφιακές έξοδοι, δίνοντας ως έξοδο 0 ή 5V. Αυτό γίνεται με τη χρήση της συνάρτησης digitalWrite(Pin,Value), όπου το Pinαφορά στο νούμερο της θύρας που θα δώσουμε τάση εξόδου από 0 έως 5V.Οι τιμές που θα πάρει το Pinθα είναι δύο:

1. LOW(0V στην έξοδο)
2. HIGH(5V στην έξοδο)

Για παράδειγμα digitalWrite(ledPin,HIGH);

,όπου η αντίστοιχη θύρα θα οριστεί ως εξόδου από την αρχή στην setup()

pinMode(10,OUTPUT);

Τα ίδια ισχύουν και για την ψηφιακή είσοδο μόνο που εδώ μιλάμε για είσοδο. Η συνάρτηση που θα χρησιμοποιούμε είναι η digitalRead(Pin),όπου Pinαναφέρεται στο νούμερο για την οποία θα πάρουμε είσοδο, ενώ η συνάρτηση επιστρέφει με το όνομα της την τιμή εισόδου. Οι τιμές που θα πάρει το Pin θα είναι όπως αυτές στην έξοδο **LOW**και**HIGH.**

Παράδειγμα:
Val=digitalRead(ledPin);

,όπου η αντίστοιχη θύρα θα οριστεί ώς εισόδου απο την αρχή στην setup()

pinMode(10,INPUT);

Αναλογική έξοδος(PMW)

Κάποια απο τα pins του arduino έχουν την ένδειξη PMW, δηλαδή μπορούν να προσομοιώσουν την αναλογική έξοδο μέσω παλμοκωδικής διαμόρφωσης. Έτσι με τιμές απο το 0 έως το 255 προσομοιώνουμε το διάστημα απο 0 έως 5V. Αυτό μπορούμε να το κάνουμε με τη χρήση της συνάρτησης analogWrite(Pin,Value), όπου το Pin αναφέρεται στο νούμερο της θύρας για την οποία θα δώσουμε ρεύμα εξόδου, ενώ η τάση εξόδου κυμαίνεται απο 0V μέχρι 5V, οι οποίες τιμές της τάσης αναλογικά αναπαρίστανται με τιμές στη μεταβλητή Value.Η τιμή 0 δίνει 0V στην έξοδο(Pin), ενώ η τιμή 255 δίνει την τάση 5Vστην έξοδο (Pin). Αναλογικά μπορούμε να δώσουμε ενδιάμεσες τάσεις (π.χ. 122 για τάση 2,5V):
analogWrite(ledPin,122);

Στο arduinoμας τα pinsπου είναι αναλογικά είναι τα 3,5,6,9,10,11.

Όπως και στη ψηφιακή είσοδο και έξοδο έτσι και εδώ θα πρέπει πάντα να δηλώνουμε την αντίστοιχη θύρα και να την ορίζουμε ως εξόδου στην setup():

pinMode(10,OUTPUT);

Αναλογική είσοδος

Το Arduino έχει 6 αναλογικές εισόδους, οι οποίες χαρακτηρίζονται με τα σύμβολα A0, A1, A2, A3,A4, A5. Μπορούμε να συνδέσουμε κάποιο αναλογικό εξάρτημα (π.χ. ένα ποτενσιόμετρο) και να το διαβάσουμε ως είσοδο. Αυτό γίνεται με χρήση της συνάρτησης analogRead(Pin), όπου το όρισμα Pin αναφέρεται στο νούμερο της θύρας για την οποία θα πάρουμε είσοδο, ενώ η συνάρτηση επιστρέφει με το όνομά της την τιμή εισόδου. Η τιμή εισόδου κυμαίνεται από 0 μέχρι και 1023. Συνήθως χρησιμοποιούμε μια μεταβλητή για να καταχωρήσουμε την τιμή.

Για παράδειγμα:

```
int r = analogRead(A1);
```

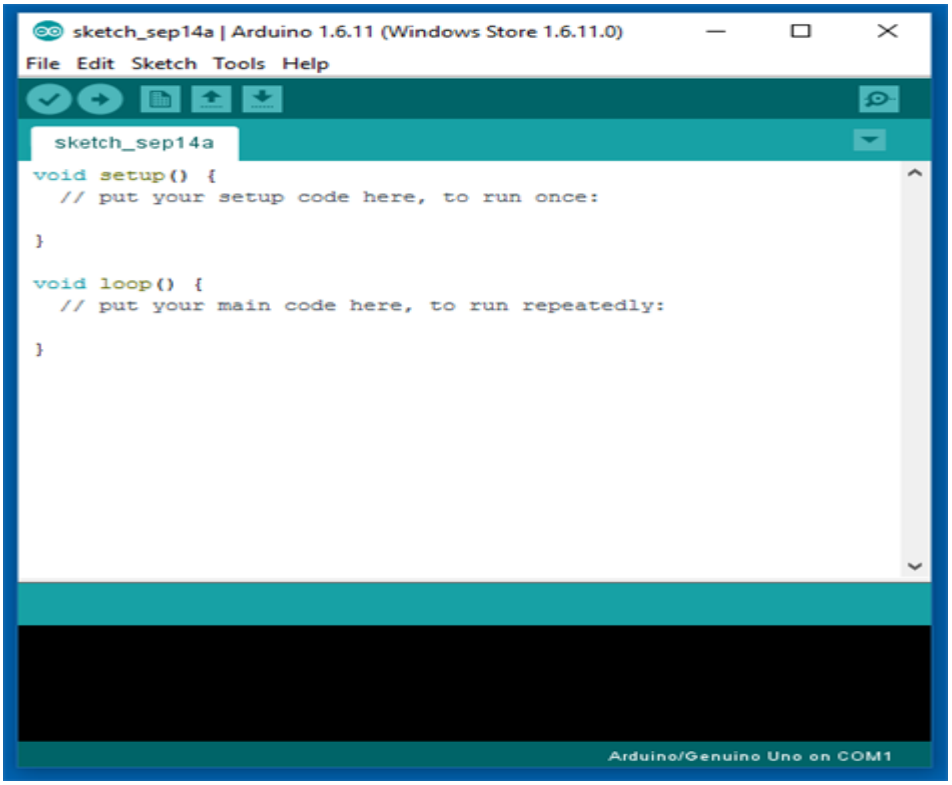
Όπως και στη ψηφιακή είσοδο και έξοδο έτσι και εδώ θα πρέπει πάντα να δηλώνουμε την αντίστοιχη θύρα και να την ορίζουμε ως εξόδου στην setup():

Για παράδειγμα:

```
pinMode(A1, INPUT);
```

3.4 Μεταφόρτωση του προγράμματος στη μονάδα μας

Προκειμένου να υλοποιήσουμε το κύκλωμά μας θα πρέπει αφού το έχουμε σχεδιάσει και συνδέσει με τον υπολογιστή μας να γράψουμε και τον κώδικα που θα εκτελέσει το arduino.Αυτό για να γίνει θα πρέπει να χρησιμοποιήσουμε το πρόγραμμα του Arduinογράφοντας σε γλώσσα ArduinoIDE.



Arduino IDE

Αφού συνδέσουμε την πλακέτα μας με τον υπολογιστή μέσω USB και γράψουμε τον κώδικα που θέλουμε θα πρέπει να τον μεταφέρουμε στην πλακέτα και να προχωρήσει στην υλοποίηση.

Πρώτα επιλέγουμε την πλακέτα μας από το μενού *Εργαλεία* του Arduino και την σειριακή θύρα - είναι η σειριακή θύρα που έχει αντιστοιχίσει το λειτουργικό μας στην πλακέτα Arduino που συνδέεται μέσω του USB καλωδίου. Αν χρησιμοποιείτε Windows αυτή θα είναι της μορφής COMX (π.χ. COM3, COM11), ενώ στο Linux η θύρα θα εμφανιστεί ως /dev/ttyXXX.

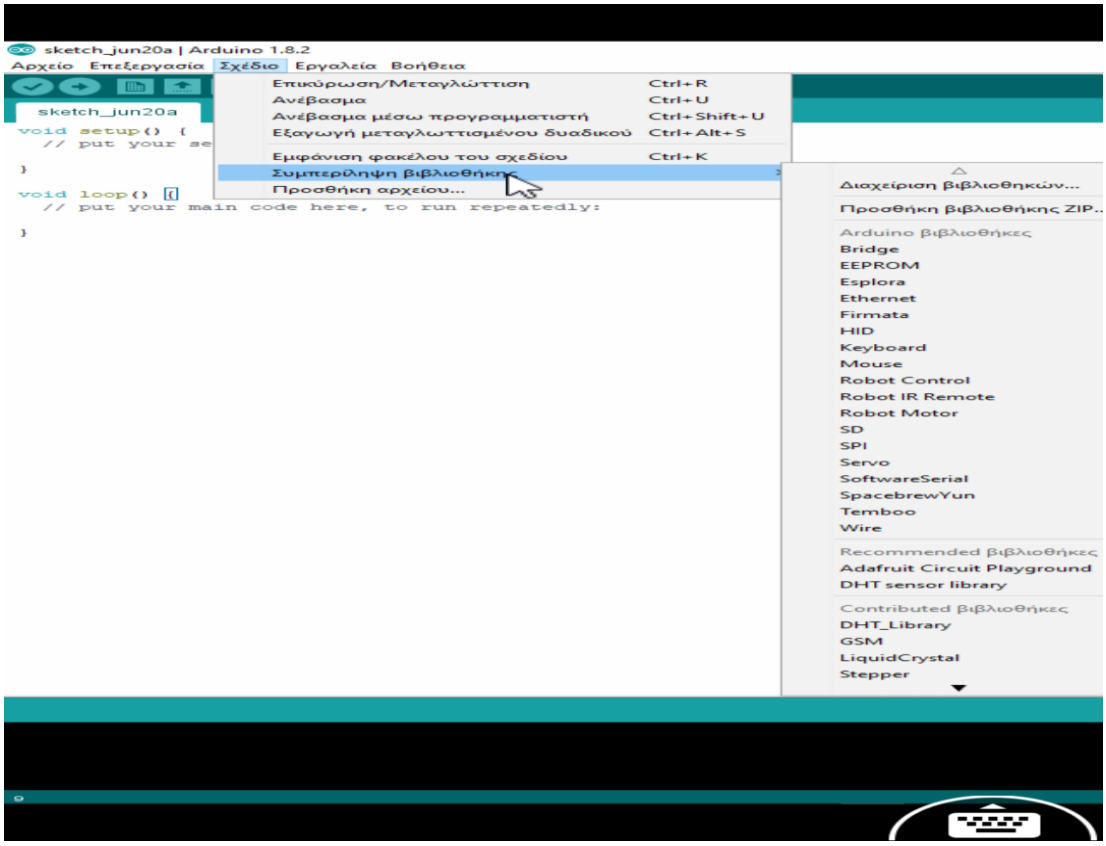
Έπειτα επιλέγουμε το σήμα με το τσεκ (Επικύρωση) προκειμένου να ελέγχει ο κώδικας για τυχόν λάθη τα οποία θα εμφανιστούν στην περιοχή/οθόνη κάτω απο το πρόγραμμα μας. Αφού γίνει ο έλεγχος θα επιλέξουμε το σήμα με το βελάκι (Ανέβασμα) προκειμένου να μεταφορτώσουμε τον κώδικα στην πλακέτα και να υλοποιηθεί.

3.5 Βιβλιοθήκες στην arduino IDE

Όπως σε κάθε γλώσσα προγραμματισμού έτσι και στην arduino IDE υπάρχει μια συλλογή από έτοιμα υποπρογράμματα που χρησιμοποιούνται για την ανάπτυξη λογισμικού. Οι βιβλιοθήκες περιέχουν υποβοηθητικό κώδικα και δεδομένα, παρέχοντας, με αυτόν τον τρόπο, υπηρεσίες σε προγράμματα. Αυτό επιτρέπει τον διαμοιρασμό και τη χρήση του κώδικα και των δεδομένων με αρθρωτό τρόπο. Η έννοια της βιβλιοθήκης είναι αναπόσπαστο τμήμα του δομημένου προγραμματισμού και αναπτύχθηκε παράλληλα με αυτόν. Κάποια εκτελέσιμα αρχεία (executables) είναι προγράμματα και βιβλιοθήκες ταυτόχρονα, αλλά οι περισσότερες βιβλιοθήκες δεν είναι εκτελέσιμες. Τα εκτελέσιμα αρχεία και οι βιβλιοθήκες αναφέρονται το ένα στον κώδικα και τα δεδομένα του άλλου μέσω μιας διαδικασίας που ονομάζεται σύνδεση και την πραγματοποιεί ο Χρήστης. Τα σύγχρονα λειτουργικά συστήματα παρέχουν βιβλιοθήκες που υλοποιούν την πλειονότητα των υπηρεσιών του συστήματος. Έτσι, ο περισσότερος κώδικας που χρησιμοποιούν οι σύγχρονες εφαρμογές παρέχεται από αυτές τις βιβλιοθήκες και δεν χρειάζεται να γραφεί από την αρχή για κάθε νέο πρόγραμμα.

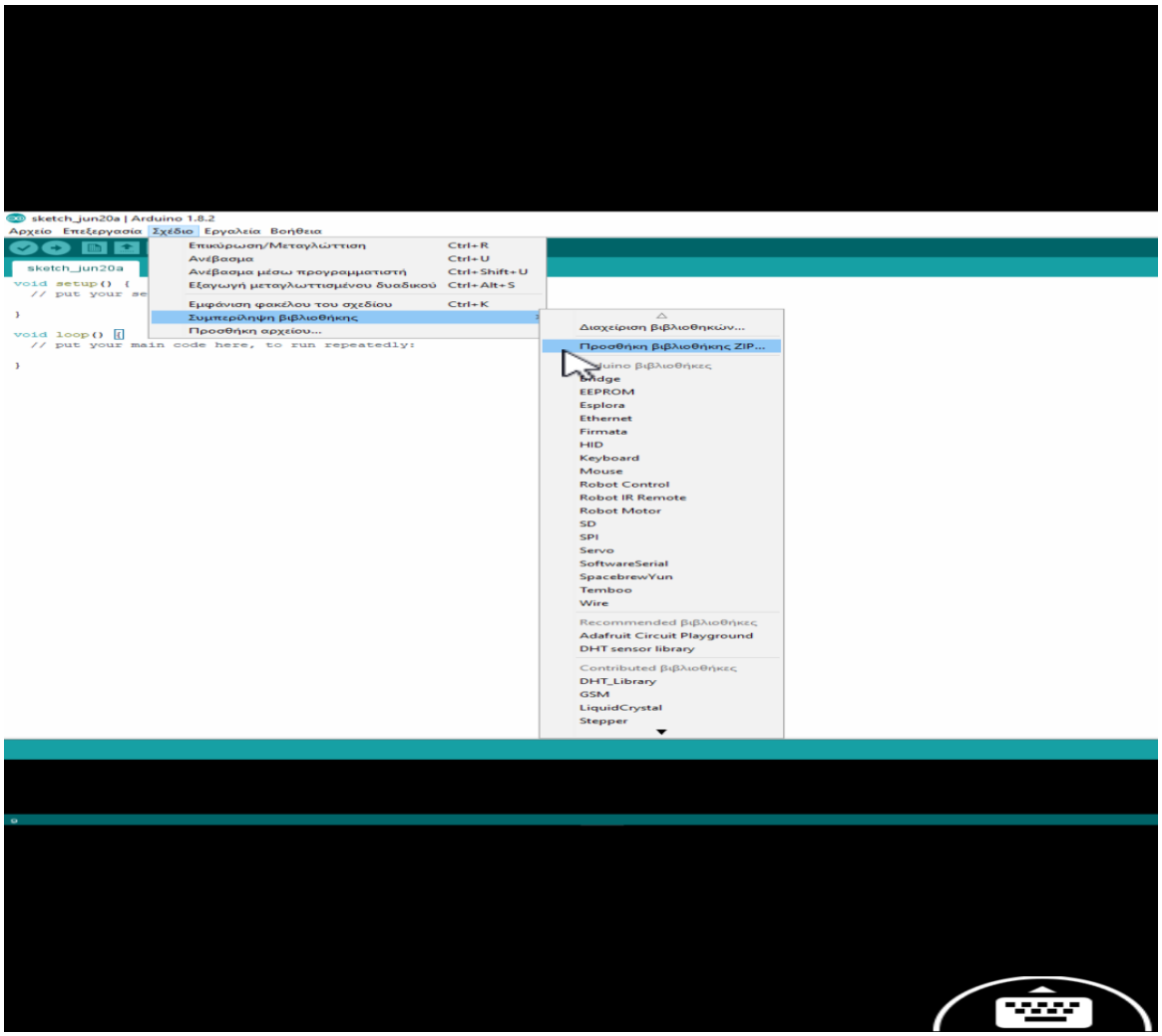
Η χρήση βιβλιοθήκης στην arduinoIDEείναι πολύ απλή. Η arduinomας προσφέρει απο την αρχή κάποιες βιβλιοθήκες οι οποίες είναι εγκατεστημένες και μπορούμε να τις καλέσουμε όποτε εμείς τις χρειαστούμε.

Όταν χρειαστούμε μια βιβλιοθήκη θα πρέπει να ακολουθήσουμε πατώντας στο μενού ην κεφαλίδα **Σχέδιο** και μετά **Συμπερίληψη βιβλιοθήκης** μας εμφανίζεται μια λίστα με όλες τις εγκατεστημένες βιβλιοθήκες. Το μόνο που έχουμε να κάνουμε είναι να επιλέξουμε την βιβλιοθήκη που θέλουμε.

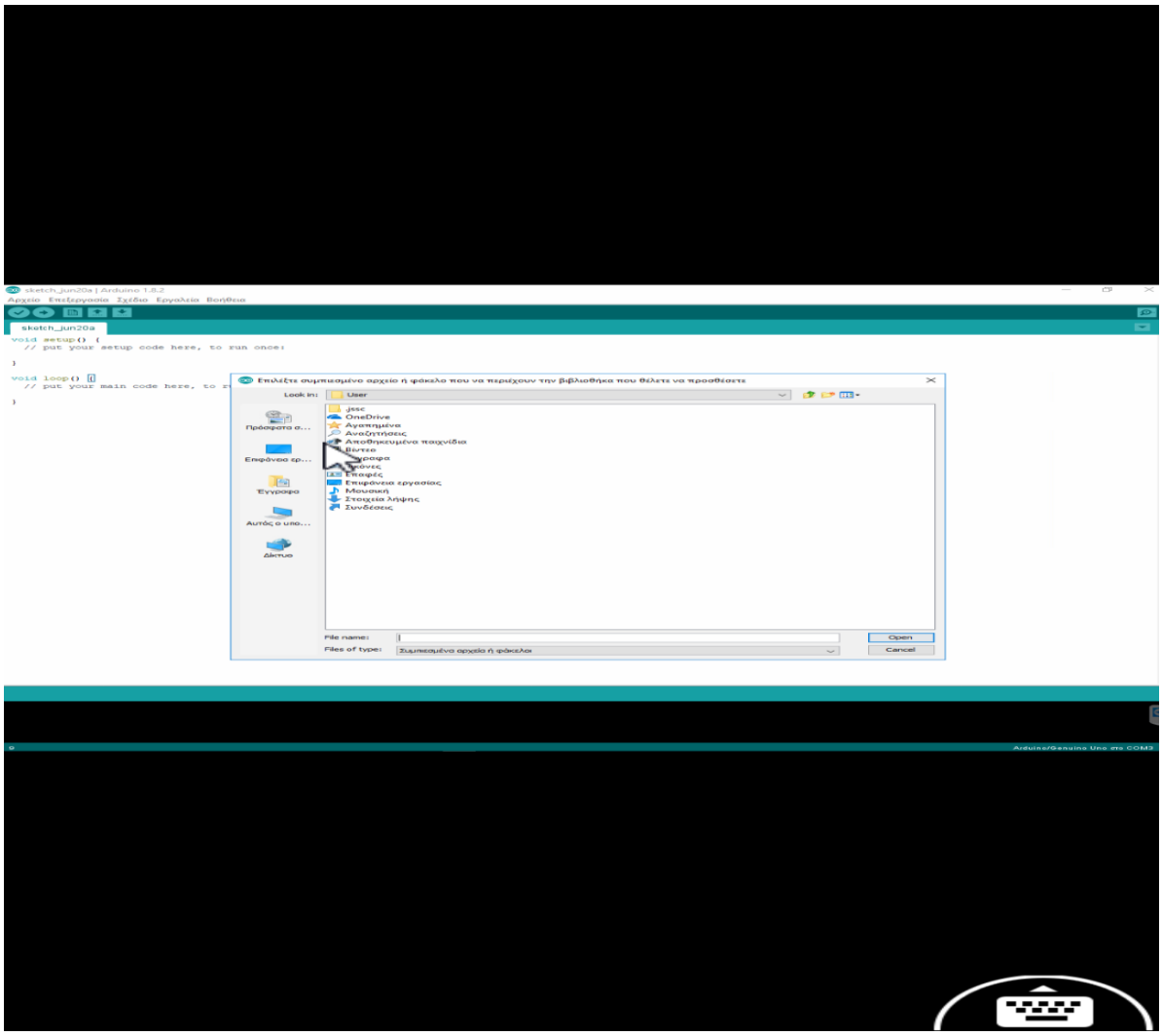


Βήμα 1: Σχέδιο, Συμπερίληψη βιβλιοθήκης

Στην περίπτωση που η βιβλιοθήκη που χρειαζόμασταν είναι περασμένη μέσα στο arduinoθα χρειαστεί να το κάνουμε εμείς χειροκίνητα κατεβάζοντας την απο το internet. Αφού την κατεβάσουμε θα πρέπει να την προσθέσουμε στην arduinoIDEακολουθώντας την ίδια διαδρομή με αυτή της επιλογής μια βιβλιοθήκης. Πατώντας λοιπόν **Σχέδιο**->**Συμπερίληψη βιβλιοθήκης** ->**Προσθήκη βιβλιοθήκης ZIP…** επιλέγοντας το αρχείο .zipπου κατεβάσαμε.



Βήμα 2: Προσθήκη βιβλιοθήκης ZIP



Βήμα 3: Επιλογή βιβλιοθήκης απο φάκελο

3.6 Παραδείγματα προγραμμάτων στην Arduino IDE

Καθώς οι σημαντικότερες συναρτήσεις έχουν προαναφερθεί σε αυτό το κεφάλαιο θα αναφερθούμε περισσότερα σε διάφορα παραδείγματα πάνω στη γλώσσα προγραμματισμού που χρησιμοποιούμε στο arduino την arduinoIDE.

Ένα απλό πρόγραμμα μπορεί να είναι η δημιουργία μιας λειτουργιάς που να επαναλαμβάνει διαφορές ενέργειες κατά παραγγελία. Όπως το παρακάτω κομμάτι κώδικα όπου θα ανάψει ένα λαμπάκι και θα σβήσει δυο φόρες:

```
void blinkLED()
{
digitalWrite(13, HIGH);
delay(1000);
digitalWrite(13, LOW);
delay(1000);
digitalWrite(13, HIGH);
delay(1000);
digitalWrite(13, LOW);
delay(1000);
}
```

και ολοκληρωμένο:

```
#define LED 13
#define del 200

void setup()
{
pinMode(LED, OUTPUT);
}
void blinkLED()
{
digitalWrite(LED, HIGH);
delay(del);
digitalWrite(LED, LOW);
delay(del);
digitalWrite(LED, HIGH);
delay(del);
digitalWrite(LED, LOW);
delay(del);
}
void loop()
{
blinkLED();
delay(1000);
}
```

Όταν η λειτουργία blinkLED() καλεσθεί στην voidloop() το arduino θα εκτελέσει τις εντολές στα πλαίσια της voidblinkLED().Με άλλα λόγια δημιουργούμε την δικιά μας λειτουργία και την χρησιμοποιούμε όποτε την χρειαστούμε.

Δημιουργία μιας λειτουργίας που να ορίζει το πόσες φορές θα ανάψει και θα κλείσει ένα λαμπάκι.

```
void blinkLED(int cycles, int del)
{
for (int z = 0 ; z < cycles ; z++ )
{
digitalWrite(LED, HIGH);
delay(del);
digitalWrite(LED, LOW);
delay(del);
}
}
```

Με τη δημιουργία αυτής της νέας λειτουργίας voidblinkLED() μπορούμε να δεχτούμε δύο ακέραιες μεταβλητές τη cycles (τον αριθμό που θα αναβοσβήσει το λαμπάκι) και τη del(και τον χρόνο που θα περάσει μέχρι να ανάψει αφού έχει σβήσει το λαμπάκι). Εάν θελήσουμε να αναβοσβήσει το λαμπάκι 12 φορές, με μια delay(100) θα γράψουμε την λειτουργία blinkLED(12,100) όπως γίνεται παρακάτω πιο αναλυτικά:

```
#define LED 13

void setup()

{
pinMode(LED, OUTPUT);
}
void blinkLED(int cycles, int del)
{
for (int z = 0 ; z < cycles ; z++ )
{
digitalWrite(LED, HIGH);
delay(del);
digitalWrite(LED, LOW);
delay(del);
}
}
void loop()
{
blinkLED(12, 100);
delay(1000);
}
```

Δημιουργία μίας λειτουργίας όπου να επιστρέφει μια τιμή

Αυτό μπορεί να πραγματοποιηθεί με την ίδια λογική της analogRead() όπου επιστρέφει μια τιμή ανάμεσα στο 0 και το 1023 μετρώντας μια αναλογική είσοδο. Η voidπου εμφανίζεται στην αρχή της λειτουργίας μας δείχνει ότι η λειτουργία δεν επιστρέφει κάποια τιμή. Παρακάτω θα δημιουργήσουμε κάποιες λειτουργίες ώστε να μας επιστρέφεται κάποια τιμή.

Βαθμοί Celsius σε Fahrenheit:

```
floatconvertTemp(floatcelsius)//Όνομα της λειτουργίας
{
float fahrenheit = 0;
fahrenheit = (1.8 * celsius) + 32;
return fahrenheit;
}
```

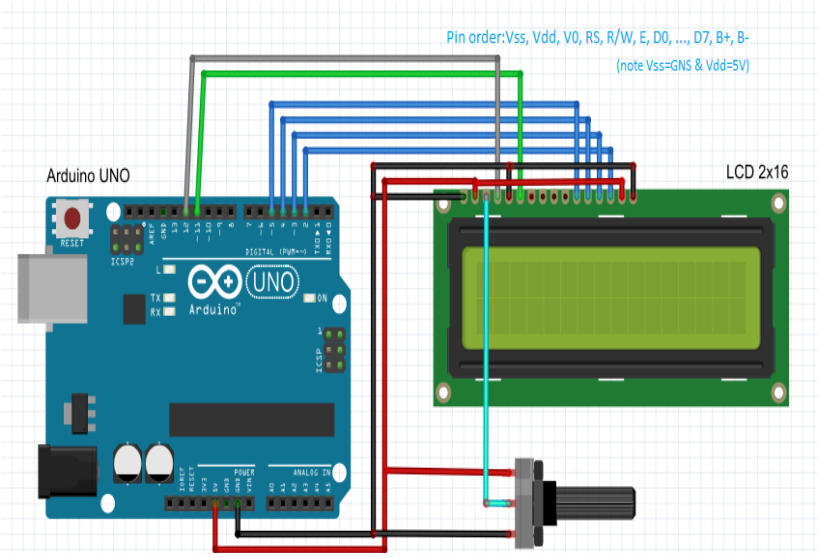
η τιμή που θα μας επιστραφεί θα είναι τύπου float
εάν θέλαμε να μετατρέψουμε μια τιμή σε fahrenaheit όπως το 40 θα γράφαμε το εξής:

```
temp f=convertTemp(40);
```

το 40 θα πάει στην θέση του Celsiuskαι αφού πραγματοποιηθεί η πράξη θα μας επιστραφεί η τελική τιμή σε fahrenheit.

Χρήση οθόνης LCD16x2

Αρχικά για να μπορέσουμε να χρησιμοποιήσουμε την οθόνη αυτή θα πρέπει εισάγουμε την βιβλιοθήκη "LiquidCrystal" για οθόνες που είναι συμβατές με το Hitachi HD44780 driver. Οι οθόνες που είναι συμβατές με το Hitachi driver μπορούν να οδηγηθούν με δυο λειτουργίες, 4-bit και 8-bit. Η συνδεσμολογία με 4-bit χρησιμοποιεί 7 I/O pins του Arduino ενώ η λειτουργία 8-bit χρησιμοποιεί 11.



Παράδειγμα 5: Λειτουργία επιστροφής τιμής

Για να την εκτύπωση μηνύματος στην οθόνη, χρειαζόμαστε απλά την λειτουργία με τα 4-bit, έτσι σε αυτό το tutorial θα εργαστούμε πάνω σε αυτή.

Το κύκλωμα θα είναι ως εξής:
Η εντολή lcd.begin(16,2) αρχικοποιεί την βιβλιοθήκη με τις γραμμές και στήλες της οθόνης

που έχουμε. Για παράδειγμα, αν είχαμε μια οθόνη με 20 στήλες και 4 γραμμές (20x4) θα έπρεπε να αλλάξουμε την εντολή σε lcd.begin(20x4).

Η εντολή lcd.print("--minima--") εκτυπώνει ένα μήνυμα στην πρώτη γραμμή και πρώτη στήλη της οθόνης. Το "minima" θα πρέπει να είναι με λατινικούς χαρακτήρες (δεν επιτρέπετε χρήση Ελληνικών) και να έχει μέγιστο μήκος όσο και το μέγεθος της οθόνης σε στήλες. Για παράδειγμα μια οθόνη με 16 στήλες μπορεί να εμφανίσει στην πρώτη γραμμή μήνυμα με μέγιστο μήκος 16 χαρακτήρες (μαζί με τα κενά), ενώ μια οθόνη με 20 στήλες μπορεί να εμφανίσει μήνυμα με μέγιστο μήκος 20 χαρακτήρες.

Η εντολή lcd.setCursor(0,1) μετακινεί τον δείκτη στην πρώτη στήλη και δεύτερη γραμμή της οθόνης. Αν είχαμε μια οθόνη 20x4 και θέλαμε να ξεκινήσουμε το μήνυμα μας στην πέμπτη στήλη της τρίτης γραμμής θα αλλάζαμε την εντολή σε lcd.setCursor(4,2).

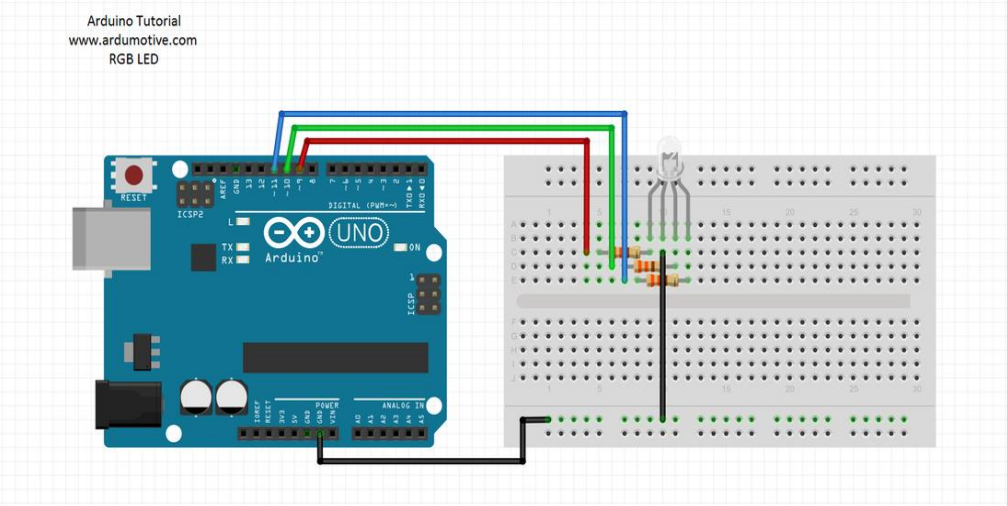
Ο κώδικας θα είναι ως εξής:

```
#include<LiquidCrystal.h>

LiquidCrystallcd(12, 11, 5, 4, 3, 2);

voidsetup() {
lcd.begin(16, 2);
lcd.print("Hello World!");

}
voidloop() {
lcd.setCursor(0, 1);
lcd.print("Ardumotive");

}
```

Χρήση RGBled

Ένα RGBLED (Red, Green, Blue) διαθέτει 4 ακροδέκτες, έναν για κάθε χρώμα και έναν κοινής καθόδου. Στο εσωτερικό του κρύβει τρία led, ένα για το κάθε χρώμα, τα οποία μπορούν να συνδυαστούν για να δώσουν ένα μεγάλο φάσμα χρωμάτων στον χρήστη. Ουσιαστικά πρόκειται για 3 σε 1 led.

Η μορφή του κυκλώματος θα είναι η εξής:



Παράδειγμα 6: RGB Led

ΚΩΔΙΚΑΣ

```
intredPin=11;
intgreenPin=10;
intbluePin=9;

void setup(){
pinMode(11,OUTPUT);
pinMode(10,OUTPUT);
pinMode(9,OUTPUT);
}

void loop(){

analogWrite(11,255); //το λαμπάκι θα ανάψει με το κόκκινο χρώμα
analogeWrite(10,0);
analogeWrite(9,0);

analogWrite(11,0);  //το λαμπάκι θα ανάψει με το πράσινο χρώμα
analogeWrite(10,255);
analogeWrite(9,0);

analogWrite(11,0);     //το λαμπάκι θα ανάψει με το μπλε χρώμα
analogeWrite(10,0);
analogeWrite(9,255);
delay(2000);
}
```

Διάφορα παραδείγματα με τη χρήση των βασικών δομών επανάληψης

- **do while():** σε αυτήν τη δομή πρώτα εκτελείτε ο κώδικας και μετά ελέγχεται εάν ικανοποιεί την συνθήκη, αν ναι τότε γίνεται επανάληψη εάν όχι τότε σταματάει. Ένα παράδειγμα είναι το παρακάτω:

```
int a = 0; // an integer
do
{
delay(1000);
a = a + 1;
} while ( a < 100 );
```

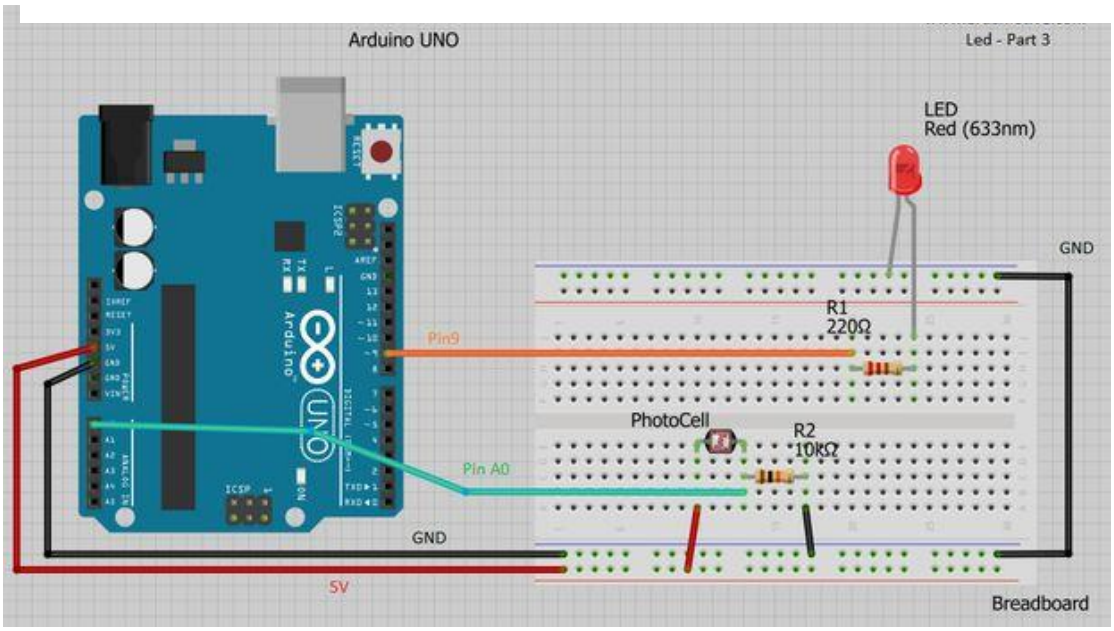
- **for():** σε αυτή τη δομή μπορούμε να επαναλάβουμε ένα σύνολο εντολών είτε μετρώντας τις επαναλήψεις είτε ελέγχοντας κάθε φορά μια συνθήκη. Ένα παράδειγμα μπορεί να είναι το εξής (με χρήση πίνακα):

```
void loop(){
int x = 1;
for (int i = 0; i> -1; i = i + x){
analogWrite(Pumping, i);
if (i == 255) x = -1;
delay(10);
}
}
```

- **if()**

  Ένα καλό παράδειγμα για τη χρήση της δομής επανάληψης σε κώδικα arduinομπορεί να είναι η χρήση φωτοαντήστασης.

Παράδειγμα 7: Photoresistor_Led




```
constintpResistor = A0;
constintledPin=9;

int value;
photoresistor (0-1023)
```

```
void setup(){

pinMode(ledPin, OUTPUT);
pinMode(pResistor, INPUT);

}

void loop(){
 value = analogRead(pResistor);

if (value > 25){
digitalWrite(ledPin, LOW);  //Turn led off
}
else{
digitalWrite(ledPin, HIGH); //Turn led on
}
delay(500);
}
```

Κεφάλαιο 4 – Έξυπνο Θερμοκήπιο

Τα υλικά που θα χρειαστούμε για τη δημιουργία του κυκλώματός μας θα είναι τα εξής:

Genuine uno

Breadboard

Καλώδια για το κύκλωμά μας

1(x) κόκκινο led

1(x) αντίσταση 220 Ω

1(x) fan 180mm 12V

1(x) μπαταρία 9V

1(x) τρανζιστοράκι NPN TR 2222A

1(x) δίοδο (e.g. IN4001)

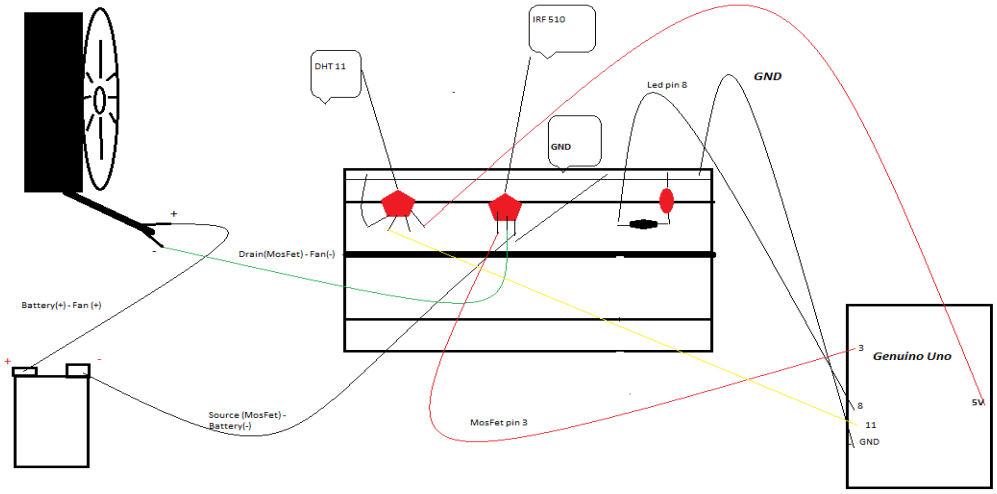
2(x) κομμάτια εύκαμπτου σωλήνα - Silicone Tube Transparent 2x4mm

1(x) Μοτερ που εξάγει αέρα- air pump motor (6V)

Αισθητήρας μέτρησης υγρασίας εδάφους (ground humidity)

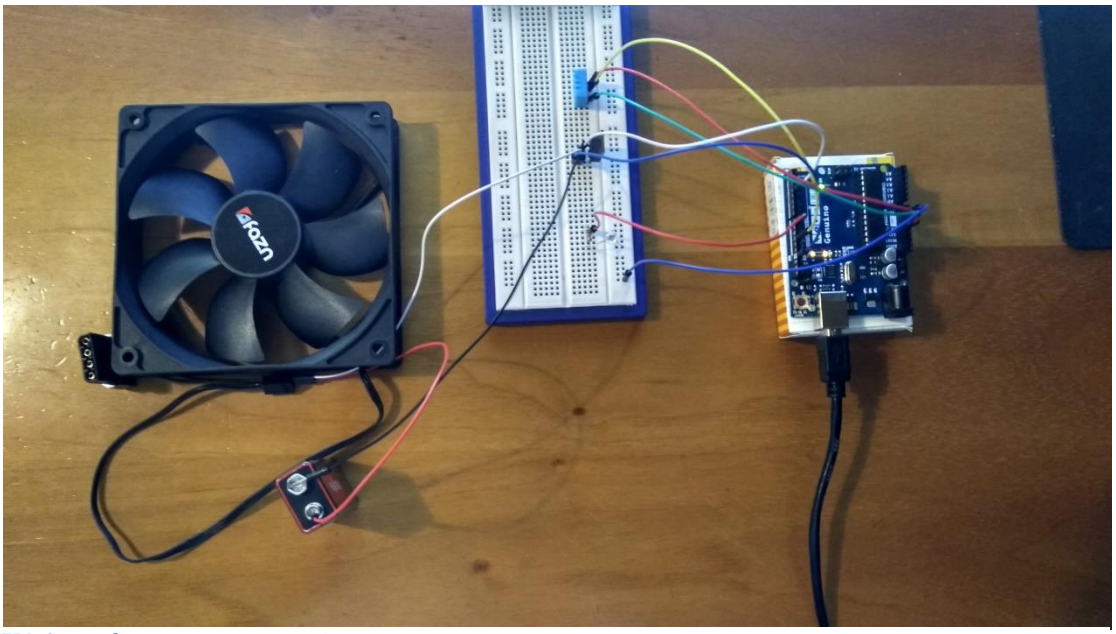
Αισθητήρας μέτρησης θερμοκρασίας και υγρασίας περιβάλλοντος (DHT 11)

4.1 Κώδικας και σχηματική αναπαράσταση του κυκλώματος, για τον αισθητήρα θερμοκρασίας και υγρασίας περιβάλλοντος:



Σχηματική αναπαράσταση 1

Πρώτα όσον αφορά τη δημιουργία του κώδικα, θα πρέπει να εισάγουμε την βιβλιοθήκη για τον αισθητήρα DHT 11 αφού την έχουμε κατεβάσει και συμπεριλάβει στην arduino IDE. Η παραπάνω διαδικασία έχει προαναφερθεί σε προηγουμενο κεφάλαιο της εργασίας.



Κύκλωμα fan_DHT 11_IRF 510_Battery

Αναλυτικότερα:

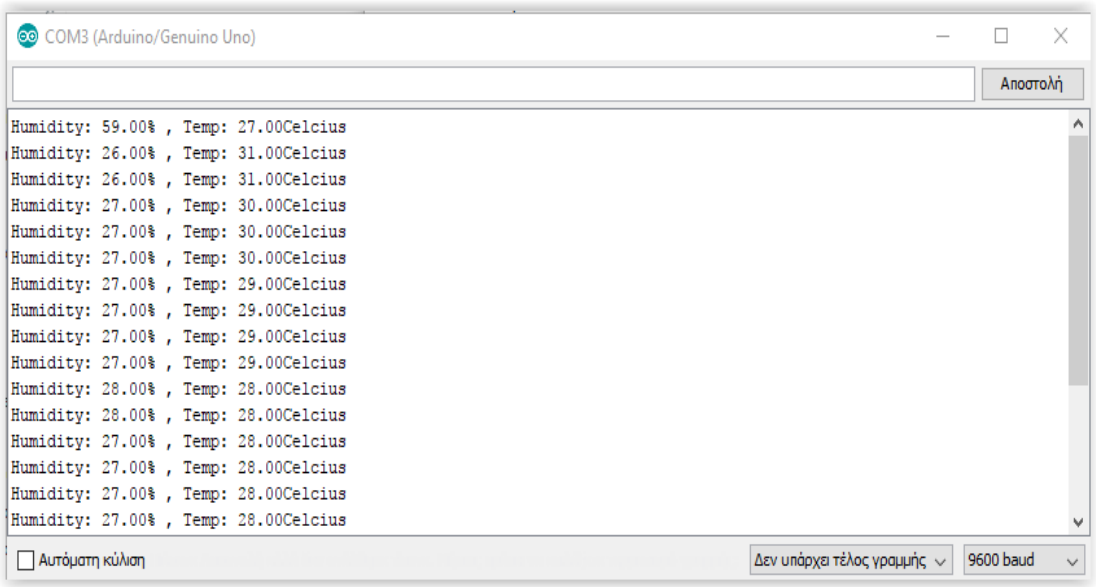
Επεξήγηση κώδικα και αποτελεσμάτων σειριακής οθόνης:

Αρχικά για τον κώδικα :

```
#include <dht.h> /*Εισαγωγή βιβλιοθήκης για να μπορέσουμε να χρησιμοποιήσουμε τον αισθητήρα μας*/

dht DHT;

#define DHT11_PIN 2 /*Σε ποιό pin του arduino θα συνδεθεί ο αισθητήρας */


const int Gate=3; /*Δηλώνουμε σε ποιό pin θα συνδεθεί το fan*/

const int led=4; /*Δηλώνουμε σε ποιό pin θα συνδεθεί το led*/


float hum; /*Αποθηκεύει την τιμή της υγρασίας για τον αισθητήρα*/

float temp; /*Αποθηκεύει την τιμή της θερμοκρασίας για τον αισθητήρα*/


void setup() {

  Serial.begin(9600); /*Για να εμφανίζονται τα αποτελέσματα απο τις τιμες που θέλουμε στην σειριακή οθόνη του υπολογιστή*/


  pinMode(2,OUTPUT);


  pinMode(3,OUTPUT);


  pinMode(4,OUTPUT); /*καταχωρούμε ξεχωριστά τον αισθητήρα, το led και το fan ως εξόδους*/


  digitalWrite(3,LOW); /*Το fan θα είναι κλειστό*/
```

```
}

void loop() {
  int chk=DHT.read11(DHT11_PIN);

  hum=DHT.humidity;
  temp=DHT.temperature; /*Διαβάζει τα δεδομένα και τα αποθηκεύει για τις μεταβλητές temp και hum*/

  Serial.print("Humidity: ");
  Serial.print(hum);
  Serial.print("% , Temp: ");
  Serial.print(temp);
  Serial.println("Celcius"); /*Εκτυπώνουμε τις τιμές των μεταβλητών που έχουμε ορίσει temp και hum αντίστοιχα στη σειριακή
 οθόνη του υπολογιστή  μας */

  delay(2000); /*αναμονή για 2000 milleseconds (2 δευτερόλεπτα)*/

  if (temp>=30){ /* Με την if ορίζουμε πως εαν η θερμοκρασία που εκτυπωθεί είναι μεγαλύτερη ή ίση με το 30... */
    digitalWrite(4,HIGH);  / *Το κόκκινο led θα ανάψει ως προειδοποίηση...*/
    delay(1000);  /*θα περιμένουμε 1 δευτερόλεπτο... */
    digitalWrite(3,HIGH);  /*...και θα τεθεί σε λειτουργία το fan προκειμένου να πέσει πάλι η θερμοκρασία. */
    }
   else if(temp<=28){  /*Συνεχίζοντας με την else if όταν η θερμοκρασία πέσει στους 28 βαθμούς ή πιο κάτω...*/
    digitalWrite(4,LOW);  /*το led θα σβήσει... */
    delay(1000);  /*και τέλος μετά απο ένα δεύτερο...*/
    digitalWrite(3,LOW); /*θα σταματήσει και το fan*/
```

```
    }
}
```

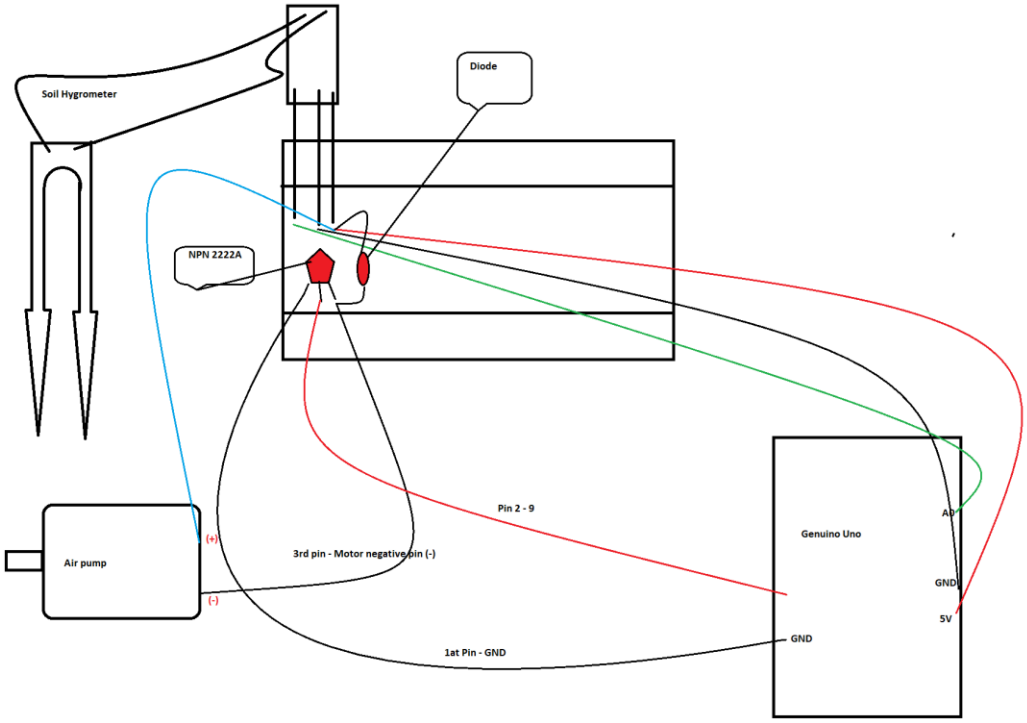


Αποτελέσματα σειριακής οθόνης

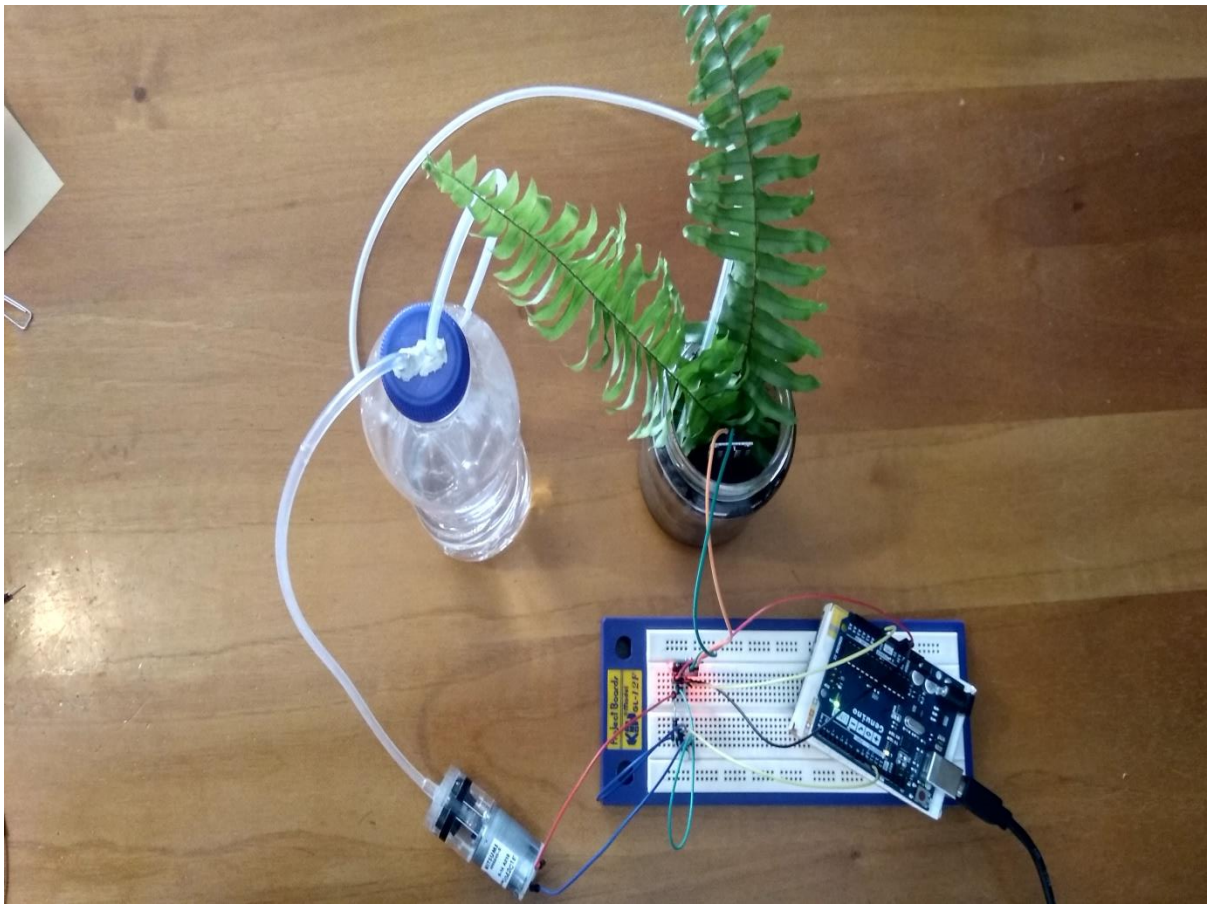
Αποτελέσματα σειριακής οθόνης παράδειγμα:

Αρχικά στους 27 βαθμούς το λαμπάκι και το fan θα παραμείνουν κλειστά αφού όλα λειτουργούν σωστά. Στην επόμενη μέτρηση όμως όπου η θερμοκρασία βρίσκεται στους 31 βαθμούς το led και το fan θα τεθούν σε λειτουργία προκειμένου να επαναφερθεί η θερμοκρασία.

4.2 Κώδικας και σχηματική αναπαράσταση αισθητήρα μέτρησης υγρασίας εδάφους (Ground humidity):



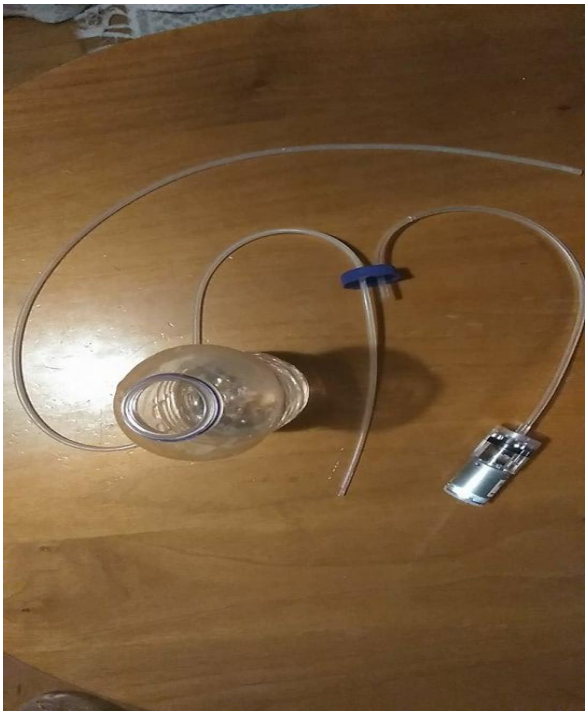
Σχηματική αναπαράσταση 2



Κύκλωμα led_Airpump_Hydro Sensor

Αρχικά για την ολοκλήρωση του κυκλώματος μας θα χρειαστεί να κατασκευάσουμε ένα δοχείο όπου θα έχουμε αποθηκευμένο νερό για να ποτίζουμε όταν το ρυθμίσουμε. Χρησιμοποιώντας ένα μπουκάλι για δοχείο περνάμε τα δύο κομμάτια του εύκαμπτου σωλήνα το ένα το συνδέουμε με το μοτεράκι που θα βγάζει αέρα και το άλλο για να περνάει το νερό στο χώμα.

Airpump 1

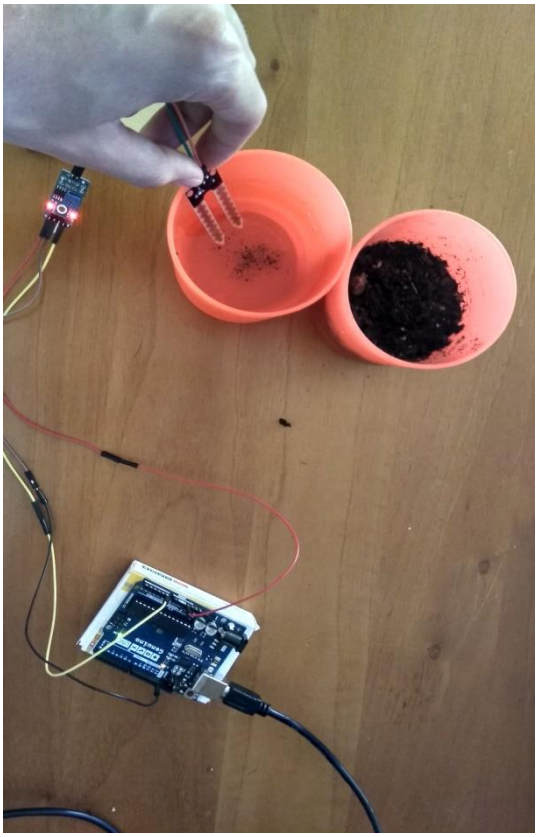


Airpump 2

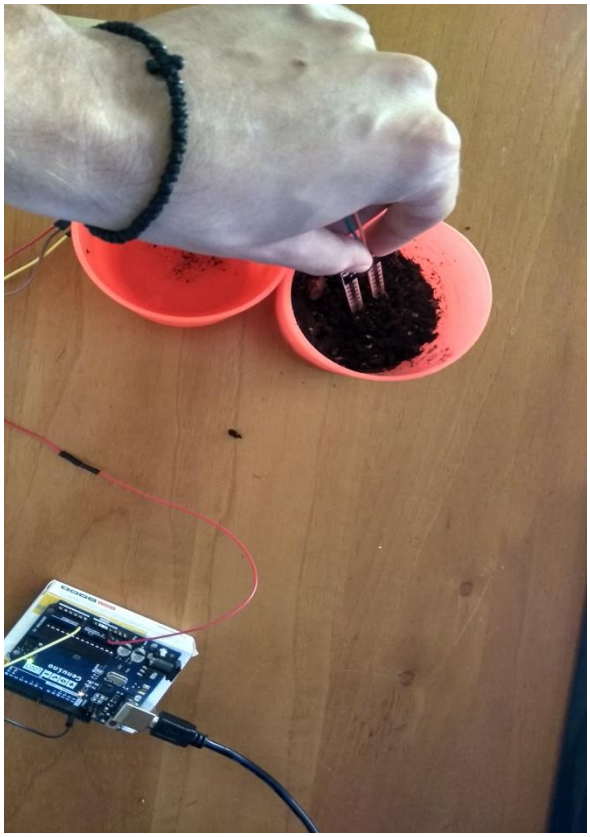
Παρακάτω ο κώδικας:

```
const int hygrometer=A0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  int sensorValue=analogRead(A0);
  Serial.println(sensorValue);
  delay(100);
}
```

Πρώτα θα τρέξουμε το παρακάτω πρόγραμμα για να δούμε τις τιμές που παίρνει ο αισθητήρας όταν βρίσκεται σε χώμα χωρίς υγρασία και όταν βρίσκεται στο νερό ή σε χώμα με υγρασία.

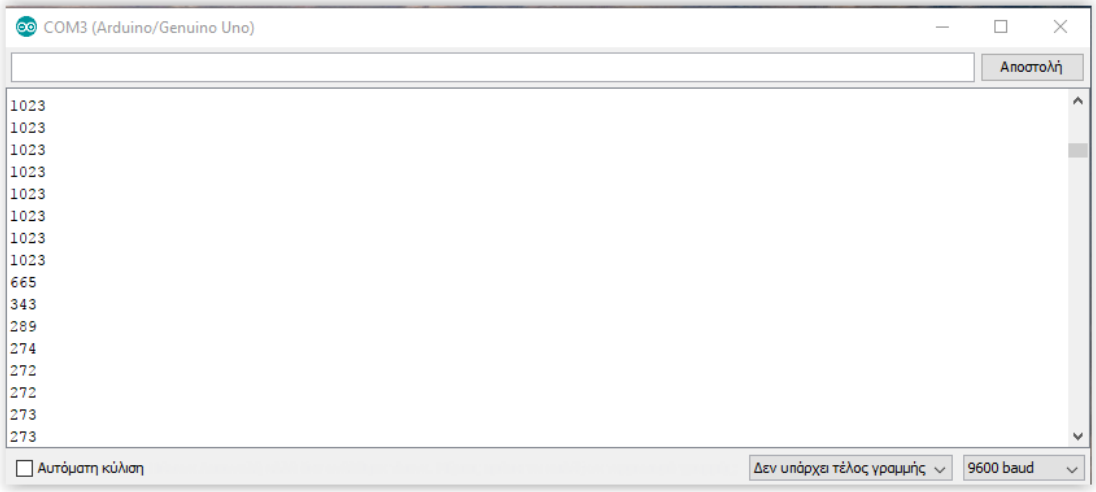


Hydro sensor 1



Hydro sensor 2

Αντίστοιχα και οι τμές:

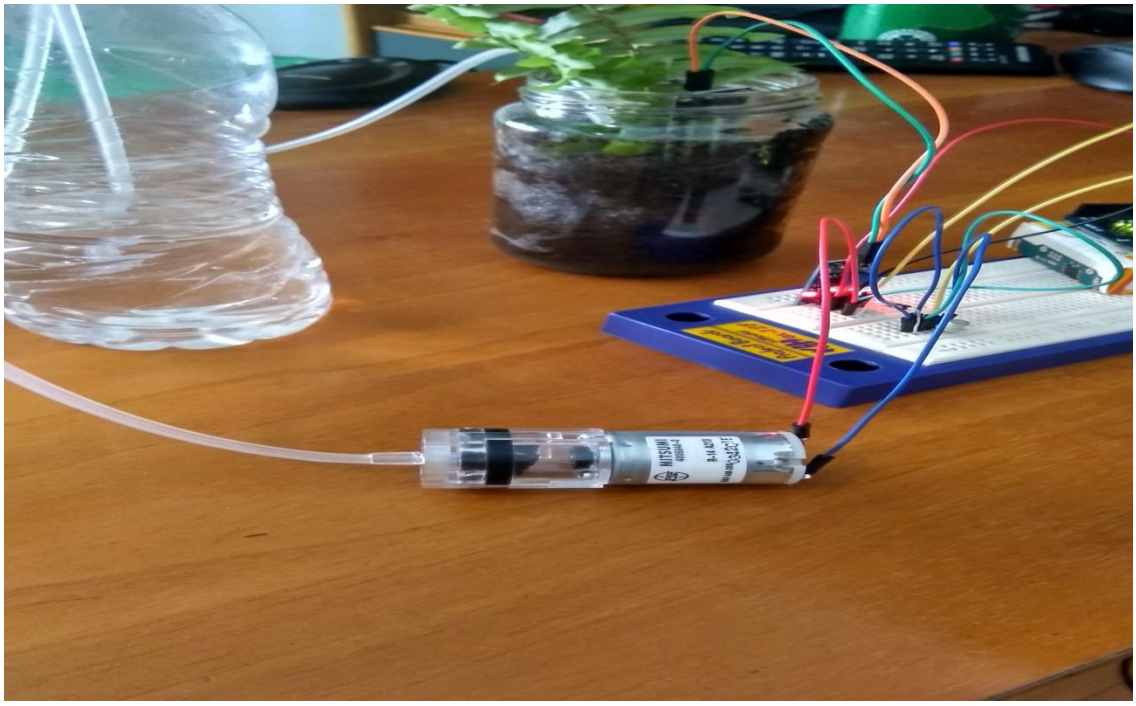


Αποτελέσματα Hydro sensor 1

Τώρα αφού ξέρουμε τις τιμές που παίρνει ο αισθητήρας θα προχωρήσουμε στην δημιουργία του κανονικού κώδικά μας:

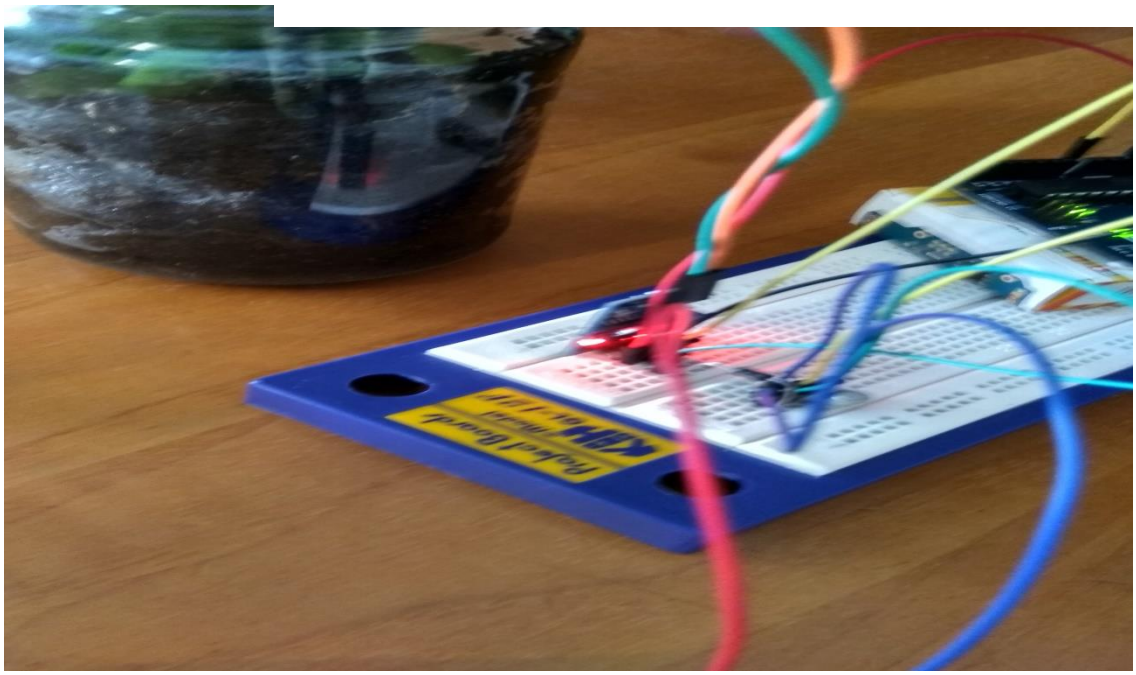
```
const int airPump = 9;  //Σύνδεση του μοτέρ στο pin 9 του arduino

const int hygrometer = A0;  //Σύνδεση του αισθητήρα με την αναλογική θύρα του arduino

int value;  //Μεταβλητή value όπου θα αφορά την τιμή της υγρασίας του εδάφους

void setup(){

  Serial.begin(9600); //Για να εμφανίζονται τα αποτελέσματα απο τις τιμες που θέλουμε στην σειριακή οθόνη του υπολογιστή

  pinMode(airPump,OUTPUT); //Καταχωρούμε το μοτέρ ώς έξοδο

}


void loop(){

  /* Σύμφωνα με τον προηγούμενο πίνακα που βρήκαμε απο τον παραπάνω κώδικα όταν το χώμα είναι υγρό, θα κρατήσουμε την τιμή 400 αλλά μπορούμε και να την αλλάξουμε.*/


  value = analogRead(hygrometer); //Διαβάζει την αναλογική τιμή

  value = constrain(value,400,1023); //Κρατάει τις βασικές τιμές 400 για υγρό χώμα και 1023 για ξυρό χώμα

  value = map(value,400,1023,100,0);  /*Με την map μετατρέπουμε το 400 σε 100 και το 1023 σε 0 καθώς μπορεί να δεχτεί τιμές απο 0 έως 255*/

  Serial.print("Soil humidity: ");
```

```
  Serial.print(value);

  Serial.println('%');

  delay(2000);  //Εκτυπώνουμε τις τιμές της υγρασίας του χώματος μετά το μήνυμα που έχουμε θέσει "Soil humidity: " με ρυθμό ανανέωσης 2 δευτερολέπτων


  if (value<20){ /*Με την if ορίζουμε εαν η τιμή της υγρασίας του χώματος πέσει κάτω απο το 30 ...*/

    Serial.print("Low soil humidity!!!"); /*... θα εμφανιστεί το μήνυμα "Low soil humidity!!!"... */

    digitalWrite(airPump,HIGH); /*και θα τεθεί σε λειτουργία το μοτέρ αέρα που έχουμε συνδέσει  στο κύκλωμα μας προκειμένου να τροφοδοτηθεί με νερό το χώμα μέσω του δοχείου*/

  }

  else if(value>20) /*Συνεχίζοντας με την else if όταν η τιμή της υγρασίας ανέβει πάνω απο το 20... */

  {

    Serial.print("The soil humidity it's ok!!!"); /*θα εμφανιστεί το μήνυμα "The soil humidity it's ok!!!"....*/

    digitalWrite(airPump,LOW); /*και θα κλείσει το μοτέρ μας.*/

}
```



Airpump_Hydro sensor

Soil humidity Led

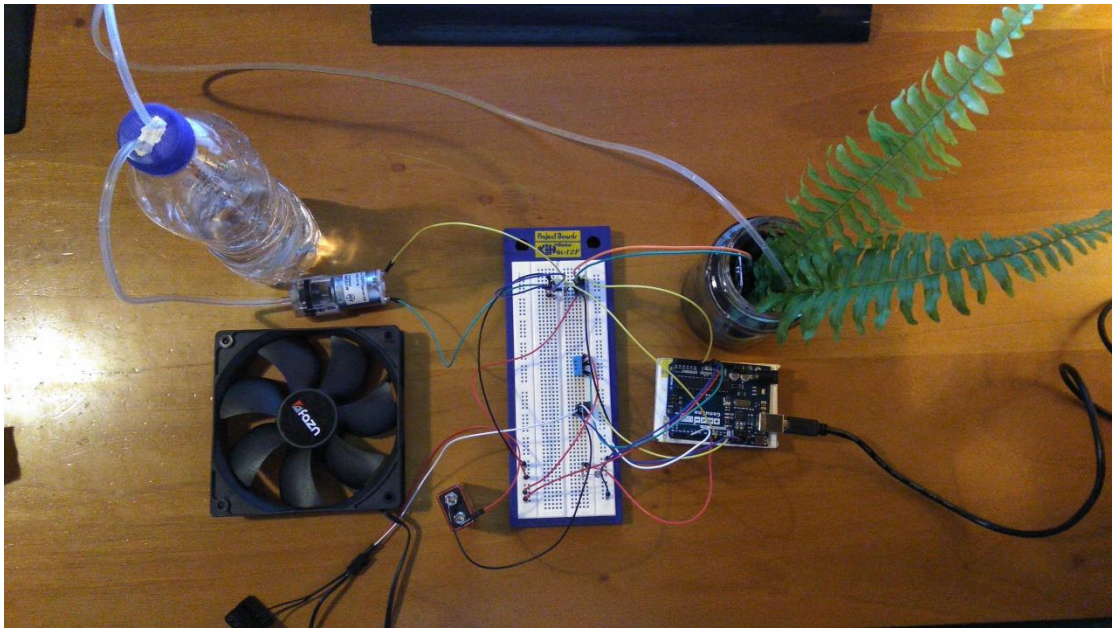


4.3

Υλοποίηση έξυπνου θερμοκηπίου

Για την δημιουργία του έξυπνου θερμοκηπίου θα χρησιμοποιήσουμε τα προηγούμενα κυκλώματά μας συνδέοντας τα και δημιουργώντας ένα κύκλωμα για το θερμοκήπιό μας. Θα συνδέσουμε βέβαια και τον κώδικα κάθε κυκλώματος σε ένα.

Με τη λέξη έξυπνο εννοούμε ότι το θερμοκήπιο θα μπορεί να δρά μόνο του (αυτοματοποιημένα). Συγκεκριμένα θα ελέγχει θερμοκρασία και υγρασία περιβάλλοντος με τον αισθητήρα DHT 11 και την υγρασία του εδάφους με τον αντίστοιχο αισθητήρα. Αφού έχουμε τις τιμές απο τους αισθητήρες μας, το arduino θα δώσει την κατάλληλη εντολή σε κάθε περίπτωση, όπως να δώσει ρεύμα στο led -για να μας προειδοποιήσει πως η θερμοκρασία του περιβάλλοντος είναι πάνω απο το όριο- και στο fan ώστε να βοηθήσει στην επαναφορά της θερμοκρασίας. Ακόμα θα μπορεί να ξεκινήσει η ροή αέρα μέσα στο δοχείο που έχουμε τοποθετήσει το νερό και μέσω του εύκαμπτου σωλήνα να περνάει το νερό που πιέζεται απο τον αέρα του μοτέρ και να περνά στο έδαφος, για το οποίο μας έχει ειδοποιήσει ο αισθητήρας πως έχει χαμηλότερη υγρασία απο το όριο που έχουμε θέσει.



Συνολικό Κύκλωμα

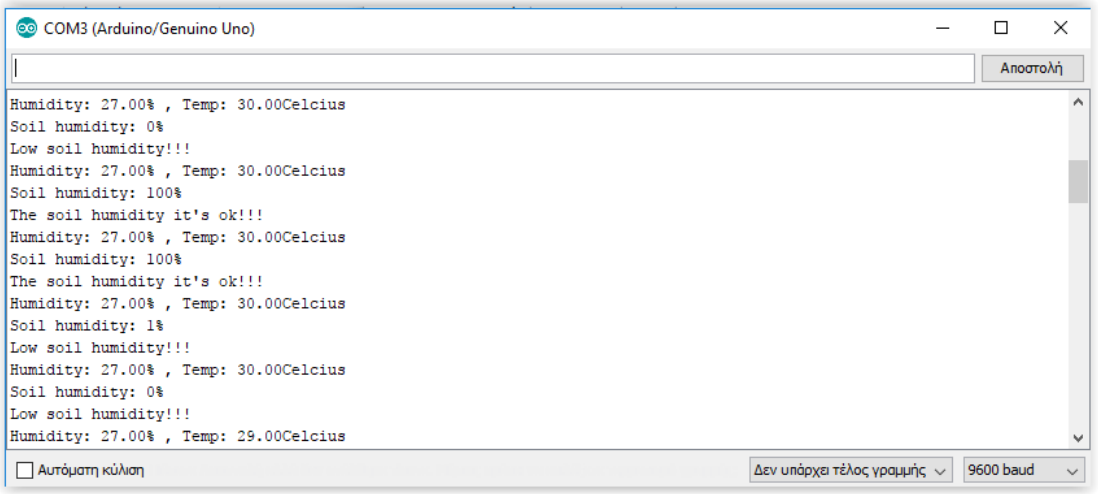
Παρακάτω ο κώδικας:

```
#include <dht.h>

dht DHT;

#define DHT11_PIN 2


const int Gate=3;

const int led=4; /* hum_temp_fan*/

const int airPump = 9;

const int hygrometer = A0;


int value;


float hum;

float temp;
```

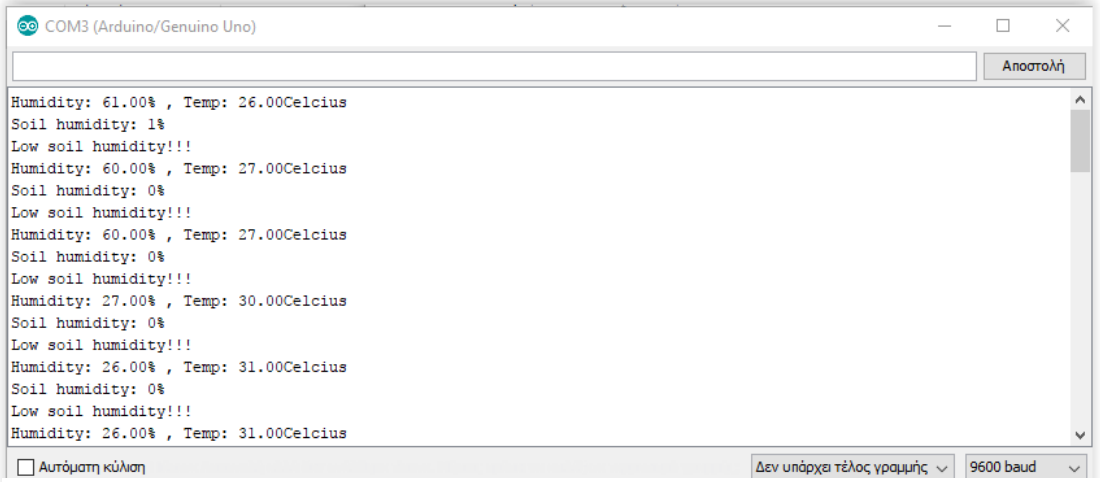
```
void setup() {
  Serial.begin(9600);
  pinMode(3,OUTPUT);
  pinMode(2,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(airPump,OUTPUT);

  digitalWrite(3,LOW);
}
void loop() {
  int chk=DHT.read11(DHT11_PIN);

  hum=DHT.humidity;
  temp=DHT.temperature;

  Serial.print("Humidity: ");
  Serial.print(hum);
  Serial.print("% , Temp: ");
  Serial.print(temp);
  Serial.println("Celcius");

  value = analogRead(hygrometer);
  value = constrain(value,400,1023);
  value = map(value,400,1023,100,0);

  Serial.print("Soil humidity: ");
  Serial.print(value);
  Serial.println('%');
```

```
  delay(2000);

if (temp>=30){ /* fan*/
    digitalWrite(4,HIGH);
    delay(1000);
    digitalWrite(3,HIGH);
    }
   else if(temp<=28){
    digitalWrite(4,LOW);
    delay(1000);
    digitalWrite(3,LOW);
    }

if (value<20){
    Serial.println("Low soil humidity!!!");
    digitalWrite(airPump,HIGH);

  }
  else if(value>20)
  {
    Serial.println("The soil humidity it's ok!!!");
    digitalWrite(airPump,LOW);
    }


}
```

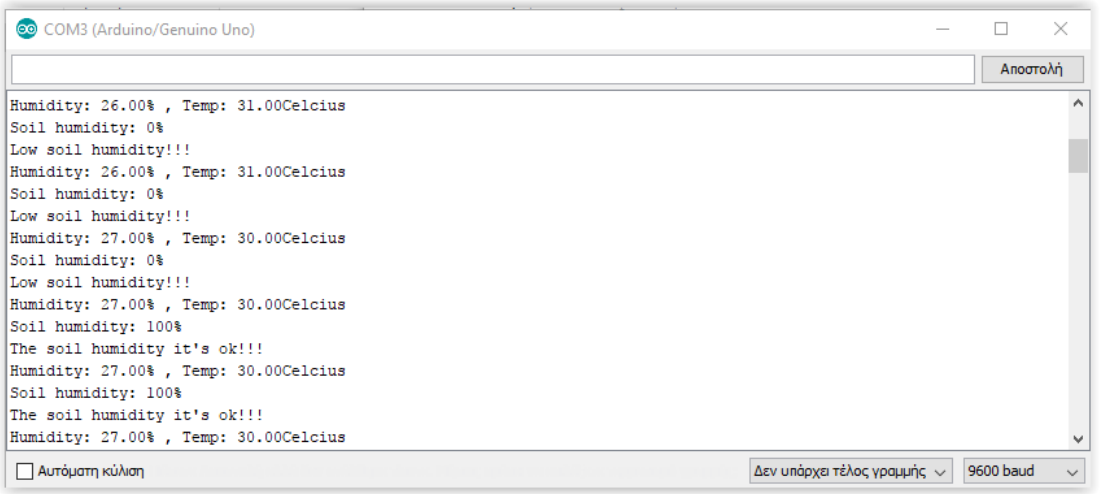
Και οι τιμές που εμφανίζονται στην σειριακή οθόνη του υπολογιτή μας:



Αποτελέσματα 1



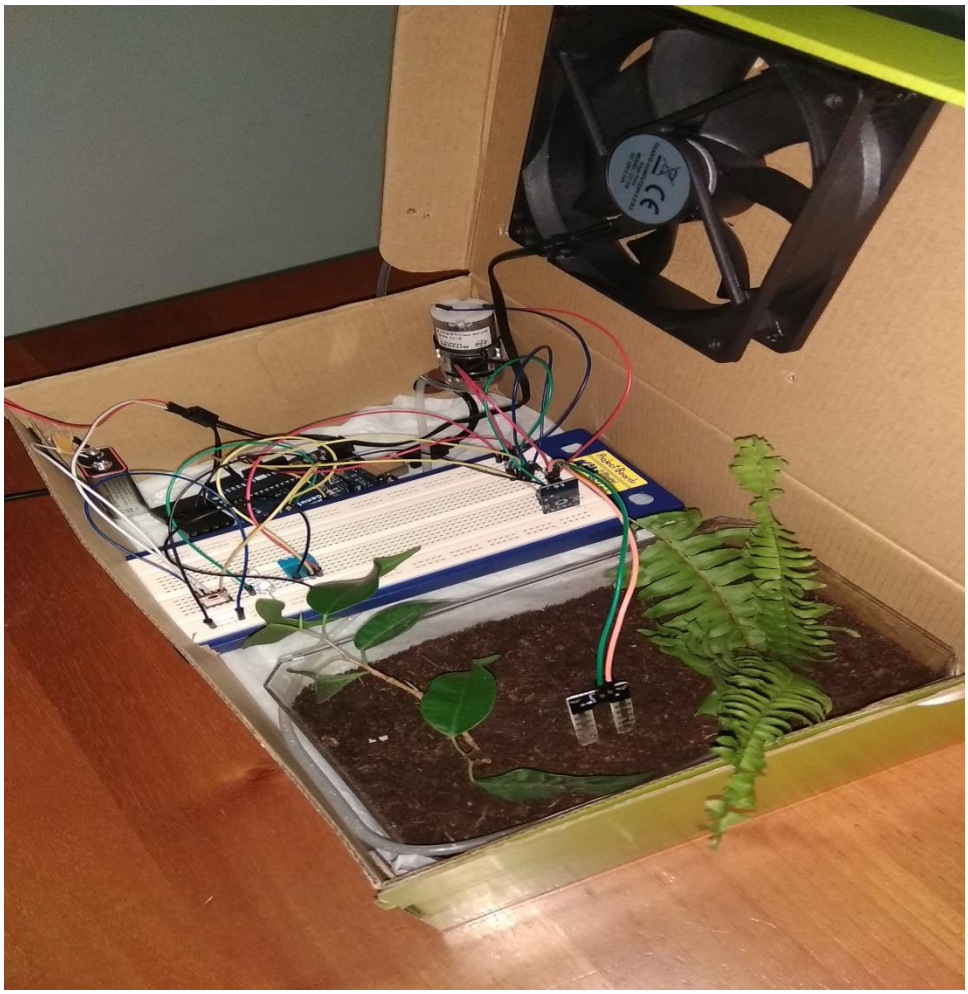
Αποτελέσματα 2



Αποτελέσματα 3

Αναλυτικότερα:

4.3.1 Χρήση του θερμοκηπίου στην πράξη

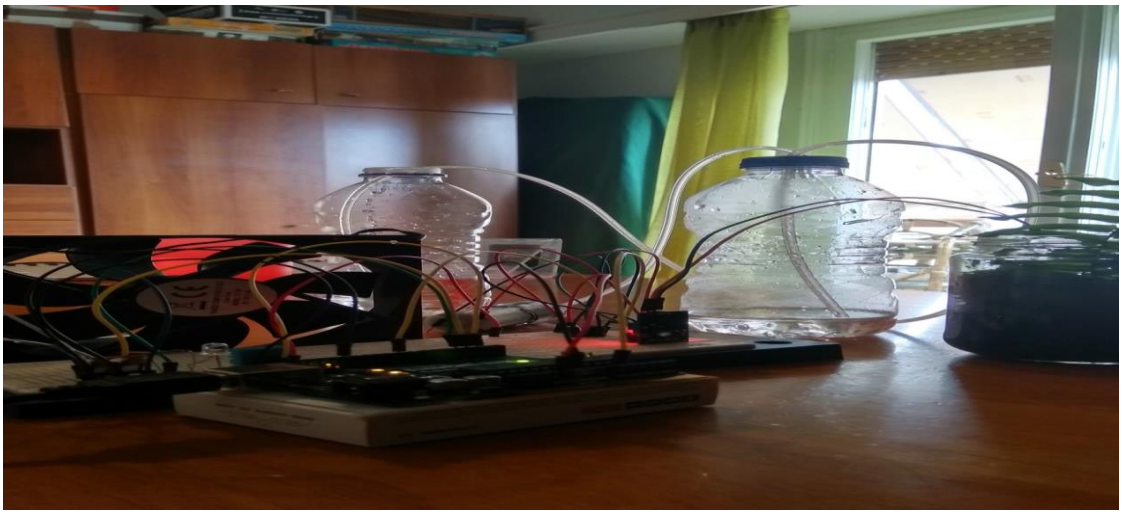


Έξυπνο θερμοκήπιο

Η διαδικασία έχει ώς εξής:

Αφού το arduino δεχτεί τις τιμές θερμοκρασίας και υγρασίας του χώρου και την υγρασία του χώματος απο τον κάθε αισθητήρα, ελέγχει εάν οι τιμές αυτές ξεπερνάνε τις τιμές που έχουμε θέσει με τις δομές επανάληψης και δρα ανάλογα με την κατάσταση. Εάν για παράδειγμα η τιμή της θερμοκρασίας του θερμοκηπίου είναι μεγαλύτερη του 30 τότε θα ανάψει το κόκκινο led για να μας προειδοποιήσει και θα ξεκινήσει το fan προκειμένου να επαναφέρει την θερμοκρασία στα επιτρεπτά επίπεδα που έχουμε ορίσει. Ακόμη στην περίπτωση που η

υγρασία του εδάφους σύμφωνα με τον αισθητήρα μας είναι χαμηλότερη απο το όριο που έχουμε θέσει τότε θα ξεκινήσει το μοτέρ αέρα προκειμένου να χορηγηθεί νερό στο έδαφος και να επαναφερθεί και αυτό στις τιμές που θέλουμε και έχουμε ορίσει. Όταν η υγρασία του εδάφους είναι εντάξει ο αισθητήρας μας έχει φροντίσει να μας ειδοποιήσει με ένα δεύτερο led που διαθέτει εκτός απο αυτό της ένδειξης λειτουργίας του



Έξυπνο θερμοκήπιο σε λειτουργία 1

Έξυπνο θερμοκήπιο 2

ΣΥΜΠΕΡΑΣΜΑΤΑ

Ώς συμπέρασμα απο τη συγκεκριμένη πτυχιακή εργασία έχουμε πρώτον ότι το ενδιαφέρον και η θέληση για να μάθει και να μπορεί να χρησιμοποιήσει ένα άτομο το arduino είτε για χόμπι είτε για επάγγελμα είναι εφικτό καθώς μιλάμε για ένα ευέλικτο και εύκολο στη χρήση υλικό και λογισμικό. Δεύτερον όσον αφορά το έξυπνο θερμοκήπιο που κατασκευάστηκε είχε ώς σκοπό την όσο λιγότερη γίνεται ανθρώπινη συμμετοχή σε ένα σύστημα το οποίο θα βοηθάει στην καλλιέργεια προϊόντων εξελίσσοντας τον τομέα της γεωργίας με τη χρήση αισθητήρων και αντικαθιστώντας ένα σημερινό θερμοκήπιο ως έχει, ώστε ο χρήστης να μπορεί να καλλιεργεί τα δικά του προϊόντα εξοικονομώντας έτσι και χρήματα που θα χρειάζονταν για να τα αγοράσει.

ΑΝΑΦΟΡΕΣ

Βιβλιογραφία:

- Random_Nerd_Tutorials_eBook_by_Rui_Santos
- Arduino Workshop A Hands On Introduction with 65 Projects
- Ανάπτυξη εφαρμογών με το Arduino
- Arduino For Dummies
- Arduino: 101 Beginners Guide: How to get started with Your Arduino (Tips, Tricks, Projects and More!)
- Arduino Programming in 24 Hours, Sams Teach Yourself
- Getting Started with Arduino: The Open Source Electronics Prototyping Platform (Make)
- Exploring Arduino: Tools and Techniques for Engineering Wizardry

Ιστότοποι:

- http://www.ardumotive.com/
- https://grobotronics.com/
- http://howtomechatronics.com/
- https://www.arduino.cc/
- http://fritzing.org/home/