

Σύστημα αυτόματου ποτίσματος τροφοδοτούμενου από δεξαμενή συλλογής νερού



2ο Γυμνάσιο Μοσχάτου

Σχολικό έτος 2022 - 23

Η κατανομή των υδατικών πόρων στην Ελλάδα

Η χώρα μας διαθέτει συνολικά επαρκείς υδατικούς πόρους, επιφανειακούς και υπόγειους, αλλά η γεωμορφολογική της σύνθεση, η γεωλογική δομή, η άνιση κατανομή των βροχοπτώσεων στο χώρο και στο χρόνο και τέλος η άνιση κατανομή των δραστηριοτήτων, και επομένως των υδατικών αναγκών, κυρίως στο χώρο, μειώνουν σημαντικά τη διαθέσιμη ποσότητα και δυσκολεύουν την αξιοποίηση.

Τα ιδιαίτερα χαρακτηριστικά της χώρας, σε σχέση με τους υδατικούς πόρους, από τα οποία προκύπτουν και οι δυσκολίες διαχείρισής τους, μπορούν να ταξινομηθούν ως εξής :

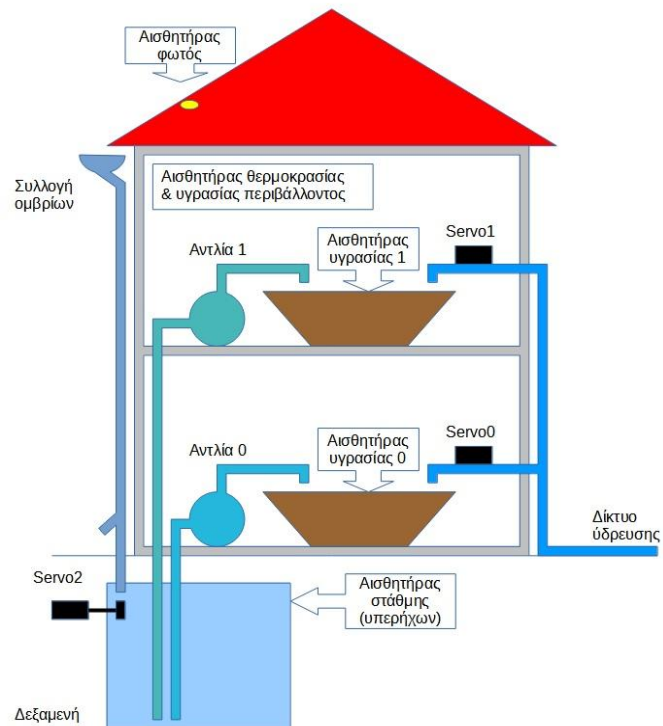
- Ως προς την κατανομή των βροχοπτώσεων, η οποία είναι άνιση ως προς το χώρο και το χρόνο. Συγκεκριμένα η Δυτική Ελλάδα δέχεται μεγαλύτερα ύψη βροχών από ότι η Ανατολική.
- Ως προς την κατανομή της ζήτησης, η οποία είναι άνιση ως προς το χώρο και ως προς το χρόνο.
- Ως προς τη μορφολογία της επιφάνειας της χώρας, με τη δημιουργία πολλών μικρών υδάτινων ρευμάτων.

Σχετικά με την κατανομή των βροχών παρατηρούνται σημαντικές διαφοροποιήσεις από περιοχή σε περιοχή. Έτσι, στα νότια και τα νησιά το φθινόπωρο και το χειμώνα έχουμε το μέγιστο της βροχής, ενώ το καλοκαίρι και μέρος της άνοιξης επικρατεί σχεδόν ξηρασία. Στα βόρεια η χρονική αυτή κατανομή αλλάζει σημαντικά, όπου φθινόπωρο-χειμώνας-άνοιξη έχουν σχεδόν ισοκατανομή της βροχής, με το καλοκαίρι να παραμένει σχεδόν ξερό. Είναι φυσικό, ότι κάτω από αυτές τις συνθήκες, για να γίνει δυνατή η ανάπτυξη των καλλιεργειών από τα μέσα της άνοιξης μέχρι το τέλος τουλάχιστον του καλοκαιριού, χρειάζεται η εφαρμογή αρδεύσεων.

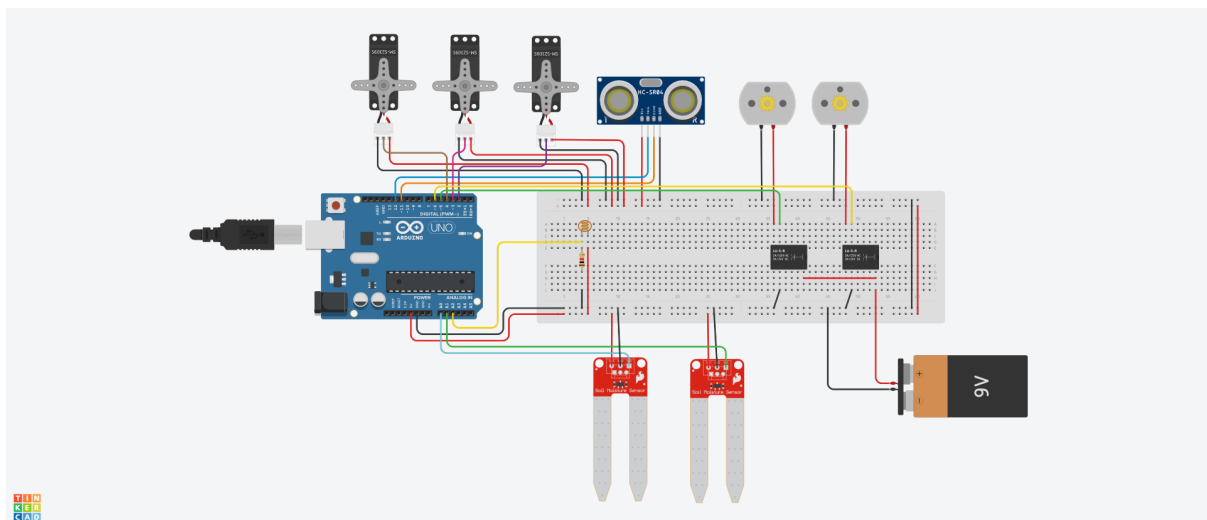
Περιγραφή του συστήματος αυτόματου ποτίσματος

Παραδοσιακά σε περιοχές που παρατηρείται μεγαλύτερη ξηρασία οι κάτοικοι συγκεντρώναν το νερό της βροχής σε στέρνες, όπως μας μαρτυρούν και λογοτεχνικές πηγές. Στο σύστημα αυτόματου ποτίσματος που σχεδιάσαμε συγκεντρώνουμε το νερό της βροχής σε δεξαμενή, από όπου ποτίζουμε τα φυτά που βρίσκονται σε δύο διαφορετικά παρτέρια.

Το σύστημά μας, ανιχνεύει την υγρασία του εδάφους σε δύο σημεία και ανάλογα με το ποσοστό υγρασίας ενεργοποιεί τις αντίστοιχες αντλίες για να τα ποτίσει. Η διαδικασία του ποτίσματος πραγματοποιείται τις βραδινές ώρες για να έχουμε μικρότερη εξάτμιση, επομένως να είναι πιο αποδοτική. Αν η δεξαμενή είναι άδεια τα παρτέρια ποτίζονται από το δίκτυο ύδρευσης.



Για την προσομοίωση της λειτουργίας και του προγραμματισμού των επιμέρους στοιχείων, καθώς και του πλήρους συστήματος, πριν την τελική συναρμολόγηση, χρησιμοποιήθηκε το ελεύθερο λογισμικό TinkerCAD. Στο παρακάτω σχέδιο βλέπουμε το κύκλωμα, με το οποίο έγινε η προσομοίωση στο TinkerCAD.



Στην τελική υλοποίηση τοποθετήθηκαν μονάδες ρελέ στη θέση των ρελέ που φαίνονται στο σχέδιο και ο αισθητήρας DHT11, τα οποία δεν υπάρχουν στο λογισμικό της προσομοίωσης.

Για την κατασκευή του συστήματος χρησιμοποιήσαμε:

Περιγραφή	Τεμάχια	Τιμή μονάδας	Κόστος
Arduino Uno SMD Compatible - CH340	1	9.90 €	9.90 €
Φωτοαντίσταση LDR 5mm	1	0.20 €	0.20 €
Αντίσταση 10KΩ	1	0,02 €	0,02 €
Αισθητήρα υγρασίας εδάφους	2	1,90 €	3,80 €
Αισθητήρα υπερήχων HC-SR04	1	2.50 €	2.50 €
Αισθητήρα υγρασίας και θερμοκρασίας DHT11	1	2,40 €	2,40 €
Servo micro MG90S	3	4,90 €	14,70 €
Αντλίες υγρών 5V	2	4,90 €	9,80 €
Σωλήνας σιλικόνης (6x8mm) μ	2	1,20 €	2,40 €
Relay module - 1 channel 5V high level trigger	2	1.90 €	3,80 €
Μπαταριοθήκη 4AA με καλώδια	1	0,60 €	0,60 €
Μπαταρίες AA (4τεμ)	1	1,80 €	1,80 €
Διακόπτης ON-ON	1	0,80 €	0,80 €
Διάτρητη πλακέτα σε σχήμα breadboard	1	3,80 €	3,80 €
Συνολικό κόστος			56,52 €

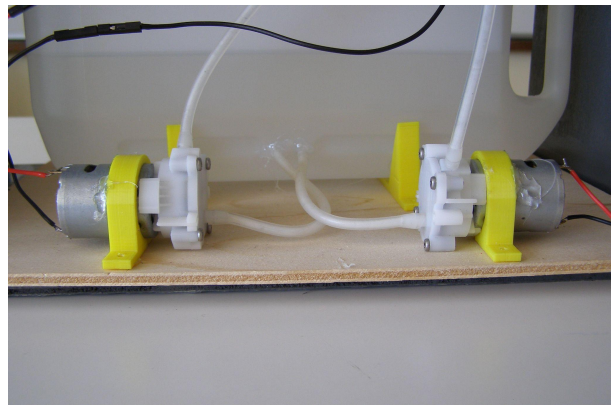
Στον παραπάνω πίνακα δεν περιλαμβάνεται το κόστος της καλωδίωσης, το οποίο εξαρτάται από το μέγεθος της μακέτας που θα κατασκευαστεί.

Κατασκευή του μοντέλου επίδειξης

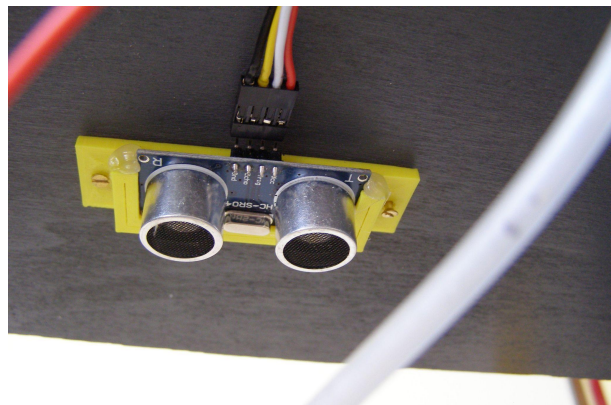
Για την επίδειξη της λειτουργίας του συστήματος κατασκευάσαμε ένα ξύλινο ομοίωμα σπιτιού με υπόγειο, ισόγειο και πρώτο όροφο. Το ομοίωμα βάφτηκε με λευκό χρώμα στους δύο ορόφους και γκρι στο υπόγειο.



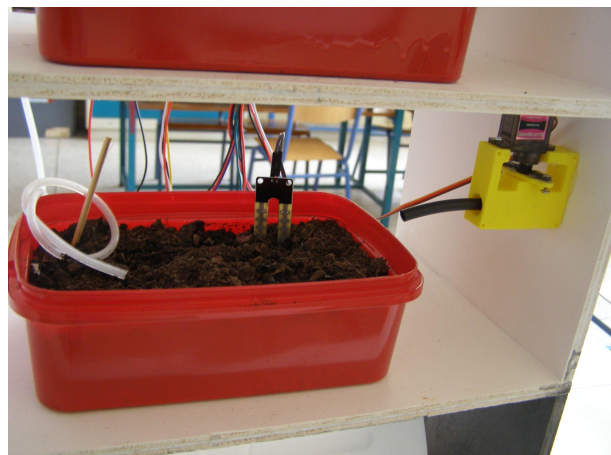
Στο υπόγειο του κτιρίου τοποθετήσαμε τις αντλίες του συστήματος και ένα πλαστικό δοχείο για δεξαμενή, στο οποίο ανοίξαμε τις υποδοχές για τις απαραίτητες σωληνώσεις.



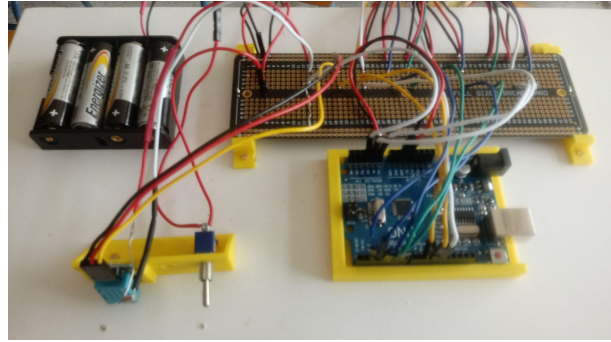
Ο αισθητήρας υπερήχων τοποθετήθηκε στην οροφή του υπογείου και ανιχνεύει το ύψος του νερού της δεξαμενής από το αντίστοιχο άνοιγμα στο δοχείο.



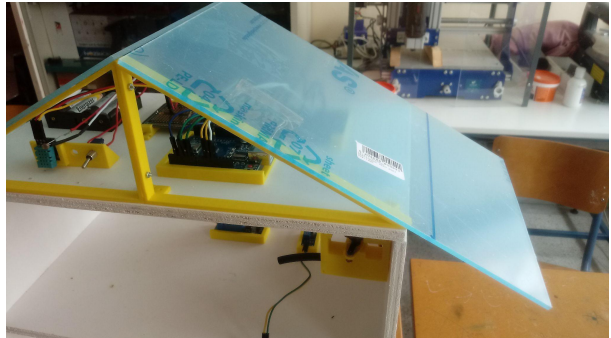
Στους ορόφους του κτιρίου τοποθετήσαμε πλαστικά δοχεία με χώμα για τα παρτέρια, όπου βάλαμε τους αισθητήρες υγρασίας εδάφους. Σε κάθε όροφο έχουμε και ένα διακόπτη για το νερό του δικτύου ύδρευσης ελεγχόμενο από servo.



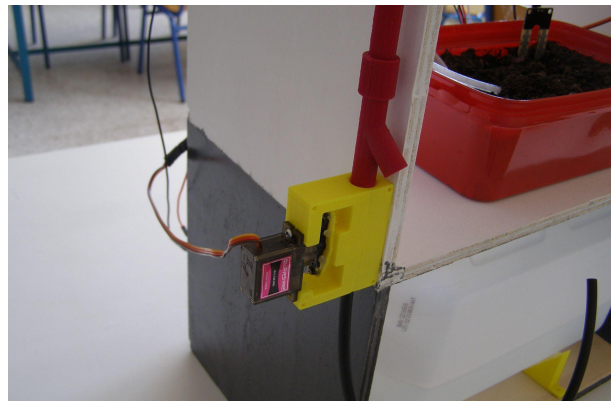
Στο πάνω μέρος του ομοιώματος τοποθετήσαμε την πλακέτα του Arduino Uno και την πλακέτα στην οποία καταλήγουν όλες οι καλωδιώσεις του συστήματος. Επίσης, τοποθετήσαμε τον αισθητήρα θερμοκρασίας-υγρασίας, τη φωτοαντίσταση και τις μπαταρίες τροφοδοσίας των αντλιών με το διακόπτη τους



Η σκεπή του ομοιώματος κατασκευάστηκε από διαφανές plexiglass.

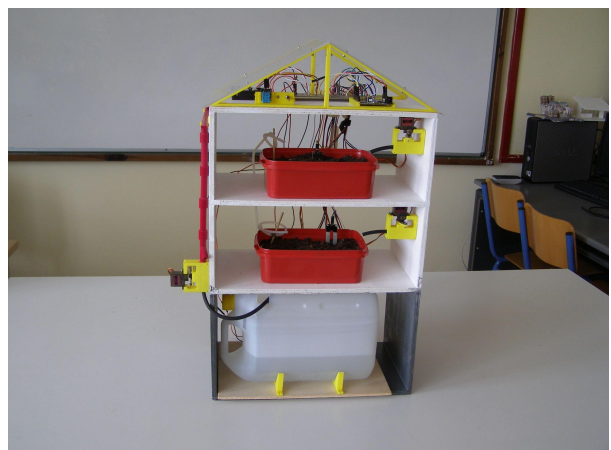


Στο εξωτερικό μέρος τοποθετήθηκε ομοίωμα της υδρορροής με ένα διακόπτη ελεγχόμενο από servo για τον έλεγχο της παροχής νερού στη δεξαμενή.



Οι βάσεις για τους αισθητήρες και τα υπόλοιπα στοιχεία του συστήματος, καθώς και η βάση της στέγης και η υδρορροή, σχεδιάστηκαν με το πρόγραμμα TinkerCAD και εκτυπώθηκαν με τον 3D printer, ενώ για τους διακόπτες νερού χρησιμοποιήθηκε ελεύθερο σχέδιο από την ιστοσελίδα Thingiverse.com (<https://www.thingiverse.com/thing:1624066>).

Η μακέτα του συστήματος ολοκληρωμένη φαίνεται στην παρακάτω φωτογραφία.



Προγραμματισμός του συστήματος αυτόματου ποτίσματος

Για τον προγραμματισμό του συστήματος χρησιμοποιήθηκε το λογισμικό Arduino IDE.

Ένα απλό πρόγραμμα που επαναλαμβάνεται στο Arduino αποτελείται από δύο συναρτήσεις:

- Τη συνάρτηση `setup()`, που εκτελείται μόνο μια φορά στην αρχή του προγράμματος, και καθορίζει τις αρχικές ρυθμίσεις του συστήματος.
- Τη συνάρτηση `loop()`, που περιέχει τον κυρίως κώδικα του προγράμματός μας, και εκτελείται κυκλικά μέχρι την απενεργοποίηση ή την επανεκκίνηση του Arduino.

Στο πρώτο μέρος του προγράμματός μας εισάγουμε τις απαραίτητες βιβλιοθήκες για την εκτέλεσή του και δημιουργούμε τα αντικείμενα των `servo` που χρειαζόμαστε, ορίζοντας και τις αρχικές τους θέσεις.

```
//Including libraries
#include <Servo.h>
#include <DHT.h>
```

```
//Creating servos and setting initial positions
Servo servo0;
Servo servo1;
Servo servo2;
int servo0Pos = 90;
int servo1Pos = 90;
int servo2Pos = 0;
```

Έπειτα ορίζουμε τους ακροδέκτες όπου θα συνδεθούν τα `servo`, τα ρελέ και οι αισθητήρες του κυκλώματος.

```
//Setting pins
const int servo0Pin=2;
const int servo1Pin=3;
const int servo2Pin=4;
const int relay0Pin=5;
const int relay1Pin=6;
const int dhtPin=10;
const int echoPin=11;
const int trigPin=12;
const int hygro0Pin=A0;
const int hygro1Pin=A1;
const int lightPin=A2;
```

Επίσης ορίζουμε το όνομα και τον τύπο των μεταβλητών, καθώς και τον τύπο του αισθητήρα DHT που θα χρησιμοποιήσουμε.

```
//Setting the variables
int hygro0;
int hygro1;
int light;
```

```
long duration;  
long distance;  
bool water0;  
bool water1;  
float hum;  
float temp;
```

```
//Setting the DHT sensor type  
#define dhtType DHT11  
DHT dht(dhtPin, dhtType);
```

Τέλος καθορίζουμε τις τιμές των μεταβλητών, όπου θα λαμβάνονται οι κρίσιμες αποφάσεις, ώστε, σε περίπτωση αλλαγής των ορίων λειτουργίας της δεξαμενής για παράδειγμα, να τροποποιούνται οι τιμές μόνο μια φορά και όχι σε όποιο σημείο του προγράμματος πρέπει να πάρουμε μια απόφαση.

```
//Setting the critical values  
const int minDistance=5;  
const int maxDistance=10;  
const int minLight=15;
```

Στην ενότητα setup, πρώτα ενεργοποιούμε την σειριακή θύρα του Arduino και τον αισθητήρα υγρασίας-θερμοκρασίας DHT11. Στη συνέχεια συνδέουμε τα Servo με τους αντίστοιχους ακροδέκτες και τα τοποθετούμε στην αρχική τους θέση. Τέλος καθορίζουμε τον τρόπο λειτουργίας των ψηφιακών ακροδεκτών που έχουμε ορίσει ότι θα χρησιμοποιήσουμε.

```
void setup() {  
  //Initialize serial communication and DHT sensor  
  Serial.begin(9600);  
  dht.begin();  
  
  //Attaching servos and writing initial positions  
  servo0.attach(servo0Pin);  
  servo1.attach(servo1Pin);  
  servo2.attach(servo2Pin);  
  servo0.write(servo0Pos);  
  servo1.write(servo1Pos);  
  servo2.write(servo2Pos);  
  
  //Initialize digital pins  
  pinMode(relay0Pin, OUTPUT);  
  pinMode(relay1Pin, OUTPUT);  
  pinMode(trigPin, OUTPUT);  
  pinMode(echoPin, INPUT);  
}
```


Στην ενότητα loop ξεκινάμε με την ανάγνωση των ενδείξεων από τους αισθητήρες του συστήματος, δηλαδή το ύψος της δεξαμενής νερού, την υγρασία του εδάφους στα παρτέρια και την φωτεινότητα.

```
void loop() {  
  //Read water level with ultrasonic sensor  
  digitalWrite(trigPin, LOW);  
  delayMicroseconds(2);  
  digitalWrite(trigPin, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigPin, LOW);  
  duration = pulseIn(echoPin, HIGH);  
  distance = (duration/2) / 29.1;  
  delay(100);  
  
  //Read moisture of plants at floor 0  
  hygro0 = analogRead(hygro0Pin);  
  hygro0 = constrain(hygro0,400,1023);  
  hygro0 = map(hygro0,400,1023,100,0);  
  delay(100);  
  
  //Read moisture of plants at floor 1  
  hygro1 = analogRead(hygro1Pin);  
  hygro1 = constrain(hygro1,400,1023);  
  hygro1 = map(hygro1,400,1023,100,0);  
  delay(100);  
  
  //Read light  
  light = analogRead(lightPin);  
  delay(100);  
  
  //Reading environment temperature and humidity  
  //Reading temperature or humidity takes about 250 milliseconds!  
  hum = dht.readHumidity();  
  temp = dht.readTemperature();  
  delay(250);
```

Ελέγχουμε στη συνέχεια το ύψος του νερού στη δεξαμενή και, στην περίπτωση που ανιχνεύσουμε το ελάχιστο όριο που έχουμε θέσει, διακόπτουμε την παροχή νερού από την υδρορροή, για να αποφύγουμε την υπερχειλίση της δεξαμενής.

```
  //Check tank overflow  
  if (distance<=minDistance){  
    servo2Pos=90;  
    servo2.write(servo2Pos);  
    delay(100);  
  }  
  else{
```

```

servo2Pos=0;
servo2.write(servo2Pos);
delay(100);
}

```

Μετά ελέγχουμε την υγρασία του πρώτου παρτεριού και καθορίζουμε αν τα φυτά χρειάζονται πότισμα.

```

//Check plants in floor 0
if (light<minLight){
    if (hygro0<30){
        water0=true;
    }
}
if (hygro0>90){
    water0=false;
}

```

Αν τα φυτά μας πρέπει να ποτιστούν, ελέγχουμε αν η δεξαμενή έχει νερό και ενεργοποιούμε την αντίστοιχη αντλία. Στην περίπτωση που η δεξαμενή δεν έχει νερό ανοίγουμε το δίκτυο ύδρευσης.

```

//Watering plants in floor 0
if (water0==true){
    if (distance<maxDistance){
        digitalWrite(relay0Pin, HIGH);
        delay(100);
        servo0Pos=90;
        servo0.write(servo0Pos);
        delay(100);
    }
    else{
        digitalWrite(relay0Pin, LOW);
        delay(100);
        servo0Pos=0;
        servo0.write(servo0Pos);
        delay(100);
    }
}

```

Επαναλαμβάνουμε τη ίδια διαδικασία και για το δεύτερο παρτέρι.

```

//Check plants in floor 1
if (light<minLight){
    if (hygro1<30){
        water1=true;
    }
}

```

```

if (hygro1>90){
    water1=false;
}

//Watering plants in floor 1
if (water1==true){
    if (distance<maxDistance){
        digitalWrite(relay1Pin, HIGH);
        delay(100);
        servo1Pos=90;
        servo1.write(servo1Pos);
        delay(100);
    }
    else{
        digitalWrite(relay1Pin, LOW);
        delay(100);
        servo1Pos=0;
        servo1.write(servo1Pos);
        delay(100);
    }
}

```

Όταν η υγρασία του εδάφους φτάσει στην τιμή που έχουμε καθορίσει, τότε το πότισμα στο αντίστοιχο παρτέρι διακόπτεται.

```

//Stop watering in floor 0
if (water0==false){
    digitalWrite(relay0Pin, LOW);
    delay(100);
    servo0Pos=90;
    servo0.write(servo0Pos);
    delay(100);
}

```

```

//Stop watering in floor 1
if (water1==false){
    digitalWrite(relay1Pin, LOW);
    delay(100);
    servo1Pos=90;
    servo1.write(servo1Pos);
    delay(100);
}

```

Στο τέλος της ενότητας οι ενδείξεις των αισθητήρων διαβιβάζονται στον συνδεδεμένο υπολογιστή μέσω της σειριακής θύρας του Arduino.

```

//Print out the values
Serial.print("Water depth ");

```

```
Serial.print(distance);  
Serial.println(" cm");  
Serial.print("Soil humidity 0: ");  
Serial.print(hygro0);  
Serial.println("%");  
Serial.print("Soil humidity 1: ");  
Serial.print(hygro1);  
Serial.println("%");  
Serial.print("Light: ");  
Serial.println(light);  
Serial.print("Environment Humidity: ");  
Serial.print(hum);  
Serial.println("%");  
Serial.print("Environment Temperature: ");  
Serial.print(temp);  
Serial.println("°C ");
```

Ακολουθεί μια παύση ενός δευτερολέπτου για το κλείσιμο της ενότητας και την εκτέλεσή της από την αρχή.

```
    delay(1000);  
}
```

Βιβλιογραφία

1. [Γ. Βούτσιнос, Γ. Καλκάνης, Κ. Κοσμάς, Κ. Σούτσας, “Γεωλογία & Διαχείριση Φυσικών Πόρων”, Ινστιτούτο Τεχνολογίας Υπολογιστών & Εκδόσεων «Διόφαντος»](#)
2. [Βασίλης Νούσης, “ARDUINO για αρχάριους”, 2019](#)
3. [Εμμανουήλ Πουλάκης, “Προγραμματίζοντας με τον μικροελεγκτή Arduino”, 2015](#)