



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ

ΘΕΜΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ:

«ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΑΥΤΟΜΑΤΗΣ ΓΛΑΣΤΡΑΣ ΦΥΤΩΝ ΕΛΕΓΧΟΜΕΝΗΣ  
ΜΕΣΩ ΔΙΑΔΙΚΤΥΟΥ»

ΟΝΟΜΑ ΦΟΙΤΗΤΗ:

ΓΙΑΝΝΗΣ ΣΟΦΙΑΝΟΣ



ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

ΑΒΡΑΑΜ ΧΑΤΖΟΠΟΥΛΟΣ

## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/Η κάτωθι υπογεγραμμένος/η .....του .....  
φοιτητής του Τμήματος ..... του  
Πανεπιστημίου Δυτικής Αττικής, πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου,  
δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα,  
όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή  
μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν  
λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου  
συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε, ο οποίος φέρει και την  
ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα, σε περίπτωση που το Ίδρυμα του έχει  
απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση  
του Τμήματος με νέα απόφασή της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου  
την εκπόνηση Π.Ε με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν  
λόγω Π.Ε πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την  
ημερομηνία ανάθεσής της.

**Ο Δηλών**

**Ημερομηνία**

## ΠΕΡΙΛΗΨΗ

Στόχος της παρούσας πτυχιακής είναι η υλοποίηση μιας αυτόματης γλάστρας η οποία θα πραγματοποιεί τις εξής λειτουργίες:

1. παρακολούθηση των συνθηκών διαβίωσης του φυτού, όπως θερμοκρασία, υγρασία, περιβάλλοντος και υγρασία εδάφους,
2. αποστολή των ανωτέρω δεδομένων στο διαδίκτυο έτσι ώστε να δίνεται η δυνατότητα απομακρυσμένης παρακολούθησης -μέσω κατάλληλα διαμορφωμένης ιστοσελίδας- από τον άνθρωπο,
3. αυτόματο έλεγχο της υγρασίας εδάφους του φυτού και αυτόματο πότισμα, ενεργοποιείται η αντλία νερού όταν η τιμή της υγρασίας κυμανθεί κάτω από το επιθυμητό όριο,
4. και δυνατότητα για απομακρυσμένο – χειροκίνητο έλεγχο του φωτισμού (λαμπτήρας LED) και του ποτιστικού (αντλία νερού) του φυτού.

Σκοπός αυτής της εργασίας είναι η διευκόλυνση έως και η απαλλαγή του ανθρώπου από την καθημερινή απασχόληση που απαιτείται για την συντήρηση του φυτού καθώς και η δημιουργία ιδανικών συνθηκών για την βέλτιστη ανάπτυξή του.

Για να επιτευχθεί η απομακρυσμένη παρακολούθηση των συνθηκών στις οποίες βρίσκεται το φυτό χρησιμοποιήσαμε το μικροελεγκτή Arduino Mega 2560 ως εξυπηρετητή (server). Πάνω στον μικροελεγκτή έχει τοποθετηθεί μία πλακέτα Ethernet shield η οποία δίνει την δυνατότητα στο Arduino να συνδέεται στο διαδίκτυο. Η πλακέτα Ethernet shield φέρει κατάλληλη υποδοχή για κάρτα μνήμης micro SD στην οποία έχουμε αποθηκεύσει τον κώδικα της ιστοσελίδας.

Η συλλογή των δεδομένων (θερμοκρασία, υγρασία) υλοποιείται με συμβατά προς το μικροελεγκτή μας αισθητήρια, τα οποία αποστέλλουν αρχικώς τα δεδομένα στον μικροελεγκτή και στη συνέχεια -μέσω του Ethernet shield και κατάλληλων διαμορφωμένων αρχείων xml- στην ιστοσελίδα.

Μελλοντικά θα μπορούσαμε να βελτιώσουμε το έργο με την τοποθέτηση κάμερας έτσι ώστε να έχουμε την δυνατότητα οπτικής επαφής με το φυτό έτσι ώστε να μπορούμε να διακρίνουμε συμπτώματα από τυχόν ασθένειες και παράσιτα οι οποίες παρουσιάζονται στον κορμό ή/και στα φύλλα του φυτού. Επιπλέον αν θέλαμε να πάμε ένα βήμα παρακάτω, θα μπορούσαμε να τοποθετήσουμε και έναν απομακρυσμένο χειροκίνητο μηχανισμό ραντίσματος που θα ψεκάζει το φυτό με το κατάλληλο φάρμακο για την καταπολέμηση των ασθενειών, τον έλεγχο εντόμων και τελικώς τη φυτοπροστασία.

## **ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ**

Arduino, Αυτόματη γλάστρα, IoT (Internet of Things).

## **ABSTRACT**

The aim of this paper is to implement an automatic pot which will perform the following functions:

1. monitoring the plant's living conditions, such as temperature, humidity, environment and soil moisture,
2. sending the above data on the Internet so that it is possible to remotely monitor - through a properly configured web page - human beings,
3. automatic control of plant soil humidity and automatic watering, activates the water pump when the humidity value is below the desired limit,
4. and possibility for remote - manual control of the lighting (LED bulb) and the irrigation (water pump) of the plant.

The purpose of this work is to facilitate absolve man from the daily employment wich required for the maintenance of the plant as well as the creation of ideal conditions for the optimal growth of the plant.

To achieve remote monitoring of plant conditions we used the Arduino Mega 2560 microcontroller as a server. An Ethernet shield has been placed on the microcontroller which enables Arduino to connect to the internet. In addition, there is a micro SD card slot on which we have stored the website code. To collect data such as temperature and humidity, we use sensors that are compatible with our microcontroller, which send the data to our microcontroller and from it sent via xml files to the web page.

In the future, we could improve the work by placing a camera so that we can visual contact with the plant so that we can discern symptoms of any illness that occurs on the trunk or leaves of the plant. In addition, if we wanted to go one step further, we could also put a sprinkle mechanism that will be triggered by a button from the website when it locates any disease in the plant so that the plant is sprinkled with a specific medicine to combat it .

# KEYWORDS

Arduino, Automated pot, IoT (Internet of Things).

## Περιεχόμενα

ΠΕΡΙΛΗΨΗ.....	iii
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ.....	iv
ABSTRACT .....	v
KEYWORDS .....	vi
ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ .....	viii
ΚΕΦΑΛΑΙΟ 1-Εισαγωγή.....	1
Ο μικροελεγκτής Arduino mega 2560.....	2
Βιβλιοθήκες .....	2
Απομακρυσμένη παρακολούθηση .....	2
ΚΕΦΑΛΑΙΟ 2 – Θεωρητικό υπόβαθρο .....	2
DHT 11 .....	3
Αισθητήριο υγρασίας εδάφους .....	4
ΚΕΦΑΛΑΙΟ 3 – Κατασκευή αυτόματης γλάστρας.....	6
3.1 Σχέδιο Γλάστρα.....	6
3.2 Ηλεκτρονικό κύκλωμα.....	9
3.3 Λίστα υλικών .....	10
ΚΕΦΑΛΑΙΟ 4 – Λογισμικό αυτόματης γλάστρας .....	11
4.1 Διάγραμμα ροής.....	11

4.2 Κώδικας Arduino .....	14
4.3 Βιβλιοθήκες Arduino .....	21
Τι είναι το Client Request και το Server Response. [9] .....	23
Server Response .....	24
4.4 Κώδικας Ιστοσελίδας.....	26
ΚΕΦΑΛΑΙΟ 5 – Επίλογος .....	36
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	37

## ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ

Εικόνα 1.DHT 11 .....	3
Εικόνα 2.DHT 11 Pins.....	4
Εικόνα 3. Αισθητήριο Υγρασίας Εδάφους.....	4
Εικόνα 4. Soil Moisture Sensor Pins .....	5
Εικόνα 5.Πρόσοψη γλάστρας.....	7
Εικόνα 6. Κάτοψη γλάστρας .....	8
Εικόνα 7. Ηλεκτρονικό κύκλωμα.....	9
Εικόνα 8.Βιβλιοθήκες.....	22
Εικόνα 9 - Web File and Address of IP, mac and ethernet port .....	23
Εικόνα 10- Δήλωση βιβλιοθήκης DHT 11 .....	25
Εικόνα 11.sketch .....	33
Εικόνα 12. κώδικας σελίδας.....	34
Εικόνα 13.Περιβάλλον Ιστοσελίδας.....	34
Εικόνα 15. Όργανο ένδειξης θερμοκρασίας.....	35
Εικόνα 14. Όργανο ένδειξης υγρασίας εδάφους .....	35



## ΚΕΦΑΛΑΙΟ 1-Εισαγωγή

Ο σκοπός αυτής της εργασίας είναι η δημιουργία μιας αυτοματοποιημένης γλάστρας η οποία ποτίζεται αυτόματα με βάση την υγρασία εδάφους της. Οι συνθήκες της θερμοκρασίας και υγρασίας του περιβάλλοντος καθώς και η υγρασία του χώματος της γλάστρας καταγράφονται από τα αισθητήρια, αποστέλλονται μέσω του διαδικτύου σε έναν server και εμφανίζονται στο πρόγραμμα πλοήγησης (browser) της φορητής συσκευής του χρήστη (πελάτη – client). Έτσι ο χρήστης θα έχει την δυνατότητα να παρακολουθεί τις συνθήκες που επικρατούν στο περιβάλλον του φυτού και να ελέγχει εάν το αυτόματο πότισμα λειτουργεί κανονικά.

Με αυτόν τον τρόπο λύνεται το πρόβλημα της απασχόλησης ανθρώπινου δυναμικού για την διαδικασία ενός απλού ποτίσματος. Έτσι ο μόνος λόγος που θα χρειάζεται η ανθρώπινη επίβλεψη είναι για τον εντοπισμό τυχόν ασθένειας του φυτού.

## **ΚΕΦΑΛΑΙΟ 2 – Θεωρητικό υπόβαθρο**

Στο παρόν κεφάλαιο εισάγουμε τον αναγνώστη στα απαραίτητα δομικά εξαρτήματα τα οποία αξιοποιήθηκαν για την ολοκλήρωση της παρούσης πτυχιακής εργασίας όπως :μικροελεγκτής Arduino και τα αισθητήρια θερμοκρασίας υγρασίας.

### **Μικροελεγκτής Arduino Mega 2560**

Το Arduino Mega 2560 είναι ένας μικροελεγκτής μονής πλακέτας βασισμένος στον ATmega 2560. Διαθέτει 54 ψηφιακές ακίδες εισόδου/εξόδου από τις οποίες οι 15 μπορούν να χρησιμοποιηθούν ως έξοδοι PWM, 16 αναλογικές εισοδοι, 4 UART (σειριακές θύρες υλικού), ταλαντωτή κρυστάλλου 16 MHZ, σύνδεση USB, υποδοχή τροφοδοσίας και ένα κουμπί επαναφοράς. Περιέχει όλα τα απαραίτητα για την υποστήριξη του μικροελεγκτή και μπορεί να ενεργοποιηθεί είτε συνδέοντάς το με ένα καλώδιο στον υπολογιστή είτε με μπαταρία [1].

### **Βιβλιοθήκες**

Το περιβάλλον του Arduino μπορεί να επεκταθεί μέσω της χρήσης βιβλιοθηκών, όπως και οι περισσότερες πλατφόρμες προγραμματισμού. Οι βιβλιοθήκες είναι μια συλλογή κωδικών που μας διευκολύνει να συνδεθούμε με έναν αισθητήρα, μία οθόνη, έναν κινητήρα και άλλα υλικά. Για παράδειγμα η ενσωματωμένη βιβλιοθήκη Liquid Crystal διευκολύνει την ομιλία σε οθόνες LCD χαρακτήρων. Στο διαδίκτυο υπάρχουν διαθέσιμες εκατοντάδες επιπλέον βιβλιοθήκες για λήψη, τις οποίες θα πρέπει να εγκαταστήσουμε στο Arduino προκειμένου να τις χρησιμοποιήσουμε. Στην εργασία αυτή χρησιμοποιήσαμε τις βιβλιοθήκες SPI.h, Ethernet2.h, SD.h, Adafruit\_sensor.h και DHT.h, την λειτουργία των οποίων αναλύουμε στο κεφάλαιο 5 [2] [3].

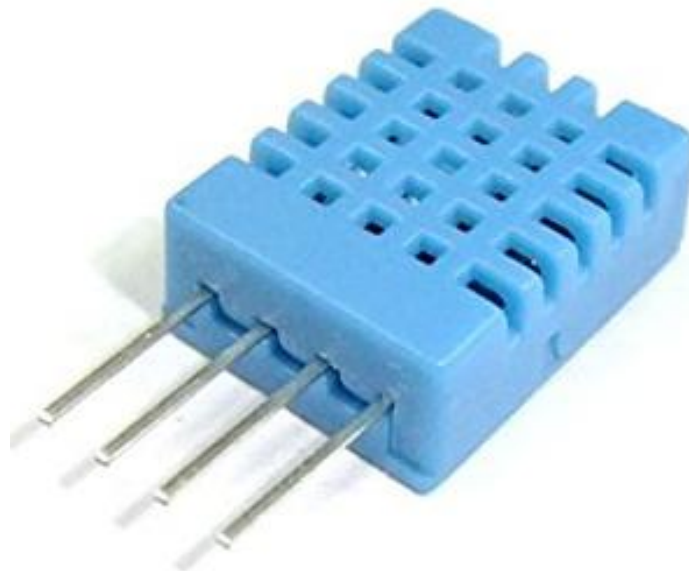
### **Απομακρυσμένη παρακολούθηση**

Για την επίτευξη της απομακρυσμένης παρακολούθησης των συνθηκών της γλάστρας υπέστη επιτακτική η ανάγκη σύνδεσης του μικροελεγκτή Arduino στο διαδίκτυο. Για αυτόν το λόγο

χρησιμοποιήσαμε την πλακέτα Ethernet Shield, η οποία εφαρμόζεται πάνω στις ακίδες εισόδου/εξόδου του Arduino. Η Ethernet Shield συνδέεται μέσω ενός καλωδίου Ethernet με το Router επιτρέποντας στο Arduino να συνδεθεί με το διαδίκτυο. Επιπλέον η πλακέτα αυτή έχει θύρα υποδοχής κάρτας micro SD στην οποία αποθηκεύσαμε τον κώδικα της ιστοσελίδας που δημιουργήσαμε. Έτσι χρησιμοποιούμε τον Arduino ως εξυπηρετητή (server) για να έχουμε την δυνατότητα πρόσβασης στην ιστοσελίδα από οποιοδήποτε τερματικό, φορητή συσκευή όπως κινητό τηλέφωνο (smart phone), ταμπλέτα (tablet) ή/και ηλεκτρονικό υπολογιστή.

## **Αισθητήριο DHT 11**

Το DHT-11 είναι ένας , χαμηλού κόστους, αισθητήρας για την ανάγνωση της υγρασίας και θερμοκρασίας στον χώρο. Στο εσωτερικό του κρύβει έναν αισθητήρα υγρασίας και ένα θερμίστορ (μεταβλητή αντίσταση που η τιμή της αλλάζει σε σχέση με την θερμοκρασία) “διαβάζοντας” έτσι τον αέρα που το περιβάλλει [4].

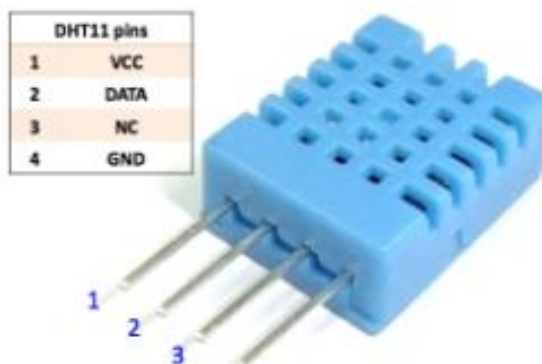


*Εικόνα 1.DHT 11*

Οι συνδεσμολογία του DHT-11 με το μικροελεγκτή είναι απλή: ο πρώτος ακροδέκτης από αριστερά στα +5V, ο δεύτερος ακροδέκτης (data) σε μια ψηφιακή είσοδο και ο τέρμα δεξιά στην γείωση του κυκλώματος.

## Τεχνικές πληροφορίες:

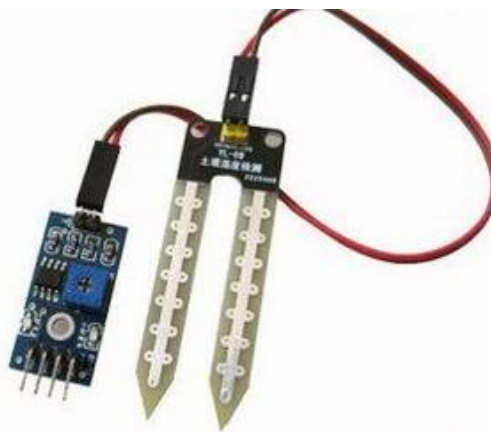
- Πηγή : 3-5V
- Μέγιστο ρεύμα: 2.5mA
- Υγρασία: 20-80%, ακρίβεια 2-5%
- Θερμοκρασία: 0 έως 50°C, ακρίβεια  $\pm 0.5^\circ$



Εικόνα 2.DHT 11 Pins

## Αισθητήριο υγρασίας εδάφους

Στην εικόνα που ακολουθεί βλέπουμε το αισθητήριο μέτρησης υγρασίας εδάφους η τιμές του οποίου καθορίζουν στην εργασία μας πότε θα ενεργοποιηθεί η αντλία νερού για το αυτόματο πότισμα .

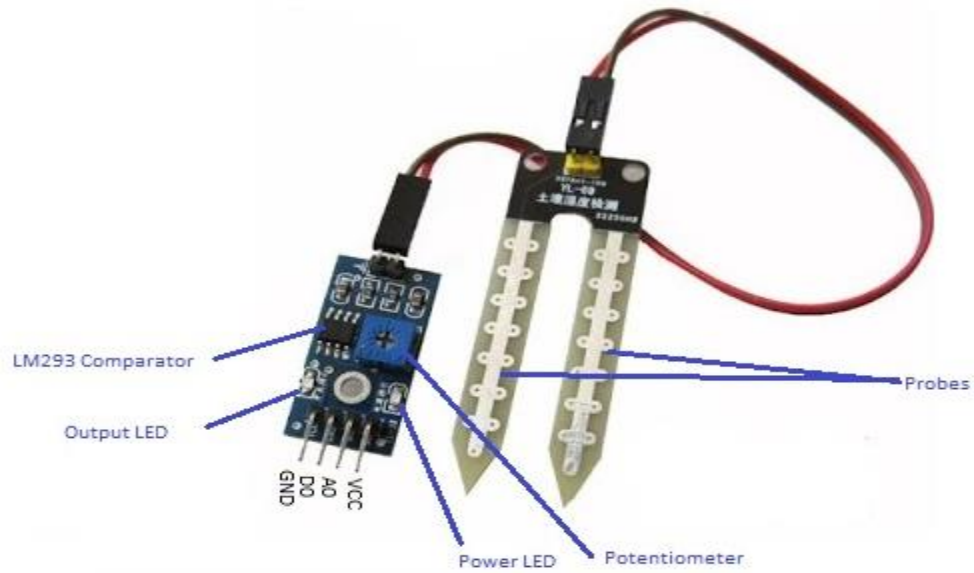


Εικόνα 3. Αισθητήριο Υγρασίας Εδάφους

Ο αισθητήρας υγρασίας χρησιμοποιείται για τη μέτρηση της περιεκτικότητας σε νερό (υγρασία) του εδάφους. Όταν το έδαφος έχει έλλειψη νερού, η έξοδος (τάση) της μονάδας είναι σε υψηλό επίπεδο, αλλιώς η έξοδος είναι σε χαμηλή στάθμη. Αυτός ο αισθητήρας υπενθυμίζει στον χρήστη να κάνει πότισμα των φυτών και παρακολουθεί την περιεκτικότητα σε υγρασία του εδάφους. Έχει χρησιμοποιηθεί ευρέως στη γεωργία, την άρδευση των καλλιεργειών και τη βοτανική κηπουρική [5].

### Χαρακτηριστικά :

- Τάση λειτουργίας: **5V**
- Ρεύμα λειτουργίας: **<20mA**
- Τύπος διασύνδεσης: **Αναλογικός**
- Θερμοκρασία λειτουργίας :**10°C~30°C**



Εικόνα 4. Soil Moisture Sensor Pins

## ΚΕΦΑΛΑΙΟ 3 – Κατασκευή αυτόματης γλάστρας

Στο «Κεφάλαιο 3 – Κατασκευή αυτόματης γλάστρας» παρουσιάζουμε αναλυτικά το κατασκευαστικό μέρος (hardware) της αυτόματης γλάστρας και συγκεκριμένα τα ηλεκτρονικά εξαρτήματα που χρειάστηκαν για την υλοποίηση του έργου, το σχέδιο της κατασκευής και το ηλεκτρονικό κύκλωμα.

### 3.1 Σχέδιο Γλάστρα

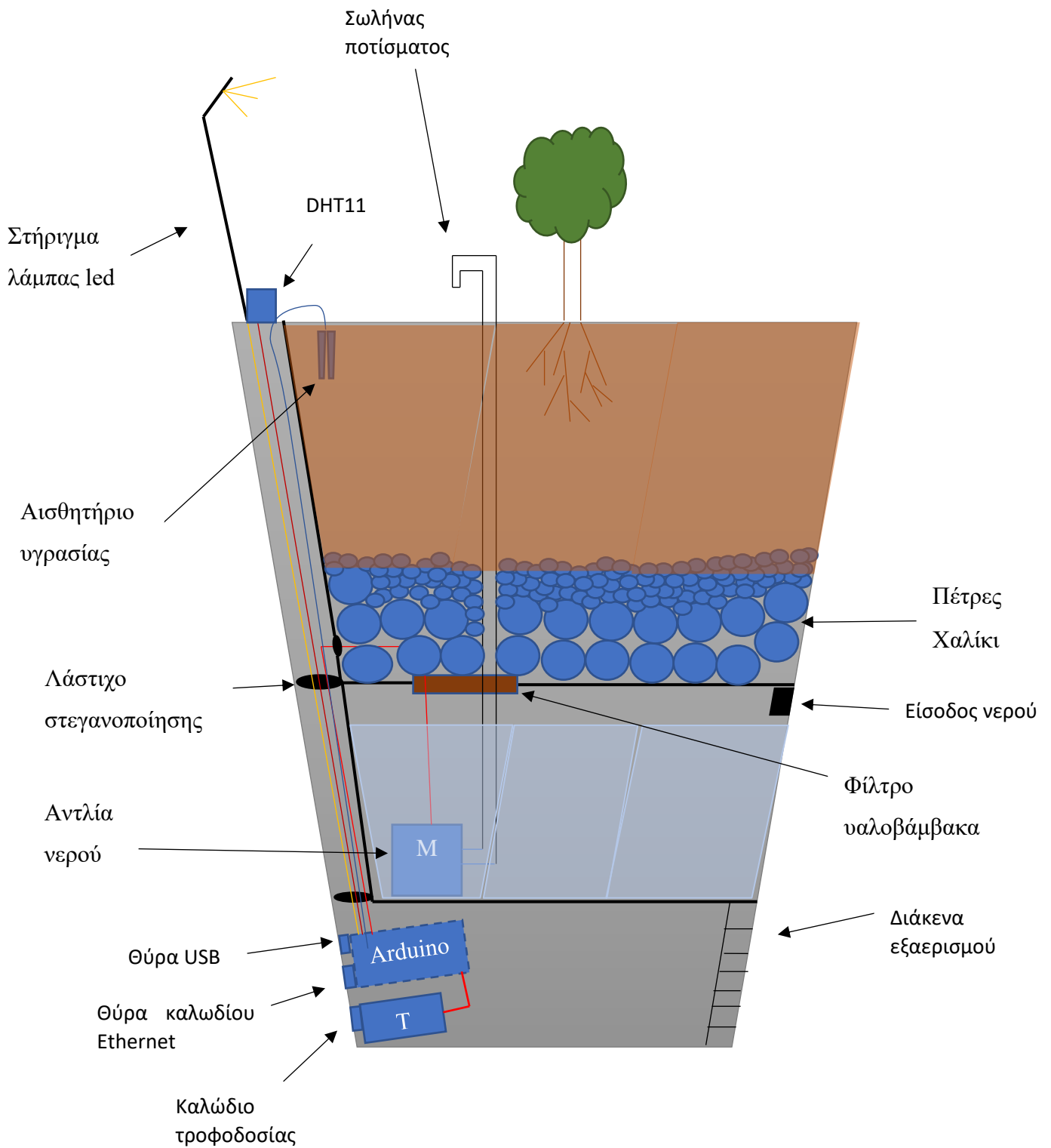
Στην Εικόνα 5.Πρόσοψη γλάστρας Εικόνα 5 παρατηρούμε την γλάστρα η οποία αποτελείται από τρία επίπεδα κατασκευής: το κατώτερο, μεσαίο και ανώτερο επίπεδο.

Το εσωτερικό του κατώτερου επιπέδου είναι κενό, το μεσαίο επίπεδο αποθηκεύει το νερό του ποτίσματος της γλάστρας και το ανώτερο επίπεδο περιέχει υλικά αποστράγγισης (πέτρες) και το χώμα στήριξης του φυτού.

Στο εσωτερικό του κατωτέρου επιπέδου έχουν τοποθετηθεί ο μικροελεγκτής Arduino Mega 2560 καθώς και το απαραίτητο τροφοδοτικό του. Στην δεξιά πλευρά του τοιχώματος υπάρχουν διάκενα προκειμένου να εισέρχεται κρύος αέρας στο εσωτερικό και να μην υπερθερμαίνεται ο μικροελεγκτής. Από την άλλη πλευρά υπάρχουν διάκενα που εφάπτονται των θυρών USB, Ethernet και τροφοδοσίας του Arduino όπου αφενός συνδέονται το καλώδιο Ethernet στο router και αφετέρου το καλώδιο του τροφοδοτικού στο δίκτυο ηλεκτρικής ενέργειας.

Στο μεσαίο επίπεδο έχουμε εισάγει νερό και μία υποβρύχια αντλία νερού έτσι ώστε να μεταφέρει το νερό στο ανώτερο επίπεδο στο οποίο βρίσκεται το φυτό. Στο πάνω μέρος του μεσαίου επιπέδου υπάρχει ένα κενό στο οποίο έχουμε τοποθετήσει ένα συρμάτινο πλέγμα και πάνω σε αυτό υαλοβάμβακα προκειμένου να επιστρέφει φιλτραρισμένο το νερό που δεν απορροφάται από το χώμα και το φυτό.

Πάνω δεξιά βρίσκεται η είσοδος νερού για να γεμίζουμε την δεξαμενή της αυτόματης γλάστρας.

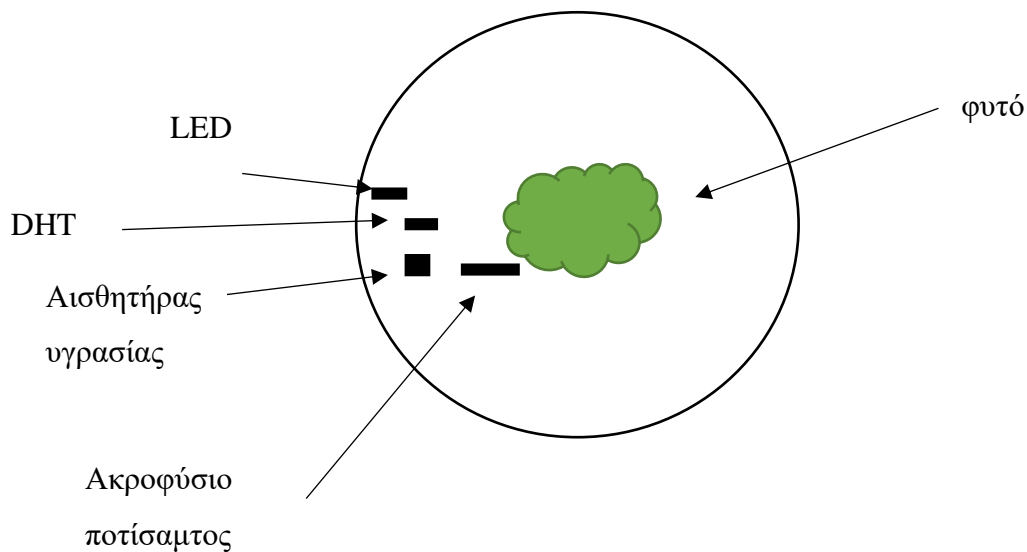


Εικόνα 5. Πρόσοψη γλάστρας

Στο κάτω μέρος του ανώτερου επιπέδου έχουμε τοποθετήσει πέτρες και χαλίκια αποστράγγισης έτσι ώστε το χώμα που έχουμε τοποθετήσει από πάνω τους να μην περνά στο φίλτρο και από εκεί στο μεσαίο επίπεδο. Επίσης σε αυτό το επίπεδο βρίσκονται τα αισθητήρια υγρασίας εδάφους, θερμοκρασίας και υγρασίας ατμόσφαιρας καθώς και η λάμπα LED.

Τέλος να αναφέρουμε ότι στο αριστερό τμήμα της γλάστρας υπάρχει ένας ενιαίος χώρος μακρόστενος ο οποίος είναι στεγανός και ξεκινά από το κατώτερο επίπεδο φτάνοντας μέχρι το ανώτερο. Αυτό το τμήμα χρησιμοποιείται για την καλωδίωση του μικροελεγκτή με τα υπόλοιπα ηλεκτρονικά μέρη του έργου.

Στην Εικόνα 6 παρουσιάζεται η κάτοψη του μηχανολογικού σχεδίου της αυτόματης γλάστρας

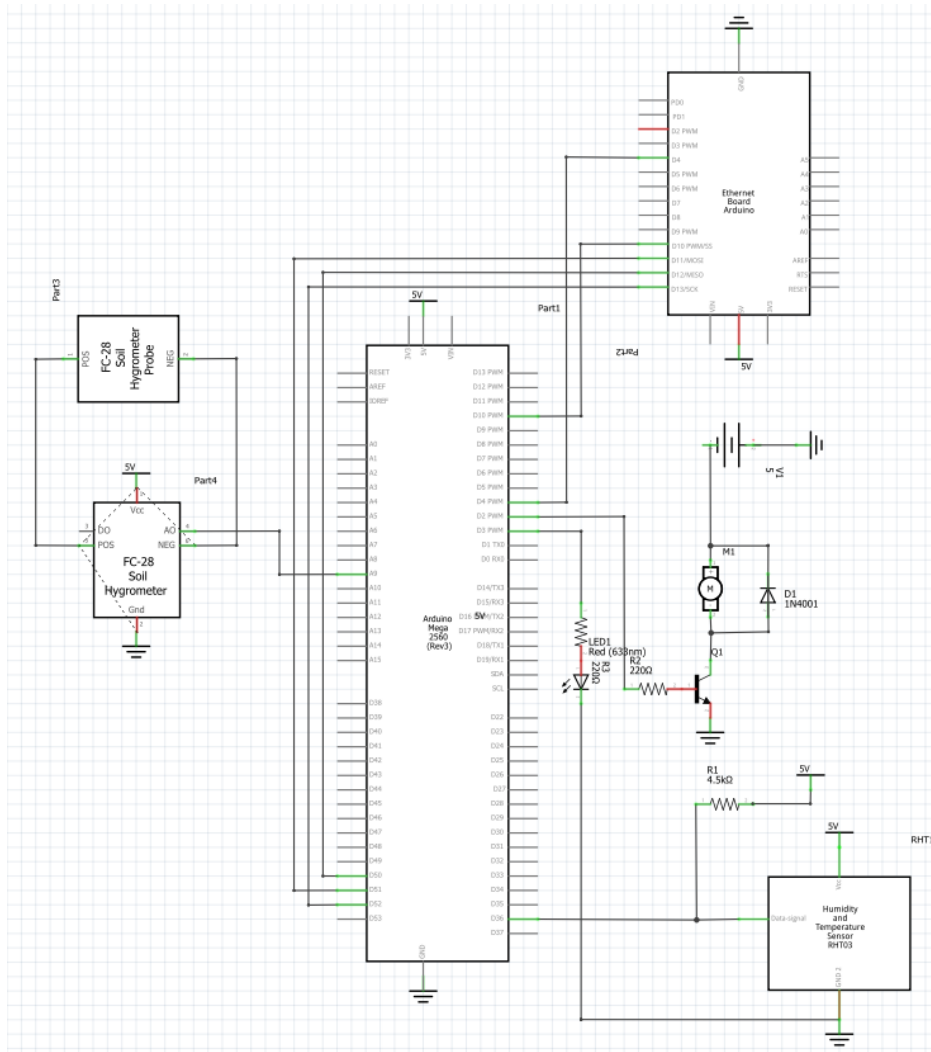


Εικόνα 6. Κάτοψη γλάστρας



### 3.2 Ηλεκτρονικό κύκλωμα

Στην παρακάτω εικόνα παρουσιάζεται το ηλεκτρονικό κύκλωμα της εργασίας.



Εικόνα 7. Ηλεκτρονικό κύκλωμα

Η διάδος χρησιμοποιείται για να προστατέψει το τρανζίστορ από το ρεύμα επαγωγής, το οποίο δημιουργείτε από το κινητήρα dc κατά την απότομη διακοπή της τροφοδοσίας του [6]. Εν μέρη ο κινητήρας είναι ένας επαγωγέας. Εάν το τρανζίστορ κόψει το ρεύμα ακαριαία, τότε το ρεύμα πρέπει να συνεχίσει να ρέει καθώς ο επαγωγέας τείνει να διατηρήσει την αρχική ροή του ρεύματος. Εάν δεν είχαμε τοποθετήσει την διάδο το ρεύμα θα πέρναγε στο τρανζίστορ το οποίο θα καιγόταν. Τώρα που τοποθετήσαμε όμως την διάδο το ρεύμα ρέει μέσα από αυτήν και ξαναγυρνάει στο επαγωγέα και αυτό συνεχίζεται μέχρι να μηδενιστεί το ρεύμα λόγω της αντίστασης του καλωδίου.

### 3.3 Λίστα υλικών

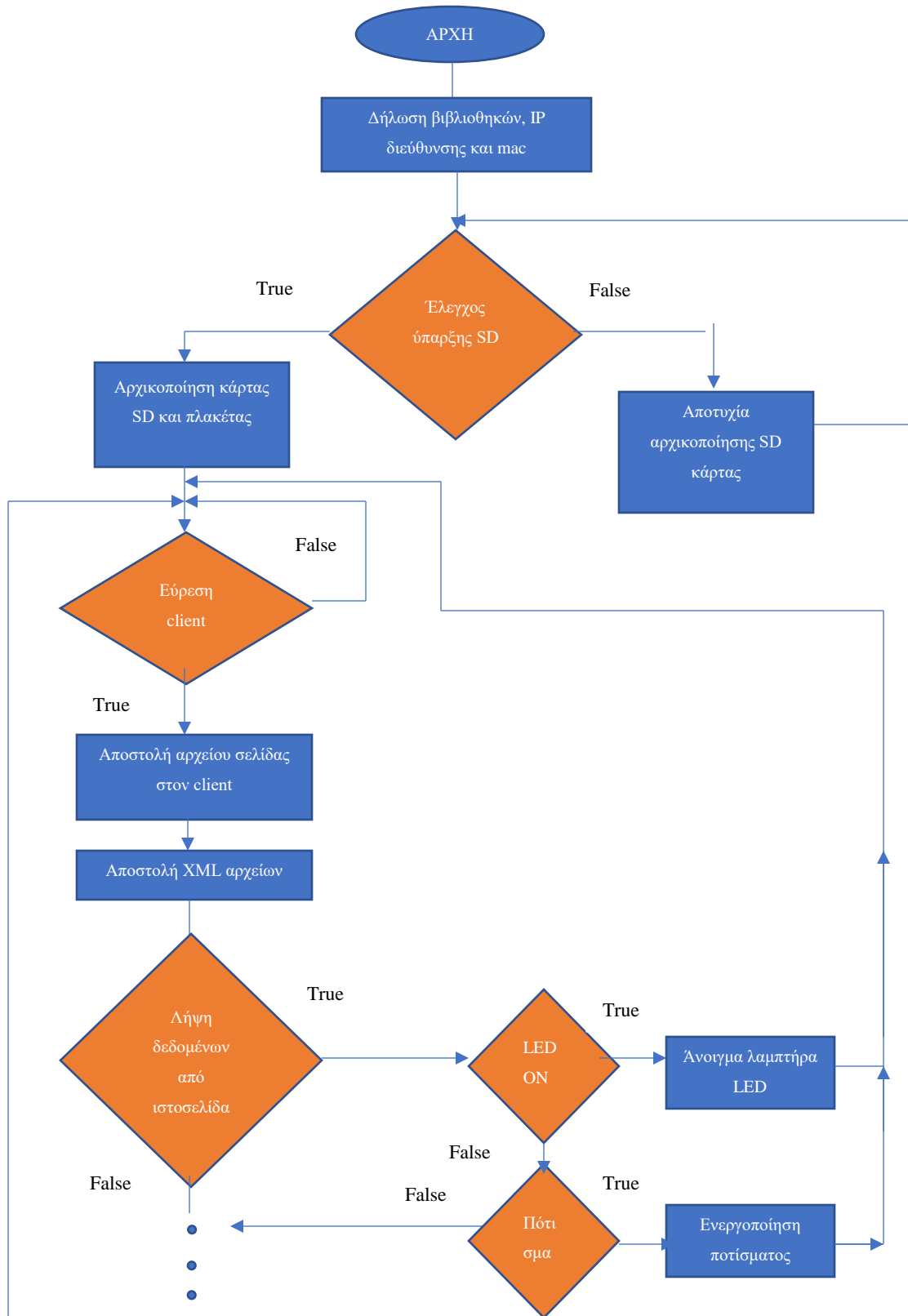
Για την υλοποίηση της κατασκευής της αυτόματης γλάστρας χρησιμοποιήθηκαν τα παρακάτω υλικά:

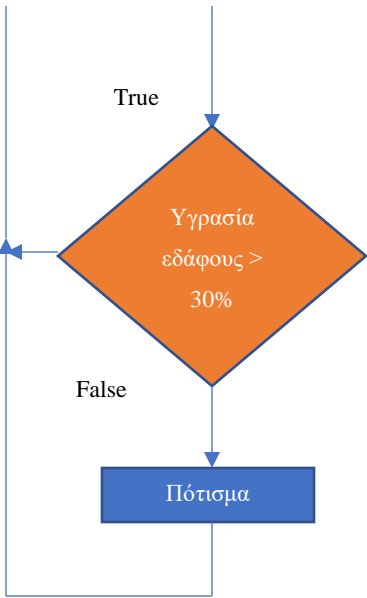
- Γλάστρα πλαστικής κατασκευής
- Υαλοβάμβακας
- Πέτρες και χαλίκια αποστράγγισης
- Χώμα για φυτά
- Arduino Mega 2560
- Ethernet shield
- Κάρτα micro SD
- Soil moisture sensor
- DHT 11 sensor
- LED ανάπτυξης φυτών
- Αντιστάσεις: 1x10KΩ , 2x220Ω
- Πυκνωτής 100pF
- Αντλία νερού τάσης 6V
- Τροφοδοτικό 12V / 1A
- Τρανζίστορ 2N2222
- Δίοδος 1N4001

## **ΚΕΦΑΛΑΙΟ 4 – Λογισμικό αυτόματης γλάστρας**

Στο «Κεφάλαιο 4 – Λογισμικό αυτόματης γλάστρας» παρουσιάζουμε αναλυτικά το λογισμικό - πρόγραμμα (software) της αυτόματης γλάστρας και συγκεκριμένα το διάγραμμα ροής (flow chart), τον κώδικα (C++) του Arduino και τον κώδικα (HTML, CSS, Javascript) της ιστοσελίδας απεικόνισης των συνθηκών διαβίωσης του φυτού.

### **4.1 Διάγραμμα ροής**





## 4.2 Κώδικας Arduino

```
#include <SPI.h>
#include <Ethernet2.h>
#include <SD.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>

#define DHTTYPE DHT11
DHT dht(36, DHTTYPE);

//Καθορισμός μεγέθους του ταμειυτήρα (buffer) ο οποίος χρησιμοποιείται για την προσωρινή
αποθήκευση των HTTP αιτήσεων.
#define REQ_BUF_SZ 60

//Η διεύθυνση MAC της Ethernet shield.
byte mac[] = { 0x90, 0xA2, 0xDA, 0x11, 0x15, 0x7F };
IPAddress ip(192, 168, 1, 122); // IP την οποία καθορίζουμε εμείς με βάση το
δίκτυό μας.
EthernetServer server(8085 ); // Δημιουργία server στην πόρτα 8085.
File webFile; // Το αρχείο της σελίδας στην κάρτα SD.
char HTTP_req[REQ_BUF_SZ] = {0}; //Δημιουργία πίνακα (60X1) με στοιχεία null.
char req_index = 0; // Δήλωση περιεχομένου HTTP αιτήματος
boolean LED_state[3] = {0}; // LED_state[3] : Είναι ένας πίνακας αποθήκευσης της
κατάστασης των LED.Ο αριθμός μέσα στις αγκύλες συμβολίζει το πλήθος των γραμμών του
πίνακα.
int soil_moisture;
int sensor_pin = A0; //Καθορισμός pin αισθητηρίου υγρασίας εδάφους.Το A0
είναι αναλογικό pin.

// Η συνάρτηση setup() καλείται όταν ξεκινά το sketch.
Χρησιμοποιείται για να δηλώσουμε μεταβλητές, τις καταστάσεις των pin, για την έναρξη
χρησιμοποίησης των βιβλιοθηκών, και άλλα.

//Η συνάρτηση setup() θα τρέξει μόνο μία φορά, αφού
ενεργοποιήσουμε το Arduino είτε μετά από κάποια επανεκκίνηση [7].
```

```

void setup()
{
  Serial.begin(9600); //ορίζουμε τον ρυθμό αποστολής των δεδομένων ανά
  δευτερόλεπτο (baud) για την σειριακή μετάδωσή τους [8].

  // Αρχικοποίηση της κάρτας SD.
  Serial.println("Initializing SD card...");
  if (!SD.begin(4)) { //Εάν δεν γίνει η αρχικοποίηση το γράφει στην σειριακή οθόνη
του arduino.
    Serial.println("ERROR - SD card initialization failed!");
    return;
  }
  Serial.println("SUCCESS - SD card initialized.");

  // Έλεγχος για το αρχείο "index.htm".
  if (!SD.exists("index.htm")) {
    Serial.println("ERROR - Can't find index.htm file!"); // Εάν δεν βρεθεί το αρχείο
της sd κάρτας το εκτυπώνει
στη συριακή οθόνη του arduino.
    return;
  }
  Serial.println("SUCCESS - Found index.htm file.");

  // Έναρξη διαβάσματος αισθητηρίου DHT11
  dht.begin();

  //Καθορισμός των PIN για το LED και την αντλία.
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);

  Ethernet.begin(mac, ip); // Αρχικοποίηση της πλακέτας Ethernet.
  server.begin(); // Ο Server ξεκινά να "ακούει" για πελάτες
(clients).
}

```

// Αφού έχουμε δημιουργήσει την συνάρτηση setup(), η οποία αρχικοποιεί και ορίζει τις αρχικές τιμές, η συνάρτηση loop() κάνει αυτό ακριβώς που αναφέρει το όνομά της, επαναλαμβάνεται συνεχώς, επιτρέποντας στο πρόγραμμα να αλλάζει και να ανταποκρίνεται. Την χρησιμοποιούμε για να ελέγξουμε ενεργά το Arduino [3].

```
void loop()
```

```
{
```

```
    EthernetClient client = server.available(); // Εάν υπάρξει κάποιος πελάτης (client),  
    τότε θα υπάρξουν διαθέσιμα bytes για ανάγνωση [4].
```

```
    if (client) { // Εάν υπάρχει πελάτης.
```

```
        boolean currentLineIsBlank = true; //Δημιουργία μιας boolean μεταβλητής δίνοντάς  
        της την τιμή true.
```

```
        while (client.connected()) { //Εφόσον ο πελάτης είναι συνδεδεμένος.
```

```
            if (client.available()) { // και εάν υπάρχουν δεδομένα διαθέσιμα για διάβασμα.
```

```
                char c = client.read(); // Διάβασε ένα byte (χαρακτήρα) από τον πελάτη  
                (client).
```

```
// Αποθήκευσε στην προσωρινή μνήμη (buffer) το πρώτο κομμάτι του αιτήματος HTTP.Όπου το  
buffer εδώ είναι η αλληλουχία (string) με την ονομασία "HTTP_req".
```

```
    if (req_index < (REQ_BUF_SZ - 1)) {
```

```
        HTTP_req[req_index] = c; // Αποθήκευση του χαρακτήρα του HTTP αιτήματος.
```

```
        req_index++;
```

```
    }
```

```
//Εκτύπωσε τον χαρακτήρα του HTTP αιτήματος στην σειριακή οθόνη.
```

```
    Serial.print(c);
```

```
    // Η τελευταία σειρά αιτήματος του πελάτη (του client request) είναι κενή  
    και τελειώνει με \n.
```

```
    //Απαντά στον πελάτη (client) μόνο αφότου λάβουμε και την τελευταία σειρά.
```

```
    if (c == '\n' && currentLineIsBlank) {
```

```
        //Στέλνει μία πρότυπη (στάνιαρ) HTTP απάντηση (response header).
```

```
        client.println("HTTP/1.1 200 OK");
```

```
        //Το υπόλοιπο της απάντησης (response header) που θα σταλεί στον client  
εξαρτάται από το αν το αρχείο που ζητείται είναι η σελίδα ή το αρχείο XML.
```

```
    if (StrContains(HTTP_req, "ajax_inputs")) {
```

```
        // Στέλνει την υπόλοιπη HTTP απάντηση ή αλλιώς HTTP επικεφαλίδα (header).
```

```
        client.println("Content-Type: text/xml");
```

```
        client.println("Connection: keep-alive");
```

```
        client.println();
```

```
        SetLEDS(); // Στέλνει το πρόγραμμα στην συνάρτηση SetLEDS().
```



```

XML_response(client); // Στέλνει στον client το XML αρχείο στο οποίο
εμπεριέχονται οι καταστάσεις των εισόδων.

        Watering ( ) ;

    }

else { // web page request
    // Στέλνει την υπόλοιπη HTTP απάντηση ή αλλιώς HTTP επικεφαλίδα (header).
    client.println("Content-Type: text/html");
    client.println("Connection: keep-alive");
    client.println();
    // Στέλνει την σελίδα.

    webFile = SD.open("index.htm"); // Άνοιγμα του αρχείου της σελίδας το
    οποίο βρίσκεται στην SD.

    if (webFile) {
        while (webFile.available()) {
            client.write(webFile.read()); // Στέλνει την σελίδα στον πελάτη
(client).
        }
        webFile.close();
    }
}

//προβάλλει το HTTP έτοιμα που έλαβε στην σειριακή οθόνη.
Serial.print(HTTP_req);

// Μηδενισμός της προσωρινής μνήμης (buffer) και όλων των στοιχείων της μέχρι
το 0.

req_index = 0;
StrClear(HTTP_req, REQ_BUF_SZ);
break;

}

//Κάθε ληφθείσα γραμμή κειμένου από τον πελάτη τελειώνει με \r\n
if (c == '\n') {
    // τελευταίος χαρακτήρας σε σειρά του ληφθέντος κειμένου.
    currentLineIsBlank = true;
}

else if (c != '\r') {

    //Ένας χαρακτήρας κειμένου ελήφθη από τον πελάτη
    currentLineIsBlank = false;
}

```

```

    }
}
}
delay(1); // Δίνουμε στον web browser χρόνο να λάβει
τα δεδομένα.

client.stop(); // Κλείσιμο της σύνδεσης.
} // end if (client)
}

// Ελέγχει αν το HTTP έτοιμα αλλάζει την κατάσταση
των LEDs.

//Επίσης αποθηκεύει την κατάσταση τους.

void SetLEDs(void)
{

// LED 1 (pin 2)

if (StrContains(HTTP_req, "LED1=1")) { // Οι μεταβλητές "LED1=1" και "LED1=2"
βρίσκονται μέσα στις κεφαλίδες του HTTP αιτήματος (HTTP Header Response).

LED_state[0] = 1; // Αποθήκευση κατάστασης LED.

digitalWrite(2, HIGH);

}

else if (StrContains(HTTP_req, "LED1=0")) {

LED_state[0] = 0; // Αποθήκευση κατάστασης LED.

digitalWrite(2, LOW);

}

// Αποστολή του XML αρχείου.

void XML_response(EthernetClient cl)
{

//Δήλωση μεταβλητών τύπου float για τις τιμές της υγρασίας και θερμοκρασίας του
περιβάλλοντος.

float humidity, temperature;

//dht11

humidity = dht.readHumidity(); //Διαβάζει την υγρασία και αποθηκεύει την
τιμή στην μεταβλητή humidity.

temperature = dht.readTemperature(); //Διαβάζει την θερμοκρασία και αποθηκεύει την
τιμή στην μεταβλητή temperature.

```

```

//soil_moisture

soil_moisture = analogRead(sensor_pin); //Διαβάζει μία τιμή από το αισθητήριο, η οποία
είναι σε αναλογική μορφή και την αποθηκεύει στην μεταβλητή soil_moisture.

soil_moisture = map(soil_moisture, 550, 0, 0, 100); // Αντιστοιχούμε τις τιμές του
αισθητηρίου στο εύρος από 0 - 100, καθώς η υγρασία μετριέται σε ποσοστό επί της εκατό.
Όταν πείραμε μετρήσεις από ξηρό έδαφος η τιμή του αισθητήρα ήταν 550 και στο υγρό έδαφος
10. Οπότε αντιστοιχίσαμε αυτές τις τιμές στην κλίμακα των 100.

//Αποστολή αρχείου XML.

cl.print("<?xml version = \"1.0\" ?>");

cl.print("<inputs>");

//---Τιμές θερμοκρασίας και υγρασίας---//

cl.print("<analog>");
cl.print(temperature);
cl.print("</analog>");
Serial.print("\n");
Serial.print(temperature);
Serial.print("\n");

cl.print("<analog>");
cl.print(humidity);
cl.print("</analog>");
Serial.print("\n");
Serial.print(humidity);
Serial.print("\n");

//----soil moisture----//

cl.print("<analog>");
cl.print(soil_moisture);
cl.print("</analog>");
Serial.print("\n");
Serial.print("soil_moisture");
Serial.print(soil_moisture);
Serial.print("%");
Serial.print("\n");

```

```

//---Είσοδοι κατάστασης Led ---//
//LED 1
cl.print("<LED>");

if (LED_state[0]) {
    cl.print("on");
}
else {
    cl.print("off");
}
cl.println("</LED>");
//Αυτόματο πότισμα όταν η υγρασία εδάφους βρίσκεται στο 30% και κάτω
void watering();
{
If ( soil_moisture <= 30%)
{
    digitalWrite ( 3, high ) ;
}
else
{
digitalWrite ( 3, low ) ;
}

}

//Θέτει κάθε στοιχείο του string ως 0 (καθαρισμός στήλης)
void StrClear(char *str, char length)
{
    for (int i = 0; i < length; i++) {
        str[i] = 0;
    }
}

//Ψάχνει για το string sfind στο string str
//Επιστρέφει 1 εάν βρει το string
//Επιστρέφει 0 εάν δεν βρει το string

```

```

char StrContains(char *str, char *sfind)
{
    char found = 0;
    char index = 0;
    char len;

    len = strlen(str);

    if (strlen(sfind) > len) {
        return 0;
    }
    while (index < len) {
        if (str[index] == sfind[found]) {
            found++;
            if (strlen(sfind) == found) {
                return 1;
            }
        }
        else {
            found = 0;
        }
        index++;
    }

    return 0;
}

```

## Περιβάλλον ανάπτυξης κώδικα Arduino

Το περιβάλλον ανάπτυξης κώδικα Arduino IDE (ολοκληρωμένο περιβάλλον ανάπτυξης - Integrated Development Environment [9] ) μπορεί να επεκταθεί μέσω της χρήσης βιβλιοθηκών. Οι βιβλιοθήκες είναι μια συλλογή κωδικών που μας διευκολύνει να ελέγξουμε έναν αισθητήρα, μία οθόνη, έναν κινητήρα και άλλα ηλεκτρονικά αισθητήρια - εξαρτήματα.

Στην

Εικόνα 8 παρουσιάζεται ο τρόπος εισαγωγής των βιβλιοθηκών που χρησιμοποιήσαμε στον πρόγραμμά μας.

```
#include <SPI.h>
#include <Ethernet2.h>
#include <SD.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
```

*Εικόνα 8.Βιβλιοθήκες*

**Βιβλιοθήκη Serial Peripheral Interface (SPI):** Συμπεριλαμβάνουμε στο πρόγραμμά μας την βιβλιοθήκη SPI στην οποία εμπεριέχεται το πρωτόκολλο SPI. Το SPI είναι ένα πρωτόκολλο σύγχρονης σειριακής μετάδοσης δεδομένων, το οποίο συνήθως χρησιμοποιείται από μικροελεγκτές για την γρήγορη επικοινωνία μεταξύ άλλων περιφερειακών συσκευών οι οποίες βρίσκονται σε μικρή απόσταση [10].

**Βιβλιοθήκη Ethernet2.h:** Έχει σχεδιαστεί να δουλεύει με το Ethernet shield2 του μικροελεγκτή Arduino και μας επιτρέπει να συνδεθούμε στο διαδίκτυο.

```
// MAC address from Ethernet shield
sticker under board

byte mac[] = { 0x90, 0xA2, 0xDA, 0x11,
0x15, 0x7F };

IPAddress ip(192, 168, 1, 122); // IP
address, may need to change depending
on network

EthernetServer server(8085 ); //
create a server at port 80

File webFile; // the web page file on
the SD card
```

*Εικόνα 9 - Web File and Address of IP, mac and ethernet port*

**byte mac[] = { 0x90, 0xA2, 0xDA, 0x11, 0x15, 0x7F }; :** Δήλωση της διεύθυνσης mac της Ethernet shield. Κάθε συσκευή έχει μία μοναδική διεύθυνση mac (Media Access Control) ή αλλιώς φυσική διεύθυνση, προκειμένου να έχει την δυνατότητα επικοινωνίας με άλλες συσκευές στο φυσικό επίπεδο.

**Τι είναι το Client Request και το Server Response [11].**

### Client Request

Όταν σερφάρουμε στην διεύθυνση IP του εξυπηρέτη (server) του Arduino, ο φυλλομετρητής (web browser) του πελάτη (client) θα στείλει ένα αίτημα, όπως αυτό που βλέπουμε παρακάτω, στον εξυπηρέτη (server).

```
GET / HTTP/1.1\r\n
Host: 10.0.0.20\r\n
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:17.0) Gecko/20100101
Firefox/17.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
Accept-Language: en-ZA,en-GB;q=0.8,en-US;q=0.5,en;q=0.3\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
\r\n
```

Οι πληροφορίες του αιτήματος θα διαφέρουν, ανάλογα με τον φυλλομετρητή (browser) και το λειτουργικό σύστημα από το οποίο στέλνεται το αίτημα.

Οι χαρακτήρες `\r\n` που βλέπουμε στο τέλος κάθε σειράς στο παραπάνω αίτημα, είναι μη ορατή χαρακτήρες (μη εκτυπώσιμη χαρακτήρες). Ο χαρακτήρας `\r` συμβολίζει την επιστροφή κρατουμένου και ο χαρακτήρας `\n` την τροφοδοσία σειράς (line feeding or newline character).

Η τελευταία γραμμή του αιτήματος περιέχει τα σύμβολα `\r\n` χωρίς την προσθήκη κάποιου κειμένου. Αυτή είναι η κενή γραμμή την οποία μόλις διαβάσει το Arduino στέλνει την απάντηση στον πελάτη φυλλομετρητή (client web browser).

Με άλλα λόγια, το πρόγραμμα του Arduino διαβάζει κάθε χαρακτήρα από το παραπάνω αίτημα και όταν εντοπίσει την κενή γραμμή καταλαβαίνει ότι εδώ τελειώνει το αίτημα.

## Server Response

Αφού ληφθεί το αίτημα για την ιστοσελίδα από τον πελάτη (client), ο εξυπηρετής (server) πρώτα στέλνει μία πρότυπη απάντηση HTTP και μετά την ιστοσελίδα.

Η απάντηση του εξυπηρετητή, όπου στην συγκεκριμένη περίπτωση είναι το Arduino, είναι η εξής:

```
HTTP/1.1 200 OK\r\n
Content-Type: text/html\r\n
Connection: close\r\n
\r\n
```

Ξανά οι μη ορατοί χαρακτήρες `\r\n` φαίνονται στην παραπάνω απάντηση. Η `println()` συνάρτηση στο πρόγραμμα του Arduino προσθέτει αυτομάτως τους χαρακτήρες `\r\n` στο τέλος της κάθε γραμμής. Η κενή `println()` συνάρτηση στο τέλος της HTTP απάντησης απλά στέλνει τους χαρακτήρες `\r\n` χωρίς κάποιο κείμενο.

**Βιβλιοθήκη SD.h:** Η βιβλιοθήκη αυτή σου επιτρέπει να διαβάζεις και να γράφεις στην SD κάρτα. Η βιβλιοθήκη υποστηρίζει το σύστημα αρχείων FAT16 και FAT32 [12].

**Βιβλιοθήκη Adafruit\_Sensor.h:** Αυτή η βιβλιοθήκη περιέχει τους απαραίτητους drivers (οδηγούς) έτσι ώστε να λειτουργούν σωστά τα αισθητήρια θερμοκρασίας και υγρασίας DHT11.



**Βιβλιοθήκη DHT.h:** Η βιβλιοθήκη αυτή χρησιμοποιείται επίσης για την σωστή λειτουργία του αισθητηρίου DHT11.

```
#define DHTTYPE DHT11
DHT dht(36, DHTTYPE);
#define REQ_BUF_SZ 60
```

*Εικόνα 10- Δήλωση βιβλιοθήκης DHT 11*

**#define DHTTYPE DHT11:** Δηλώνουμε ότι θα χρησιμοποιήσουμε το μοντέλο DHT11, καθώς υπάρχουν και άλλα DHT αισθητήρια.

**DHT dht (36, DHTTYPE):** Καθορίζουμε σε ποιο pin του Arduino θα συνδεθεί το αισθητήριο DHT11.

**#define REQ\_BUF\_SZ 60 :** Καθορίζει το μέγεθος της ενδιάμεσης μνήμης (buffer) στην οποία αποθηκεύονται προσωρινά τα αιτήματα του πρωτοκόλλου HTTP (Hypertext Transfer Protocol).

## 4.4 Κώδικας Ιστοσελίδας

```
<!DOCTYPE html>

<html>

<head>

  <title>Automated Pot</title>

  <script>

    strLED1 = "";

    strLED2 = "";

    var LED1_state = 0;

    var LED2_state = 0;

    var data_val = 0;

    var data_val2= 0;

    var data_val3= 0;

<!-- Gauge Code Starts -->

var Gauge=function(b){function l(a,b){for(var c in b)"object"==typeof b[c]&&"[object Array]"!==Object.prototype.toString.call(b[c])&&"renderTo"!==c?("object"!==typeof a[c]&&(a[c]={}),l(a[c],b[c])):a[c]=b[c]}function q(){z.width=b.width;z.height=b.height;A=z.cloneNode(!0);B=A.getContext("2d");C=z.width;D=z.height;t=C/2;u=D/2;f=t<u?t:u;A.i8d=!1;B.translate(t,u);B.save();a.translate(t,u);a.save()}function v(a){var b=new Date;G=setInterval(function(){var c=(new Date-b)/a.duration;l<c&&(c=1);var f=("function"==

typeof

a.delta?a.delta:M[a.delta])(c);a.step(f);l==c&&clearInterval(G)},a.delay||10)}function k(){G&&clearInterval(G);var a=I-n,h=n,c=b.animation;v({delay:c.delay,duration:c.duration,delta:c.fn,step:function(b){n=parseFloat(h)+a*b;E.draw()}})}function e(a){return a*Math.PI/180}function g(b,h,c){c=a.createLinearGradient(0,0,0,c);c.addColorStop(0,b);c.addColorStop(1,h);return c}function p(){var m=93*(f/100),h=f-m,c=91*(f/100),e=88*(f/100),d=85*(f/100);a.save();b.glow&&(a.shadowBlur=h,a.shadowColor=

"rgba(0,

0,

0,

0.5)");a.beginPath();a.arc(0,0,m,0,2*Math.PI,!0);a.fillStyle=g("#ddd","#aaa",m);a.fill();a.restore();a.beginPath();a.arc(0,0,c,0,2*Math.PI,!0);a.fillStyle=g("#fafafa","#ccc",c);a.fill();a.beginPath();a.arc(0,0,e,0,2*Math.PI,!0);a.fillStyle=g("#eee","#f0f0f0",e);a.fill();a.beginPath();a.arc(0,0,d,0,2*Math.PI,!0);a.fillStyle=b.colors.plate;a.fill();a.save()}function w(a){var h=!1;a=0===b.majorTicksFormat.dec?Math.round(a).toString():a.toFixed(b.majorTicksFormat.dec);return l<b.majorTicksFormat["int"]?

(h=-1<a.indexOf("."),-1<a.indexOf("-")?"-"+(b.majorTicksFormat["int"]+b.majorTicksFormat.dec+2+(h?1:0)-a.length)+a.replace("-","")):""+(b.majorTicksFormat["int"]+b.majorTicksFormat.dec+1+(h?1:0)-a.length)+a)}function d){var
```

```

m=81*(f/100);a.lineWidth=2;a.strokeStyle=b.colors.majorTicks;a.save();if(0===b.majorTi
cks.length){for(var h=(b.maxValue-
b.minValue)/5,c=0;5>c;c++)b.majorTicks.push(w(b.minValue+h*c));b.majorTicks.push(w(b.m
axValue))}for(c=0;c<b.majorTicks.length;++c)a.rotate(e(45+c*(270/(b.majorTicks.length-
1))))),a.beginPath(),a.moveTo(0,m),a.lineTo(0,m-
15*(f/100)),a.stroke(),a.restore(),a.save();b.strokeTicks&&(a.rotate(e(90)),a.beginPat
h(),a.arc(0,0,m,e(45),e(315),!1),a.stroke(),a.restore(),a.save())}function J(){var
m=81*(f/100);a.lineWidth=1;a.strokeStyle=b.colors.minorTicks;a.save();for(var
h=b.minorTicks*(b.majorTicks.length-
1),c=0;c<h;++c)a.rotate(e(45+c*(270/h))),a.beginPath(),a.moveTo(0,m),a.lineTo(0,m-
7.5*(f/100)),a.stroke(),a.restore(),a.save()}function s(){for(var
m=55*(f/100),h=0;h<b.majorTicks.length;++h){var c=
F(m,e(45+h*(270/(b.majorTicks.length-1)))));a.font=20*(f/200)+"px
Arial";a.fillStyle=b.colors.numbers;a.lineWidth=0;a.textAlign="center";a.fillText(b.ma
jorTicks[h],c.x,c.y+3)}function x(a){var
h=b.valueFormat.dec,c=b.valueFormat["int"];a=parseFloat(a);var
f=0>a;a=Math.abs(a);if(0<h){a=a.toFixed(h).toString().split(".");h=0;for(c=
=a[0].length;h<c;++h)a[0]="0"+a[0];a=(f?"-
":"")+a[0]+"."+a[1]}else{a=Math.round(a).toString();h=0;for(c=
=a.length;h<c;++h)a="0"+a;a=(f?"-":"")+a}return a}function F(a,b){var c=
Math.sin(b),f=Math.cos(b);return{x:0*f-a*c,y:0*c+a*f}}function N(){a.save();for(var
m=81*(f/100),h=m-15*(f/100),c=0,g=b.highlights.length;c<g;c++){var
d=b.highlights[c],r=(b.maxValue-b.minValue)/270,k=e(45+(d.from-
b.minValue)/r),r=e(45+(d.to-
b.minValue)/r);a.beginPath();a.rotate(e(90));a.arc(0,0,m,k,r,!1);a.restore();a.save();
var l=F(h,k),p=F(m,k);a.moveTo(l.x,l.y);a.lineTo(p.x,p.y);var
p=F(m,r),n=F(h,r);a.lineTo(p.x,p.y);a.lineTo(n.x,n.y);a.lineTo(l.x,l.y);a.closePath();
a.fillStyle=d.color;a.fill();
a.beginPath();a.rotate(e(90));a.arc(0,0,h,k-
0.2,r+0.2,!1);a.restore();a.closePath();a.fillStyle=b.colors.plate;a.fill();a.save();}
function K(){var
m=12*(f/100),h=8*(f/100),c=77*(f/100),d=20*(f/100),k=4*(f/100),r=2*(f/100),l=function(
){a.shadowOffsetX=2;a.shadowOffsetY=2;a.shadowBlur=10;a.shadowColor="rgba(188,
143,
143,
0.45)"};l();a.save();n=0>n?Math.abs(b.minValue-n):0<b.minValue?n-
b.minValue:Math.abs(b.minValue)+n;a.rotate(e(45+n/(b.maxValue-
b.minValue)/270));a.beginPath();a.moveTo(-r,-d);a.lineTo(-k,
0);a.lineTo(-1,c);a.lineTo(1,c);a.lineTo(k,0);a.lineTo(r,-
d);a.closePath();a.fillStyle=g(b.colors.needle.start,b.colors.needle.end,c-
d);a.fill();a.beginPath();a.lineTo(-0.5,c);a.lineTo(-1,c);a.lineTo(-k,0);a.lineTo(-r,-
d);a.lineTo(r/2-2,-d);a.closePath();a.fillStyle="rgba(255,
255,
255,
0.2)";a.fill();a.restore();l();a.beginPath();a.arc(0,0,m,0,2*Math.PI,!0);a.fillStyle=g
("#f0f0f0","#ccc",m);a.fill();a.restore();a.beginPath();a.arc(0,0,h,0,2*Math.PI,!0);a.
fillStyle=g("#e8e8e8","#f5f5f5",h);a.fill()}
function L(){a.save();a.font=40*(f/200)+"px Led";var b=x(y),h=a.measureText("-
"+x(0)).width,c=f-33*(f/100),g=0.12*f;a.save();var d=-h/2-0.025*f,e=c-g-
0.04*f,h=h+0.05*f,g=g+0.07*f,k=0.025*f;a.beginPath();a.moveTo(d+k,e);a.lineTo(d+h-
k,e);a.quadraticCurveTo(d+h,e,d+h,e+k);a.lineTo(d+h,e+g-
k);a.quadraticCurveTo(d+h,e+g,d+h-
k,e+g);a.lineTo(d+k,e+g);a.quadraticCurveTo(d,e+g,d,e+g-
k);a.lineTo(d,e+k);a.quadraticCurveTo(d,e,d+k,e);a.closePath();d=a.createRadialGradien
t(0,c-0.12*f-0.025*f+(0.12*f+0.045*f)/
2,f/10,0,c-0.12*f-
0.025*f+(0.12*f+0.045*f)/2,f/5);d.addColorStop(0,"#888");d.addColorStop(1,"#666");a.st
rokeStyle=d;a.lineWidth=0.05*f;a.stroke();a.shadowBlur=0.012*f;a.shadowColor="rgba(0,
0,
0,
1)";a.fillStyle="#babab2";a.fill();a.restore();a.shadowOffsetX=0.004*f;a.shadowOffsetY
=0.004*f;a.shadowBlur=0.012*f;a.shadowColor="rgba(0,
0,
0,
0.3)";a.fillStyle="#444";a.textAlign="center";a.fillText(b,-

```

```

0, c); a.restore() } Gauge.Collection.push(this); this.config={renderTo:null, width:200, height:200, title:!1,

maxValue:100, minValue:0, majorTicks:[], minorTicks:10, strokeTicks:!0, units:!1, valueFormat:{"int":3, dec:2}, majorTicksFormat:{"int":1, dec:0}, glow:!0, animation:{delay:10, duration:250, fn:"cycle"}, colors:{plate:"#fff", majorTicks:"#444", minorTicks:"#666", title:"#888", units:"#888", numbers:"#444", needle:{start:"rgba(240, 128, 128, 1)", end:"rgba(255, 160, 122, .9)"}}, highlights:[{from:20, to:60, color:"#eee"}, {from:60, to:80, color:"#ccc"}, {from:80, to:100, color:"#999"}]; var y=0, E=this, n=0, I=0, H=!1; this.setValue=

function(a){n=b.animation?y:a; var d=(b.maxValue-b.minValue)/100; I=a>b.maxValue?b.maxValue+d:a<b.minValue?b.minValue-d:a; y=a; b.animation?k():this.draw(); return this}; this.setRawValue=function(a){n=y=a; this.draw(); return this}; this.clear=function(){y=n=I=this.config.minValue; this.draw(); return this}; this.getValue=function(){return y}; this.onready=function(){l(this.config, b); this.config.minValue=parseFloat(this.config.minValue); this.config.maxValue=parseFloat(this.config.maxValue); b=this.config; n=y=b.minValue; if(!b.renderTo) throw Error("Canvas element was not specified when creating the Gauge object!"); var z=b.renderTo.tagName?b.renderTo:document.getElementById(b.renderTo), a=z.getContext("2d"), A, C, D, t, u, f, B; q(); this.updateConfig=function(a){l(this.config, a); q(); this.draw(); return this}; var M={linear:function(a){return a}, quad:function(a){return Math.pow(a, 2)}, quint:function(a){return Math.pow(a, 5)}, cycle:function(a){return 1-Math.sin(Math.acos(a))}, bounce:function(a){a:{a=1-a; for(var b=0, c=1; b+=c, c/=2) if(a>=(7-4*b)/11){a=-Math.pow((11-6*b-11*a)/4, 2)+Math.pow(c, 2); break} a=void 0} return 1-a}, elastic:function(a){a=1-a; return 1-Math.pow(2, 10*(a-1))*Math.cos(30*Math.PI/3*a)}}; G=null; a.lineCap="round"; this.draw=function(){if(!A.i8d){B.clearRect(-t, -u, C, D); B.save(); var g={ctx:a; a=B; p(); N(); J(); d(); s(); b.title&&(a.save(), a.font=24*(f/200)+"px Arial", a.fillStyle=b.colors.title, a.textAlign="center", a.fillText(b.title, 0, -f/4.25), a.restore()); b.units&&(a.save(), a.font=22*(f/200)+"px Arial", a.fillStyle=b.colors.units, a.textAlign="center", a.fillText(b.units, 0, f/3.25), a.restore()); A.i8d=!0; a=g.ctx; delete g.ctx} a.clearRect(-t, -u, C, D); a.save(); a.drawImage(A, -t, -u, C, D); if(Gauge.initialized)L(), K(), H|| (E.onready&&E.onready(), H=!0); else var e=setInterval(function(){Gauge.initialized&&(clearInterval(e), L(), K(), H|| (E.onready&&E.onready(), H=!0))}, 10); return this}; Gauge.initialized=!1;

(function(){var b=document, l=b.getElementsByTagName("head")[0], q=!1!=navigator.userAgent.toLocaleLowerCase().indexOf("msie"), v="@font-face {font-family:'Led'; src: url('fonts/digital-7-mono."+ (q?"eot":"ttf")+ "."); }", k=b.createElement("style"); k.type="text/css"; if(q) l.appendChild(k), l=k.styleSheet, l.cssText=v; else{try{k.appendChild(b.createTextNode(v))} catch(e){k.cssText=v} l.appendChild(k); l=k.styleSheet?k.styleSheet:k.sheet||b.styleSheets[b.styleSheets.length-1]} var g=setInterval(function(){if(b.body){clearInterval(g);

var e=b.createElement("div"); e.style.fontFamily="Led"; e.style.position="absolute"; e.style.height=e.style.width=0; e.style.overflow="hidden"; e.innerHTML="."; b.body.appendChild(e); setTimeout(function(){Gauge.initialized=!0; e.parentNode.removeChild(e)}, 250)}, 1)}() ; Gauge.Collection=[];

Gauge.Collection.get=function(b){if("string"===typeof b) for(var l=0, q=this.length; l<q; l++) {if((this[l].config.renderTo.tagName?this[l].config.renderTo:document.getElementById(this[l].config.renderTo)).getAttribute("id")==b) return this[l]} else return "number"===typeof b?this[b]:null}; function domReady(b){window.addEventListener?window.addEventListener("DOMContentLoaded", b, !1):window.attachEvent("onload", b)} domReady(function(){function b(b){for(var e=b[0], d=1, g=b.length; d<g; d++) e+=b[d].substr(0, 1).toUpperCase()+b[d].substr(1, b[d].length-1); return l=document.getElementsByTagName("canvas"), q=0, v=l.length; q<v; q++) if("canvas-gauge"===l[q].getAttribute("data-type")) {var

```

```

k=l[q],e={},g,p=parseInt(k.getAttribute("width"),10),w=parseInt(k.getAttribute("height"),10);e.renderTo=k;p&&(e.width=p);w&&(e.height=w);p=0;for(w=k.attributes.length;p<w;p++)if(g=k.attributes.item(p).nodeName,"data-type"!=g&&"data-"==g.substr(0,5)){var d=g.substr(5,g.length-5).toLowerCase().split("-");if(g=k.getAttribute(g))switch(d[0]){case "colors":d[1]&&(e.colors||(e.colors={}),"needle"==d[1]?(d=g.split(/\s+/),e.colors.needle=d[0]&&d[1]?{start:d[0],end:d[1]}:g):(d.shift(),e.colors[b(d)]=g));break;case "highlights":e.highlights||(e.highlights=[]);g=g.match(/(?:?:-?\d*\.)?(-?\d+){1,2}){2}(?:?:#|0x)?(?:[0-9A-F|a-f]){3,8}|rgba?\(.*\?)\)/g;for(var d=0,J=g.length;d<J;d++){var s=g[d].replace(/^\s+|\s+$/g,"").split(/\s+/),x={};s[0]&&"!"=s[0]&&(x.from=s[0]);s[1]&&"!"=s[1]&&(x.to=s[1]);s[2]&&"!"=s[2]&&(x.color=s[2]);e.highlights.push(x)}break;case "animation":d[1]&&(e.animation||(e.animation={}),"fn"==d[1]&&/^s*function\s*\(/.test(g)&&(g=eval("(+g+)"))),e.animation[d[1]]=g);break;default:d=b(d);if("onready"==d)continue;if("majorTicks"==d)g=g.split(/\s+/);else if("strokeTicks"==d||"glow"==d)g="true"==g?!0:1;else if("valueFormat"==d)if(g=g.split("."),2==g.length)g={"int":parseInt(g[0],10),dec:parseInt(g[1],10)};else continue;e[d]=g}}e=new Gauge(e);k.getAttribute("data-value")&&e.setRawValue(parseFloat(k.getAttribute("data-value")));k.getAttribute("data-onready")&&(e.onready=function(){eval(this.config.renderTo.getAttribute("data-onready"))});e.draw()});window.Gauge=Gauge;
<!-- Gauge Code Ends -->

```

function GetArduinoIO() // Συνάρτηση Javascript η οποία στέλνει XML αιτήματα προκειμένου να λάβει τις τιμές για τα gauges και τα leds.

```

{
nocache = "&nocache=" + Math.random() * 1000000;
var request = new XMLHttpRequest();
request.onreadystatechange = function()
{
if (this.readyState == 4) {
if (this.status == 200) {
if (this.responseXML != null) { //Επειδή ένα αρχείο xml έχει σταλθεί πίσω από το Arduino, η συνάρτηση τώρα ελέγχει εάν η απάντηση xml (XML response) περιέχει δεδομένα ή απλό κείμενο απάντησης.
//---code for gauges---//
//Λήψη τιμών από το xml αρχείο που στέλνει το Arduino για τα gauges.
data_val = this.responseXML.getElementsByTagName('analog')[0].childNodes[0].nodeValue;
data_val2 = this.responseXML.getElementsByTagName('analog')[1].childNodes[0].nodeValue;
data_val3 = this.responseXML.getElementsByTagName('analog')[2].childNodes[0].nodeValue;

```

```

//---code for leds---//

//Λήψη τιμών από το xml αρχείο που στέλνει το Arduino για τα LEDs.

        if
(this.responseXML.getElementsByTagName('LED')[0].childNodes[0].nodeValue === "on") {
            document.getElementById("LED1").innerHTML = "LED is ON
";

            LED1_state = 1;
        }
        else {
OFF ";
            document.getElementById("LED1").innerHTML = "LED is
            LED1_state = 0;
        }

        if
(this.responseXML.getElementsByTagName('LED')[1].childNodes[0].nodeValue === "on") {
            document.getElementById("LED2").innerHTML = "Watering
pot is ON ";

            LED2_state = 1;
        }
        else {
pot is OFF ";
            document.getElementById("LED2").innerHTML = "Watering
            LED2_state = 0;
        }
    }
}

request.open("GET", "ajax_inputs" + strLED1 + strLED2 + nocache, true); //
Αποστολή αιτήματος κάθε ένα δευτερόλεπτο.

request.send(null);

setTimeout('GetArduinoIO()', 1000);

strLED1 = "";

```

```

strLED2 = "";
strLED3 = "";

}

function GetButton1()
{
    if (LED1_state === 1) {
        LED1_state = 0;
        strLED1 = "&LED1=0";
    }
    else {
        LED1_state = 1;
        strLED1 = "&LED1=1";
    }
}

function GetButton2()
{
    if (LED2_state === 1) {
        LED2_state = 0;
        strLED2 = "&LED2=0";
    }
    else {
        LED2_state = 1;
        strLED2 = "&LED2=1";
    }
}

</script>

<style type="text/css">
    h2 {
        color: brown; //Χρώμα επικεφαλίδας.
    }

```

```

        </style>
</head>
<body onload="GetArduinoIO()">

    <div class="IO_box">

        <h2>Button</h2>

        <button type="button" id="LED1" onclick="GetButton1()">LED 1 is OFF
</button><br /><br />

        <button type="button" id="LED2" onclick="GetButton2()">LED 2 is OFF
</button><br /><br />

        <canvas id="an_gauge_1" data-title="temperature" data-units="Temp. &deg;C"
width="250" height="500" data-major-ticks="-40 -30 -20 -10 0 10 20 30 40 50 60 70 80
90 100 110 120 130 140 150" data-type="canv-gauge" data-min-value="-40" data-max-
value="150" data-highlights="-40 0 #4D89F2, 0 10 #25B8D9, 10 30 #0BB950, 30 40 #cc5, 40
150 #f33" data-onready="setInterval( function() {
Gauge.Collection.get('an_gauge_1').setValue(data_val);}, 200);"></canvas>

        <canvas id="an_gauge_2" data-title="humidity" data-units="hum. %"
width="250" height="500" data-major-ticks="0 10 20 30 40 50 60 70 80 90 100" data-
type="canv-gauge" data-min-value="0" data-max-value="100" data-highlights="0 0 #4D89F2,
0 10 #25B8D9, 10 30 #0BB950, 30 40 #cc5, 40 150 #f33" data-onready="setInterval(
function() { Gauge.Collection.get('an_gauge_2').setValue(data_val2);}, 200);"></canvas>

        <canvas id="an_gauge_3" data-title="soil moisture" data-units="moi. %"
width="250" height="500" data-major-ticks="-40 -30 -20 -10 0 10 20 30 40 50 60 70 80
90 100 110 120 130 140 150" data-type="canv-gauge" data-min-value="-40" data-max-
value="150" data-highlights="-40 0 #4D89F2, 0 10 #25B8D9, 10 30 #0BB950, 30 40 #cc5, 40
150 #f33" data-onready="setInterval( function() {
Gauge.Collection.get('an_gauge_3').setValue(data_val3);}, 200);"></canvas>

    </div>

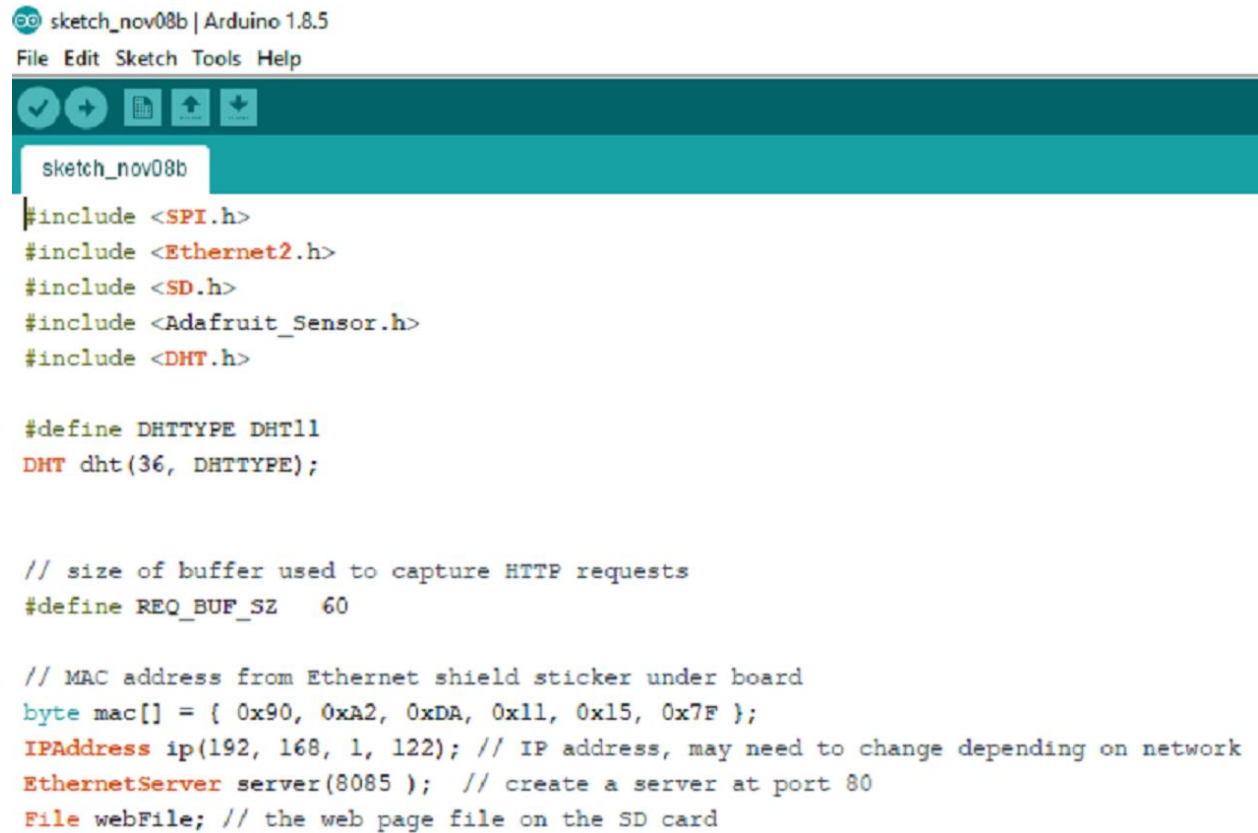
    <p class="small_text">By G.SOFIANOS</p>

</body>
</html>

```



Παρακάτω παραθέτουμε κάποια screenshot από τον κώδικα του Arduino, του html αρχείου και από τον browser.



```
sketch_nov08b
#include <SPI.h>
#include <Ethernet2.h>
#include <SD.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>

#define DHTTYPE DHT11
DHT dht(36, DHTTYPE);

// size of buffer used to capture HTTP requests
#define REQ_BUF_SZ 60

// MAC address from Ethernet shield sticker under board
byte mac[] = { 0x90, 0xA2, 0xDA, 0x11, 0x15, 0x7F };
IPAddress ip(192, 168, 1, 122); // IP address, may need to change depending on network
EthernetServer server(8085 ); // create a server at port 80
File webFile; // the web page file on the SD card
```

*Εικόνα 11.sketch*

Στην παρακάτω εικόνα φαίνεται ένα κομμάτι του κώδικα της ιστοσελίδας το οποίο απαρτίζεται από τις γλώσσες προγραμματισμού html, Javascript και CSS. Το πρόγραμμα που χρησιμοποιήθηκε για την συγγραφή του κώδικα είναι το sublime text.

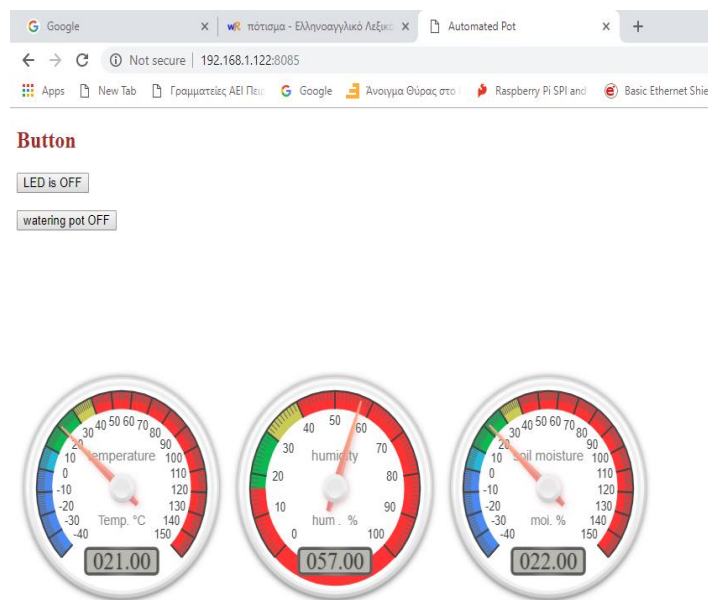
```

C:\Users\g_sof\Desktop\Automated Pot\8_nov_18\index.htm - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
index.htm
37 g.substr(0,5)){var d=g.substr(5,g.length-5).toLowerCase().split("-");if(g.k.getAttribute(g))switch(d[0]){case "colors":d[1]&&(e.colors||(e.
d[1]>{start:d[0],end:d[1]}:g):(d.shift(),e.colors[b(d)]:g));break;case "highlights":e.highlights||(e.highlights=[]);g.g.match(/(?:-?d
\))/g);for(var d=0,J=g.length;d<J;d++){var s=g[d].replace(/\s+|\s+$/g,"").split(/\s+/),x={};
38 s[0]&&"!-s[0]&&(x.from-s[0]);s[1]&&"!-s[1]&&(x.to-s[1]);s[2]&&"!-s[2]&&(x.color-s[2]);e.highlights.push(x);break;case "animation":d[1]&&
test(g)&&(g.eval("(+g+"))),e.animation[d[1]]:g);break;default:d-b(d);if("onready"--d)continue;if("majorTicks"--d)g.g.split(/\s+/);else
valueFormat"--d)if(g.g.split("."),2-g.length)g.g={"int":parseInt(g[0],10),dec:parseFloat(g[1],10)};else continue;
39 e[d]g)e=new Gauge(e);k.getAttribute("data-value")&&e.setRawValue(parseFloat(k.getAttribute("data-value")));k.getAttribute("data-onready")
data-onready"}));e.draw()});window.Gauge=Gauge;
40 <!-- Gauge Code Ends -->
41
42
43
44 function GetArduinoIO()
45 {
46     nocache = "&nocache=" + Math.random() * 1000000;
47     var request = new XMLHttpRequest();
48     request.onreadystatechange = function()
49     {
50         if (this.readyState == 4) {

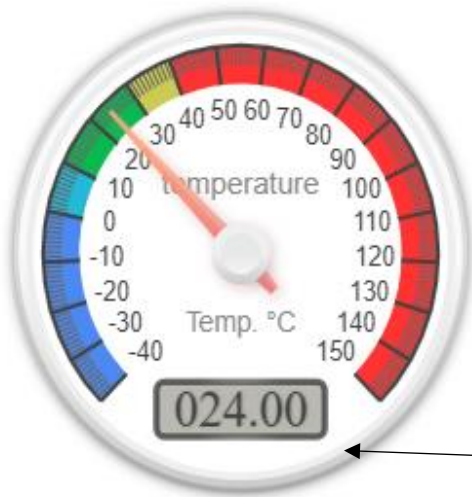
```

Εικόνα 12. κώδικας σελίδας

Σε αυτήν την εικόνα βλέπουμε το περιβάλλον της ιστοσελίδας, όπου μπορούμε να διακρίνουμε τα τρία gauges για την υγρασία, θερμοκρασία και υγρασία εδάφους καθώς και τα δύο κουμπιά όπου το ένα χρησιμοποιείται για την ενεργοποίηση του LED και το άλλο για την λειτουργία της αντλίας νερού.

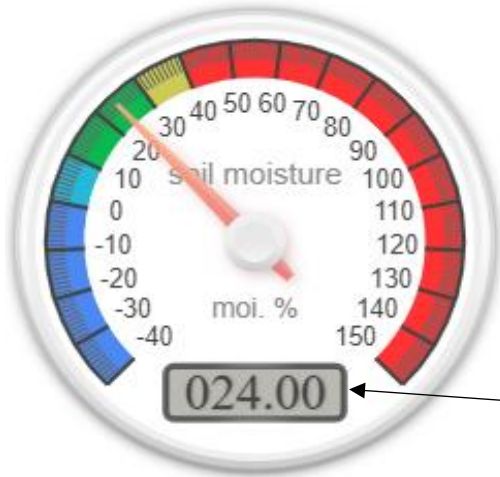


Εικόνα 13. Περιβάλλον Ιστοσελίδας



Ένδειξη  
θερμοκρασίας

Εικόνα 15. Όργανο ένδειξης θερμοκρασίας



Ένδειξη υγρασίας  
εδάφους με ψηφιακή  
μορφή.

Εικόνα 14. Όργανο ένδειξης υγρασίας εδάφους

## ΚΕΦΑΛΑΙΟ 5 – Επίλογος

Τα πλεονεκτήματα χρήσεως της αυτόματης γλάστρας είναι εμφανή τόσο στον άνθρωπο όσο και στο φυτό. Με την απομακρυσμένη παρακολούθηση της κατάστασης του φυτού και του αυτόματου ποτίσματος ο άνθρωπος δεν χρειάζεται να παρευρίσκεται στον χώρο που είναι το φυτό για να το φροντίσει. Έτσι μπορεί να αξιοποιήσει τον χρόνο που θα αφιέρωνε για την φροντίδα του φυτού, σε κάποια άλλη ασχολία. Όσον αφορά το φυτό δημιουργούνται καλύτερες συνθήκες για την ανάπτυξή του διότι ο άνθρωπος έχει την δυνατότητα να γνωρίζει ακριβώς την υγρασία του εδάφους. Αυτό σημαίνει ότι μπορεί να ποτίζει το φυτό πιο συχνά ή πιο αραιά με βάση τα επίπεδα υγρασίας που είναι καλύτερα για την ανάπτυξή του.

Ένα από τα αρνητικά της αυτόματης γλάστρας είναι ότι το Arduino συνδέεται με το router μέσω καλωδίου Ethernet πράγμα που σημαίνει ότι θα πρέπει να έχουμε κοντά ένα router και ένα καλώδιο εκτεθειμένο στις περιβαλλοντικές συνθήκες που θα επικρατούν στον κήπο .

Αυτό βέβαια λύνεται με την χρήση μίας Wi-Fi πλακέτας έτσι ώστε να επικοινωνεί το Arduino με το router ασύρματα. Μία ακόμη βελτίωση που θα μπορούσαμε να κάνουμε στην γλάστρα είναι η προσθήκη κάμερας έτσι ώστε να παρατηρούμε και οπτικά την κατάσταση του φυτού. Με αυτόν τον τρόπο θα μπορούσαμε να εντοπίσουμε τυχόν αρρώστιες που θα εμφάνιζε το φυτό, παρατηρώντας τα φύλλα του, πάνω στα οποία εμφανίζονται συνήθως τα σημάδια κάποιας αρρώστιας. Ακόμη θα μπορούσαμε να τοποθετήσουμε δίπλα στην γλάστρα μηχανισμό ραντίσματος ενός φαρμάκου, ώστε να είναι εφικτός ο απομακρυσμένος ψεκασμός του φυτού σε περίπτωση κάποιας ασθένειας.

## Βιβλιογραφία

- 1 «Arduino Mega 2560 Rev 3,» [Ηλεκτρονικό]. Available: <https://store.arduino.cc/arduino-mega-2560-rev3>. [Πρόσβαση 8 1 2019].  
«Arduino - Libraries,» [Ηλεκτρονικό]. Available: <https://www.arduino.cc/en/guide/libraries>. [Πρόσβαση 8 1 2019].
- 2 «Arduino - Libraries,» [Ηλεκτρονικό]. Available: <https://www.arduino.cc/en/reference/libraries>. [Πρόσβαση 8 1 2019].
- 3 «Arduino - Libraries,» [Ηλεκτρονικό]. Available: <https://www.arduino.cc/en/reference/libraries>. [Πρόσβαση 8 1 2019].
- 4 ArduinoautoMotive, «Θερμοκαρσία και Υγρασία - DHT 11 - Ardumotive Arduino GreekPlaygroun,» [Ηλεκτρονικό]. Available: <http://www.ardumotive.com/how-to-use-dht-11-sensor-gr.html>. [Πρόσβαση 4 12 2018].
- 5 «arduino-soil-moisture-sensor,» 31 7 2018. [Ηλεκτρονικό]. Available: <http://www.circuitstoday.com/arduino-soil-moisture-sensor>. [Πρόσβαση 12 11 2018].
- 6 O. Lathrop, «arduino-purpose-of-the-diode-and-capacitor-in-this-motor-circuit,» [Ηλεκτρονικό]. Available: <https://electronics.stackexchange.com/questions/95140/purpose-of-the-diode-and-capacitor-in-this-motor-circuit>. [Πρόσβαση 27 11 2018].
- 7 «Arduino Reference,» Arduino, [Ηλεκτρονικό]. Available: <https://www.arduino.cc/reference/en/language/structure/sketch/setup/>. [Πρόσβαση 12 11 18].
- 8 Arduino, «Arduino-begin,» [Ηλεκτρονικό]. Available: <https://www.arduino.cc/en/Serial/Begin>. [Πρόσβαση 12 11 18].
- 9 Βικιπαίδεια, «Ολοκληρωμένο περιβάλλον ανάπτυξης - Βικιπαίδεια,» [Ηλεκτρονικό]. Available: [https://el.wikipedia.org/wiki/%CE%9F%CE%BB%CE%BF%CE%BA%CE%BB%CE%B7%CF%81%CF%89%CE%BC%CE%AD%CE%BD%CE%BF\\_%CF%80%CE%B5%CF%81%CE%B9%CE%B2%CE%AC%CE%BB%CE%BB%CE%BF%CE%](https://el.wikipedia.org/wiki/%CE%9F%CE%BB%CE%BF%CE%BA%CE%BB%CE%B7%CF%81%CF%89%CE%BC%CE%AD%CE%BD%CE%BF_%CF%80%CE%B5%CF%81%CE%B9%CE%B2%CE%AC%CE%BB%CE%BB%CE%BF%CE%)

- BD\_%CE%B1%CE%BD%CE%AC%CF%80%CF%84%CF%85%CE%BE%CE%B7%CF%82. [Πρόσβαση 15 1 2019].
- 10 «Arduino-SPI,» Arduino, 08 11 18. [Ηλεκτρονικό]. Available: <https://www.arduino.cc/en/Reference/SPI>.
- 11 «Basic Ethernet Shield Arduino Web Server,» 21 1 2013. [Ηλεκτρονικό]. Available: <https://startingelectronics.org/tutorials/arduino/ethernet-shield-web-server-tutorial/basic-web-server/>. [Πρόσβαση 12 11 2018].
- 12 «Arduino - SD,» Arduino, 8 11 18. [Ηλεκτρονικό]. Available: <https://www.arduino.cc/en/Reference/SD>.
- 13 «Moisture Sensor - eProLabs Wiki,» [Ηλεκτρονικό]. Available: [https://wiki.eprolabs.com/index.php?title=Moisture\\_Sensor](https://wiki.eprolabs.com/index.php?title=Moisture_Sensor). [Πρόσβαση 04 12 2018].
- 14 [Ηλεκτρονικό]. Available: <https://store.arduino.cc/arduino-mega-2560-rev3>. [Πρόσβαση 1 2019].
- 15 J. Duckett, «HTML & CSS design and build websites,» Indianapolis, John Wiley and Sons, 2014, p. 500.