

# Fielduino (Digital Farming)

OPEN ROBOTICS

1ΟΣ ΠΑΝΕΛΛΗΝΙΟΣ ΔΙΑΓΩΝΙΣΜΟΣ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΡΟΜΠΟΤΙΚΗΣ & PHYSICAL COMPUTING ΑΝΟΙΧΤΩΝ ΤΕΧΝΟΛΟΓΙΩΝ

### Στοιχεία έργου

Όνομα έργου: Fielduino (Digital Farming)

Όνομα σχολείου: ΙΕΚ Μεσολογγίου (<http://iekmesol.sch.gr/>)

Βαθμίδα εκπαίδευσης: Μεταδευτεροβάθμια

Link στο github: <https://github.com/iekmesol/Fielduino>

Link έργου: [https://robotics.ellak.gr/robotics\\_participant/diek-mesolongiou/](https://robotics.ellak.gr/robotics_participant/diek-mesolongiou/)

**Δημιουργοί:** Ειδικότητα **Τεχνικός Η/Υ** στο πλαίσιο του μαθήματος Διαθεματική εργασία (εργαστήριο) του Δ' Εξαμήνου - 2019Α

Μαθητευόμενοι:

- Κούστας Γρηγόριος
- Παπαλόπουλος Ιωάννης
- Ρεβέκη Γεωργία
- Τίτας Θεόδωρος
- Μωραΐτης Γεώργιος

Εκπαιδευτής: Καραχρήστος Χριστόφορος

**Άδεια χρήσης:** GNU General Public License v3.0

## Μέρος 1

### Εισαγωγή

Ο πρωτογενής τομέας συνδέεται καθημερινά όλο και περισσότερο με την ανάπτυξη των νέων τεχνολογιών. Μεγάλες εταιρείες του χώρου της τεχνολογίας διαβλέπουν το μέλλον της «έξυπνης παραγωγής και καλλιέργειας». Η έξυπνη γεωργία (smart farming) αποτελεί μια καινοτόμα προσέγγιση για τη διαχείριση των γεωργικών δραστηριοτήτων με χρήση νέων τεχνολογιών. Μπορεί να οδηγήσει σε σημαντική μείωση του κόστους μαζί με σημαντική ποιοτική και ποσοτική βελτίωση της γεωργικής παραγωγής. Αξιοποιεί τις σύγχρονες τεχνολογίες και την επιστημονική γνώση με τέτοιο τρόπο ώστε να λαμβάνονται οι πιο κατάλληλες αποφάσεις σχετικά με την παραγωγή, ωφελώντας έτσι την οικονομία και το περιβάλλον. Ένα τέτοιο έργο μπορεί να βοηθήσει την τοπική κοινωνία του Μεσολογγίου, της ευρύτερης περιοχής αλλά και των περισσότερων περιοχών της χώρας μας οι οποίες είναι αγροτικές περιοχές και η οικονομία τους στηρίζεται σε μεγάλο βαθμό στην αγροτική παραγωγή.

Μερικά από τα βασικά πλεονεκτήματά της είναι τα εξής:

- Μείωση στη χρήση φυτοφαρμάκων, λιπασμάτων και νερού άρδευσης
- Αποτελεσματική αντιμετώπιση των απειλών που επηρεάζουν την παραγωγή
- Ποιοτική και ποσοτική βελτίωση της παραγωγής
- Προστιθέμενη αξία στα γεωργικά προϊόντα
- Μείωση των περιβαλλοντικών επιπτώσεων
- Συμμόρφωση με το ρυθμιστικό πλαίσιο της ΕΕ

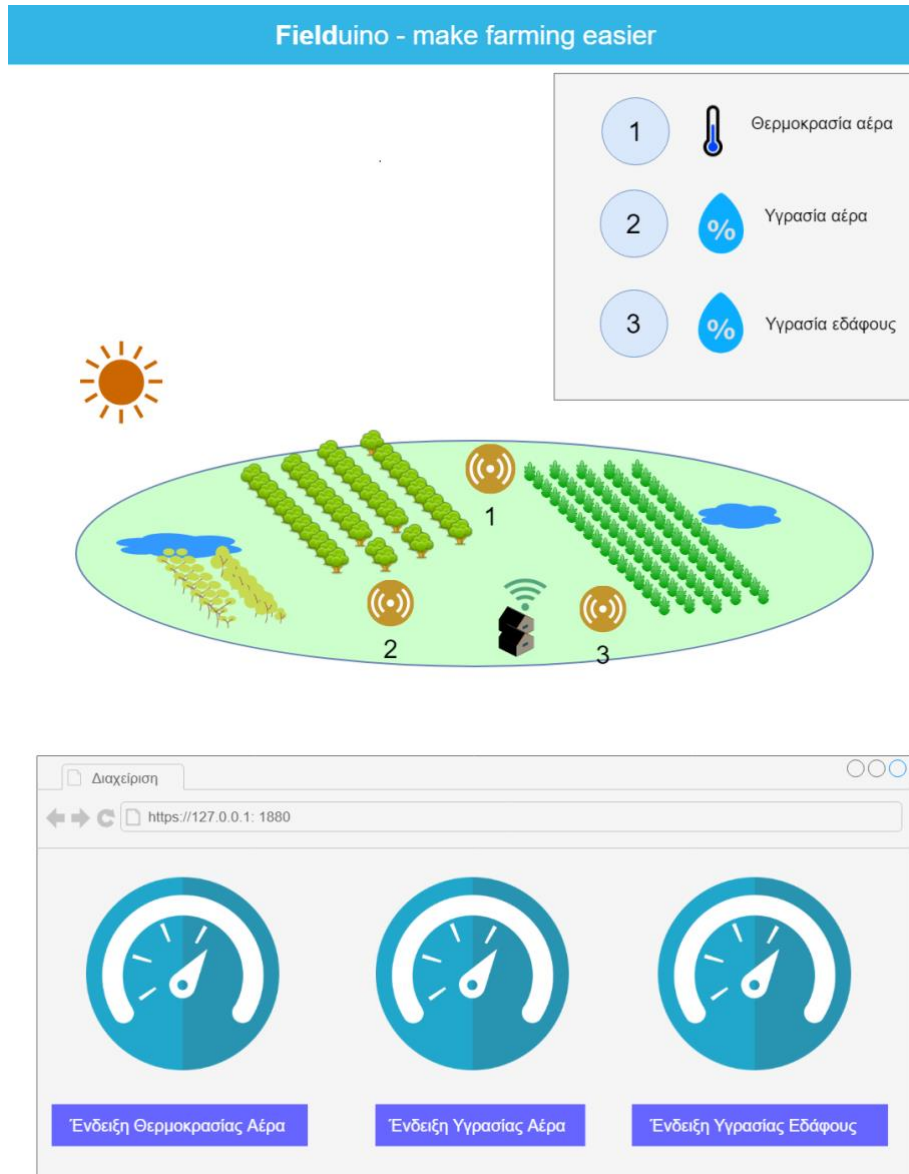
### Περιγραφή του έργου

Το προτεινόμενο έργο στόχο έχει να δώσει τη δυνατότητα σε κάποιον που ασχολείται με αγροτικές εργασίες να αναπτύξει ένα σύστημα μέσω του οποίου θα μπορεί να παρακολουθεί εξ αποστάσεως μέσω διαδικτύου και καθ' όλη τη διάρκεια της ημέρας, διάφορες περιβαλλοντικές (και μη) τιμές που τον ενδιαφέρουν σχετικά με τις καλλιέργειές του και σχετίζονται με μετρήσεις από αισθητήρες. Στη συγκεκριμένη υλοποίηση χρησιμοποιήθηκαν οι μετρήσεις από τρεις περιβαλλοντικούς δείκτες οι οποίες είναι:

- Θερμοκρασία αέρα,
- Υγρασία αέρα,
- Υγρασία εδάφους.

Θα πρέπει να τονιστεί πως αυτοί οι μετρούμενοι δείκτες είναι ενδεικτικοί και μπορούν πολύ εύκολα να αντικατασταθούν από άλλες μετρήσεις ή ακόμη και να επεκταθούν. Παράλληλα θα πρέπει να τονιστεί πως το συγκεκριμένο έργο εστιάζει στην δημιουργία ενός οδηγού υλοποίησης ενός τέτοιου συστήματος και όχι της πλήρους ανάπτυξης ενός τελικού παραγωγικού ολοκληρωμένου συστήματος.

Αυτό σημαίνει πως για τις μετρήσεις χρησιμοποιούνται οι τιμές που λαμβάνονται από ένα σημείο ενδιαφέροντος σε ένα χωράφι δηλαδή καταγράφονται μια μέτρηση από κάθε αισθητήρα. Το συγκεκριμένο έργο μπορεί να επεκταθεί και να συλλέγει τιμές από διάσπαρτους αισθητήρες σε κάποια



γεωργική έκταση και να εμπλουτιστεί κατάλληλα με κάποιο μοντέλο επεξεργασίας δεδομένων που θα δίνει τη δυνατότητα στο χρήστη να εξάγει περισσότερο χρήσιμα συμπεράσματα. Για παράδειγμα μπορεί ο ενδιαφερόμενος να συλλέγει τιμές υγρασίας εδάφους από διάφορα σημεία και να τα επεξεργάζεται κατάλληλα ώστε να εξάγει μέγιστες και ελάχιστες τιμές, μέσους όρους τιμών κτλ.

### Χρονοπρογραμματισμός έργου

Την ανάπτυξη του έργου τη χωρίσαμε σε 3 πακέτα εργασίας. Στο πρώτο πακέτο εργασίας ασχοληθήκαμε με την ανάπτυξη του υλικού το οποίο απαιτείται και το οποίο περιλαμβάνει τη

διασύνδεση των αισθητήρων με τον μικροελεγκτή Arduino UNO έτσι ώστε να καταγράφονται οι τιμές από τους αισθητήρες. Στο δεύτερο πακέτο εργασίας ασχοληθήκαμε με την διασύνδεση του προηγούμενου συστήματος με τη διεπαφή του χρήστη η οποία θα είναι μια web-based εφαρμογή και από την οποία ο χρήστης θα μπορεί να βλέπει τις τιμές που καταγράφουν οι αισθητήρες αλλά και να αλληλοεπιδρά. Τα δύο αυτά πακέτα εργασίας ολοκληρώθηκαν παράλληλα καθώς αποτελούν ανεξάρτητα συστήματα. Στο τρίτο πακέτο εργασίας το οποίο ξεκίνησε αφού ολοκληρώθηκαν τα δύο πρώτα πακέτα εργασίας προσπαθήσαμε να τα συνδέσουμε όλα αυτά σε ένα ενιαίο σύστημα του οποίου τα επιμέρους συστήματα επικοινωνούν μεταξύ τους μέσω ασύρματης σύνδεσης. Τέλος ασχοληθήκαμε με την τεκμηρίωση του project.

### **Κόστος υλοποίησης**

Arduino UNO: €7-20

Αισθητήρας Θερμοκρασίας-Υγρασίας DHT11: €4.00

SparkFun Soil Moisture Sensor (with Screw Terminals): €6.99

## Μέρος 2

### Υλοποίηση

Παρακάτω αναλύονται τα υλικά μέρη που χρησιμοποιήθηκαν:

#### 1. Μικροελεγκτής Arduino UNO και προγραμματιστικό περιβάλλον Arduino IDE

Ένας μικροελεγκτής είναι μια ηλεκτρονική πλακέτα με ένα chip ενσωματωμένο, το οποίο μπορεί να προγραμματιστεί ώστε να κάνει πολλά διαφορετικά πράγματα, όπως να διαβάζει πληροφορίες από αισθητήρες (για παράδειγμα την ανίχνευση κίνησης όταν κάποιος κινείται μέσα σε ένα δωμάτιο κ.τ.λ.). Όλα αυτά τα δεδομένα εισόδου μπορούν να χρησιμοποιηθούν για να ελεγχθούν στη συνέχεια διάφορες συσκευές ή απλά για να αποθηκευτούν και να απεικονιστούν σαν δεδομένα εξόδου (για παράδειγμα όταν ανιχνεύεται κίνηση από κάποιον αισθητήρα ανίχνευσης κίνησης, ένα λαμπάκι ανάβει για να προειδοποιήσει).

Το Arduino UNO είναι ένας συνδυασμός ενός ανοικτού κώδικα μικροελεγκτή, ο οποίος βασίζεται σε ένα chip και αποτελείται από διαφορετικά μέρη, και μιας πλακέτας πάνω στην οποία είναι συγκολλημένα αυτά. Ο συνδυασμός αυτός δημιουργεί έναν μικρό υπολογιστή.

Το Arduino απαρτίζεται από δύο μέρη:

- Την πλακέτα Arduino, η οποία αποτελεί το υλικό (hardware) πάνω στο οποίο θα δουλεύετε και θα μπορείτε να προγραμματίσετε.
- Το Arduino IDE (Integrated Development Environment - Περιβάλλον προγραμματισμού) το οποίο αποτελεί το κομμάτι λογισμικού (software) που υποστηρίζει το Arduino και που τρέχει στον υπολογιστή σας.

Στο σύνδεσμο που ακολουθεί μπορείτε να βρείτε πληροφορίες σε μορφή pdf για το Arduino UNO και το Arduino IDE: <https://github.com/iekmesol/Fielduino/blob/master/Arduino.zip>

Στο video που ακολουθεί μπορείτε να παρακολουθήσετε μια σύντομη παρουσίαση της πλακέτας Arduino Uno: <https://youtu.be/NVNMVjMSWIQ>

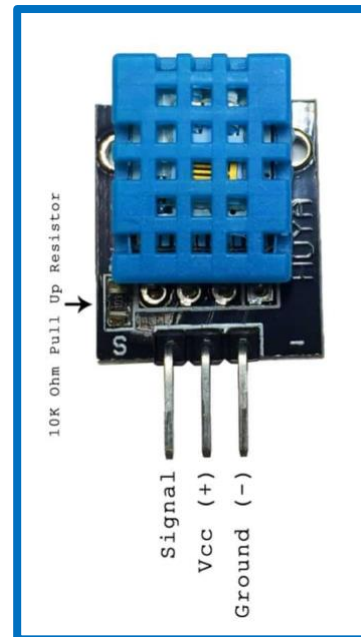
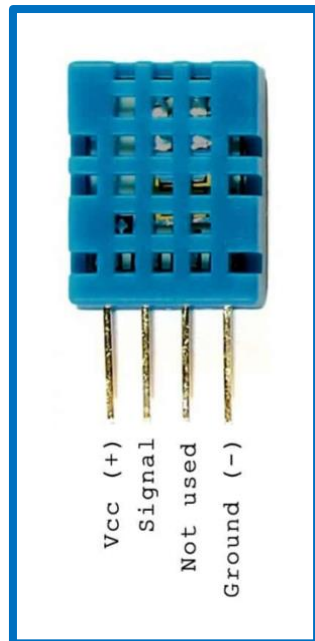
Στο video που ακολουθεί μπορείτε να παρακολουθήσετε μια σύντομη παρουσίαση των βασικών εξαρτημάτων της πλακέτας Arduino UNO: <https://youtu.be/wjAj0KmcDQA>

Στα videos που ακολουθούν μπορείτε να βρείτε οδηγίες σχετικές με την εγκατάσταση του Arduino IDE (Windows και Mac):

- Windows: <https://youtu.be/KtwBAKj06gU>
- Macintosh: [https://youtu.be/OrYXuBXe-\\_I](https://youtu.be/OrYXuBXe-_I)

## 2. Θερμοκρασία αέρα – Υγρασία Αέρα (DHT11 αισθητήρας)

Ο DHT-11 είναι ένας βασικός, χαμηλού κόστους, αισθητήρας για την εύρεση της σχετικής υγρασίας και της θερμοκρασίας στον χώρο. Υπάρχουν 2 διαφορετικοί τύποι DHT11 αισθητήρα. Ο ένας τύπος έχει 4 ακίδες (pins) και ο άλλος 3 ακίδες (pins) όπως φαίνονται στην εικόνα:



Στην υλοποίησή μας θα χρησιμοποιήσουμε τον DHT-11 με τις 3 ακίδες.

### Τεχνικές πληροφορίες:

Πηγή: 3-5.5V

Μέγιστο ρεύμα: 2.5mA

Υγρασία: 20-90% RH\*, ακρίβεια  $\pm 5\%$  RH

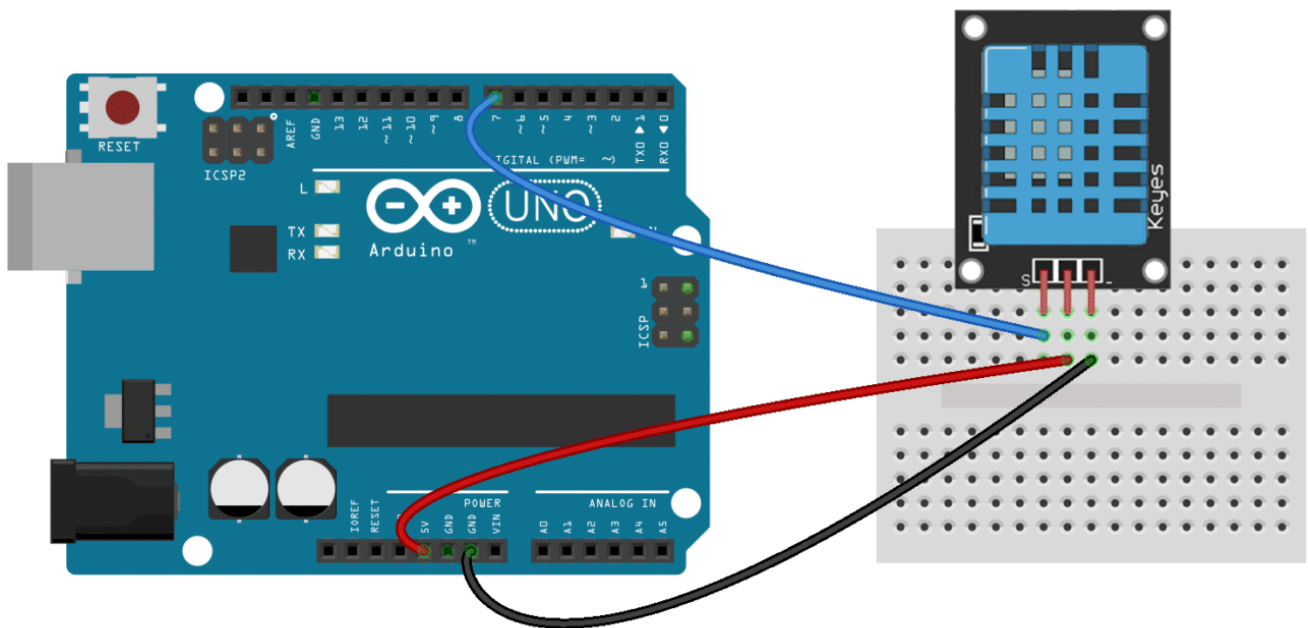
Θερμοκρασία (εύρος): 0 έως 50°C, ακρίβεια  $\pm 2\%$  °C

\*RH – Relative Humidity (Σχετική υγρασία): Ο DHT11 αισθητήρας μετράει την σχετική υγρασία. Η σχετική υγρασία είναι η ποσότητα υδρατμών στον αέρα σε σχέση με το σημείο κορεσμού της ποσότητας των υδρατμών του αέρα. Στο σημείο κορεσμού ο υδρατμός αρχίζει να συμπυκνώνεται και να συσσωρεύεται σε επιφάνειες που σχηματίζουν δροσιά. Το σημείο κορεσμού αλλάζει με τη θερμοκρασία του αέρα. Ο ψυχρός αέρας μπορεί να συγκρατήσει λιγότερους υδρατμούς πριν επέλθει κορεσμός. Ο θερμός αέρας μπορεί να συγκρατήσει περισσότερους υδρατμούς πριν κορεσθεί. Η σχετική

υγρασία εκφράζεται σε ποσοστό. Στο 100% RH, συμβαίνει συμπύκνωση και στο 0% RH, ο αέρας είναι εντελώς στεγνός.

- **Σύνδεση DHT-11 3 ακίδων με Arduino UNO**

**Βήμα 1:** Συνδέστε την ακίδα S (Signal) στην ψηφιακή ακίδα 7 του Arduino UNO. Μπορείτε να χρησιμοποιήσετε όποια ψηφιακή ακίδα επιθυμείτε από το Arduino UNO στην περίπτωση που η 7 είναι πιασμένη.



**Βήμα 2:** Συνδέστε την vcc ακίδα (μεσαία ακίδα) με την 5v ακίδα τροφοδοσίας του Arduino UNO.

**Βήμα 3:** Συνδέστε την ground ακίδα με την GND ακίδα του Arduino UNO.

- **Προγραμματισμός DHT-11 σε Arduino UNO**

Πριν ξεκινήσετε τον προγραμματισμό του DHT-11 θα πρέπει να εγκαταστήσετε την βιβλιοθήκη DHTLib. Η βιβλιοθήκη περιλαμβάνει όλες τις συναρτήσεις που είναι απαραίτητες για τη λήψη της υγρασίας αέρα και της θερμοκρασίας από τον αισθητήρα.

Κατεβάστε τη βιβλιοθήκη από έναν από τους παρακάτω συνδέσμους:

- [Σύνδεσμος 1](#)
- [Σύνδεσμος 2](#)

Για να εγκαταστήσετε τη βιβλιοθήκη ανοίξτε το Arduino IDE. Στη συνέχεια πηγαίνατε Σχέδιο → Συμπερίληψη βιβλιοθήκης → Προσθήκη βιβλιοθήκης .ZIP και επιλέξτε το DHTLib.zip αρχείο.

Φορτώστε το παρακάτω πρόγραμμα στο Arduino UNO:



```
#include <dht.h>
dht DHT;
#define DHT11_PIN 7 //Καθορισμός ακίδας S

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

void loop()
{
  int chk = DHT.read11(DHT11_PIN);
  Serial.print("Temperature = ");
  Serial.println(DHT.temperature); //Επιστρεφόμενη τιμή θερμοκρασίας
  Serial.print("Humidity = ");
  Serial.println(DHT.humidity); //Επιστρεφόμενη τιμή υγρασίας αέρα
  delay(1000); // Διάβασμα τιμών ανά ένα δευτερόλεπτο
}
```

Τέλος ανοίξτε την Παρακολούθηση σειριακής (Serial Monitor). Αν όλα έχουν πάει καλά θα πρέπει να βλέπετε τις παρακάτω ενδείξεις να ανανεώνονται ανά ένα δευτερόλεπτο.

**Temperature = 26.00**

**Humidity = 37.00**

### 3. Υγρασία εδάφους (Soil Moisture Sparkfun)

Ο **SparkFun Soil Moisture Sensor** είναι ένας απλός αισθητήρας για τη μέτρηση της υγρασίας στο έδαφος. Τα δύο άκρα του λειτουργούν ως ανιχνευτές για τον αισθητήρα, ενώ λειτουργούν μαζί ως μεταβλητές αντιστάσεις. Όσο περισσότερο νερό υπάρχει στο έδαφος, τόσο καλύτερη είναι η αγωγιμότητα μεταξύ των άκρων και θα προκύψει χαμηλότερη αντίσταση και υψηλότερη έξοδος SIG.

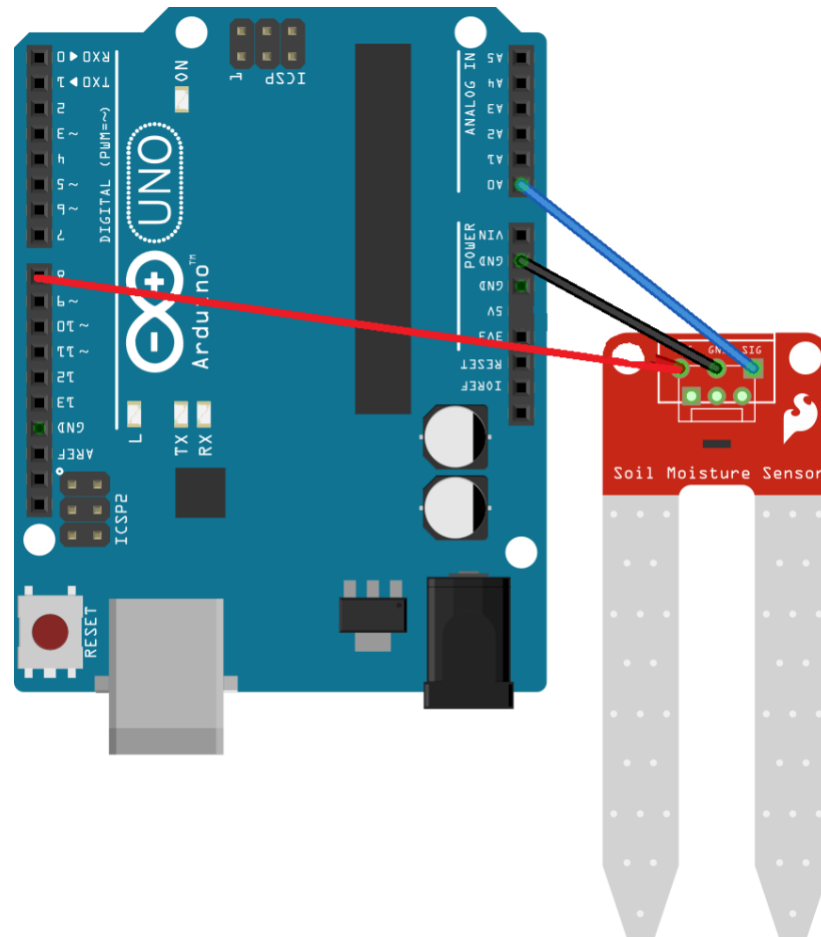
Στην υλοποίησή μας χρησιμοποιήσαμε την εκδοχή του αισθητήρα με τις βίδες (**SparkFun Soil Moisture Sensor - with Screw Terminals**)

Τεχνικές πληροφορίες:

Πηγή: 3-5.5V

- **Σύνδεση Soil Moisture Sparkfun με Arduino UNO**

Ο αισθητήρας είναι πολύ εύκολος στη σύνδεσή του με το Arduino UNO.



**Βήμα 1:** Συνδέστε την vcc ακίδα (μεσαία ακίδα) με την ψηφιακή ακίδα 8 (και όχι με την 5v ακίδα τροφοδοσίας) του Arduino UNO. Ο λόγος για τον οποίο τροφοδοτούμε τον αισθητήρα υγρασίας με μια ψηφιακή ακίδα (HIGH voltage ~5v) και όχι κατευθείαν από πηγή 3.3v ή 5v είναι το διαφορετικό εύρος τιμών μεταξύ των δύο τάσεων. Αν τροφοδοτήσουμε τον αισθητήρα με 3.3v και συνδέσουμε το vcc στην πηγή 3.3v τότε το εύρος τιμών που θα πάρουμε είναι από 0 (στεγνό) μέχρι ~550 (κορεσμένο με υγρασία). Αντίστοιχα αν τροφοδοτήσουμε τον αισθητήρα με 5v και συνδέσουμε το vcc στην πηγή 5v τότε το εύρος τιμών που θα πάρουμε είναι από 0 (στεγνό) μέχρι ~880 (κορεσμένο με υγρασία). Για να μην αντιμετωπίσετε προβλήματα συμβατότητας κατά τον προγραμματισμό επιλέξτε να συνδέσετε το vcc σε κάποια ψηφιακή ακίδα (με HIGH voltage 5v).

**Βήμα 2:** Συνδέστε την ground ακίδα με την GDN ακίδα του Arduino UNO.

**Βήμα 3:** Η ακίδα SIG παρέχει αναλογικό σήμα και μπορεί να συνδεθεί με όποια αναλογική ακίδα (στην υλοποίησή μας με το A0). Η τιμή που παίρνετε πίσω εξαρτάται από την τάση που εφαρμόζετε στον αισθητήρα.

- **Προγραμματισμός Soil Moisture Sparkfun σε Arduino UNO**

Στον προηγούμενο κώδικα προσθέστε τις παρακάτω εντολές (έντονη επισήμανση)

```
#include <dht.h>
dht DHT;
#define DHT11_PIN 7 //Καθορισμός ακίδας S
int val = 0; //value for storing moisture value
int soilPin = A0;//Δήλωση μεταβλητής για την ακίδα SIG
int soilPower = 8;//Μεταβλητή για τη διαχείριση της τάσης του αισθητήρα
void setup(){
  Serial.begin(9600);
  pinMode(soilPower, OUTPUT);//Ορίζουμε την ψηφιακή ακίδα 8 σαν έξοδο
  digitalWrite(soilPower, LOW);//Ορίζουμε σε LOW έτσι ώστε καμία τάση να μην διέρχεται
}
void loop()
{
  int chk = DHT.read11(DHT11_PIN);
  Serial.print("Temperature = ");
  Serial.println(DHT.temperature); //Επιστρεφόμενη τιμή θερμοκρασίας
  Serial.print("Humidity = ");
  Serial.println(DHT.humidity); //Επιστρεφόμενη τιμή υγρασίας αέρα
  Serial.print("Soil Moisture = ");
  Serial.println(readSoil()); ///Επιστρεφόμενη τιμή υγρασίας εδάφους
  delay(1000); // Διάβασμα τιμών ανά ένα δευτερόλεπτο
}
int readSoil()
{
  digitalWrite(soilPower, HIGH);//D8 "On"
  delay(10);//wait 10 milliseconds
```

```
val = analogRead(soilPin);//Διαβάσε την SIG τιμή απο τον αισθητήρα  
digitalWrite(soilPower, LOW);//D8 "Off"  
return val;//επεστρεψε την τρέχουσα τιμή του αισθητήρα  
}
```

Τέλος ανοίξτε την Παρακολούθηση σειριακής (Serial Monitor). Αν όλα έχουν πάει καλά θα πρέπει να βλέπετε τις παρακάτω ενδείξεις να ανανεώνονται ανά ένα δευτερόλεπτο.

**Temperature = 25.00**

**Humidity = 36.00**

**Soil Moisture = 0**

#### 4. Προγραμματισμός δειγματοληψίας τιμών αισθητήρων

Μπορείτε να ορίσετε το χρονικό διάστημα μεταξύ των μετρήσεων που παίρνει το σύστημα από τους αισθητήρες. Επειδή η συνάρτηση `delay(1000)` που χρησιμοποιείται δεν είναι η καλύτερη λύση αφαιρέθηκε η εν λόγω συνάρτηση και αντικαταστάθηκε από το παρακάτω κώδικα:

```
void loop()  
{  
    const unsigned long xMinutes = 1* 1000UL;  
    static unsigned long lastSampleTime = 0 - xMinutes;  
    unsigned long now = millis();  
    if (now - lastSampleTime >= xMinutes)  
    {  
        lastSampleTime += xMinutes;  
        int chk = DHT.read11(DHT11_PIN);  
        Serial.print("Temperature = ");  
        Serial.println(DHT.temperature); //Επιστρεφόμενη τιμή θερμοκρασίας  
        Serial.print("Humidity = ");  
        Serial.println(DHT.humidity); //Επιστρεφόμενη τιμή υγρασίας αέρα  
        Serial.print("Soil Moisture = ");  
        Serial.println(readSoil()); ////Επιστρεφόμενη τιμή υγρασίας εδάφους
```

```
}  
}
```

Ο υπόλοιπος κώδικας μένει ως έχει. Για να ρυθμίσετε το χρονικό διάστημα μεταξύ των τιμών του αρκεί να τροποποιήσετε την παρακάτω εντολή:

```
const unsigned long xMinutes = 1* 1000UL;
```

Αν για παράδειγμα θέλετε να παίρνετε τιμές από τους αισθητήρες ανά 5 λεπτά αρκεί να τροποποιήσετε την εντολή σε:

```
const unsigned long xMinutes = 5*60* 1000UL;
```

Αν θέλετε να παίρνετε τιμές από τους αισθητήρες ανά 10 λεπτά αρκεί να τροποποιήσετε την εντολή σε:

```
const unsigned long xMinutes = 10*60* 1000UL;
```

Κατεβάστε τον πλήρη κώδικα της εφαρμογής για το Arduino UNO στο παρακάτω link:

- [Fielduino.ino](#)

## 5. Προβλήματα κατά τη σύνδεση

Πιθανά προβλήματα κατά τη σύνδεση του Arduino UNO με τον υπολογιστή είναι η σωστή επιλογή πλακέτας (Arduino/ Genuino UNO ή όποιο διατίθεται) και θύρας. Η θύρα θα πρέπει να χρησιμοποιηθεί και παρακάτω στη ρύθμιση του Node-Red κόμβου σειριακής εισόδου.

## 6. Node.js (<https://nodejs.org/en/>)

Για την ανάπτυξη της web εφαρμογής χρησιμοποιήσαμε ένα asynchronous event driven JavaScript runtime περιβάλλον, το οποίο ονομάζεται Node.js και έχει σχεδιαστεί για την ανάπτυξη εφαρμογών διαδικτύου. Είναι λογισμικό ανοικτού κώδικα και επιτρέπει τη δημιουργία διαδικτυακών εργαλείων. Ο βασικός του στόχος ήταν να μειώσει την πολυπλοκότητα των κλασικών server εφαρμογών. Ο λόγος που χρησιμοποιήσαμε το Node.js είναι για να μπορέσουμε στη συνέχεια να χρησιμοποιήσουμε το προγραμματιστικό εργαλείο Node-Red.

Στον σύνδεσμο που ακολουθεί υπάρχουν οδηγίες εγκατάστασης του Node.js:

<https://nodejs.org/en/download/package-manager/>

## 7. Node-Red

Το Node-Red είναι ένα προγραμματιστικό εργαλείο το οποίο χρησιμοποιείται για τη διασύνδεση των συσκευών και online υπηρεσιών με εύκολο τρόπο. Παρέχει browser-based συντάκτη που δίνει τη δυνατότητα διασύνδεσης flows μεταξύ τους κάνοντας χρήση κόμβων που ονομάζει nodes. Αυτοί οι κόμβοι που στην περίπτωσή μας θα είναι οι τιμές που περνούν από την τοπική σειριακή θύρα.

Στον σύνδεσμο που ακολουθεί υπάρχουν οδηγίες εγκατάστασης του Node-Red:

<https://nodered.org/docs/getting-started/installation>

Στον σύνδεσμο που ακολουθεί υπάρχουν οδηγίες για το πώς να εκκινήσετε το Node-Red:

<https://nodered.org/docs/getting-started/running>

Προτού χρησιμοποιήσετε το Node-Red θα πρέπει να εγκαταστήσετε κάποια επιπλέον nodes απαραίτητα για το project. Παρακάτω παρουσιάζεται μια λίστα από αυτά:

- **node-red-node-serialport**: Διαβάζει δεδομένα από μια τοπική σειριακή πόρτα
- **node-red-dashboard**: Παρέχει ένα σύνολο nodes για τη γρήγορη και εύκολη δημιουργία πίνακα ελέγχου live δεδομένων

Η εγκατάσταση όλων των απαραίτητων nodes γίνεται με τον ίδιο τρόπο μέσω του συντάκτη. Για να γίνει αυτό επιλέξτε Manage Palette από το menu (πάνω δεξί μέρος της οθόνης) και στη συνέχεια επιλέξτε την install καρτέλα στην palette. Μπορείτε τώρα να ψάξετε για τα nodes που είναι απαραίτητα για το project μας.

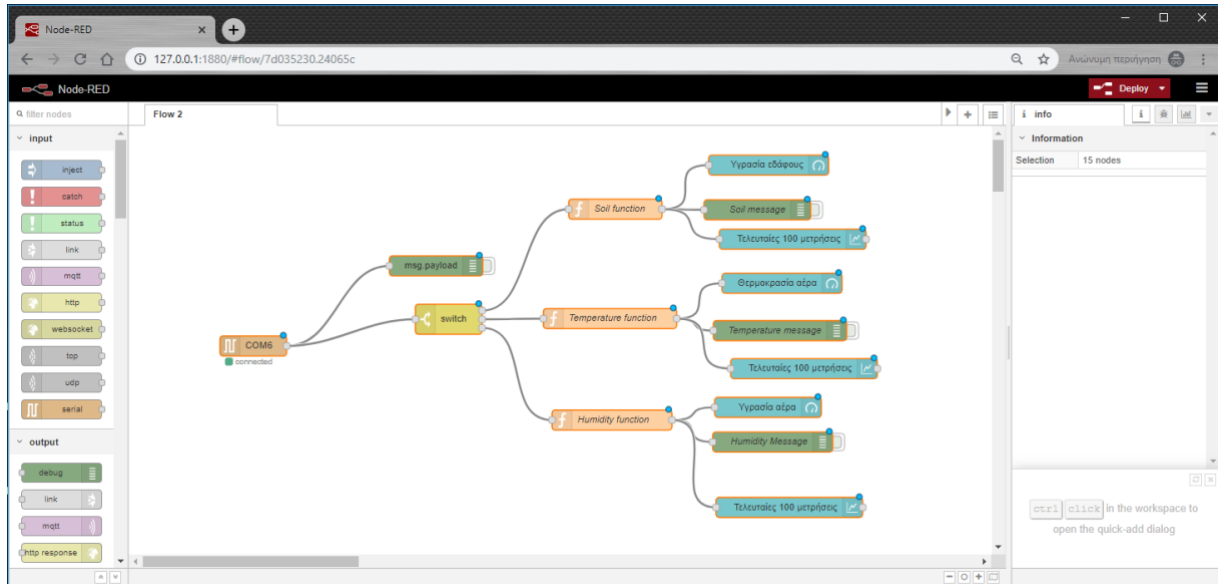
- **Node-Red: Flow**

Παρακάτω παρουσιάζεται ο κώδικας που υλοποιεί το project μας στο Node-Red

- **Fielduino.json**

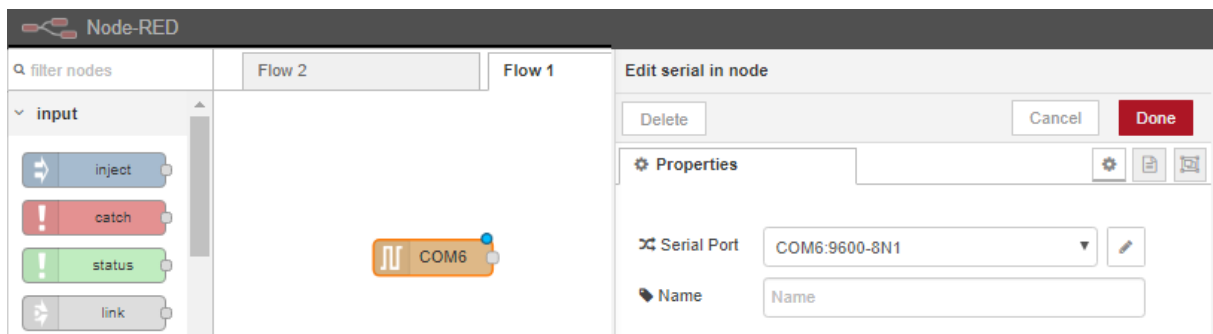
Για να τον προσθέσετε απλά σύρτε το αρχείο στο Node-Red συντάκτη ή προσθέστε το από το menu στο δεξί πάνω μέρος της οθόνης → import → clipboard.

Αν όλα έχουν πάει καλά θα πρέπει να δείτε την παρακάτω οθόνη:



Στη συνέχεια θα εξηγήσουμε τη λογική του παραπάνω flow έτσι ώστε να αντιληφθεί ο χρήστης τη λογική πίσω από αυτό και να είναι σε θέση να το τροποποιήσει – προσαρμόσει στις δικές του ανάγκες.

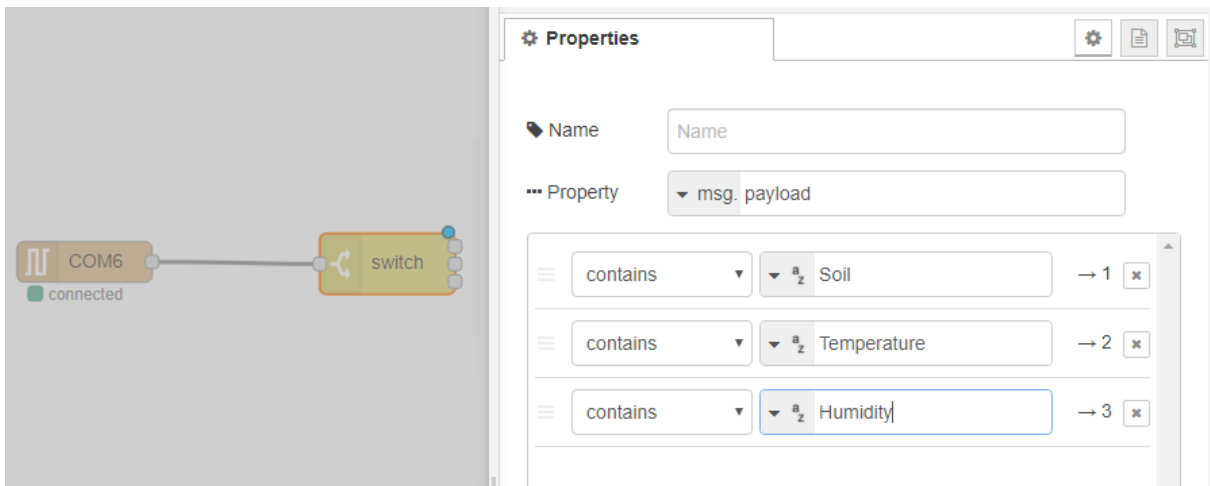
Το πρώτο σημείο της υλοποίησης είναι να μπορέσετε να πάρετε τις τιμές που προωθούνται από το Arduino στη σειριακή πόρτα. Για το σκοπό αυτό προσθέτουμε ένα serial node το οποίο έχει η δυνατότητα να διαβάζει τιμές από τη σειριακή πόρτα.



Στις επιλογές του node επιλέξτε την ίδια σειριακή πόρτα που έχετε χρησιμοποιήσει και στο Arduino UNO. Για να λειτουργήσει σωστά (πράσινη ένδειξη) θα πρέπει η σειριακή έξοδος του Arduino UNO να είναι κλειστή. Αν όλα έχουν πάει καλά στον debugger του Node-Red μπορείτε να δείτε τις τιμές που έρχονται από το Arduino UNO στην παρακάτω μορφή:

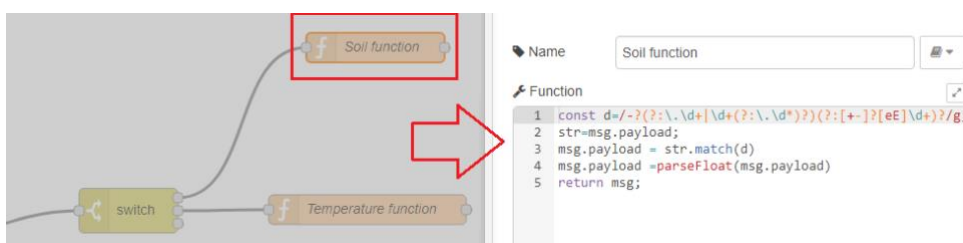
```
3/5/2019, 10:10:02 π.μ.node: b1b04261.9fa1cmsg.payload : string[21]
"Temperature = 25.00€"
3/5/2019, 10:10:02 π.μ.node: b1b04261.9fa1cmsg.payload : string[18]
"Humidity = 37.00€"
3/5/2019, 10:10:02 π.μ.node: b1b04261.9fa1cmsg.payload : string[19]
"Soil Moisture = 0€"
```

Παρατηρούμε πως οι τιμές έρχονται με τη μορφή συμβολοσειράς και με παραπανήσια πληροφορία. Ιδανικά από κάθε γραμμή μας χρειάζεται μόνο η τιμή. Για παράδειγμα από το "Temperature = 25.00€" αρκεί μόνο να απομονώσουμε την τιμή 25.00. Για το σκοπό αυτό χρησιμοποιούμε το node switch.



Στις επιλογές του node switch δημιουργούμε τρεις εκφράσεις έτσι ώστε να μας δώσει τρεις διαφορετικές εξόδους. Η λογική πίσω από αυτό είναι να μπορεί το Node-Red τις τιμές που παίρνει από τη σειριακή είσοδο να τις ανακατευθύνει σε διαφορετικές εξόδους ανάλογα με το αν στο σώμα τους περιέχουν τις τιμές Soil, Temperature και Humidity αντίστοιχα. Έτσι για παράδειγμα την τιμή "Temperature = 25.00€" την ανακατευθύνει στη δεύτερη έξοδο.

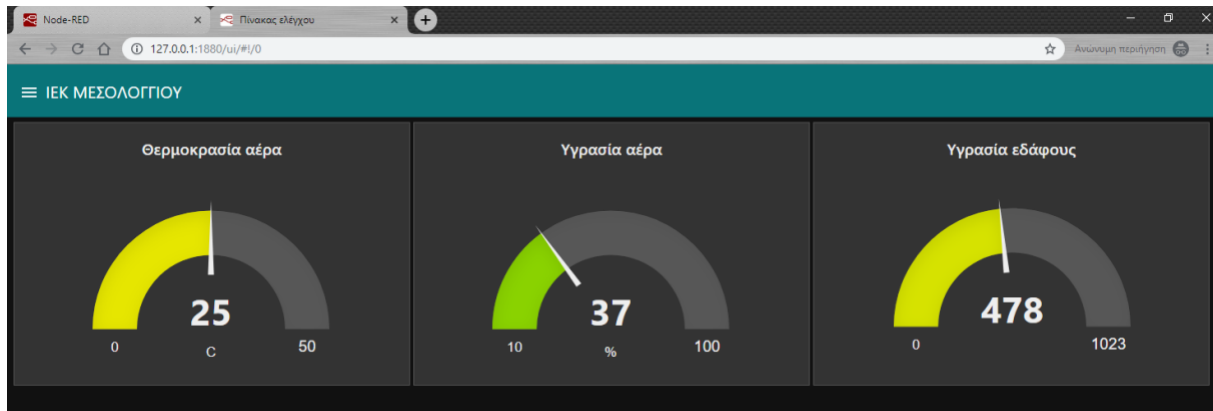
Στη συνέχεια σε κάθε έξοδο εφαρμόζουμε μια regular expression για να μπορέσουμε να απομονώσουμε μόνο την αριθμητική τιμή που μας ενδιαφέρει.



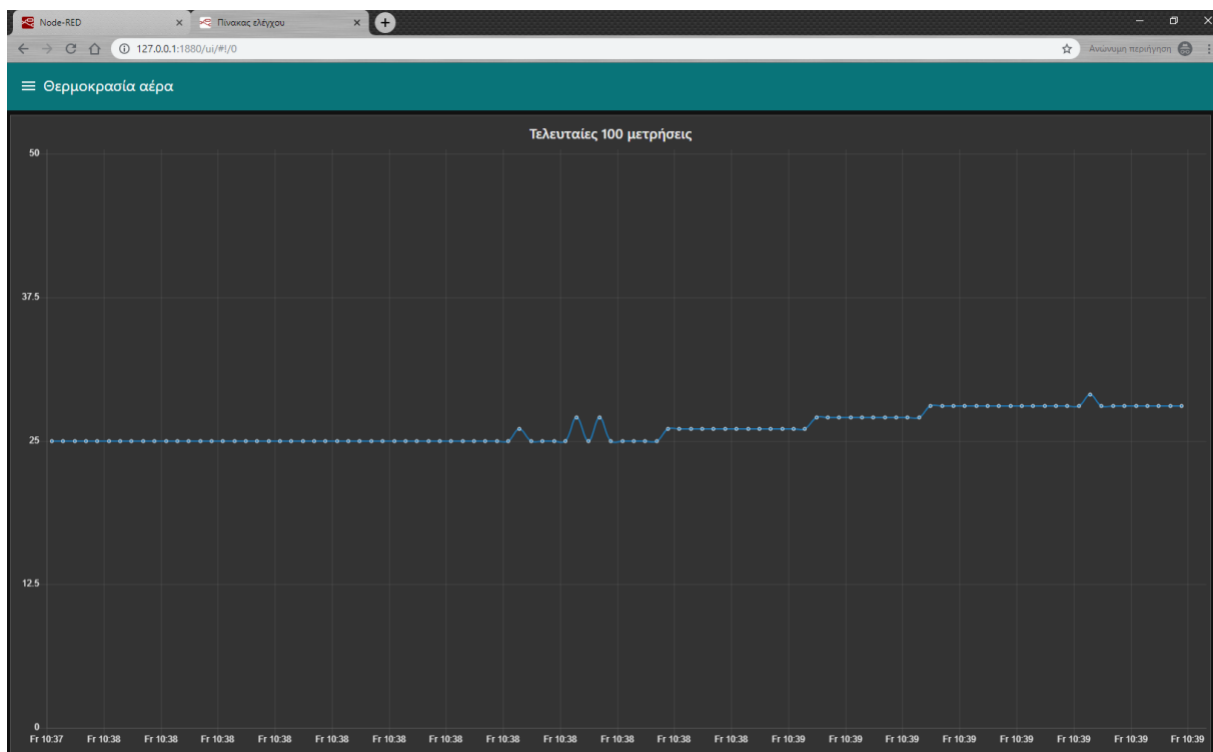
- **Node-Red: Dashboard**

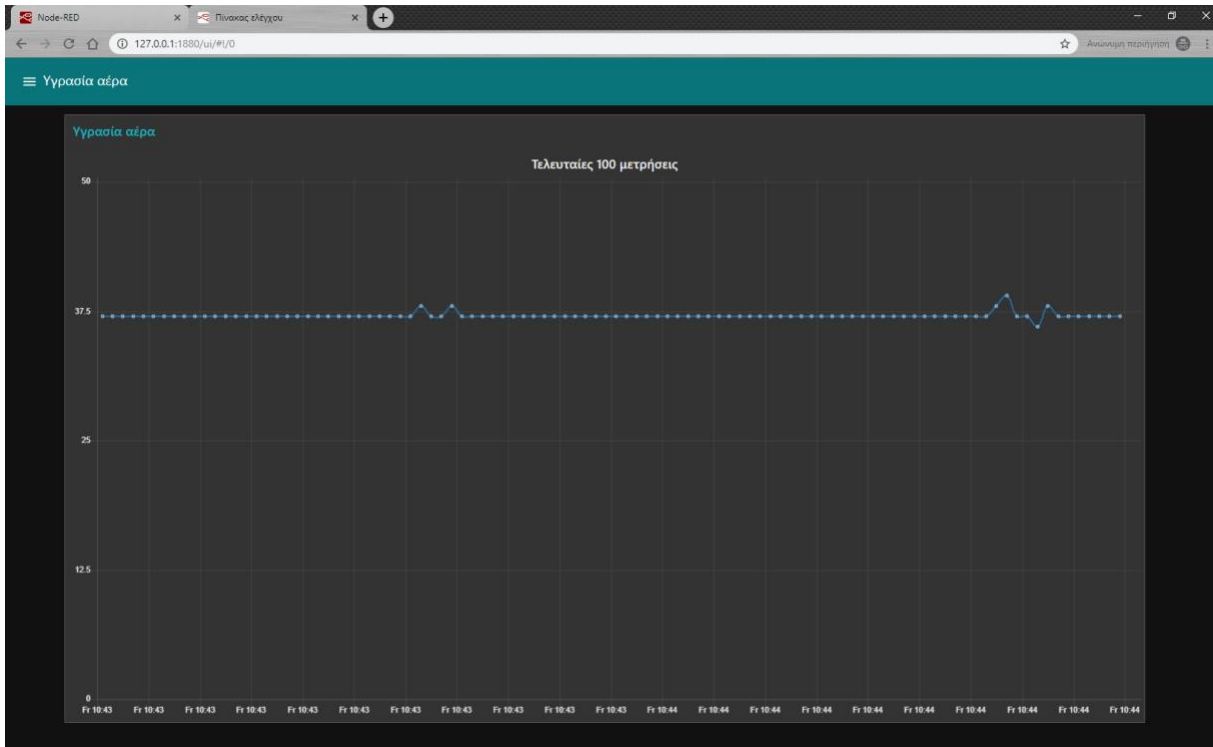


Τέλος δημιουργούμε τους πίνακες ελέγχου για τις τιμές που διαβάζουμε από τους αισθητήρες. Για πρόσβαση στον πίνακα ελέγχου σε μια νέα καρτέλα πληκτρολογούμε τη διεύθυνση που τρέχει ο server μας προσθέτοντας στο τέλος /ui (στο συγκεκριμένο παράδειγμα <http://127.0.0.1:1880/ui>). Για τις τρεις τιμές που παίρνουμε από τους αισθητήρες (Θερμοκρασίας αέρα, Υγρασίας αέρα, Υγρασίας εδάφους) δημιουργούμε μια συγκεντρωτική οθόνη που μας δείχνει την τρέχουσα κατάσταση:

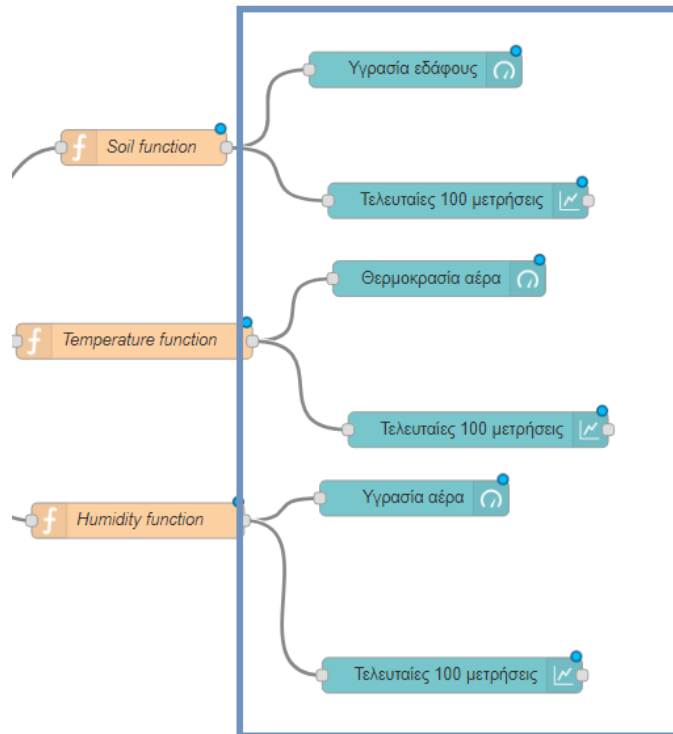


Για κάθε μια από τις τρεις τιμές που παίρνουμε από τους αισθητήρες δημιουργούμε και μια επιπλέον οθόνη στον πίνακα ελέγχου που μας δείχνει τις τελευταίες 100 τιμές σε γράφημα έτσι ώστε να μπορούμε να αντιληφθούμε τις μεταβολές στο πέρασμα της χρονικής διάρκειας (ώρα, ημέρα κτλ.) που έχουμε ορίσει. Παρακάτω φαίνονται οι τρεις οθόνες:





Ο τρόπος δημιουργίας των παραπάνω οθονών υλοποιείται με την παραμετροποίηση των παρακάτω nodes:



Για τις ρυθμίσεις του κάθε node μπορείτε να ανατρέξετε στις ρυθμίσεις του πατώντας διπλό κλικ πάνω του.

## Μέρος 3

### Πιθανές επεκτάσεις του project

Στην ενότητα αυτή αναφέρονται μερικές πιθανές επεκτάσεις που μπορούν να προστεθούν στο εν λόγω project χωρίς αυτό να σημαίνει πως είναι και μοναδικές. Δεσμευόμαστε σε επόμενο διαγωνισμό να επεκτείνουμε τις λειτουργίες του συγκεκριμένου έργου και να προσθέσουμε νέες δυνατότητες ώστε να το κάνουμε περισσότερο αλληλεπιδραστικό και λειτουργικό για τον τελικό χρήστη. Στόχος μας είναι να μπορεί ο οποιοσδήποτε με χαμηλό κόστος και με ελάχιστες γνώσεις πληροφορικής να μπορέσει να αναπτύξει και να προσαρμόσει ένα σύστημα έξυπνης γεωργίας στα μέτρα του.

- **Προσθήκη νέων αισθητήρων**

Είναι σημαντικό να μπορεί κάποιος εύκολα να τροποποιήσει την υπάρχουσα εφαρμογή και να προσθέσει τους δικούς του αισθητήρες ώστε να καλύψει τις δικές του ανάγκες. Παραδείγματα τέτοιων αισθητήρων είναι αισθητήρας μέτρησης pH κτλ.

- **Προσθήκη βάσης δεδομένων**

Μια ακόμη σημαντική προσθήκη στο project είναι και η δυνατότητα αποθήκευσης των καταγεγραμμένων τιμών σε μια βάση δεδομένων για περαιτέρω επεξεργασία. Στο συγκεκριμένο έργο η δυνατότητα αυτή είναι εφικτή καθώς ο χρήστης μπορεί να προσθέσει στο Node-Red διάφορες επιλογές ΒΔ: node-red-node-sqlite, node-red-node-mysql, node-red-contrib-mssql, node-red-mongodb. Προσθέτωντας ΒΔ ο χρήστης μπορεί να ανατρέχει σε προηγούμενες τιμές ώστε να κάνει συγκρίσεις (π.χ. θερμοκρασίες την ίδια ημέρα τα τελευταία 10 χρόνια) κτλ.

- **Προσθήκη οθονών αλληλεπίδρασης**

Μέσω του dashboard και των δυνατοτήτων που παρέχει μπορεί ο χρήστης πολύ εύκολα να προσθέσει νέες οθόνες παρουσίασης της πληροφορίας (Pie chart, Polar area chart, Bar chart κτλ)

- **Αυτόματη ενημέρωση του χρήστη (triggered events)**

Δίνεται η δυνατότητα δημιουργίας triggered events για τους χρήστες. Για παράδειγμα αν η θερμοκρασία περάσει πάνω από κάποια τιμή μπορεί να στέλνεται email στο χρήστη ή κάποια άλλη ειδοποίηση μέσω social media. Τα triggered events μπορούν να προστεθούν και στις οθόνες του dashboard (π.χ. διαφορετικός χρωματισμός για διαφορετικές τιμές).

- **Προσθήκη μοντέλου επεξεργασίας δεδομένων**

Ο χρήστης όπως αναφέρθηκε και νωρίτερα μπορεί να τροποποιήσει το μοντέλο δεδομένων και να προσθέσει παραπάνω από έναν αισθητήρες σε διάφορα σημεία του χωραφιού έτσι ώστε να παίρνει πολλαπλές τιμές θερμοκρασίας, υγρασίας κτλ και να μπορεί έτσι να έχει καλύτερη επίβλεψη. Μέσω της προτεινόμενης υποδομής δίνεται η δυνατότητα με πολύ εύκολο τρόπο να δημιουργηθεί μοντέλο επεξεργασίας δεδομένων. Για παράδειγμα μπορούν οι εισερχόμενες τιμές από 3 αισθητήρες υγρασίας εδάφους να επεξεργαστούν μαζί ώστε να υπολογιστεί μέση τιμή υγρασίας εδάφους ή να τροποποιηθεί στη συνέχεια η ποσότητα νερού ανά σημείο.

- **Προσθήκη ενεργοποιητών και αυτοματισμών**

Πολύ σημαντικό επίσης σημείο το οποίο μπορεί να καλυφθεί σε επόμενες εκδόσεις του project είναι η προσθήκη αυτοματισμών στο σύστημα. Όπως το σύστημα μπορεί να διαβάζει τις μετρούμενες τιμές από τους αισθητήρες, με τον ίδιο τρόπο μπορεί να χειρίζεται ενεργοποιητές. Για παράδειγμα μπορεί ο χρήστης να ενεργοποιεί μέσω της web διεπαφής εξ αποστάσεως το αυτόματο πότισμα, η μπορεί να ενεργοποιεί το κλείσιμο των σκιάστρων σε περίπτωση υψηλής θερμοκρασίας κτλ. Παράλληλα οι ενέργειες αυτές μπορούν να πραγματοποιούνται και με αυτόματο τρόπο. Για παράδειγμα να ενεργοποιείται το αυτόματο πότισμα καθημερινά συγκεκριμένες ώρες ή σε περίπτωση που η θερμοκρασία είναι υψηλή και η υγρασία εδάφους χαμηλή.

## Σημεία που δεν καλύφθηκαν στο συγκεκριμένο project

Στο έργο αυτό περιγράφηκε ένα σύστημα παρακολούθησης γεωργικών καλλιεργειών. Περιγράφηκε ο τρόπος με τον οποίο συνδέονται οι διάφοροι αισθητήρες και προγραμματίζονται μέσω του μικροελεγκτή Arduino UNO. Περιγράφηκε επίσης ο τρόπος με τον οποίο αναπτύχθηκε η web διεπαφή καθώς και πως τα δύο αυτά μέρη επικοινωνούν μεταξύ τους.

Το σημείο που δεν καλύφθηκε καθώς φεύγει πέρα από τα όρια της περιγραφής του έργου, όμως αποτελεί πολύ σημαντικό κομμάτι του είναι πως ο μικροελεγκτής Arduino UNO συνδέεται με το Server

στον οποίο τρέχει η εφαρμογή μας (Node.js, Node-Red). Στην περιγραφή μας εμείς συνδέουμε απευθείας το Arduino UNO με το laptop στο οποίο τρέχει το Node.js.



Σε πραγματικές συνθήκες κάτι τέτοιο δεν είναι ιδιαίτερα βολικό καθώς δεν είναι ιδιαίτερα βολικό να έχει κάποιος στο χωράφι συνδεδεμένο το Arduino UNO με το laptop. Κάτι που δυσκολεύει το προηγούμενο σενάριο είναι και η απουσία συνήθως internet σε γεωργικές εκτάσεις.

Πιθανές εναλλακτικές λύσεις σε αυτό θα μπορούσαν να είναι οι εξής:

- Χρήση σαν Server ενός Raspberry pi πάνω στο οποίο θα στηθούν οι υπηρεσίες και θα συνδεθεί μέσω usb το Arduino UNO. Αυτή η λύση προϋποθέτει την πρόσβαση στο internet με κάποιο τρόπο του Raspberry pi
- Σύνδεση Bluetooth dongle ή wifi module στο Arduino UNO για αποστολή των τιμών των αισθητήρων σε κάποιο Server (π.χ. Raspberry pi ή άλλο) σε μικρή σχετικά απόσταση
- Σύνδεση του Arduino UNO στο internet μέσω Arduino GSM Shield. Αυτή η λύση είναι ιδανική για απομονωμένες γεωργικές εκτάσεις.