



Διεθνές Πανεπιστήμιο της Ελλάδος

Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων

Πτυχιακή Εργασία

“Μετεωρολογικός σταθμός για αυτοματοποιημένη άρδευση με χρήση
GPRS”

Γουλάρας Απόστολος

Επιβλέπων καθηγητής: Γιακουμής Άγγελος

Ακαδημαϊκό έτος 2023-2024

Περίληψη

Αντικείμενο της παρούσας πτυχιακής εργασίας είναι η δημιουργία ενός μετεωρολογικού σταθμού με τη χρήση του Arduino, όπου τα δεδομένα μεταφέρονται στην πλατφόρμα ThingSpeak, ενώ παράλληλα επιτυγχάνεται αυτόματη άρδευση ενός αγροτεμαχίου μέσω του GPRSShield.

Αρχικά, παρουσιάζονται οι βασικές λειτουργίες της κατασκευής, ενώ παρέχονται πληροφορίες σχετικά με τον μετεωρολογικό σταθμό, το Arduino και το ThingSpeak.

Στη συνέχεια, εξετάζονται τα υλικά που επιλέχθηκαν για την κατασκευή, οι ιδιότητές τους και οι λόγοι επιλογής τους.

Στο κατασκευαστικό μέρος, αναλύεται ο τρόπος συνδεσμολογίας των αισθητήρων και των εξαρτημάτων στην πλακέτα.

Στο κομμάτι του κώδικα, παρέχονται πληροφορίες για το ArduinoIDE, και αναλύονται τμηματικά κομμάτια του κώδικα.

Η εργασία ολοκληρώνεται με την παρουσίαση των λειτουργιών του συστήματος και της κατασκευής, συμπεριλαμβανομένου του τελικού κόστους.

Στο τελευταίο κεφάλαιο, προτείνονται τρόποι βελτίωσης της κατασκευής και παρουσιάζονται τα τελικά συμπεράσματα.

Λέξεις Κλειδιά: Μετεωρολογικός σταθμός, Arduino, ThingSpeak, ArduinoUnoR3, PrototypeShieldv.5, GPRSShieldv1.0, DHT11, Soil Hygrometer / Moisture Detection Module, MH-RD Raindrops module, ZTS-3000-FSJT-V05 wind speed sensor, Tongling single channel 5V relay module, ArduinoIDE

Abstract

The focus of this thesis is the creation of a meteorological station using Arduino, where data is transmitted to the ThingSpeak platform, and automatic irrigation of a field is achieved through the GPRS Shield.

Initially, the fundamental functions of the system are presented, along with information about the meteorological station, Arduino, and ThingSpeak.

Next, the materials chosen for the construction are examined, including their properties and the reasons for their selection.

In the construction section, the method of connecting sensors and components to the board is analyzed.

In the coding section, information about the Arduino IDE is provided, and the code is analyzed in detail.

The thesis concludes with a presentation of the system and construction functionalities, including the final cost.

The concluding chapter proposes improvements to the construction and presents the concluding remarks.

Key Words: Weather Station, Arduino, ThingSpeak, ArduinoUnoR3, PrototypeShieldv.5, GPRSShieldv1.0, DHT11, Soil Hygrometer / Moisture Detection Module, MH-RD Raindrops module, ZTS-3000-FSJT-V05 wind speed sensor, Tongling single channel 5V relay module, ArduinoIDE

Περιεχόμενα

| | |
|---|----|
| 1. Εισαγωγή..... | 8 |
| 1.1 Πως μπορείς να δημιουργήσεις έναν μετεωρολογικό σταθμό με τη χρήση Arduino...10 | |
| 1.2 Τι είναι ο μετεωρολογικός σταθμός | 11 |
| 1.3 Τι είναι το Arduino..... | 12 |
| 1.4 Τι είναι το ThingSpeak..... | 13 |
| 2. Επιλογή υλικών κατασκευής..... | 14 |
| 2.1 ArduinoUnoR3..... | 14 |
| 2.2 PrototypeShield v.5..... | 16 |
| 2.3 GPRSShield v1.0..... | 17 |
| 2.4 DHT11 temperature and humidity module..... | 18 |
| 2.5 Soil Hygrometer / Moisture Detection Module..... | 19 |
| 2.6 MH-RD Raindrops module..... | 21 |
| 2.7 ZTS-3000-FSJT-V05 wind speed sensor..... | 22 |
| 2.8 Tongling single channel 5V relay module..... | 23 |
| 3. Συνδεσμολογία..... | 25 |
| 3.1 Συνδεσμολογία Prototype Shield v.5..... | 25 |
| 3.2 Συνδεσμολογία GPRS Shield v1.0..... | 25 |
| 3.3 Συνδεσμολογία DHT11 temperature and humidity module..... | 27 |
| 3.4 Συνδεσμολογία Soil Hygrometer / Moisture Detection Module..... | 28 |
| 3.5 Συνδεσμολογία MH-RD Raindrops module..... | 29 |
| 3.6 Συνδεσμολογία ZTS-3000-FSJT-V05 wind speed sensor..... | 31 |
| 3.7 Συνδεσμολογία Tongling single channel 5V relay module..... | 32 |

| | |
|---|----|
| 3.8 Τελική κατασκευή..... | 33 |
| 3.9 Block Διάγραμμα..... | 34 |
| 4. Λογισμικό..... | 36 |
| 4.1 Το περιβάλλον προγραμματισμού ArduinoIDE..... | 36 |
| 4.2 Σχολιασμός του κώδικα..... | 37 |
| 4.2.1 Βιβλιοθήκες και μεταβλητές..... | 38 |
| 4.2.2 void setup()..... | 39 |
| 4.2.3 void loop()..... | 41 |
| 4.2.4 void configureGPRS()..... | 42 |
| 4.2.5 voidsendData() : συλλογή δεδομένων..... | 45 |
| 4.2.6 voidsendData() : αποστολή δεδομένων..... | 47 |
| 4.2.7 voidctrlRelay() : έλεγχος ρελέ..... | 49 |
| 4.2.8 voidctrlRelay() : αυτόματη απενεργοποίηση ρελέ..... | 52 |
| 4.2.9 voidShowSerialData()..... | 53 |
| 4.2.10 voidSendSMS()..... | 54 |
| 5. Παρουσίαση λειτουργίας..... | 56 |
| 5.1 Ενεργοποίηση του μετεωρολογικού σταθμού..... | 56 |
| 5.2 Μεταφόρτωση του κώδικα..... | 58 |
| 5.3 Ασύρματος έλεγχος ρελέ..... | 60 |
| 6. Κόστος υλικών κατασκευής μετεωρολογικού σταθμού Arduino..... | 62 |
| 7. Βελτιώσεις και αναβαθμίσεις..... | 63 |
| 8. Συμπεράσματα..... | 64 |
| Βιβλιογραφικές αναφορές..... | 67 |

| | |
|-------------------------|----|
| Διαδικτυακές πηγές..... | 69 |
| Παράρτημα..... | 70 |
| Κώδικας Arduino..... | 70 |

Κατάλογος εικόνων

| | |
|---|----|
| Εικόνα 1: ArduinoUnoR3 inout..... | 15 |
| Εικόνα 2: Το Arduino Uno R3..... | 16 |
| Εικόνα 3: Το PrototypeShield v.5..... | 17 |
| Εικόνα 4: Το GPRS Shield v1.0..... | 18 |
| Εικόνα 5: Το DHT11 temperature and humidity module..... | 19 |
| Εικόνα 6: Το Soil Hygrometer / Moisture Detection Module..... | 20 |
| Εικόνα 7: Το MH-RD Raindrops module..... | 21 |
| Εικόνα 8: Ο ZTS-3000-FSJT-V05 wind speed sensor..... | 23 |
| Εικόνα 9: Το Tongling single channel 5V relay module..... | 24 |
| Εικόνα 10: Συνδεσμολογία PrototypeShield v.5 με Arduino..... | 25 |
| Εικόνα 11: Τοποθέτηση Sim στο GPRSShield v1.0..... | 26 |
| Εικόνα 12: Κλείδωμα Sim στο GPRSShield v1.0..... | 26 |
| Εικόνα 13: Συνδεσμολογία GPRSShield v1.0 με PrototypeShield v.5..... | 27 |
| Εικόνα 14: Συνδεσμολογία DHT11 με PrototypeShield v.5..... | 28 |
| Εικόνα 15: Συνδεσμολογία αισθητήρα εδάφους με MH-Sensor-Series..... | 28 |
| Εικόνα 16: Συνδεσμολογία MH-Sensor-Series εδάφους με PrototypeShield v.5..... | 29 |
| Εικόνα 17: Συνδεσμολογία MH-RD Raindrops module με MH-Sensor-Series..... | 30 |
| Εικόνα 18: Συνδεσμολογία MH-Sensor-Series βροχόπτωσης με PrototypeShield v.5..... | 34 |

| | |
|---|----|
| Εικόνα 19: Συνδεσμολογία wind speed sensor με PrototypeShield v.5..... | 32 |
| Εικόνα 20: Συνδεσμολογία Tongling single channel 5V relay module με PrototypeShield v.5.. | 32 |
| Εικόνα 21: Συνολικές συνδέσεις 1..... | 33 |
| Εικόνα 22: Συνολικές συνδέσεις 2..... | 33 |
| Εικόνα 23: Συνολικές συνδέσεις 3..... | 34 |
| Εικόνα 24: Block Διάγραμμα..... | 35 |
| Εικόνα 25: Βιβλιοθήκες και μεταβλητές κώδικα (1-16)..... | 38 |
| Εικόνα 26: void setup() (17-27)..... | 40 |
| Εικόνα 27: void loop() (28-35)..... | 41 |
| Εικόνα 28: void configureGPRS() (36-61)..... | 42 |
| Εικόνα 29: void configureGPRS() (62-92)..... | 44 |
| Εικόνα 30: void sendData() (93-126)..... | 46 |
| Εικόνα 31: void sendData() (127-151)..... | 48 |
| Εικόνα 32: void ctrlRelay() (152-196)..... | 50 |
| Εικόνα 33: void ctrlRelay() (197-216)..... | 52 |
| Εικόνα 34: void ShowSerialData() (217-224)..... | 53 |
| Εικόνα 35: void sendSMS() (225-240)..... | 54 |
| Εικόνα 36: Ο μετεωρολογικός σταθμός..... | 57 |
| Εικόνα 37: Ο αισθητήρας ανέμου του μετεωρολογικού σταθμού..... | 57 |
| Εικόνα 38: Μεταφόρτωση κώδικα..... | 58 |
| Εικόνα 39: Πρώτο σύνολο μετρήσεων του κώδικα στο Serial Monitor..... | 58 |
| Εικόνα 40: 10 ^ο σύνολο μετρήσεων του κώδικα στο Serial Monitor..... | 59 |
| Εικόνα 41: Διαγράμματα των 10 συνόλων μετρήσεων στο ThingSpeak..... | 59 |

| | |
|---|----|
| Εικόνα 42: Δείγμα απάντησης στον έλεγχο του ρελέ μέσω Serial Monitor..... | 60 |
| Εικόνα 43: Δείγμα συνομιλίας SMS με το Arduino..... | 61 |

Κατάλογος πινάκων

| | |
|----------------|----|
| Πίνακας 1..... | 26 |
| Πίνακας 2..... | 27 |
| Πίνακας 3..... | 29 |
| Πίνακας 4..... | 30 |
| Πίνακας 5..... | 31 |
| Πίνακας 6..... | 32 |
| Πίνακας 7..... | 62 |

1. Εισαγωγή

Σε έναν διαρκώς εξελισσόμενο κόσμο, η γεωργία, αν και ένας από τους πιο παραδοσιακούς τομείς, οφείλει να προσαρμοστεί στις σύγχρονες ανάγκες. Η χρήση της τεχνολογίας έχει πλέον γίνει απαραίτητη για την αποτελεσματική διαχείριση των καλλιεργειών, και ένας από τους πιο σημαντικούς τρόπους για να επιτευχθεί αυτό είναι μέσω της εγκατάστασης μετεωρολογικών σταθμών στα αγροτεμάχια. Αυτοί οι σταθμοί παρέχουν κρίσιμες πληροφορίες για τις περιβαλλοντικές συνθήκες, επιτρέποντας στους αγρότες να λαμβάνουν ενημερωμένες αποφάσεις που βελτιστοποιούν την παραγωγή.

Οι μετεωρολογικοί σταθμοί προσφέρουν στους αγρότες τη δυνατότητα να παρακολουθούν σε πραγματικό χρόνο δεδομένα όπως η θερμοκρασία, η υγρασία και η ένταση των ανέμων. Με τη χρήση αυτών των πληροφοριών, οι καλλιεργητές μπορούν να προσαρμόσουν τις πρακτικές τους, μειώνοντας τις σπατάλες σε νερό και άλλους πόρους, ενώ παράλληλα αυξάνουν την αποδοτικότητα και τη βιωσιμότητα της παραγωγής τους.

Η χρήση αυτών των σταθμών συμβάλλει επίσης στη μείωση των κινδύνων που προκύπτουν από απρόβλεπτες καιρικές μεταβολές. Με την έγκαιρη ανίχνευση και πρόβλεψη κρίσιμων φαινομένων, οι αγρότες μπορούν να λάβουν άμεσα μέτρα για την προστασία των καλλιεργειών τους, διασφαλίζοντας την ποιότητα και την ποσότητα της παραγωγής τους σε έναν απαιτητικό και μεταβαλλόμενο περιβάλλον.

Μέσα σε αυτό το πλαίσιο, η παρούσα πτυχιακή εργασία στοχεύει στην ανάλυση της κατασκευής και του προγραμματισμού ενός μετεωρολογικού σταθμού με τη χρήση Arduino. Το σύστημα αυτό θα συλλέγει και θα αποστέλλει τα δεδομένα του στην πλατφόρμα ThingSpeak μέσω του GPRSShield για ανάλυση και αποθήκευση. Επιπλέον, μέσω SMS ο χρήστης θα έχει τη

δυνατότητα να ελέγχει το ρελέ το οποίο είναι συνδεδεμένο στο σύστημα, για την άρδευση του αγροτεμαχίου.

1.1 Πως μπορείς να δημιουργήσεις έναν μετεωρολογικό σταθμό με τη χρήση Arduino

Ο σκοπός της πτυχιακής εργασίας είναι η ανάπτυξη ενός ολοκληρωμένου μετεωρολογικού σταθμού για αυτοματοποιημένη άρδευση, αξιοποιώντας τις δυνατότητες του μικροελεγκτή Arduino. Το σύστημα αυτό θα παρακολουθεί και θα αναλύει τα καιρικά φαινόμενα σε έναν αγροτικό χώρο, συλλέγοντας κρίσιμα μετεωρολογικά δεδομένα όπως η θερμοκρασία, η υγρασία του αέρα και του εδάφους, τα επίπεδα βροχόπτωσης και η ταχύτητα του ανέμου. Αυτά τα δεδομένα θα παρέχουν μια πλήρη εικόνα των περιβαλλοντικών συνθηκών που επικρατούν στο χωράφι, επιτρέποντας τη λήψη τεκμηριωμένων αποφάσεων.

Για τη συλλογή των δεδομένων, θα χρησιμοποιηθούν αισθητήρες υψηλής ακρίβειας, οι οποίοι, σε συνδυασμό με ειδικά προγραμματισμένο κώδικα στο Arduino, θα εξασφαλίσουν αξιόπιστες και ακριβείς μετρήσεις. Τα δεδομένα θα μεταφέρονται μέσω GPRS και θα αποθηκεύονται στην πλατφόρμα ThingSpeak, η οποία προσφέρει τη δυνατότητα παρακολούθησης και ανάλυσης δεδομένων ¹IoT σε πραγματικό χρόνο. Ο χρήστης θα μπορεί να παρακολουθεί τις μετρήσεις καθώς καταγράφονται ζωντανά, να αναλύει τις τάσεις και να λαμβάνει αποφάσεις βασισμένες σε πραγματικά δεδομένα.

Επιπλέον, το σύστημα ενσωματώνει έναν ρελέ που ελέγχεται ασύρματα μέσω SMS για τη διαχείριση μιας αντλίας νερού. Αυτό επιτρέπει στο χρήστη να ενεργοποιεί ή να απενεργοποιεί την αντλία από απόσταση, εξασφαλίζοντας αυτόματη ή χειροκίνητη άρδευση του χωραφιού ανάλογα με τις ανάγκες των καλλιεργειών και τις καιρικές συνθήκες. Έτσι, το σύστημα συμβάλλει στη βελτιστοποίηση της χρήσης νερού, αποτρέποντας την υπερβολική άρδευση και εξοικονομώντας πόρους.

¹ Internet of Things

1.2 Τι είναι ο μετεωρολογικός σταθμός

Οι μετεωρολογικοί σταθμοί αποτελούν κρίσιμα εργαλεία για την κατανόηση και παρακολούθηση των κλιματολογικών συνθηκών και φαινομένων. Αυτές οι εγκαταστάσεις είναι σχεδιασμένες για τη συνεχή καταγραφή και ανάλυση των μετεωρολογικών δεδομένων, όπως η θερμοκρασία, η υγρασία, η ταχύτητα και κατεύθυνση του ανέμου, η βροχόπτωση και η ατμοσφαιρική πίεση. Με τη βοήθεια προηγμένων αισθητήρων και τεχνολογικών συστημάτων, οι μετεωρολογικοί σταθμοί παρέχουν ακριβή και αξιόπιστα δεδομένα, τα οποία είναι ζωτικής σημασίας για την πρόβλεψη του καιρού και την ανάλυση των κλιματολογικών τάσεων.

Το βασικό έργο ενός μετεωρολογικού σταθμού είναι η συνεχής παρακολούθηση των καιρικών συνθηκών, παρέχοντας δεδομένα απαραίτητα για την καθημερινή πρόγνωση του καιρού, αλλά και για μακροπρόθεσμες μελέτες της κλιματικής αλλαγής. Οι μετεωρολογικοί σταθμοί, σε συνδυασμό με δίκτυα επικοινωνιών, επιτρέπουν τη συλλογή και μετάδοση δεδομένων σε πραγματικό χρόνο, διευκολύνοντας την άμεση ανάλυση και ανταπόκριση σε καιρικά φαινόμενα.

Με την εξέλιξη της τεχνολογίας, οι μετεωρολογικοί σταθμοί έχουν εξελιχθεί σε ολοκληρωμένα κέντρα έρευνας, συνεισφέροντας στην κατανόηση των φυσικών διεργασιών που διαμορφώνουν το κλίμα της Γης. Αυτοί οι σταθμοί παίζουν σημαντικό ρόλο στην επιστημονική έρευνα, ιδίως όσον αφορά τις επιπτώσεις της κλιματικής αλλαγής, ενώ ταυτόχρονα συμβάλλουν στην ανάπτυξη προηγμένων μοντέλων πρόγνωσης καιρού.

Επιπλέον, οι μετεωρολογικοί σταθμοί έχουν σημαντική εκπαιδευτική αξία. Παρέχουν τη δυνατότητα σε μαθητές και επισκέπτες να αποκτήσουν γνώση γύρω από τη μετεωρολογία και την κλιματολογία, προσφέροντας μια μοναδική ευκαιρία για σύνδεση της θεωρητικής γνώσης με την πρακτική εφαρμογή. Μέσω διαδραστικών παρουσιάσεων και εκπαιδευτικών

δραστηριοτήτων, οι επισκέπτες μπορούν να κατανοήσουν καλύτερα τη λειτουργία των μετεωρολογικών σταθμών και την επίδραση της ατμόσφαιρας στην καθημερινότητά μας.

1.3 Τι είναι το Arduino

Το Arduino είναι μια ανοιχτού κώδικα ηλεκτρονική πλατφόρμα που βασίζεται σε ευέλικτο και εύχρηστο υλικό και λογισμικό. Αρχικά αναπτύχθηκε για να διευκολύνει την δημιουργία διαδραστικών εφαρμογών από ανθρώπους που δεν έχουν εκτενή τεχνική γνώση στον προγραμματισμό. Το Arduino προσφέρει μια απλή και προσιτή μέθοδο για την εκμάθηση της ηλεκτρονικής, καθώς παρέχει μια σειρά από πλακέτες ανάπτυξης και αισθητήρες που μπορούν να συνδυαστούν με άλλες συσκευές, επιτρέποντας στους χρήστες να πειραματιστούν και να δημιουργήσουν χωρίς περιορισμούς.

Το βασικό χαρακτηριστικό του Arduino είναι η δυνατότητά του να συνδέεται με διάφορους αισθητήρες και ενεργοποιητές, επιτρέποντας τη κατασκευή διαδραστικών εφαρμογών που αντιδρούν στις μεταβολές του περιβάλλοντος. Το Arduino μπορεί να διαχειριστεί διάφορες εισόδους, όπως φως, θερμοκρασία και ήχο, και να ελέγξει εξόδους, όπως LED, κινητήρες και οθόνες.

Η ανοιχτή φύση του Arduino έχει συμβάλει στη ραγδαία διάδοσή του σε όλο τον κόσμο, ενθαρρύνοντας τη δημιουργία μιας μεγάλης και ενεργής κοινότητας που περιλαμβάνει εκπαιδευτικά ιδρύματα, ερευνητές, καλλιτέχνες και επαγγελματίες της βιομηχανίας. Αυτή η κοινότητα έχει αναπτύξει και μοιραστεί χιλιάδες διαφορετικά σχέδια και προγράμματα, προωθώντας την ιδέα ότι οποιοσδήποτε, ανεξαρτήτως επιπέδου γνώσεων, μπορεί να δημιουργήσει λειτουργικά και χρήσιμα ηλεκτρονικά έργα.

1.4 Τι είναι το ThingSpeak

Το ThingSpeak είναι μια πλατφόρμα Internet of Things (IoT) που αναπτύχθηκε από την MathWorks και κυκλοφόρησε το 2010. Ο στόχος της είναι να παρέχει ένα ανοιχτό και εύχρηστο περιβάλλον για τη συλλογή, την ανάλυση και την οπτικοποίηση δεδομένων από διάφορες συσκευές και αισθητήρες. Το ThingSpeak έχει σχεδιαστεί για να υποστηρίζει τη σύνδεση συσκευών IoT μέσω του πρωτοκόλλου HTTP, επιτρέποντας την εύκολη αποστολή δεδομένων στο cloud και την ανάλυσή τους σε πραγματικό χρόνο.

Η πλατφόρμα προσφέρει έναν ισχυρό συνδυασμό εργαλείων για την αποθήκευση και την ανάλυση δεδομένων, καθώς και για την οπτικοποίησή τους μέσω γραφημάτων και πινάκων. Οι χρήστες μπορούν να δημιουργήσουν «κανάλια» για τη συλλογή δεδομένων, τα οποία μπορούν να προγραμματιστούν για να λαμβάνουν δεδομένα από αισθητήρες, πλακέτες ανάπτυξης όπως το Arduino, ή άλλες συσκευές IoT. Επιπλέον, το ThingSpeak υποστηρίζει τη χρήση εφαρμογών MATLAB για την ανάλυση των δεδομένων, διευκολύνοντας τη δημιουργία εξελιγμένων αναφορών και την εφαρμογή προηγμένων αλγορίθμων ανάλυσης. Μέσω αυτής της πλατφόρμας, οι χρήστες μπορούν να παρακολουθούν και να ανταγωνίζονται με ζωντανά δεδομένα, κάνοντας την πιο ολοκληρωμένη διαχείριση και ανάλυση δεδομένων IoT.

2. Επιλογή υλικών κατασκευής

Για τη πλήρη ικανοποίηση των ζητούμενων της κατασκευής ενός μετεωρολογικού πάρκου, επιλέχθηκαν τα παρακάτω υλικά.

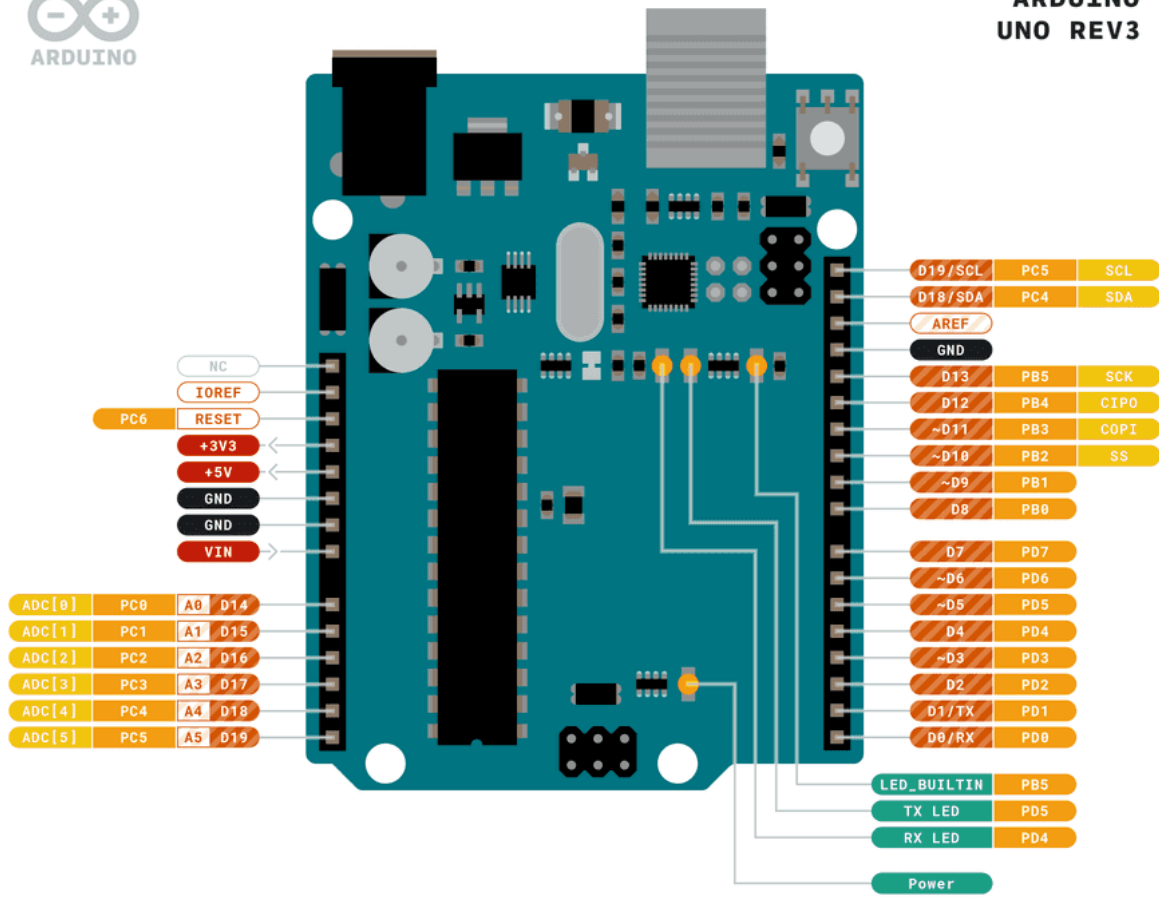
- Arduino Uno R3
- Prototype Shield v.5
- GPRS Shield v1.0
- DHT11 temperature and humidity module
- Soil Hygrometer / Moisture Detection Module
- MH-RD Raindrops module
- ZTS-3000-FSJT-V05 wind speed sensor
- Tongling single channel 5V relay module

2.1 ArduinoUnoR3

Η πλακέτα ArduinoUnoR3 είναι βασισμένη στο μικροελεγκτή ATmega328P και αποτελεί την πιο δημοφιλή πλακέτα Arduino. Διαθέτει 14 ψηφιακά pin εισόδου και εξόδου, από τα οποία τα 6 μπορούν να χρησιμοποιηθούν ως PWM εισόδους, 6 αναλογικές εισόδους και έναν κεραμικό συνηχητή των 16MHz. Επιπλέον, μπορεί να τροφοδοτηθεί από καλώδιο USB ή με μπαταρία μέσω powerjack.



ARDUINO UNO REV3



| | | | |
|--------|--------------|-------------|------------------------|
| Ground | Internal Pin | Digital Pin | Microcontroller's Port |
| Power | SWD Pin | Analog Pin | |
| LED | Other Pin | Default | |

ARDUINO.CC
CC BY SA
This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Εικόνα 1: ArduinoUnoR3 Pinout

Η συγκεκριμένη πλακέτα επιλέχθηκε για την κατασκευή λόγω του πλήθους των διαθέσιμων pin, που παρέχουν ευελιξία και επιτρέπουν την εύκολη επέκταση και αναβάθμιση του συνολικού συστήματος. Επιπλέον, η αξιοπιστία της πλακέτας, η οποία απορρέει από τη δημοτικότητα και την ευρεία χρήση της, αποτέλεσε έναν κρίσιμο παράγοντα στην επιλογή αυτή. Η ευρεία αποδοχή της πλακέτας στον κόσμο της ανάπτυξης ηλεκτρονικών έργων διασφαλίζει

την υποστήριξη και τη σταθερότητα της εφαρμογής, καθιστώντας την ιδανική για το συγκεκριμένο έργο.



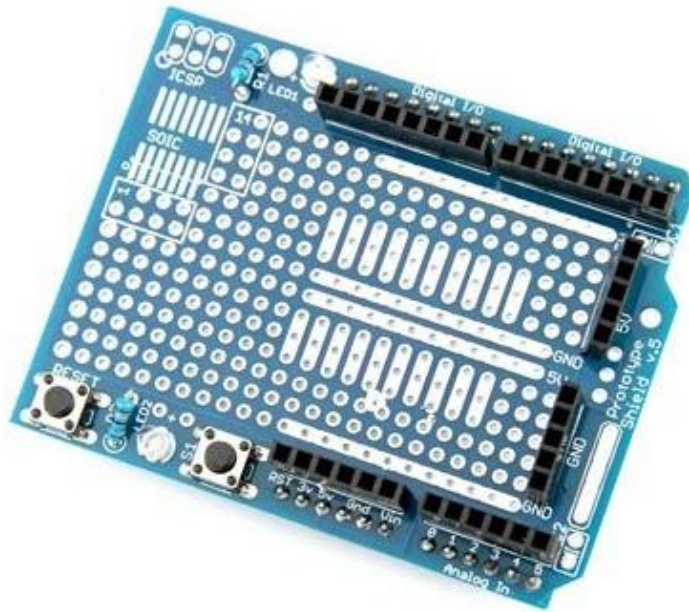
Εικόνα 2: Το ArduinoUnoR3

2.2 Prototype Shield v.5

Το Prototype Shield v.5, σχεδιασμένο από την Adafruit, είναι μια επέκταση που τοποθετείται πάνω από την πλακέτα ArduinoUnoR3, διευκολύνοντας τη δημιουργία πρωτοτύπων κυκλωμάτων. Διαθέτει ενσωματωμένη πειραματική πλακέτα (breadboard), πλήρη πρόσβαση σε όλα τα ψηφιακά και αναλογικά pin του Arduino, και επιπλέον σειρές pin headers για εύκολη σύνδεση εξαρτημάτων. Περιλαμβάνει επίσης θέσεις για διόδους LED, κουμπιά και ένα reset button, καθιστώντας το ιδανικό για δοκιμές και ανάπτυξη εφαρμογών.

Η συγκεκριμένη επέκταση επιλέχθηκε για τη διευκόλυνση της κατασκευής, καθώς απλοποιεί σημαντικά τη συνδεσμολογία και μειώνει την ανάγκη για εξωτερικά breadboards,

εξοικονομώντας έτσι πολύτιμο χώρο. Παράλληλα, επιτρέπει την τακτοποίηση των κυκλωμάτων με τρόπο που βελτιώνει τη συνολική εργονομία του σχεδίου, καθιστώντας την κατασκευή πιο συμπαγή και οργανωμένη.



Εικόνα 3: Το Prototype Shield v.5

2.3 GPRSShieldv1.0

Το GPRSShield v1.0 είναι ένα πρόσθετο για πλακέτες Arduino που επιτρέπει την ασύρματη επικοινωνία μέσω δικτύων κινητής τηλεφωνίας, παρέχοντας πρόσβαση σε υπηρεσίες όπως η αποστολή και λήψη δεδομένων, φωνητικών κλήσεων και μηνυμάτων SMS. Βασισμένο στο SIM900 module, ένα τετραπλό GSM/GPRS module, μπορεί να λειτουργήσει σε συχνότητες 850/900/1800/1900 MHz, καθιστώντας το κατάλληλο για χρήση σε παγκόσμιο επίπεδο. Επιπλέον, έχει 12 GPIO (General Purpose Input/Output, γενικού σκοπού είσοδος/έξοδος), 2 PWM και ελέγχεται από εντολές AT.

Το GPRSShield v1.0 επιλέχθηκε για την εργασία λόγω της προηγμένης δυνατότητάς του να αποστέλλει δεδομένα στη διαδικτυακή πλατφόρμα ThingSpeak, επιτρέποντας τη συνεχή και αξιόπιστη ροή πληροφοριών από το Arduino προς τον διακομιστή. Αυτή η λειτουργία είναι κρίσιμη για την παρακολούθηση και ανάλυση των δεδομένων σε πραγματικό χρόνο, εξασφαλίζοντας ότι οι μετρήσεις που συλλέγονται από τους αισθητήρες καταγράφονται και αξιολογούνται άμεσα και απομακρυσμένα μέσω του διαδικτύου.

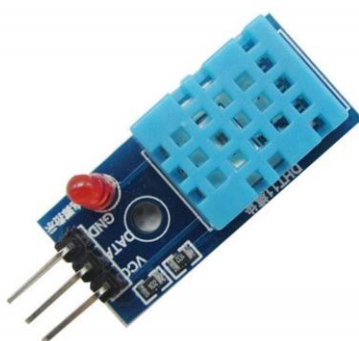


Εικόνα 4: Το GPRSShield v1.0

2.4 DHT11 temperature and humidity module

Ο αισθητήρας θερμοκρασίας και υγρασίας DHT11 διαθέτει ένα σύμπλεγμα αισθητήρων θερμοκρασίας και υγρασίας με βαθμονομημένη ψηφιακή έξοδο. Χρησιμοποιώντας μια αποκλειστική τεχνική ψηφιακής απόκτησης σήματος και τεχνολογία ανίχνευσης θερμοκρασίας και υγρασίας, εξασφαλίζει υψηλή αξιοπιστία και εξαιρετική μακροχρόνια σταθερότητα, καταγράφοντας τη θερμοκρασία από 0°C έως 50°C και την υγρασία της ατμόσφαιρας από 20% έως και 90% με ακρίβεια $\pm 1^\circ\text{C}$ και $\pm 1\%$ αντίστοιχα. Επιπλέον, ο αισθητήρας έχει 3 pin, το VCC για την τάση, το DAT για τη ψηφιακή έξοδο και το GND για την γείωση.

Ο αισθητήρας επιλέχθηκε λόγω της ευχρηστίας του, της ευκολίας στην ανεύρεσή του και της εξαιρετικής μακροχρόνιας αξιοπιστίας του. Η απλότητα στην εγκατάσταση και η δυνατότητα άμεσης ενσωμάτωσης τον καθιστούν ιδανική επιλογή για εφαρμογές που απαιτούν αξιόπιστες μετρήσεις θερμοκρασίας και υγρασίας. Επιπλέον, η ευρεία διαθεσιμότητα του αισθητήρα στην αγορά διασφαλίζει ότι η προμήθεια και η αντικατάστασή του είναι εύκολες και οικονομικές, κάνοντάς τον μία από τις πιο πρακτικές λύσεις για διαρκή χρήση.



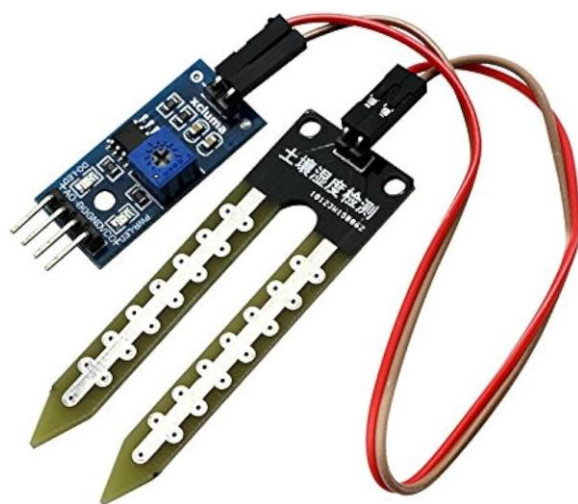
Εικόνα 5: Το DHT11 temperature and humidity module

2.5 Soil Hygrometer / Moisture Detection Module

Ο αισθητήρας Soil Moisture Detection Module μετράει τη ποσότητα νερού που βρίσκεται μέσα στο έδαφος δημιουργώντας ένα ηλεκτρικό πεδίο στο έδαφος. Επειδή τα μόρια του νερού είναι πολικά, τα μη δεσμευμένα μόρια νερού στο έδαφος περιστρέφονται ώστε να ευθυγραμμιστούν με τις γραμμές του ηλεκτρικού πεδίου. Η περιστροφή των μη δεσμευμένων μορίων νερού απαιτεί ενέργεια, η οποία αποθηκεύεται ως δυναμική ενέργεια στα ευθυγραμμισμένα μόρια νερού. Όσο περισσότερο νερό υπάρχει στο έδαφος, τόσο περισσότερη ενέργεια αποθηκεύεται και τόσο υψηλότερη είναι η συνολική διηλεκτρική σταθερά του εδάφους. Με αυτό τον τρόπο ο αισθητήρας μπορεί να καταγράψει με ακρίβεια την υγρασία του εδάφους.

Επιπλέον, για τη χρήση του αισθητήρα εκμεταλλευόμαστε 3 από τα 4 pin, το VCC για τη τάση, το GND για τη γείωση και το A0 για την αναλογική έξοδο, όπου όσο μεγαλύτερη η καταγραφή τόσο χαμηλότερη η υγρασία του εδάφους, ενώ το D0 μένει αχρησιμοποίητο.

Η επιλογή του αισθητήρα Soil Hygrometer / Moisture Detection Module από την OEM έγινε για αρκετούς σημαντικούς λόγους πέρα από την προσιτή του τιμή. Καταρχάς, ο συγκεκριμένος αισθητήρας παρέχει αξιόπιστες και ακριβείς μετρήσεις της υγρασίας του εδάφους, που είναι κρίσιμες για την ακριβή παρακολούθηση και διαχείριση της υγρασίας. Η σχεδίασή του εξασφαλίζει αντοχή και σταθερή απόδοση, ενώ η ευρεία διαθεσιμότητά του και η υποστήριξη από τον κατασκευαστή διευκολύνουν την ενσωμάτωσή του στο σύστημα και την επίλυση τυχόν τεχνικών ζητημάτων. Επίσης, η αναγνωρισμένη ποιότητα και αξιοπιστία του προϊόντος σε διάφορες συνθήκες περιβάλλοντος ενισχύει την εμπιστοσύνη στη λειτουργικότητά του και την αποτελεσματικότητα της εφαρμογής.

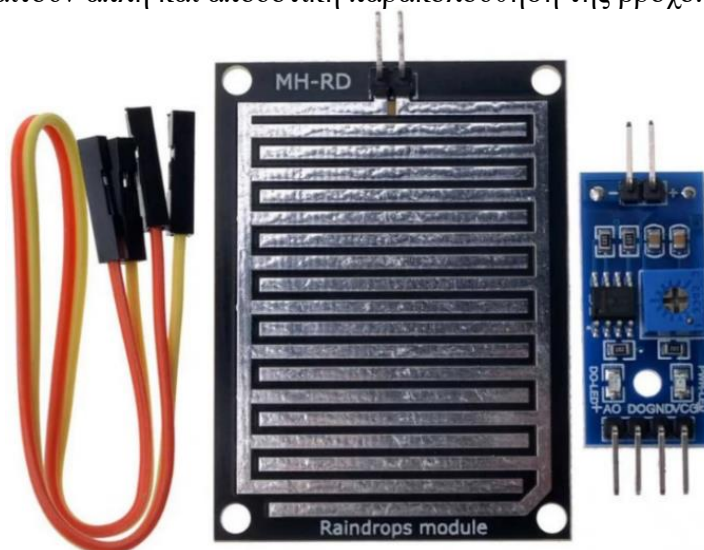


Εικόνα 6: Το Soil Hygrometer / Moisture Detection Module

2.6 MH-RD Raindrops module

Ο αισθητήρας MH-RD Raindrops module είναι ένας αισθητήρας καταγραφής της βροχόπτωσης. Χρησιμοποιώντας ηλεκτρονικές διατάξεις, ο αισθητήρας ανιχνεύει τις σταγόνες της βροχής και μετατρέπει αυτή την πληροφορία σε ηλεκτρικό σήμα. Αυτό το σήμα μπορεί στη συνέχεια να χρησιμοποιηθεί για την παρακολούθηση της έντασης της βροχόπτωσης και την ενεργοποίηση ή απενεργοποίηση σχετικών συστημάτων ή ειδοποιήσεων. Επιπλέον, όπως και για το Soil Sensor, για τη χρήση του αισθητήρα εκμεταλλευόμαστε 3 από τα 4 pin, το VCC για τη τάση, το GND για τη γείωση και το A0 για την αναλογική έξοδο, όπου όσο μεγαλύτερη η καταγραφή τόσο χαμηλότερη η υγρασία του εδάφους, ενώ το D0 μένει αχρησιμοποίητο.

Ο αισθητήρας MH-RD Raindrops Module επιλέχθηκε λόγω της αξιοπιστίας και της σταθερότητάς του στην ανίχνευση της βροχόπτωσης. Προσφέρει ακριβείς μετρήσεις με συνεπή απόδοση, ακόμα και σε ποικιλία καιρικών συνθηκών, εξασφαλίζοντας αξιόπιστα δεδομένα για την ένταση της βροχής. Η εγκατάσταση και η χρήση του είναι ιδιαίτερα εύκολες, κάνοντάς τον ιδανικό για γρήγορη ενσωμάτωση σε διάφορα συστήματα. Αυτό τον καθιστά κατάλληλο για εφαρμογές που απαιτούν απλή και αποδοτική παρακολούθηση της βροχόπτωσης.



Εικόνα 7: Το MH-RD Raindrops module

2.7 ZTS-3000-FSJT-V05 wind speed sensor

Ο αισθητήρας ταχύτητας ανέμου ZTS-3000-FSJT-V05 λειτουργεί με τη βοήθεια περιστρεφόμενων πτερυγίων που είναι συνδεδεμένα σε οριζόντιους βραχίονες, οι οποίοι με τη σειρά τους είναι προσαρτημένοι σε κάθετο άξονα. Όταν ο άνεμος φυσάει, τα πτερύγια περιστρέφονται, προκαλώντας την περιστροφή του άξονα. Η ταχύτητα περιστροφής του άξονα, η οποία εξαρτάται από τη δύναμη του ανέμου, μετατρέπεται σε ηλεκτρικό σήμα. Το σήμα αυτό αντιπροσωπεύει την ταχύτητα του ανέμου με ακρίβεια και μπορεί να μετρήσει ταχύτητες έως και 30 μέτρα ανά δευτερόλεπτο (m/s). Μέσω της μέτρησης των περιστροφών του άξονα, υπολογίζεται η ταχύτητα του ανέμου. Δεδομένου ότι οι ταχύτητες του ανέμου δεν είναι συνεχείς αλλά παρουσιάζουν διακυμάνσεις (ριπές και παύσεις), η ταχύτητα του ανέμου συνήθως μετράται και αναλύεται κατά μέσο όρο σε σύντομες χρονικές περιόδους για να παρέχει μια ακριβή εκτίμηση.

Η επιλογή του αισθητήρα ταχύτητας ανέμου ZTS-3000-FSJT-V05 βασίστηκε σε τρεις βασικούς παράγοντες: την ακρίβεια, την ανθεκτικότητα και την ευκολία συνδεσμολογίας του. Ο συγκεκριμένος αισθητήρας προσφέρει υψηλή ακρίβεια στις μετρήσεις της ταχύτητας του ανέμου, γεγονός που διασφαλίζει αξιόπιστα δεδομένα για την ανάλυση και παρακολούθηση. Επιπλέον, η ανθεκτική κατασκευή του τον καθιστά κατάλληλο για χρήση σε διάφορες καιρικές συνθήκες, εξασφαλίζοντας μακροχρόνια απόδοση και αξιοπιστία. Τέλος, η απλή και άμεση διαδικασία σύνδεσής του με τα ηλεκτρονικά συστήματα ή τους μικροελεγκτές διευκολύνει την ενσωμάτωσή του, καθιστώντας τον ιδανική επιλογή για έργα που απαιτούν γρήγορη και εύκολη εγκατάσταση χωρίς την ανάγκη περίπλοκης διαχείρισης καλωδίων.



Εικόνα 8: Ο ZTS-3000-FSJT-V05 wind speed sensor

2.8 Tongling single channel 5V relay module

Η πλακέτα διαθέτει ένα ρελέ που λειτουργεί με τάση 5V, ενώ το σήμα εισόδου μπορεί να προέρχεται απευθείας από έξοδο μικροελεγκτή που λειτουργεί είτε στα 3V είτε στα 5V για τον έλεγχο του ρελέ. Κάθε ρελέ είναι ικανό να εναλλάσσει ποικιλία φορτίων υψηλής τάσης και υψηλής έντασης ρεύματος, που λειτουργούν σε 110V ή 220VAC, όπως φώτα, ανεμιστήρες και κινητήρες. Η κατάσταση του ρελέ υποδεικνύεται από ένα κόκκινο LED. Στην παρούσα κατασκευή, το ρελέ χρησιμοποιείται για την αυτόματη ενεργοποίηση και απενεργοποίηση μιας αντλίας νερού, η οποία διαχειρίζεται την άρδευση του χωραφιού. Αυτή η εφαρμογή επιτρέπει την απομακρυσμένη και αυτοματοποιημένη ρύθμιση της άρδευσης, προσαρμόζοντας τη λειτουργία της αντλίας ανάλογα με τις ανάγκες του αγρού και τις καιρικές συνθήκες.

Η επιλογή του Tongling single channel 5V relay module έγινε λόγω της απλότητας και της αποτελεσματικότητάς του, καθώς διαθέτει ένα μοναδικό ρελέ, το οποίο επαρκεί για τις ανάγκες της συγκεκριμένης κατασκευής, εξαλείφοντας την ανάγκη για επιπλέον ρελέ. Η σήμανση κατάστασης με κόκκινο LED είναι ιδιαίτερα χρήσιμη, καθώς παρέχει άμεση οπτική

ένδειξη της λειτουργίας του ρελέ, διευκολύνοντας την παρακολούθηση της κατάστασης της αντλίας. Επιπλέον, το module είναι οικονομικό, προσφέροντας μια προσιτή λύση χωρίς να θυσιάζεται η ποιότητα και η αποτελεσματικότητα, καθιστώντας το ιδανική επιλογή για την αυτοματοποιημένη άρδευση του χωραφιού.



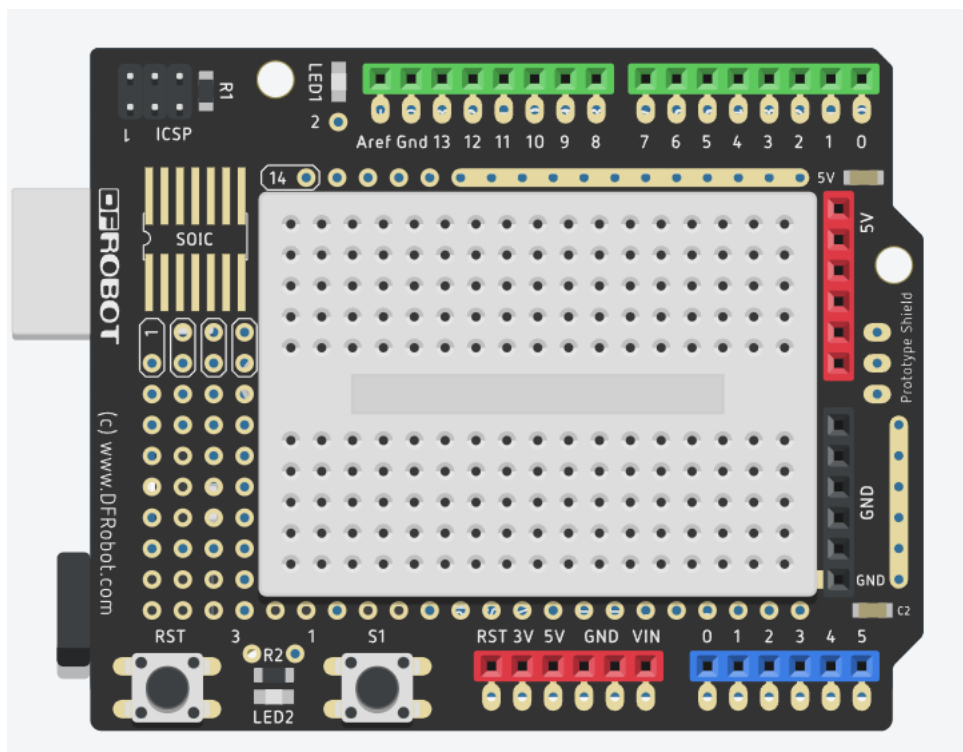
Εικόνα 9: Το Tongling single channel 5V relay module

3. Συνδεσμολογία

Σε αυτό το κεφάλαιο θα παρατεθεί η συνδεσμολογία κάθε αισθητήρα ξεχωριστά με εικόνες από το Tinkercad ή φωτογραφίες επεξεργασμένες στο Photoshop, ενώ στο τέλος θα υπάρχουν φωτογραφίες με το τελικό αποτέλεσμα.

3.1 Συνδεσμολογία PrototypeShield v.5

Το PrototypeShield v.5 συνδέεται ακριβώς πάνω από τη πλακέτα Arduino χωρίς τη χρήση κάποιου καλωδίου.



Εικόνα 10: Συνδεσμολογία PrototypeShield v.5 με Arduino

3.2 Συνδεσμολογία GPRSShield v1.0

Αρχικά για να λειτουργήσει το GPRSShield, πριν συνδέσουμε τα καλώδια, θα χρειαστεί να εισάγουμε τη κάρτα sim στο κάτω μέρος της επέκτασης όπως φαίνεται στις εικόνες 11 και 12.



Εικόνα 11: Τοποθέτηση Sim στο GPRSShieldv1.0

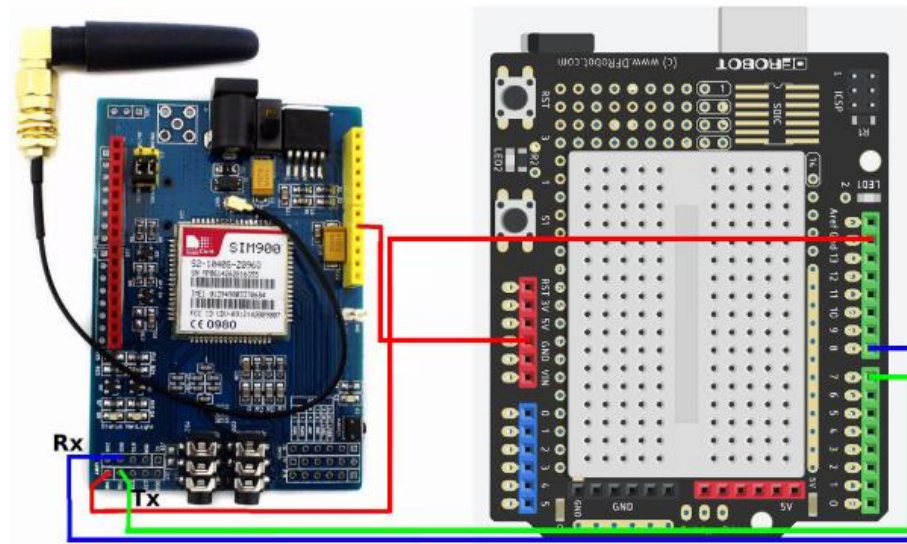


Εικόνα 12: Κλείδωμα Sim στο GPRSShieldv1.0

Έπειτα, συνδέουμε τα καλώδια με βάση τον πίνακα 1.

| GPRS Shield v1.0 | Prototype Shield v.5 |
|------------------|----------------------|
| GND | Analog GND |
| Sim900 GND | Digital GND |
| RXD | D8 |
| TXD | D7 |

Πίνακας 1



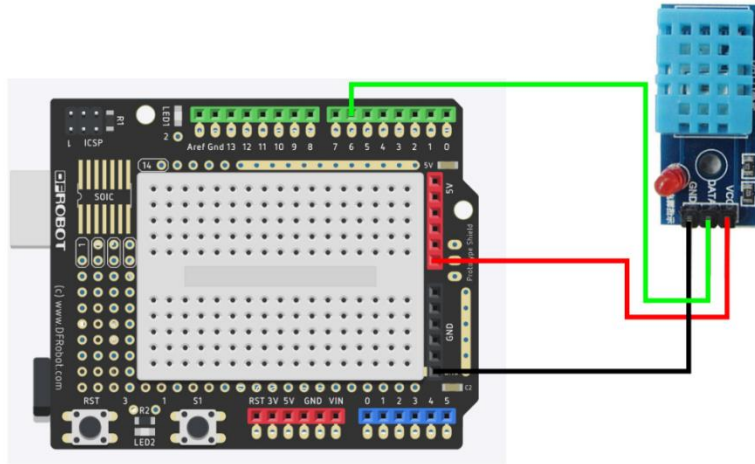
Εικόνα 13: Συνδεσμολογία GPRSShieldv1.0 με PrototypeShieldv.5

3.3 Συνδεσμολογία DHT11 temperature and humidity module

Η συνδεσμολογία του DHT11 είναι πολύ απλή και παρουσιάζεται στον πίνακα 2.

| DHT 11 | Prototype Shield v.5 |
|--------|----------------------|
| VCC | 5V |
| DAT | D6 |
| GND | GND |

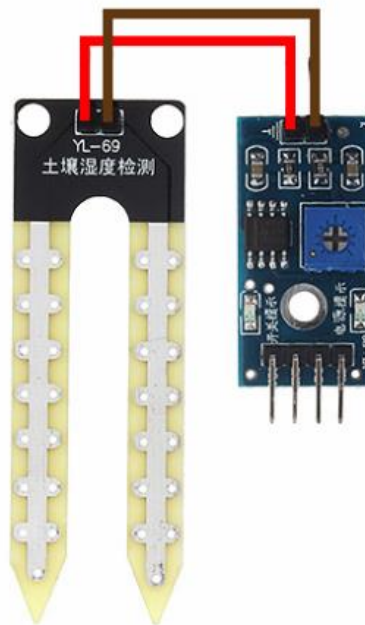
Πίνακας 2



Εικόνα 14: Συνδεσμολογία DHT11 με PrototypeShield v.5

3.4 Συνδεσμολογία Soil Hygrometer / Moisture Detection Module

Ο αισθητήρας Soul Moisture Detection Module αποτελείται από δύο κομμάτια, τον αισθητήρα και το εξάρτημα MH-Sensor-Series. Αρχικά πρέπει να συνδέσουμε τον αισθητήρα με το MH-Sensor-Series με βάση την εικόνα 15.

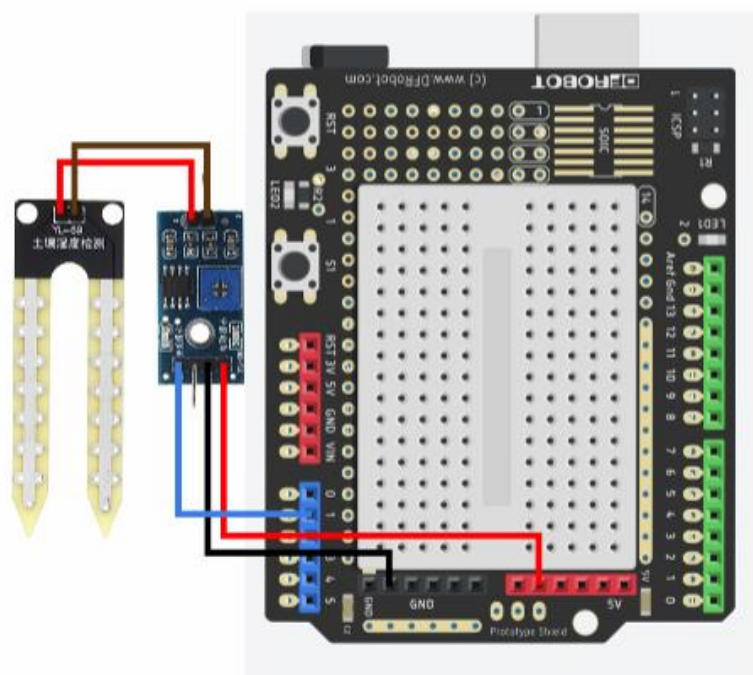


Εικόνα 15: Συνδεσμολογία αισθητήρα εδάφους με MH-Sensor-Series

Στη συνέχεια, η σύνδεση του MH-Sensor-Series γίνεται σύμφωνα με τον πίνακα 3

| MH-Sensor-Series | Prototype Shield v.5 |
|------------------|----------------------|
| VCC | 5V |
| GND | GND |
| A0 | A1 |

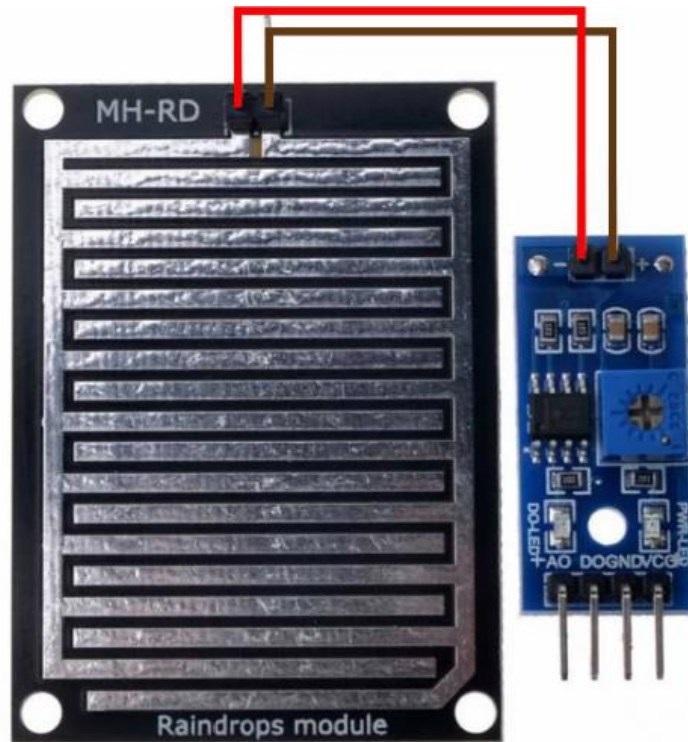
Πίνακας 3



Εικόνα 16: Συνδεσμολογία MH-Sensor-Series εδάφους με PrototypeShield v.5

3.5 Συνδεσμολογία MH-RDRaindropsmodule

Παρομοίως με τον αισθητήρα της υγρασίας εδάφους, και ο MH-RDRaindropsmodule πρέπει πρώτα να συνδεθεί με το MH-Sensor-Series, όπως φαίνεται στην παρακάτω εικόνα.

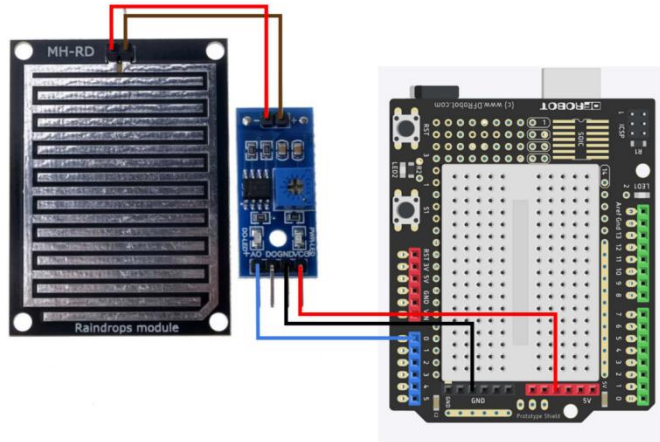


Εικόνα 17: Συνδεσμολογία MH-RD Raindrops module με MH-Sensor-Series

Έπειτα, ακολουθώντας τον πίνακα 4 συνδέουμε το MH-Sensor-Series στο Arduino.

| MH-Sensor-Series | Prototype Shield v.5 |
|------------------|----------------------|
| VCC | 5V |
| GND | GND |
| A0 | A0 |

Πίνακας 4



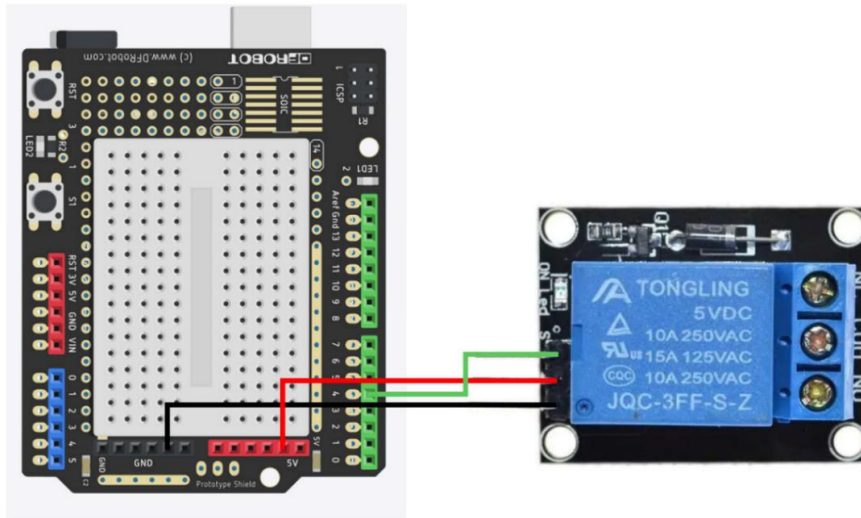
Εικόνα 18: Συνδεσμολογία MH-Sensor-Series βροχόπτωσης με PrototypeShield v.5

3.6 Συνδεσμολογία ZTS-3000-FSJT-V05 wind speed sensor

Για τη σωστή σύνδεση του αισθητήρα οφείλουμε να γνωρίζουμε πως το μαύρο καλώδιο είναι για τη γείωση, το καφέ καλώδιο είναι για την τάση και το μπλε καλώδιο είναι η αναλογική έξοδος, ενώ το κίτρινο καλώδιο δεν μας είναι χρήσιμο. Ο αισθητήρας θα συνδεθεί με το PrototypeShield v.5 όπως φαίνεται στον πίνακα 5.

| ZTS-3000-FSJT-V05 | Prototype Shield v.5 |
|-------------------|----------------------|
| GND(Μαύρο) | GND |
| VCC (Καφέ) | 5V |
| A0 (Μπλε) | A2 |

Πίνακας 5



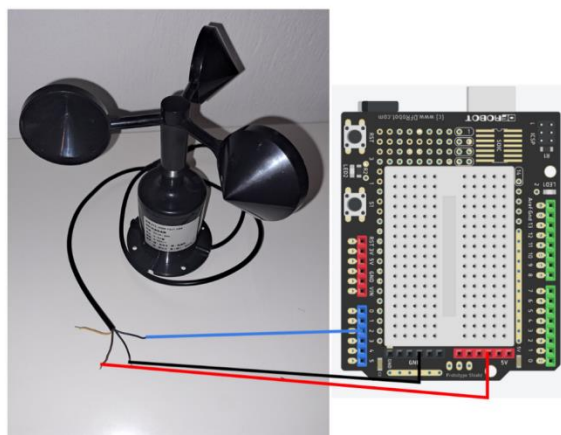
Εικόνα 19: Συνδεσμολογία wind speed sensor με Prototype Shield v.5

3.7 Συνδεσμολογία Tongling single channel 5V relay module

Για την σύνδεση του ρελέ θα χρειαστεί να ακολουθήσουμε τις ενδείξεις του πίνακα 6.

| Tongling single channel 5V relay module | Prototype Shield v.5 |
|---|----------------------|
| - | GND |
| + | 5V |
| S | D4 |

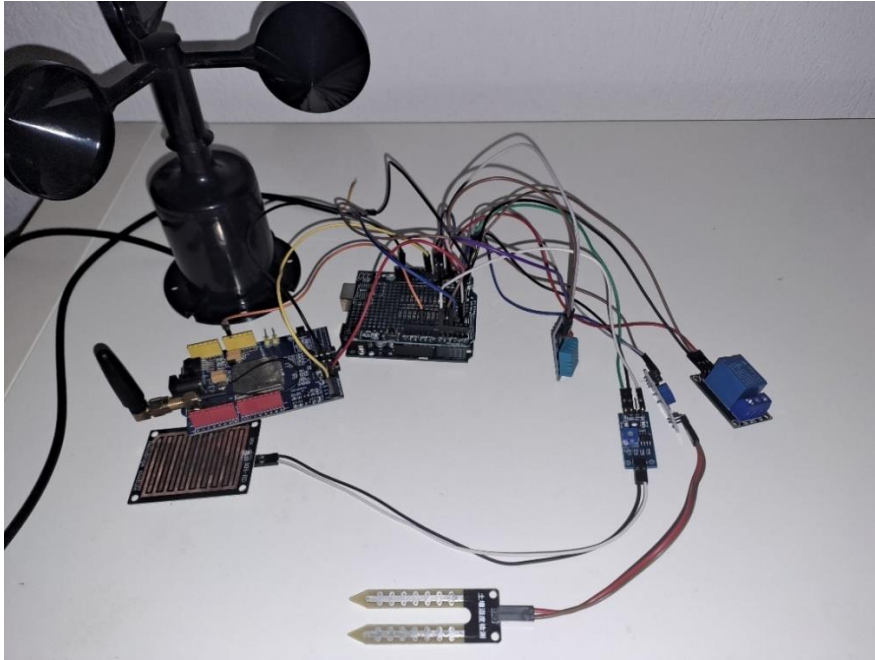
Πίνακας 6



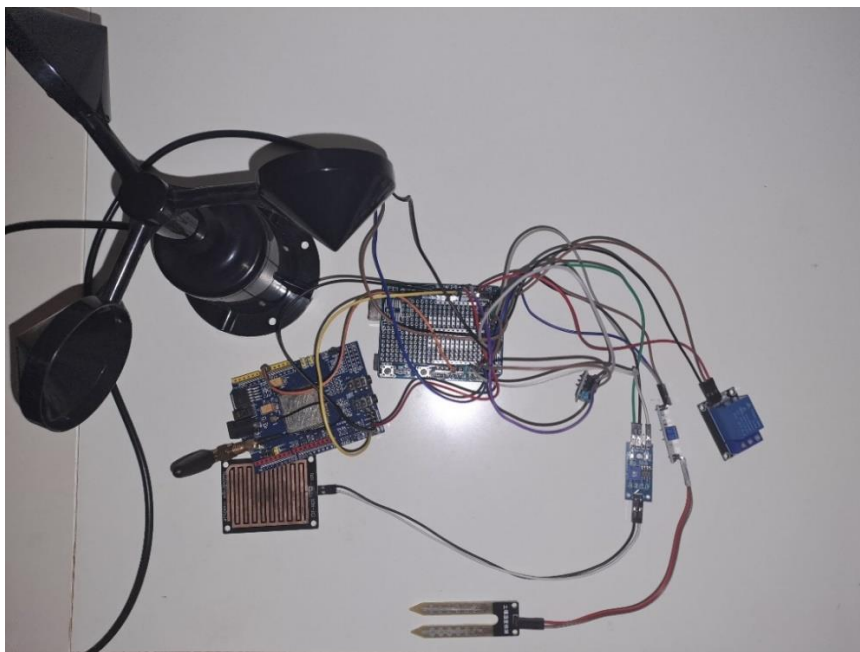
Εικόνα 20: Συνδεσμολογία Tongling single channel 5V relay module με Prototype Shield v.5

3.8 Τελική κατασκευή

Η τελική μορφή του συστήματος παρουσιάζεται στις παρακάτω εικόνες, προσφέροντας μια πλήρη και σαφή απεικόνιση της υλοποίησης.



Εικόνα 21: Συνολικές συνδέσεις 1



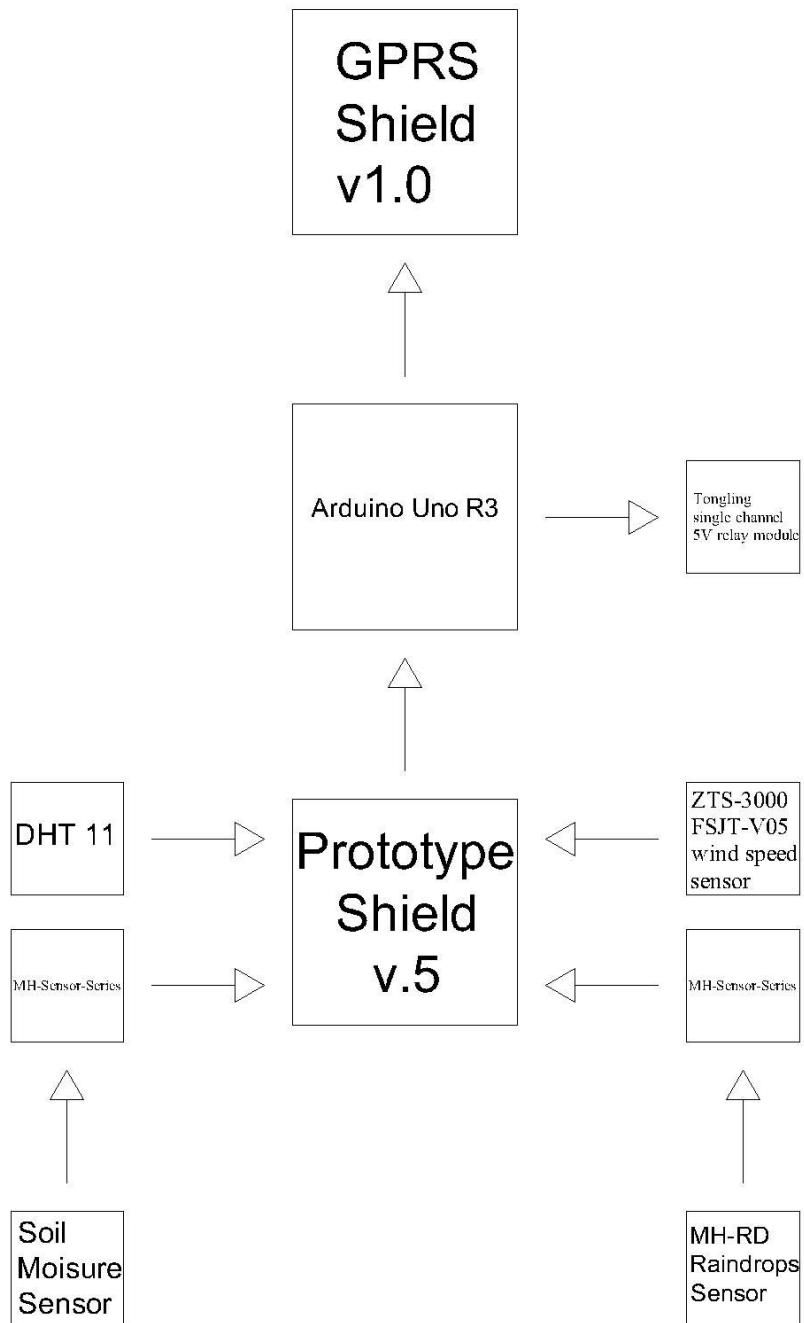
Εικόνα 22: Συνολικές συνδέσεις 2



Εικόνα 23: Συνολικές συνδέσεις 3

3.9 Block Διάγραμμα

Επιπλέον, παρακάτω παρατίθεται ένα block διάγραμμα που απεικονίζει την αλληλεπίδραση των επιμέρους στοιχείων της κατασκευής.



Εικόνα 24: Block Διάγραμμα

4. Λογισμικό

Για τις απαιτήσεις της εργασίας, επιλέχθηκε το περιβάλλον προγραμματισμού ArduinoIDE, το οποίο βασίζεται στη γλώσσα προγραμματισμού C++, προκειμένου να υλοποιηθεί ο προγραμματισμός του ArduinoUnoR3. Στη συνέχεια, θα παρατεθούν πληροφορίες σχετικά με το ArduinoIDE, ενώ ακολούθως θα ακολουθήσει αναλυτική επεξήγηση του κώδικα

4.1 Το περιβάλλον προγραμματισμού ArduinoIDE

Το ArduinoIDE (Integrated Development Environment) αποτελεί το επίσημο περιβάλλον ανάπτυξης λογισμικού για την πλατφόρμα Arduino. Πρόκειται για ένα εύχρηστο εργαλείο που επιτρέπει στους χρήστες να προγραμματίσουν μικροελεγκτές της σειράς Arduino χρησιμοποιώντας κυρίως τη γλώσσα προγραμματισμού C++. Το ArduinoIDE είναι σχεδιασμένο με σκοπό την απλότητα, καθιστώντας το ιδανικό τόσο για αρχάριους όσο και για προχωρημένους προγραμματιστές. Μέσα από την εύκολη και άμεση πρόσβαση που προσφέρει, οι χρήστες μπορούν να δημιουργήσουν, να μεταγλωττίσουν και να φορτώσουν τον κώδικά τους σε συσκευές Arduino, αξιοποιώντας το φιλικό περιβάλλον του κειμενογράφου και τα ενσωματωμένα εργαλεία που διαθέτει.

Ένα από τα κύρια χαρακτηριστικά του ArduinoIDE είναι η υποστήριξη πολλαπλών πλατφορμών, καθώς είναι συμβατό με Windows, macOS και Linux. Αυτό επιτρέπει στους χρήστες να αναπτύσσουν τις εφαρμογές τους ανεξάρτητα από το λειτουργικό σύστημα που χρησιμοποιούν. Το περιβάλλον προσφέρει μια ολοκληρωμένη λύση, καθώς περιλαμβάνει όλες τις απαραίτητες λειτουργίες για τη διαχείριση του hardware, μέσω βιβλιοθηκών που διευκολύνουν την αλληλεπίδραση με εξωτερικά εξαρτήματα όπως αισθητήρες, οθόνες και κινητήρες.

Η γλώσσα προγραμματισμού που χρησιμοποιείται στο ArduinoIDE είναι μια παραλλαγή της C++, με ειδικές προσαρμογές που την καθιστούν πιο προσιτή και εύκολη στη χρήση. Αυτή η προσέγγιση επιτρέπει στους χρήστες να εστιάσουν περισσότερο στη δημιουργικότητα και την ανάπτυξη των εφαρμογών τους, παρά στη διαχείριση πολύπλοκων τεχνικών λεπτομερειών. Η κοινότητα του Arduino, η οποία είναι ιδιαίτερα ενεργή, προσφέρει πλούσιο υλικό, όπως παραδείγματα και tutorials, που ενισχύουν την εμπειρία των χρηστών.

Το ArduinoIDE είναι ένα λογισμικό ανοικτού κώδικα, πράγμα που σημαίνει ότι ο κώδικας του είναι διαθέσιμος σε όλους για μελέτη, τροποποίηση και βελτίωση. Αυτή η ανοικτή φύση του IDE, σε συνδυασμό με την υποστήριξη από την παγκόσμια κοινότητα χρηστών, καθιστά δυνατή την ανάπτυξη και την προσαρμογή του σε διαφορετικές ανάγκες. Παράλληλα, το περιβάλλον υποστηρίζει την επέκταση των δυνατοτήτων του μέσω πρόσθετων (plugins) και τη διασύνδεση με άλλα εργαλεία ανάπτυξης, επιτρέποντας στους πιο προχωρημένους χρήστες να εκτελούν πιο περίπλοκα έργα.

Συνολικά, το ArduinoIDE προσφέρει ένα ιδανικό εργαλείο για την ανάπτυξη έργων που βασίζονται στο Arduino. Η απλότητά του, η διαθεσιμότητά του σε πολλαπλές πλατφόρμες και η πλούσια υποστήριξη από την κοινότητα το καθιστούν ένα από τα πιο δημοφιλή IDE για προγραμματισμό μικροελεγκτών, υποστηρίζοντας ένα ευρύ φάσμα χρηστών από αρχάριους έως προχωρημένους προγραμματιστές.

4.2 Σχολιασμός του κώδικα

Σε αυτό το κεφάλαιο, θα παρουσιαστεί ο κώδικας τμηματικά, συνοδευόμενος από φωτογραφίες που απεικονίζουν τα επιμέρους μέρη του. Κάθε τμήμα του κώδικα θα εξεταστεί

αναλυτικά και θα εξηγηθεί με σαφήνεια, προκειμένου να κατανοηθούν πλήρως οι λειτουργίες και οι ενέργειές του.

4.2.1 Βιβλιοθήκες και μεταβλητές

Στην παρακάτω εικόνα παρουσιάζεται η διάταξη και η ρύθμιση των απαραίτητων πληροφοριών για την ομαλή λειτουργία του συστήματος, περιλαμβάνοντας τις συνδέσεις των pin, τις ρυθμίσεις παραμέτρων, και τις συνδέσεις με τις συσκευές.

```
1  #include <SoftwareSerial.h>
2  #include <string.h>
3  #include <dht.h>
4
5  SoftwareSerial gprsSerial(7,8);
6  dht DHT;
7
8  #define dht11_pin 6
9
10 const int windsensorpin = A2;
11 const float vtagereference = 5.0;
12 const int analogmaxvalue = 1023;
13 const float maxwindspeed = 30;
14 int relayPin = 4;
15 const char* phoneNumber = "+306949308321";
16
```

Εικόνα 25: Βιβλιοθήκες και μεταβλητές κώδικα (1-16)

- Αρχικά, στις σειρές 1 έως 3 περιλαμβάνονται οι απαραίτητες βιβλιοθήκες για την ομαλή λειτουργία του κώδικα.
- Στη σειρά 5 δημιουργείται το αντικείμενο `gprsSerial`, το οποίο θα χρησιμοποιηθεί για την επικοινωνία με το `GPRSShield`. Ορίζεται ότι τα pin 7 και 8 θα χρησιμοποιηθούν για την

σειριακή επικοινωνία, με το pin 7 να είναι το RX (λήψη) και το pin 8 να είναι το TX (πομπός).

- Στη σειρά 6 δημιουργείται το αντικείμενο DHT το οποίο θα χρησιμοποιηθεί για την επικοινωνία με τον αισθητήρα DHT11, ενώ, στην σειρά 8 αντιστοιχείται ο αισθητήρας DHT11 με το pin 6.
- Στη σειρά 10 δημιουργείται η σταθερά windsensorpin με τιμή A2. Αυτή η σταθερά καθορίζει τον αναλογικό ακροδέκτη A2, ο οποίος συνδέεται με τον αισθητήρα ανέμου για τη μέτρηση της ταχύτητας του ανέμου.
- Στις σειρές 10 με 13 ορίζονται τέσσερις σταθερές οι οποίες είναι απαραίτητες για την ανάγνωση και επεξεργασία δεδομένων από τον αισθητήρα ανέμου, καθώς και για την μετατροπή των μετρήσεων από αναλογική σε ψηφιακή μορφή.
- Στη σειρά 14 αντιστοιχείται η μεταβλητή relayPin με το pin4 για την ενεργοποίηση και απενεργοποίηση του συστήματος άρδευσης.
- Στη σειρά 15 δημιουργείται η σταθερά phone Number, η οποία είναι ένας δείκτης σε μια συμβολοσειρά χαρακτήρων (string) που περιέχει τον αριθμό τηλεφώνου που θα χρησιμοποιηθεί για την αποστολή SMSστη συνέχεια.

4.2.2 voidsetup()

Στη συνέχεια, εξηγείται η συνάρτηση setup() στην οποία αρχικοποιούνται οι σειριακές επικοινωνίες, ρυθμίζεται και απενεργοποιείται το ρελέ και στέλνονται εντολές AT για την ρύθμιση του GPRS.

```

17 void setup()
18 {
19   Serial.begin(9600);
20   gprsSerial.begin(9600);
21   pinMode(relayPin, OUTPUT);
22   digitalWrite(relayPin, LOW);
23   gprsSerial.println("AT+CMGF=1");
24   gprsSerial.println("AT+CNMI=2,2,0,0,0");
25   delay(1000);
26 }
27

```

Εικόνα 26: void setup() (17-27)

- Στη σειρά 17 δημιουργούμε τη συνάρτηση setup() η οποία θα εκτελεστεί μία φορά κατά την αρχική εκκίνηση του προγράμματος.
- Στη σειρά 19 ξεκινάει η σειριακή επικοινωνία με ταχύτητα 9600 baud, επιτρέποντας την επικοινωνία μεταξύ του μικροελεγκτή και του υπολογιστή μέσω της θύρας USB για την παρακολούθηση ή αποστολή δεδομένων.
- Στη σειρά 20 ξεκινάει η σειριακή επικοινωνία με το GPRSShield v1.0 με ταχύτητα 9600baud, για την αποστολή και λήψη δεδομένων.
- Στη σειρά 21 ρυθμίζεται το pin που καθορίζεται από τη μεταβλητή “relayPin” ως έξοδο, ενώ στη σειρά 22 θέτεται η συγκεκριμένη έξοδο σε χαμηλή στάθμη, που σημαίνει πως απενεργοποιείται το ρελέ.
- Στη σειρά 23 στέλνεται η εντολή “AT+CMGF=1” στο GPRS για να τεθεί σε λειτουργία κειμένου, για την αποστολή και λήψη SMS.

- Στη σειρά 24 στέλνεται η εντολή “AT+CNMI=2,2,0,0,0” στο GPRS για τη ρύθμιση των παραμέτρων του για την εμφάνιση νέων μηνυμάτων SMSστη σειριακή έξοδο όταν παραλαμβάνονται.
- Στη σειρά 25 εισάγεται μία καθυστέρηση 1 δευτερολέπτου για να επιτραπεί στο σύστημα να ολοκληρώσει τυχόν τρέχουσες εργασίες και να εξασφαλίσουμε ότι οι εντολές έχουν εκτελεστεί σωστά πριν προχωρήσει στο επόμενο βήμα.

4.2.3 voidloop()

Σε αυτό το κομμάτι του κώδικα επεξηγείται η συνάρτηση loop() η οποία οφείλεται για τη συνεχή λειτουργία του συστήματος.

```
28 void loop()  
29 {  
30     configureGPRS();  
31     delay(1000);  
32     sendData();  
33     delay(1000);  
34 }  
35
```

Εικόνα 27: void loop() (28-35)

- Στις σειρές 28 έως 34 δημιουργούμε τη συνάρτηση loop() στην οποία, καλούνται διαδοχικά οι συναρτήσεις “configureGPRS()” και “sendData()”, οι οποίες θα αναλυθούν στη συνέχεια, με καθυστέρηση 1 δευτερολέπτου μεταξύ τους. Αυτή η διαδικασία επαναλαμβάνεται συνεχώς.

4.2.4 void configureGPRS()

Στις εικόνες 28 και 29 επεξηγείται η συνάρτηση `configureGPRS()` η οποία εκτελεί μια σειρά από εντολές AT μέσω του GPRS για την αρχικοποίηση και τη ρύθμιση της σύνδεσης δικτύου. Περιλαμβάνει εντολές για τον έλεγχο της κατάστασης SIM, της σύνδεσης στο δίκτυο και της σύνδεσης στο διαδίκτυο. Επιπλέον, ρυθμίζει την APN, δημιουργεί σύνδεση TCP με τον διακομιστή `api.thingspeak.com` για την αποστολή των δεδομένων. Σε κάθε βήμα, υπάρχει καθυστέρηση για την ολοκλήρωση των εντολών, και χρησιμοποιεί τη συνάρτηση `ctrlRelay()`, την οποία θα δούμε στη συνέχεια. Για λόγους ευανάγνωσης, θα παραλειφθεί η επανειλημμένη χρήση των εντολών `ctrlRelay()`, `delay()`, `ShowSerialData()` και θα επεξηγηθούν μία φορά.

```
36 void configureGPRS()
37 {
38     gprsSerial.println("AT");
39     delay(1000);
40     ctrlRelay();
41     delay(1000);
42     gprsSerial.println("AT+CPIN?");
43     delay(1000);
44     ctrlRelay();
45     delay(1000);
46     gprsSerial.println("AT+CREG?");
47     delay(1000);
48     ctrlRelay();
49     delay(1000);
50     gprsSerial.println("AT+CGATT?");
51     delay(1000);
52     ctrlRelay();
53     delay(1000);
54     gprsSerial.println("AT+CIPSHUT");
55     delay(1000);
56     ctrlRelay();
57     delay(1000);
58     gprsSerial.println("AT+CIPSTATUS");
59     delay(2000);
60     ctrlRelay();
61     delay(1000);
```

Εικόνα 28: void configureGPRS() (36-61)

- Στη σειρά 36 δημιουργούμε τη συνάρτηση “configureGPRS()” που διαχειρίζεται τη ρύθμιση του GPRS μέσω εντολών AT.
- Στη σειρά 38 στέλνεται η εντολή “AT” στο GPRS για να ελεγχθεί αν ανταποκρίνεται σωστά.
- Στη σειρά 39 εισάγεται μια καθυστέρηση 1 δευτερολέπτου για να επιτρέψουμε στο GPRS να επεξεργαστεί την προηγούμενη εντολή.
- Στη σειρά 40 καλείται η συνάρτηση “ctrlRelay()”.
- Στη σειρά 42 στέλνεται η εντολή “AT+CPIN?” στο GPRS για να ελεγχθεί η κατάσταση της κάρτας SIM.
- Στη σειρά 46 στέλνεται η εντολή “AT+CREG?” για να ελεγχθεί η κατάσταση της εγγραφής του GPRS στο δίκτυο κινητής τηλεφωνίας.
- Στη σειρά 50 στέλνεται η εντολή “AT+CGATT?” για να ελεγχθεί αν το GPRS είναι συνδεδεμένο στο GPRS δίκτυο.
- Στη σειρά 54 στέλνεται η εντολή “AT+CIPSHUT” για να κλείσει οποιοσδήποτε ανοιχτές συνδέσεις TCP/IP στο GPRS.
- Στη σειρά 58 στέλνεται η εντολή “AT+CIPSTATUS” για να ελεγχθεί την τρέχουσα κατάσταση των συνδέσεων TCP/IP.

```

62   gprsSerial.println("AT+CIPMUX=0");
63   delay(2000);
64   ctrlRelay();
65   delay(1000);
66   ShowSerialData();
67   gprsSerial.println("AT+CSTT=\"internet.vodafone.gr\"");
68   delay(1000);
69   ctrlRelay();
70   delay(1000);
71   ShowSerialData();
72   gprsSerial.println("AT+CIICR");
73   delay(3000);
74   ctrlRelay();
75   delay(1000);
76   ShowSerialData();
77   gprsSerial.println("AT+CIFSR");
78   delay(2000);
79   ctrlRelay();
80   delay(1000);
81   ShowSerialData();
82   gprsSerial.println("AT+CIPSPRT=0");
83   delay(3000);
84   ctrlRelay();
85   delay(1000);
86   ShowSerialData();
87   gprsSerial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\"");
88   delay(6000);
89   ctrlRelay();
90   delay(1000);
91   ShowSerialData();
92 }

```

Εικόνα 29: void configure GPRS() (62-92)

- Στη σειρά 62 στέλνεται η εντολή “AT+CIPMUX=0” για να ρυθμιστεί το GPRS σε λειτουργία single connection (μία σύνδεση TCP/IP).
- Στη σειρά 66 καλείται η συνάρτηση ShowSerialData() η οποία θα εξηγηθεί αργότερα.
- Στη σειρά 67 στέλνεται η εντολή “AT+CSTT=“internet.vodafone.gr”” για να ρυθμιστεί η APN (Access Point Name) του δικτύου κινητής τηλεφωνίας.

- Στη σειρά 72 στέλνεται η εντολή “AT+CIICR” για να φέρει σε λειτουργία τη σύνδεση δικτύου.
- Στη σειρά 77 στέλνεται η εντολή “AT+CIFSR” για την απόκτηση της τοπικής IP διεύθυνσης του GPRSShield.
- Στη σειρά 82 στέλνεται η εντολή “AT+CIPSPRT=0” για να απενεργοποιηθεί η αναφορά πρωτοκόλλου της σύνδεσης TCP/IP.
- Στη σειρά 87 στέλνεται η εντολή “AT+CIPSTART” για να ξεκινήσει μια σύνδεση TCP με τον διακομιστή “api.thingspeak.com” στη θύρα 80 για την αποστολή δεδομένων.

4.2.5 voidsendData(): συλλογή δεδομένων

Συνεχίζοντας με τη ροή του κώδικα, το επόμενο κομμάτι που θα αναλυθεί αφορά τη συνάρτηση sendData(), η οποία συλλέγει δεδομένα από διάφορους αισθητήρες, όπως θερμοκρασία, υγρασία, υγρασία εδάφους, βροχόπτωση και ταχύτητα ανέμου. Τα δεδομένα αυτά υπολογίζονται, μετατρέπονται σε ποσοστά ή άλλες μονάδες μέτρησης, και στη συνέχεια αποστέλλονται μέσω της σειριακής θύρας για παρακολούθηση ή περαιτέρω επεξεργασία.

```

93
94 void sendData()
95 {
96
97     int chk = DHT.read11 (dht11_pin);
98     int rr = analogRead(A0);
99     int sr = analogRead(A1);
100    int windsensorvalue = analogRead(windsensorpin);
101    float voltage = windsensorvalue * (voltage/reference / analogmaxvalue);
102    float windspeed = (voltage / voltage/reference) * maxwindspeed;
103    float soilmoisturepercent = map(sr, 0 , 1023, 100, 0);
104    float rainpercent = map(rr, 0, 1023, 100, 0);
105    delay(100);
106
107    Serial.print("Temperature = ");
108    Serial.print(DHT.temperature);
109    Serial.println(" °C");
110
111    Serial.print("Humidity = ");
112    Serial.print(DHT.humidity);
113    Serial.println(" %");
114
115    Serial.print("Soil Moisture = ");
116    Serial.print(soilmoisturepercent);
117    Serial.println(" %");
118
119    Serial.print("Rain Reading = ");
120    Serial.print(rainpercent);
121    Serial.println(" %");
122
123    Serial.print("Wind Speed =");
124    Serial.print(windspeed);
125    Serial.println(" m/s ");
126

```

Εικόνα 30: void sendData() (93-126)

- Στη σειρά 94 δημιουργούμε τη συνάρτηση sendData() η οποία συλλέγει τις μετρήσεις των αισθητήρων και στη συνέχεια τις στέλνει στην ιστοσελίδα ThingSpeak.
- Στη σειρά 97 διαβάζονται οι τιμές της θερμοκρασίας και υγρασίας από τον αισθητήρα DHT11 και αποθηκεύονται στη μεταβλητή “chk”.
- Στη σειρά 98 διαβάζεται η αναλογική τιμή από το pin A0, το οποίο είναι συνδεδεμένο στον αισθητήρα βροχόπτωσης και αποθηκεύει την τιμή στη μεταβλητή “rr”.

- Στη σειρά 99 διαβάζεται η αναλογική τιμή από το pinA1, το οποίο είναι συνδεδεμένο στον αισθητήρα υγρασίας του εδάφους και αποθηκεύει την τιμή στη μεταβλητή “sr”.
- Στη σειρά 100 διαβάζεται η αναλογική τιμή από το pin που ορίστηκε παραπάνω ως windsensorpin και αποθηκεύεται η τιμή στη μεταβλητή “windsensorvalue”.
- Στις σειρές 101 και 102 υπολογίζουμε τη ταχύτητα του ανέμου με βάση την τιμή που συλλέξαμε από τον αισθητήρα ανέμου.
- Στις σειρές 103 και 104 μετατρέπουμε τις αναλογικές τιμές των αισθητήρων υγρασίας εδάφους και βροχόπτωσης αντίστοιχα σε ποσοστό (0 – 100%).
- Στη σειρά 105 εισάγεται μια καθυστέρηση 100 χιλιοστών του δευτερολέπτου.
- Στις σειρές 107 έως 109 εκτυπώνεται η θερμοκρασία σε βαθμούς κελσίου.
- Στις σειρές 111 έως 113 εκτυπώνεται η υγρασία της ατμόσφαιρας σε ποσοστό %.
- Στις σειρές 115 έως 117 εκτυπώνεται το ποσοστό υγρασίας του εδάφους.
- Στις σειρές 119 έως 121 εκτυπώνεται το ποσοστό βροχόπτωσης
- Στις σειρές 123 έως 125 εκτυπώνεται η ταχύτητα του ανέμου σε m/s (μέτρα ανά δευτερόλεπτο).

4.2.6 voidsendData() : αποστολή δεδομένων

Στη συνέχεια της συνάρτησης sendData() αναλύεται η διαδικασία αποστολής δεδομένων στο ThingSpeak μέσω GPRS. Το πρόγραμμα αρχίζει την αποστολή των δεδομένων, δημιουργεί και στέλνει ένα μήνυμα που περιλαμβάνει τις μετρήσεις των αισθητήρων σε στη πλατφόρμα συλλογής δεδομένων που επιλέχθηκε. Μετά την αποστολή, ολοκληρώνεται η μετάδοση με ειδικό χαρακτήρα για να δηλωθεί το τέλος του μηνύματος. Τέλος κλείνει η σύνδεση με τον διακομιστή.

```

127 gprsSerial.println("AT+CIPSEND");
128 delay(4000);
129 ShowSerialData();
130
131
132 String str="GET https://api.thingspeak.com/update?api_key=EWGBSN427DRR6C3D&field1=" + String(DHT.temperature)
133                                     + "&field2="+String(DHT.humidity)
134                                     + "&field3="+String(soilmoisturepercent)
135                                     + "&field4="+String(rainpercent)
136                                     + "&field5=" + String(windspeed);
137 Serial.println(str);
138 gprsSerial.println(str);
139 delay(4000);
140 ShowSerialData();
141
142 gprsSerial.println((char)26);
143 delay(5000);
144 gprsSerial.println();
145 ShowSerialData();
146
147 gprsSerial.println("AT+CIPSHUT");
148 delay(100);
149 ShowSerialData();
150 }
151

```

Εικόνα 31: void sendData() (127-151)

- Στη σειρά 127 στέλνεται η εντολή “AT+CIPSEND” για να ξεκινήσει η αποστολή δεδομένων στον διακομιστή μέσω της ενεργής σύνδεσης GPRS.
- Στη σειρά 128 εισάγεται καθυστέρηση 4 δευτερολέπτων για να δοθεί χρόνος στην εντολή αποστολής να ξεκινήσει.
- Στη σειρά 129 καλείται η συνάρτηση ShowSerialData().
- Στη σειρά 132 δημιουργείται ένα αίτημα HTTPGET ως συμβολοσειρά (string) για την πλατφόρμα ThingSpeak, ενσωματώνοντας τις τιμές των αισθητήρων για θερμοκρασία, υγρασία, υγρασία εδάφους, βροχόπτωση και ταχύτητα ανέμου.
- Στη σειρά 137 εκτυπώνεται το αίτημα στη σειριακή θύρα για έλεγχο.
- Στη σειρά 138 στέλνεται το αίτημα μέσω του GPRS στον διακομιστή της πλατφόρμας ThingSpeak.

- Στη σειρά 142 στέλνεται ο χαρακτήρας Ctrl+Z, του οποίου ο κωδικός ASCII είναι 26, μέσω του GPRS για να δηλώσει το τέλος του μηνύματος που αποστέλλεται στον διακομιστή.
- Στη σειρά 144 στέλνεται μια κενή γραμμή μέσω του GRPS, διαχωρίζοντας την επόμενη εντολή και διασφαλίζοντας ότι η μετάδοση έχει ολοκληρωθεί.
- Στη σειρά 147 στέλνεται η εντολή "AT+CIPSHUT" για να κλείσει η σύνδεση TCP και να τερματιστεί η επικοινωνία με τον διακομιστή.

4.2.7 voidctrlRelay() : έλεγχος ρελέ

Το παρακάτω τμήμα του κώδικα ελέγχει για εισερχόμενα SMS μέσω του GPRS module. Οι πρώτες γραμμές του κώδικα χρησιμοποιούνται για debugging, εκτυπώνοντας πληροφορίες στη σειριακή θύρα σχετικά με την κατάσταση και το περιεχόμενο των ληφθέντων μηνυμάτων. Εάν εντοπιστεί ένα μήνυμα, το πρόγραμμα διαβάζει το περιεχόμενό του και, ανάλογα με την εντολή που περιλαμβάνεται ("Open" ή "Close"), ανοίγει ή κλείνει ένα ρελέ, ενημερώνοντας τον αποστολέα με ένα απαντητικό SMS.

```

152 void ctrlRelay()
153 {
154   int windsensorvalue = analogRead(windsensorpin);
155   float voltage = windsensorvalue * (voltageReference / analogmaxvalue);
156   float windspeed = (voltage / voltageReference) * maxwindspeed;
157   int rr = analogRead(A0);
158   float rainpercent = map(rr, 0, 1023, 100, 0);
159   delay(100);
160
161   Serial.println("Checking SMS");
162   delay(100);
163   while (gprsSerial.available())
164   {
165     String response = gprsSerial.readString();
166     if (response.indexOf("+CMT:") != -1)
167     {
168       Serial.println("Received");
169       delay(100);
170       Serial.println("Full response: ");
171       Serial.println(response);
172       delay(1000);
173
174       int msgStartIndex = response.indexOf("\",\"") + 3;
175       String msgContent = response.substring(msgStartIndex);
176       Serial.println("Message content: ");
177       Serial.println(msgContent);
178       delay(1000);
179
180       if (msgContent.indexOf("Open") != -1)
181       {
182         Serial.println("Relay open");
183         digitalWrite(relayPin, HIGH);
184         sendSMS(phoneNumber, "Relay opened.");
185         delay(1000);
186       }
187       else if (msgContent.indexOf("Close") != -1)
188       {
189         Serial.println("Relay closed");
190         digitalWrite(relayPin, LOW);
191         sendSMS(phoneNumber, "Relay closed.");
192         delay(1000);
193       }
194     }
195   }
196 }

```

Εικόνα 32: void ctrlRelay() (152-196)

- Στη σειρά 152 δημιουργούμε τη συνάρτηση ctrlRelay() η οποία είναι υπεύθυνη για το χειρισμό του ρελέ και την ενημέρωση του χρήστη για τυχών αλλαγές του μέσω SMS
- Στις σειρές 154 διαβάζεται η αναλογική τιμή από το pin που ορίστηκε παραπάνω ως windsensorpin και αποθηκεύεται η τιμή στη μεταβλητή “windsensorvalue”. Ενώ, στις σειρές 155 και 156 υπολογίζουμε τη ταχύτητα του ανέμου με βάση την τιμή που συλλέξαμε από τον αισθητήρα ανέμου.

- Στη σειρά 157 διαβάζεται η αναλογική τιμή από το pin A0, το οποίο είναι συνδεδεμένο στον αισθητήρα βροχόπτωσης και αποθηκεύει την τιμή στη μεταβλητή “tr”. Ενώ στη σειρά 158 μετατρέπουμε την αναλογική τιμή του αισθητήρα βροχόπτωσης σε ποσοστό (0 – 100%).
- Στη σειρά 161 εκτυπώνεται το μήνυμα “CheckingSMS” στη σειριακή θύρα, ενημερώνοντας ότι αρχίζει ο έλεγχος για εισερχόμενα SMS.
- Στη σειρά 163 γίνεται έλεγχος αν υπάρχουν διαθέσιμα δεδομένα για ανάγνωση από το GPRS. Εάν υπάρχουν, ο βρόχος “while” θα τα διαβάσει.
- Στη σειρά 165 διαβάζονται όλα τα διαθέσιμα δεδομένα από το GPRSως συμβολοσειρά και αποθηκεύονται στη μεταβλητή “response”.
- Στη σειρά 166 ελέγχεται εάν η συμβολοσειρά “response” περιέχει το κείμενο “+CMT:” το οποίο υποδηλώνει την άφιξη ενός νέου SMS.
- Στη σειρά 168 εκτυπώνεται το μήνυμα “Received” στη σειριακή θύρα, υποδεικνύοντας ότι ένα SMS έχει ληφθεί.
- Στις σειρές 170 και 171 εκτυπώνει το κείμενο “Fullresponse:” στη σειριακή θύρα και έπειτα εκτυπώνει ολόκληρο το περιεχόμενο του SMSστη σειριακή θύρα.
- Στη σειρά 174 γίνεται εύρεση της θέσης στην οποία ξεκινά το περιεχόμενο του μηνύματος SMS.
- Στη σειρά 175 δημιουργείται μια νέα συμβολοσειρά “msgContent” που περιέχει μόνο το περιεχόμενο του μηνύματος, ξεκινώντας από τη θέση που βρέθηκε προηγουμένως.
- Στη σειρά 176 εκτυπώνεται το μήνυμα “Messagecontent:” στη σειριακή θύρα. Στη συνέχεια, στη σειρά 177 εκτυπώνεται το περιεχόμενο του μηνύματος χωρίς τα επιπρόσθετα στοιχεία του SMS.

- Στις σειρές 180 έως 185 ελέγχεται εάν το περιεχόμενο του μηνύματος SMS περιέχει τη λέξη “Open”. Εάν ναι, τότε θα ενεργοποιήσει το ρελέ και θα καλέσει τη συνάρτηση sendSMS() για να στείλει στο προκαθορισμένο τηλέφωνο ένα SMS, ενημερώνοντας ότι το ρελέ έχει ανοίξει.
- Στις σειρές 187 έως 192 ελέγχεται εάν το περιεχόμενο του μηνύματος περιέχει τη λέξη “Close”. Εάν ναι, τότε θα απενεργοποιήσει το ρελέ και θα καλέσει τη συνάρτηση sendSMS() για να στείλει SMS το οποίο θα ενημερώνει ότι το ρελέ έχει κλείσει.

4.2.8 void ctrlRelay() : αυτόματη απενεργοποίηση ρελέ

Στο επόμενο κομμάτι του CtrlRelay() γίνεται ο έλεγχος της ταχύτητας και της βροχόπτωσης. Εάν οι μετρήσεις είναι μεγαλύτερες από 10m/s και 50% αντίστοιχα, τότε το ρελέ θα κλείσει και ο χρήστης θα ενημερωθεί μέσω SMS.

```

197   int relayState = digitalRead(relayPin);
198   if (relayState == HIGH)
199   {
200     if (windspeed > 10)
201     {
202       delay(5000);
203       digitalWrite(relayPin, LOW);
204       sendSMS(phoneNumber, "Relay closed due to high wind speed");
205       delay(1000);
206     }
207     if (rainpercent > 50)
208     {
209       delay(5000);
210       digitalWrite(relayPin, LOW);
211       sendSMS(phoneNumber, "Relay closed due to rain");
212       delay(1000);
213     }
214   }
215 }
216

```

Εικόνα 33: void ctrlRelay() (197-216)

- Στη σειρά 197 διαβάζεται η κατάσταση του ρελέ (ανοιχτό ή κλειστό) και αποθηκεύεται στη μεταβλητή relayState

- Στη σειρά 198 γίνεται έλεγχος της κατάστασης του ρελέ και αν το ρελέ είναι ανοιχτό, τότε, εκτελείται ο κώδικας που ακολουθεί.
- Στις σειρές 200 έως 205 γίνεται έλεγχος της ταχύτητας του ανέμου. Εάν η ταχύτητα είναι μεγαλύτερη από 10 m/s, τότε το ρελέ θα κλείσει και θα σταλθεί SMS ενημερώνοντας πως το ρελέ έχει κλείσει λόγω υψηλής ταχύτητας ανέμου.
- Στις σειρές 207 έως 212 γίνεται έλεγχος της του ποσοστού της βροχόπτωσης. Εάν το ποσοστό της βροχόπτωσης είναι μεγαλύτερο από 50%, τότε το ρελέ θα κλείσει και θα σταλθεί SMS ενημερώνοντας πως το ρελέ έκλεισε λόγω βροχόπτωσης.

4.2.9 voidShowSerialData()

Η βοηθητική συνάρτηση ShowSerialData() διαβάζει δεδομένα από το GPRSmodule και τα εκτυπώνει στη σειριακή θύρα, επιτρέποντας την παρακολούθηση των δεδομένων σε πραγματικό χρόνο, για σκοπούς ελέγχου και debugging.

```

217 void ShowSerialData()
218 {
219     while (gprsSerial.available() != 0)
220     {
221         Serial.write(gprsSerial.read());
222     }
223 }
224

```

Εικόνα 34: void ShowSerialData() (217-224)

- Στη σειρά 217 δημιουργούμε τη συνάρτηση ShowSerialData().
- Στη σειρά 219 εκτελείται ένας βρόχος “while” όσο υπάρχουν διαθέσιμα δεδομένα στο GPRS. Η συνθήκη “gprsSerial.available() !=0” ελέγχει εάν ο αριθμός των διαθέσιμων byte για ανάγνωση είναι διαφορετικός από μηδέν.

- Στη σειρά 221 η εντολή “gprsSerial.read()” διαβάζει ένα byte από τα διαθέσιμα δεδομένα του GPRS και η εντολή “Serial.write()” το στέλνει στη σειριακή θύρα.

4.2.10 voidsendSMS()

Τέλος, η συνάρτηση sendSMS() χρησιμοποιείται για την αποστολή ενός SMS μέσω του GPRSmodule. Ξεκινά στέλνοντας την απαραίτητη εντολή AT για να καθορίσει τον αριθμό παραλήπτη, ακολουθούμενη από την αποστολή του μηνύματος. Στη συνέχεια, η συνάρτηση στέλνει το χαρακτήρα 26 (Ctrl+Z) για να υποδείξει το τέλος του μηνύματος. Τέλος, η συνάρτηση εκτυπώνει οποιαδήποτε απάντηση από το GPRSmodule στη σειριακή θύρα, επιτρέποντας την παρακολούθηση της διαδικασίας αποστολής του SMS.

```
225 void sendSMS(const char* number, const char* message)
226 {
227     gprsSerial.print("AT+CMGS=\"");
228     gprsSerial.print(number);
229     gprsSerial.println("\");
230     delay(1000);
231     gprsSerial.print(message);
232     delay(3000);
233     gprsSerial.write(26);
234     delay(2000);
235     while (gprsSerial.available())
236     {
237         char c = gprsSerial.read();
238         Serial.print(c);
239     }
240 }
```

Εικόνα 35: void sendSMS() (225-240)

- Στη σειρά 225 ορίζεται η συνάρτηση “sendSMS()”, η οποία δέχεται δύο παραμέτρους: “number” , δηλαδή τον αριθμό τηλεφώνου του παραλήπτη που ορίσαμε στην αρχή και “message” , δηλαδή το μήνυμα που θα στείλει.

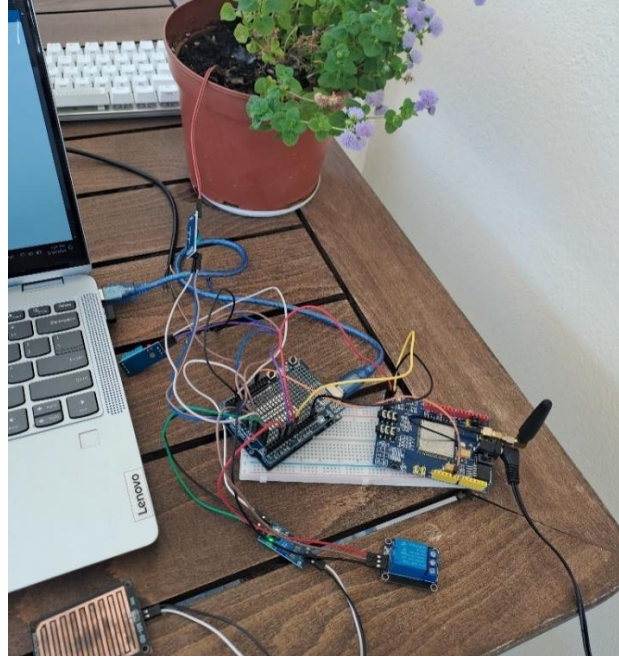
- Στη σειρά 227 στέλνεται η εντολή AT στο GPRS για να ξεκινήσει τη διαδικασία αποστολής SMS. Το "AT+CMGS=\"\" είναι το πρώτο μέρος της εντολής που καθορίζει ότι πρόκειται να ακολουθήσει ο αριθμός τηλεφώνου παραλήπτη.
- Στη σειρά 228 στέλνεται ο αριθμός τηλεφώνου του παραλήπτη που έχει περαστεί ως παράμετρος στη συνάρτηση.
- Στη σειρά 229 ολοκληρώνεται η εντολή με το κλείσιμο των εισαγωγικών και την αποστολή της στο GPRS.
- Στη σειρά 231 στέλνεται το περιεχόμενο του μηνύματος SMS στο GPRS. Το μήνυμα είναι αυτό που έχει περαστεί ως παράμετρος στη συνάρτηση.
- Στη σειρά 233 στέλνεται ο χαρακτήρας 26 (Ctrl+Z) στο GPRS για να υποδείξει το τέλος του μηνύματος.
- Στη σειρά 235 γίνεται έλεγχος της ύπαρξης διαθέσιμων δεδομένων στη σειριακή θύρα του GPRS. Εάν ναι, τότε στη σειρά 237 διαβάζεται ένας χαρακτήρας από τα διαθέσιμα δεδομένα του GPRS και τον αποθηκεύει στη μεταβλητή "c".
- Τέλος, στη σειρά 238 εκτυπώνεται το "c" στη σειριακή θύρα του Arduino.

5. Παρουσίαση λειτουργίας

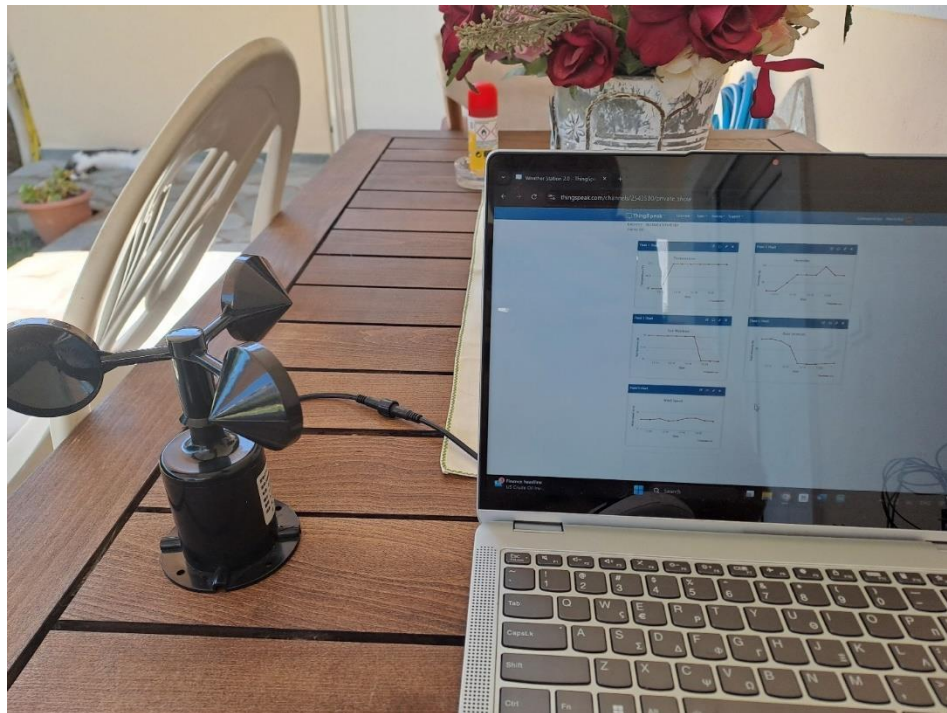
Σε αυτό το κεφάλαιο, αναλύεται η διαδικασία ενεργοποίησης του μετεωρολογικού σταθμού, περιγράφοντας τα απαραίτητα βήματα για τη σωστή λειτουργία του συστήματος και παρουσιάζεται η διαδικασία μεταφόρτωσης του κώδικα στον μικροελεγκτή. Επιπλέον, εξετάζεται η παρακολούθηση των μετρήσεων σε πραγματικό χρόνο μέσω του SerialMonitor, καθώς και η αποστολή και απεικόνιση των δεδομένων στην πλατφόρμα ThingSpeak, όπου οι μετρήσεις οργανώνονται σε πίνακες για ευκολότερη ανάλυση. Τέλος, γίνεται αναφορά στον απομακρυσμένο έλεγχο του ρελέ, ο οποίος επιτρέπει την διαχείριση των λειτουργιών άρδευσης εξ αποστάσεως, προσφέροντας ευελιξία και αυτοματοποίηση στον έλεγχο του συστήματος.

5.1 Ενεργοποίηση του μετεωρολογικού σταθμού

Με την ολοκλήρωση του σχεδιασμού του κώδικα, σειρά έχει η σύνδεση του Arduino με τον υπολογιστή μέσω του κατάλληλου καλωδίου USB και τη τροφοδοσία του GPRSShield με φορτιστή ισχύος 1A. Το επόμενο βήμα της διαδικασίας είναι η ενεργοποίηση της σύνδεσης στο δίκτυο του GPRS με το πάτημα του κουμπιού ενεργοποίησης για περίπου 2 δευτερόλεπτα. Η επιτυχής σύνδεση στο δίκτυο σηματοδοτείται από την καθορισμένη ακολουθία του netlightLED, η οποία είναι 64ms αναμμένο και 3000ms σβηστό.



Εικόνα 36: Ο μετεωρολογικός σταθμός



Εικόνα 37: Ο αισθητήρας ανέμου του μετεωρολογικού σταθμού

5.2 Μεταφόρτωση του κώδικα

Στη συνέχεια, γίνεται ανάρτηση του κώδικα στο Arduino μέσω του IDE. Με την επιτυχή μεταφόρτωση του κώδικα, το Arduino αρχίζει να συλλέγει δεδομένα από τους συνδεδεμένους αισθητήρες και τα στέλνει στο κανάλι του ThingSpeak για ανάλυση σε πραγματικό χρόνο. Η διαδικασία της συλλογής δεδομένων και της αποστολής τους διαρκεί ένα λεπτό και δέκα δευτερόλεπτα.



```
1 #include <SoftwareSerial.h>
2 #include <string.h>
3 #include <dht.h>
4
5 SoftwareSerial gprsSerial(7,8);
6 dht DHT;
7
8 #define dht11_pin 6
9
10 const int windsensorpin = A2;
11 const float voltagereference = 5.0;
12 const int analogmaxvalue = 1023;
13 const float maxwindspeed = 30;
14 int relayPin = 4;
15 const char* phoneNumber = "+306987696901";
16
17 void setup()
18 {
19   Serial.begin(9600);
20   gprsSerial.begin(9600);
21   pinMode(relayPin, OUTPUT);
22   digitalWrite(relayPin, LOW);
23   gprsSerial.println("AT+CMGF=1");
24   gprsSerial.println("AT+CNMI=2,2,0,0,0");
25   delay(1000);
26 }
27
28 void loop()
29 {
30   configureGPRS();
31   delay(1000);
```

Εικόνα 38: Μεταφόρτωση κώδικα

```
Temperature = 36.00 °C
Humidity = 33.00 %
Soil Moisture = 71.00 %
Rain Reading = 0.00 %
Wind Speed =12.40 m/s
AT+CIPSEND
GET https://api.thingspeak.com/update?api_key=EWGBSN427DRR6C3D&field1=36.00&field2=33.00&field3=71.00&field4=0.00&field5=12.40
GET https://api.thingspeak.com/update?api_key=EWGBSN427DRR6C3D&
SEND OK
335
```

Εικόνα 39: Πρώτο σύνολο μετρήσεων του κώδικα στο SerialMonitor

```

Temperature = 37.00 °C
Humidity = 31.00 %
Soil Moisture = 72.00 %
Rain Reading = 9.00 %
Wind Speed =14.63 m/s
AT+CIPSEND
GET https://api.thingspeak.com/update?api_key=EWGBSN427DRR6C3D&field1=37.00&field2=31.00&field3=72.00&field4=9.00&field5=14.63
GET https://api.thingspeak.com/update?api_key=EWGBSN427DRR6C3D&
SEND OK

```

Εικόνα 40: 10^ο σύνολο μετρήσεων του κώδικα στο SerialMonitor



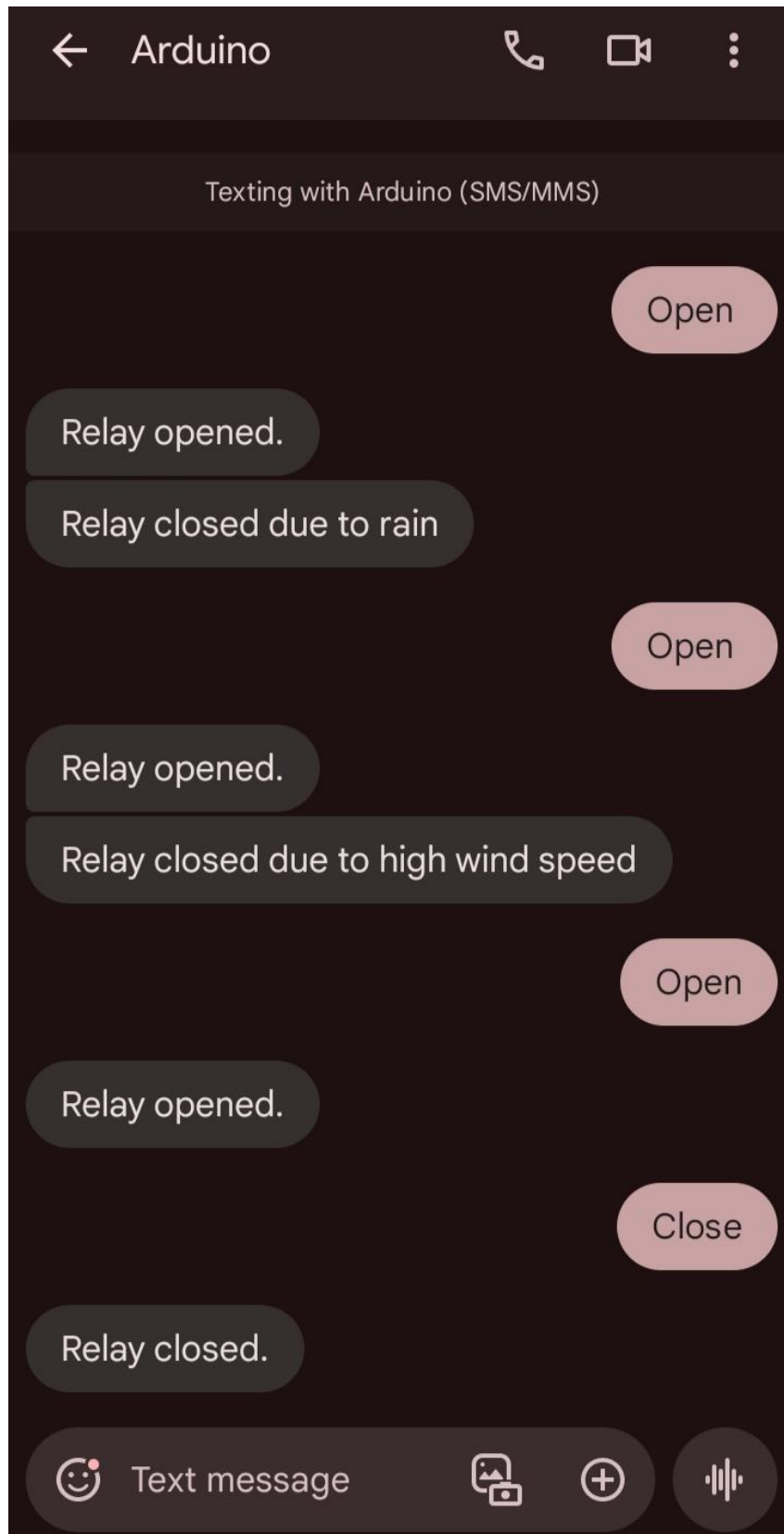
Εικόνα 41: Διαγράμματα των 10 συνόλων μετρήσεων στο ThingSpeak

5.3 Ασύρματος έλεγχος ρελέ

Κατά τη διάρκεια της διαδικασίας αρχικοποίησης και ρύθμισης του δικτύου, ο χρήστης έχει τη δυνατότητα να στείλει σε SMS τα μηνύματα “Open” ή “Close” στο GPRS, ώστε να ανοίξει ή να κλείσει το ρελέ αντίστοιχα, για την άρδευση του χωραφιού. Όμως, στο χρονικό διάστημα στο οποίο το Arduino συλλέγει τα δεδομένα και τα αποστέλλει στην πλατφόρμα, η ενέργεια χειρισμού του ρελέ είναι προσωρινά αδύνατη.

```
+CMT: "+306987696901", "", "24/08/14,13:02:58+12"  
Open  
  
Message content:  
", "24/08/14,13:02:58+12"  
Open  
  
Relay open  
AT+CMGS="+306987696901"  
  
> Relay opened.  
+CMGS: 32  
  
OK
```

Εικόνα 42: Δείγμα απάντησης στον έλεγχο του ρελέ μέσω SerialMonitor



Εικόνα 43: Δείγμα συνομιλίας SMS με το Arduino

6. Κόστος υλικών κατασκευής μετεωρολογικού σταθμού Arduino

| Υλικά | Κόστος (€) |
|---|--------------|
| Arduino Uno R3 | 16,80 |
| Prototype Shield v.5 | 3,80 |
| GPRS Shield v1.0 | 29,90 |
| DHT11 | 3,90 |
| Soil Moisture detection module | 1,90 |
| MH-RD Raindrops module | 1,80 |
| ZTS-3000-FSJT-V05 wind speed sensor | 24,85 |
| Tongling single channel 5V relay module | 1,80 |
| Καλώδια | 3 |
| Σύνολο | 87,75 |

Πίνακας 7

Όπως φαίνεται στον παραπάνω πίνακα, το συνολικό κόστος της κατασκευής υπολογίζεται περίπου στα 88€. Επομένως, μπορούμε να συμπεράνουμε πως είναι μια αρκετά οικονομική λύση για έναν ερασιτέχνη.

7. Βελτιώσεις και αναβαθμίσεις

Με την ολοκλήρωση της κατασκευής του μετεωρολογικού σταθμού, είναι σημαντική η σκέψη βελτίωσης και εξέλιξής του. Αρχικά, θα μπορούσαν να προστεθούν επιπλέον αισθητήρες για τη συλλογή περισσότερων δεδομένων, όπως ένας αισθητήρας ατμοσφαιρικής πίεσης, ένας αισθητήρας ηλιοφάνειας και ένας βροχογράφος. Επίσης, η κατασκευή θα μπορούσε να αναβαθμιστεί ώστε να γίνει φορητή, τροφοδοτώντας την με ηλιακή ενέργεια και μπαταρίες, καθιστώντας την κατάλληλη για χρήση σε απομακρυσμένα χωράφια.

Μια σημαντική βελτίωση που πρέπει να γίνει, αφορά τον κώδικα. Επί του παρόντος, όταν το Arduino στέλνει δεδομένα στην πλατφόρμα ThingSpeak, δεν είναι δυνατός ο έλεγχος του ρελέ μέσω SMS. Το πρόβλημα αυτό μπορεί να επιλυθεί με την εφαρμογή τεχνικών non-blockingprogramming, επιτρέποντας την ταυτόχρονη αποστολή δεδομένων και τον έλεγχο του ρελέ, βελτιώνοντας έτσι τη συνολική απόδοση και τη χρηστικότητα του συστήματος.

Επιπρόσθετα, με την επέκταση της συλλογής δεδομένων και τη βελτίωση της αποδοτικότητας στην αποστολή τους, καθώς και στον έλεγχο του συστήματος άρδευσης, μπορεί να ενσωματωθεί ένα σύστημα για την αποστολή περισσότερων αυτοματοποιημένων ειδοποιήσεων μέσω SMS σε περίπτωση έντονων καιρικών φαινομένων. Το συγκεκριμένο θα διασφαλίσει ότι ο αγρότης θα είναι πλήρως ενημερωμένος, χωρίς να χρειάζεται να ελέγξει χειροκίνητα την πλατφόρμα ThingSpeak για την κατάσταση του χωραφιού του.

Τέλος, με την ενσωμάτωση αλγορίθμων για την ανάλυση των δεδομένων, είναι εφικτή η προσθήκη μιας λειτουργίας πρόβλεψης των καιρικών συνθηκών, τα αποτελέσματα της οποίας θα εμφανίζονται επίσης στο κανάλι του ThingSpeak.

8. Συμπεράσματα

Η ιστορία των μετεωρολογικών σταθμών ξεκινά από την ανάγκη του ανθρώπου να κατανοήσει και να προβλέψει τις καιρικές συνθήκες, οι οποίες επηρεάζουν άμεσα την καθημερινή ζωή και τις δραστηριότητές του. Από τα αρχαία χρόνια, όπου οι παρατηρήσεις γίνονταν με γυμνό μάτι, μέχρι την ανακάλυψη των πρώτων μετεωρολογικών οργάνων, όπως το βαρόμετρο και το θερμόμετρο, η παρακολούθηση του καιρού αποτέλεσε κρίσιμο στοιχείο για την επιβίωση και την ανάπτυξη των κοινωνιών. Καθώς η τεχνολογία προχωρούσε, οι μετεωρολογικοί σταθμοί εξελίχθηκαν, ενσωματώνοντας όλο και πιο εξελιγμένα όργανα μέτρησης και δυνατότητες καταγραφής και ανάλυσης δεδομένων. Σήμερα, οι σύγχρονοι μετεωρολογικοί σταθμοί μπορούν να παρέχουν σε πραγματικό χρόνο ακριβείς προβλέψεις για τις καιρικές συνθήκες, παίζοντας ζωτικό ρόλο στην ασφάλεια και την ευημερία.

Η αλληλεπίδραση μεταξύ των μετεωρολογικών φαινομένων και της γεωργίας είναι θεμελιώδης. Οι καιρικές συνθήκες, όπως η βροχόπτωση, η θερμοκρασία, η υγρασία και η ταχύτητα του ανέμου, επηρεάζουν άμεσα την παραγωγικότητα και την υγεία των καλλιεργειών. Η δυνατότητα παρακολούθησης και ανάλυσης αυτών των συνθηκών επιτρέπει στους γεωργούς να λαμβάνουν πιο ενημερωμένες αποφάσεις για την άρδευση, τη λίπανση και την προστασία των φυτών από ακραία καιρικά φαινόμενα. Με αυτόν τον τρόπο, η χρήση μετεωρολογικών δεδομένων στη γεωργία δεν είναι απλώς χρήσιμη, αλλά αναγκαία για την επίτευξη βιώσιμης και αποδοτικής γεωργικής παραγωγής.

Επιπρόσθετα, με την εξέλιξη της τεχνολογίας και τη διαθεσιμότητα πιο οικονομικών και προσαρμόσιμων λύσεων, όπως το Arduino, η παρακολούθηση των μετεωρολογικών δεδομένων έχει γίνει πιο προσβάσιμη. Το Arduino, έχει επιτρέψει στους κατασκευαστές και τους ερευνητές να δημιουργήσουν προσαρμοσμένα συστήματα που μπορούν να συλλέγουν, να αναλύουν και να

ανταποκρίνονται σε δεδομένα σε πραγματικό χρόνο. Μέσα από τη πλατφόρμα ThingSpeak, αυτά τα δεδομένα μπορούν να αποθηκευτούν και να αναλυθούν, επιτρέποντας την παρακολούθηση της κατασκευής ενός χωραφιού από απόσταση, ενώ παράλληλα παρέχουν δυνατότητες αυτοματοποιημένου ελέγχου συστημάτων άρδευσης και άλλων λειτουργιών.

Επιπλέον, η χρήση του Arduino διαδραματίζει σημαντική εκπαιδευτική αξία, καθώς επιτρέπει στους μαθητές και στους ερασιτέχνες να πειραματιστούν με τη δημιουργία συστημάτων που ενσωματώνουν αισθητήρες, μικροελεγκτές και λογισμικό. Μέσα από αυτόν τον πειραματισμό, οι χρήστες αποκτούν βαθύτερη κατανόηση της τεχνολογίας και της επιστήμης πίσω από τη μετεωρολογία και τη γεωργία, ενώ παράλληλα καλλιεργούν δεξιότητες που μπορούν να χρησιμοποιηθούν σε πιο εξελιγμένες και καινοτόμες εφαρμογές. Το Arduino, με την ευελιξία και την προσαρμοστικότητα του, γίνεται εργαλείο που γεφυρώνει την απόσταση μεταξύ θεωρίας και πράξης.

Η συγκεκριμένη πτυχιακή εργασία αποτελεί μια προσπάθεια να χτιστεί μια γέφυρα μεταξύ της παραδοσιακής γεωργίας και της σύγχρονης τεχνολογίας, αξιοποιώντας έναν οικονομικό και προσιτό μετεωρολογικό σταθμό βασισμένο στο Arduino. Ο σταθμός αυτός, αν και σχεδιασμένος για ερασιτεχνική χρήση, προσφέρει αξιόπιστα δεδομένα και έχει τη δυνατότητα να στέλνει τις μετρήσεις του σε βάσεις δεδομένων στο διαδίκτυο, όπως η πλατφόρμα ThingSpeak. Η κατασκευή είναι επεκτάσιμη και αναβαθμίσιμη, δίνοντας τη δυνατότητα για την προσθήκη νέων αισθητήρων, όπως αισθητήρας ατμοσφαιρικής πίεσης ή ηλιοφάνειας, αλλά και την ενσωμάτωση ηλιακής ενέργειας για τη δημιουργία ενός φορητού και βιώσιμου σταθμού.

Ένα δίκτυο από τέτοιους μετεωρολογικούς σταθμούς θα μπορούσε να αναπτυχθεί, συνεισφέροντας ουσιαστικά στη συλλογή δεδομένων για μετεωρολογικές προβλέψεις και στατιστικές αναλύσεις. Παράλληλα, παρουσιάστηκαν τεχνικές και μέθοδοι προγραμματισμού

του μικροελεγκτή Arduino που επιτρέπουν την αυτοματοποίηση της άρδευσης, την αποστολή ειδοποιήσεων μέσω SMS και τον απομακρυσμένο έλεγχο μέσω δικτύου. Το έργο αυτό καταδεικνύει πώς η τεχνολογία, όταν χρησιμοποιείται σωστά, μπορεί να βελτιώσει την καθημερινότητα, διευκολύνοντας τη γεωργία και συμβάλλοντας στην ανάπτυξη πιο αποδοτικών και βιώσιμων καλλιεργητικών πρακτικών.

Με τη χρήση αυτού του μετεωρολογικού σταθμού, έγινε ένα σημαντικό βήμα προς τη δημιουργία ενός οικονομικά προσιτού, αλλά ταυτόχρονα ισχυρού εργαλείου για την υποστήριξη της γεωργίας. Ο σταθμός όχι μόνο προσφέρει αξιόπιστα δεδομένα για τις καιρικές συνθήκες αλλά επίσης ανοίγει τον δρόμο για πιο εξελιγμένες και καινοτόμες λύσεις στον τομέα της γεωργίας, επιδεικνύοντας πώς απλά τεχνολογικά μέσα μπορούν να οδηγήσουν σε σημαντικές βελτιώσεις στη διαχείριση των καλλιεργειών και την πρόγνωση των καιρικών φαινομένων.

Βιβλιογραφικές αναφορές

Δάρμα, Σ. Α. (2015). *Ανάλυση μετεωρολογικών μετρήσεων μέσω αισθητήρων με τη χρήση του μικροελεγκτή Arduino*.

Καραχάλιος, Θ. (2018). *Δημιουργία αυτόνομου συστήματος καταγραφής καιρικών μεγεθών*.

Μαργαρίτης, Ε. (2018). *Διδακτικά σενάρια για την κατανόηση φυσικών εννοιών με χρήση Arduino*.

Πασχάλης, Δ. (2016). *Arduino και αυτοματισμοί*. Αθήνα: Τεχνικές Εκδόσεις.

Τσιώρης, Ι. (2019). *Arduino: Εισαγωγή στον προγραμματισμό και τις εφαρμογές*. Θεσσαλονίκη: Εκδόσεις Ζήτη.

Παπαδάκης, Ν. (2017). *Μετεωρολογία και Κλιματολογία*. Αθήνα: Παπαζήσης.

Χριστοδουλάκης, Σ. (2015). *Κλιματική Αλλαγή και Μετεωρολογικά Φαινόμενα*. Αθήνα: Εκδόσεις Κριτική.

Βασιλείου, Μ. (2014). *Έξυπνα Συστήματα και Arduino: Από τη Θεωρία στην Πράξη*. Αθήνα: Κλειδάριθμος.

Μακρίδης, Π. (2013). *Arduino στην Εκπαίδευση και τις Εφαρμογές του*. Πάτρα: Εκδόσεις Νέα Γνώση.

Banzi, M. (2011). *GettingStartedwithArduino* (2η έκδοση). Sebastopol: O'Reilly Media.

Blum, J. (2013). *Exploring Arduino: Tools and Techniques for Engineering Wizardry*. Indianapolis: Wiley.

Monk, S. (2013). *Programming Arduino: Getting Started with Sketches* (2η έκδοση). New York: McGraw-Hill Education.

Ahrens, C. D. (2019). *Meteorology Today: An Introduction to Weather, Climate, and the Environment*. Boston: Cengage Learning.

World Meteorological Organization (WMO). (2017). *Guide to Meteorological Instruments and Methods of Observation* (8η έκδοση). Geneva: WMO.

Wallace, J. M., & Hobbs, P. V. (2006). *Atmospheric Science: An Introductory Survey* (2η έκδοση). San Diego: Academic Press.

Ramamurthy, P. (2014). *Smart Agriculture with IoT and Arduino*. Birmingham: Packt Publishing.

McRoberts, M. (2011). *Beginning Arduino* (2η έκδοση). New York: Apress.

Stull, R. B. (2016). *Practical Meteorology: An Algebra-based Survey of Atmospheric Science*. Vancouver: University of British Columbia.

Burroughs, W. J. (1999). *Weather Cycles: Real or Imaginary?* Cambridge: Cambridge University Press.

Διαδικτυακές πηγές

<https://docs.arduino.cc/hardware/uno-rev3/#features>

<https://grobotronics.com/proto-shield-for-arduino-uno-with-mini-breadboard.html?sl=en>

https://wiki.seeedstudio.com/GPRS_Shield_v1.0/

<https://grobotronics.com/gsm-gprs-shield-for-arduino-sim900.html?sl=en>

<https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>

<https://soilsensor.com/>

<https://einstronic.com/product/rain-drop-weather-sensor-module/>

<https://education.nationalgeographic.org/resource/anemometer/>

<https://www.faranux.com/product/single-channel-5v-relay-module-com41/>

<https://www.instructables.com/Soil-Moisture-Measurement-With-Arduino/>

<https://thingspeak.com/>

<https://www.youtube.com/watch?v=rAZwZxiOaKY>

<https://www.youtube.com/watch?v=aYQKJFromXs>

<https://www.youtube.com/watch?v=dJJAQxyryoQ>

<https://www.youtube.com/watch?v=EFvbS6XzTVo>

<https://www.youtube.com/watch?v=Ax2Oe5eBjow>

<https://www.youtube.com/watch?v=n-RkWRUw62g>

<https://github.com/RobTillaart/Arduino/tree/master/libraries/DHTlib>

Παράρτημα

Κώδικας Arduino

```
#include <SoftwareSerial.h>

#include <string.h>

#include <dht.h>

SoftwareSerial gprsSerial(7,8);

dht DHT;

#define dht11_pin 6

const int windsensorpin = A2;

const float vtagereference = 5.0;

const int analogmaxvalue = 1023;

const float maxwindspeed = 30;

int relayPin = 4;

const char* phoneNumber = "+3069xxxxxxxx";

void setup()

{

  Serial.begin(9600);

  gprsSerial.begin(9600);
```

```
pinMode(relayPin, OUTPUT);  
digitalWrite(relayPin, LOW);  
gprsSerial.println("AT+CMGF=1");  
gprsSerial.println("AT+CNMI=2,2,0,0,0");  
delay(1000);  
}
```

```
void loop()  
{  
  configureGPRS();  
  delay(1000);  
  sendData();  
  delay(1000);  
}
```

```
void configureGPRS()  
{  
  gprsSerial.println("AT");  
  delay(1000);  
  ctrlRelay();  
  delay(1000);  
  gprsSerial.println("AT+CPIN?");  
  delay(1000);  
}
```

```
ctrlRelay();  
delay(1000);  
gprsSerial.println("AT+CREG?");  
delay(1000);  
ctrlRelay();  
delay(1000);  
gprsSerial.println("AT+CGATT?");  
delay(1000);  
ctrlRelay();  
delay(1000);  
gprsSerial.println("AT+CIPSHUT");  
delay(1000);  
ctrlRelay();  
delay(1000);  
gprsSerial.println("AT+CIPSTATUS");  
delay(2000);  
ctrlRelay();  
delay(1000);  
gprsSerial.println("AT+CIPMUX=0");  
delay(2000);  
ctrlRelay();  
delay(1000);  
ShowSerialData();
```

```
gprsSerial.println("AT+CSSTT=\"internet.vodafone.gr\"");
delay(1000);
ctrlRelay();
delay(1000);
ShowSerialData();
gprsSerial.println("AT+CIICR");
delay(3000);
ctrlRelay();
delay(1000);
ShowSerialData();
gprsSerial.println("AT+CIFSR");
delay(2000);
ctrlRelay();
delay(1000);
ShowSerialData();
gprsSerial.println("AT+CIPSPRT=0");
delay(3000);
ctrlRelay();
delay(1000);
ShowSerialData();
gprsSerial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\");
delay(6000);
ctrlRelay();
```

```

delay(1000);

ShowSerialData();

}

void sendData()

{

    int chk = DHT.read11 (dht11_pin);

    int rr = analogRead(A0);

    int sr = analogRead(A1);

    int windsensorvalue = analogRead(windsensorpin);

    float voltage = windsensorvalue * (voltage / analogmaxvalue);

    float windspeed = (voltage / voltage / voltage) * maxwindspeed;

    float soilmoisturepercent = map(sr, 0 , 1023, 100, 0);

    float rainpercent = map(rr, 0, 1023, 100, 0);

    delay(100);

    Serial.print("Temperature = ");

    Serial.print(DHT.temperature);

    Serial.println(" °C");

    Serial.print("Humidity = ");

    Serial.print(DHT.humidity);

```

```
Serial.println(" %");
```

```
Serial.print("Soil Moisture = ");
```

```
Serial.print(soilmoisturepercent);
```

```
Serial.println(" %");
```

```
Serial.print("Rain Reading = ");
```

```
Serial.print(rainpercent);
```

```
Serial.println(" %");
```

```
Serial.print("Wind Speed =");
```

```
Serial.print(windspeed);
```

```
Serial.println(" m/s ");
```

```
gprsSerial.println("AT+CIPSEND");
```

```
delay(4000);
```

```
ShowSerialData();
```

```
String str="GET
```

```
https://api.thingspeak.com/update?api_key=EWGBSN427DRR6C3D&field1=" +
```

```
String(DHT.temperature)
```

```
+"&field2="+String(DHT.humidity)
```

```

        + "&field3=" + String(soilmoisturepercent)
        + "&field4=" + String(rainpercent)
        + "&field5=" + String(windspeed);

    Serial.println(str);

    gprsSerial.println(str);

    delay(4000);

    ShowSerialData();

    gprsSerial.println((char)26);

    delay(5000);

    gprsSerial.println();

    ShowSerialData();

    gprsSerial.println("AT+CIPSHUT");

    delay(100);

    ShowSerialData();
}

void ctrlRelay()
{
    int windsensorvalue = analogRead(windsensorpin);

    float voltage = windsensorvalue * (voltage / analogmaxvalue);

    float windspeed = (voltage / voltage) * maxwindspeed;
}

```

```
int rr = analogRead(A0);

float rainpercent = map(rr, 0, 1023, 100, 0);

delay(100);

Serial.println("Checking SMS");

delay(100);

while (gprsSerial.available())
{
    String response = gprsSerial.readString();

    if (response.indexOf("+CMT:") != -1)
    {
        Serial.println("Received");

        delay(100);

        Serial.println("Full response: ");

        Serial.println(response);

        delay(1000);

        int msgStartIndex = response.indexOf("\"", 3) + 3;

        String msgContent = response.substring(msgStartIndex);

        Serial.println("Message content: ");

        Serial.println(msgContent);

        delay(1000);
    }
}
```

```
if (msgContent.indexOf("Open") != -1)
{
  Serial.println("Relay open");
  digitalWrite(relayPin, HIGH);
  sendSMS(phoneNumber, "Relay opened.");
  delay(1000);
}
else if (msgContent.indexOf("Close") != -1)
{
  Serial.println("Relay closed");
  digitalWrite(relayPin, LOW);
  sendSMS(phoneNumber, "Relay closed.");
  delay(1000);
}
}
```

```
int relayState = digitalRead(relayPin);
if (relayState == HIGH)
{
  if (windspeed > 10)
  {
    delay(5000);
  }
}
```

```

digitalWrite(relayPin, LOW);

sendSMS(phoneNumber, "Relay closed due to high wind speed");

delay(1000);

}

if (rainpercent > 50)

{

    delay(5000);

    digitalWrite(relayPin, LOW);

    sendSMS(phoneNumber, "Relay closed due to rain");

    delay(1000);

}

}

}

void ShowSerialData()

{

    while (gprsSerial.available() != 0)

    {

        Serial.write(gprsSerial.read());

    }

}

void sendSMS(const char* number, const char* message)

```

```
{  
  gprsSerial.print("AT+CMGS=\");  
  gprsSerial.print(number);  
  gprsSerial.println("\");  
  delay(1000);  
  gprsSerial.print(message);  
  delay(3000);  
  gprsSerial.write(26);  
  delay(2000);  
  while (gprsSerial.available())  
  {  
    char c = gprsSerial.read();  
    Serial.print(c);  
  }  
}
```