



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΑΝΑΠΤΥΞΗ ΣΕΝΑΡΙΟΥ ΙΟΤ ΜΕ ARDUINO ΓΙΑ ΠΕΡΙΒΑΛΛΟΝΤΙΚΟ ΕΛΕΓΧΟ

ΔΗΜΗΤΡΗΣ ΜΑΥΡΟΠΟΛΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Νικόλαος Τζιρίτας
Επίκουρος Καθηγητής

Λαμία έτος



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΑΝΑΠΤΥΞΗ ΣΕΝΑΡΙΟΥ ΙΟΤ ΜΕ ARDUINO ΓΙΑ ΠΕΡΙΒΑΛΛΟΝΤΙΚΟ ΕΛΕΓΧΟ

ΔΗΜΗΤΡΗΣ ΜΑΥΡΟΠΟΔΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Νικόλαος Τζιρίτας
Επίκουρος Καθηγητής

Λαμία έτος



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

IOT SCEANARIO DEVELOPMENT WITH
ARDUINO FOR ENVIROMENTAL CONTROL

DIMITRIS MAVROPODIS

FINAL THESIS

ADVISOR

Nikolaos Tziritas
Assistant Professor

Lamia year

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 21.7.2022

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΠΕΡΙΛΗΨΗ

Η πτυχιακή έχει θέμα την ανάπτυξη ενός σεναρίου IOT με Arduino για περιβαλλοντικό έλεγχο. Στόχος ήταν η παρακολούθηση των συνθηκών στο περιβάλλον που έχει τοποθετηθεί το Arduino Uno και η αποστολή των δεδομένων τους σε μια διαδικτυακή πλατφόρμα IoT. Για την υλοποίηση χρησιμοποιήθηκε ο μικροελεγκτής Arduino Uno μαζί με ένα breadboard, έναν αισθητήρα θερμοκρασίας και υγρασίας DHT11 Module ,ένα ESP8266 wifi system on chip το οποίο παρέχει ασύρματη σύνδεση στην πλακέτα με το διαδίκτυο και κάποια καλώδια. Το ολοκληρωμένο κύκλωμα καταγράφει την θερμοκρασία και την υγρασία του περιβάλλοντος και ταυτόχρονα την ανεβάζει στην πλατφόρμα Thingspeak IOT όπου παρατηρούνται οι μετρήσεις σε πραγματικό χρόνο. Υπάρχει η δυνατότητα για απομακρυσμένη παρακολούθηση των μετρήσεων που έχουν γίνει από το κύκλωμα αφού όλες καταγράφονται στην πλατφόρμα Thingspeak IOT. Το Arduino προγραμματίστηκε στο περιβάλλον Arduino IDE με τη χρήση εντολών που υποστηρίζονται από διάφορες βιβλιοθήκες. Η ίδια υλοποίηση έγινε και με μια πλακέτα NodeMCU V3. Στην υλοποίηση με την πλακέτα NodeMCU έγινε χρήση μόνο του αισθητήρα θερμοκρασίας και υγρασίας DHT11 Module διότι η πλακέτα διαθέτει ενσωματωμένο το ESP8266 wifi system on chip. Η ενσωμάτωση του ESP8266 πάνω στο NodeMCU κάνει το κύκλωμα απλούστερο, μικρότερο σε όγκο και την υλοποίηση του κώδικα πιο εύκολη. Ο προγραμματισμός του NodeMCU γίνεται και αυτός στο περιβάλλον Arduino IDE.

ABSTRACT

The thesis is about the development of an IOT scenario with Arduino for environmental control. The goal was to monitor the conditions in the environment where the Arduino Uno is placed and send its data to an IoT web platform. For the implementation, the Arduino Uno microcontroller was used together with a breadboard, a temperature and humidity DHT11 Module, an ESP8266 wifi system on chip which provides wireless connection to the board with the internet and some cables. The integrated circuit records the temperature and humidity of the environment and simultaneously uploads it to the Thingspeak IOT platform where the measurements are observed in real time. There is the possibility for remote monitoring of the measurements made by the circuit since they are all recorded on the Thingspeak IOT platform. Arduino was programmed in the Arduino IDE environment using commands supported by various libraries. The same implementation was done with a NodeMCU V3 board. In the implementation with the NodeMCU board only the temperature and humidity sensor DHT11 Module was used because the board has an integrated ESP8266 wifi system on chip. The integration of the ESP8266 on the NodeMCU makes the circuit simpler, smaller in volume and the implementation of the code easier. The programming of the NodeMCU is also done in the Arduino IDE environment

Table of Contents

ΠΕΡΙΛΗΨΗ	I
ABSTRACT	III
<u>ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ</u>	<u>2</u>
1.1 ΤΙ ΕΙΝΑΙ ΤΟ ARDUINO UNO (ΠΕΡΙΓΡΑΦΗ).....	2
1.2 ΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ARDUINO	2
1.3 Η ΔΙΑΤΑΞΗ ΤΟΥ ARDUINO – ΔΙΑΓΡΑΜΜΑ	3
1.4 Η ΔΙΑΤΑΞΗ ΤΟΥ ARDUINO - ΤΡΟΦΟΔΟΣΙΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ARDUINO	3
1.5 Η ΔΙΑΤΑΞΗ ΤΟΥ ARDUINO – ANALOG IN (ΑΝΑΛΟΓΙΚΕΣ ΕΙΣΟΔΟΙ).....	5
1.6 Η ΔΙΑΤΑΞΗ ΤΟΥ ARDUINO – ΨΗΦΙΑΚΟΙ ΑΚΡΟΔΕΚΤΕΣ	6
1.6.Α ΑΝΑΛΥΣΗ ΤΗΣ ΨΗΦΙΑΚΗΣ ΛΕΙΤΟΥΡΓΙΑΣ ΤΩΝ ΑΚΡΟΔΕΚΤΩΝ ΤΟΥ ARDUINO UNO	6
1.6.Β ΑΝΑΛΥΣΗ ΤΗΣ PWM ΛΕΙΤΟΥΡΓΙΑΣ ΤΩΝ ΑΚΡΟΔΕΚΤΩΝ ΤΟΥ ARDUINO UNO	7
1.6.Γ ΑΝΑΛΥΣΗ ΤΩΝ ΠΡΩΤΟΚΟΛΛΩΝ ΕΠΙΚΟΙΝΩΝΙΑΣ ΤΟΥ ARDUINO UNO	7
1.7 Η ΔΙΑΤΑΞΗ ΤΟΥ ARDUINO- ICSP HEADER.....	9
<u>ΚΕΦΑΛΑΙΟ 2 ΤΟ ΛΟΓΙΣΜΙΚΟ IDE ΚΑΙ Ο ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΟΥ ARDUINO</u>	<u>9</u>
2.1 ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΟΥ ΚΩΔΙΚΑ ΤΟΥ ARDUINO ΚΑΙ ΒΑΣΙΚΑ ΕΡΓΑΛΕΙΑ.....	9
2.2 Η ΔΟΜΗ ΤΟΥ ΚΩΔΙΚΑ	12
2.3 Ο ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΟΥ ARDUINO	15
2.4 ΟΙ ΒΙΒΛΙΟΘΗΚΕΣ ΤΟΥ ΚΩΔΙΚΑ ARDUINO.....	15
2.5 Η ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ IDE	17
2.5.Α ΓΙΑ ΠΛΑΚΕΤΕΣ ARDUINO	17
2.5.Β ΓΙΑ NON-NATIVE ΠΛΑΚΕΤΕΣ.....	19
<u>ΚΕΦΑΛΑΙΟ 3 ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ THINGSPEAK IOT</u>	<u>20</u>
<u>ΚΕΦΑΛΑΙΟ 4 RELATED WORK.....</u>	<u>21</u>
4.1 Η ΕΝΣΩΜΑΤΩΣΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ IOT ΓΙΑ ΑΠΟΤΕΛΕΣΜΑΤΙΚΗ ΠΕΡΙΒΑΛΛΟΝΤΙΚΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗ	21
4.2 Η ΕΝΣΩΜΑΤΩΣΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ IOT ΓΙΑ ΤΗΝ ΑΠΟΤΕΛΕΣΜΑΤΙΚΗ ΠΕΡΙΒΑΛΛΟΝΤΙΚΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΣΤΗΝ ΑΓΡΟΤΙΚΗ ΚΑΙ ΚΤΗΝΟΤΡΟΦΙΚΗ ΠΑΡΑΓΩΓΗ ΚΑΙ Η ΧΡΗΣΙΜΟΤΗΤΑ ΤΗΣ	23
<u>ΚΕΦΑΛΑΙΟ 5 Η ΜΕΘΟΔΟΛΟΓΙΑ ΥΛΟΠΟΙΗΣΗΣ ΤΗΣ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΜΕ ΤΗ ΧΡΗΣΗ ΤΗΣ ΠΛΑΚΕΤΑΣ ARDUINO UNO</u>	<u>25</u>
5.1 ΤΑ ΕΞΑΡΤΗΜΑΤΑ ΠΟΥ ΑΠΑΙΤΟΥΝΤΑΙ ΓΙΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΚΥΚΛΩΜΑΤΟΣ ΓΙΑ ΤΗΝ ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ.....	25

5.2 Η ΑΝΑΛΥΣΗ ΚΑΙ Η ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΣΥΝΔΕΣΜΟΛΟΓΙΑΣ ΠΟΥ ΑΠΑΙΤΕΙΤΑΙ ΓΙΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΚΥΚΛΩΜΑΤΟΣ.....	31
5.3 Ο ΚΩΔΙΚΑΣ ΠΟΥ ΑΝΑΠΤΥΧΘΗΚΕ ΓΙΑ ΤΟ ARDUINO UNO	33

ΚΕΦΑΛΑΙΟ 6 Η ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΜΕ ΤΗ ΧΡΗΣΗ ΤΗΣ ΠΛΑΚΕΤΑΣ NODEMCU..... 36

6.1 ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΠΛΑΚΕΤΑΣ NODEMCU V3	36
6.2 ΣΥΓΚΡΙΣΗ ΤΟΥ NODEMCU ΜΕ ΤΟ ARDUINO	37
6.3 Η ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΚΥΚΛΩΜΑΤΟΣ ΤΗΣ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΜΕ ΤΟ NODEMCU	39
6.4 Ο ΚΩΔΙΚΑΣ ΠΟΥ ΑΝΑΠΤΥΧΘΗΚΕ ΓΙΑ ΤΟ NODEMCU	40

ΚΕΦΑΛΑΙΟ 7 ΣΥΜΠΕΡΑΣΜΑΤΑ..... 42

ΒΙΒΛΙΟΓΡΑΦΙΑ..... 44

ΔΙΑΔΙΚΤΥΑΚΕΣ ΠΗΓΕΣ..... 45

ΚΕΦΑΛΑΙΟ 1 Εισαγωγή

1.1 Τι είναι το Arduino uno (περιγραφή)

Το Arduino/Genuino Uno είναι ένας μικροεπεξεργαστής ο οποίος βασίζεται στον ATmega328P[3]. Έχει 14 ψηφιακές εισόδους/εξόδους (από τις οποίες 6 μπορούν να χρησιμοποιηθούν σαν έξοδοι PWM) ,6 αναλογικές εισόδους,16 MHz crystal quartz ,μια θύρα USB , μια θύρα για τροφοδοσία ρεύματος, εναν ICSP header και ένα κουμπί επαναφοράς. Το όνομα “UNO” προήλθε απο την ιταλική γλώσσα και σημαίνει ένα.Το συγκεκριμένο όνομα επιλέχτηκε για να σημειώσει την κυκλοφορία του λογισμικού Arduino (IDE) 1.0. Η πλακέτα UNO και η έκδοση του λογισμικού Arduino (IDE) 1.0 ήταν οι εκδόσεις αναφοράς του Arduino, οι οποίες διαδέχθηκαν από νεότερες εκδόσεις. Η υπολογιστική πλατφόρμα Arduino είναι μια πλατφόρμα ανοιχτού κώδικα[1]. Χρησιμοποιεί την δική της γλώσσα προγραμματισμού η οποία είναι μια απλοποιημένη εκδοχή της γλώσσας C++

1.2 Τα Χαρακτηριστικά του Arduino

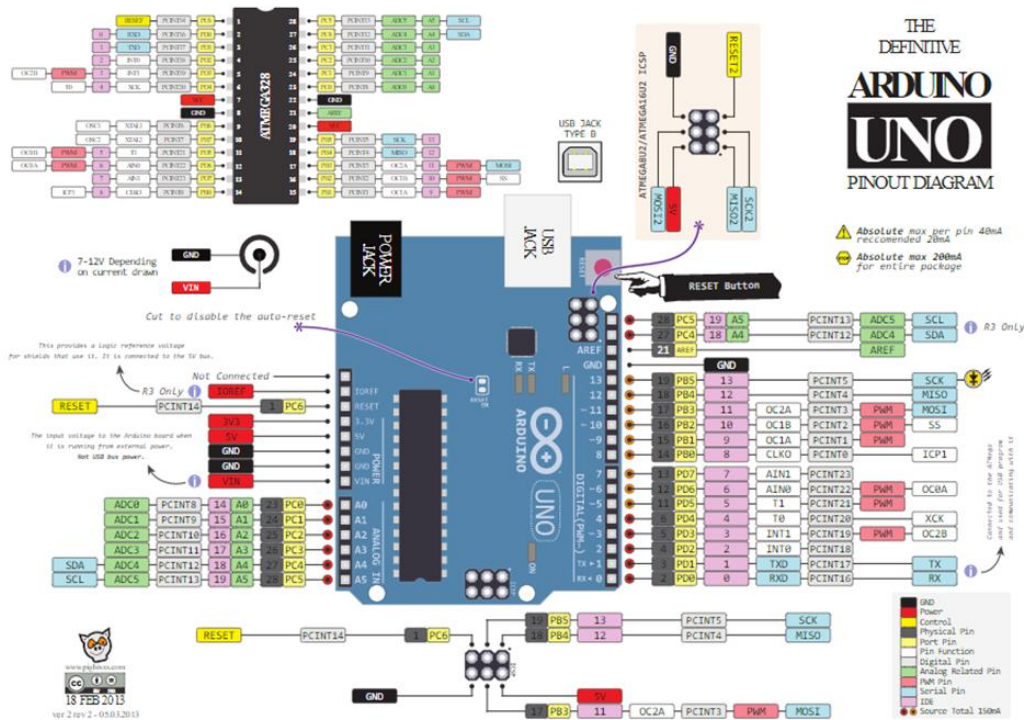
Το Arduino διαθέτει τον επεξεργαστή ATmega328[3]. Ο επεξεργαστής αυτός έχει ταχύτητα ρολογιού 16MHz. Η τάση λειτουργίας του κυμαίνεται μεταξύ των 5V(ελάχιστη) και των 20V(μέγιστη). Η χρήση του ωστόσο στα 20V μπορεί να προκαλέσει ζημιές και δεν προτείνεται. Η προτεινόμενη τάση λειτουργίας κυμαίνεται μεταξύ των 7V και 12V. Διαθέτει 6 αναλογικούς ακροδέκτες εισόδου και 14 ψηφιακούς ακροδέκτες που μπορούν να χρησιμοποιηθούν είτε σαν εισοδοι είτε σαν έξοδοι. Το συνεχές ρεύμα που διαθέτει ο κάθε ακροδέκτης εισόδου/εξόδου ενώ το συνεχές ρεύμα στον 3.3V ακροδέκτη είναι 50mA. Διαθέτει μία μνήμη SRAM 2KB , μια μνήμη EEPROM 1KB και μία μνήμη flash των 32KB εκ των οποίων 0.5KB χρησιμοποιούνται ως boot loader.

Microcontroller	ATmega328
Clock Speed	16MHz
Operating Voltage	5V
Maximum supply Voltage (not recommended)	20V
Supply Voltage (recommended)	7-12V
Analog Input Pins	6
Digital Input/Output Pins	14
DC Current per Input/Output Pin	40mA
DC Current in 3.3V Pin	50mA
SRAM	2KB
EEPROM	1KB
Flash Memory	32KB of which 0.5KB used by boot loader

Εικόνα 1.1: Χαρακτηριστικά του Arduino(<https://www.researchgate.net/>)

1.3 Η διάταξη του Arduino – Διάγραμμα

Στην παρακάτω εικόνα φαίνονται αναλυτικά οι ακροδέκτες του Arduino Uno.



Εικόνα 1.2: Η διάταξη των ακροδεκτών του Arduino (<https://commons.wikimedia.org/>)

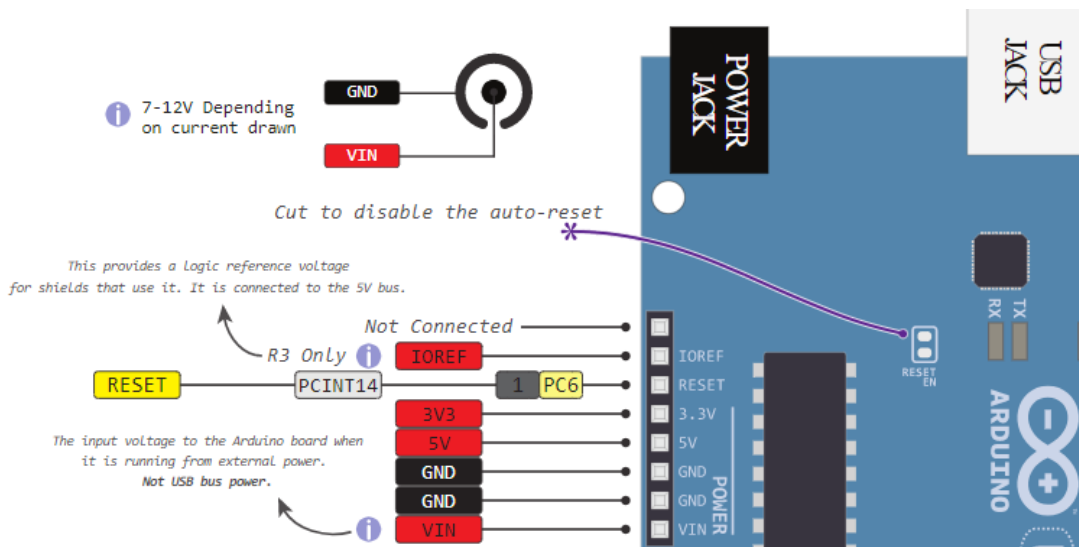
1.4 Η διάταξη του Arduino - Τροφοδοσία Χαρακτηριστικά του Arduino

Υπάρχουν 3 τρόποι για να τροφοδοτηθεί με ρεύμα το Arduino UNO[4]:

Υποδοχή φισ-Η υποδοχή φισ μπορεί να χρησιμοποιηθεί για να τροφοδοτηθεί με ρεύμα η πλακέτα Arduino. Η υποδοχή φισ συνήθως συνδέεται με μετασχηματιστή απο μια πρίζα. Η πλακέτα μπορεί να τροφοδοτηθεί απο 5 μέχρι 20 volt αλλά ο κατασκευαστής προτείνει η τάση να είναι ανάμεσα στα 7-12 volt. Πάνω από 12 volt υπάρχει πιθανότητα να υπερθερμανθούν οι ρυθμιστές τάσης και κάτω απο 7 volt ίσως η τάση να είναι ανεπαρκής.

Ακροδέκτης VIN- Αυτός ο ακροδέκτης χρησιμοποιείται για να τροφοδοτηθεί με ρεύμα το Arduino UNO από μια εξωτερική πηγή ρεύματος. Η τάση πρέπει να βρίσκεται στο εύρος που αναφέρθηκε παραπάνω.

Καλώδιο USB- όταν συνδέεται σε υπολογιστή παρέχει 5 volt στα 500mA.



Εικόνα 1.3: Οι ακροδέκτες του Arduino και του Power Jack (<https://commons.wikimedia.org/>)

Υπάρχει δίοδος προστασίας της πολικότητας ανάμεσα στο θετικό της υποδοχής φικ και τον ακροδέκτη VIN η οποία έχει ονομαστική ένταση ρεύματος 1 Ampere.

Η ισχύς της τροφοδοσίας που χρησιμοποιείται καθορίζει την ισχύ που είναι διαθέσιμη για το κύκλωμα. Για παράδειγμα τροφοδοτώντας το κύκλωμα χρησιμοποιώντας την θύρα USB το ρεύμα περιορίζεται στα 500mA. Πρέπει να ληφθεί υπόψιν πως αυτό χρησιμοποιείται για να τροφοδοτηθεί η MCU, τα περιφερειακά της ,τους ρυθμιστές που βρίσκονται πάνω στην πλακέτα και τα εξαρτήματα που είναι συνδεδεμένα πάνω της. Όταν τροφοδοτείται το κύκλωμα από την υποδοχή φικ ή τον ακροδέκτη VIN η μέγιστη διαθέσιμη χωρητικότητα καθορίζεται απο τους ρυθμιστές των 5 και 3,3 volt που βρίσκονται πάνω στην πλακέτα Arduino.

5v and 3v3- Παρέχουν ρυθμισμένη 5 και 3,3 volt τάση για την τροφοδοσία εξωτερικών εξαρτημάτων σύμφωνα με τις προδιαγραφές του κατασκευαστή.

GND- Στους ακροδέκτες του Arduino UNO υπάρχουν 5 GND ακροδέκτες , οι οποίοι είναι όλοι διασυνδεδεμένοι.

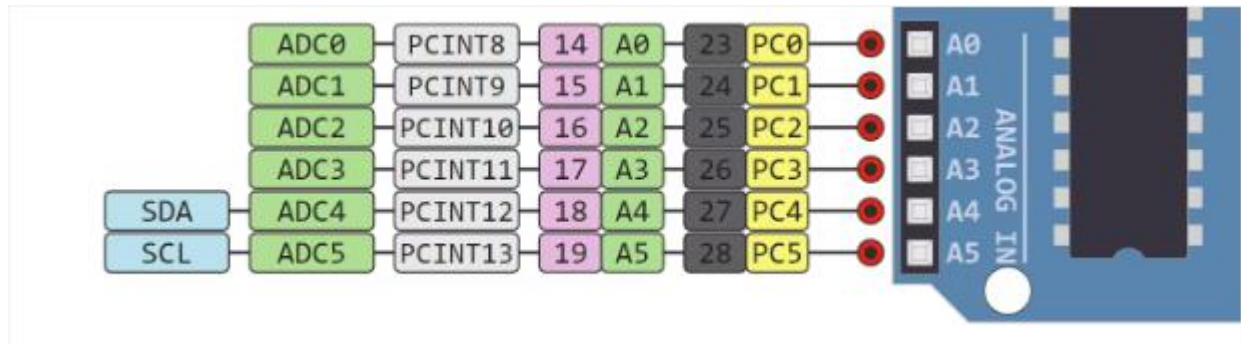
Οι ακροδέκτες GND όλοι χρησιμοποιούνται για να κλείσουν το ηλεκτρικό κύκλωμα και για να παρέχουν ένα κοινό λογικό επίπεδο αναφοράς μέσα στο κύκλωμα. Πάντα πρέπει να είμαστε σίγουροι πως όλοι οι ακροδέκτες GND (του Arduino, των περιφερειακών και των εξαρτημάτων) είναι συνδεδεμένοι ο ένας με τον άλλο και έχουν κοινή γείωση.

RESET-επαναφέρει το Arduino

IOREF-Αυτός ο ακροδέκτης είναι η αναφορά εισόδου/εξόδου. Παρέχει την αναφορά της τάσης με την οποία λειτουργεί ο μικροεπεξεργαστής.

1.5 Η διάταξη του Arduino – Analog IN (Αναλογικές εισοδοι)

Το Arduino Uno έχει 6 αναλογικούς ακροδέκτες, οι οποίοι χρησιμοποιούν ένα ADC (μετατροπέα από Αναλογικό σε ψηφιακό). Αυτοί οι ακροδέκτες χρησιμοποιούνται σαν αναλογικές εισοδοι αλλά μπορούν να λειτουργήσουν και σαν ψηφιακοί εισοδοι ή ψηφιακοί έξοδοι.



Εικόνα 1.4: Οι 6 αναλογικές εισοδοι του Arduino(<https://commons.wikimedia.org/>)

Η μετατροπή από αναλογικό σήμα σε ψηφιακό σήμα(ADC)

ADC σημαίνει αναλογικός σε ψηφιακός μετατροπέας. Ο ADC είναι ένα ηλεκτρονικό κύκλωμα που χρησιμοποιείται για να μετατρέψει αναλογικά σήματα σε ψηφιακά. Αυτή η ψηφιακή παρουσίαση των αναλογικών σημάτων επιτρέπει στον επεξεργαστή (ο οποίος είναι μια ψηφιακή συσκευή) να μετρήσει το αναλογικό σήμα και να το χρησιμοποιήσει μέσα στην λειτουργία του.

Οι ακροδέκτες του Arduino A0-A5 είναι ικανοί να διαβάσουν αναλογικές τάσεις. Στο Arduino ο ADC έχει ανάλυση 10 bit, που σημαίνει ότι μπορεί να αναπαραστήσει αναλογικό σήμα σε 1024 ψηφιακά επίπεδα. Ο ADC μετατρέπει τη τάση σε bits τα οποία ο μικροεπεξεργαστής μπορεί να καταλάβει.[2]

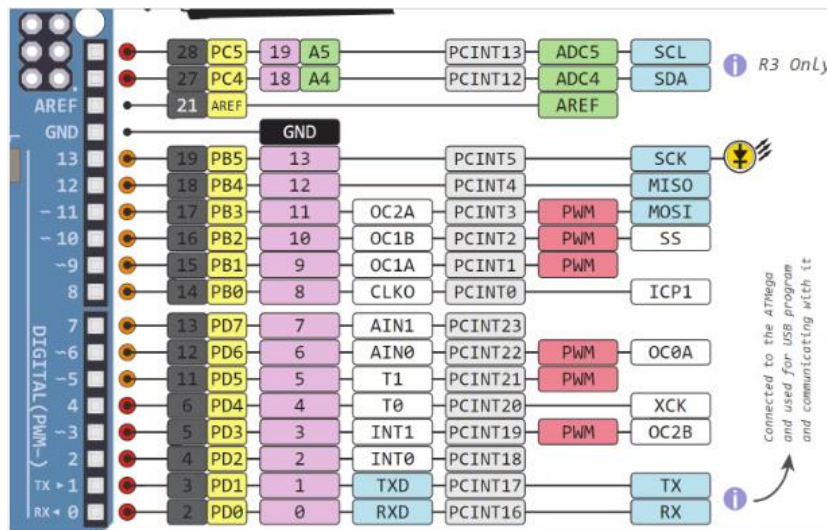
Ένα κοινό παράδειγμα ενός ADC είναι το Voice over IP (VoIP). Κάθε έξυπνο κινητό έχει ένα μικρόφωνο που μετατρέπει τα ηχητικά κύματα (φωνή) σε αναλογική τάση. Αυτή περνάει μέσα από τον ADC της συσκευής, μετατρέπεται σε ψηφιακά δεδομένα που μεταδίδονται στην μεριά λήψης του διαδικτύου.

1.6 Η διάταξη του Arduino – Ψηφιακοί ακροδέκτες

Οι ακροδέκτες 0-13 του Arduino Uno χρησιμοποιούνται σαν ψηφιακές εισοδοί / έξοδοι. Ο ακροδέκτης 13 του Arduino Uno είναι συνδεδεμένος στο ενσωματωμένο LED του Arduino.

Στο Arduino Uno οι ακροδέκτες 3,5,6,9,10,11 έχουν δυνατότητα PWM.

- Κάθε ακροδέκτης μπορεί να παρέχει μέγιστο ρεύμα 40 mA. Αλλά το συνιστώμενο ρεύμα είναι 20mA.
- Η μεγαλύτερη απόλυτη τιμή ρεύματος που παρέχεται για όλους τους ακροδέκτες μαζί είναι 200mA



Εικόνα 1.5: Οι ψηφιακοί ακροδέκτες του Arduino (<https://commons.wikimedia.org/>)

1.6.α Ανάλυση της ψηφιακής λειτουργίας των ακροδεκτών του Arduino Uno

Το ψηφιακό είναι ένας τρόπος αναπαράστασης της τάσης σε 1 bit που η τιμή του είναι είτε 1 είτε 0. Οι ψηφιακοί ακροδέκτες στο Arduino είναι ακροδέκτες σχεδιασμένοι να μπορούν να διαμορφωθούν σαν εισοδοί ή έξοδοι ανάλογα με τις ανάγκες του χρήστη. Οι ψηφιακοί εισοδοί είναι είτε αναμμένοι (ON) είτε είναι σβηστοί (OFF). Όταν είναι στο ON είναι σε μία κατάσταση υψηλής τάσης των 5V και όταν είναι στο OFF είναι σε μία κατάσταση χαμηλής τάσης των 0V.

Στο Arduino, όταν οι ακροδέκτες είναι διαμορφωμένοι σαν **έξοδοι** είναι ρυθμισμένοι στα 0 ή στα 5 Volt.

Στο Arduino, όταν οι ακροδέκτες είναι διαμορφωμένοι σαν **είσοδοι**, η τάση τους παρέχεται από μία εξωτερική συσκευή. Η τάση μπορεί να μεταβάλλεται ανάμεσα στο 0-5 Volt η οποία μετατρέπεται σε ψηφιακή αναπαράσταση (0 ή 1). Για να το καθορίσουμε αυτό υπάρχουν 2 όρια:

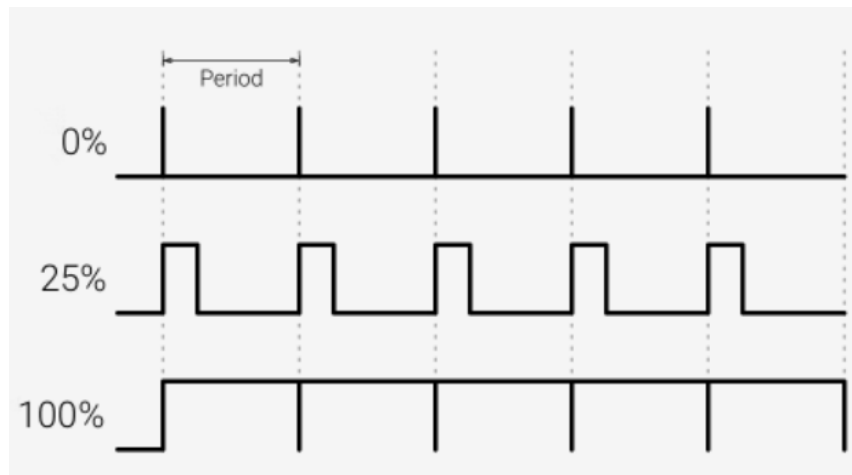
- Κάτω από 0,8V – θεωρείται 0
- Πάνω από 2V – θεωρείται 1

Όταν συνδέεται ένα εξάρτημα σε έναν ψηφιακό ακροδέκτη, πρέπει να είναι βέβαιο πως τα λογικά επίπεδα ταιριάζουν. Αν η τάση είναι ανάμεσα στα όρια η τιμή που επιστρέφεται θα είναι απροσδιόριστη.

1.6.β Ανάλυση της PWM λειτουργίας των ακροδεκτών του Arduino Uno

Γενικά το Pulse Width Modulation (PWM) είναι μία τεχνική διαμόρφωσης που χρησιμοποιείται για να κωδικοποιήσει ένα μήνυμα σε ένα σήμα παλμού. Μια PWM εξαρτάται από δύο παράγοντες κλειδιά: την **συχνότητα** και τον **κύκλο λειτουργίας**. Η συχνότητα του PWM καθορίζει το πόσο χρειάζεται για να ολοκληρωθεί ένας κύκλος(περίοδος) και πόσο γρήγορα το σήμα διακυμαίνεται από την υψηλή τάση(HIGH) στην χαμηλή τάση (LOW). Ο κύκλος λειτουργίας καθορίζει για πόση ώρα θα μείνει το σήμα στο HIGH μέσα σε μία περίοδο. Ο κύκλος λειτουργίας παρουσιάζεται ως ποσοστό επι τοις εκατό.

Στο Arduino οι ακροδέκτες που έχουν ενεργοποιημένη την PWM παράγουν μια σταθερή συχνότητα των 500Hz , ενώ ο κύκλος λειτουργίας αλλάζει ανάλογα με τις παραμέτρους που έχει ορίσει ο χρήστης. Παρατίθεται το ανάλογο σχήμα παρακάτω



Εικόνα 1.6: Γραφική παράσταση του PWM(<https://www.circuito.io/>)

Τα PWM σήματα χρησιμοποιούνται για τον έλεγχο της ταχύτητας κινητήρων DC, για τη ρύθμιση της έντασης της φωτεινότητας των LED και για άλλα.

1.6.γ Ανάλυση των πρωτοκόλλων επικοινωνίας του Arduino Uno

Σειριακό(TTL)- Οι ψηφιακοί ακροδέκτες 0 και 1 είναι οι σειριακοί ακροδέκτες του Arduino Uno.Χρησιμοποιούνται από την ενσωματωμένη μονάδα USB της πλακέτας.

Τι είναι η σειριακή επικοινωνία;

Η σειριακή επικοινωνία χρησιμοποιείται για να ανταλλάξει δεδομένα μεταξύ της πλακέτας Arduino και άλλων σειριακών συσκευών όπως υπολογιστών, οθονών, αισθητήρων κλπ. Κάθε πλακέτα Arduino έχει τουλάχιστον μία σειριακή θύρα. Η σειριακή επικοινωνία προκύπτει στους ακροδέκτες 0(RX) και 1(TX) καθώς επίσης και στην θύρα USB. Το Arduino υποστηρίζει την σειριακή επικοινωνία μέσω των ψηφιακών ακροδεκτών μέσω της βιβλιοθήκης SoftwareSerial. Αυτό επιτρέπει στο χρήστη να συνδέσει πολλαπλές σειριακές συσκευές και έτσι να παραμένει η κεντρική σειριακή θύρα ελεύθερη για τη USB.

Software serial και hardware serial – Οι περισσότεροι μικροελεγκτές έχουν υλικό σχεδιασμένο να επικοινωνεί και με άλλες σειριακές συσκευές. Οι software serial θύρες χρησιμοποιούν ένα σύστημα ακροδέκτη-αλλαγής διακοπής για να επικοινωνούν. Υπάρχει ενσωματωμένη βιβλιοθήκη για Software serial επικοινωνία. Η software serial χρησιμοποιείται από τον επεξεργαστή για να προσομοιώσει παραπάνω σειριακές θύρες. Το μοναδικό μειονέκτημα με την Software serial είναι ότι απαιτεί περισσότερη επεξεργασία και δεν μπορεί να υποστηρίξει τις ίδιες υψηλές ταχύτητες όπως η hardware serial.

SPI - Οι SS/SCK/MISO/MOSI ακροδέκτες είναι αποκλειστικά για SPI επικοινωνία.

Βρίσκονται στους ψηφιακούς ακροδέκτες 10-13 του Arduino Uno και στους ICSP headers.

Τι είναι η SPI;

Η σειριακή περιφερειακή διασύνδεση(SPI) είναι ένα πρωτόκολλο σειριακών δεδομένων που χρησιμοποιείται από μικροελεγκτές για να επικοινωνήσει με μία ή περισσότερες συσκευές σε μίας σύνδεσης bus. Η SPI μπορεί να χρησιμοποιηθεί για να συνδέσει δύο μικροελεγκτές. Στην SPI σύνδεση bus, υπάρχει πάντα μία συσκευή που ορίζεται σαν Κύρια συσκευή και οι υπόλοιπες σαν δευτερεύουσες. Στις περισσότερες περιπτώσεις, ο μικροελεγκτής είναι η κύρια συσκευή. Ο SS (Slave Select) ακροδέκτης καθορίζει ποια συσκευή είναι η κύρια και με ποια επικοινωνεί την συγκεκριμένη στιγμή.

Οι SPI ενσωματωμένες συσκευές πάντα έχουν τους ακόλουθους ακροδέκτες:

MISO (Master In Slave Out) - Μια γραμμή για να στέλνει δεδομένα στην Κύρια συσκευή.

MOSI (Master Out Slave In) - Η κύρια γραμμή για αποστολή δεδομένων σε περιφερειακές συσκευές.

SCK (Serial Clock) – Ένα σήμα ρολογιού που δημιουργείται από την Κύρια συσκευή για να συγχρονίσει την μετάδοση των δεδομένων.

I2C – Οι SCL/SDA ακροδέκτες χρησιμοποιούνται αποκλειστικά για I2C επικοινωνία. Πάνω στο Arduino Uno βρίσκονται στους αναλογικούς ακροδέκτες A4 και A5.

Τι είναι το I2C;

Το I2C είναι ένα πρωτόκολλο επικοινωνίας που συχνά αναφέρεται σαν το «I2C bus». Το I2C πρωτόκολλο σχεδιάστηκε για να επιτρέψει την επικοινωνία μεταξύ εξαρτημάτων πάνω στο κύκλωμα μιας πλακέτας. Με το I2C υπάρχουν δυο καλώδια που αναγράφονται σαν SCL και SDA.

Η SCL είναι η γραμμή του ρολογιού η οποία σχεδιάστηκε για να συγχρονίζει τις μεταφορές δεδομένων.

Η SDA είναι η γραμμή που χρησιμοποιείται για τη μετάδοση δεδομένων.

Κάθε συσκευή στο I2C bus έχει μία μοναδική διεύθυνση, πάνω από 255 συσκευές μπορούν να συνδεθούν στην ίδια σύνδεση bus

Aref – Η τάση αναφοράς για τις αναλογικές εισόδους

Interrupt - INT0 και INT1. Το Arduino Uno έχει δύο εξωτερικούς ακροδέκτες διακοπής.

External Interrupt – Μια εξωτερική διακοπή είναι μια διακοπή συστήματος που συμβαίνει όταν υπάρχουν εξωτερικές παρεμβολές.

Παρεμβολές μπορούν να προκύψουν από τον χρήστη ή από άλλες συσκευές υλικού στο δίκτυο. Συχνές χρήσεις για αυτές τις διακοπές στο Arduino είναι το διάβασμα μιας συχνότητας ενός τετράγωνου κύματος που έχει δημιουργηθεί από κωδικοποιητές ή για το ξύπνημα του επεξεργαστή από ένα εξωτερικό συμβάν.

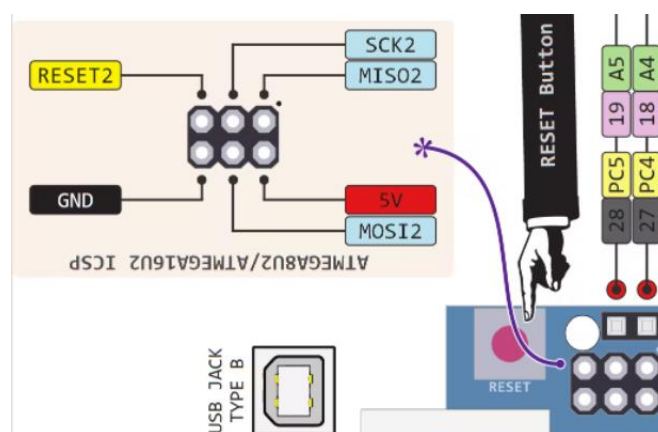
Το Arduino έχει δύο μορφές διακοπής:

- Την Εξωτερική
- Την αλλαγή ακροδέκτη

Υπάρχουν δυο εξωτερικοί ακροδέκτες διακοπής στον ATmega168/328 που ονομάζονται INT0 και INT1. Και οι δυο INT0 και INT1 είναι συνδεδεμένοι στους ακροδέκτες 2 και 3. Σε αντίθεση η διακοπή αλλαγής ακροδέκτη μπορεί να ενεργοποιηθεί σε οποιονδήποτε από τους ακροδέκτες.

1.7 Η διάταξη του Arduino- ICSP Header

ICSP σημαίνει εσωτερικός προγραμματισμός συστήματος. Το όνομα προέρχεται από τις λέξεις In-System Programming headers (ISP). Κατασκευαστές όπως η Atmel που εργάζονται με την Arduino έχουν εξελίξει τον δικό τους εσωτερικό σειριακό προγραμματισμό. Αυτοί οι ακροδέκτες επιτρέπουν στο χρήστη να προγραμματίσει το υλικολογισμικό της πλακέτας Arduino. Υπάρχουν 6 ICSP ακροδέκτες πάνω στην πλακέτα Arduino που μπορούν να συνδεθούν με μία συσκευή προγραμματισμού μέσω ενός καλωδίου προγραμματισμού.



Εικόνα 1.7: Απεικόνιση του ICSP Header (<https://commons.wikimedia.org/>)

ΚΕΦΑΛΑΙΟ 2 Το Λογισμικό IDE και ο προγραμματισμός του Arduino

2.1 Το περιβάλλον του κωδικα του Arduino και βασικα εργαλεία

Τι γλώσσα είναι το Arduino;

Ο κώδικας Arduino είναι γραμμένος σε C++ με την προσθήκη ειδικών μεθόδων και συναρτήσεων, τις οποίες θα αναφερθούν στη συνέχεια. Η C++ είναι μια γλώσσα προγραμματισμού που μπορεί να διαβαστεί από τον άνθρωπο. Όταν ο χρήστης δημιουργεί ένα «sketch» (το όνομα που δίνεται στα αρχεία κώδικα Arduino), υποβάλλεται σε επεξεργασία και μεταγλωττίζεται στη γλώσσα μηχανής.

Arduino IDE

Το Arduino Integrated Development Environment (IDE) [5] είναι το κύριο πρόγραμμα επεξεργασίας κειμένου που χρησιμοποιείται για τον προγραμματισμό του Arduino. Εκεί ο χρήστης πληκτρολογεί τον κώδικα πριν τον φορτώσει στην πλακέτα που θέλει να προγραμματίσει. Ο κώδικας Arduino αναφέρεται ως **sketches**.

```
arduinotest11
#include <DHT.h>
#include <SoftwareSerial.h>

String agAdi = "KarrasHome";           // Edw grafoume to onoma tou diktyou mas
String agSifresi = "2695031430a";     // Edw grafoume ton kwdiko tou diktyou mas

int rxPin = 10;                        // ESP8266 RX pin
int txPin = 11;                        // ESP8266 TX pin
int dht11Pin = 2;

String ip = "184.106.153.149";        // Thingspeak ip adress
float temper, humi;

//dht11 DHT11;
#define DHTPIN 2
SoftwareSerial esp(rxPin, txPin);     // Edw ksekinanei to setuptw twv pins ths seiriakhs epikoinwnias
DHT dht(DHTPIN, DHT11);
void setup() {

  Serial.begin(9600);                 // Edw ksekinanei h epikoinwnia mas me th serial port
  Serial.println("Started");
  esp.begin(115200);                 // Ksekiname th seiriakh epikoinwnia me to ESP8266
  esp.println("AT");                 // Elegxoume to module me thn entolh AT
  Serial.println("AT sent ");
  while(!esp.find("OK")){           // Perimenoume mexri to module na einai etoimo
    esp.println("AT");
    Serial.println("ESP8266 Not Find.");
  }
```

Εικόνα 2.1: Παράδειγμα του κώδικα Arduino

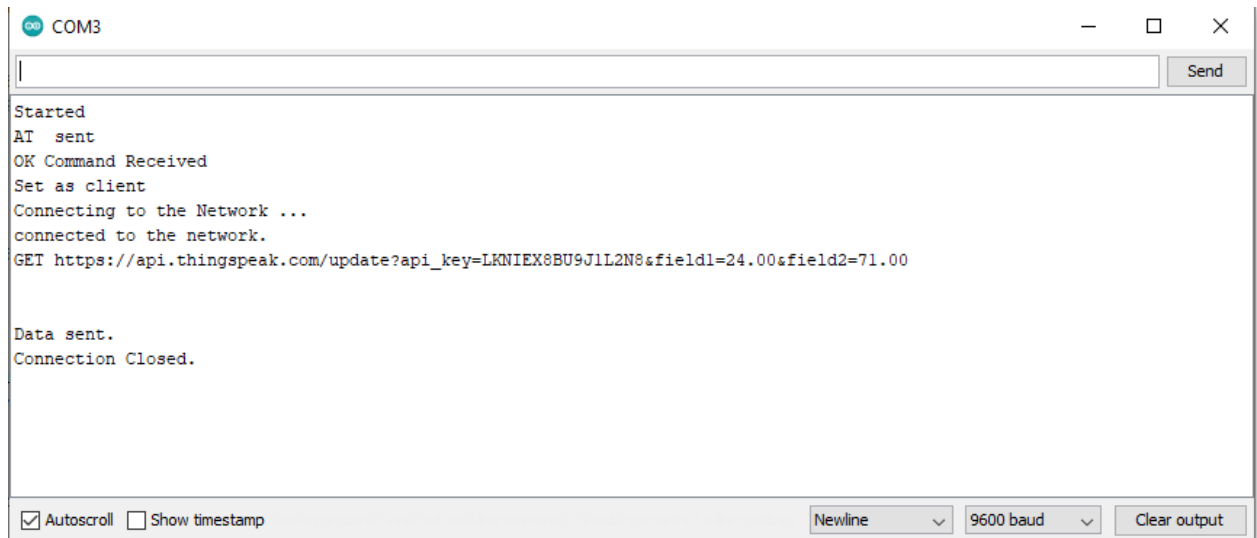
Το IDE έχει μινιμαλιστικό σχεδιασμό. Υπάρχουν μόνο 5 επικεφαλίδες στη γραμμή μενού, καθώς και μια σειρά από κουμπιά από κάτω που επιτρέπουν στο χρήστη να επαληθεύσει και να ανεβάσει τα σκίτσα του. Ουσιαστικά, το IDE μεταφράζει και μεταγλωττίζει τα sketches σε κώδικα που μπορεί να κατανοήσει το Arduino. Μόλις μεταγλωττιστεί ο κώδικας Arduino, στη συνέχεια μεταφορτώνεται στη μνήμη της πλακέτας.

Ο χρήστης για να μεταγλωτίσει το sketch του χρειάζεται να πατήσει μόνο ένα κουμπί όπως φαίνεται και παρακάτω.

Εάν υπάρχουν σφάλματα στον κώδικα Arduino, θα εμφανιστεί ένα προειδοποιητικό μήνυμα που θα ζητά από τον χρήστη να κάνει αλλαγές. Οι περισσότεροι νέοι χρήστες αντιμετωπίζουν συχνά δυσκολίες με τη μεταγλώττιση λόγω του αυστηρού συντακτικού του Arduino. Εάν γίνει λάθος στο συντακτικό όταν χρησιμοποιείται το Arduino, ο κώδικας δεν θα μεταγλωττιστεί και θα εμφανιστεί ένα μήνυμα σφάλματος.

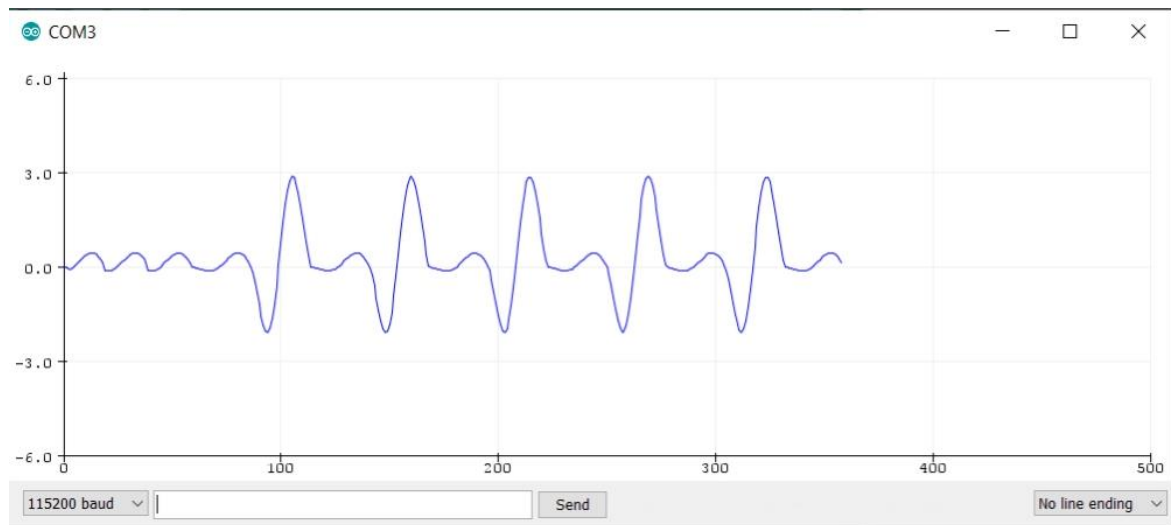
Serial Monitor και Serial Plotter

Το serial monitor του Arduino μπορεί να ανοίξει κάνοντας κλικ στο εικονίδιο του μεγεθυντικού φακού στην πάνω δεξιά πλευρά του IDE ή κάτω από τα εργαλεία. Η σειριακή οθόνη χρησιμοποιείται κυρίως για την αλληλεπίδραση με την πλακέτα Arduino χρησιμοποιώντας τον υπολογιστή και είναι ένα εξαιρετικό εργαλείο για παρακολούθηση και τον εντοπισμό σφαλμάτων σε πραγματικό χρόνο. Για να γίνει χρήση της οθόνης, πρέπει να χρησιμοποιηθεί η κλάση Serial.



Εικόνα 2.2: Εμφάνιση των δεδομένων της εκτέλεσης κώδικα στο serial monitor του IDE

Το Arduino serial plotter είναι ένα άλλο στοιχείο του Arduino IDE, το οποίο επιτρέπει τη δημιουργία ενός γραφήματος σε πραγματικό χρόνο των σειριακών δεδομένων. Το serial plotter διευκολύνει πολύ στην ανάλυση των δεδομένων μέσω μιας οπτικής οθόνης. Υπάρχει η δυνατότητα δημιουργίας γραφημάτων, γραφημάτων αρνητικών τιμών και η διεξαγωγή μιας ανάλυσης κυματομορφής.



Εικόνα 2.3: Κυματομορφή στο serial plotter του IDE

Αποσφαλμάτωση του κώδικα Arduino και Hardware

Σε αντίθεση με άλλες πλατφόρμες προγραμματισμού, το Arduino δεν διαθέτει ενσωματωμένο πρόγραμμα εντοπισμού σφαλμάτων. Οι χρήστες μπορούν είτε να χρησιμοποιήσουν λογισμικό τρίτων κατασκευαστών είτε να χρησιμοποιήσουν το serial monitor για να τυπώσουν τις ενεργές διαδικασίες του Arduino για την παρακολούθηση και τον εντοπισμό σφαλμάτων.

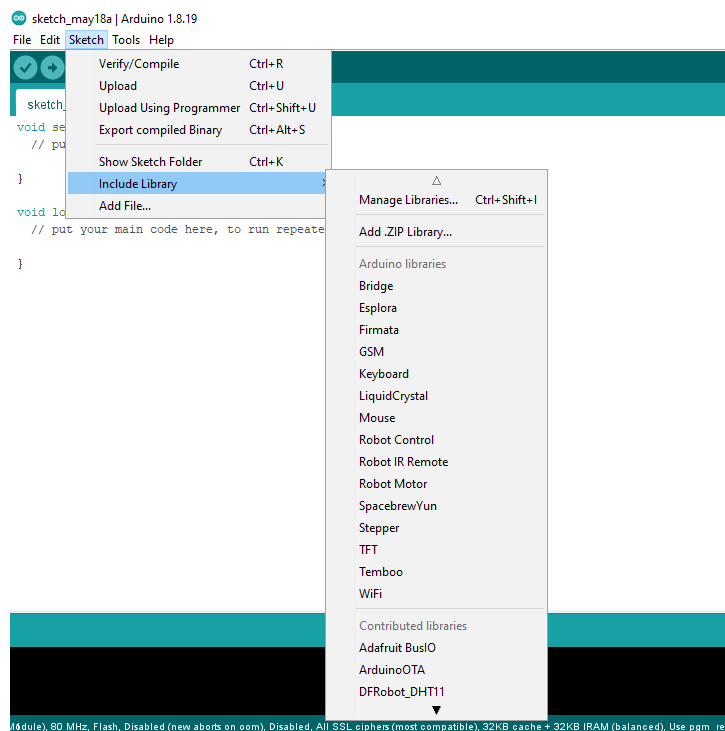
Χρησιμοποιώντας την κλάση Serial, ο χρήστης μπορεί να τυπώνει στη σειριακή οθόνη σχόλια αποσφαλμάτωσης και τις τιμές των μεταβλητών. Στα περισσότερα μοντέλα Arduino, αυτό μπορεί να επιτευχθεί με τη χρήση των σειριακών ακροδεκτών 0 και 1 που είναι συνδεδεμένοι στη θύρα USB.

2.2 Η δομή του κώδικα

Βιβλιοθήκες

Στο Arduino, όπως και σε άλλες κορυφαίες πλατφόρμες προγραμματισμού, υπάρχουν ενσωματωμένες βιβλιοθήκες που παρέχουν βασικές λειτουργίες. Επιπλέον, είναι δυνατή η εισαγωγή άλλων βιβλιοθηκών έτσι επιτυγχάνεται η επέκταση των δυνατοτήτων και των χαρακτηριστικών της πλακέτας Arduino. Αυτές οι βιβλιοθήκες χωρίζονται χονδρικά σε βιβλιοθήκες που αλληλεπιδρούν με ένα συγκεκριμένο εξάρτημα και σε αυτές που υλοποιούν νέες λειτουργίες.

Για την εισαγωγή μιας νέας βιβλιοθήκης, ο χρήστης πρέπει να μεταβεί στο Sketch > Include Library



Εικόνα 2.4: Εισαγωγή βιβλιοθήκης στο IDE του Arduino

Επιπλέον, στο επάνω μέρος του αρχείου .ino, πρέπει να χρησιμοποιηθεί το «#include» για να συμπεριληφθούν εξωτερικές βιβλιοθήκες. Επίσης μπορούν να χρησιμοποιηθούν τροποποιημένες βιβλιοθήκες για χρήση σε μεμονωμένα σκίτσα.

Ορισμός Ακροδεκτών

Ο χρήστης για να χρησιμοποιήσει τους ακροδέκτες του Arduino, πρέπει να ορίσει ποιος ακροδέκτης χρησιμοποιείται και ποια είναι η λειτουργικότητά του. Ένας πρακτικός τρόπος για να οριστούν οι ακροδέκτες που χρησιμοποιούνται είναι ο παρακάτω:

```
#define pinName pinNumber
```

Η λειτουργικότητα είναι είτε είσοδος είτε έξοδος και ορίζεται με τη χρήση της μεθόδου `pinMode ()` στο κομμάτι του `setup`.

Δηλώσεις:

Μεταβλητές

Κάθε φορά που χρησιμοποιείται το Arduino, πρέπει να δηλωθούν οι καθολικές (global) μεταβλητές και τα στιγμιότυπα (instances) που θα χρησιμοποιηθούν αργότερα. Με λίγα λόγια, μια μεταβλητή κάνει δυνατή την ονομασία και την αποθήκευση μιας τιμής που θα χρησιμοποιηθεί στο μέλλον. Ένα παράδειγμα είναι η αποθήκευση δεδομένων που αποκτήθηκαν από έναν αισθητήρα για να χρησιμοποιηθούν αργότερα. Για να δηλωθεί μια μεταβλητή απλά ορίζεται ο τύπος, το όνομα και η αρχική της τιμή. Αξίζει να σημειωθεί ότι η δήλωση καθολικών μεταβλητών δεν είναι απόλυτη αναγκαιότητα. Ωστόσο, συνιστάται η δήλωση των μεταβλητών για την διευκόλυνση στη χρήση των τιμών παρακάτω.

Instances

Στον προγραμματισμό λογισμικού, μια **κλάση** είναι μια συλλογή συναρτήσεων και μεταβλητών που διατηρούνται μαζί σε ένα μέρος. Κάθε κλάση έχει μια ειδική συνάρτηση γνωστή ως **κατασκευαστής**, η οποία χρησιμοποιείται για τη δημιουργία ενός **στιγμιότυπου** της κλάσης. Για να χρησιμοποιηθούν οι συναρτήσεις της κλάσης, πρέπει να δηλωθεί ένα στιγμιότυπο για αυτήν.

Setup()

Κάθε Arduino sketch πρέπει να έχει μια συνάρτηση `setup`[\[6\]](#). Αυτή η συνάρτηση καθορίζει την αρχική κατάσταση του Arduino κατά την εκκίνηση και εκτελείται μόνο μία φορά.

Εδώ θα ορίσει ο χρήστης τα ακόλουθα:

1. Την λειτουργικότητα των ακροδεκτών με τη χρήση της συνάρτησης `pinMode`
2. Την αρχική κατάσταση των ακροδεκτών
3. Την αρχικοποίηση των μεταβλητών
4. Την αρχικοποίηση των μεταβλητών
5. Την λογική του κώδικα

Loop()

Η συνάρτηση `loop` [\[6\]](#) είναι επίσης απαραίτητη για κάθε sketch του Arduino και εκτελείται μόλις ολοκληρωθεί η `setup()`. Είναι η κύρια συνάρτηση και όπως υποδηλώνει και το όνομά της, εκτελείται σε βρόχο ξανά και ξανά. Ο βρόχος περιγράφει την κύρια λογική του κυκλώματος του χρήστη.

Για παράδειγμα:

```
void setup() {  
  
  Serial.begin(9600); // We are starting our communication  
  Serial.println("Started");  
  esp.begin(115200);  
  esp.println("AT");  
  Serial.println("AT sent ");  
  while(!esp.find("OK")){  
    esp.println("AT");  
    Serial.println("ESP8266 Not Find.");  
  }  
  Serial.println("OK Command Received");  
  esp.println("AT+CWMODE=1");  
  while(!esp.find("OK")){  
    esp.println("AT+CWMODE=1");  
    Serial.println("Setting is ....");  
  }  
  Serial.println("Set as client");  
  Serial.println("Connecting to the Network ...");  
  esp.println("AT+CWJAP=\"" +agAdi+"\", \"" +agSifresi+"\"");  
  while(!esp.find("OK"));  
  Serial.println("connected to the network.");  
  delay(1000);  
}
```

Εικόνα 2.5: Παράδειγμα του κώδικα Arduino

```
void loop() {  
  esp.println("AT+CIPSTART=\"TCP\", \"" +ip+"\", 80"); // Syndeomaste sto I  
  if(esp.find("Error")){ // Elegxoume gia cor  
    Serial.println("AT+CIPSTART Error");  
  }  
  //DHT11.read(dht11Pin);  
  float h = dht.readHumidity();  
  float t = dht.readTemperature();  
  temper = t;  
  humi = h;  
  String veri = "GET https://api.thingspeak.com/update?api_key=LKNIEX8BU9J1L2N8";  
  veri += "sfield1=";  
  veri += String(temper); // H metavlith ths thermokrasias pou stelnoume  
  veri += "sfield2=";  
  veri += String(humi); // H metavlith ths ygrasias pou stelnoume  
  veri += "\r\n\r\n";  
  esp.print("AT+CIPSEND="); // Dinoume to megeth  
  esp.println(veri.length()+2);  
  delay(2000);  
  if(esp.find(">")){ // Oi entoles autes  
    esp.print(veri); // Stelnoume ta ded  
    Serial.println(veri);  
    Serial.println("Data sent.");  
    delay(1000);  
  }  
  Serial.println("Connection Closed.");  
  esp.println("AT+CIPCLOSE"); // Kleinoume th synde  
  delay(1000); // Perimenoume ena le  
}
```

Εικόνα 2.6: Παράδειγμα του κώδικα Arduino

Σημείωση: Η χρήση του όρου void σημαίνει πως η συνάρτηση δεν επιστρέφει τιμές.

2.3 Ο προγραμματισμός του Arduino

Η βασική λογική του κώδικα Arduino είναι σαν μια δομή «If-then» και μπορεί να χωριστεί σε 4 μέρη:

- **Setup**- γράφεται συνήθως στο κομμάτι setup του κώδικα Arduino, και εκτελεί πράγματα που χρειάζεται να γίνουν μια φορά, όπως για παράδειγμα η προσαρμογή ενός αισθητήρα.
- **Input**- στο ξεκίνημα του βρόγχου διαβάζει τις εισόδους. Αυτές οι τιμές θα χρησιμοποιηθούν σαν συνθήκες («if») όπως η ανάγνωση του περιβάλλοντος φωτισμού από ένα LDR χρησιμοποιώντας την AnalogRead().
- **Manipulate Data** – αυτό το κομμάτι χρησιμοποιείται για να μετατρέψει τα δεδομένα σε μια πιο κατάλληλη μορφή ή για να πραγματοποιήσει υπολογισμούς. Για παράδειγμα η AnalogRead() δίνει μετρήσεις από 0-1023 ο οποίες μπορούν να αντιστοιχιστούν στην κλίμακα από 0-255 ώστε να χρησιμοποιηθούν σαν PWM
- **Output** – Αυτό το κομμάτι ορίζει το τελικό αποτέλεσμα του λογικού («τότε») σύμφωνα με τα δεδομένα που υπολογίστηκαν στο προηγούμενο βήμα. Όπως για παράδειγμά με τα LDR και PWM, ενεργοποιείται ένα LED μόνο όταν το επίπεδο του περιβάλλοντος φωτός πέσει κάτω από ένα συγκεκριμένο όριο.

2.4 Οι βιβλιοθήκες του κώδικα Arduino

Δομή Βιβλιοθήκης

Μια βιβλιοθήκη είναι ένας φάκελος που αποτελείται από αρχεία C++ (.cpp) κώδικα αρχείων και αρχεία κεφαλίδας C++ (.h).

Το **.h** αρχείο περιγράφει τη δομή της βιβλιοθήκης και δηλώνει όλες τις μεταβλητές και τις συναρτήσεις της.

Το **.cpp** αρχείο περιέχει την υλοποίηση της συνάρτησης

Εισαγωγή Βιβλιοθηκών

Το πρώτο πράγμα που πρέπει να γίνει από το χρήστη είναι να βρεί τη βιβλιοθήκη που θέλει να χρησιμοποιήσει από τις πολλές βιβλιοθήκες που είναι διαθέσιμες στο διαδίκτυο. Αφού την κατεβάσει στον υπολογιστή, πρέπει απλώς να ανοίξει το Arduino IDE και να κάνει κλικ στο Sketch > Include Library > Manage Libraries. Στη συνέχεια, μπορεί να επιλέξει τη βιβλιοθήκη που θέλει να εισάγει στο IDE. Μόλις ολοκληρωθεί η διαδικασία, η βιβλιοθήκη θα είναι διαθέσιμη στο μενού του sketch.

Από το Software στο Hardware

Υπάρχουν πολλά που μπορούν να ειπωθούν για τις δυνατότητες λογισμικού του Arduino, αλλά είναι σημαντικό να θυμόμαστε ότι η πλατφόρμα αποτελείται τόσο από λογισμικό όσο και από υλικό. Αυτά τα δύο συνεργάζονται παράλληλα για να τρέξουν ένα πολύπλοκο λειτουργικό σύστημα.

Κώδικας → Μεταγλώττιση → Φόρτωση → Εκτέλεση

Στον πυρήνα του Arduino, βρίσκεται η δυνατότητα μεταγλώττισης και εκτέλεσης του κώδικα.

Αφού γραφτεί ο κώδικας στο IDE, πρέπει να τον ανεβεί στο Arduino. Κάνοντας κλικ στο κουμπί Μεταφόρτωση (το εικονίδιο με το δεξί βέλος), θα μεταγλωττιστεί ο κώδικας και θα ανεβεί εάν περάσει από τη μεταγλώττιση. Μόλις ολοκληρωθεί η μεταφόρτωση το πρόγραμμα θα ξεκινήσει να εκτελείται αυτόματα.

Παρακάτω φαίνεται η διαδικασία βήμα βήμα:

Αρχικά, μεταγλωττίζεται ο κώδικας. Για να γίνει αυτό, απλώς κάνουμε κλικ στο εικονίδιο check (ή κάνουμε κλικ στο sketch > Verify / Compile στη γραμμή μενού).



Εικόνα 2.7: Η μεταγλώττιση του κώδικα Arduino

Μπορεί να διακριθεί το εικονίδιο check που βρίσκεται επάνω αριστερά κάτω από την ετικέτα «File» στην ενότητα μενού. Μόλις γίνει αυτό, το Arduino θα αρχίσει να μεταγλωττίζει. Μόλις τελειώσει, θα εμφανιστεί ένα μήνυμα ολοκλήρωσης που μοιάζει με αυτό της παρακάτω εικόνας:

```
Done compiling.
Executable segment sizes:
ICACHE : 32768      - flash instruction cache
IROM   : 243592     - code in flash          (default or ICACHE_FLASH_ATTR)
IRAM   : 28837 / 32768 - code in IRAM          (IRAM_ATTR, ISRs...)
DATA   : 1504      - initialized variables (global, static) in RAM/HEAP
RODATA : 1300 / 81920 - constants              (global, static) in RAM/HEAP
BSS    : 26128     - zeroed variables      (global, static) in RAM/HEAP
Sketch uses 275233 bytes (26%) of program storage space. Maximum is 1044464 bytes.
Global variables use 28932 bytes (35%) of dynamic memory, leaving 52988 bytes for local variables. Maximum is 81920 bytes.
```

Εικόνα 2.8: Η κονσόλα του Arduino IDE

Όπως μπορούμε να δούμε, η μπλε γραμμή στο κάτω μέρος της σελίδας γράφει «done compiling». Εάν ο κώδικας δεν εκτελεστεί, θα ειδοποιηθούμε στην ίδια ενότητα και ο προβληματικός κώδικας θα επισημανθεί για επεξεργασία.

Αφού έχει μεταγλωττιστεί το sketch, είναι ώρα να ανέβει στην πλακέτα.

- Επιλέγουμε τη σειριακή θύρα στην οποία είναι συνδεδεμένο το Arduino. Για να το κάνουμε αυτό, κάνουμε κλικ στο Tools > Serial port στο μενού για να ορίσουμε τη σειριακή θύρα που έχετε επιλέξει (όπως φαίνεται παραπάνω). Στη συνέχεια, μπορούμε να ανεβάσουμε το μεταγλωττισμένο sketch.
- Για να ανεβάσουμε το sketch, κάνουμε κλικ στο εικονίδιο μεταφόρτωσης δίπλα στο τικ. Εναλλακτικά, μπορούμε να μεταβούμε στο μενού και να κάνουμε κλικ στο File> upload. Τα LED του Arduino θα αναβοσβήνουν μόλις μεταφερθούν τα δεδομένα.

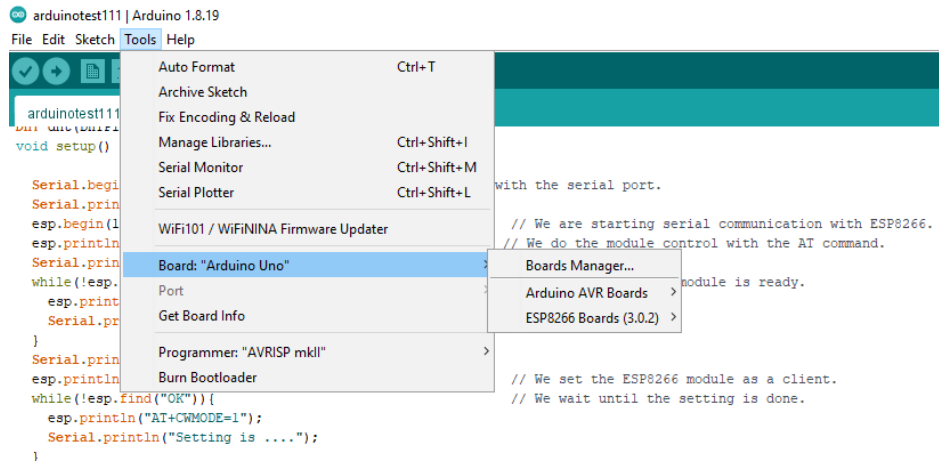
Μόλις ολοκληρωθεί, θα εμφανιστεί ένα μήνυμα ολοκλήρωσης που θα μας ενημερώνει ότι το Arduino ολοκλήρωσε τη μεταφόρτωση.

2.5 Η εγκατάσταση του IDE

2.5.α Για πλακέτες Arduino

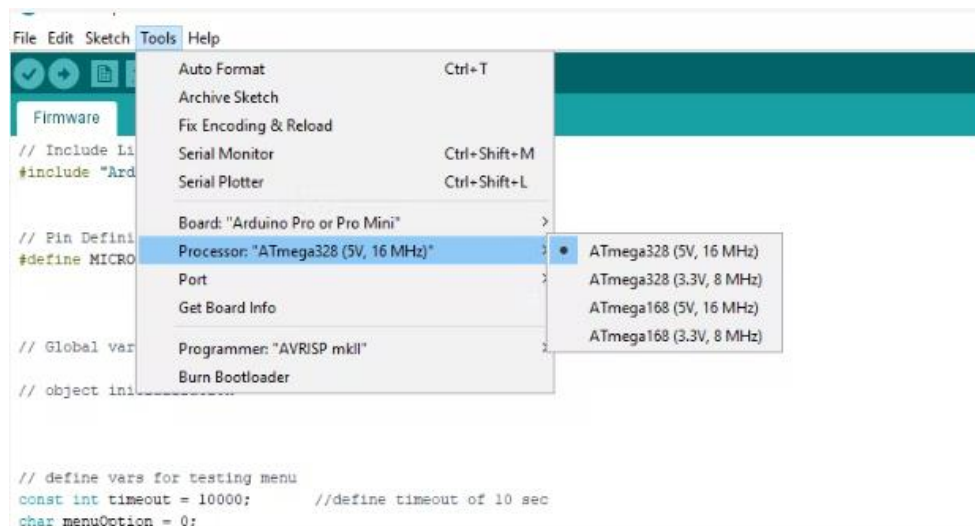
Για να συνδεθεί μια πλακέτα Arduino στον υπολογιστή χρειάζεται ένα καλώδιο USB. Όταν χρησιμοποιείται το Arduino UNO, το USB μεταφέρει τα δεδομένα του προγράμματος απευθείας στην πλακέτα. Το καλώδιο USB χρησιμοποιείται για την τροφοδοσία του Arduino. Μπορεί επίσης να τροφοδοτηθεί με ρεύμα το Arduino μέσω μιας εξωτερικής πηγής τροφοδοσίας.

Για να ανεβεί ο κώδικας, υπάρχουν ορισμένες ρυθμίσεις που πρέπει να διαμορφωθούν. Επιλέγουμε την πλακέτα μας - Πρέπει να ορίσουμε ποια πλακέτα Arduino πρόκειται να χρησιμοποιήσουμε. Αυτό γίνεται κάνοντας κλικ στην επιλογή Tools > Board > Η πλακέτα που έχουμε.



Εικόνα 2.9: Επιλογή της κατάλληλης πλακέτας

Επιλέγουμε τον επεξεργαστή της πλακέτας μας - υπάρχουν ορισμένες πλακέτες (για παράδειγμα Arduino pro-mini) για τις οποίες πρέπει να καθορίσουμε ποιο μοντέλο επεξεργαστή έχουμε. Στην περιοχή tools > processor > επιλέγουμε το μοντέλο που έχουμε.



Εικόνα 2.10: Επιλογή του κατάλληλου επεξεργαστή στην πλακέτα Arduino pro-mini (<https://www.circuito.io/>)

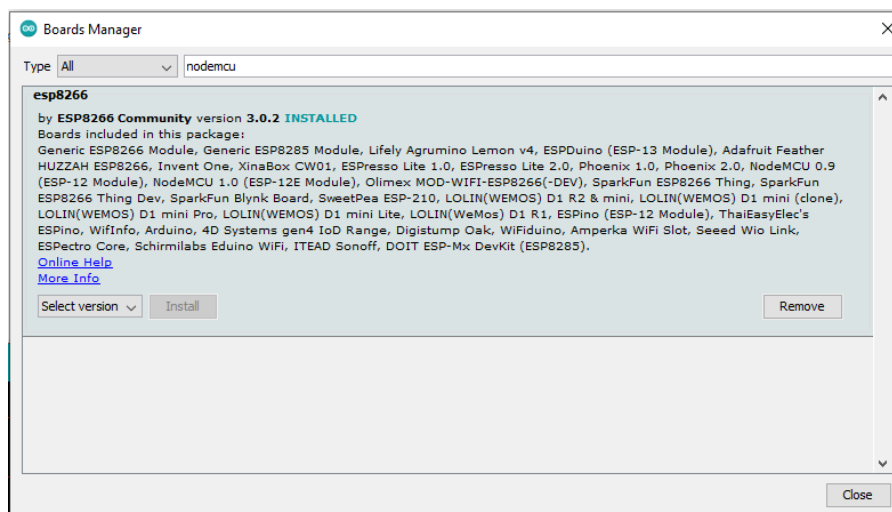
Επιλέγουμε τη θύρα μας - για να επιλέξουμε τη θύρα στην οποία είναι συνδεδεμένη η πλακέτα μας, επιλέγουμε tools > Port > COMX Arduino (Αυτή είναι η σειριακή θύρα του Arduino).

2.5.β Για non-native πλακέτες

Ορισμένες πλακέτες δεν είναι προεγκατεστημένες στο Arduino IDE, επομένως θα πρέπει να τις εγκαταστήσουμε για να μπορέσουμε να ανεβάσουμε κώδικα.

Για να εγκαταστήσουμε μια Non -Native πλακέτα όπως το NodeMCU, πρέπει να κάνουμε τα εξής:

1. Κλικ στο Tools > Boards > Boards Manager
2. Αναζητάμε στη γραμμή αναζήτησης την πλακέτα που θέλουμε να προσθέσουμε και κάνουμε κλικ στο «install». Ορισμένες πλακέτες δεν μπορούν να βρεθούν μέσω του Board Manager. Σε αυτήν την περίπτωση, θα χρειαστεί να προστεθούν χειροκίνητα. Για να γίνει αυτό πρέπει να πράξουμε ως εξής:
 - Κλικ στο Files > Preferences
 - Στο πεδίο Additional Boards Manager κάνουμε επικόλληση τη διεύθυνση URL του πακέτου εγκατάστασης της πλακέτας που θέλουμε να εγκαταστήσουμε. Για παράδειγμα, για το NodeMCU, προσθέτουμε την ακόλουθη διεύθυνση URL:
http://arduino.esp8266.com/stable/package_esp8266com_index.json
 - Κλικ στο OK
 - Πάμε στο Tools > Boards > Boards Manager
 - Αναζητάμε στη γραμμή αναζήτησης την πλακέτα που θέλουμε να προσθέσουμε και κάνουμε κλικ στο «install».



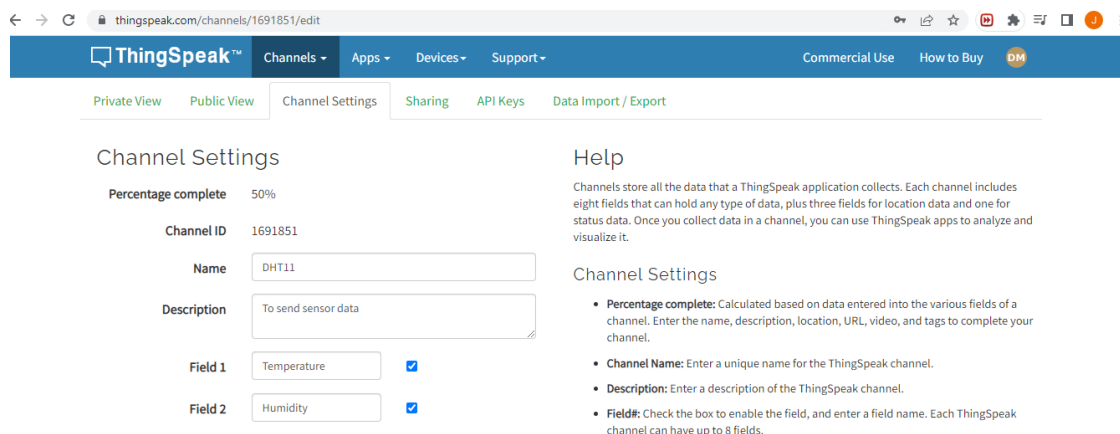
Εικόνα 2.11: Εγκατάσταση Non-Native πλακέτας στο Board Manager

Μόλις ολοκληρωθεί αυτό το βήμα, θα δούμε τις εγκατεστημένες πλακέτες στη λίστα των πλακετών κάτω από το tools.

ΚΕΦΑΛΑΙΟ 3 Περιγραφή της πλατφόρμας Thingspeak IOT

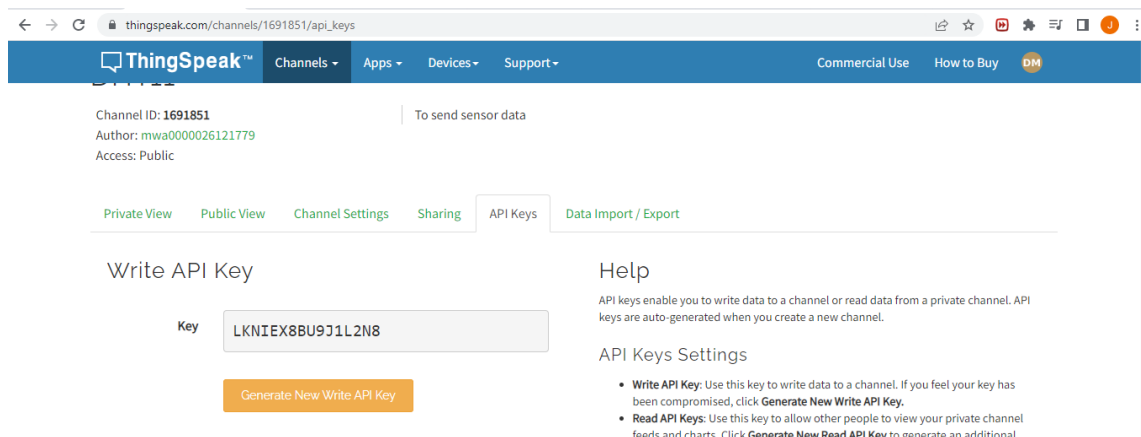
Το ThingSpeak είναι μια υπηρεσία πλατφόρμας ανάλυσης στοιχείων IOT που επιτρέπει την συγκέντρωση, την οπτικοποίηση και την ανάλυση ζωντανών ροών δεδομένων στο cloud. Μπορούμε να στείλουμε δεδομένα στο ThingSpeak από τις συσκευές μας, να δημιουργήσουμε άμεση οπτικοποίηση ζωντανών δεδομένων και να στείλουμε ειδοποιήσεις.

Για να στείλουμε δεδομένα στο Thingspeak πρέπει να δημιουργήσουμε ένα κανάλι στην πλατφόρμα. Στη συνέχεια πρέπει να ορίσουμε τα πεδία στα οποία θα στείλουμε τα δεδομένα μας.



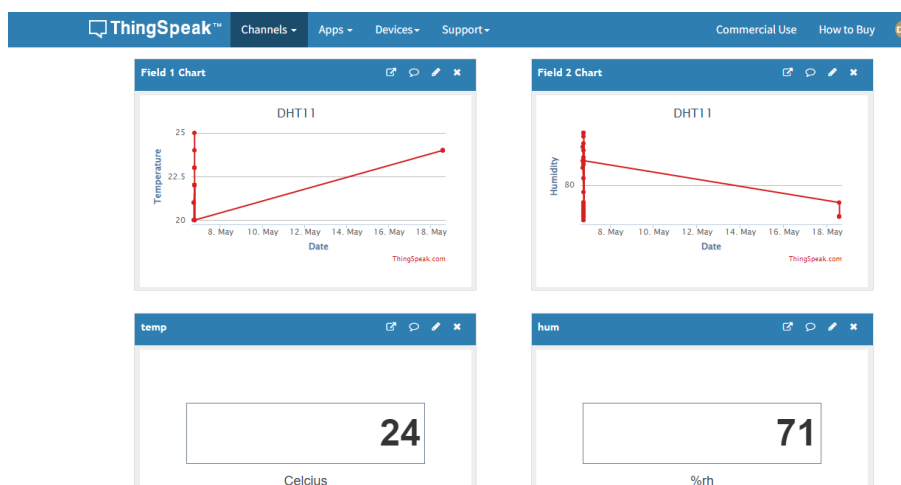
Εικόνα 3.1: Οι ρυθμίσεις καναλιού στο Thingspeak

Το κανάλι μας διαθέτει ένα μοναδικό αριθμό κλειδί API με τον οποίο χρησιμοποιώντας τον στον κώδικά μας μπορούμε να στείλουμε τα δεδομένα στο κανάλι μας. Στον κώδικα χρησιμοποιήθηκε μόνο το write Api key επειδή δεν χρειάζεται να διαβαστεί κάποιο δεδομένο από το Thingspeak. Ο αριθμός του write Api key είναι: LKNIEX8BU9J1L2N8



Εικόνα 3.2: Το write API key που χρησιμοποιείται στο Thingspeak

Αφού έχουμε κάνει τα παραπάνω και με την προϋπόθεση πως ο κώδικας μας δεν έχει λάθη και η συνδεσμολογία είναι σωστή τα δεδομένα θα σταλούν στα πεδία που έχουμε ορίσει .



Εικόνα 3.3: Οι γραφικές παραστάσεις και οι τιμές των δεδομένων που έχουν σταλεί στο Thingspeak

ΚΕΦΑΛΑΙΟ 4 Related Work

4.1 Η ενσωμάτωση της τεχνολογίας ΙΟΤ για αποτελεσματική περιβαλλοντική παρακολούθηση

Με την αύξηση της αστικοποίησης, τα περιβαλλοντικά ζητήματα είναι σημαντικές ανησυχίες που πρέπει να αντιμετωπίσουν οι οργανισμοί. Η τεχνολογία ΙοΤ αποδεικνύεται αρκετά αποτελεσματική για τον μετριάσμό των προβλημάτων που σχετίζονται με το περιβάλλον. Το ΙοΤ είναι ένα τεράστιο θέμα που περιλαμβάνει την καλύτερη χρήση των πόρων, των λειτουργιών που βασίζονται σε αισθητήρες και των στρατηγικών που σχετίζονται με την αναβάθμιση των επιχειρήσεων. Η παρακολούθηση του περιβάλλοντος είναι η πιο ωφέλιμη εφαρμογή ΙοΤ. Χρησιμοποιεί προηγμένες συσκευές αισθητήρων για τον εντοπισμό της παρουσίας ρύπων στον αέρα και το νερό και προάγει την καλύτερη βιωσιμότητα. Έτσι γίνεται εφικτή η διατήρηση της ασφάλειας και της καθαριότητας των εγκαταστάσεων χρησιμοποιώντας την έξυπνη λύση της περιβαλλοντικής παρακολούθησης.

Σε κλειστούς χώρους εργασίας, όπως υπόγεια ορυχεία, χώρους γραφείων ή δωμάτια, καθίσταται απαραίτητο να διατηρηθεί μια ευνοϊκή ατμόσφαιρα. Τα δεδομένα του αέρα συλλέγονται μέσω αισθητήρων και αποστέλλονται σε μια πλατφόρμα cloud μέσω πυλών. Οι οργανισμοί μπορούν εύκολα να συλλέξουν τις πληροφορίες σε μια φιλική προς το χρήστη μορφή μέσω ενός συνδεδεμένου πίνακα ελέγχου και να τις αναλύσουν για καλύτερα αποτελέσματα.

Πολλές περιβαλλοντικές τάσεις είναι πολύπλοκες και δύσκολο να αντιληφθούν. Υπάρχουν πολλές προκλήσεις στη χρήση παραδοσιακών τεχνικών για την παρακολούθηση των περιβαλλοντικών συνθηκών. Με την έλευση της τεχνολογικής εποχής και των έξυπνων

στρατηγικών διαχείρισης, έχει γίνει ευκολότερη η παρακολούθηση σε πραγματικό χρόνο του περιβάλλοντος των ανθρώπων. Το IoT παίζει μεγάλο ρόλο στη βελτίωση της ποιότητας της ατμόσφαιρας. Διαθέτει έξυπνες και καινοτόμες στρατηγικές για τη μέτρηση της ποιότητας του αέρα και για την επεξεργασία του νερού, οι οποίες συμβάλλουν σε μεγάλο βαθμό στην καλυτέρευση της βιωσιμότητας. Επιπλέον, το IoT χρησιμοποιεί μια προσέγγιση που βασίζεται σε δεδομένα για να παρέχει χρήσιμες πληροφορίες και προγνωστικά αποτελέσματα. Όσο καλύτερα το σύστημα μπορεί να παρακολουθεί, να μετράει και να καταλογίζει δεδομένα, τόσο καλύτερα θα γίνει η κατανόηση των πληροφοριών για βελτιώσεις.

Η έξυπνη περιβαλλοντική παρακολούθηση μπορεί να χωριστεί σε διάφορες κατηγορίες:

Παρακολούθηση ποιότητας νερού:

Το νερό είναι ο πιο απαραίτητος πόρος για την επιβίωση[9]. Απαιτεί συνεχή ποιοτική παρακολούθηση και κατάλληλες τεχνικές διαχείρισης για αποτελεσματικές πρακτικές διατήρησης της καθαρής σύστασης του. Η τεχνολογία IoT περιλαμβάνει έξυπνους μετρητές νερού, αισθητήρες και συστήματα άρδευσης που εγκαθίστανται σε σπίτια και έξυπνα κτίρια για την παρακολούθηση σε πραγματικό χρόνο των προτύπων κατανάλωσης νερού. Επίσης, εγκαθίστανται προηγμένες συσκευές αισθητήρων για τη μέτρηση των επιπέδων pH, των επιπέδων οξυγόνου και άλλων συστατικών, όπως μολυσματικών ουσιών που ενδέχεται να μην είναι ασφαλείς για σκοπούς της κατανάλωσης του. Η ανίχνευση τέτοιων παραμέτρων προσφέρει αξιόπιστες περιπτώσεις χρήσης που προωθούν την καλυτέρευση της ανθρώπινης υγείας και συμβάλλουν στη βελτίωση του βιοτικού επιπέδου.

Παρακολούθηση ποιότητας αέρα:

Στο περιβάλλον, [8] η ποιότητα του αέρα πρέπει να είναι καθαρή και υγιής για την αναπνοή, διαφορετικά μπορεί να οδηγήσει σε σοβαρά προβλήματα υγείας. Ακόμη και όταν η εργασία γίνεται σε κλειστό χώρο, ο περιβάλλον αέρας πρέπει να είναι καθαρός και φρέσκος. Οι λύσεις περιβαλλοντικής παρακολούθησης που λειτουργούν με IoT, όταν εγκαθίστανται σε συγκεκριμένους χώρους, μπορούν να χρησιμοποιηθούν για τον εντοπισμό των σωματιδίων στον αέρα και τη λήψη των δεδομένων μέσω αισθητήρων. Οι οργανισμοί μπορούν να μεταφέρουν τα δεδομένα σε μια πλατφόρμα cloud για επεξεργασία. Χρήσιμες πληροφορίες αποστέλλονται στον πίνακα ελέγχου του χρήστη. Η όλη διαδικασία επιτρέπει την παρακολούθηση του αέρα σε πραγματικό χρόνο, τη γρήγορη λήψη αποφάσεων σε περίπτωση τυχόν προβλημάτων, την αναζωογόνηση του αέρα και τον έλεγχο του αερισμού για καλύτερη κυκλοφορία του αέρα.

Ενεργειακή παρακολούθηση:

Τα προϊόντα που λειτουργούν με IoT είναι οικονομικά, αποδοτικά και μπορούν να εφαρμοστούν για να βελτιώσουν την ατμόσφαιρα. Η τεχνολογία [14] που βασίζεται σε αισθητήρες είναι η πιο αποτελεσματική για σκοπούς εξοικονόμησης κόστους που σχετίζονται με την κατανάλωση ενέργειας και τη βελτιστοποίηση της παραγωγικότητας των εργαζομένων. Ένα έξυπνο σύστημα περιβαλλοντικής παρακολούθησης παρέχει όλες τις λεπτομέρειες για τα πρότυπα κατανάλωσης ενέργειας και τις περιοχές όπου η ενέργεια χρησιμοποιείται άσκοπα, διευκολύνοντας τη μείωση της χρήσης ενέργειας. Επίσης, με την τεχνολογική πρόοδο, οι αισθητήρες έχουν ξεπεράσει τους παραδοσιακούς τρόπους ενεργοποίησης και απενεργοποίησης των φωτών. Με τη χρήση μιας λύσης που λειτουργεί

με IoT, γίνεται εφικτός ο έλεγχος εξ αποστάσεως τη χρήσης της ενέργειας, με αποτέλεσμα την εξοικονόμηση των πόρων.

Ανίχνευση τοξικών αερίων:

Το IoT για περιβαλλοντική παρακολούθηση είναι ο καταλληλότερος τρόπος για να έχουμε τον απομακρυσμένο έλεγχο ενός περιβάλλοντος. Οι κλειστοί χώροι συχνά εκτίθενται σε επιβλαβή αέρια, [8]τα οποία μπορούν να κοστίσουν σε υποδομές και σε μεγαλύτερα επίπεδα σε ανθρώπινες ζωές. Ωστόσο, η προ ανίχνευση τέτοιων επιβλαβών αερίων παρέχει μια σαφέστερη εικόνα των μελλοντικών συμβάντων και μπορεί να μειώσει τις καταστροφικές επιπτώσεις με σημαντικό κόστος. Οι αισθητήρες είναι τα βασικά στοιχεία που επιτρέπουν την αυτοματοποιημένη και σε πραγματικό χρόνο αναγνώριση τέτοιων αερίων, η οποία με τη σειρά της παρέχει ένα ιστορικό των δεδομένων για ανάλυση, επιτρέποντας προγνωστικά αποτελέσματα.

4.2 Η ενσωμάτωση της τεχνολογίας IoT για την αποτελεσματική περιβαλλοντική παρακολούθηση στην αγροτική και κτηνοτροφική παραγωγή και η χρησιμότητα της

Ως αποτέλεσμα της μείωσης του αγροτικού εργατικού δυναμικού, έχει ενεργοποιηθεί η υιοθέτηση λύσεων σύνδεσης στο Διαδίκτυο στις γεωργικές πρακτικές, για να μειωθεί η ανάγκη για χειρωνακτική εργασία. Οι λύσεις IoT επικεντρώνονται στο να βοηθήσουν τους αγρότες να καλύψουν το χάσμα ζήτησης προσφοράς, διασφαλίζοντας υψηλές αποδόσεις, κερδοφορία και προστασία του περιβάλλοντος. Η προσέγγιση της χρήσης της τεχνολογίας IoT για τη διασφάλιση της βέλτιστης χρήσης των πόρων για την επίτευξη υψηλών αποδόσεων των καλλιεργειών και τη μείωση του λειτουργικού κόστους ονομάζεται γεωργία ακριβείας. Οι τεχνολογίες IoT στη γεωργία περιλαμβάνουν εξειδικευμένο εξοπλισμό, ασύρματη συνδεσιμότητα, λογισμικό και υπηρεσίες πληροφορικής.

Η έξυπνη γεωργία [13]που βασίζεται σε τεχνολογίες IoT δίνει τη δυνατότητα στους καλλιεργητές και τους αγρότες να μειώσουν τα απόβλητα και να βελτιώσουν την παραγωγικότητα που εξαρτώνται από την ποσότητα του λιπάσματος που χρησιμοποιείται μέχρι τον αριθμό των διαδρομών που έχουν κάνει τα αγροτικά οχήματα κατά συνέπεια υπάρχει αποτελεσματική χρήση πόρων όπως το νερό, ο ηλεκτρισμός κ.λπ. Το έξυπνο σύστημα IoT farming solutions είναι ένα σύστημα που έχει κατασκευαστεί για την παρακολούθηση του αγρού με τη βοήθεια αισθητήρων (φως, υγρασία, θερμοκρασία, υγρασία εδάφους, υγεία των καλλιεργειών κ.λπ.) και αυτοματοποίηση του συστήματος άρδευσης. Οι αγρότες μπορούν να παρακολουθούν τις συνθήκες του αγρού από οπουδήποτε. Μπορούν επίσης να επιλέξουν μεταξύ χειροκίνητων και αυτοματοποιημένων επιλογών για τη λήψη των απαραίτητων ενεργειών με βάση αυτά τα δεδομένα. Για παράδειγμα, εάν το επίπεδο υγρασίας του εδάφους μειωθεί, ο γεωργός μπορεί να

ενεργοποιήσει αισθητήρες για να ξεκινήσει την άρδευση. Η έξυπνη γεωργία είναι εξαιρετικά αποδοτική σε σύγκριση με τη συμβατική προσέγγιση.

Το IoT έχει τη δυνατότητα να μεταμορφώσει τη γεωργία σε πολλές πτυχές και αυτές είναι οι κύριες:

- **Δεδομένα που συλλέγονται από έξυπνους αισθητήρες γεωργίας**, [7] σε αυτήν την προσέγγιση διαχείρισης αγροκτημάτων, βασικό στοιχείο είναι οι αισθητήρες, τα συστήματα ελέγχου, η ρομποτική, τα αυτόνομα οχήματα, το αυτοματοποιημένο υλικό, η τεχνολογία μεταβλητού ρυθμού, οι ανιχνευτές κίνησης, η κάμερα με κουμπιά και οι φορητές συσκευές. Αυτά τα δεδομένα μπορούν να χρησιμοποιηθούν για την παρακολούθηση της κατάστασης της επιχείρησης γενικά, καθώς και για την απόδοση του προσωπικού, την αποδοτικότητα του εξοπλισμού. Η ικανότητα πρόβλεψης της παραγωγής της παραγωγής επιτρέπει τον προγραμματισμό για καλύτερη διανομή προϊόντων.
- **Γεωργικά Drones**, τα drones εδάφους και τα drones αέρος χρησιμοποιούνται στη γεωργία προκειμένου να ενισχυθούν διάφορες γεωργικές πρακτικές: αξιολόγηση της υγείας των καλλιεργειών, άρδευση, παρακολούθηση των καλλιεργειών, ψεκασμός καλλιεργειών, φύτευση και ανάλυση εδάφους και αγρού.
- **Παρακολούθηση ζώων και γεωγραφική περιήφραξη**, οι ιδιοκτήτες φάρμας μπορούν να χρησιμοποιήσουν ασύρματες εφαρμογές IoT για τη συλλογή δεδομένων σχετικά με την τοποθεσία, την ευημερία και την υγεία των βοοειδών τους. Αυτές οι πληροφορίες βοηθούν στην πρόληψη της εξάπλωσης νόσων και επίσης μειώνουν το κόστος εργασίας.
- **Έξυπνα θερμοκήπια**, ένα έξυπνο θερμοκήπιο σχεδιασμένο με τη βοήθεια του IoT παρακολουθεί και ελέγχει έξυπνα το κλίμα, εξαλείφοντας την ανάγκη για χειροκίνητη παρέμβαση.
- **Αναλυτικά Προγνωστικά στοιχεία για έξυπνη γεωργία**, [7] η πρόβλεψη καλλιέργειας παίζει βασικό ρόλο. Βοηθά τον αγρότη να αποφασίσει το μελλοντικό σχέδιο σχετικά με την παραγωγή της καλλιέργειας, την αποθήκευση, τις τεχνικές μάρκετινγκ και τη διαχείριση κινδύνου. Για την πρόβλεψη του ρυθμού παραγωγής του τεχνητού δικτύου καλλιέργειας χρησιμοποιούνται πληροφορίες που συλλέγονται από αισθητήρες του αγροκτήματος. Αυτές οι πληροφορίες περιλαμβάνουν παραμέτρους όπως το έδαφος, η θερμοκρασία, η πίεση, η βροχόπτωση και η υγρασία. Οι αγρότες μπορούν να λάβουν ακριβή δεδομένα εδάφους είτε από τον πίνακα ελέγχου είτε από μια προσαρμοσμένη εφαρμογή για κινητά.

Οι αγρότες έχουν αρχίσει να συνειδητοποιούν ότι το IoT είναι η κινητήρια δύναμη για την αύξηση της γεωργικής παραγωγής με οικονομικά αποδοτικό τρόπο.

ΚΕΦΑΛΑΙΟ 5 Η μεθοδολογία υλοποίησης της παρακολούθησης περιβάλλοντος με τη χρήση της πλακέτας Arduino Uno

5.1 Τα εξαρτήματα που απαιτούνται για την υλοποίηση του κυκλώματος για την παρακολούθηση του περιβάλλοντος

Τα εξαρτήματα που απαιτούνται είναι το Arduino Uno, ένας αισθητήρας θερμοκρασίας-υγρασίας DHT11 που θα καταγράφει τα δεδομένα από τον περιβάλλοντα χώρο, ένα wifi transmitter chip ESP8266 01 με το οποίο θα επιτευχθεί η ασύρματη επικοινωνία του Arduino Uno με το δίκτυο που θα συνδεθεί, ένα breadboard στο οποίο θα τοποθετηθεί ο αισθητήρας DHT11 και μερικά καλώδια με τα οποία θα γίνει η διασύνδεση των επιμέρους εξαρτημάτων.

Τα εξαρτήματα που αποτελούν το κύκλωμα:

- Το Arduino UNO
- Ο αισθητήρας DHT11
- Το chip ESP8266 01
- Το breadboard
- Τα καλώδια

To Arduino UNO

Το Arduino Uno αποτελεί το βασικό κομμάτι του κυκλώματος. Παρέχει την τροφοδοσία του ρεύματος στα επιμέρους εξαρτήματα. Λαμβάνει, αποκωδικοποιεί και επεξεργάζεται τα δεδομένα που δέχεται από το περιβάλλον.

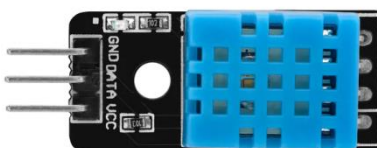


Arduino UNO R3 - SMD

Εικόνα 5.1: Η πλακέτα Arduino Uno (<https://www.twinschip.com/>)

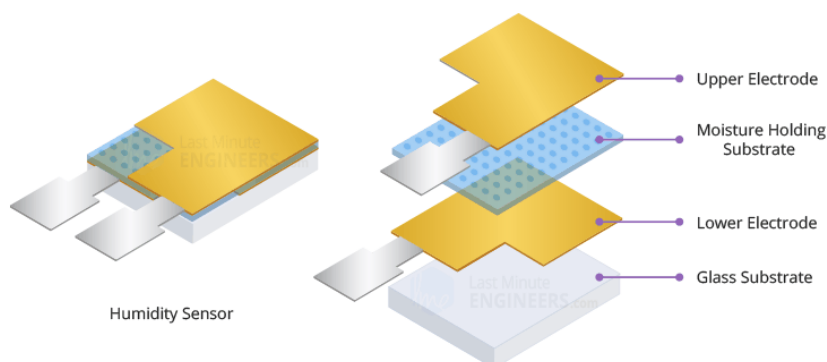
Ο αισθητήρας θερμοκρασίας και υγρασίας DHT11

Αυτός ο αισθητήρας είναι ήδη βαθμονομημένος και δεν απαιτεί επιπλέον εξαρτήματα. Έτσι μπορούμε να ξεκινήσουμε τη μέτρηση της σχετικής υγρασίας και θερμοκρασίας αμέσως.



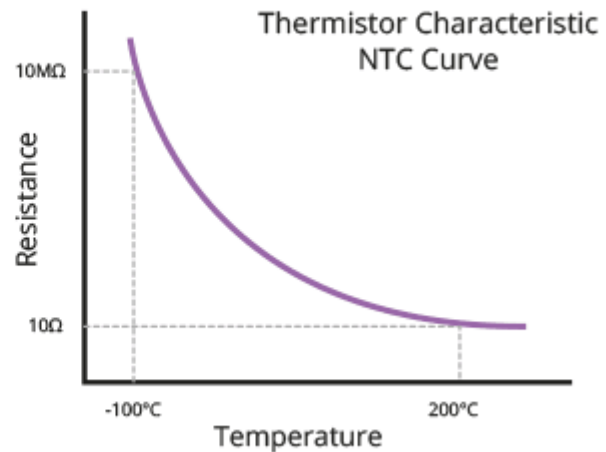
Εικόνα 5.2: Ο αισθητήρας DHT11(<https://www.az-delivery.de/>)

Μέσα στον DHT11, [3] υπάρχει ένα στοιχείο ανίχνευσης υγρασίας μαζί με ένα θερμίστορ. Το στοιχείο ανίχνευσης υγρασίας έχει δύο ηλεκτρόδια με ένα υπόστρωμα συγκράτησης υγρασίας τοποθετημένο ανάμεσά τους. Τα ιόντα απελευθερώνονται από το υπόστρωμα καθώς οι υδρατμοί απορροφώνται από αυτό. Έτσι έχουμε σαν αποτέλεσμα την αύξηση της αγωγιμότητας μεταξύ των ηλεκτροδίων. Η αλλαγή στην αντίσταση μεταξύ των δύο ηλεκτροδίων είναι ανάλογη της σχετικής υγρασίας. Η υψηλότερη σχετική υγρασία μειώνει την αντίσταση μεταξύ των ηλεκτροδίων, ενώ η χαμηλότερη σχετική υγρασία αυξάνει την αντίσταση μεταξύ των ηλεκτροδίων.



Εικόνα 5.3: Το εσωτερικό του αισθητήρα DHT11 (<https://lastminuteengineers.com/>)

Ο DHT11 περιέχει επίσης ένα NTC/Thermistor για τη μέτρηση της θερμοκρασίας. Το θερμίστορ είναι μια θερμική αντίσταση της οποίας η αντίσταση αλλάζει δραστικά με τη θερμοκρασία. Ο όρος «NTC» σημαίνει «αρνητικός συντελεστής θερμοκρασίας(Negative Temperature Coefficient)», που σημαίνει ότι η αντίσταση μειώνεται με την αύξηση της θερμοκρασίας.

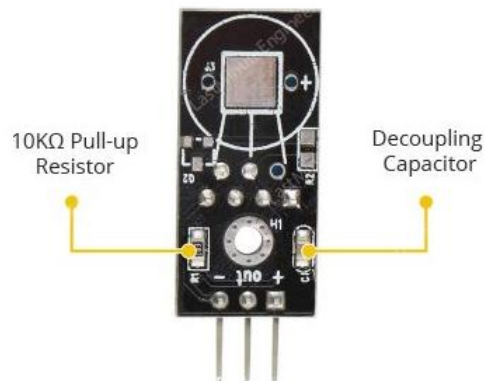


Εικόνα 5.4: Η γραφική παράσταση της συμπεριφοράς του θερμίστορ στις αλλαγές της θερμοκρασίας (<https://lastminuteengineers.com/>)

Στην άλλη πλευρά, υπάρχει ένα μικρό PCB με ένα 8-bit SOIC-14 IC. Αυτό το IC μετρά και επεξεργάζεται το αναλογικό σήμα με αποθηκευμένους συντελεστές βαθμονόμησης, κάνει μετατροπή από αναλογικό σε ψηφιακό και εκπέμπει ένα ψηφιακό σήμα με τη θερμοκρασία και την υγρασία. Ο DHT11 μπορεί να μετρήσει θερμοκρασία από 0°C έως 50°C με ακρίβεια $\pm 2,0^\circ\text{C}$ και υγρασία από 20% έως 80% με ακρίβεια 5%. Ο ρυθμός δειγματοληψίας του DHT11 είναι 1 Hz, που σημαίνει ότι μπορούμε να λαμβάνουμε νέα δεδομένα από αυτόν κάθε ένα δευτερόλεπτο.

Υποστηρικτικό κύκλωμα

Η μονάδα συνοδεύεται από όλα τα απαραίτητα κυκλώματα υποστήριξης έτσι είναι έτοιμη να λειτουργήσει χωρίς επιπλέον εξαρτήματα.



Εικόνα 5.5: Το εσωτερικό κύκλωμα του αισθητήρα DHT11 (<https://lastminuteengineers.com/>)

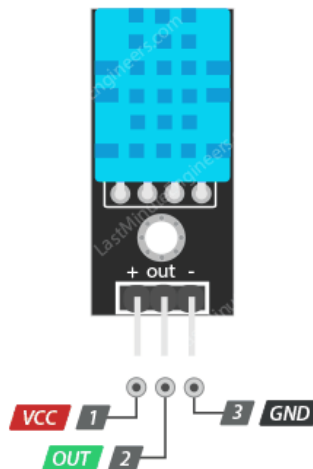
Διάταξη των ακροδεκτών του DHT11

Ο DHT11 έχει μόνο 3 ακροδέκτες:

Ο **VCC** ακροδέκτης τροφοδοτεί τον αισθητήρα με ρεύμα. Συνιστάται τροφοδοσία 5V, αν και η τάση τροφοδοσίας κυμαίνεται από 3,3V έως 5,5V. Στην περίπτωση τροφοδοσίας με 5V, μπορούμε να τοποθετήσουμε τον αισθητήρα σε απόσταση έως και 20 μέτρα. Ωστόσο, με τάση τροφοδοσίας 3,3 V, το μήκος του καλωδίου δεν πρέπει να είναι μεγαλύτερο από 1 μέτρο. Διαφορετικά, η πτώση τάσης γραμμής θα οδηγήσει σε σφάλματα στη μέτρηση.

Ο **OUT** ακροδέκτης χρησιμοποιείται για την επικοινωνία μεταξύ του αισθητήρα και του Arduino.

Ο **GND** ακροδέκτης συνδέεται με τη γείωση του Arduino



Εικόνα 5.6: Οι ακροδέκτες του αισθητήρα DHT11 (<https://lastminuteengineers.com/>)

To chip ESP8266 01

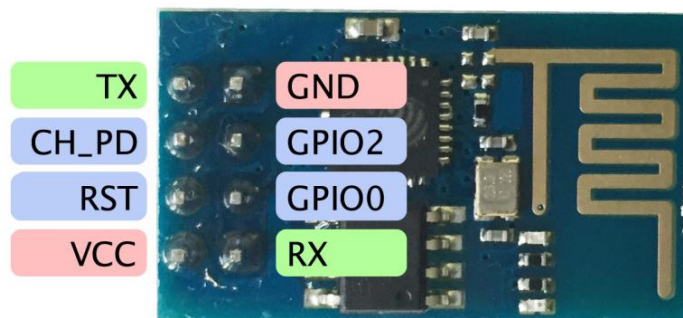
Τι είναι το ESP8266;

Το ESP8266 [6] είναι ένα wifi SOC (σύστημα σε τσιπ- system on a chip) που παράγεται από την Espressif Systems. Είναι ένα ολοκληρωμένο τσιπ που έχει σχεδιαστεί για να παρέχει πλήρη σύνδεση στο διαδίκτυο σε ένα μικρό πακέτο. Το ESP8266 μπορεί να χρησιμοποιηθεί ως εξωτερική μονάδα Wi-Fi, χρησιμοποιώντας το πρότυπο λογισμικό AT Commands συνδέοντάς το με οποιονδήποτε μικροελεγκτή χρησιμοποιώντας το σειριακό UART (στην δική μας περίπτωση ο μικροελεγκτής είναι το Arduino UNO) ή να χρησιμεύσει απευθείας ως μικροελεγκτής με δυνατότητα Wifi, προγραμματίζοντας ένα νέο λογισμικό χρησιμοποιώντας το SDK που παρέχεται. Οι GPIO ακροδέκτες επιτρέπουν Analog IO και Digital IO καθώς και PWM, SPI, I2C, κλπ.

Τεχνικά χαρακτηριστικά:

- 802.11 b / g / n
- Wi-Fi Direct (P2P), soft-AP
- Ενσωματωμένο TCP / IP πρωτόκολλο στοίβας
- Ενσωματωμένο TR switch, balun, LNA, ενισχυτή ισχύος και αντιστοίχιση δικτύου

- Ενσωματωμένο PLL, ρυθμιστή τάσης και εξαρτήματα διαχείρισης ισχύος
- 802.11b mode + 19.5dBm ισχύ εξόδου
- Ενσωματωμένος αισθητήρας θερμοκρασίας
- Υποστήριξη ποικιλίας κεραιών
- Το ρεύμα διαρροής είναι μικρότερο από 10uA
- Ενσωματωμένη CPU 32-bit χαμηλής κατανάλωσης: μπορεί να λειτουργήσει ως επεξεργαστής εφαρμογών
- SDIO 2.0, SPI, UART
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU, A-MSDU συνάθροιση με 0,4 Εντός αφύπνισης
- 2ms, σύνδεση και μεταφορά πακέτων δεδομένων
- κατανάλωση ενέργειας σε κατάσταση αναμονής μικρότερη από 1,0 mW (DTIM3)



Εικόνα 5.7: Οι ακροδέκτες του chip ESP8266 (<http://fabacademy.org/>)

Χρησιμοποιώντας το ESP8266 με το Arduino

Πολλές βιβλιοθήκες έχουν αναπτυχθεί για να χρησιμοποιούν το ESP8266 ως λειτουργική μονάδα για το Arduino. Επίσης, υπάρχει μια θύρα του Arduino IDE για απευθείας προγραμματισμό του ESP

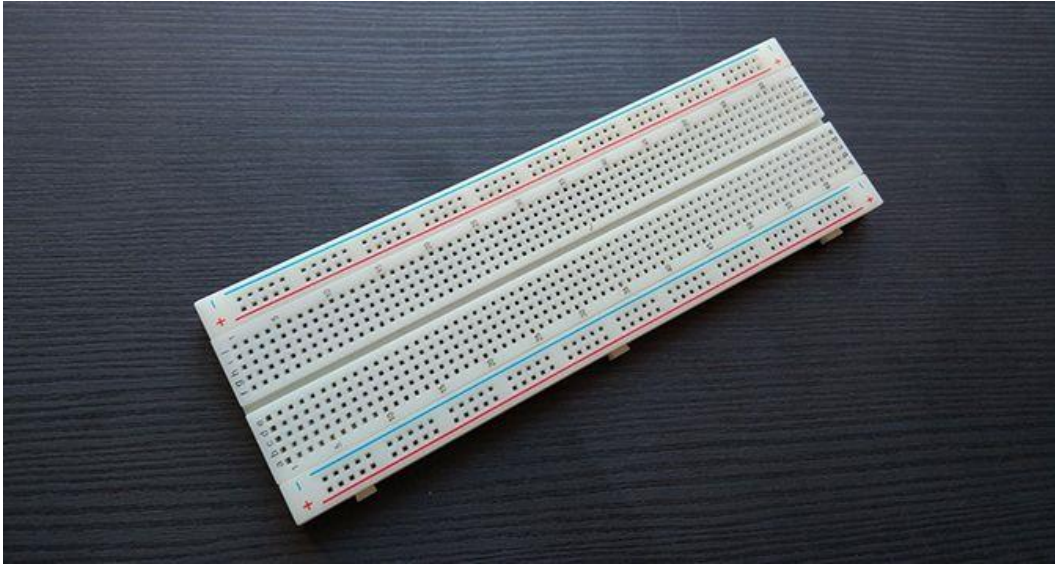
Χρησιμοποιώντας το ESP8266 μέσω σειριακής θύρας

Μπορούμε να αλληλεπιδράσουμε με το ESP8266 στέλνοντας AT commands μέσω των παρεχόμενων ακροδεκτών TX και RX χρησιμοποιώντας το Arduino. Χρειάζεται προσοχή στην τάση των γραμμών σήματος, καθώς το ESP8266 λειτουργεί μόνο με 3,3V και οι ακροδέκτες δεν αντέχουν στα 5V.

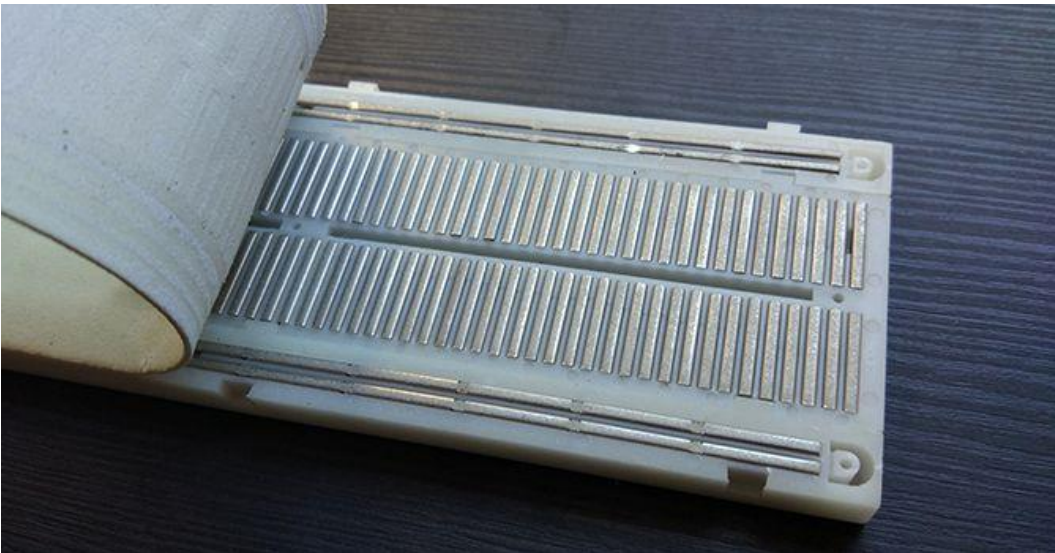
To breadboard

Τι είναι το breadboard;

Το Breadboard [5] είναι ένα είδος πλακέτας που χρησιμοποιείται για τη δημιουργία πρωτοτύπων ή την κατασκευή κυκλωμάτων. Μας επιτρέπει να τοποθετούμε εξαρτήματα και να κάνουμε συνδέσεις στην πλακέτα για να δημιουργούμε κυκλώματα χωρίς συγκολλήσεις. Οι τρύπες στο breadboard εξασφαλίζουν τις συνδέσεις μας κρατώντας με φυσικό τρόπο τα εξαρτήματα ή τα καλώδια που τοποθετούμε τα οποία συνδέονται ηλεκτρικά μέσα στην πλακέτα. Η ονομασία breadboard προήλθε από τους νέους σχεδιαστές κυκλωμάτων που χρησιμοποιούσαν στα πρώτα χρόνια του σχεδιασμού κυκλωμάτων την σανίδα κοπής ψωμιού (breadboard) με καρφωμένες βίδες και τυλιγμένα καλώδια γύρω τους για να δημιουργήσουν ολοκληρωμένα κυκλώματα

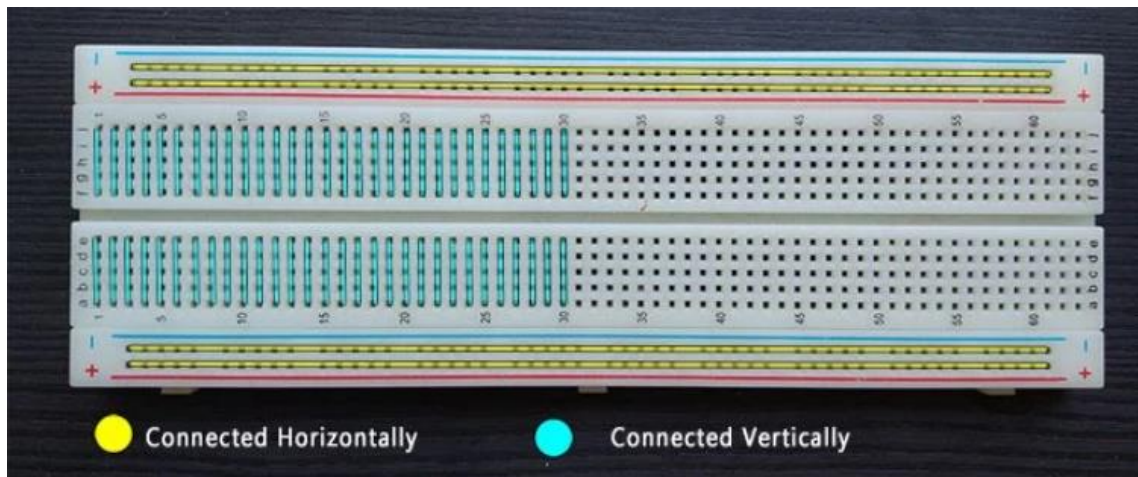


Εικόνα 5.8: Το Breadboard (<https://www.makeuseof.com/>)



Εικόνα 5.9: Το εσωτερικό του Breadboard (<https://www.makeuseof.com/>)

Τα δυο μεγάλα καλώδια και απο τις δυο πλευρές που διατρέχουν το Breadboard κατά μήκος του συνήθως χρησιμοποιούνται για να συνδέσουν μια πηγή ρεύματος στην πλακέτα. Συνήθως αναφέρονται σαν γραμμές ισχύος(power rails). Τα μικρότερα καλώδια που είναι τοποθετημένα κάθετα στην πλακέτα χρησιμοποιούνται για να συνδέσουμε τα εξαρτήματα του κυκλώματος μας.



Εικόνα 5.10: Οπτικοποίηση των καλωδίων του Breadboard (<https://www.makeuseof.com/>)

Οι γραμμές ισχύος βρίσκονται οριζόντια ως δύο σειρές στο επάνω και στο κάτω μέρος. Εν τω μεταξύ, οι κάθετες στήλες βρίσκονται προς τα μέσα καθώς μετακινούμαστε προς τα κάτω στην πλακέτα.

Τα καλώδια

Η αλληλεπίδραση μεταξύ των εξαρτημάτων του κυκλώματος θα γίνει με τη χρήση διαφόρων καλωδίων. Τα καλώδια που θα χρησιμοποιηθούν διαθέτουν ακροδέκτες σύνδεσης στις άκρες τους. Αυτό έχει ως αποτέλεσμα η σύνδεση μεταξύ των ακροδεκτών στο κύκλωμα να γίνεται χωρίς την χρήση κάποιου συγκολλητικού μέσου (π.χ. καλαί). Έτσι υπάρχει η δυνατότητα της άμεσης και εύκολης τροποποίησης του κυκλώματος χωρίς την χρήση εξειδικευμένων εργαλείων. Τα καλώδια αυτά προορίζονται κυρίως για τη δοκιμή των ολοκληρωμένων κυκλωμάτων και όχι για μόνιμη εγκατάσταση στο κύκλωμα, ωστόσο στην περίπτωση του δικού μας κυκλώματος είναι κατάλληλα χωρίς να επηρεάζεται σε κάποιο τομέα η λειτουργικότητα του κυκλώματος.



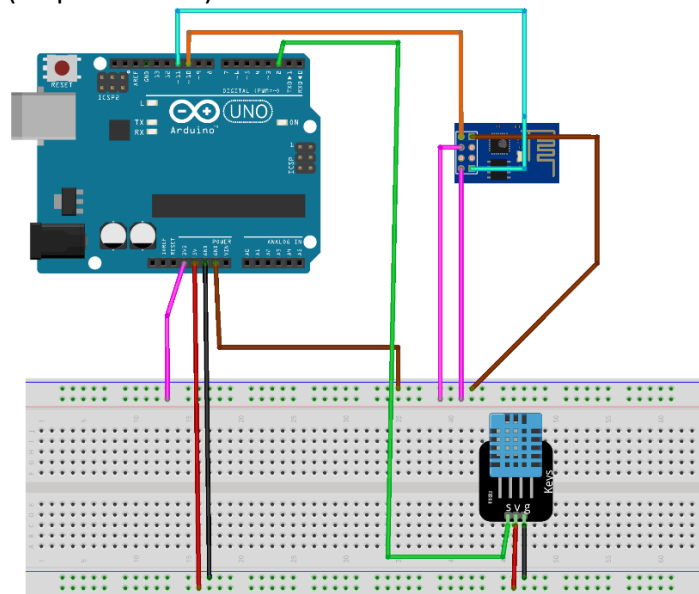
Εικόνα 5.11: Καλώδια με ακροδέκτες σύνδεσης (<https://export.rsdelivers.com/>)

5.2 Η ανάλυση και η περιγραφή της συνδεσμολογίας που απαιτείται για την υλοποίηση του κυκλώματος

Για να λειτουργήσουν τα εξαρτήματα πρέπει να δημιουργούν ένα κλειστό κύκλωμα. Ο DHT11 αισθητήρας είναι τοποθετημένος πάνω στο breadboard και τροφοδοτείται με

ρεύμα από το Arduino με τον ακροδέκτη 5V μέσω ενός κόκκινου καλωδίου που συνδέεται στον ακροδέκτη VCC του DHT11. Η μεταφορά των δεδομένων από τον αισθητήρα DHT11 γίνεται με τη σύνδεση του ακροδέκτη DATA του αισθητήρα DHT11 και του ακροδέκτη D2 του Arduino μέσω του πράσινου καλωδίου. Το κύκλωμα κλείνει με τη σύνδεση του ακροδέκτη γείωσης GND του DHT11 με τον ακροδέκτη GND του Arduino μέσω του μαύρου καλωδίου

Το chip ESP8266 τροφοδοτείται με ρεύμα από τον ακροδέκτη 3V3 του Arduino που συνδέεται στον ακροδέκτη VCC του chip μέσω του ροζ καλωδίου που έχει τοποθετηθεί στο breadboard. Στον ακροδέκτη 3V3 του Arduino συνδέεται επίσης με ροζ καλώδιο και ο ακροδέκτης CH_PD του ESP8266. Ο ακροδέκτης D10 του Arduino συνδέεται με τον ακροδέκτη UTXD του ESP8266 μέσω του πορτοκαλί καλωδίου. Ο ακροδέκτης D11 του Arduino συνδέεται με τον ακροδέκτη URXD του ESP8266 μέσω του κυανού καλωδίου. Το κύκλωμα κλείνει με τη σύνδεση του ακροδέκτη γείωσης GND του Arduino με τον ακροδέκτη GND του ESP8266 μέσω του καφέ καλωδίου. **Πρέπει να σημειωθεί πως ο αισθητήρας DHT11 και το chip ESP8266 είναι απαραίτητο να συνδεθούν σε διαφορετικές γειώσεις πάνω στο Arduino.** Στην περίπτωση που συνδεθούν στην ίδια γείωση δεν θα λειτουργήσει το κύκλωμα διότι δεν παρέχεται η σωστή τάση για να λειτουργήσουν. Για αυτό το λόγο ο DHT11 είναι συνδεδεμένος στον πρώτο ακροδέκτη γείωσης του Arduino (μαύρο καλώδιο), ενώ το chip ESP8266 είναι συνδεδεμένο στο δεύτερο ακροδέκτη γείωσης του Arduino (καφέ καλώδιο).



Εικόνα 5.12: Το ολοκληρωμένο κύκλωμα με το Arduino Uno και τα επιμέρους εξαρτήματα (υλοποιήθηκε με το πρόγραμμα Fritzing)

Πίνακας καλωδιώσεων

Arduino	Χρώμα καλωδίου	DHT11	ESP8266
5V	κόκκινο	VCC	
GND	μαύρο	GND	
D2	πράσινο	DATA	
3V3	ρόζ		VCC
3V3	ρόζ		CH_PD
D10	πορτοκαλί		UTXD
D11	κυανό		URXD
GND	καφέ		GND

5.3 Ο κώδικας που αναπτύχθηκε για το Arduino Uno

Στις επόμενες εικόνες θα αναρτηθεί ο κώδικας που υλοποιεί την παρακολούθηση του περιβάλλοντος και την αποστολή των δεδομένων στην διαδικτυακή πλατφόρμα Thingspeak IOT . Ο κώδικας μέσω του serial monitor μας ενημερώνει εάν το Arduino συνδέθηκε στο δίκτυο που έχουμε ορίσει ,εάν λαμβάνει τις εντολές που του έχουμε ορίσει και αν στέλνονται τα δεδομένα μας στην πλατφόρμα Thing speak. Για να εμφανιστούν τα δεδομένα στο serial monitor πρέπει το baud που θα οριστεί να είναι 9600 αφού έτσι το έχουμε ορίσει στην εντολή Serial.begin(9600);

```
arduinotest111
#include <DHT.h>
#include <SoftwareSerial.h>

String agAdi = "KarrasHome"; // Edw grafoyme to onoma tou diktyou mas
String agSifresi = "2695031430a"; // Edw grafoyme ton kwdiko tou diktyou mas

int rxPin = 10; // ESP8266 RX pin
int txPin = 11; // ESP8266 TX pin
int dht11Pin = 2;

String ip = "184.106.153.149"; // Thingspeak ip adress
float temper, humi;

//dht11 DHT11;
#define DHTPIN 2
SoftwareSerial esp(rxPin, txPin); // Edw ksekinaei to setup twv pins ths seiriakhs epikoinwnias
DHT dht(DHTPIN, DHT11);
void setup() {

  Serial.begin(9600); // Edw ksikinaei h epikoinwnia mas me th serial port
  Serial.println("Started");
  esp.begin(115200); // Ksekiname th seiriakh epikoinwnia me to ESP8266
  esp.println("AT"); // Elegxoume to module me thn entolh AT
  Serial.println("AT sent ");
  while(!esp.find("OK")){ // Perimenoume mexri to module na einai etoimo
    esp.println("AT");
    Serial.println("ESP8266 Not Find.");
  }
```

Εικόνα 5.13: Ο κώδικας που υλοποιεί την παρακολούθηση περιβάλλοντος

```
arduinotest111
}
Serial.println("OK Command Received");
esp.println("AT+CWMODE=1"); // Orizoume to ESP8266 module san client
while(!esp.find("OK")){ // Perimenoume mexri na ginei h rythmish
  esp.println("AT+CWMODE=1");
  Serial.println("Setting is ....");
}
Serial.println("Set as client");
Serial.println("Connecting to the Network ...");
esp.println("AT+CWMJAP=\""+agAdi+"\",\""+agSifresi+"\"); // Syndeomaste sto diktyo mas
while(!esp.find("OK")); // Perimenoume mexri na ginei h syndesh me to diktyo
Serial.println("connected to the network.");
delay(1000);
}
void loop() {
  esp.println("AT+CIPSTART=\"TCP\", \""+ip+"\",80"); // Syndeomaste sto Thingspeak.
  if(esp.find("Error")){ // Elegxoume gia connection error.
    Serial.println("AT+CIPSTART Error");
  }
  //DHT11.read(dht11Pin);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  temper = t;
  humi = h;
  String veri = "GET https://api.thingspeak.com/update?api_key=LKNIEX8BU9J1L2N8"; // H entolh gia to Thingspeak.
  veri += "%field1=";
  veri += String(temper);// H metavlith ths thermokrasias pou stelnoume
  veri += "%field2=";
```

Εικόνα 5.14: Ο κώδικας που υλοποιεί την παρακολούθηση περιβάλλοντος

```

veri += String(humi);    // Η μεταβλητή της υγρασίας που στέλνουμε
veri += "\r\n\r\n";
esp.print("AT+CIPSEND=");           // Δίνουμε το μέγεθος των δεδομένων που στέλνουμε στο ESP
esp.println(veri.length()+2);
delay(2000);
if(esp.find(">")){                  // Οι εντολές αυτές πραγματοποιούνται όταν το ESP8266 είναι έτοιμο
    esp.print(veri);                 // Στέλνουμε τα δεδομένα
    Serial.println(veri);
    Serial.println("Data sent.");
    delay(1000);
}
Serial.println("Connection Closed.");
esp.println("AT+CIPCLOSE");         // Κλείνουμε τη σύνδεση
delay(1000);                        // Περιμένουμε ένα λεπτό για να στείλουμε νέα δεδομένα
}

```

Εικόνα 5.15: Ο κώδικας που υλοποιεί την παρακολούθηση περιβάλλοντος

Παρακάτω βλέπουμε την εκτέλεση του κώδικα στο serial monitor και στο Thingspeak
Serial monitor



```

COM3
Started
AT sent
OK Command Received
Set as client
Connecting to the Network ...
connected to the network.
GET https://api.thingspeak.com/update?api_key=LKNIEX8BU9J1L2N8&field1=28.00&field2=65.00

Data sent.
Connection Closed.

```

Εικόνα 5.16: Στιγμιότυπο της εκτέλεσης του κώδικα στο serial monitor του IDE

```

COM3
Started
AT sent
OK Command Received
Set as client
Connecting to the Network ...
connected to the network.
GET https://api.thingspeak.com/update?api_key=LKNIEX8BU9JIL2N8&field1=28.00&field2=65.00

Data sent.
Connection Closed.
Connection Closed.
Connection Closed.
Connection Closed.
GET https://api.thingspeak.com/update?api_key=LKNIEX8BU9JIL2N8&field1=28.00&field2=64.00

Data sent.
Connection Closed.
GET https://api.thingspeak.com/update?api_key=LKNIEX8BU9JIL2N8&field1=28.00&field2=65.00

Data sent.
Connection Closed.
GET https://api.thingspeak.com/update?api_key=LKNIEX8BU9JIL2N8&field1=28.00&field2=65.00

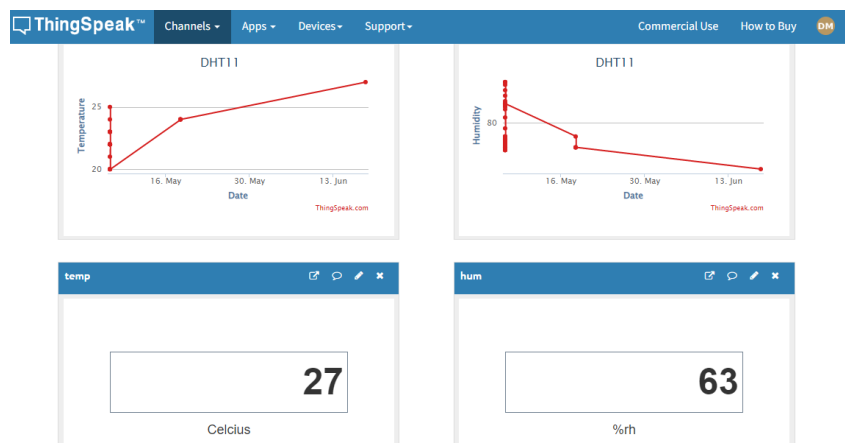
Data sent.
Connection Closed.
Connection Closed.
Connection Closed.
Connection Closed.
GET https://api.thingspeak.com/update?api_key=LKNIEX8BU9JIL2N8&field1=28.00&field2=64.00

Data sent.
Connection Closed.

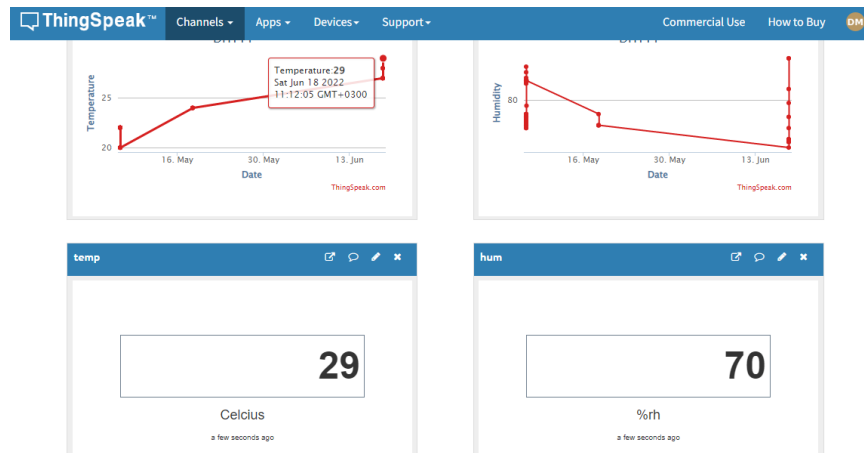
```

Εικόνα 5.17: Στιγμιότυπο της εκτέλεσης του κώδικα στο serial monitor του IDE

Τα δεδομένα που αποστέλλονται στο Thingspeak εμφανίζονται είτε με τη μορφή γραφικών παραστάσεων είτε με τη μορφή κονσόλας που εμφανίζει την τιμή των δεδομένων. Η μορφή που εμφανίζονται τα δεδομένα εξαρτάται από την επιθυμητή μορφή που έχει επιλέξει ο χρήστης. Κατά την αποστολή των δεδομένων έγιναν μετρήσεις με όργανα για να διαπιστωθεί η ακρίβεια των δεδομένων που καταγράφονται και αποστέλλονται. Οι δοκιμές έγιναν φυσώντας πάνω στον αισθητήρα θερμοκρασίας ή πλησιάζοντας κοντά του μια πηγή θερμότητας. Με αυτόν τον τρόπο είναι δυνατή η παρατήρηση στην αλλαγή των τιμών που διαβάζουν οι αισθητήρες.



Εικόνα 5.18: Τα δεδομένα που απεικονίζονται στο Thingspeak



Εικόνα 5.19: Τα δεδομένα που απεικονίζονται στο Thingspeak

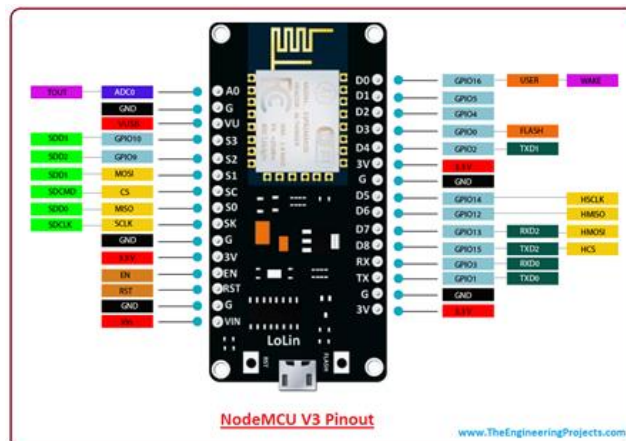
ΚΕΦΑΛΑΙΟ 6 Η υλοποίηση της παρακολούθησης περιβάλλοντος με τη χρήση της πλακέτας NodeMCU

6.1 Περιγραφή της πλακέτας NodeMCU V3

Το NodeMCU βασίζεται στο Espressif ESP8266-12E WiFi System-On-Chip(στην υλοποίηση με Arduino χρησιμοποιήθηκε η μονάδα ESP8266 01, ουσιαστικά είναι η ίδια μονάδα wifi απλά εδώ είναι εγκατεστημένη πάνω στην πλακέτα) με εγκατεστημένο ένα open-source, Lua-based firmware. Είναι τέλειο για IOT εφαρμογές και άλλες περιπτώσεις όπου η ασύρματη σύνδεση είναι απαραίτητη. Η πλακέτα έχει πολλές ομοιότητες με το Arduino , είναι εξοπλισμένες και οι δυο πλακέτες με μικροελεγκτές που μπορούν να προγραμματιστούν μέσω του Arduino IDE



Εικόνα 6.1: Το NodeMCU V3 (<https://www.circuito.io/>)



Εικόνα 6.2: Το διάγραμμα ακροδεκτών του NodeMCU V3(<https://www.theengineeringprojects.com/>)

6.2 Σύγκριση του NodeMCU με το Arduino

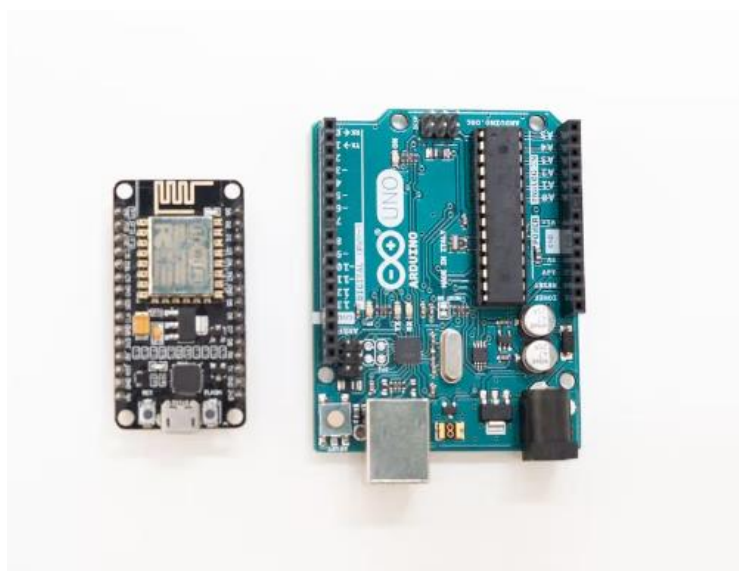
Το NodeMCU έχει πολλά διακριτά πλεονεκτήματα. Ωστόσο πρέπει να δηλώσουμε ότι το NodeMCU, τεχνικά, δεν είναι μια συσκευή, αλλά μάλλον το firmware που φορτώνεται σε μία. Στην πράξη το firmware NodeMCU είναι διαθέσιμο προς πώληση προεγκατεστημένο στην πλακέτα.

Υπήρξαν πολλές διαφορετικές μονάδες ESP όλα αυτά τα χρόνια, η καθεμία με τα δικά της πλεονεκτήματα και μειονεκτήματα. Ωστόσο, υπήρχαν μόνο δύο τύποι πλακών NodeMCU: οι εκδόσεις 0.9 και 1.0. Η έκδοση 0.9 είναι μπλε και έρχεται εγκατεστημένη με το τσιπ ESP-12, ενώ η 1.0 είναι μαύρη και συνοδεύεται από το ESP-12E (που σημαίνει «enhanced» δηλαδή βελτιωμένο). Υπάρχουν αρκετές βασικές διαφορές μεταξύ των δύο τσιπ, η πιο αξιοσημείωτη είναι ότι η νεότερη έκδοση έρχεται με 22 ακροδέκτες και η προηγούμενη με μόλις 16.

Δεδομένου ότι το ESP8266 είναι μια πιο πρόσφατη έκδοση από το Arduino, δεν προκαλεί έκπληξη το γεγονός ότι έχει ισχυρότερες προδιαγραφές. Υπάρχει ένας επεξεργαστής RISC 32 bit χρονισμένος στα 80 MHz, μαζί με γενναιόδωρα μεγάλη RAM και υποστήριξη για έως και 16 mb εξωτερικής αποθήκευσης flash. Η συσκευή είναι ιδιαίτερα χρήσιμη για εφαρμογές IoT, χάρη στο μικροσκοπικό της μέγεθος και την ενσωματωμένη υποστήριξη WiFi.

Πίνακας Σύγκρισης των δύο πλακετών

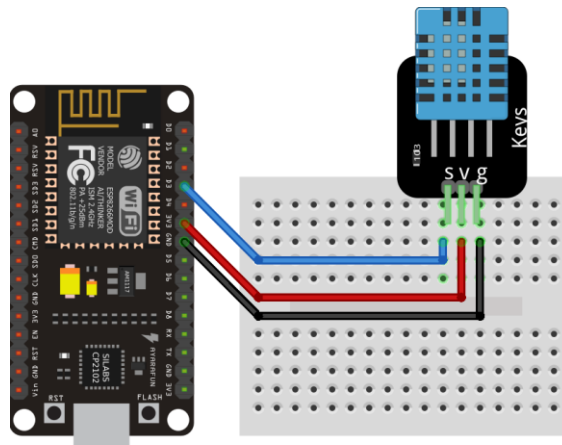
Λειτουργία	NodeMCU V3	Arduino Uno
Microcontroller	Tensilica 32-bit Xtensa LX106	ATmega328P
Flash Memory	4 MB	32 KB (0.5 KB is used for Bootloader)
SRAM	128 KB	2 KB
EEPROM	Not Available 512 bytes (Software)	1 KB
CLOCK Speed	80 MHz or 160 MHz	16 MHz
Operating Voltage	3.3V DC	5V DC
Input Voltage	7-12 DC	6V-20V DC
Current Consumption	15 μ A – 400 mA	45 mA – 80 mA
DC current per I/O Pin	12 mA	40 mA
Digital IO Pins	17	14
Analog Input Pins	1	6
UARTs	2	1
SPI	2	1
I2C	1	1
PWM	4	6
WiFi	Yes	No
Bluetooth	No	No



Εικόνα 6.3: Αριστερά το NodeMCU V3 και δεξιά το Arduino Uno (<https://www.circuito.io/>)

6.3 Η υλοποίηση του κυκλώματος της παρακολούθησης περιβάλλοντος με το NodeMCU

Φαίνεται ξεκάθαρα πως η υλοποίηση με το NodeMCU είναι πολύ πιο απλή. Το ενσωματωμένο τσιπ ESP8266 μειώνει τον όγκο των καλωδίων που χρησιμοποιούνται καθώς χρειάζονται μόνο 3 καλώδια για την υλοποίηση του κυκλώματος. Στην υλοποίηση με το Arduino UNO η καλωδίωση είναι πολύπλοκότερη γιατί χρειάζεται επιπλέον καλωδίωση και για το ESP8266 αφού δεν είναι ενσωματωμένο στην πλακέτα.



Εικόνα 6.4: Το ολοκληρωμένο κύκλωμα του NodeMCU V3 με τον αισθητήρα θερμοκρασίας και υγρασίας DHT11(υλοποιήθηκε με το πρόγραμμα Fritzing)

Στον ακροδέκτη D3 του NodeMCU συνδέεται με μπλε καλώδιο ο ακροδέκτης DATA του αισθητήρα DHT11. Ο ακροδέκτης 3V3 του NodeMCU τροφοδοτεί με ρεύμα τον αισθητήρα DHT11 μέσω ενός κόκκινου καλωδίου που συνδέεται με τον ακροδέκτης VCC του αισθητήρα. Το κύκλωμα κλείνει συνδέοντας τον ακροδέκτη GND του NodeMCU και τον ακροδέκτη GND του αισθητήρα μέσω του μαύρου καλωδίου.

Πίνακας καλωδιώσεων

NodeMCU V3	Χρώμα Καλωδίου	DHT11
D3	μπλέ	DATA
3V3	κόκκινο	VCC
GND	μαύρο	GND

6.4 Ο κώδικας που αναπτύχθηκε για το NodeMCU

Όπως και με το Arduino έτσι και με το NodeMCU ο κώδικας θα υλοποιηθεί μέσω του λογισμικού Arduino IDE. Οι εντολές μοιάζουν αλλά δεν είναι ίδιες. Το ενσωματωμένο wifi chip ESP8266 του NodeMCU δεν χρειάζεται το πλήθος των εντολών που απαιτήθηκαν στον προγραμματισμό του Arduino. Το λογισμικό του NodeMCU είναι φτιαγμένο με γνώμονα τη σύνδεση μέσω wifi , με αποτέλεσμα ο κώδικας να είναι απλούστερος στην υλοποίηση σε σχέση με το Arduino. Υπο το οποίο χρειάζεται ένα εξωτερικό εξάρτημα για να συνδεθεί στο διαδίκτυο.

```
nodenew

#include <DHT.h> // Including library for dht
#include <ESP8266WiFi.h>

String apiKey = "LKNIEY8BU9J1L2N8"; // edw einai to write api key tou thingspeak

const char *ssid = "KarrasHome"; // edw einai kwdikoi tou wifi
const char *pass = "2695031430a";
const char* server = "api.thingspeak.com";

#define DHTPIN 0 //pin pou syndeetai o dht11

DHT dht(DHTPIN, DHT11);

WiFiClient client;

void setup()
{
  Serial.begin(115200);
  delay(10);
  dht.begin();

  Serial.println("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, pass);
```

Εικόνα 6.5: Ο κώδικας που λειτουργεί το NodeMCU

```
while (WiFi.status() != WL_CONNECTED)
{
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

}

void loop()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t))
  {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  if (client.connect(server,80) // "184.106.153
  {
    String postStr = apiKey;
    postStr += "&field1=";
    postStr += String(t);
    postStr += "&field2=";
```

Εικόνα 6.6: Ο κώδικας που λειτουργεί το NodeMCU

```

        postStr += String(h);
        postStr += "\r\n\r\n";

        client.print("POST /update HTTP/1.1\n");
        client.print("Host: api.thingspeak.com\n");
        client.print("Connection: close\n");
        client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
        client.print("Content-Type: application/x-www-form-urlencoded\n");
        client.print("Content-Length: ");
        client.print(postStr.length());
        client.print("\n\n");
        client.print(postStr);

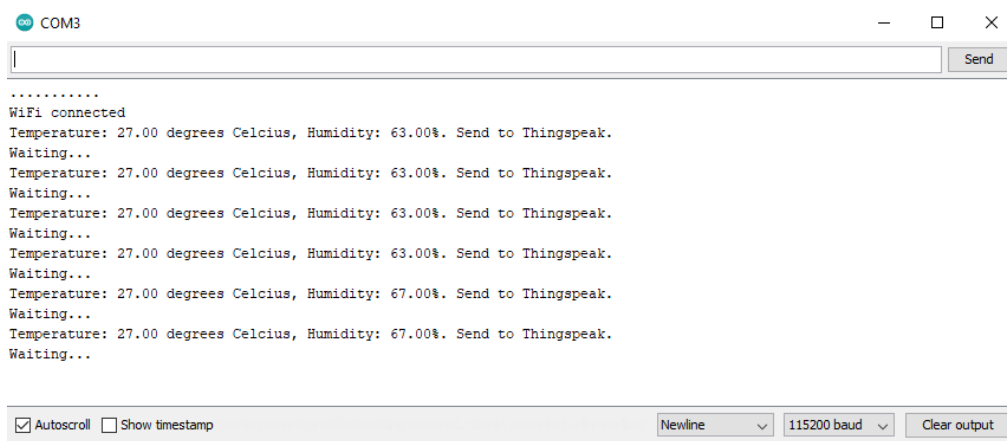
        Serial.print("Temperature: ");
        Serial.print(t);
        Serial.print(" degrees Celcius, Humidity: ");
        Serial.print(h);
        Serial.println("%. Send to Thingspeak.");
    }
    client.stop();

    Serial.println("Waiting...");

    //to thingspeak xreiazetai to ligotero 15 sec delay anamesa sthn apostolh twm dedomenwn
    delay(1000);
}

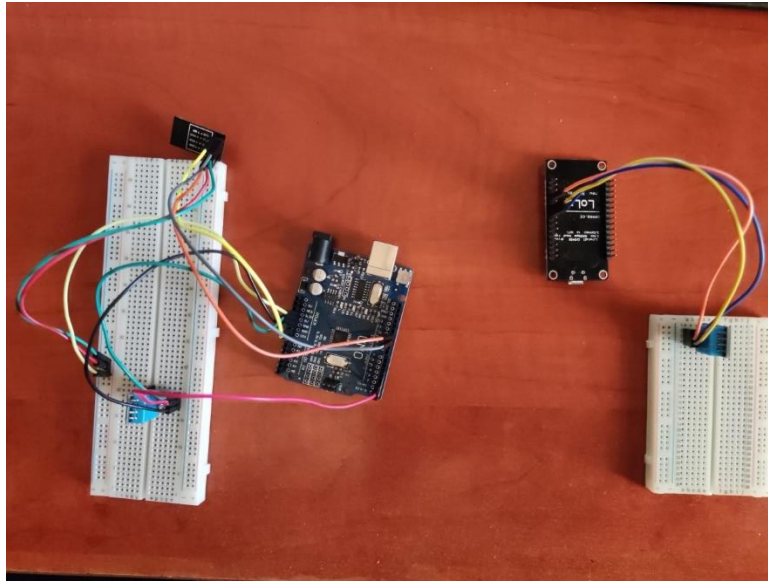
```

Εικόνα 6.7: Ο κώδικας που λειτουργεί το NodeMCU



Εικόνα 6.8: Το serial monitor του NodeMCU

Το NodeMCU στέλνει τα δεδομένα στο Thingspeak IOT όπως το Arduino. Στην παρακάτω φωτογραφία απεικονίζεται στα αριστερά το κύκλωμα με το Arduino Uno και στα δεξιά το κύκλωμα με το NODEMCU



Εικόνα 6.9: Αριστερά το ολοκληρωμένο κύκλωμα με το Arduino Uno και δεξιά το ολοκληρωμένο κύκλωμα με το NodeMCU V3

ΚΕΦΑΛΑΙΟ 7 Συμπεράσματα

Η πτυχιακή εργασία υλοποίησε την ανάπτυξη ενός σεναρίου ΙΟΤ για περιβαλλοντικό έλεγχο. Αυτό επιτεύχθηκε με τη χρήση του Arduino Uno και ενός αισθητήρα. Ανάλογα με τις ανάγκες και τις απαιτήσεις μας μπορούμε να δημιουργήσουμε και να προγραμματίσουμε το δικό μας αυτόνομο ηλεκτρονικό κύκλωμα.

Το Arduino είναι μια ευέλικτη πλατφόρμα η οποία απευθύνεται τόσο στον ερασιτέχνη προγραμματιστή ή ηλεκτρονικό όσο και σε κάποιον με προχωρημένες γνώσεις προγραμματισμού. Ο προγραμματισμός της πλακέτας είναι σχετικά εύκολος καθότι είναι μια πλατφόρμα ανοιχτού κώδικα. Το μόνο που χρειάζεται είναι βασικές γνώσεις προγραμματισμού και η γνώση της βιβλιοθήκης που θα χρησιμοποιήσει ο χρήστης στο Arduino IDE.

Οι δυνατότητες του Arduino σε σχέση με το κόστος απόκτησής του είναι απίστευτα καλές. Αυτό το κάνει προσβάσιμο σχεδόν σε όλους, επιτρέποντας στον χρήστη να πειραματιστεί και να υλοποιήσει εφαρμογές που θα του καλύψουν κάποια ανάγκη της καθημερινής του ζωής ή θα του προσφέρουν γνώση πάνω στην δομή των ηλεκτρονικών κυκλωμάτων και μικροελεγκτών

BIBΛΙΟΓΡΑΦΙΑ

1. M. Banzi, Getting Started with arduino. " O'Reilly Media, Inc.", 2009
2. A. M. Gibb, New media art, design, and the Arduino microcontroller: A malleable tool. PhD thesis, Pratt Institute, 2010
3. P. D. Minns, C Programming For the PC the MAC and the Arduino Microcontroller System. Author House, 2013
4. A. U. ARDUINO UNO, "Front. arduino uno board, " 2012
- 5 . J. Boxall, Arduino Workshop: A Hands-on Introduction with 65 Projects. No Starch Press, 2013
- 6 . Programming Arduino: Getting Started with Sketches, Second Edition 2nd Edition, Kindle Edition by Simon Monk
7. IoT Based Environmental Monitoring System using Arduino UNO and Thingspeak IJSTE - International Journal of Science Technology and Engineering
8. Design of an IoT based Real Time Environment Monitoring System using Legacy Sensors,Aparajita Das, Manash Pratim Sarma, Kandarpa Kumar Sarma and Nikos Mastorakis, October 2018
- 9.Water Quality Monitoring System Based on IOT ,Vaishnavi V. Daigavane and Dr. M.A Gaikwad, Advances in Wireless and Mobile Communications. ISSN 0973-6972 Volume 10, Number 5 (2017), pp. 1107-1116

ΔΙΑΔΙΚΤΥΑΚΕΣ ΠΗΓΕΣ

- 1) <https://create.arduino.cc>
- 2) <https://www.circuito.io>
- 3) <https://lastminuteengineers.com>
- 4) <https://roboindia.com>
- 5) <https://www.circuitbread.com>
- 6) <http://fabacademy.org>
- 7) <https://www.makeuseof.com>
- 8) <https://github.com>
- 9) <https://forum.fritzing.org>
- 10) [https://commons.wikimedia.org/wiki/File:Pinout of ARDUINO Board and ATMega328PU.svg](https://commons.wikimedia.org/wiki/File:Pinout_of_ARDUINO_Board_and_ATMega328PU.svg)
- 11) <https://www.researchgate.net>
- 12) <https://www.twinschip.com/>
- 13) <https://www.iotsworldcongress.com/>
- 14) <https://www.techtarget.com/>
- 15) <https://www.theengineeringprojects.com/>