



Σχεδίαση και κατασκευή μετεωρολογικού σταθμού και αποστολή δεδομένων σε πλατφόρμα IoT.



ΚΟΝΤΟΓΙΩΡΓΟΣ ΣΠΥΡΙΔΩΝ

16 Εξάμηνο αμ: 14035, email: kontogiorgos@hotmail.com

ΥΠΟΓΡΑΦΗ

X

Επιβλέπων Καθηγητής: Γλαβάς Ευρπίδης

Πίνακας περιεχομένων

1. Εισαγωγικές Έννοιες	6
1.1. Συστήματα μετρήσεων.....	6
1.2. Θερμοκρασία	6
1.3. Υγρασία	6
1.4. Τι είναι ένας Μετεωρολογικός σταθμός	7
2. Ενσωματωμένα συστήματα	7
2.1. Τα ενσωματωμένα συστήματα ως μέρος της καθημερινότητας.....	7
2.2. Ιδιαιτερότητες των ενσωματωμένων συστημάτων	8
2.3. Οι μικροελεγκτές	9
2.4. Το Arduino.....	9
2.5. Περιγραφή της πλακέτας Arduino	10
2.6. Integrated Development Environment (IDE).....	11
2.7. Η ηλεκτρονική πλατφόρμα NodeMCU 1.0 – 12E.....	14
2.7.1. Πλατφόρμες ανάπτυξης λογισμικού	16
3. Internet of Things	18
3.1. Ασύρματα Δίκτυα Αισθητήρων	18
3.2. Τι είναι το Internet of Things.....	18
3.3. Η πλατφόρμα ThingSpeak	20
4. Περιγραφή Εξαρτημάτων	24
4.1. NodeMCU 1.0 - 12E.....	24
4.2. Sensor DHT11 Temperature & Humidity.....	25
4.3. Sensor ds18B20 waterproof Temperature	26
4.4. Sensor rain drop module	27
4.5. Άλλα Εξαρτήματα.....	28
5. Υλοποίηση	29
5.1. Βιβλιοθήκες	29
5.2. Διάγραμμα Ροής.....	29
5.3. Συνδεσμολογία κυκλώματος.....	33
5.4. Λειτουργία συστήματος.....	34
5.5. Παρατηρήσεις γραφημάτων μετεωρολογικού σταθμού	37
6. Συμπεράσματα	39
Βιβλιογραφία	40
Παράρτημα Κώδικα.....	41
Παράρτημα εικόνας Μετεωρολογικού Σταθμού	45

Εικόνα 1 Μετεωρολογικός σταθμός στην Νάξο	7
Εικόνα 2 Τα μέρη από τα οποία αποτελείται ένα ενσωματωμένο σύστημα	9
Εικόνα 3 Περιγραφή της πλακέτας Arduino	11
Εικόνα 4 Arduino IDE.....	12
Εικόνα 5 Επιλογή της κατάλληλης πλακέτας.....	13
Εικόνα 6 Επιλογή του κατάλληλου Port	14
Εικόνα 7 ESP-12E.....	15
Εικόνα 8 Πλακέτα NodeMCU 1.0.....	15
Εικόνα 9 Εγκατάσταση ESP8266 package.....	16
Εικόνα 10 Επιλογή πλακέτας NodeMCU 1.0 12E	17
Εικόνα 11 Επιλογή θύρας.....	18
Εικόνα 12 Registration στην πλατφόρμα ThingSpeak	20
Εικόνα 13 Δημιουργία νέου καναλιού	21
Εικόνα 14 Επεξεργασία καναλιού	22
Εικόνα 15 Καρτέλα Γραφημάτων των μετρήσεων μας.....	23
Εικόνα 16 Κλειδί ταυτοποίησης API key.....	23
Εικόνα 17 Πλακέτα NodeMcu 1.0 ESP 12E	24
Εικόνα 18 DHT11	25
Εικόνα 19 Σύνδεση MCU - DHT11	25
Εικόνα 20 Πίνακας κατανάλωσης ενέργειας DHT11.....	26
Εικόνα 21 ds18B20 sensor	26
Εικόνα 22 Sensor Rain Drop Module.....	27
Εικόνα 23 Διάγραμμα Ροής setup ().....	30
Εικόνα 24 Διάγραμμα Ροής της συνάρτησης loop ()	32
Εικόνα 25 Συνδεσμολογία κυκλώματος.....	34
Εικόνα 26 Σύνδεση στο NodeMcu	35
Εικόνα 27 configure WiFi	35
Εικόνα 28 Καταχώρηση Διαπιστευτηρίων WiFi.....	36
Εικόνα 29 Συνδέθηκε στο δίκτυο που καταχωρήσαμε.....	36
Εικόνα 30 Γράφημα rain drop sensor.....	37
Εικόνα 31 Γράφημα ds18b20 waterproof temperature.....	38
Εικόνα 32 Γράφημα dht11 θερμοκρασίας.....	38
Εικόνα 33 Γράφημα dht11 υγρασίας.....	39
Εικόνα 34 Τελικό αποτέλεσμα του μετεωρολογικού σταθμού	45

ΔΗΛΩΣΗ ΠΝΕΥΜΑΤΙΚΗΣ ΙΔΙΟΚΤΗΣΙΑΣ

Η παρούσα εργασία αποτελεί προϊόν αποκλειστικά δικής μου προσπάθειας. Όλες οι πηγές που χρησιμοποιήθηκαν περιλαμβάνονται στη βιβλιογραφία και γίνεται ρητή αναφορά σε αυτές μέσα στο κείμενο όπου έχουν χρησιμοποιηθεί.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέπων καθηγητή μου κ. Γλαβά Ευρπίδη, για την βοήθεια, τη καθοδήγηση και την πολύ καλή συνεργασία που είχαμε καθ' όλο το χρονικό διάστημα εκπόνησης της διπλωματικής αυτής εργασίας. Οι συμβουλές του, οι οδηγίες του αλλά και οι κατευθύνσεις του συνέβαλαν στο μέγιστο για την επιτυχή κατά την άποψη μου ολοκλήρωση της εργασίας αυτής. Θα ήθελα επίσης να ευχαριστήσω όλους τους καθηγητές του τμήματος, για την πολύτιμη γνώση που μου μεταλαμπάδευσαν και τη βοήθεια που μου πρόσφεραν κάθε φορά που ζητήθηκε.

ΠΕΡΙΛΗΨΗ

Σε αυτή την εργασία θα παρουσιαστεί η μελέτη, ο σχεδιασμός και η κατασκευή ενός μετεωρολογικού σταθμού ο οποίος θα παράγει μετρήσεις θερμοκρασιών, υγρασίας και βροχής και θα τις μεταδίδει σε μία πλατφόρμα IoT, πιο συγκεκριμένα το ThingSpeak. Θα αναφερθούμε στους μικροελεγκτές όπως τον ATmega328P αλλά και τον ESP8266EX όπως και στις πλακέτες Arduino και NodeMCU 1.0 καθώς και στο λογισμικό με το οποίο θα προγραμματίσουμε τον μικροελεγκτή μας ESP8266EX, το Arduino IDE. Επίσης θα μιλήσουμε για τα δίκτυα αισθητήρων και το διαδίκτυο των πραγμάτων το οποίο ορίζεται από τους ερευνητές ως την φυσική εξέλιξη του διαδικτύου. Στη συνέχεια θα γίνει λεπτομερής αναφορά στους αισθητήρες και σε όλα τα εξαρτήματα που χρησιμοποιήθηκαν για την λειτουργία του συστήματος. Επιπλέον θα παρουσιαστεί η λειτουργία του συστήματος, τα διαγράμματα ροής, ο κώδικας αλλά και πολλές βασικές συναρτήσεις που χρησιμοποιήθηκαν για να υλοποιηθεί ο μετεωρολογικός σταθμός. Τέλος θα γίνουν παρατηρήσεις στα γραφήματα που συλλέξαμε από τις μετρήσεις των αισθητηρίων και θα αναφέρουμε κάποια συμπεράσματα και μελλοντικές επεκτάσεις που μπορούν να γίνουν για την εξέλιξη της εργασίας.

Λέξεις κλειδιά: Ενσωματωμένα συστήματα, Διαδίκτυο των πραγμάτων, Μικροελεγκτές, NodeMCU, Μετεωρολογικός σταθμός

1. Εισαγωγικές Έννοιες

1.1. Συστήματα Μετρήσεων

Ο άνθρωπος από το ξεκίνημα της γέννησης του και κατά την πορεία της ζωής του έχει την όρεξη να ψάχνει και να προσπαθεί να πληροφορηθεί όμως δεν αρκεί μόνο να λαμβάνει πληροφορίες αλλά θέλει να μπορεί και να τις επεξεργαστεί. Για το λόγο αυτό συλλέγει πληροφορίες από το περιβάλλον και τις επεξεργάζεται μέσω των μετρητικών του συστημάτων, τις αισθήσεις. Για παράδειγμα την όραση, την ακοή, την αφή, την όσφρηση και τη γεύση. Αφού ληφθούν οι πληροφορίες, καταλήγουν στον επεξεργαστή του ανθρώπου δηλαδή τον εγκέφαλο και μετά την επεξεργασία γίνεται η λήψη των αποφάσεων (για παράδειγμα επιλογή τροφής). Το βιολογικό σύστημα μέτρησης του ανθρώπου επιτρέπει την ολοκλήρωση πολλών απλών και πολύπλοκων εργασιών στην καθημερινή του ζωή. Διαθέτει ικανότητες όπως είναι η γρήγορη επεξεργασία των δεδομένων αλλά μαθαίνει να προσαρμόζεται και στην ανοχή λαθών. Έτσι λοιπόν σχεδιάστηκαν μετρητικά συστήματα τα οποία θα εργάζονται όπως ο άνθρωπος. Δηλαδή θα μπορούν να συλλέξουν πληροφορία, θα μπορούν να την επεξεργαστούν και να πάρουν κάποια απόφαση. Θα μπορούσαμε να πούμε ότι ένα μετρητικό σύστημα πληροφορίας σχεδιάζεται ως εξής. Αρχικά λαμβάνει το σήμα εισόδου που ονομάζεται μετρήσιμη ποσότητα η οποία είναι μία φυσική ή χημική ποσότητα (πχ. Μετατόπιση, πίεση, συγκέντρωση αερίου). Η ποσότητα αυτή ανιχνεύεται από τον αισθητήρα ή τον μεταλλάκτη. Ως αισθητήρας ορίζεται η ηλεκτρονική συσκευή η οποία λαμβάνει ερεθίσματα από το περιβάλλον και τα μετατρέπει σε ηλεκτρική τάση. Στη συνέχεια αυτό το σήμα επειδή είναι αναλογικό χρειάζεται μία αναλογική επεξεργασία. Ως επεξεργαστής ορίζεται η τροποποίηση του ηλεκτρικού σήματος που στέλνει ο αισθητήρας χωρίς να αλλάζει η ενέργεια του σήματος. Τέλος το σήμα από τον επεξεργαστή καταλήγει σε ένα σύστημα απεικόνισης για να λάβει ο χειριστής κάποιες πληροφορίες (πχ. LCD οθόνη) αλλιώς μπορεί να αποθηκευτεί (EPROM, RAM, μαγνητικός δίσκος, χαρτί) για περαιτέρω μελέτη. [1]

1.2. Θερμοκρασία

Η θερμοκρασία είναι το βασικό χαρακτηριστικό της ατμόσφαιρας και ο κύριος συντελεστής για τον καθορισμό της βλάστησης, του κλίματος και της ζωής σε ένα τόπο. Η θερμοκρασία μετριέται με το θερμόμετρο και η μονάδα μέτρησης της είναι οι βαθμοί Κελσίου C. Όμως η μετεωρολογία χρησιμοποιεί εξειδικευμένα θερμόμετρα που μετράνε την μέγιστη και την ελάχιστη θερμοκρασία από την τελευταία τους ρύθμιση. Επιπλέον αυτές οι μετρήσεις καταγράφονται συνεχώς σε ειδικούς θερμογράφους μέσω μιας βαθμονομημένης χάρτινης ταινίας. [2]

1.3. Υγρασία

Ο ατμοσφαιρικός αέρας περιέχει υδρατμούς και η ποσότητα του εξαρτάται από τον τόπο και την ώρα. Δεν μπορεί να έχει απεριόριστη ποσότητα υδρατμών όμως μπορεί να έχει μία μέγιστη δυνατή περιεκτικότητα υδρατμών ανάλογα την θερμοκρασία. Όταν ο αέρας φτάσει σε αυτή την μέγιστη τιμή ονομάζεται κορεσμένος. Όμως όσο πιο ψυχρός είναι ο αέρας τόσο λιγότερη ποσότητα υδρατμών μπορεί να κρατήσει. Αν μία μάζα υγρού και θερμού αέρα ψυχθεί οι υδρατμοί θα συμπυκνωθούν σε σταγονίδια και θα δημιουργήσουν το νέφος. Αντίθετα αν δεν συγκεντρωθούν σε ψυχρά σωματίδια θα δημιουργήσουν την δρόσο. Η θερμοκρασία στην οποία ο αέρας καθώς ψύχεται φτάνει στο κορεσμό ονομάζεται σημείο δρόσου. Όπως και η θερμοκρασία έτσι και η υγρασία είναι πολύ σημαντική για το κλίμα και την ζωή ενός τόπου για αυτό το λόγο ο άνθρωπος δημιούργησε υγρόμετρα για να μπορεί να καταγράφει την υγρασία στο περιβάλλον. Τα υγρόμετρα μετράνε την υγρασία και η μονάδα μέτρησης της είναι το ποσοστό (%). [2]

1.4. Τι είναι ένας Μετεωρολογικός σταθμός

Μετεωρολογικός σταθμός είναι ένα επίγειο σημείο στο οποίο πραγματοποιούνται μετεωρολογικές παρατηρήσεις. Είναι μία εγκατάσταση η οποία περιέχει μετεωρολογικά όργανα είτε μέσα σε μετεωρολογικό κλωβό είτε εκτός, επιστημονικό προσωπικό και η θέση αυτών των σταθμών ορίζεται από την κεντρική Εθνική Μετεωρολογική Υπηρεσία της κάθε χώρας ώστε όλοι οι σταθμοί να δημιουργήσουν ένα δίκτυο μετεωρολογίας για όλη τη χώρα. Κάθε σταθμός έχει μοναδικό αριθμό ταυτότητας ο οποίος είναι διεθνής και με αυτόν απεικονίζεται στους μετεωρολογικούς χάρτες. Στην Ελλάδα μετεωρολογικοί σταθμοί υπάρχουν στις μεγαλύτερες πόλεις, στα λιμάνια και στα αεροδρόμια. Οι μετρήσεις του κάθε σταθμού κωδικοποιούνται και στέλνονται στην κεντρική υπηρεσία στην οποία καταχωρούνται και μελετώνται για να εξάγουν συμπεράσματα για την πρόβλεψη του καιρού. [3]



Εικόνα 1 Μετεωρολογικός σταθμός στην Νάξο

2. Ενσωματωμένα συστήματα

2.1. Τα ενσωματωμένα συστήματα ως μέρος της καθημερινότητας

Όταν πρωτοεμφανίστηκαν τα ενσωματωμένα συστήματα χρησιμοποιούνταν ως συσκευές ή ως προϊόντα χαμηλών επιδόσεων και γενικά με της σημερινές επιδόσεις όπως τα ρολόγια, οι αριθμομηχανές κ.α. Σήμερα τα ενσωματωμένα συστήματα βρίσκονται παντού στη ζωή των ανθρώπων. Χρησιμοποιούνται σε πολύπλοκες συσκευές όπως φορητά τερματικά πολυμέσων, ψηφιακοί προσωπικοί βοηθοί κ.α. Έχουν γίνει μέρος της καθημερινότητας των περισσότερων ανθρώπων διευκολύνοντας έτσι την ζωή τους. Για αυτό το λόγο πολλοί επιστήμονες έχουν χαρακτηρίσει τα ενσωματωμένα συστήματα ως επανάσταση των ηλεκτρονικών συσκευών. Αυτή τη στιγμή κατασκευάζονται από μεγάλες εταιρίες εκατομμύρια υπολογιστές σε όλο το κόσμο που αυτό σημαίνει ότι κατασκευάζονται δισεκατομμύρια ενσωματωμένα συστήματα για να καλύψουν τις ανάγκες όλων αυτών των συσκευών. Μάλιστα έρευνες έχουν δείξει ότι στις τεχνολογικά αναπτυγμένες χώρες, αν μόνο το 40% της κοινωνίας διαθέτει προσωπικό υπολογιστή, κάθε σπίτι έχει τουλάχιστον 30 ενσωματωμένα συστήματα. Βέβαια η χρήση τους διαφέρει από τους υπολογιστές

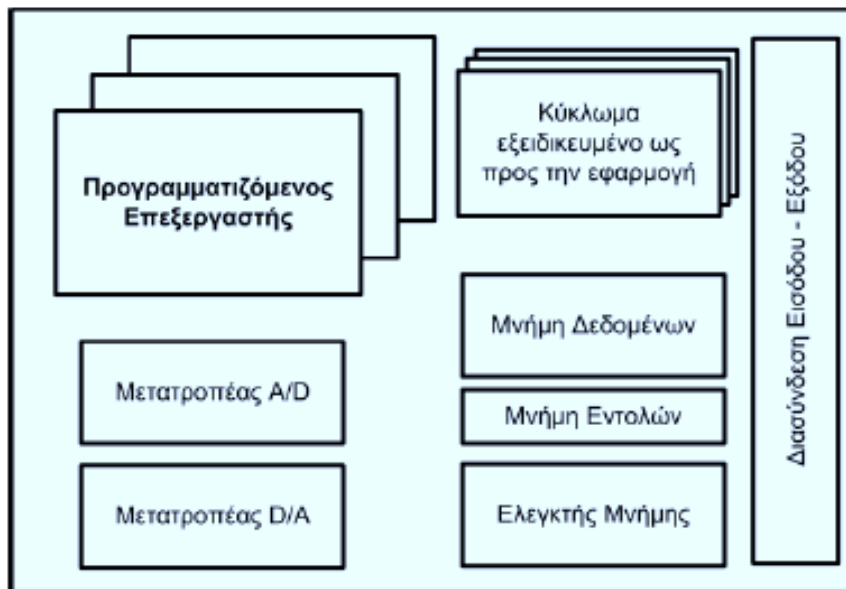
γενικού σκοπού. Τα ενσωματωμένα συστήματα είναι ειδικού σκοπού και χρησιμοποιούνται για πολύ συγκεκριμένες εφαρμογές και για περιορισμένη χρήση. Για αυτό το λόγο το ενσωματωμένο σύστημα ορίζεται ως το ψηφιακό σύστημα που εκτελεί κάποια λειτουργία και δεν είναι προσωπικός υπολογιστής, φορητός υπολογιστής ή κεντρικός διακομιστής. Κάποιες συσκευές που περιέχουν τέτοια συστήματα, παρουσιάζονται παρακάτω:

- Τηλεοράσεις και βίντεο: Οι τηλεοράσεις περιέχουν ενσωματωμένα συστήματα για την επεξεργασία των δεδομένων, την αποστολή των μηνυμάτων στην οθόνη, την ρύθμιση των καναλιών και την τροφοδοσία των κυκλωμάτων με ρεύμα.
- Κινητά / ασύρματα τηλέφωνα: Τα κινητά περιέχουν ενσωματωμένα συστήματα για την αποκωδικοποίηση και κωδικοποίηση της φωνής για την λειτουργία της συσκευής αλλά και για την εκτέλεση των εφαρμογών της.
- Οχήματα: Τα σύγχρονα αυτοκίνητα διαθέτουν πλήθος από ενσωματωμένα συστήματα για την προστασία των επιβατών, για τις βοηθητικές ενδείξεις και για βελτίωση της οδικής συμπεριφοράς του οχήματος.
- Συσκευές νοικοκυριού: Όλες οι σύγχρονες οικιακές συσκευές περιέχουν στο εσωτερικό τους ψηφιακά ενσωματωμένα συστήματα για τον έλεγχο και την λειτουργία των συσκευών. Τέτοιες συσκευές είναι το πλυντήριο, το ψυγείο, η ηλεκτρική σκούπα.
- Εκτυπωτές / φαξ / φωτογραφικές μηχανές: Χωρίς τα ενσωματωμένα συστήματα θα ήταν αδύνατη η εκτύπωση εγγράφων, η αποστολή δεδομένων για το φαξ αλλά και πολλά χαρακτηριστικά που βοηθούν τις φωτογραφικές μηχανές όπως είναι το ζουμ ή η συλλογή των φωτογραφιών στην μνήμη της φωτογραφικής μηχανής. [4]

2.2. Ιδιαιτερότητες των ενσωματωμένων συστημάτων

Κάποιες από τις ιδιαιτερότητες των ενσωματωμένων συστημάτων που τα κάνουν τόσο δημοφιλή είναι η κατασκευή από υλικό υψηλών επιδόσεων σε συνδυασμό με ένα εξειδικευμένο λογισμικό. Και τα δύο μαζί έχουν μικρό χρόνο σχεδιασμού και εισόδου στην αγορά (Short time to market). Προηγουμένως αναλύθηκαν κάποια παραδείγματα συσκευών που περιέχουν ενσωματωμένα συστήματα. Όμως επικρατεί ένα χάσμα στο πως ορίζονται οι συσκευές γενικού σκοπού και συσκευές ειδικού σκοπού. Για παράδειγμα ένας υπολογιστής που εκτελεί μόνο ένα πρόγραμμα, είναι ενσωματωμένο σύστημα; Για αυτό το λόγο ένας πιο αυστηρός ορισμός αναφέρει ότι τα ενσωματωμένα συστήματα εμπεριέχονται μέσα στους υπολογιστές και χρησιμοποιούνται για την εκτέλεση μίας λειτουργίας. Μάλιστα βρίσκονται τοποθετημένοι στην ίδια επιφάνεια με τους πυρήνες των επεξεργαστών, μνημών, στα περιφερειακά όπως GPIO, μετατροπείς αναλογικών/ψηφιακών κυκλωμάτων καταλήγοντας σε συστήματα πάνω στο υλικό πυριτίου (System on Chip, SoC). Μία δεύτερη ιδιαιτερότητα είναι ότι έχουν αυξημένους περιορισμούς. Δηλαδή σχεδιάζονται για να εκτελούν αδιαλείπτως μία λειτουργία, δεν μπορούν να χρησιμοποιηθούν για να εκτελέσουν διαφορετικές λειτουργίες από αυτές που έχουν σχεδιαστεί. Για παράδειγμα ένας επεξεργαστής πλυντηρίου δεν μπορεί να χρησιμοποιηθεί για αναπαραγωγή πολυμέσων. Σημαντικά χαρακτηριστικά των ενσωματωμένων συστημάτων είναι ότι πρέπει να είναι πολύ μικρά σε μέγεθος αφού τοποθετούνται σε φορητές συσκευές και να έχουν όσο το δυνατόν μικρότερο κόστος ώστε να προτιμώνται από τους καταναλωτές, επιπλέον να έχουν μικρή κατανάλωση ενέργειας, να έχουν υψηλές επιδόσεις και να είναι αξιόπιστα. Τα ενσωματωμένα συστήματα σε σχέση με τους υπολογιστές έχουν δικά τους λειτουργικά συστήματα πραγματικού χρόνου (RTOS) τα οποία διαφέρουν από τα λειτουργικά συστήματα (windows, linux) διότι είναι πολύ μικρά σε μέγεθος (μόλις κάποια KB) και βρίσκονται στην μνήμη ROM σε σχέση με τα παραπάνω λειτουργικά συστήματα που είναι αποθηκευμένα σε κάποιο σκληρό δίσκο. Επίσης είναι μη φιλικά προς τον χρήστη ή τον προγραμματιστή διότι δεν μπορούν να τροποποιηθούν. Τέλος στην παρακάτω εικόνα διακρίνεται ένα ενσωματωμένο σύστημα και τα μέρη τα οποία αποτελείται. Ένα ενσωματωμένο σύστημα συνήθως

αποτελείται από επεξεργαστή, μετατροπείς σήματος αναλογικού σε ψηφιακού και το αντίθετο, από μνήμες εντολών και δεδομένων, από τον ελεγκτή μνήμης, από εξειδικευμένα κύκλωμα και τέλος τις διασυνδέσεις εισόδου – εξόδου. Τα πιο σημαντικά δομοστοιχεία που περιέχει είναι ο επεξεργαστής και η μνήμη επειδή ο σχεδιασμός και η χρήση αυτών επηρεάζει την απόδοση αλλά και την κατανάλωση του ενσωματωμένου. [4]



Εικόνα 2 Τα μέρη από τα οποία αποτελείται ένα ενσωματωμένο σύστημα

2.3. Οι Μικροελεγκτές

Ένας μικροεπεξεργαστής είναι η καρδιά του υπολογιστή και η βασική του αρμοδιότητα είναι η εκτέλεση των εντολών. Οι μικροεπεξεργαστές δεν λειτουργούν αυτόνομα, στηρίζονται στα υπόλοιπα περιφερειακά και υποσυστήματα του υπολογιστή όπως η μητρική πλακέτα, η μνήμη, ο σκληρός δίσκος, η οθόνη, πληκτρολόγιο, ποντίκι κ.α. αντίθετα ο μικροελεγκτής περιέχει τον μικροεπεξεργαστή και όλα τα περιφερειακά σε ένα ολοκληρωμένο κύκλωμα. Βασικό χαρακτηριστικό των μικροελεγκτών είναι ότι μπορούν να αλληλοεπιδρούν με τον φυσικό κόσμο ενώ ένας υπολογιστής δεν μπορεί. Μπορούν να παίρνουν πληροφορίες από τα ερεθίσματα που λαμβάνουν οι αισθητήρες και να ορίζουν πως θα λειτουργήσουν. Μπορούν να ελέγξουν leds, κινητήρες κ.α. Ένα τέτοιο παράδειγμα είναι το κλιματιστικό το οποίο μέσω του χειριστηρίου ενεργοποιείται και ο χρήστης μπορεί να ορίσει την επιθυμητή θερμοκρασία. Στη συνέχεια ο αισθητήρας που βρίσκεται μέσα στο κλιματιστικό ελέγχει συνεχώς την θερμοκρασία του χώρου και με βάση την τιμή της ενεργοποιεί ή απενεργοποιεί τα μηχανολογικά μέρη του κλιματιστικού. Οι μικροελεγκτές στην σημερινή εποχή χρησιμοποιούνται ευρέως σε εφαρμογές: στον βιομηχανικό έλεγχο, σε δίκτυα αισθητήρων για περιβαλλοντολογικούς ελέγχους, σε συστήματα αυτόματων συναλλαγών (πχ. ATM τράπεζες), σε συστήματα μετρήσεων, έξυπνα σπίτια, παιχνιδιομηχανές κ.α. [5]

2.4. Το Arduino

Το Arduino δημιουργήθηκε από έναν φοιτητή στην πόλη της Ιβρέα στην Ιταλία το 2003. Το Arduino πήρε το όνομα του από τον στρατιωτικό διοικητή της Ιβρέα και αργότερα βασιλιά της Ιταλίας Arduin. Ο φοιτητής στα πλαίσια μιας εργασίας ανέπτυξε το Wiring project. Αυτό το πρότζεκτ είχε ως σκοπό την υλοποίηση ενός απλού συστήματος με μικροελεγκτή που να προσφέρεται για εύκολη ανάπτυξη εφαρμογών. Το Arduino που ήταν μία επέκταση του Wiring project χρησιμοποιήθηκε και

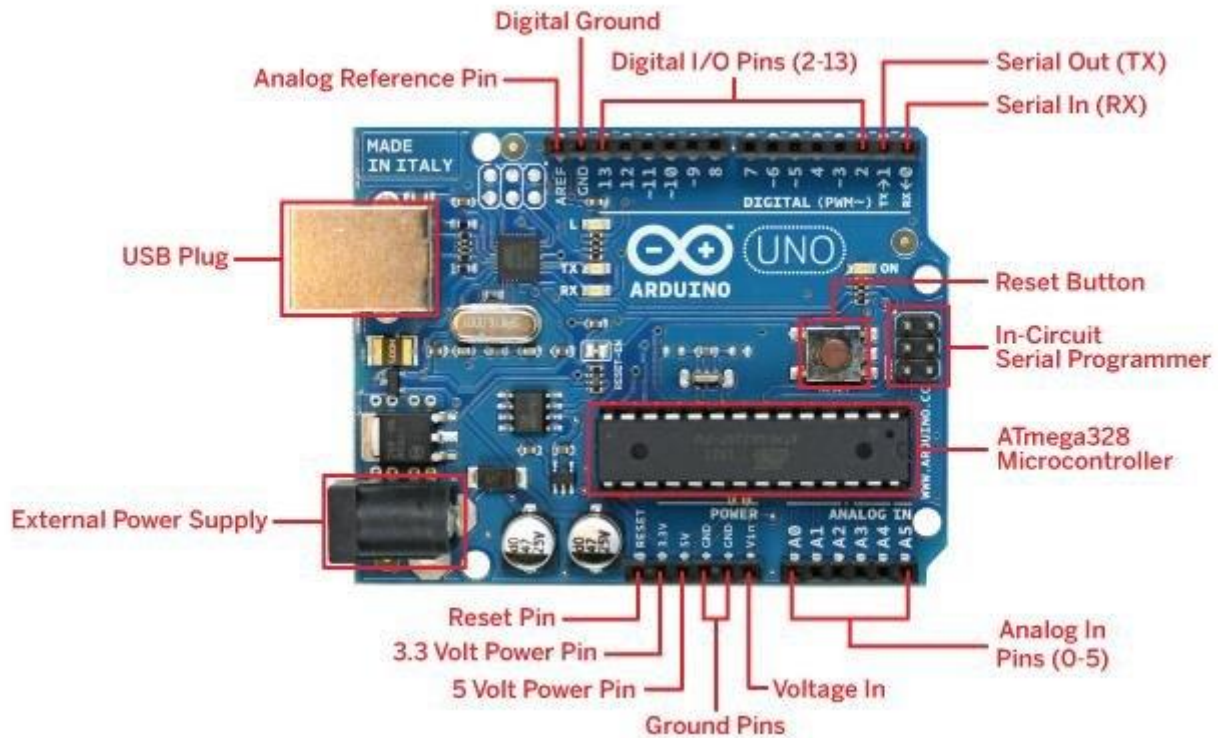
έγινε γνωστό για φοιτητές και μηχανικούς που δεν είχαν μεγάλη εμπειρία σε αυτά τα συστήματα. Το Arduino είναι μια πλατφόρμα ανάπτυξης έργων η οποία περιλαμβάνει την πλακέτα για τον μικροελεγκτή (βασίζεται σε μικροελεγκτές AVR της Atmel) και ένα περιβάλλον ανάπτυξης κώδικα τα οποία είναι ανοιχτού κώδικα και το λογισμικό διατίθεται δωρεάν. Η γλώσσα προγραμματισμού είναι η Wiring η οποία είναι μία παραλλαγή της C++ ενώ μέσα από μια μεγάλη σειρά βιβλιοθηκών μπορούν να αυξηθούν οι δυνατότητες και οι λειτουργίες του προγραμματισμού. Η πιο mainstream σειρά του Arduino είναι η ηλεκτρονική πλατφόρμα Arduino UNO. Τα βασικά πλεονεκτήματα της πλακέτας είναι:

- Χαμηλό κόστος: Η original πλακέτα κοστίζει περίπου 20 ευρώ, υπάρχουν όμως κλώνοι με τις ίδιες λειτουργίες που το κόστος τους δεν ξεπερνά τα 10 ευρώ.
- Συμβατότητα: Μπορεί να λειτουργήσει σε κάθε λειτουργικό σύστημα (Windows, Linux, Mac).
- Απλότητα: Εύκολο και απλό περιβάλλον ανάπτυξης κώδικα για αρχάριους χρήστες.
- Επεκτάσιμο λογισμικό ανοιχτού κώδικα: Λόγω του ότι τα προγράμματα και το λογισμικό είναι ανοιχτού κώδικα, υπάρχουν στο διαδίκτυο πληθώρα από tutorials και υπάρχει μεγάλη κοινότητα που μπορεί να κατευθύνει κάποιον αρχάριο χρήστη για την δημιουργία των δικών του προγραμμάτων. [5]

2.5. Περιγραφή της πλακέτας Arduino

Η καρδιά της πλακέτας Arduino είναι ο μικροελεγκτής AVR ATmega328 είναι ένας 8-μπιτος ελεγκτής που λειτουργεί σε τάση 5V και παρέχει στην πλακέτα 6 αναλογικές εισόδους και 14 ψηφιακές εξόδους. Η πλακέτα συνδέεται μέσω USB καλωδίου με τον υπολογιστή. Ο χρήστης από τον υπολογιστή μπορεί να προγραμματίσει και να μεταφορτώσει τον κώδικα στον μικροελεγκτή μέσω του καλωδίου usb. Άραξ και γίνει η μεταφόρτωση το καλώδιο usb δεν είναι απαραίτητο παρά μόνο για να τροφοδοτήσει την πλακέτα. Ωστόσο αν χρησιμοποιήσουμε διαφορετική πηγή, όπως μπαταρία η πλακέτα μπορεί να λειτουργήσει αυτόνομα. Τα βασικά στοιχεία που απαρτίζουν την πλακέτα είναι τα εξής:

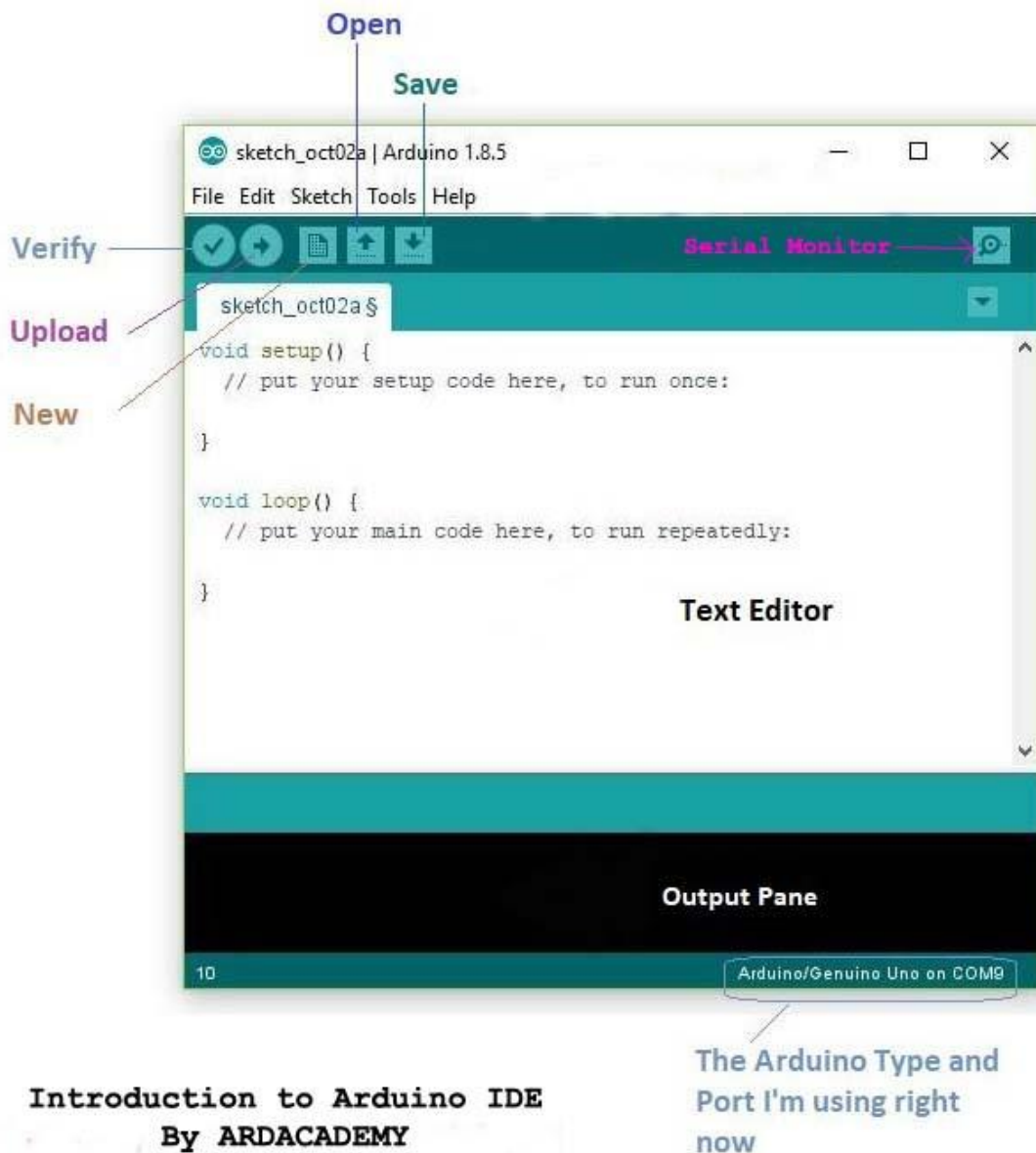
- ATmega328: όπως προαναφέρθηκε είναι ο μικροελεγκτής που διαθέτει η πλακέτα Arduino.
- Ψηφιακές θύρες εισόδου / εξόδου: Είναι τοποθετημένα στη σειρά στο πάνω μέρος της πλακέτας όπως διακρίνεται στην εικόνα. Η αρίθμηση τους ξεκινάει από το 0 έως το 13.
- Αναλογικές θύρες εισόδου: Βρίσκονται στο κάτω μέρος της πλακέτας όπως διακρίνεται στην εικόνα και λειτουργούν μόνο ως εισοδοί. Η αρίθμηση τους ξεκινάει από το 0 έως το 6.
- Θύρα USB: Χρησιμοποιείται για την αποστολή δεδομένων, την μεταφόρτωση του κώδικα αλλά και την τροφοδοσία όλης της πλακέτας.
- Εξωτερική τροφοδοσία: Εκτός από την θύρα usb η πλακέτα περιέχει και μία θύρα barrel power jack όπου μπορεί να τροφοδοτήσει την πλακέτα μας από 5 έως 12 V. Αν δεχτεί τάση μεγαλύτερη των 5V η πλακέτα θα συνεχίσει να λειτουργεί κανονικά διότι υπάρχει κύκλωμα που σταθεροποιεί την τάση.
- Κουμπί reset: Στην επιφάνεια της πλακέτας υπάρχει κουμπί reset. Χρησιμοποιείται για την επανεκκίνηση του προγράμματος.
- Ακροδέκτες ISP: Από αυτούς τους ακροδέκτες μπορεί να προγραμματιστεί ο μικροελεγκτής μέσω ISP.
- Leds: Στην επιφάνεια της πλακέτας υπάρχουν led Tx & Rx, led εκκίνησης της πλακέτας καθώς και ένα ενσωματωμένο led που είναι συνδεδεμένο με το ψηφιακό pin 13 και υπάρχει στην πλακέτα για δοκιμή, αν δεν υπάρχει πρόσθετος εξοπλισμός από τον χρήστη.
- Τάσεις: Η πλακέτα διαθέτει pin τάσης 3.3 V, 5V και pin Vin όπου μας δίνεται η δυνατότητα να τροφοδοτήσουμε το σύστημα μας με τάση διαφορετική από τα υπόλοιπα pin τάσης που υπάρχουν. [6]



Εικόνα 3 Περιγραφή της πλακέτας Arduino

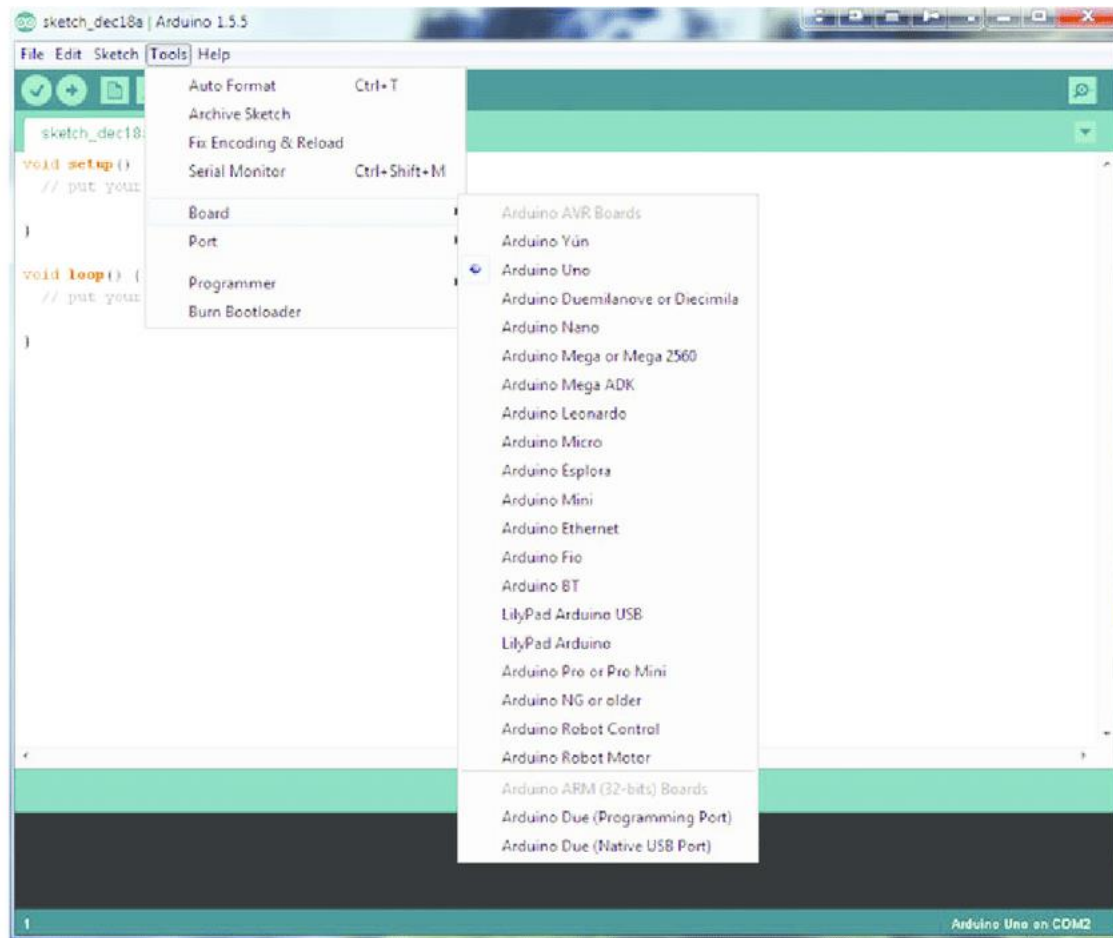
2.6. Integrated Development Environment (IDE)

Το Integrated Development Environment (IDE) είναι το ολοκληρωμένο περιβάλλον ανάπτυξης του Arduino. Είναι μια απλή εφαρμογή η οποία έχει μικρό μέγεθος (περίπου 180 MB) και λειτουργεί σε όλα τα λειτουργικά συστήματα (Windows, Linux, Mac). Η ροή του λογισμικού είναι η εξής, ο χρήστης γράφει τον κώδικα που στη συνέχεια γίνεται compile για τον έλεγχο συντακτικών λαθών, γίνεται εκτελέσιμο και μεταφορτώνεται στον μικροελεγκτή. Πιο αναλυτικά, κατά την εκκίνηση του IDE στο παράθυρο που θα ανοίξει, το μεγαλύτερο μέρος καταλαμβάνει ο editor δηλαδή το μέρος στο οποίο γράφουμε τον κώδικα. Στο κάτω μέρος του software υπάρχει ένα μαύρο πλαίσιο όπου εμφανίζονται τα μηνύματα ανάλογα με την πορεία της αποσφαλμάτωσης και της μεταφόρτωσης ενώ στο πάνω μέρος υπάρχει το κυρίως μενού και από κάτω τα κουμπιά επικύρωση, ανέβασμα, τα βοηθητικά κουμπιά ανοίγματος νέου πρότζεκτ, αποθήκευσης και το κουμπί της παρακολούθησης του serial monitor.

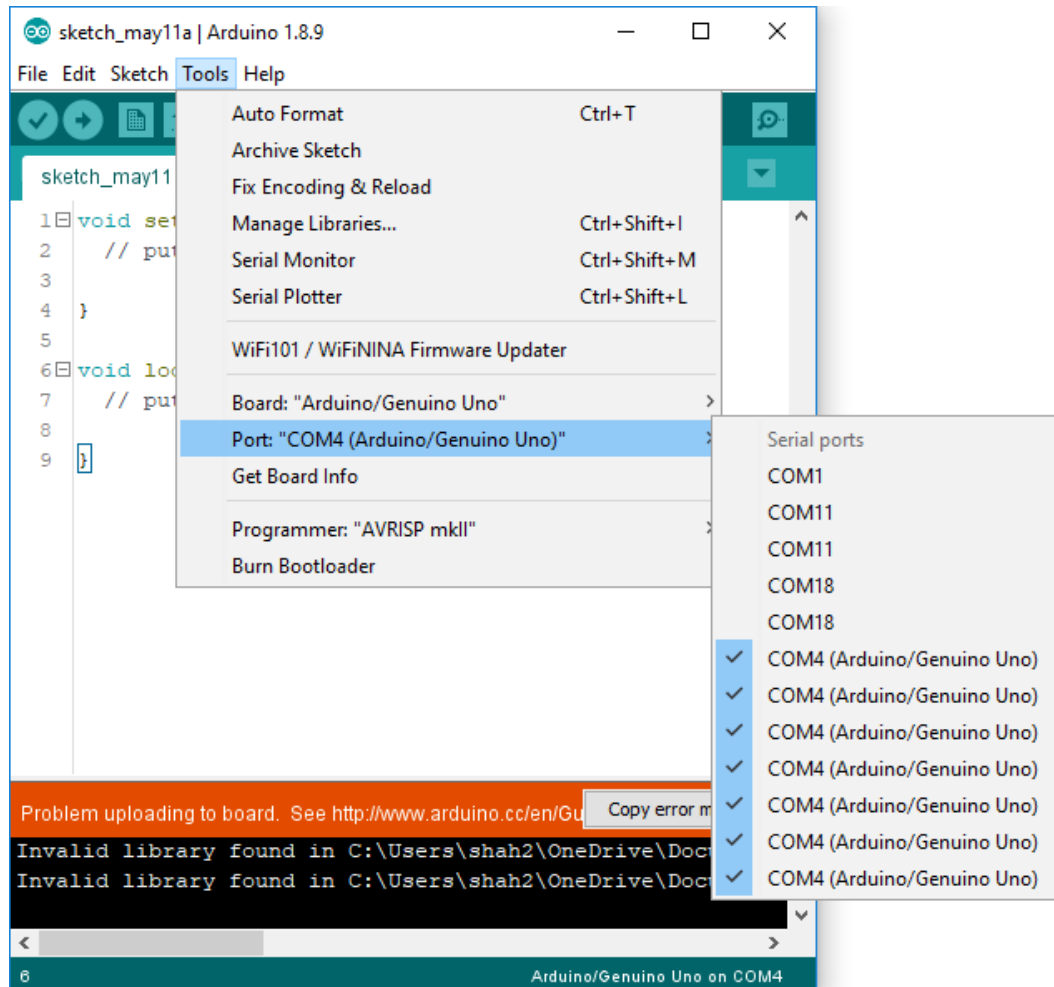


Εικόνα 4 Arduino IDE

Πριν γράψουμε κώδικα όμως θα πρέπει να έχουμε συνδέσει την πλακέτα με τον υπολογιστή μέσω του usb καλωδίου. Επόμενο βήμα είναι να επιλέξουμε την συσκευή που διαθέτουμε, στο software. Για παράδειγμα αν έχουμε στην κατοχή μας την πλακέτα Arduino UNO R3 τότε επιλέγουμε tools > board > Arduino UNO (εικόνα 4) στη συνέχεια πρέπει να επιλέξουμε την θύρα επικοινωνίας η οποία βρίσκεται στην καρτέλα tools > serialPort (εικόνα 5). Αν το λειτουργικό σύστημα που χρησιμοποιούμε είναι Windows θα πρέπει να δούμε μια εικονική θύρα COM ενώ αν είναι Linux θα πρέπει να δούμε tty/ACM. Αφού ολοκληρώσουμε την επιλογή της συσκευής και της πόρτας το μόνο που παραμένει να κάνουμε είναι να φορτώσουμε τον κώδικα στον μικροελεγκτή πατώντας το κουμπί verify. Αν δεν υπάρχουν σφάλματα στον κώδικα μας στη συνέχεια πατάμε το κουμπί upload. [6]



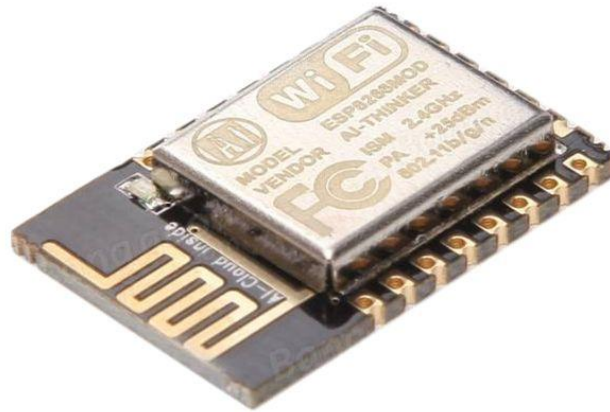
Εικόνα 5 Επιλογή της κατάλληλης πλακέτας



Εικόνα 6 Επιλογή του κατάλληλου Port

2.7. Η ηλεκτρονική πλατφόρμα NodeMCU 1.0 – 12E

Το NodeMCU 1.0 είναι μια ηλεκτρονική πλατφόρμα που κατασκευάζεται από την εταιρία Espressif Systems. Είναι ιδανικό για έργα IoT (Internet of Things) λόγω του ότι συνδυάζει μικροελεγκτή και πρωτόκολλα TCP / IP μαζί. Το τσιπ που βρίσκεται πάνω στην πλακέτα ονομάζεται ESP-12EX και περιέχει το τσιπ ESP8266EX και τον μικροεπεξεργαστή Tensilica L106 32-bit RISC που λειτουργεί με ταχύτητα ρολογιού από 80 έως 160 MHz και υποστηρίζει RTOS. Επίσης έχει 128 KB ram και 4 MB flash μνήμη για την αποθήκευση προγραμμάτων και δεδομένων και ο πομποδέκτης WiFi υποστηρίζει τα πρότυπα 802.11b / g / n.

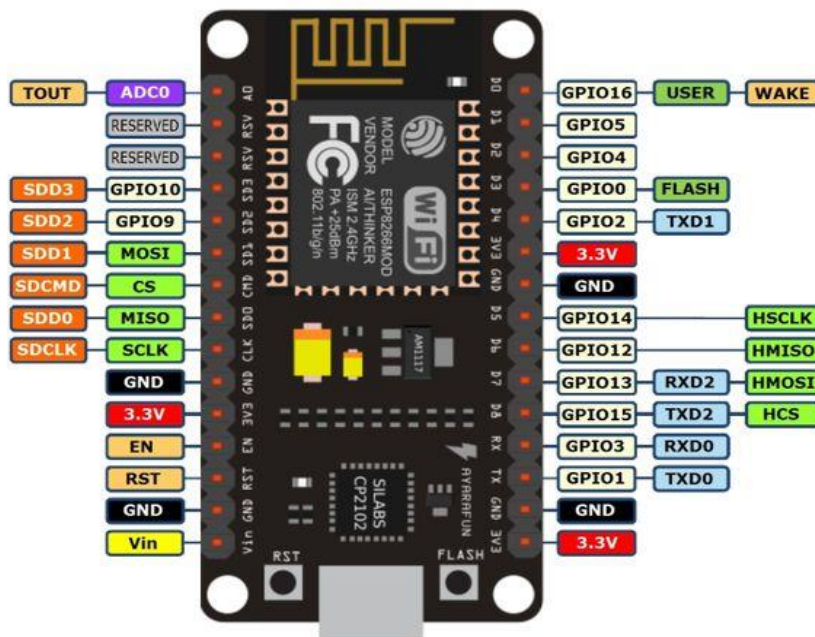


Εικόνα 7 ESP-12E

Τα περιφερειακά που περιέχει το τσιπ ESP8266EX είναι:

- 17 ακροδέκτες GPIO.
- Υποστηρίζει τα πρωτόκολλα SPI, I2C, I2S, UART, SDIO.
- Έξοδοι PWM για τον έλεγχο led και κινητήρων.
- ADC channel, για μετατροπή αναλογικού σε ψηφιακό σήμα με ανάλυση 10 bit.
- Η τάση λειτουργίας είναι 2.5 - 3.3 Volt με ρυθμιστή τάσης LDO για να διατηρείται σταθερή η τάση στα 3.3 Volt.
- Operating current: 80 mA.
- Operating Temperature range: -40 έως 125 C.
- Security WPA / WPA2.
- Encryption WEP / TKIP / AES.
- Network protocols IPV4, TCP / UDP / HTTP.

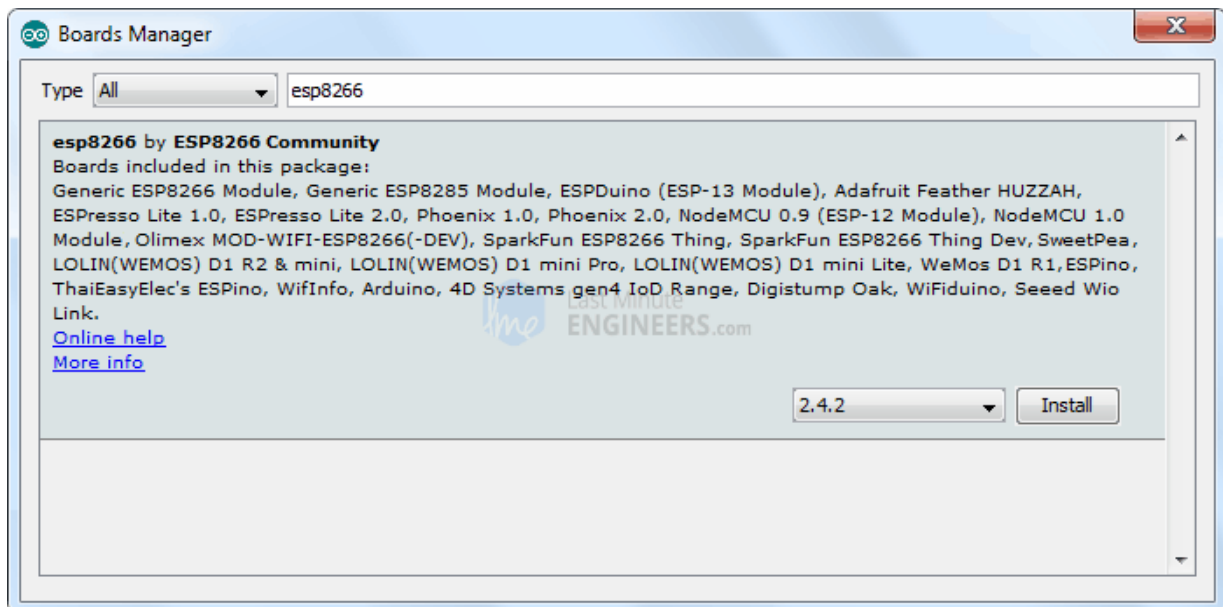
[7]



Εικόνα 8 Πλακέτα NodeMCU 1.0

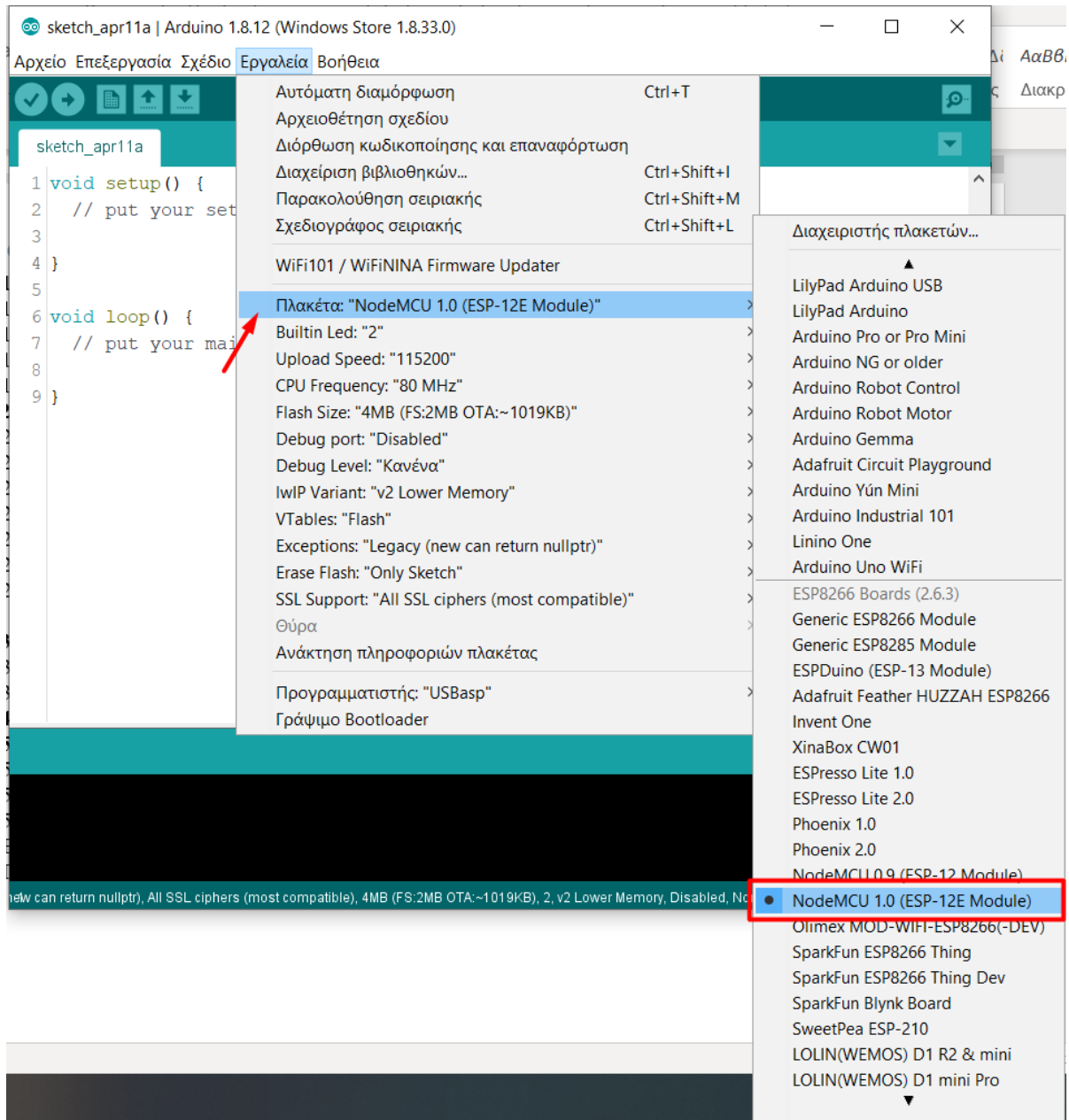
2.7.1. Πλατφόρμες ανάπτυξης λογισμικού

Υπάρχουν διάφορες πλατφόρμες ανάπτυξης κώδικα για τον μικροελεγκτή ESP8266. Κάποια από αυτά είναι το Espruino – JavaScript SDK, το Mongoose OS (το οποίο συνίσταται από την εταιρία Espressif), το Visual Studio μέσω του framework platform IO, το Arduino IDE κ.α. Ο προγραμματισμός της παρούσας εργασίας υλοποιήθηκε με το λογισμικό Arduino IDE. Για να προγραμματίσουμε και να αλληλοεπιδράσουμε με τον μικροελεγκτή πρέπει να εγκαταστήσουμε το Arduino IDE και συγκεκριμένα την έκδοση 1.6.4 ή κάποια νεότερη. Στη συνέχεια πρέπει να συμπεριλάβουμε το package του ESP8266 από το Boards Manager και να το εγκαταστήσουμε. Μόλις ολοκληρωθεί η εγκατάσταση ο μικροελεγκτής μπορεί να χρησιμοποιηθεί.

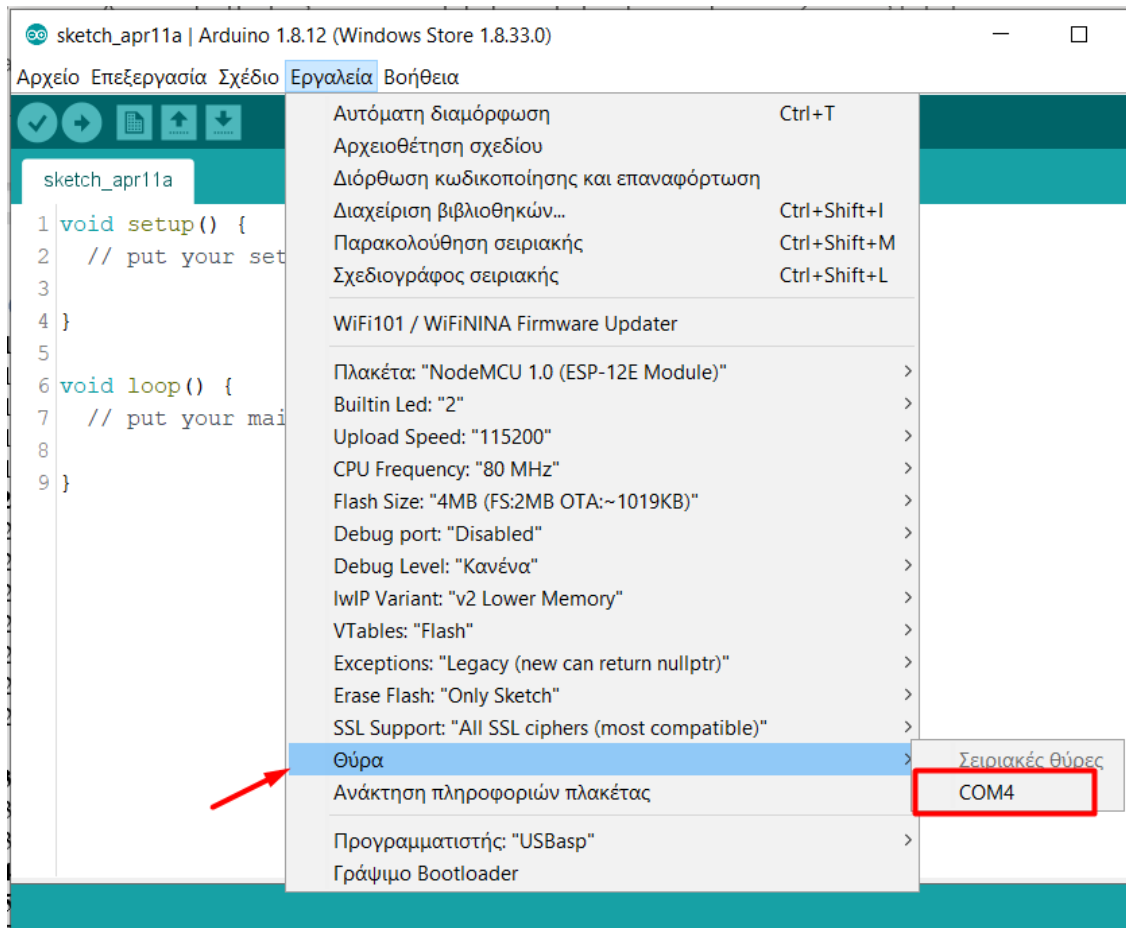


Εικόνα 9 Εγκατάσταση ESP8266 package

Στη συνέχεια αφού έχουμε εγκαταστήσει το esp8266 package θα εκτελέσουμε την ίδια διαδικασία όπως με το Arduino UNO για να επιλέξουμε αυτή τη φορά την πλακέτα NodeMCU 1.0 και το κατάλληλο port. Στην παρακάτω εικόνα φαίνεται η διαδικασία αυτή. Από το κυρίως μενού επιλέγουμε την καρτέλα tools > boards > NodeMCU 1.0 12E και στην ίδια καρτέλα επιλέγουμε την εικονική θύρα του υπολογιστή μας, tools > port > com. [7]



Εικόνα 10 Επιλογή πλακέτας NodeMCU 1.0 12E



Εικόνα 11 Επιλογή θύρας

3. Internet of Things

3.1. Ασύρματα Δίκτυα Αισθητήρων

Τα ασύρματα δίκτυα αισθητήρων (Wireless Sensor Network – WSN) αποτελούνται από συσκευές που περιλαμβάνουν έναν ή πολλούς αισθητήρες τα οποία έχουν ως σκοπό να λαμβάνουν ερεθίσματα, να παρακολουθούν το περιβάλλον, να συλλέγουν τα δεδομένα και να τα μεταδίδουν απομακρυσμένα. Κύριο χαρακτηριστικό αυτών των συσκευών είναι η χαμηλή επεξεργαστική ισχύ και η πολύ χαμηλή κατανάλωση ενέργειας. Τα ασύρματα δίκτυα αισθητήρων χωρίζονται σε δύο κατηγορίες, τα αδόμητα και τα δομημένα δίκτυα. Τα αδόμητα δίκτυα περιλαμβάνουν πολλούς αισθητήρες οι οποίοι συνδέονται μεταξύ τους και συνήθως περιέχουν πρωτόκολλα δρομολόγησης πληροφορίας και πρωτόκολλα για την βέλτιστη διαχείριση της ενέργειας ενώ τα δομημένα δίκτυα αποτελούνται από αισθητήρες οι οποίοι συνδέονται και αλληλοεπιδρούν με έναν κεντρικό σταθμό. Μία σημαντική ιδιαιτερότητα των WSN που έχει δημιουργήσει ερευνητικές προκλήσεις είναι η ασφάλεια των πληροφοριών που μεταδίδονται από τα δίκτυα αισθητήρων διότι οι μηχανικοί πρέπει να λάβουν υπόψιν την μικρή επεξεργαστική ισχύ παράλληλα με την χαμηλή κατανάλωση ενέργειας. [8]

3.2. Τι είναι το Internet of Things

Το διαδίκτυο των πραγμάτων (IoT) αναφέρθηκε για πρώτη φορά σε ομιλία του από τον Kevin Ashton το 1999. Ο Ashton πίστευε ότι οι συσκευές και τα αντικείμενα αν γίνουν έξυπνα με κάποιο τρόπο θα μπορέσουν να λειτουργήσουν με μεγαλύτερη προσοχή, ακρίβεια και πολύ μικρή έως

μηδαμινή ανοχή στο λάθος σε σχέση με τον άνθρωπο. Θεωρούσε ότι αν υπήρχαν υπολογιστές που να γνωρίζουν τα πάντα για τα αντικείμενα τους θα μπορούσαμε να παρακολουθήσουμε και να μετρήσουμε τα πάντα, θα γνωρίζαμε τότε χρειάζονται συντήρηση ή κάποια επισκευή. Στην ουσία αυτή είναι και η βασική ιδέα του IoT το οποίο φαίνεται από πολλούς ειδικούς πως είναι η φυσική εξέλιξη του διαδικτύου. Βέβαια υπάρχει και το ερώτημα πως μπορεί να λειτουργήσει το IoT και αν είναι μία συγκεκριμένη υπηρεσία. Το διαδίκτυο των πραγμάτων είναι ένα δίκτυο στο οποίο είναι συνδεδεμένοι υπολογιστές και κάθε είδους αντικείμενο αρκεί να περιέχει κάποιο ενσωματωμένο σύστημα αλλά και να μπορεί να συνδεθεί στο δίκτυο. Μέσω της διασύνδεσης θα μπορούν να επικοινωνούν, να αλληλοεπιδρούν και να ανταλλάσσουν πληροφορίες μεταξύ τους, χωρίς να χρειάζεται να επέμβει ο ανθρώπινος παράγοντας. Τα αντικείμενα μπορεί να είναι διάφορες οικιακές συσκευές, μέσα μεταφοράς, μέσα τηλεπικοινωνίας, πόρτες, παράθυρα, βιβλία, αυτοκίνητα κ.α. Με αυτό το τρόπο το διαδίκτυο θα γίνει διαδραστικό, ένα τεράστιο ιεραρχικά οργανωμένο «νευρικό σύστημα» που θα χρησιμοποιεί συσκευές με αισθητήρες και ενεργοποιητές που θα συνεργάζονται για έξυπνες υπηρεσίες για τις επιχειρήσεις, την υγεία, τις μεταφορές, την διανομή, την κατανάλωση ενέργειας κ.α. Ακόμα μέσω του διαδικτύου των πραγμάτων μπορεί να γίνει η συλλογή τεράστιων δεδομένων, να ολοκληρώσει εργασίες και να χρησιμοποιηθεί για την βελτίωση των επιχειρηματικών διεργασιών αλλά και να αποτελέσει αφορμή για νέα επιχειρηματική καινοτομία.

Το IoT αποτελείται από:

- Τα αντικείμενα / πράγματα.
- Τα δίκτυα επικοινωνιών στα οποία θα συνδέονται.
- Τους υπολογιστές και τα συστήματα τα οποία θα συλλέγουν τις πληροφορίες και τα δεδομένα από τα αντικείμενα.
- Τα αντικείμενα θα είναι είτε φυσικές οντότητες είτε εικονικές.
- Θα διαθέτουν ταυτότητα για τον αυτόματο εντοπισμό τους.
- Δεν θα είναι κακόβουλα
- Θα περιέχουν πρωτόκολλα για την ασφάλεια και την ιδιωτικότητα των δεδομένων ή των ατόμων με τα οποία αλληλοεπιδρούν.
- Θα χρησιμοποιούν πρωτόκολλα επικοινωνίας και θα ανταλλάσσουν πληροφορίες μεταξύ φυσικών και ψηφιακών οντοτήτων.

Επιπλέον τα αντικείμενα μπορούν να λειτουργούν αυτόνομα, να αντιλαμβάνονται μέσω των διάφορων αισθητηρίων και να παίρνουν αποφάσεις ανάλογα τις συνθήκες ακόμη και να εξελίσσονται. Όπως προαναφέρθηκε το IoT δεν είναι μια ενιαία τεχνολογία, βασίζεται σε προ υπάρχουσες τεχνολογίες και λειτουργεί με αυτές. Κάποιες από αυτές είναι:

- Τεχνολογίες επικοινωνίας και συνεργασίας: Τα αντικείμενα έχουν τη δυνατότητα να δικτυωθούν μεταξύ τους ή με το διαδίκτυο χρησιμοποιώντας πρωτόκολλα όπως GSM, UMTS, WiFi, Bluetooth, Zigbee κ.α.
- Τεχνολογίες διευθυνσιοδότησης: Τα αντικείμενα έχουν ταυτότητα που την χρησιμοποιούν για να εντοπιστούν και να δώσουν πληροφορίες τις οποίες έχουν σχεδιαστεί να παρέχουν.
- Ταυτότητα: Τα αντικείμενα διαθέτουν μοναδικό serial number για να αναγνωρίζονται. Όσες συσκευές δεν έχουν όμως μπαταρία και είναι παθητικές μπορούν να λειτουργήσουν με τεχνολογίες RFID, NFC ή ακόμα και με barcode.
- Τηλεπισκόπηση: Οι αισθητήρες των αντικειμένων λαμβάνουν ερεθίσματα από το περιβάλλον, συλλέγουν πληροφορίες και αλληλοεπιδρούν με άλλα αντικείμενα.
- Ενεργοποίηση: Ένα βασικό πλεονέκτημα των αντικειμένων είναι ότι μπορούν να αλληλοεπιδράσουν απομακρυσμένα μετατρέποντας την ηλεκτρική τάση σε μηχανική κίνηση. Με αυτό το τρόπο μπορούν να χρησιμοποιηθούν για έλεγχο εξ αποστάσεως στον πραγματικό κόσμο.
- Εντοπισμός θέσης: Τα έξυπνα αντικείμενα μπορούν να έχουν εντοπισμό της θέσης τους μέσω τεχνολογιών GPS ή μέσω του τηλεφωνικού δικτύου.
- Διεπαφές χρήστη: Τα έξυπνα αντικείμενα μπορούν να αλληλοεπιδρούν με τους ανθρώπους.

Τέλος μέσω του IoT στο μέλλον θα δημιουργηθούν νέες αγορές, θα προκύψουν νέες ιδέες και θα βελτιώσουν υπάρχουσες υπηρεσίες κάνοντας τις έξυπνες. Μερικά παραδείγματα από κλάδους στους οποίους αναπτύσσεται το IoT αυτή την στιγμή είναι:

- Έξυπνες λύσεις στον κλάδο των μεταφορών. Για παράδειγμα με την μείωση της κίνησης στους δρόμους.
- Έξυπνα ηλεκτρικά δίκτυα που ενσωματώνουν περισσότερες ανανεώσιμες πηγές ενέργειας.
- Απομακρυσμένη παρακολούθηση ασθενών.
- Έξυπνα σπίτια.
- Αισθητήρες μηχανών παρακολούθησης, που εντοπίζουν και προβλέπουν πότε χρειάζεται συντήρηση ή επισκευή ένα μηχάνημα μειώνοντας έτσι το κόστος επισκευής. [9]

3.3. Η πλατφόρμα ThingSpeak

Το ThingSpeak είναι μία πλατφόρμα που χρησιμοποιείται για εφαρμογές που σχετίζονται αποκλειστικά με το διαδίκτυο το πραγμάτων. Είναι ελεύθερο προς χρήση στο διαδίκτυο για όλους τους χρήστες και απαιτείται η εγγραφή στην πλατφόρμα χρησιμοποιώντας ένα email για την εγγραφή. Αφού κάνουμε εγγραφή μας ζητάει να δημιουργήσουμε ένα νέο κανάλι.

The image shows a screenshot of the ThingSpeak registration page on the left and a diagram of the IoT data flow on the right.

Registration Page (Left): The page is titled "Sign up for ThingSpeak". It includes a navigation bar with "Channels", "Apps", and "Support". Below the navigation bar, there is a "Create MathWorks Account" section. The "Email Address" field is empty and has a red border with a warning icon and the message "Missing required information". Below it, there is a note: "To access your organization's MATLAB license, use your school or work email." The "Location" dropdown menu is set to "United States". The "First Name" and "Last Name" fields are also empty.

Diagram (Right): The diagram illustrates the data flow. On the left, there are "SMART CONNECTED DEVICES" represented by icons of a smartphone, a laptop, and a server. These devices send data to a central cloud labeled "DATA AGGREGATION AND ANALYTICS ThingSpeak". From the cloud, data is sent to a "MATLAB" computer icon labeled "ALGORITHM DEVELOPMENT SENSOR ANALYTICS".

Εικόνα 12 Registration στην πλατφόρμα ThingSpeak

My Channels

[New Channel](#)

Help

Collect data in a ThingSpeak channel from a device, from another channel, or from the web.

Click **New Channel** to create a new ThingSpeak channel.

Click on the column headers of the table to sort by the entries in that column or click on a tag to show channels with that tag.

Learn to [create channels](#), explore and transform data.

Learn more about [ThingSpeak Channels](#).

Εικόνα 13 Δημιουργία νέου καναλιού

Αφού πατήσουμε στην δημιουργία νέου καναλιού μας εμφανίζει ένα νέο παράθυρο για να επεξεργαστούμε το κανάλι. Σε αυτή τη σελίδα θα καταχωρήσουμε στα πεδία που εμφανίζει τα κατάλληλα στοιχεία. Στο πεδίο «Name» θα καταχωρήσουμε το όνομα του πρότζεκτ, στη συγκεκριμένη περίπτωση το όνομα είναι «measurements» και θα ανοίξουμε 4 fields για να έχουμε τέσσερα διαφορετικά γραφήματα. Ένα για κάθε μέτρηση που θα λαμβάνουμε από την συσκευή μας. Τα ονόματα που δόθηκαν για να ξεχωρίζουμε τα γραφήματα είναι τα ονόματα των αισθητήρων.

Field 1: ds18b20

Field 2: dht11_temp

Field 3: dht11_hum

Field 4: rain

New Channel

Name	<input type="text" value="measurements"/>
Description	<input type="text"/>
Field 1	<input type="text" value="ds18b20"/> <input checked="" type="checkbox"/>
Field 2	<input type="text" value="dht11_temp"/> <input checked="" type="checkbox"/>
Field 3	<input type="text" value="dht11_hum"/> <input checked="" type="checkbox"/>
Field 4	<input type="text" value="rain"/> <input checked="" type="checkbox"/>
Field 5	<input type="text"/> <input type="checkbox"/>
Field 6	<input type="text"/> <input type="checkbox"/>
Field 7	<input type="text"/> <input type="checkbox"/>
Field 8	<input type="text"/> <input type="checkbox"/>
Metadata	<input type="text"/>

Εικόνα 14 Επεξεργασία καναλιού

Όταν δημιουργήσουμε τις τιμές στα πεδία πατάμε save channel για να αποθηκευτούν οι αλλαγές. Μετά την αποθήκευση ανοίγει ένα νέο interface το οποίο μας δείχνει κάποια στοιχεία όπως το «channel ID», το «author» και τα γραφήματα που δημιουργήσαμε. Όπως φαίνεται και στην εικόνα είναι κενά διότι ακόμα δεν έχουμε συνδέσει τη συσκευή μας με το ThingSpeak για την επικοινωνία και την μετάδοση δεδομένων. Η σύνδεση θα γίνει μέσω του API key το οποίο βρίσκεται στην καρτέλα API keys στο μενού του ThingSpeak και είναι το κλειδί ταυτοποίησης της συσκευής στην πλατφόρμα. [10]

measurements

Channel ID: **1036066**
Author: [mwa0000018130287](#)
Access: Private

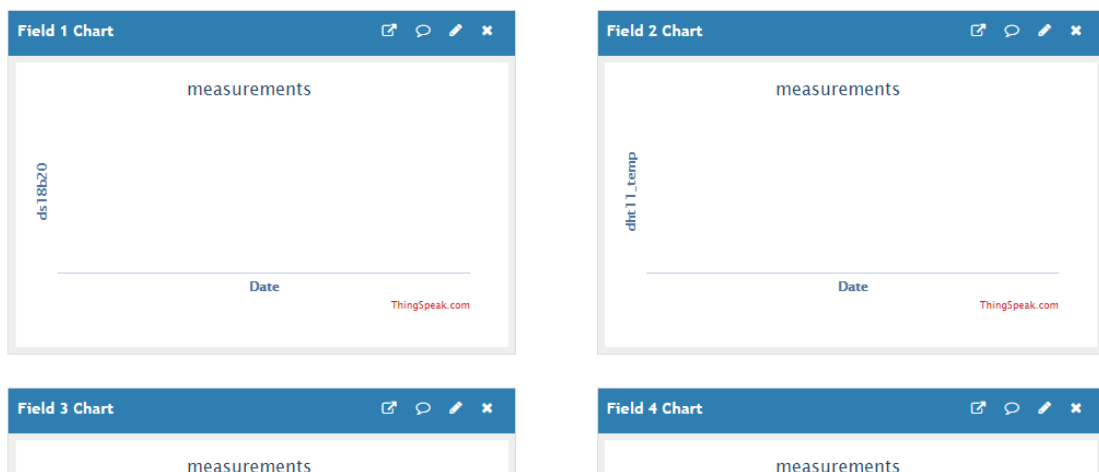
Private View Public View Channel Settings Sharing API Keys Data Import / Export

Add Visualizations Add Widgets Export recent data

MATLAB Analysis MATLAB Visualization

Channel Stats

Created: [less than a minute ago](#)
Entries: 0



Εικόνα 15 Καρτέλα Γραφημάτων των μετρήσεων μας

measurements

Channel ID: **1036066**
Author: [mwa0000018130287](#)
Access: Private

Private View Public View Channel Settings Sharing API Keys Da

Write API Key

Key 8SX3UJTBM7YE9VAP

Generate New Write API Key

Read API Keys

Key 96DNLLTOGUCBASFG

Note

Save Note Delete API Key

Add New Read API Key

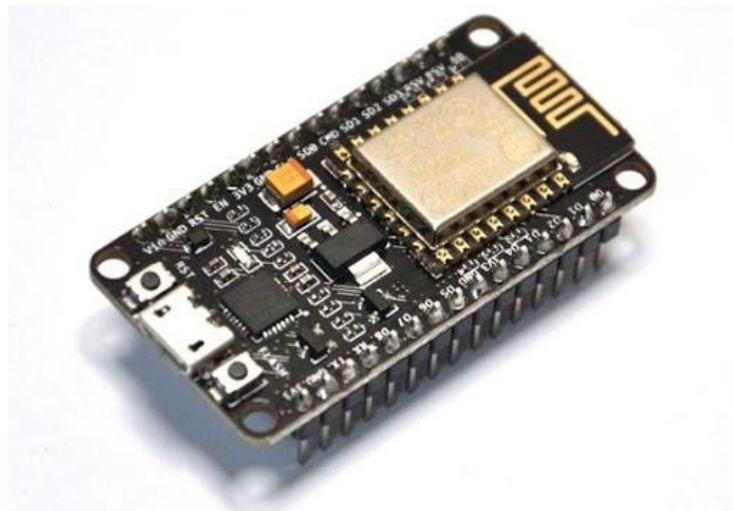
Εικόνα 16 Κλειδί ταυτοποίησης API key

4. Περιγραφή Εξαρτημάτων

Για τις ανάγκες της εργασίας χρησιμοποιήθηκαν τα εξής εξαρτήματα:

1. NodeMCU 1.0 ESP-12E
2. DHT11 Humidity & Temperature sensor
3. DS18B20 Waterproof Temperature sensor
4. Rain drop sensor module
5. Resistors
6. Wires
7. Breadboard
8. Usb A – micro B for Power Supply

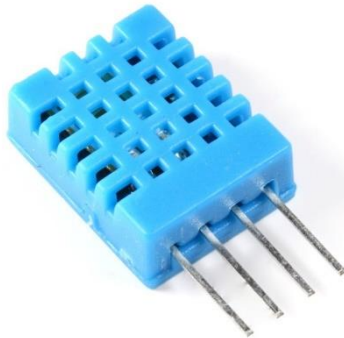
4.1. NodeMCU 1.0 - 12E



Εικόνα 17 Πλακέτα NodeMcu 1.0 ESP 12E

- Voltage: 3.3V.
- Wi-Fi Direct (P2P), soft-AP.
- Current consumption: 10uA~170mA.
- Flash memory attachable: 4MB.
- Integrated TCP/IP protocol stack.
- Processor: Tensilica L106 32-bit.
- Processor speed: 80~160MHz.
- RAM: 128 KB.
- GPIOs: 17 (multiplexed with other functions).
- Analog to Digital: 1 input with 1024 step resolution.
- +19.5dBm output power in 802.11b mode
- 802.11 support: b/g/n.
- Maximum concurrent TCP connections: 5. [7]

4.2. Sensor DHT11 Temperature & Humidity



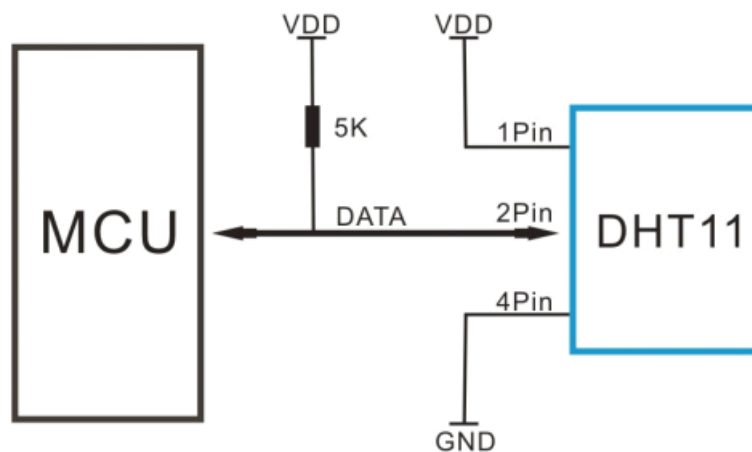
Εικόνα 18 DHT11

Ο αισθητήρας DHT11 διαθέτει αισθητήρα μέτρησης υγρασίας και θερμοκρασίας. Η έξοδος που βγάζει είναι ένα βαθμονομημένο ψηφιακό σήμα. Για την μέτρηση της θερμοκρασίας περιέχει ένα θερμίστορ τύπου NTC. Επιπλέον διαθέτει έναν 8 μπιτο μικροελεγκτή ο οποίος προσφέρει ποιότητα, γρήγορη απόκριση και αντι – παρεμβολές. Ο αισθητήρας έχει 4 ακίδες όμως μόνο οι τρεις χρειάζονται για την σύνδεση. Δύο ακίδες χρησιμοποιούνται για την τάση και γείωση και η τρίτη σε μία ψηφιακή υποδοχή του μικροελεγκτή.

Προδιαγραφές Sensor:

- Operating Voltage: 3.5V to 5.5V
- Operating current: 0.5mA (measuring) 150uA (standby)
- Output: Serial data
- Temperature Range: 0 °C to 50 °C
- Humidity Range: 20 % to 90 %
- Resolution: Temperature: 8-bit, Humidity: 8 bit
- Accuracy: $\pm 2^{\circ}\text{C}$ and $\pm 5\%$

Η εγκατάσταση και η σύνδεση είναι πολύ απλή, όπως φαίνεται και στην παρακάτω εικόνα. Ο κατασκευαστής αναφέρει ότι για απόσταση μικρότερη των 20 μέτρων συνίσταται μία αντίσταση pull up 5k ενώ για απόσταση μεγαλύτερη των 20 μέτρων (αναφορά στο καλώδιο σύνδεσης sensor – MCU) χρειάζεται απαραίτητα αυτή την αντίσταση.



Εικόνα 19 Σύνδεση MCU - DHT11

Επιπλέον σημαντικό χαρακτηριστικό του dht11 είναι μικρή κατανάλωση ενέργειας. Σε τάση λειτουργίας 3V όταν ο αισθητήρας είναι σε κατάσταση ηρεμίας καταναλώνει 100μΑ ενώ όταν παράγει μέτρηση η κατανάλωση είναι 0.5 mA.

	Conditions	Minimum	Typical	Maximum
Power Supply	DC	3V	5V	5.5V
Current Supply	Measuring	0.5mA		2.5mA
	Average	0.2mA		1mA
	Standby	100uA		150uA
Sampling period	Second	1		

Εικόνα 20 Πίνακας κατανάλωσης ενέργειας DHT11

Τέλος για την σωστή λειτουργία του αισθητήρα, κάποιες παρατηρήσεις που επισημαίνονται στο φύλλο κατασκευής του είναι:

- 1) Αν ο αισθητήρας βρίσκεται τοποθετημένος σε σημείο που υπάρχουν ατμοί και χημικά θα επηρεάσουν την ευαισθησία του και θα δημιουργήσουν βλάβη στον sensor.
- 2) Η μακροχρόνια έκθεση στο ηλιακό φως και την υπεριώδη ακτινοβολία θα προκαλέσουν βλάβη και μείωση της απόδοσης του αισθητήρα.
- 3) Η ποιότητα των καλωδίων μπορεί να επηρεάσει την ποιότητα των μετρήσεων για αυτό συνίσταται καλώδιο θωράκισης.
- 4) Για το κύκλωμα η θερμοκρασία συγκόλλησης πρέπει να είναι κάτω από 260 βαθμούς κελσίου και η επαφή να είναι μικρότερη των 10 δευτερολέπτων. [12]

4.3. Sensor ds18B20 waterproof Temperature



Εικόνα 21 ds18B20 sensor

Ο ds18b20 είναι ένας ψηφιακός αισθητήρας με ανάλυση μετρήσεων από 9 έως 12 bit. Διαθέτει στο εσωτερικό του κύκλωμα και τσιπ στο οποίο υπάρχει μία EEPROM μνήμη που αποθηκεύεται η 64 bit διεύθυνση αναγνώρισης του αισθητήρα (δηλαδή ένα μοναδικό serial number). Κάποια από τα πλεονεκτήματα αυτού του sensor είναι ότι μπορούν στο ίδιο καλώδιο να συνδεθούν πολλοί αισθητήρες μέσω του δίαυλου bus. Αυτός είναι και ένας λόγος που διαθέτει serial number. Επιπλέον μπορεί να

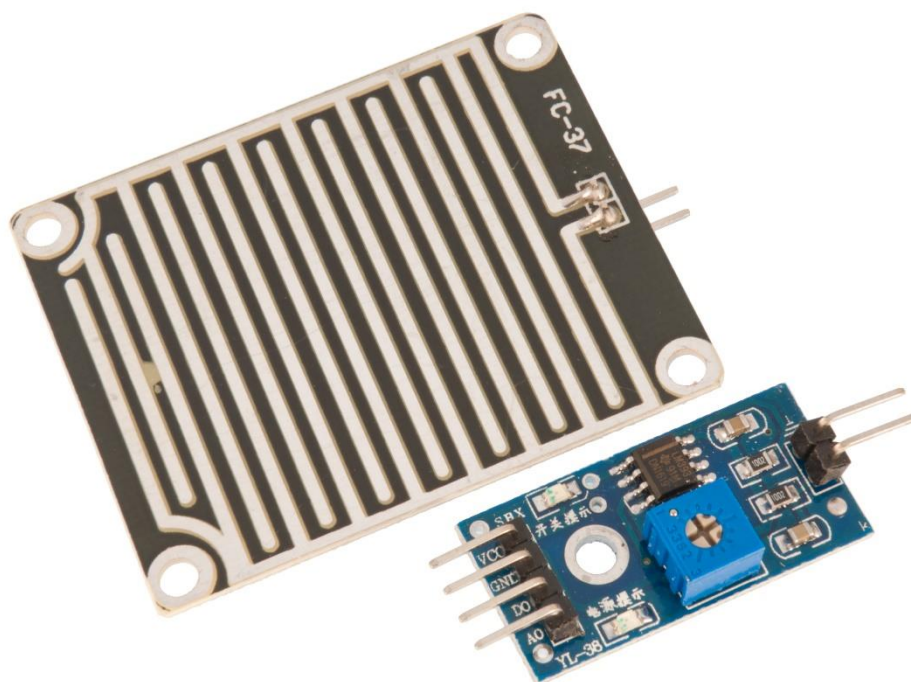
τροφοδοτηθεί απευθείας μέσω του καλωδίου δεδομένων εξαιρίζοντας την ανάγκη για εξωτερική παροχή ρεύματος (μέσω του parasite power). Λόγω της απόδοσης, της αξιοπιστίας και της λειτουργικότητας του αυτός ο αισθητήρας μπορεί να χρησιμοποιηθεί σε HVAC συστήματα, σε βιομηχανικές εφαρμογές, σε καταναλωτικά προϊόντα, σε θερμομέτρα, μετεωρολογικούς σταθμούς και πολλές ακόμα εφαρμογές. Τέλος έχει πολύ μικρή κατανάλωση ενέργειας, σε ενεργή κατάσταση η κατανάλωση είναι 1.5 mA ενώ σε κατάσταση ηρεμίας η κατανάλωση είναι 1000 nA.

- Measures Temperatures: -55°C to $+125^{\circ}\text{C}$ (-67°F to $+257^{\circ}\text{F}$)
- $\pm 0.5^{\circ}\text{C}$ Accuracy from -10°C to $+85^{\circ}\text{C}$
- Voltage Range: -0.5V to $+6.0\text{V}$

Thermometer error	Temperature conversion time:	
-10°C to $+85^{\circ}\text{C}$ $\sim \pm 0.5\text{ C}$	9-bit resolution	93.75 ms
-30°C to $+100^{\circ}\text{C}$ $\sim \pm 1\text{ C}$	10-bit resolution	187.5 ms
-55°C to $+125^{\circ}\text{C}$ $\sim \pm 2\text{ C}$	11-bit resolution	375 ms
	12-bit resolution	750 ms

Για το συγκεκριμένο πρότζεκτ παραμετροποιήσαμε τον αισθητήρα σε ανάλυση 12 bit για να πάρουμε την ακριβέστερη μέτρηση θερμοκρασίας. [11]

4.4. Sensor rain drop Module



Εικόνα 22 Sensor Rain Drop Module

Ο rain drop sensor αποτελείται από την πλαστική επιφάνεια που χρησιμοποιείται για την βροχή και μία μονάδα ελέγχου που βασίζεται στο τσιπ LM393 και ένα ποτενσιόμετρο που ρυθμίζει την ευαισθησία του sensor στην βροχή. Μέσω αυτής την μονάδας ο αισθητήρας μπορεί να λειτουργήσει είτε με αναλογική έξοδο είτε μετατρέποντας την σε ψηφιακή. Τέτοιοι αισθητήρες μπορούν να χρησιμοποιηθούν σε πολλούς τομείς όπως στην αυτοκινητοβιομηχανία για τον αυτόματο έλεγχο των υαλοκαθαριστήρων, στην γεωργία, σε μετεωρολογικούς σταθμούς και για συστήματα αυτοματισμών σε πολλές οικιακές χρήσεις. Ο αισθητήρας λειτουργεί σαν μία μεταβλητή αντίσταση που βρίσκεται

στο πλαστικό PCB το οποίο στην επάνω επιφάνεια του επικαλύπτεται από γραμμές νικελίου με αποτέλεσμα αν πέσει μία σταγόνα νερού να μειώσει την αντίσταση επειδή το νερό είναι καλός αγωγός της ηλεκτρικής ενέργειας και η παρουσία του νερού συνδέει παράλληλα τις γραμμές νικελίου επομένως όσο μικρότερη αντίσταση τόσο μειώνει και την πτώση τάσης. Αντίθετα όταν δεν υπάρχει πτώση βροχής η αντίσταση είναι αυξημένη και παίρνουμε ένα σήμα υψηλής τάσης.

Η μονάδα ελέγχου έχει 4 εξόδους για τη σύνδεση με τον μικροελεγκτή μας. Αυτές είναι:

1	VCC	Τάση τροφοδοσίας - 5V
2	GND	Σύνδεση με την γείωση
3	D0	Ψηφιακή ακίδα για ψηφιακή έξοδο
4	A0	Αναλογική ακίδα για αναλογική έξοδο

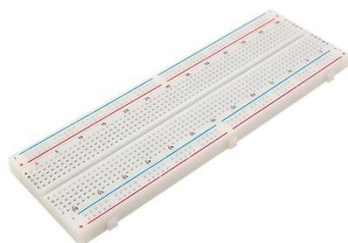
Και τα χαρακτηριστικά του αισθητήρα είναι:

- Τάση λειτουργίας 5V
- Μορφή εξόδου: Ψηφιακή έξοδος και αναλογική έξοδος τάσης AO
- Το ποτενσιόμετρο ρυθμίζει την ευαισθησία
- Χρησιμοποιεί ένα τσιπ LM393
- Κατανάλωση ενέργειας πάνω από 15 mA
- Αντιοξειδωτική, αντι-αγωγιμότητα, με μεγάλο χρόνο χρήσης
- Με οπές για εύκολη εγκατάσταση
- Μικρό μέγεθος PCB: 3,2εκ. X 1,4εκ

[13]

4.5. Άλλα Εξαρτήματα

Τέλος για την σωστή σύνδεση χρησιμοποιήσαμε breadboard, αντιστάσεις και καλώδια για την ένωση όλων των εξαρτημάτων με τον μικροελεγκτή και το breadboard και καλώδιο USB για την τροφοδοσία της πλακέτας με τον υπολογιστή.



5. Υλοποίηση

Σε αυτό το κεφάλαιο θα αναλύσουμε τις βασικές συναρτήσεις του κώδικα, τα διαγράμματα ροής, την συνδεσμολογία του κυκλώματος αλλά και την ολοκληρωμένη λειτουργία του συστήματος.

5.1. Βιβλιοθήκες

Μέσω της βιβλιοθήκης ThingSpeak στέλνουμε τις μετρήσεις στην πλατφόρμα ThingSpeak.
`#include "ThingSpeak.h"`

Βιβλιοθήκη για την χρήση του αισθητήρα DHT11 .
`#include "DHT.h"`

Βιβλιοθήκη για την χρήση του αισθητήρα ds18b20 waterproof sensor.
`#include <DallasTemperature.h>`

Μέσω της βιβλιοθήκης oneWire γίνεται η επικοινωνία του ds18b20 με τον μικροελεγκτή από τον διάλο bus.
`#include <OneWire.h>`

Βασική βιβλιοθήκη για την λειτουργία του ESP8266EX. Μέσω αυτής λειτουργούν όλες οι συναρτήσεις για το WiFi δίκτυο.
`#include <ESP8266WiFi.h>`

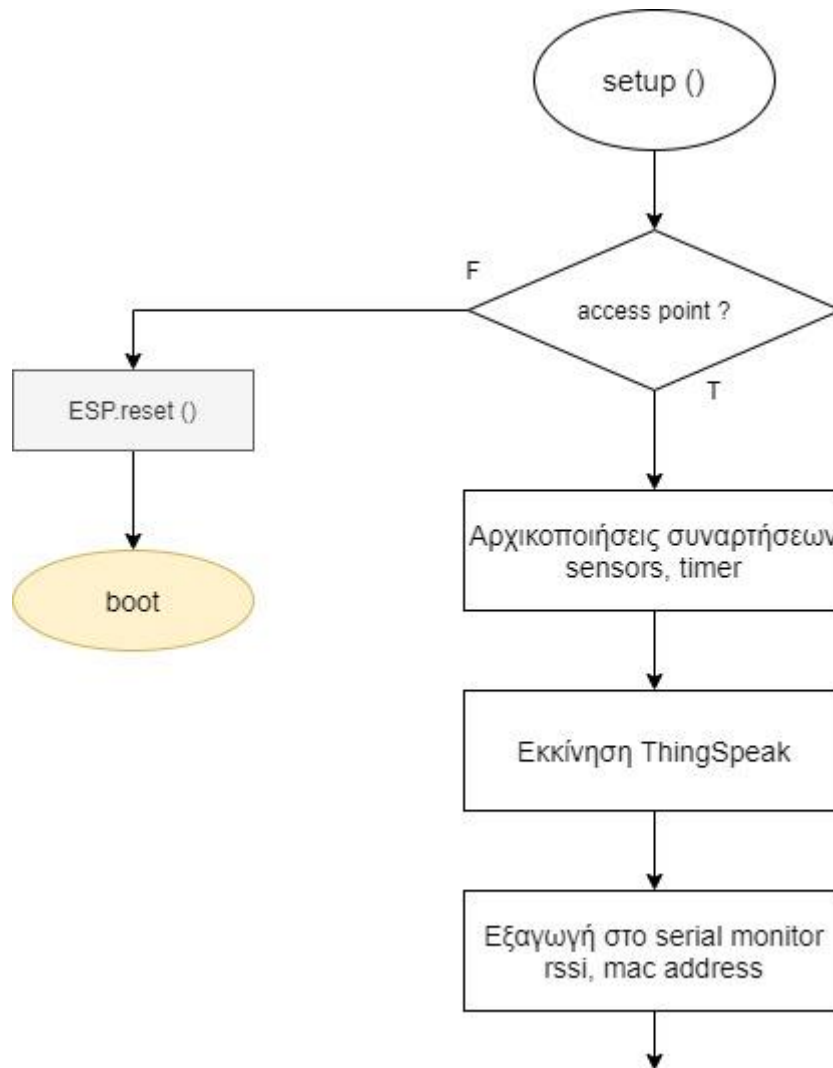
Η βιβλιοθήκη αυτή αρχικοποιεί και ενεργοποιεί τον timer του μικροελεγκτή.
`#include <Ticker.h>`

Βιβλιοθήκες για την χρήση του WiFiManager. Δηλαδή για το configuration του access point.
`#include <DNSServer.h>`
`#include <ESP8266WebServer.h>`
`#include <WiFiManager.h>`

[17], [18], [19], [20], [21], [22], [23]

5.2. Διάγραμμα Ροής

Το διάγραμμα ροής χωρίζεται σε δύο μέρη. Στις βασικές συναρτήσεις του Arduino IDE `setup ()` και `loop ()`. [14]



Εικόνα 23 Διάγραμμα Ροής setup ()

Η `setup ()` είναι η συνάρτηση των αρχικοποιήσεων και που το πρόγραμμα μας θα μπει μία φορά, όταν τελειώσει τις εντολές της `setup` θα συνεχίσει στην συνάρτηση `loop ()`. Εδώ αρχικοποιούμε τις συναρτήσεις των αισθητήρων και του `timer`. Επιπλέον αυτή την συνάρτηση γίνεται η εγκατάσταση σε ένα WiFi δίκτυο μέσω της βιβλιοθήκης `wifiManager.h` και η εκκίνηση της πλατφόρμας `ThingSpeak`. Τέλος για δική μας διευκόλυνση τυπώνουμε κάποια μηνύματα στο `serial monitor` του `Arduino IDE` όπως την `mac address` και το `rssi` (η λαμβανόμενη ισχύς του σήματος στην πλακέτα μας).

Μέσω της βιβλιοθήκης `WiFiManager.h` χρησιμοποιήσαμε την συνάρτηση:

➤ **`wifiManager.startConfigPortal ("NodeMcu")`**

Για να μπορέσουμε να μπούμε σε κατάσταση `access point` και να συνδεθούμε με το δίκτυο μας.

Μέσω της βιβλιοθήκης `ThingSpeak.h` χρησιμοποιήσαμε την συνάρτηση:

➤ **`ThingSpeak.begin (client)`**

Για να συνδεθούμε με την πλατφόρμα `ThingSpeak`.

Μέσω της βιβλιοθήκης Ticker.h χρησιμοποιήσαμε την συνάρτηση:

➤ **timer. attach (300, flag)**

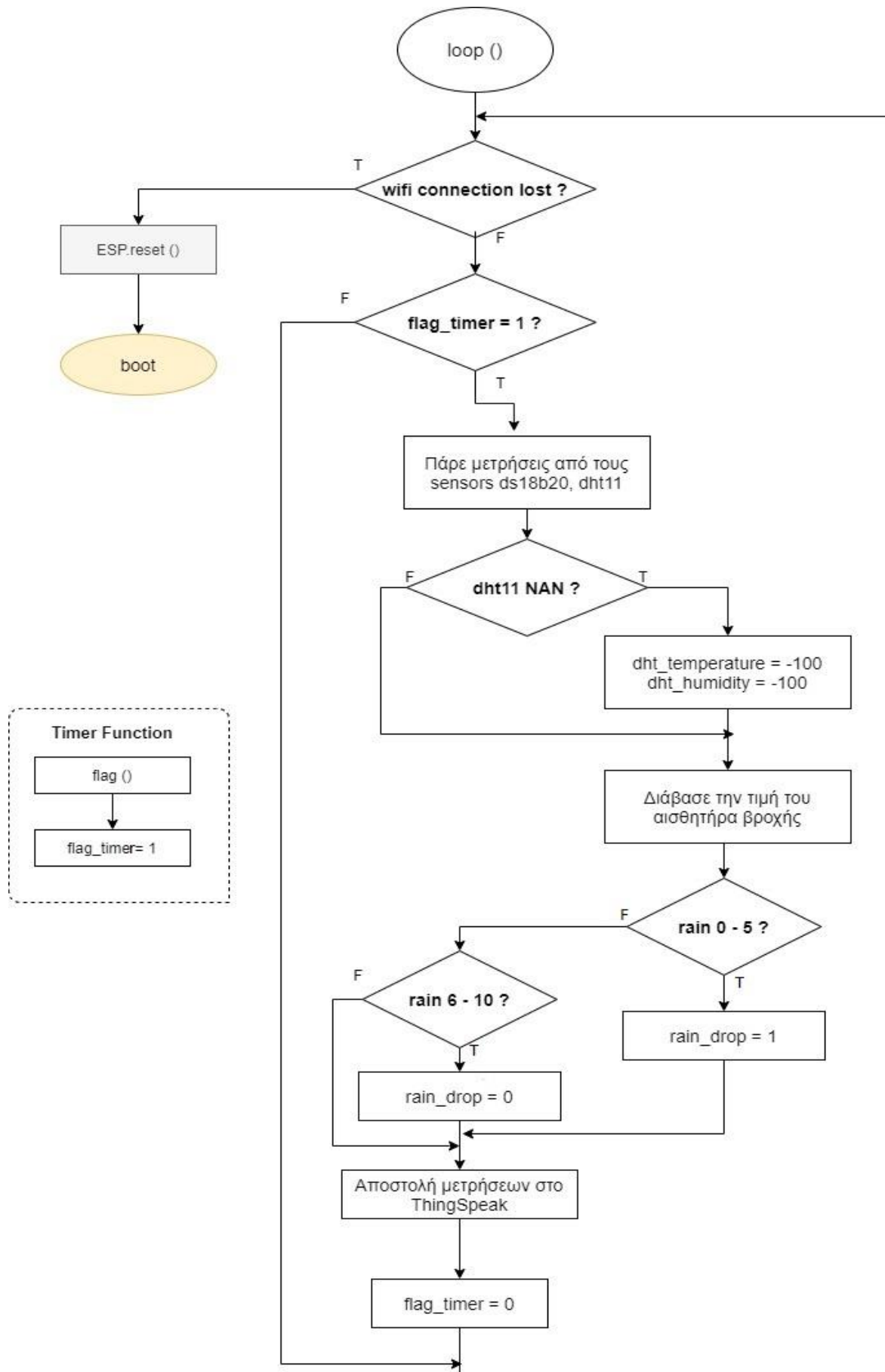
Στην οποία γίνεται η αρχικοποίηση του timer. Στις παραμέτρους της συνάρτησης ορίζουμε τον χρόνο που θα μετράει ο timer σε δευτερόλεπτα και στη δεύτερη παράμετρο ορίζουμε την συνάρτηση στην οποία θα μπει όταν φτάσει τα 300 δευτερόλεπτα (5 λεπτά, αφού θέλουμε να στέλνουμε μία μέτρηση στην πλατφόρμα ανά 5 λεπτά).

Μέσω της βιβλιοθήκης DHT.h χρησιμοποιήσαμε την συνάρτηση για να αρχικοποιήσουμε τον αισθητήρα μας:

dht. begin () // Για sensor dht11.

Μέσω της βιβλιοθήκης DallasTemperature.h και OneWire.h χρησιμοποιήσαμε την συνάρτηση για να αρχικοποιήσουμε τον αισθητήρα μας:

➤ **sensors. begin ()** // Για sensor ds18B20.



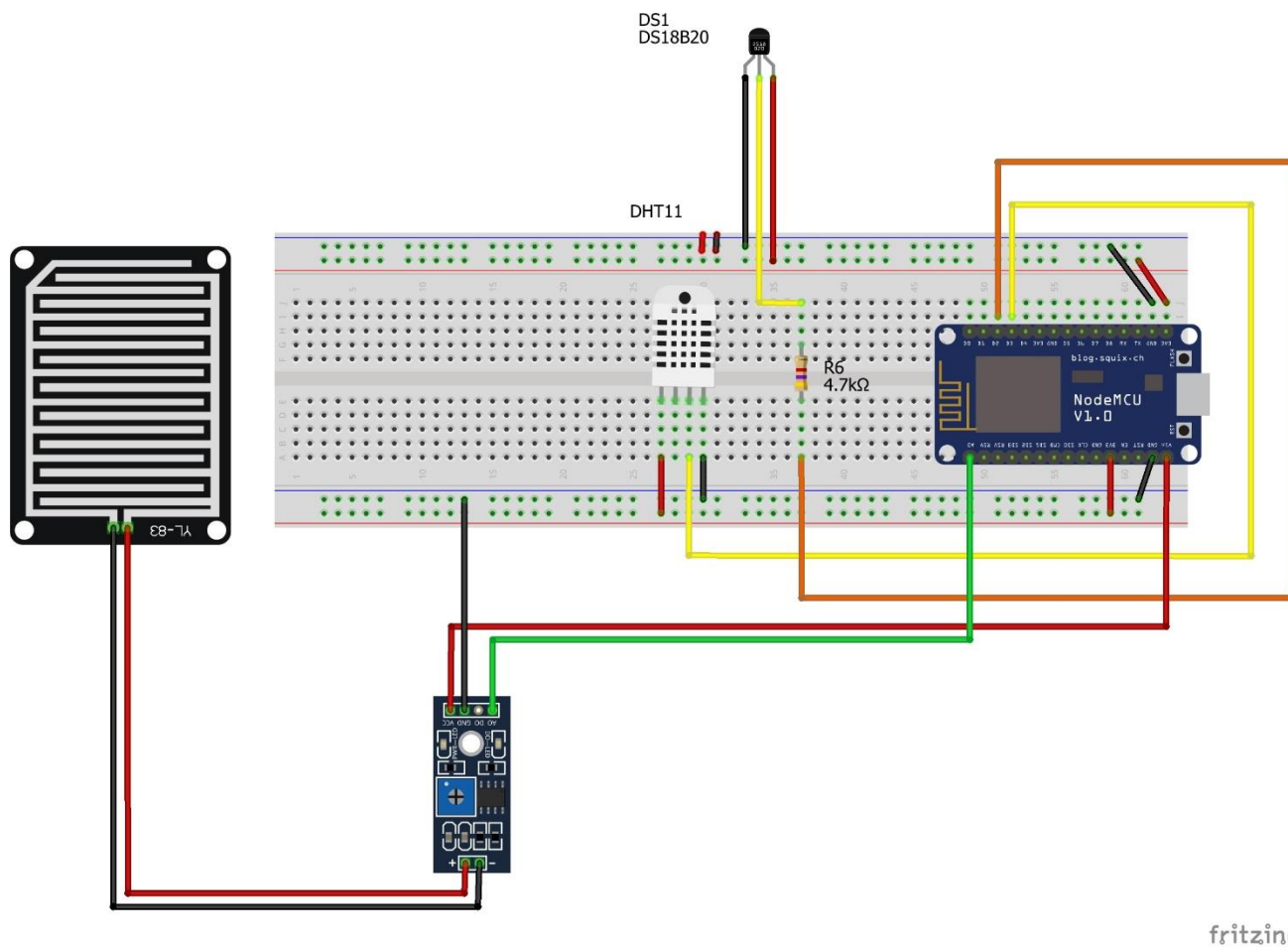
Εικόνα 24 Διάγραμμα Ροής της συνάρτησης loop ()

Η συνάρτηση `loop ()` είναι ο ατέρμων βρόχος που θα παραμείνει το πρόγραμμα μας αφού ολοκληρώσει την συνάρτηση `setup`. Σε αυτή γίνεται έλεγχος αν διακοπεί η σύνδεση με το δίκτυο και έλεγχος αν πέρασαν 5 λεπτά μέσω του `timer` για να μεταδώσουμε τις μετρήσεις στην πλατφόρμα `ThingSpeak`. Οι κύριες συναρτήσεις που χρησιμοποιήθηκαν για τον προγραμματισμό είναι:

- **ThingSpeak.setField (field, value)** // Τα ορίσματα αυτής της συνάρτησης είναι για να ορίσουμε το `field` στο οποίο απευθυνόμαστε και `value` είναι η τιμή που θα μπει σε αυτό το όρισμα.
- **ThingSpeak.writeFields(channelNumber, writeAPIKey)** // Με αυτή την συνάρτηση μεταδίδουμε το μήνυμα σε όλα τα `fields` με ορίσματα το `channel ID` και το `API KEY` του λογαριασμού μας στο `ThingSpeak`.
- **sensors.requestTemperatures ()** // Η χρήση της συνάρτησης αυτής είναι για να παράξει μία μέτρηση θερμοκρασίας από τον αισθητήρα `ds18b20`.
- **sensors.getTempCByIndex (0)** // Αυτή η συνάρτηση μας στέλνει την μέτρηση της θερμοκρασίας στο μικροελεγκτή μέσω του δίαυλου `bus`. Η παράμετρος ορίζει την αρίθμηση του αισθητήρα (επειδή στον δίαυλο `bus` μπορούμε να συνδέσουμε πολλούς αισθητήρες).
- **dht.readTemperature()** // Παράγει μία μέτρηση θερμοκρασίας από τον `sensor DHT11`
- **dht.readHumidity()** // Παράγει μία μέτρηση υγρασίας από τον `sensor DHT11`
- **map (value, fromLow, fromHigh, toLow, toHigh)** // Χρησιμοποιήθηκε για τον αναλογικό αισθητήρα `rain drop sensor`. Η λειτουργία της είναι να μετατρέπει κάποια όρια σε διαφορετικά όρια. Για παράδειγμα από 0 έως 1023 σε 0 έως 10. Περιορίζοντας τα όρια μπορούμε να είμαστε πιο ακριβείς στην εγγραφή του κώδικα μας όπως παρατηρούμε στο διάγραμμα ροής.

5.3. Συνδεσμολογία κυκλώματος

Παρακάτω εμφανίζεται η συνδεσμολογία του κυκλώματος του μετεωρολογικού σταθμού που υλοποιήθηκε για την εργασία. Η σχεδίαση έγινε με τη βοήθεια του σχεδιαστικού προγράμματος `Fritzing`. Στο σχέδιο έχουν ενσωματωθεί το `NodeMCU 1.0`, οι αισθητήρες `ds18b20` και `dht11` καθώς και ο `sensor rain` με τον `controller` του. [15]

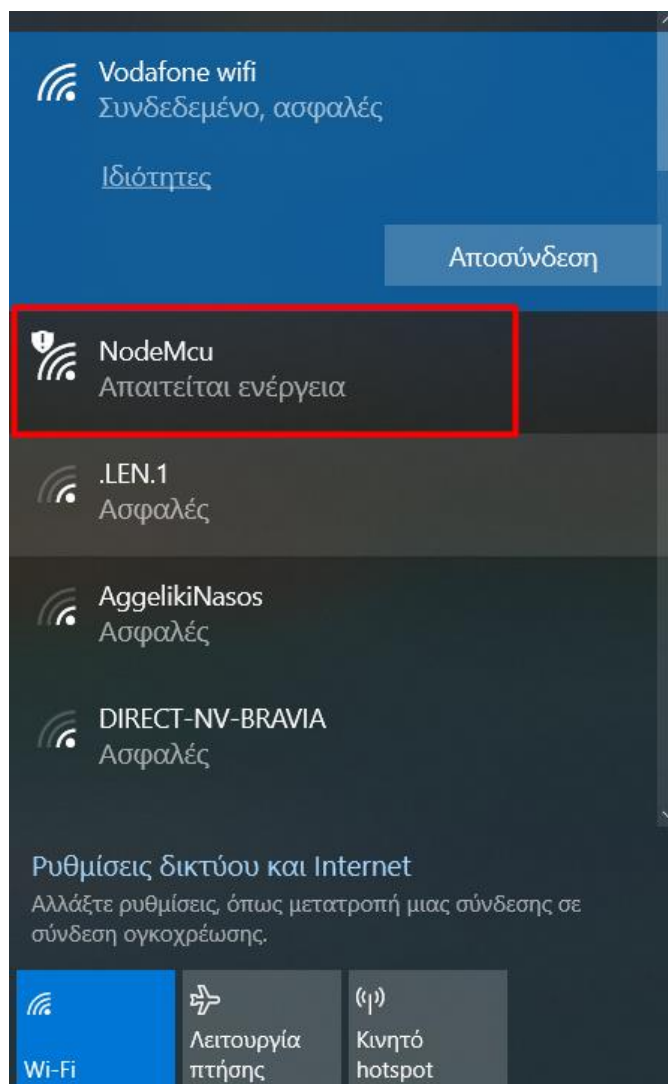


Εικόνα 25 Συνδεσμολογία κυκλώματος

fritzing

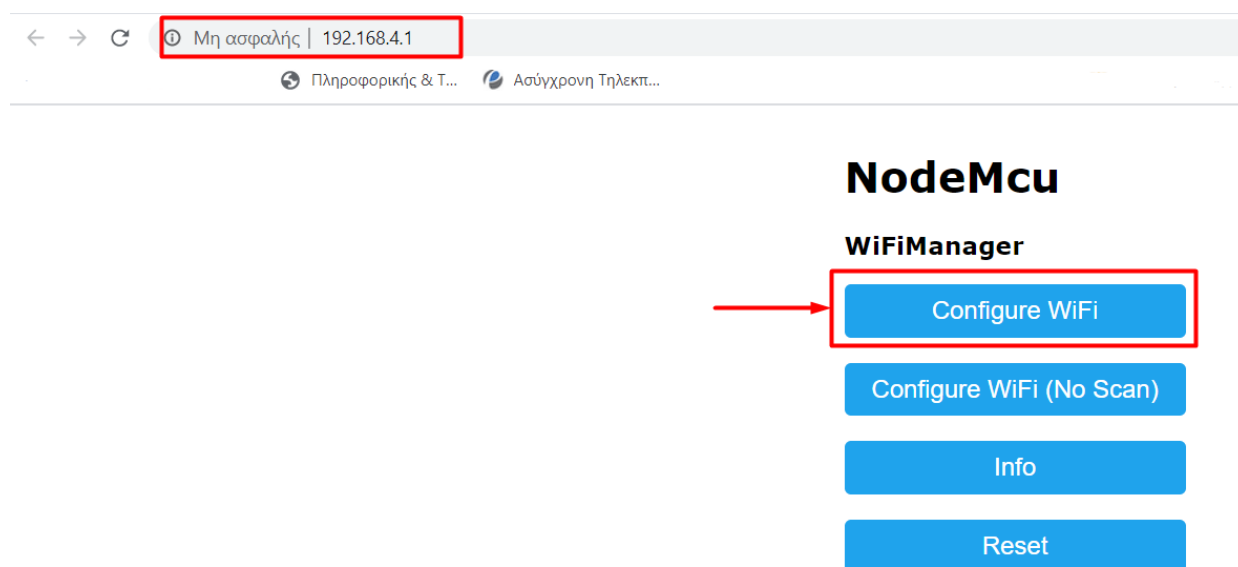
5.4. Λειτουργία συστήματος

Κατά την εκκίνηση, ο μετεωρολογικός σταθμός μας μπαίνει σε κατάσταση access point. Για να μπορέσουμε να συνδεθούμε στο interface της συσκευής και να παραμετροποιήσουμε το δίκτυο μας πρέπει να συνδεθούμε μέσω του υπολογιστή μας στο WiFi του node. Το όνομα ssid της συσκευής το ορίζουμε μέσω της βιβλιοθήκης WiFiManager.h. Μόλις συνδεθούμε, αυτόματα θα ανοίξει το interface σε έναν browser.



Εικόνα 26 Σύνδεση στο NodeMcu

Στο interface που θα ανοίξει πατάμε Configure WiFi.



Εικόνα 27 configure WiFi

Στη συνέχεια καταχωρούμε τα διαπιστευτήρια του WiFi δικτύου μας και συνδεόμαστε.

Vodafone_wifi	🔒 42%
WIND_2.4G_F13C4F	🔒 40%
COSMOTE_WiFi_Fon	36%
OTE4ACED9	🔒 28%
.LEN.1	🔒 28%
OTE313149	🔒 26%
Vodafone_2.4G-00583	🔒 26%
COSMOTE-9A18CE	🔒 26%
Forthnet-67FFC	🔒 24%
DIRECT-D2-HP DeskJet 4670 series	🔒 24%
WIND_BF3AA4	🔒 24%
XRISTINA KOTRWNIA	🔒 24%
VODAFONE_WIFI_108	🔒 20%
Hol Zte 3	🔒 20%
VODAFONE_5003	🔒 18%
TP-Link_8F6E	🔒 14%

SSID

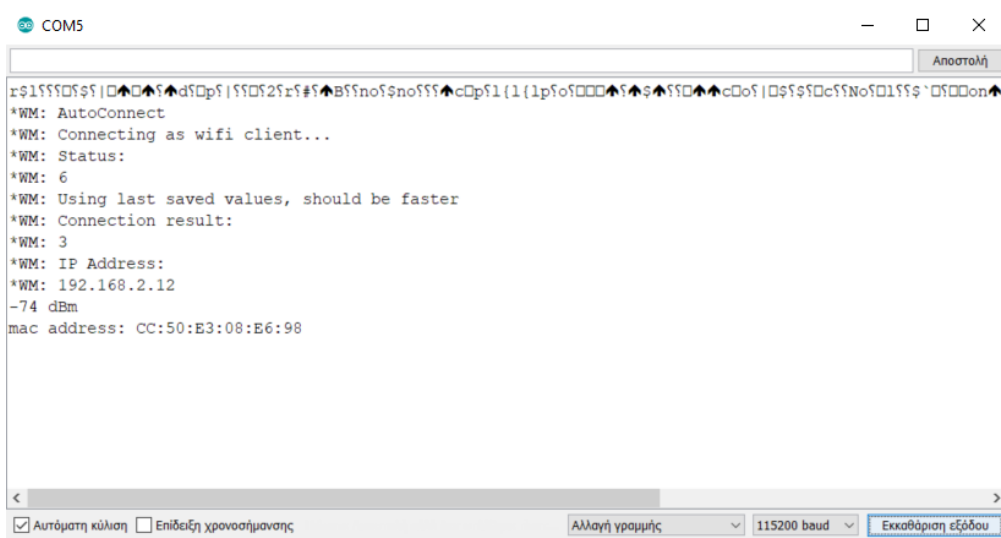
password

save

[Scan](#)

Εικόνα 28 Καταχώρηση Διαπιστευτηρίων WiFi

Αφού γίνει η σύνδεση στο serial monitor του Arduino IDE μπορούμε να δούμε την νέα IP, το RSSI και την mac address που έχει πλέον η συσκευή μας. Κάθε φορά που θα τροφοδοτούμε το σύστημα μας με ρεύμα, χρειάζεται να εκτελείται αυτή η διαδικασία για να συνδεθούμε στο δίκτυο και να ξεκινήσει η λειτουργία του μετεωρολογικού μας σταθμού.

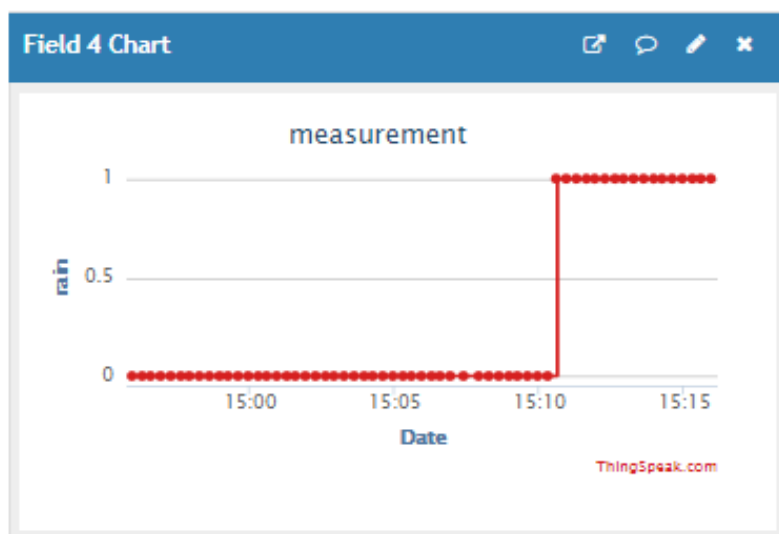


Εικόνα 29 Συνδέθηκε στο δίκτυο που καταχωρήσαμε

Αφού γίνουν οι εκτυπώσεις των στοιχείων στο serial monitor, ο κώδικας μας προχωράει στην συνάρτηση loop. Εκεί ελέγχεται κάθε φορά αν παραμένει συνδεδεμένη στο δίκτυο η συσκευή μας αλλιώς κάνει reset την πλακέτα και ξανά κάνει boot από την αρχή. Όσο ο μετεωρολογικός σταθμός βρίσκεται συνδεδεμένος στο δίκτυο ελέγχει αν ο timer μέτρησε 5 λεπτά. Αν περάσουν 5 λεπτά παίρνει μετρήσεις από όλους τους αισθητήρες και μεταδίδει τα μηνύματα σε κάθε field της πλατφόρμας ThingSpeak. Αλλιώς συνεχίζει μέσα στη loop ελέγχοντας αν υπάρχει σύνδεση στο WiFi και αν ο timer κάνει tick. Για τα αισθητήρια dht11 και rain sensor υπάρχουν συγκεκριμένοι έλεγχοι για την αποφυγή σφαλμάτων. Για παράδειγμα αν η μέτρηση που θα παράγει ο DHT11 είναι NaN (Not a Number) έχουμε προγραμματίσει τον αισθητήρα να μας δίνει τιμές -100 ως ένδειξη σφάλματος. Όστε στα γραφήματα να γνωρίζουμε πότε δεν πήραμε σωστή μέτρηση. Επιπλέον για τον rain sensor έχουμε περιορίσει την ευαισθησία του αισθητήρα ώστε να μην δείξει στο γράφημα ότι βρέχει με μία σταγόνα νερό. Μετά από δοκιμές παρατηρήθηκε ότι πρέπει να καλυφθεί περίπου το 70% της επιφάνειας του αισθητήρα για να μας δείξει ότι βρέχει. Στις παρακάτω εικόνες φαίνονται τα γραφήματα του μετεωρολογικού σταθμού ύστερα από ένα σύνολο μετρήσεων που πήραμε.

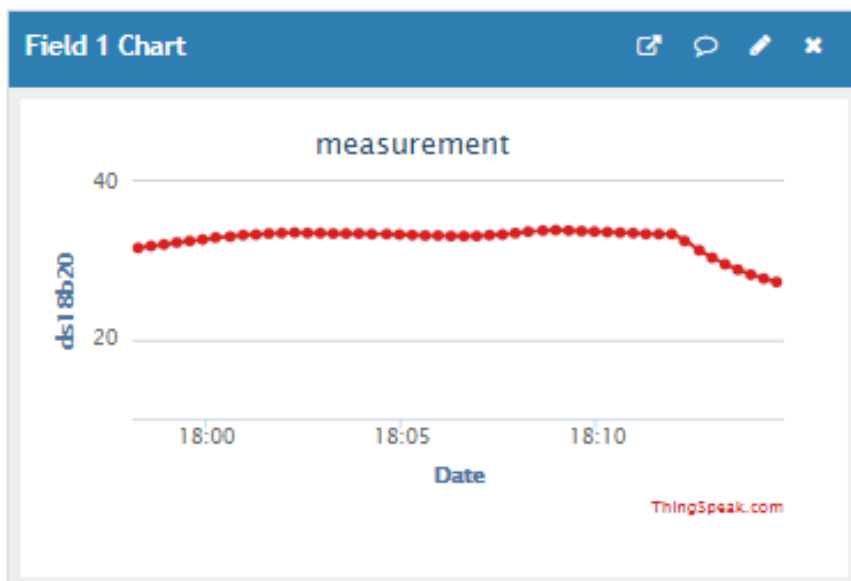
5.5. Παρατηρήσεις Γραφημάτων μετεωρολογικού σταθμού

Το παρακάτω γράφημα απεικονίζει τον αισθητήρα βροχής σε συναρτήσει με τον χρόνο. Φαίνεται σαν ένας τετραγωνικός παλμός αφού οι τιμές που του έχουμε δώσει είναι 0 και 1. Δηλαδή βρέχει ή δεν βρέχει. Επομένως παρατηρούμε ότι στις 15:10 ξεκίνησε να βρέχει.



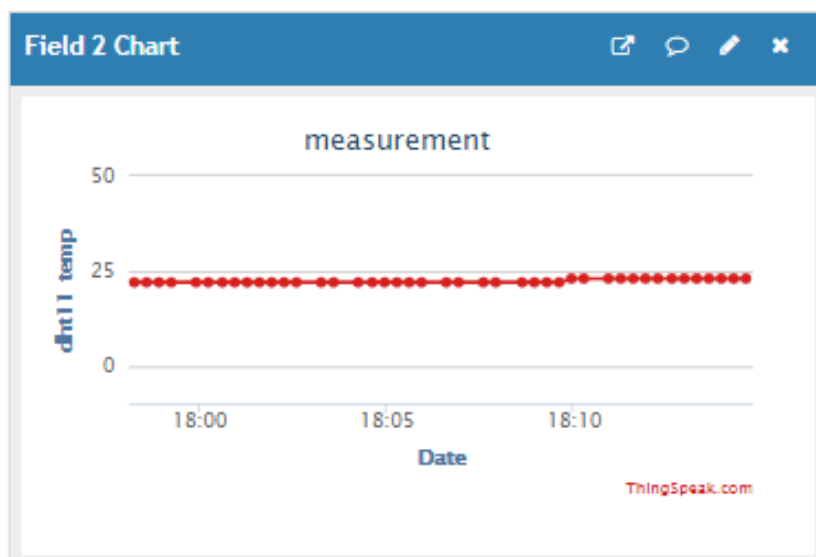
Εικόνα 30 Γράφημα rain drop sensor

Η παρακάτω γραφική παράσταση απεικονίζει την θερμοκρασία του αισθητήρα ds18b20 waterproof σε συναρτήσει με τον χρόνο. Παρατηρούμε ότι για τα όρια που επιλέξαμε σε μία απογευματινή ώρα η θερμοκρασία ήταν σταθερή στους 33 βαθμούς κελσίου ενώ μετά τις 18:10 άρχισε να μειώνεται γραμμικά φτάνοντας στους 27 βαθμούς.

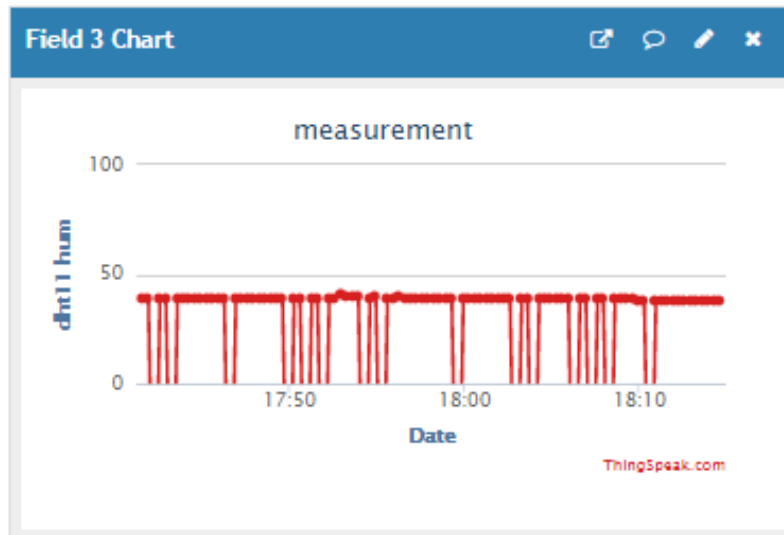


Εικόνα 31 Γράφημα ds18b20 waterproof temperature

Στις παρακάτω εικόνες παρατηρούμε τα γραφήματα του αισθητήρα dht11 θερμοκρασίας και υγρασίας που έχουμε ενσωματώσει στον μετεωρολογικό μας σταθμό. Και εδώ οι άξονες των γραφημάτων είναι θερμοκρασία σε συναρτήσει χρόνου και υγρασία σε συναρτήσει χρόνου. Στο πρώτο (εικόνα 32) παρατηρούμε ότι για το διάστημα που επιλέξαμε η θερμοκρασία είναι σταθερή με ελάχιστες αυξομειώσεις περίπου στους 23 βαθμούς κελσίου και στο γράφημα της υγρασίας παρατηρούμε ότι η υγρασία κυμαίνεται γύρω στο 40%. Ωστόσο οι κατακόρυφες γραμμές που καταλήγουν υπό το μηδέν όπως διακρίνουμε στο γράφημα είναι επειδή οι συγκεκριμένες μετρήσεις είναι ενδείξεις σφάλματος και η τιμή που έχουν πάρει είναι -100.



Εικόνα 32 Γράφημα dht11 θερμοκρασίας



Εικόνα 33 Γράφημα dht11 υγρασίας

6. Συμπεράσματα

Σκοπός της παρούσας εργασίας ήταν η σχεδίαση και κατασκευή ενός μετεωρολογικού σταθμού που θα παράγει μετρήσεις θερμοκρασίας, υγρασίας και ενδείξεις βροχής και θα τις απεικονίζει σε μία πλατφόρμα IoT όπως είναι το ThingSpeak για ανάλυση των δεδομένων και περαιτέρω μελέτη για το περιβαλλοντολογικό κλίμα της περιοχής που βρίσκεται εγκατεστημένος ο μετεωρολογικός σταθμός. Επιπλέον έγιναν παρατηρήσεις μέσω των γραφημάτων και διαπιστώθηκε η ακρίβεια αλλά και η απόδοση των αισθητηρίων σε συναρτήσει με τον ρυθμό σφαλμάτων. Επιπλέον χρησιμοποιήθηκε πλατφόρμα IoT για να μπορούμε απομακρυσμένα μέσω του διαδικτύου να ελέγχουμε και να αναλύουμε τα δεδομένα. Σαν μελλοντικές επεκτάσεις της εργασίας, θα μπορούσε ο μετεωρολογικός σταθμός να εξοπλιστεί με περισσότερα αισθητήρια ώστε να καλύψει περισσότερες ανάγκες αλλά και να μελετηθούν περισσότερα στοιχεία ενός κλίματος όπως η ταχύτητα του ανέμου, η ατμοσφαιρική πίεση, η βαρομετρική πίεση, η καθαρότητα της ατμόσφαιρας από διάφορα αέρια και τι ποσοστό σκόνης υπάρχει. Τέλος μία σημαντική μελλοντική επέκταση θα ήταν η μελέτη ενεργειακής αυτονομίας του μετεωρολογικού σταθμού και η εγκατάσταση μπαταριών και φωτοβολταϊκού πάνελ για την πλήρη αυτόνομη λειτουργία του.

Βιβλιογραφία

- [1] Julian W. Gardner (1994). *ΜΙΚΡΟΑΙΣΘΗΤΗΡΕΣ ΑΡΧΕΣ ΚΑΙ ΕΦΑΡΜΟΓΕΣ*. John Wiley & Sons Ltd
- [2] Υπουργείο Εθνικής Παιδείας και Θρησκευμάτων Κέντρο Περιβαλλοντικής Εκπαίδευσης Μακρινίτσας, Εισαγωγή στη Μετεωρολογία, Μια εκπαιδευτική προσέγγιση Μέρος Α΄: Βασικές Έννοιες, Δεκέμβριος 2007
- [3]https://el.wikipedia.org/wiki/%CE%9C%CE%B5%CF%84%CE%B5%CF%89%CF%81%CE%BF%CE%BB%CE%BF%CE%B3%CE%B9%CE%BA%CF%8C%CF%82_%CF%83%CF%84%CE%B1%CE%B8%CE%BC%CF%8C%CF%82
- [4] Δασυγένης Μ. & Σουντρής Δ. (2015). *Ενσωματωμένα συστήματα Ο αθέατος ψηφιακός κόσμος*. Ελληνικά Ακαδημαϊκά Ηλεκτρονικά Συγγράμματα και Βοηθήματα
- [5] Παπάζογλου Π. & Λιώνης Π. (2015). *Ανάπτυξη εφαρμογών με το Arduino*. Θεσσαλονίκη: Εκδόσεις Τζιόλα
- [6] Καλοφώλιας Δ. (2017). Ο προγραμματισμός του μικροελεγκτή AVR ATmega328 με την χρήση της πλατφόρμας ARDUINO. Θεσσαλονίκη: Εκδόσεις Τζιόλα.
- [7] https://www.handsontec.com/pdf_learn/esp8266-V10.pdf
- [8] https://repository.kallipos.gr/pdfviewer/web/viewer.html?file=/bitstream/11419/1026/1/05_chapter_02.pdf
- [9] https://repository.kallipos.gr/bitstream/11419/2264/3/08_kefalaio8.pdf
- [10] <https://thingspeak.com/>
- [11] <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- [12]<https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>
- [13] <https://components101.com/sensors/rain-drop-sensor-module>
- [14] <https://app.diagrams.net/>
- [15] <https://fritzing.org/home/>

Για τον προγραμματισμό:

- [16] <https://www.arduino.cc/reference/en/language/functions/math/map/>
- [17] <https://github.com/esp8266/Arduino/blob/master/libraries/Ticker/src/Ticker.h>
- [18] <https://github.com/tzapu/WiFiManager>
- [19] <https://github.com/mathworks/thingspeak-arduino/blob/master/src/ThingSpeak.h>
- [20] <https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/station-class.html>
- [21] <https://www.arduinolibraries.info/libraries/one-wire>
- [22] <https://github.com/milesburton/Arduino-Temperature-Control-Library>
- [23] <https://github.com/adafruit/DHT-sensor-library>

Παράρτημα Κώδικα

```
// ----- Libraries -----
#include "ThingSpeak.h" // Μέσω της βιβλιοθήκης thingspeak στέλνουμε τις μετρήσεις στην
πλατφόρμα ThingSpeak.
#include "DHT.h" // Βιβλιοθήκη για την χρήση του αισθητήρα DHT11 .
#include <DallasTemperature.h> // Βιβλιοθήκη για την χρήση του αισθητήρα ds18b20 waterproof
sensor.
#include <OneWire.h> // Μέσω της βιβλιοθήκης oneWire γίνεται η επικοινωνία του ds18b20 με τον
μικροελεγκτή.
#include <ESP8266WiFi.h> // Βασική βιβλιοθήκη για την λειτουργία του ESP8266EX. Μέσω αυτής
λειτουργούν όλες οι συναρτήσεις για το WiFi δίκτυο.
#include <Ticker.h> // Η βιβλιοθήκη αυτή αρχικοποιεί και ενεργοποιεί τον timer του μικροελεγκτή.
// Βιβλιοθήκες για την χρήση του WiFiManager. Δηλαδή για το configuration του access point.
#include <DNSServer.h>
#include <ESP8266WebServer.h>
#include <WiFiManager.h>

//----- Define -----
#define DHTPIN 0 // Σύνδεση δεδομένων στο ψηφιακό pin D3(GPIO 0) για τον sensor DHT11.
#define ONE_WIRE_BUS 4 // Σύνδεση δεδομένων στο ψηφιακό pin D2 (GPIO 4) για τον sensor
ds18B20.
#define DHTTYPE DHT11
#define sensor_rain A0 // Σύνδεση του αισθητήρα βροχής στο αναλογικό pin A0.

// ----- Objects -----
OneWire oneWire (ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
DHT dht (DHTPIN, DHTTYPE);
WiFiClient client;
WiFiManager wifiManager;
Ticker timer;

//----- Variables -----
float dht_temperature; // Μεταβλητή για την θερμοκρασία του dht11.
float dht_humidity; // Μεταβλητή για την υγρασία του dht11.
float ds18b20_temperature; // Μεταβλητή για την θερμοκρασία του ds18B20.
const int sensor_rain_min = 0; // Μεταβλητή για τον καθορισμό των ορίων του αναλογικού αισθητήρα
βροχής.
const int sensor_rain_max = 1023; // Μεταβλητή για τον καθορισμό των ορίων του αναλογικού
αισθητήρα βροχής.
int rain; // Μεταβλητή για τον αισθητήρα βροχής
unsigned long myChannelNumber = 1036921; // ορισμός διαπιστευτηρίων για την σύνδεση και την
αποστολή δεδομένων στην πλατφόρμα ThingSpeak.
const char * myWriteAPIKey = "W31TVDIPE7A48MN7"; // ορισμός διαπιστευτηρίων για την
σύνδεση και την αποστολή δεδομένων στην πλατφόρμα ThingSpeak.
int flag_timer; // Μεταβλητή για τον timer.
int rssi; // Μεταβλητή για την απόδοση του λαμβανόμενου σήματος WiFi.

//----- Setup -----
```

/* Στην setup () γίνονται οι αρχικοποιήσεις των συναρτήσεων των αισθητήρων αλλά και η εκκίνηση της σειριακού πρωτοκόλλου για να δούμε τα δεδομένα στο serial monitor του arduino IDE.

Επιπλέον σε αυτή την συνάρτηση ξεκινάει το σετάρισμα του WiFi δικτύου και η συσκευή μας μπαίνει σε κατάσταση access point για να συνδεθεί σε κάποιο δίκτυο.

Αφού συνδεθεί θα δούμε στο serial monitor του arduino IDE την ip, το rssi και την mac address της συσκευής.

Στη συνέχεια γίνεται η αρχικοποίηση του timer ξεκινώντας να μετράει μέχρι να φτάσει στον χρόνο που έχουμε ορίσει για να κάνει την διακοπή.

*/

```
void setup() {
  Serial.begin(115200); // Αποστολή δεδομένων μέσω του σειριακού πρωτοκόλλου για να
  μπορούμε να δούμε τα αποτελέσματα στο serial monitor.
  if (!wifiManager.startConfigPortal("NodeMcu")) // Σε αυτή την γραμμή μπαίνει σε κατάσταση
  access point η συσκευή μας για να συνδεθούμε σε ένα δίκτυο. Αν δεν μπορέσει κάνει reset την πλακέτα
  και ξεκινάει από την αρχή.
  {
    Serial.println("failed to connect and hit timeout");
    delay(3000); // καθυστέρηση 3 δευτερόλεπτα.
    ESP.reset();
  }
  dht.begin(); // Λειτουργία αναζήτησης των συνδεδεμένων αισθητηρίων dht11
  sensors.begin(); // Λειτουργία αναζήτησης των συνδεδεμένων αισθητηρίων ds18B20
  ThingSpeak.begin(client); // Εκκίνηση του thingspeak μέσω του αντικειμένου client.
  timer.attach(300, flag); // Η αρχικοποίηση του timer αποτελείται από 2 παραμέτρους, τον
  χρόνο(second) που θα κάνει το tick και την συνάρτηση που θα καλεστεί όταν φτάσει τα 300 sec.
  rssi = WiFi.RSSI(); // Μας δίνει την απόδοση του λαμβανόμενου σήματος και το καταχωρεί στην
  μεταβλητή rssi.
  Serial.print(rssi); // Εξάγουμε το αποτέλεσμα του rssi στο serial monitor.
  Serial.println(" dBm");
  Serial.print("mac address: ");
  Serial.println(WiFi.macAddress()); // Εξάγουμε απευθείας την συνάρτηση που μας εμφανίζει την
  mac address στο serial monitor.
}
```

// ----- Loop -----

/* Η loop () είναι η συνάρτηση στην οποία εκτελείται ο ατέρμων βρόχος. Δηλαδή αν μπει σε αυτή την συνάρτηση θα εκτελείται συνεχώς. Σε αυτή την συνάρτηση

αρχικά γίνεται έλεγχος αν συνεχίζουμε να είμαστε συνδεδεμένοι στο wifi δίκτυο αλλιώς κάνει reset την πλακέτα και ξεκινάει το πρόγραμμα από την αρχή.

Στη συνέχεια ελέγχουμε αν το flag_timer = 1 δηλαδή αν ήρθε η ώρα να πάρουμε τις μετρήσεις των αισθητήρων και να τις στείλουμε στην πλατφόρμα ThingSpeak.

Τέλος έχουμε ορίσει στον κώδικα μας να βλέπουμε τα αποτελέσματα και στο serial monitor.

*/

```
void loop ()
{
  int send_thingspeak, rain_drop, test;
  if (WiFi.status () == WL_CONNECTION_LOST) // Αν χαθεί η σύνδεση του WiFi κάνει reset την
  πλακέτα, αλλιώς συνέχισε στην επόμενη if.
  {
    ESP.reset();
  }
}
```

```

}
if (flag_timer == 1) // Αν το flag_timer γίνει 1 λαμβάνει όλες τις μετρήσεις και τις στέλνει στην
πλατφόρμα στο τέλος ξανά μηδενίζει το flag_timer.
{
  sensors.requestTemperatures(); // Μετατροπή μέτρησης για τον sensor ds18b20.
  ds18b20_temperature = sensors.getTempCByIndex(0); // Λήψη της μέτρησης θερμοκρασίας
ds18b20 στην μεταβλητή ds18b20_temperature.
  dht_temperature = dht.readTemperature(); // Λήψη μέτρησης θερμοκρασίας στην μεταβλητή
dht_temperature.
  dht_humidity = dht.readHumidity(); // Λήψη μέτρησης υγρασίας στην μεταβλητή dht_humidity.
  delay(20);
  // Έλεγχος για τον αισθητήρα dht11, αν προκύψει σφάλμα σε μία απ τις δύο μετρήσεις. Αν υπάρξει
σφάλμα στέλνει την τιμή -100 ως ένδειξη λάθους.
  if (isnan(dht_humidity) || isnan(dht_temperature))
  {
    dht_temperature = -100;
    dht_humidity = -100;
  }
  rain = analogRead(sensor_rain); // Διαβάζει την αναλογική τιμή του αισθητήρα βροχής.
  rain = map(rain, sensor_rain_min, sensor_rain_max, 0, 10); // Μέσω της συνάρτησης map
μικραίνουμε τα διαστήματα, δηλαδή από 0 - 1023 σε 0 - 10.
  if (rain <= 5 && rain >= 0) // Αν η τιμή που διάβασε ο αισθητήρας είναι από 0 έως 5 σημαίνει ότι
βρέχει, άρα το rain_drop γίνεται 1.
  {
    rain_drop = 1;
  }
  if (rain <= 10 && rain >= 6) // Αν η τιμή που διάβασε ο αισθητήρας είναι από 6 έως 10 σημαίνει
ότι δεν βρέχει, άρα το rain_drop γίνεται 0.
  {
    rain_drop = 0;
  }
  ThingSpeak.setField(1, ds18b20_temperature); // θέτει την μέτρηση του ds18b20 στο field 1.
  ThingSpeak.setField(2, dht_temperature); // θέτει την μέτρηση του dht11 στο field 2.
  ThingSpeak.setField(3, dht_humidity); // θέτει την μέτρηση του dht11 στο field 3.
  ThingSpeak.setField(4, rain_drop); // θέτει την μέτρηση του rain sensor στο field 4.
  test = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey); // Στέλνει και στα 4 fields
τις μετρήσεις χρησιμοποιώντας την συνάρτηση writeFields. Τα ορίσματα είναι το channel ID και το
API KEY.
  if ( test == 200) // Έλεγχος για το αν στάλθηκε το μήνυμα στο ThingSpeak. Αν επιστρέψει 200
σημαίνει ότι το μήνυμα στάλθηκε επιτυχώς.
  {
    Serial.println();
    Serial.println(" METRSH ESTALH ");
  }
  Serial.print("ds18B20_Temperature: ");
  Serial.print(ds18b20_temperature);
  Serial.println("C");
  Serial.print("DHT_Temperature: ");
  Serial.println(dht_temperature);
  Serial.print("DHT_Humidity: ");

```

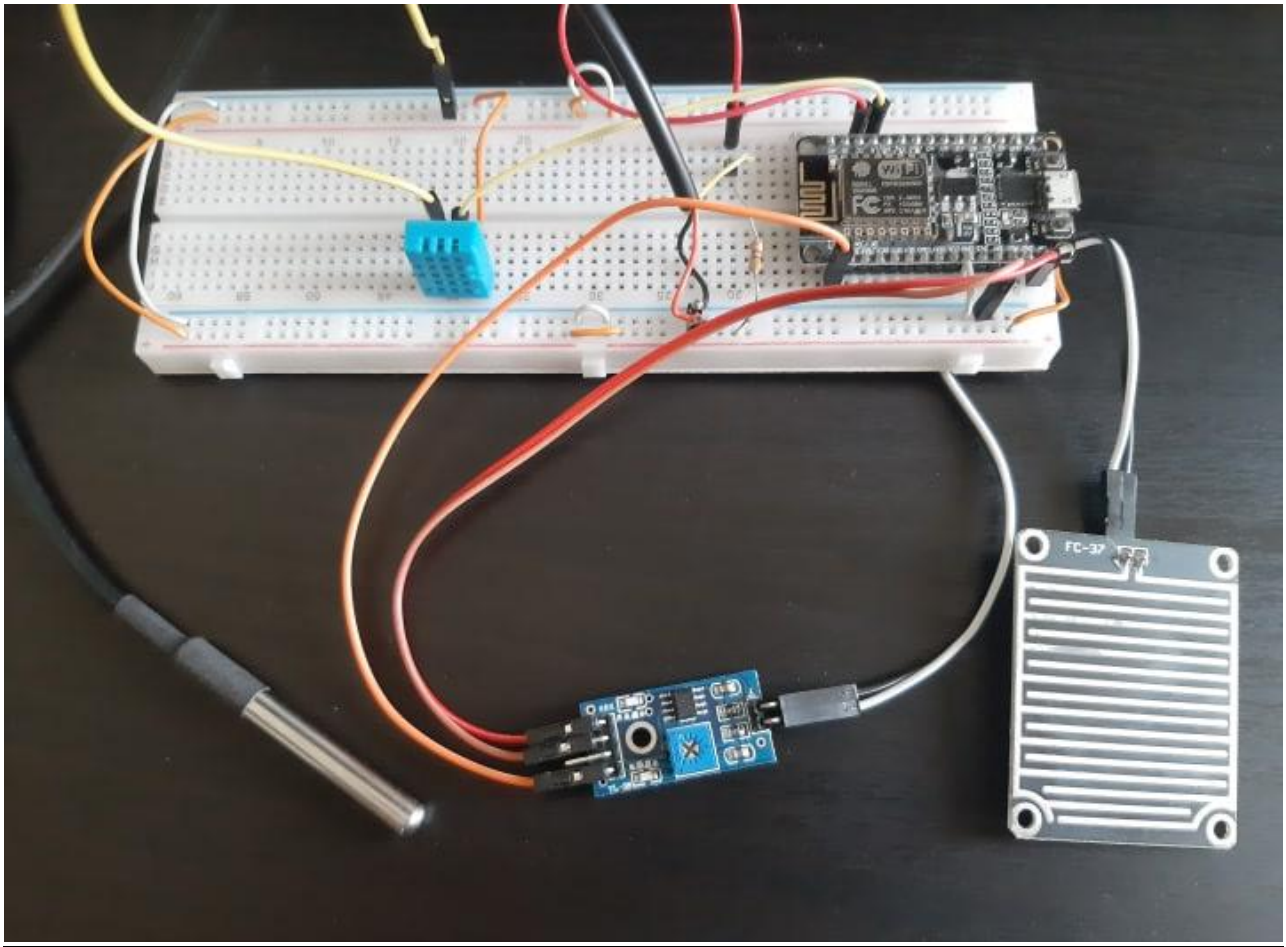
```
Serial.println(dht_humidity);  
Serial.print("rain drop: ");  
Serial.println(rain_drop);  
Serial.println();  
flag_timer = 0; // μηδενίζουμε πάλι το flag_timer.  
}
```

```
}
```

```
//----- flag -----  
// Συνάρτηση που αλλάζει την κατάσταση των μετρήσεων, μέσω του timer. Αν συμβεί η διακοπή από  
// τον timer κάνει την μεταβλητή flag_timer = 1 και μπαίνει στην if στην loop() για να πάρει μετρήσεις.  
void flag()  
{  
  flag_timer = 1;  
}
```

[16], [17], [18], [19], [20], [21], [22]

Παράρτημα εικόνας Μετεωρολογικού Σταθμού



Εικόνα 34 Τελικό αποτέλεσμα του μετεωρολογικού σταθμού