

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**  
**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**

**Αυτόνομος αισθητήρας κραδασμών, με GPS και Cloud Services**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

του

**ΚΩΤΣΙΟΥ ΙΩΑΝΝΗ**

Επιβλέπων : Αγγελίδης Παντελής

ΚΟΖΑΝΗ, ΟΚΤΩΒΡΙΟΣ 2023



**UNIVERSITY OF WESTERN MACEDONIA**  
**DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING**

## **Autonomous vibrations sensor with GPS and Cloud Services**

**THESIS**

**KOTSIOS IOANNIS**

Supervisor: Aggelidis Pantelis

KOZANI, OCTOBER 2023

## ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο: «**Αυτόνομος αισθητήρας κραδασμών, με GPS και Cloud Services**» καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Αγγελίδη Παντελή, αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να δια τηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Copyright© Ονοματεπώνυμο Φοιτητή & Επιβλέποντα, Έτος, Πόλη

Copyright© Κώτσιος Ιωάννης , Αγγελίδης Παντελής, 2023, Κοζάνη

Υπογραφή φοιτητή

ΚΩΤΣΙΟΣ ΙΩΑΝΝΗΣ

## Περίληψη

Η παρούσα διπλωματική αποτελεί μια πρακτική εφαρμογή του μικροεπεξεργαστή ESP8266 και διάφορων αισθητήρων που αλληλοεπιδρούν μεταξύ τους. Στη συγκεκριμένη εργασία, παρουσιάζεται η υλοποίηση μιας ολοκληρωμένης πλακέτας που καταγράφει δεδομένα από το περιβάλλον, τα επεξεργάζεται και στη συνέχεια τα ανεβάζει σε μια πλατφόρμα με σκοπό την εύκολη κατανόηση για τον τελικό χρήστη. Όλη αυτή η διαδικασία πραγματοποιείται χωρίς την παρέμβαση χειριστή, δηλαδή αποτελεί προϊόν έτοιμο για χρήση από οποιονδήποτε άνθρωπο χωρίς ιδιαίτερη εξειδίκευση.

Καθώς η τεχνολογία εξελίσσεται συνεχώς και είναι πλέον αναπόσπαστο κομμάτι της ζωής μας, αναζητούμε τρόπους ώστε να υλοποιήσουμε ιδέες που παλιότερα ήταν αρκετά χρονοβόρες και είχαν υψηλές απαιτήσεις σε ενέργεια.

Το κύριο μέρος της διπλωματικής είναι η δημιουργία ενός αυτόνομου ολοκληρωμένου κυκλώματος το οποίο θα καταγράφει την ποιότητα των δρόμων, θα επεξεργάζεται τα δεδομένα που συλλέγει και τέλος θα ανεβάζει τα αποτελέσματα σε μια βάση δεδομένων. Από εκεί και έπειτα τα δεδομένα αυτά θα είναι διαθέσιμα για απεικόνιση σε μια πλατφόρμα χαρτών της επιλογής μας, δείχνοντας μας τα σημεία του δρόμου όπου χρίζουν προσοχή και επισκευή.

Για την κατασκευή του κυκλώματος χρησιμοποιήθηκαν οικονομικά εξαρτήματα με μεγάλη διαθεσιμότητα, που σημαίνει ότι σε περίπτωση μαζικής παραγωγής το κόστος θα είναι χαμηλό. Επίσης, ανάλογα με τις ανάγκες του πελάτη, το λογισμικό μπορεί να διαμορφωθεί ώστε να συλλέγονται επιπλέον δεδομένα τα οποία είναι δυνατό να καταγραφούν μέσω των αισθητήρων.

**Λέξεις-κλειδιά:** ESP8266, αισθητήρες, πλακέτα, περιβάλλον, ενέργεια, τεχνολογία, καταγραφή δεδομένων

## **Abstract**

This thesis is a practical application of ESP8266 microprocessor and various sensors that interact with each other. In this particular task, the creation of a completed circuit is displayed, which captures data from the environment, it processes them and finally uploads them to a platform so that the final user can easily understand the results. This whole process doesn't need any intervention, it is a product ready to be deployed and used by anyone without and particular knowledge.

As technology is constantly being developed and is a part of our life, we search ways to implement ideas that used to be time consuming and had high energy demands.

The main part of this thesis is the creation of an autonomous completed circuit which will capture road quality, it will process the gathered data and finally it will upload the results to a database. From thereafter these data will be available for imaging to a maps platform of our choice, indicating parts of the road that need caution and repair.

For manufacturing the circuit, economical parts with high availability were used, which means that in case of mass manufacturing the cost would be low. Furthermore, the software can be modified according to the client's needs, to collect extra data which are available via the existing sensors.

**Keywords:** ESP8266, sensors, circuit board, environment, energy, technology, data logging

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, Δρ. Παντελή Αγγελίδη για την εμπιστοσύνη και την υποστήριξη που μου έδειξε ώστε να καταφέρω να σας παρουσιάσω το τελικό αποτέλεσμα της ιδέας η οποία κατάφερε να γίνει πράξη.

Επίσης, ένα μεγάλο ευχαριστώ στα κοντινά μου άτομα για την υποστήριξη που μου παρείχαν έτσι ώστε να είμαι σωστός απέναντι στις υποχρεώσεις μου και να φέρω εις πέρας αυτό το αποτέλεσμα.

**ΑΥΤΗ Η ΣΕΛΙΔΑ ΕΙΝΑΙ ΣΚΟΠΙΜΑ ΚΕΝΗ**

# Περιεχόμενα

Περίληψη.....	4
Abstract.....	5
Ευχαριστίες.....	6
Περιεχόμενα.....	8
Κατάλογος εικόνων.....	10
<b>1. Εισαγωγή.....</b>	<b>11</b>
1.2 Στόχος της εργασίας.....	11
<b>2. Θεωρητικό Υπόβαθρο.....</b>	<b>12</b>
2.1 ESP8266.....	12
2.2 Arduino IDE.....	13
2.3 ThingSpeak IoT (Internet of Things).....	16
2.4 Τα βασικά μέρη του κυκλώματος.....	17
2.4.1 NodeMCU V2 πλακέτα ανάπτυξης.....	17
2.4.2 GPS Neo-6M.....	18
2.4.3 Επιταχυνσιόμετρο MPU-6050.....	19
2.4.4 Αναγνώστης καρτών τύπου microSD.....	20
2.4.5 Ηχείο τύπου Buzzer.....	20
2.4.6 Μπαταρία λιθίου 18650.....	21
2.4.7 TP4056 μονάδα φόρτισης μπαταρίας.....	21
2.4.8 Μετατροπέας τάσης (ενισχυτής σταθερής τάσης).....	21
2.4.9 Διακόπτης τροφοδοσίας.....	22
<b>3. Από τη θεωρία στην πράξη.....</b>	<b>22</b>
3.1 Σχεδιάγραμμα.....	22
3.2 Εγκατάσταση σε πλακέτα δοκιμών.....	24
3.3 Πρωτότυπη πλακέτα για εξωτερικές δοκιμές.....	25
3.4 Τελική πλακέτα κυκλώματος.....	27
<b>4. Ανάλυση λειτουργίας κυκλώματος.....</b>	<b>30</b>
4.1 Σύντομη περιγραφή λειτουργίας.....	30
4.2 Ανάλυση κώδικα.....	31
4.2.1 Βιβλιοθήκες.....	31
4.2.2 Global μεταβλητές.....	32
4.2.3 void setup().....	33



4.2.4 void loop().....	37
<b>5. Αποτελέσματα .....</b>	<b>38</b>
<b>6. Μελλοντικά σχέδια .....</b>	<b>41</b>
<b>7. Επίλογος.....</b>	<b>41</b>
<b>Βιβλιογραφίες.....</b>	<b>44</b>
<b>Απόδοση ξενόγλωσσων όρων .....</b>	<b>45</b>

## Κατάλογος εικόνων

Εικόνα 1. Το τσιπ 8266EX .....	12
Εικόνα 2. Το ESP-12e.....	13
Εικόνα 3. Arduino IDE, αρχική σελίδα .....	14
Εικόνα 4. Arduino IDE, διαχείριση πλακετών .....	15
Εικόνα 5. Arduino IDE, διαχείριση βιβλιοθηκών.....	16
Εικόνα 6. NodeMCU V2 Development Board .....	18
Εικόνα 7. Ublox NEO-6M GPS.....	18
Εικόνα 8. Επιταχυνσίόμετρο MPU-6050.....	19
Εικόνα 9. Αναγνώστης καρτών τύπου microSD .....	20
Εικόνα 10. Ηχείο τύπου buzzer .....	20
Εικόνα 11. Μπαταρία λιθίου 18650.....	21
Εικόνα 12. Μονάδα φόρτισης μπαταρίας TP4056 .....	21
Εικόνα 13. Μετατροπέας σταθερής τάσης ρεύματος .....	22
Εικόνα 14. Διακόπτης τροφοδοσίας .....	22
Εικόνα 15. Οπτικό σχεδιάγραμμα μέσω του λογισμικού Fritzing.....	23
Εικόνα 16. Σχηματικός πίνακας κυκλώματος.....	23
Εικόνα 17. Εγκατάσταση των εξαρτημάτων σε πλακέτα δοκιμών.....	24
Εικόνα 18. Δημιουργία συνδέσεων με καλώδιο μικρής διατομής.....	25
Εικόνα 19. Τα εξαρτήματα κουμπώνουν στους θηλυκούς ακροδέκτες για δοκιμές .....	26
Εικόνα 20. Η προσωρινή πλακέτα είναι στο στάδιο των δοκιμών και αποσφαλμάτωσης.....	27
Εικόνα 21. Η τελική διάτρητη πλακέτα, τοποθετήθηκαν τα πρώτα εξαρτήματα.....	28
Εικόνα 22. Τοποθετήθηκαν οι θηλυκοί ακροδέκτες και η μπαταρία στο κύκλωμα.....	28
Εικόνα 23. Η πλακέτα ολοκληρωμένη και έτοιμη για χρήση.....	29
Εικόνα 24. Πίσω όψη της ολοκληρωμένης πλακέτας .....	30
Εικόνα 25. Οι βιβλιοθήκες, global μεταβλητές και οι κλάσεις που δημιουργήθηκαν .....	33
Εικόνα 26. Η συνάρτηση αρχικοποίησης void setup.....	33
Εικόνα 27. Οι συναρτήσεις που καλούνται κατά την εκτέλεση του void setup .....	35
Εικόνα 28. Η συνάρτηση uploadDataToThingSpeak .....	36
Εικόνα 29. Η συνάρτηση επανάληψης void loop .....	37
Εικόνα 30. Οι συναρτήσεις που καλούνται μέσα από την void loop .....	38
Εικόνα 31. Αρχική σελίδα του καναλιού στο ThingSpeak.....	39
Εικόνα 32. Χάρτης που απεικονίζει σημεία του δρόμου που χρειάζονται προσοχή.....	40
Εικόνα 33. Έκδοση 0.1, κατασκευασμένη γύρω από το Arduino Uno .....	43

## **1. Εισαγωγή**

Η παρούσα διπλωματική εργασία βασίζεται σε ένα συχνό φαινόμενο που συναντάμε στην καθημερινή μας ζωή, αυτό του κακού οδοστρώματος. Το πρόβλημα είναι ιδιαίτερα αισθητό, τόσο σε κεντρικούς δρόμους μιας πόλης όσο και σε περιοχές εκτός δήμου όπου συνήθως το πρόβλημα αυξάνεται. Σκοπός της εργασίας είναι η κατασκευή ενός ολοκληρωμένου συστήματος (πλακέτα) η οποία θα καταγράφει αυτές τις ανωμαλίες του οδοστρώματος και έπειτα θα αποστέλλει τις πληροφορίες που έχει συλλέξει στον αρμόδιο φορέα (π.χ. Δήμος) ώστε να υπάρξει αναγνώριση των δρόμων που χρίζουν επισκευή. Ένας λόγος που με ώθησε στην κατασκευή της εργασίας είναι η πόλη μου, τα Τρίκαλα. Όντας μια τεχνολογικά εξελιγμένη πόλη που υιοθετεί συνεχώς νέες καινοτόμες ιδέες π.χ. λεωφορεία χωρίς οδηγό, ηλεκτρικά οχήματα, δωρεάν Wi-Fi κτλ. , μπορεί εύκολα να ενσωματώσει αυτή την ιδέα και να την κάνει πράξη. Επιπλέον, ένα μεγάλο ποσοστό των κατοίκων επιλέγουν τα ποδήλατα για την καθημερινή μετακίνηση τους, το οποίο σημαίνει ότι οποιαδήποτε προσπάθεια επίλυσης του προβλήματος του οδοστρώματος θα συμβάλλει στην ποιότητα ζωής των πολιτών.

### **1.2 Στόχος της εργασίας**

Αυτή η εργασία έγινε με στόχο την επίλυση ενός προβλήματος που αντιμετωπίζουμε στην καθημερινότητα μας, εστιάζοντας κυρίως στην πρακτική εφαρμογή και υλοποίηση ενός τελικού προϊόντος. Παράλληλα γίνεται μια προσπάθεια να δοθούν οι βασικές θεωρητικές πληροφορίες για το ESP8266, τους επιμέρους αισθητήρες που χρησιμοποιήθηκαν καθώς και ο κώδικας που καταφέρνει να τα συνδέσει όλα μεταξύ τους και να μας δώσει το επιθυμητό αποτέλεσμα.

## 2. Θεωρητικό Υπόβαθρο

### 2.1 ESP8266

Το ESP8266 είναι ένα χαμηλού κόστους μικροτσιπ, με ενσωματωμένο λογισμικό δικτύου TCP/IP και δυνατότητες μικροελεκτή. Παράγεται από την Espressif Systems. Διαθέτει ενσωματωμένο επεξεργαστή, μνήμη τύπου flash (για την αποθήκευση προγραμμάτων και δεδομένων) και Wi-Fi για ασύρματη σύνδεση στο διαδίκτυο. Υποστηρίζει ποικίλους αισθητήρες και εξοπλισμό με σκοπό την ανάγνωση περιβαλλοντικών συνθηκών, όπως για παράδειγμα θερμοκρασία, κίνηση, υγρασία κτλ. Ο διάυλος επικοινωνίας που μπορεί να χρησιμοποιηθεί μεταξύ των περιφερειακών και του ESP8266 είναι UART, HSPI, I2C, GPIO,



Εικόνα 1. Το τσιπ 8266EX

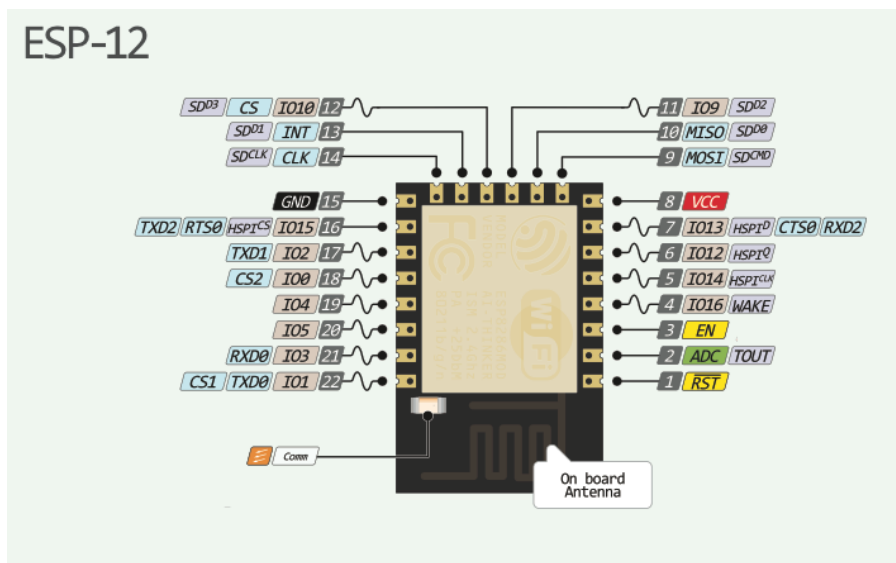
PWM. Μερικές γλώσσες προγραμματισμού που μπορούν να χρησιμοποιηθούν είναι οι C, C++, Python και Node.js.

Το ESP8266 χρησιμοποιείται ευρέως σε:

- **Έξυπνες συσκευές:** π.χ. λάμπες, διακόπτες, πρίζες, οικιακές συσκευές γενικότερα.
- **Ενσωματωμένα Συστήματα:** Συστήματα ελέγχου και παρακολούθησης.
- **Projects διαδικτύου και χόμπι:** Είναι αρκετά δημοφιλές σε ερασιτεχνικά projects που απαιτούν σύνδεση στο διαδίκτυο. Επίσης χρησιμοποιείται για τη δημιουργία πρωτότυπων κατασκευών λόγω της ευκολίας προγραμματισμού του και της συμβατότητας με πολλούς αισθητήρες.
- **Εκπαίδευση:** Χρησιμοποιείται σε σχολεία και πανεπιστημιακά ιδρύματα για να διδάξει τον προγραμματισμό και τα ασύρματα δίκτυα.

Βασικό πλεονέκτημα του ESP8266 είναι η χαμηλή κατανάλωση ενέργειας, που το καθιστά ιδανικό για τροφοδοσία μέσω μπαταριών και η τοποθέτηση σε απομακρυσμένα μέρη όπου η παροχή ρεύματος δεν είναι εφικτή. Επίσης, υποστηρίζεται από μια μεγάλη κοινότητα χρηστών και λόγω του ανοιχτού κώδικα είναι εύκολη η προσαρμογή στις απαιτήσεις του χρήστη και η πλήρης αξιοποίηση των διαθέσιμων πόρων.

Το τσιπ ESP8266EX το συναντάμε σε διάφορες εκδόσεις, συσκευασμένο ανάλογα με τις απαιτήσεις του χρήστη. Η κύρια διαφορά τους είναι ο αριθμός των εισόδων/εξόδων που είναι διαθέσιμες για χρήση. Στην παρούσα διπλωματική χρησιμοποιήθηκε το ESP-12e

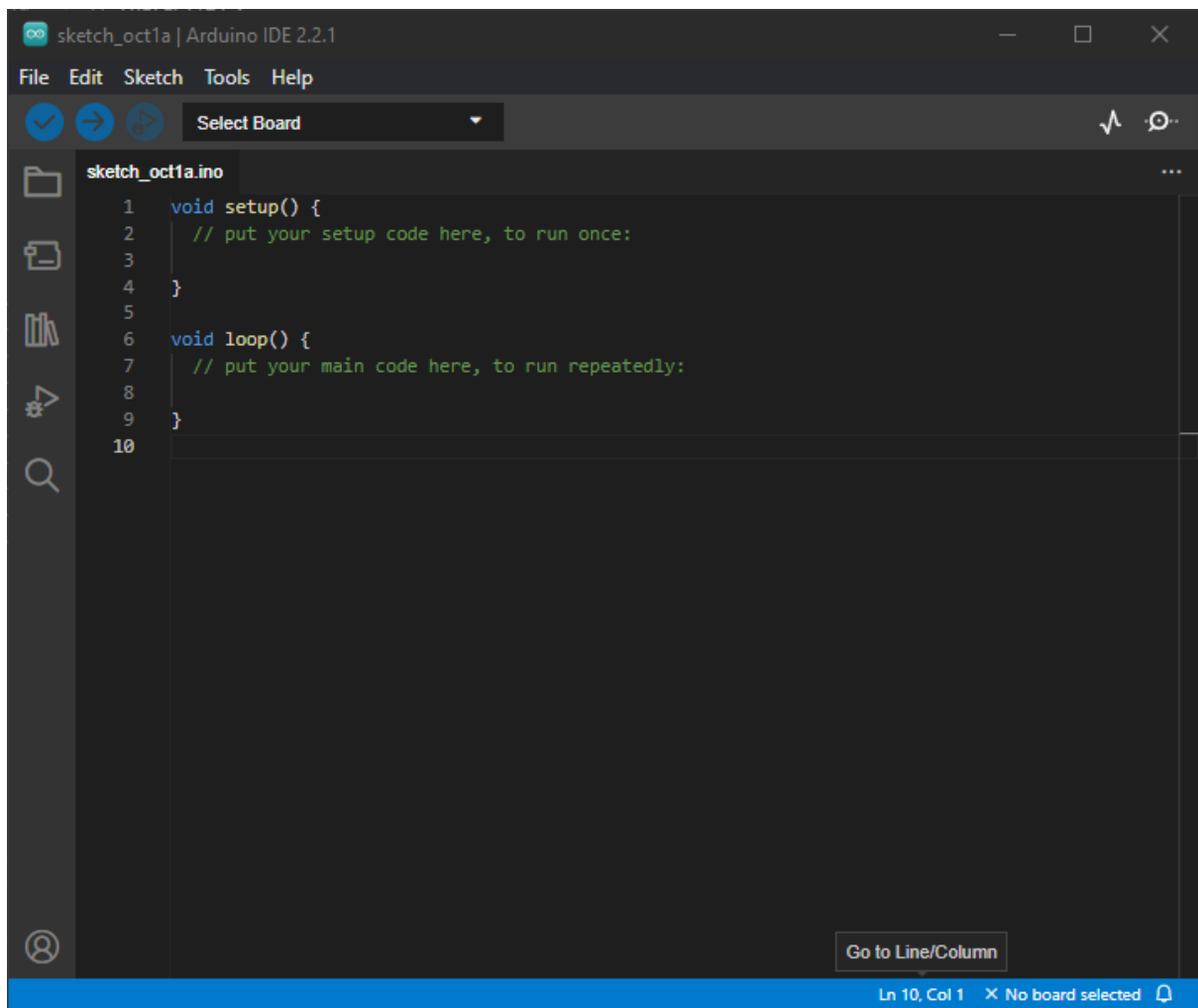


Εικόνα 2. Το ESP-12e

## 2.2 Arduino IDE

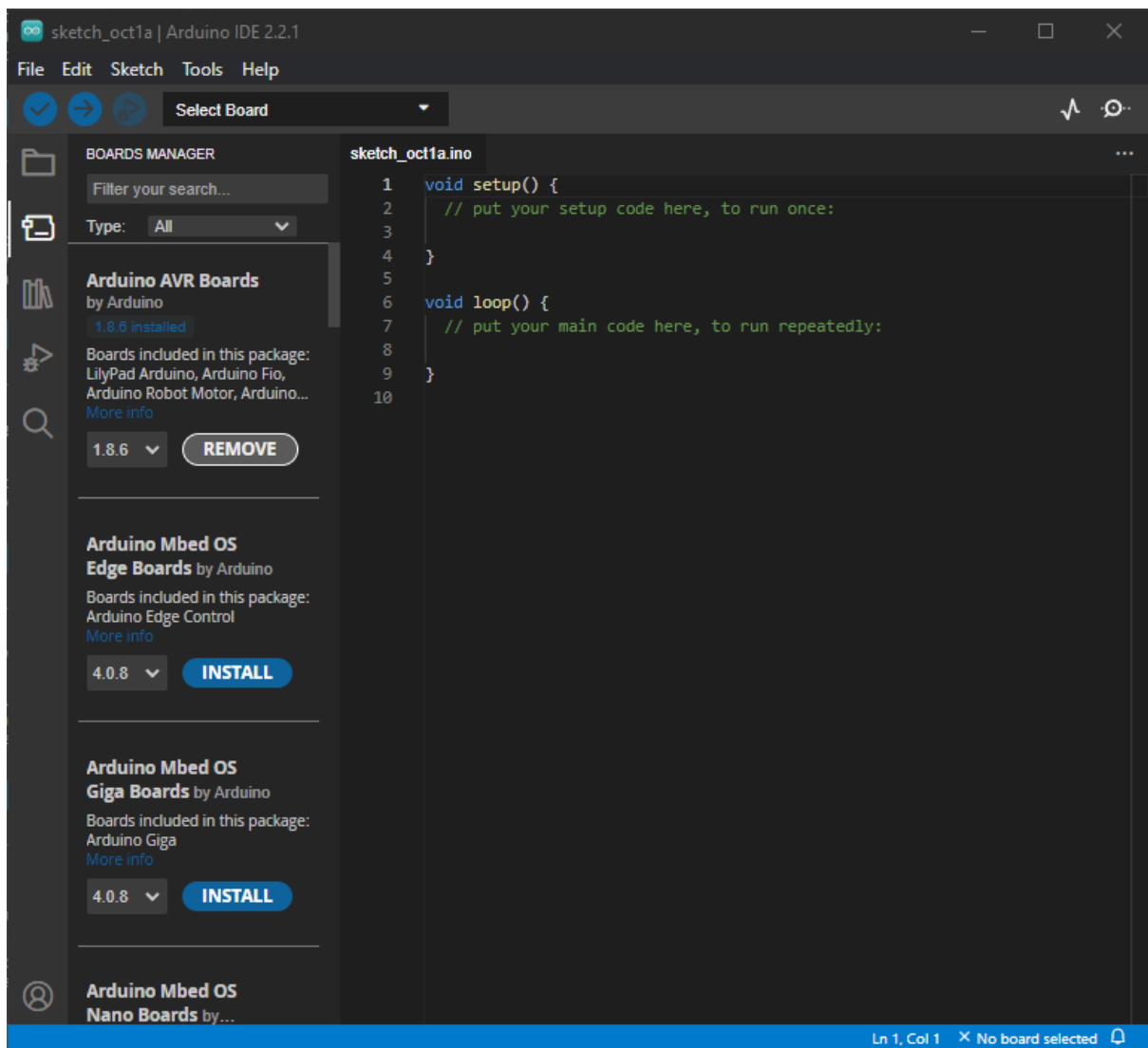
Είναι ένα ανοιχτού κώδικα περιβάλλον ανάπτυξης της εταιρείας Arduino, το οποίο χρησιμοποιείται ευρέως για τον προγραμματισμό σε Arduino, ESP8266, ESP32 και πολλών άλλων πλακετών ανοιχτού κώδικα. Είναι απλό στη χρήση του και φιλικό σε νέους χρήστες που θέλουν να έρθουν σε επαφή με τον προγραμματισμό. Περιέχει επεξεργαστή κειμένου για την εγγραφή κώδικα, περιοχή μηνυμάτων και κονσόλα κειμένου για την επικοινωνία με το ολοκληρωμένο.

Τα προγράμματα που γράφονται μέσω του Arduino IDE ονομάζονται sketches, γράφονται στον επεξεργαστή κειμένου και έχουν την κατάληξη .ino. Ο επεξεργαστής κειμένου παρέχει όλες τις βασικές λειτουργίες για την εύκολη διαμόρφωση του κώδικα μας. Η περιοχή μηνυμάτων που διαθέτει μας ειδοποιεί όταν γίνεται αποθήκευση, εξαγωγή κτλ. Η κονσόλα κειμένου που διαθέτει εμφανίζει το κείμενο εξόδου του λογισμικού καθώς και πλήρες μηνύματα σφαλμάτων και λοιπές πληροφορίες. Στην κάτω δεξιά γωνία αναγράφεται η πλακέτα που έχουμε ρυθμίσει και η σειριακή θύρα που είναι συνδεδεμένη.



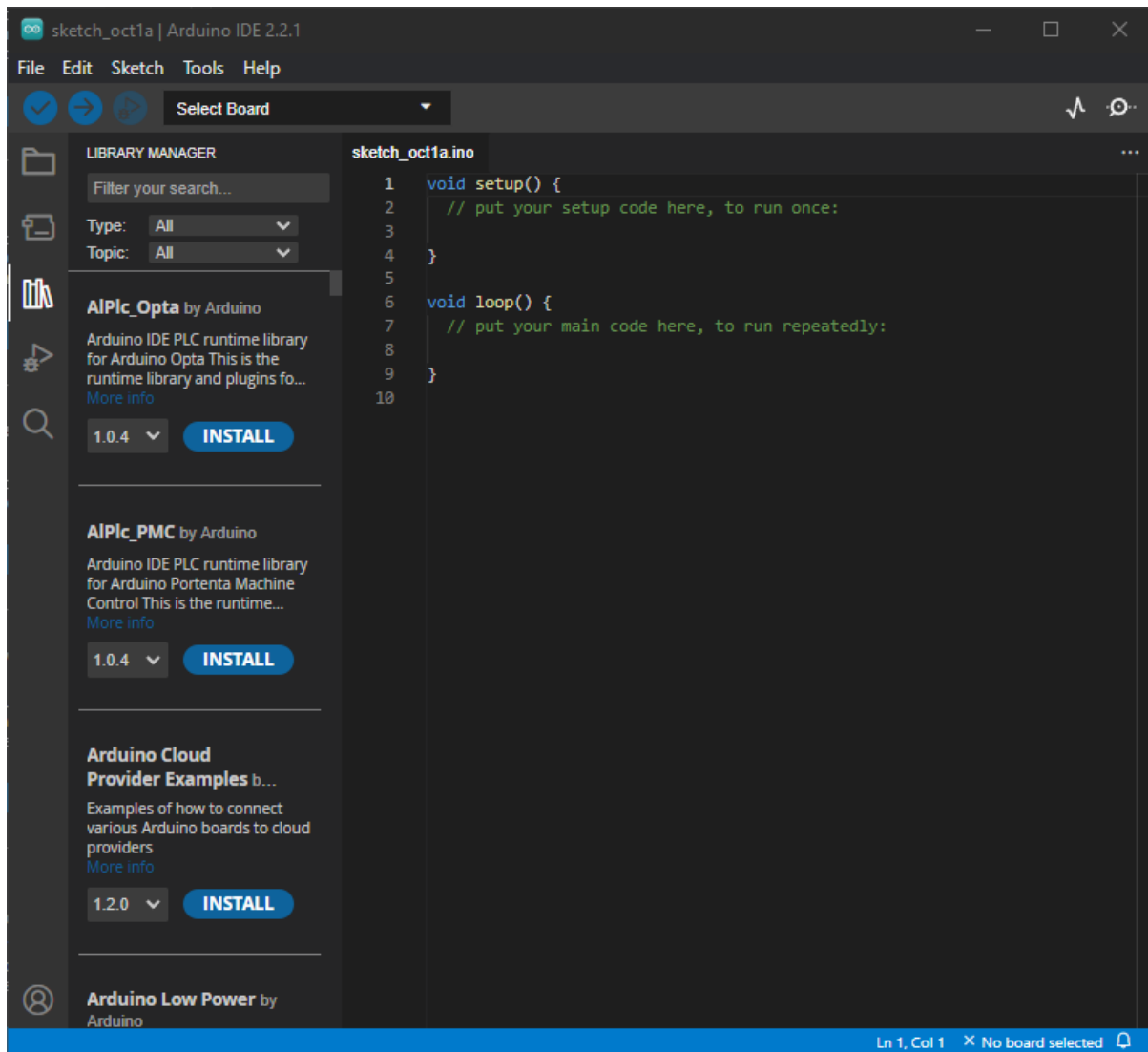
Εικόνα 3. Arduino IDE, αρχική σελίδα

Η εγκατάσταση και η εγγραφή κώδικα σε πλακέτες ανοιχτού κώδικα είναι πολύ εύκολη. Από το αριστερό μενού επιλέγουμε τη διαχείριση πλακετών, εμφανίζεται ένα αναδυόμενο παράθυρο με όλες τις διαθέσιμες πλακέτες που μπορούν να εγκατασταθούν στο πρόγραμμα, όπως φαίνεται στην εικόνα 4.



Εικόνα 4. Arduino IDE, διαχείριση πλακετών

Αφού προσθέσουμε την πλακέτα που επιθυμούμε να προγραμματίσουμε, μπορούμε να ξεκινήσουμε την εγγραφή κώδικα. Ένα πολύ σημαντικό στοιχείο του Arduino IDE είναι οι βιβλιοθήκες που διαθέτει. Αυτές είναι κομμάτια κώδικα τα οποία μας προσφέρουν επιπλέον λειτουργικότητα για χρήση στον κώδικα μας, όπως για παράδειγμα τον έλεγχο του υλικού και τη διαχείριση δεδομένων. Οι βιβλιοθήκες ανεβαίνουν στην πλακέτα μαζί με τον κώδικα μας, αυξάνοντας το χώρο που καταλαμβάνει στη μνήμη, οπότε καλό είναι να προσθέτουμε μόνο όσες μας είναι απαραίτητες για τη σωστή λειτουργία του κώδικα μας και να μην σπαταλάμε επιπλέον αποθηκευτικό χώρο. Γενικά οι βιβλιοθήκες είναι ένα από τα δυνατά σημεία του προγράμματος, επειδή μας επιτρέπει να προσθέσουμε επιπλέον λειτουργίες στον κώδικα μας ή να αλλάξουμε υλικό/αισθητήρες με μικρή επεξεργασία στο κείμενο μας.



Εικόνα 5. Arduino IDE, διαχείριση βιβλιοθηκών

## 2.3 ThingSpeak IoT (Internet of Things)

Το ThingSpeak είναι μια υπηρεσία πλατφόρμας IoT που επιτρέπει στους χρήστες να συλλέγουν, να αναλύουν και να εμφανίζουν δεδομένα από συνδεδεμένες συσκευές. Η υπηρεσία παρέχει μια απλή διεπαφή για τη συγκέντρωση δεδομένων από αισθητήρες και άλλες συσκευές IoT, καθώς και την απεικόνιση σε γραφικές παραστάσεις και αναλύσεις για την παρουσίαση των δεδομένων.

Οι βασικές λειτουργίες του ThingSpeak περιλαμβάνουν:

- **Καταγραφή δεδομένων:**  
Οι χρήστες μπορούν να στέλνουν δεδομένα από αισθητήρες και συσκευές στο ThingSpeak χρησιμοποιώντας αιτήματα HTTP (Hypertext Transfer Protocol) ή MQTT (Message Queuing Telemetry Transport). Το ThingSpeak καταγράφει αυτά τα δεδομένα και τα αποθηκεύει για μελλοντική χρήση.
- **Αναλύσεις και γραφικές παραστάσεις:**



Το ThingSpeak παρέχει εργαλεία για τη δημιουργία γραφικών παραστάσεων και αναλύσεων βάσει των δεδομένων που συλλέγονται. Οι χρήστες μπορούν να δημιουργήσουν γραφήματα, διαγράμματα, και άλλες παραστάσεις για να αναλύσουν τα δεδομένα τους.

- **Ειδοποιήσεις:**  
Οι χρήστες μπορούν να δημιουργήσουν κανόνες που ελέγχουν τα δεδομένα και να λάβουν ειδοποιήσεις (email, SMS) όταν συμβούν συγκεκριμένες συνθήκες.
- **Δημόσια ή ιδιωτική κοινότητα**  
Οι χρήστες μπορούν να επιλέξουν αν θέλουν τα δεδομένα τους να είναι δημόσια ή ιδιωτικά. Αυτό επιτρέπει την κοινή χρήση δεδομένων ή τη διατήρησή τους ιδιωτικά.
- **Ενσωμάτωση:**  
Το ThingSpeak επιτρέπει την ενσωμάτωση των γραφικών παραστάσεων και των δεδομένων σε άλλες ιστοσελίδες ή εφαρμογές μέσω API (Application Programming Interface).
- **Συνδεσιμότητα με διάφορες πηγές δεδομένων:**  
Το ThingSpeak δεν περιορίζεται μόνο στην παραλαβή δεδομένων από συνδεδεμένες συσκευές IoT. Μπορεί επίσης να λαμβάνει δεδομένα από διάφορες πηγές όπως αισθητήρες, αρχεία CSV, αλλά και από άλλες πλατφόρμες μέσω διάφορων πρωτοκόλλων.
- **Διαχείριση καναλιών και περιόδων δειγματοληψίας:**  
Οι χρήστες μπορούν να δημιουργήσουν διάφορα κανάλια για την οργάνωση των δεδομένων τους. Επιπλέον, μπορούν να καθορίσουν την περίοδο δειγματοληψίας, που καθορίζει πόσα δείγματα πρέπει να σταλούν στο ThingSpeak κάθε λεπτό.
- **Χρήση MATLAB Analytics:**  
Ένα ενδιαφέρον χαρακτηριστικό του ThingSpeak είναι η δυνατότητα χρήσης λειτουργιών ανάλυσης MATLAB για την επεξεργασία των δεδομένων. Αυτό σημαίνει ότι μπορούμε να εφαρμόσουμε προχωρημένες αναλύσεις και αλγόριθμους στα δεδομένα που συλλέγουμε.
- **Διαμόρφωση προφίλ χρήστη:**  
Οι χρήστες μπορούν να δημιουργήσουν προσαρμοσμένα προφίλ που καθορίζουν ποια δεδομένα θα είναι δημόσια και ποια θα παραμένουν ιδιωτικά. Αυτό παρέχει ένα επίπεδο απόρρητου και ελέγχου στους χρήστες του ThingSpeak.

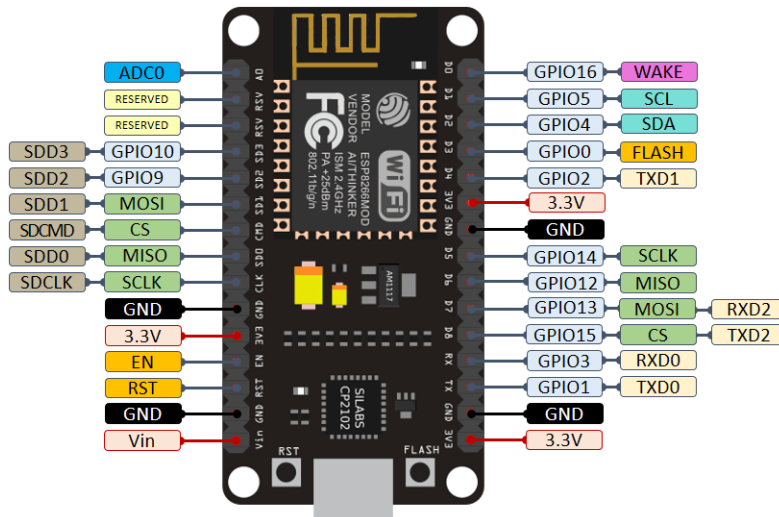
Αυτή η υπηρεσία είναι ιδιαίτερα χρήσιμη για εφαρμογές IoT, παρακολούθησης περιβάλλοντος, καταγραφής δεδομένων και άλλες εφαρμογές που απαιτούν τη συλλογή και την ανάλυση δεδομένων από συσκευές σε πραγματικό χρόνο.

## 2.4 Τα βασικά μέρη του κυκλώματος

### 2.4.1 NodeMCU V2 πλακέτα ανάπτυξης

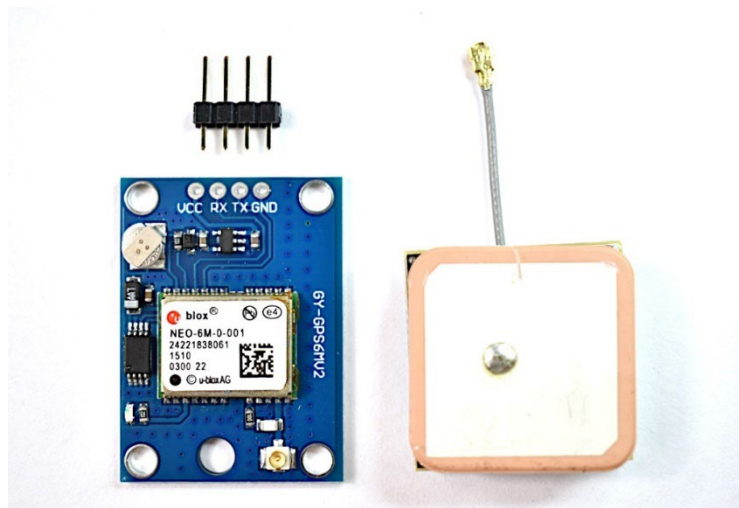
Πλακέτα ανάπτυξης που βασίζεται στο τσιπ ESP8266 12-e, έχει ενσωματωμένο προγραμματιστή για εύκολη και άμεση εγγραφή κώδικα μέσω υπολογιστή (σύνδεση μέσω της θύρας micro usb). Επίσης στην πλακέτα υπάρχει ρυθμιστής τάσης που μετατρέπει το εισερχόμενο ρεύμα σε σταθερή τιμή 3,3 Volt, κατάλληλο για το ESP8266 και τις

εισόδους/εξόδους. Όλες οι επαφές είναι διαθέσιμες ως ακροδέκτες, κατάλληλη για τοποθέτηση της πλακέτας σε μια βάση δοκιμών (breadboard) ή μιας διάτρητης πλακέτας για μόνιμη εγκατάσταση (perfboard). Τέλος, είναι συμβατό με το Arduino IDE καθώς και με πληθώρα από τις διαθέσιμες βιβλιοθήκες που υπάρχουν για εύκολη σύνδεση περιφερειακών και σύνδεση με τον κώδικα μας.



Εικόνα 6. NodeMCU V2 Development Board

## 2.4.2 GPS Neo-6M



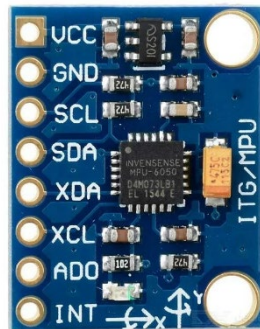
Εικόνα 7. Ublox NEO-6M GPS

Μικρό και ισχυρό GPS με εξωτερική κεραία. Μπορεί να παρακολουθεί μέχρι και 22 δορυφόρους, διαθέτει υψηλή ευαισθησία παρακολούθησης ενώ καταναλώνει μόλις 45mA. Ο ρυθμός ανανέωσης της τοποθεσίας είναι 5 το δευτερόλεπτο με 2,5 μέτρα οριζόντια ακρίβεια. Επίσης ένα σημαντικό χαρακτηριστικό του είναι η λειτουργία εξοικονόμησης ενέργειας, όπου μειώνει ακόμη περισσότερο την κατανάλωση απενεργοποιώντας και ενεργοποιώντας

επιλεκτικά συγκεκριμένα μέρη του δέκτη. Χρησιμοποιεί το στάνταρ nvmca όπου με λίγες γραμμές κώδικα αντλούμε πληροφορίες μέσω δορυφόρων (Ατομικό ρολόι, συντεταγμένες, ημισφαίρια, ταχύτητα). Η επικοινωνία γίνεται μέσω UART, με εργοστασιακό ρυθμό στα 9600. Δοκιμάστηκαν και άλλες ταχύτητες ωστόσο δεν λειτούργησαν σωστά. Στην συγκεκριμένη εργασία δεν είναι απαραίτητο να καταγράφουμε συνεχώς την τοποθεσία, παρά μόνο όταν εντοπιστεί ανωμαλία μέσω του επιταχυνσιόμετρου.

### 2.4.3 Επιταχυνσιόμετρο MPU-6050

Αποτελεί ένα ενσωματωμένο σύστημα αισθητήρων που συνδυάζει ένα τρισδιάστατο επιταχυνσιόμετρο και ένα τρισδιάστατο γυροσκόπιο. Χρησιμοποιείται ευρέως στη ρομποτική και ηλεκτρονική κοινότητα για αναγνώριση κίνησης, προσανατολισμού και κίνησης. Επίσης διαθέτει και αισθητήρα θερμοκρασίας αν και δεν χρησιμοποιείται κυρίως για αυτή τη λειτουργία του.



Εικόνα 8. Επιταχυνσιόμετρο MPU-6050

Βασικά χαρακτηριστικά του είναι:

- Επιταχυνσιόμετρο τριών κατευθύνσεων (x,y,z), για τη μέτρηση των αλλαγών στην ταχύτητα και τον προσανατολισμό. Στη συγκεκριμένη εργασία θα χρησιμοποιηθεί μόνο αυτό το χαρακτηριστικό του αισθητήρα.
- Γυροσκόπιο, το οποίο μετράει το ρυθμό περιστροφής γύρω από τους τρεις άξονες (x,y,z). Χρησιμοποιείται για τη μέτρηση αλλαγών της γωνιακής ταχύτητας.
- Ενσωματωμένος επεξεργαστής κίνησης, ο οποίος παράγει πληροφορίες για την κίνηση βάσει στα δεδομένα που καταγράφουν το επιταχυνσιόμετρο και το γυροσκόπιο.
- Το MPU-6050 επικοινωνεί με τον μικροελεκτή μέσω δίαυλου I2C και SPI

Σε αυτή την εργασία μας ενδιαφέρει οι τιμές της επιτάχυνσης, οι οποίες θα ελέγχονται συνεχώς για μεταβολές. Με την κατάλληλη φόρμουλα στον κώδικα θα εντοπίζει ανωμαλία στο οδόστρωμα και θα ξεκινάνε οι διαδικασίες καταγραφής.

#### 2.4.4 Αναγνώστης καρτών τύπου microSD

Μία κάρτα μνήμης τύπου microSD χρησιμοποιείται για την καταγραφή των δεδομένων κατά τη διάρκεια λειτουργίας της συσκευής. Ο αναγνώστης καρτών συνδέεται με το ESP8266 μέσω του δίαυλου SPI. Αν και το ESP8266 έχει ενσωματωμένη μνήμη flash μικρού μεγέθους καθιστώντας δυνατή την αποθήκευση εκεί, επέλεξα να μην το κάνω αυτό διότι οι συνεχείς εγγραφές θα φθείρουν γρηγορότερα τη μνήμη του χρίζοντας το άχρηστο.



Εικόνα 9. Αναγνώστης καρτών τύπου microSD

#### 2.4.5 Ηχείο τύπου Buzzer

Ένα μικρό ηχείο τύπου buzzer χρησιμοποιείται σε διάφορα σημεία του κώδικα ώστε να δίνει feedback στον προγραμματιστή σχετικά με τη λειτουργία της συσκευής. Χρησιμοποιήθηκε στη φάση του development, δεν εφαρμόστηκε στο τελικό σχέδιο ωστόσο παρέμεινε στην τελική πλακέτα σε περίπτωση που χρησιμοποιηθεί μελλοντικά.



Εικόνα 10. Ηχείο τύπου buzzer

## 2.4.6 Μπαταρία λιθίου 18650

Μια μπαταρία Li-On τροφοδοτεί όλο το κύκλωμα. Επειδή όμως η τάση στη οποία μπορεί να λειτουργήσει το ESP8266 κυμαίνεται στα 2.5 με 3.6 Volt θα χρησιμοποιήσουμε έναν ενισχυτή σταθερής τάσης (DC boost converter) το οποίο θα μας δίνει σταθερή τάση 5 Volt καθ' όλη τη διάρκεια ζωής της μπαταρίας. Έπειτα, το NodeMCU θα χρησιμοποιεί τον ρυθμιστή τάσης ώστε το ESP8266 και οι υπόλοιποι αισθητήρες να λαμβάνουν σταθερό ρεύμα 3.3 Volt.



Εικόνα 11. Μπαταρία λιθίου 18650

## 2.4.7 TP4056 μονάδα φόρτισης μπαταρίας

Πλακέτα μικρού μεγέθους που φορτίζει με ασφάλεια μπαταρίες λιθίου σε ικανοποιητική ταχύτητα. Διαθέτει σύγχρονη θύρα τροφοδοσίας τύπου USB C καθώς και δύο LED για εύκολη ανάγνωση της κατάστασης της μπαταρίας (πορτοκαλί σε κατάσταση φόρτισης, μπλε όταν είναι πλήρης).



Εικόνα 12. Μονάδα φόρτισης μπαταρίας TP4056

## 2.4.8 Μετατροπέας τάσης (ενισχυτής σταθερής τάσης)

Μετατροπέας που χρησιμοποιείται ευρέως σε φορητούς φορτιστές συσκευών (power bank), μετατρέπει την τάση που λαμβάνει από μια μπαταρία λιθίου (η οποία αυξομειώνεται

ανάλογα με το διαθέσιμο φορτίο που έχει διαθέσιμο) σε σταθερή τιμή 5 Volt. Το ιδανικό θα ήταν να είχαμε σταθερή τάση στα 3.3 Volt, αλλά δυστυχώς δεν βρέθηκε κατάλληλο εξάρτημα για αυτή τη λειτουργία. Δοκιμάστηκαν αντίστοιχα εξαρτήματα τα οποία θεωρητικά

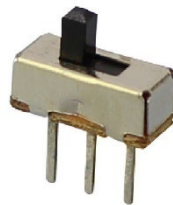


Εικόνα 13. Μετατροπέας σταθερής τάσης ρεύματος

διατηρούν την τάση εξόδου σταθερή άσχετα από την είσοδο (buck-boost), αλλά τελικά δεν λειτούργησαν σωστά και απορρίφθηκαν. Μελλοντικά αν βρεθεί το κατάλληλο εξάρτημα τότε το υπάρχον μπορεί εύκολα να αντικατασταθεί με μια μικρή αλλαγή στο κύκλωμα (Η τροφοδοσία να αλλάξει από τον ακροδέκτη Vin στον 3.3 του NodeMCU)

### 2.4.9 Διακόπτης τροφοδοσίας

Ένας μικρός διακόπτης τοποθετήθηκε ανάμεσα στον μετατροπέα τάσης και στο Vin του NodeMCU για να ενεργοποιούμε το κύκλωμα. Είναι απαραίτητο καθώς δεν θέλουμε να τροφοδοτούμε το κύκλωμα συνεχώς αφού η μπαταρία είναι μόνιμο κομμάτι του κυκλώματος.



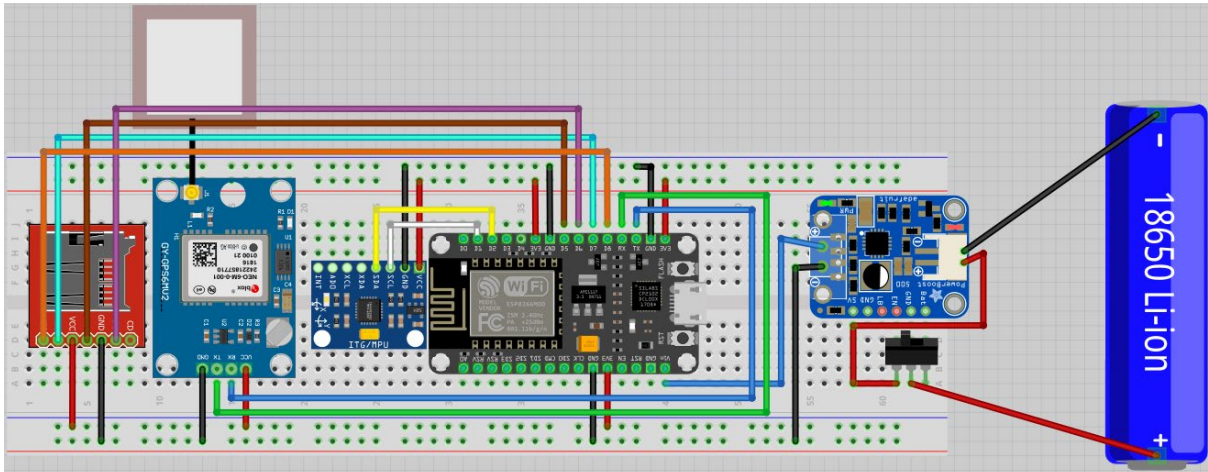
Εικόνα 14. Διακόπτης τροφοδοσίας

## 3. Από τη θεωρία στην πράξη

### 3.1 Σχεδιάγραμμα

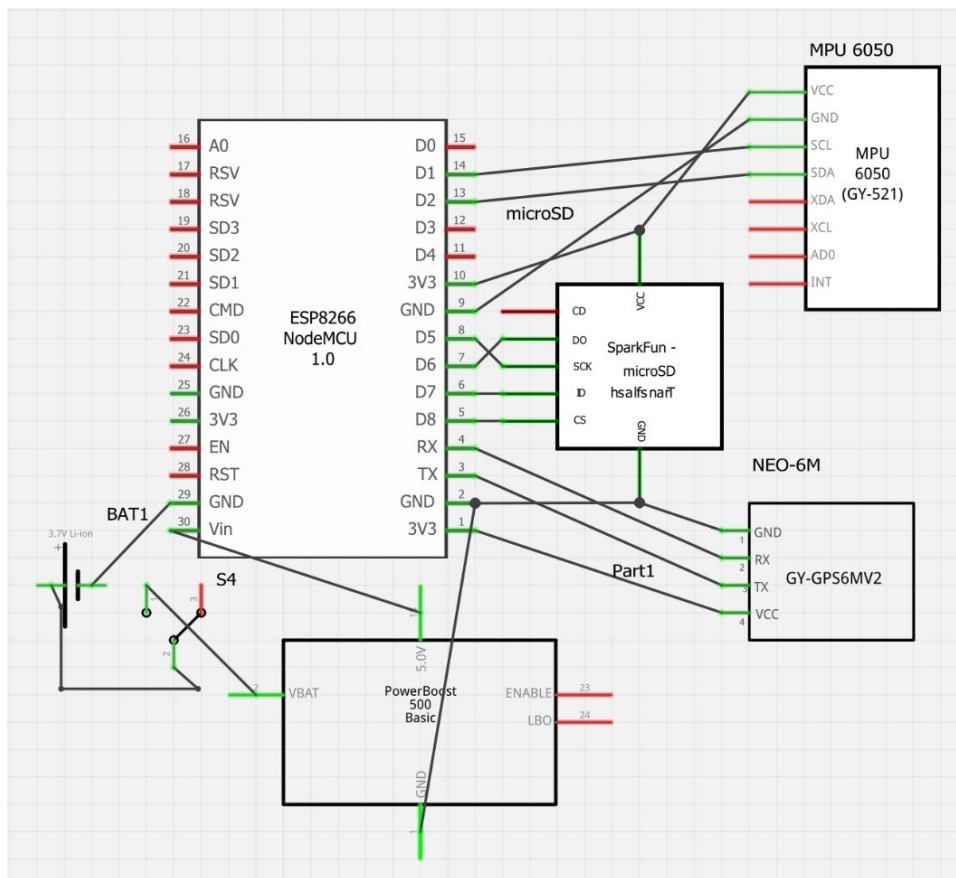
Πριν ξεκινήσουμε την κατασκευή του κυκλώματος, πρέπει να το σχεδιάσουμε ώστε να ελέγξουμε ότι όλες οι συνδέσεις που θα κάνουμε είναι σωστές και δεν θα καταστρέψουμε το hardware λόγω λανθασμένης συνδεσμολογίας. Χρησιμοποιώντας το λογισμικό Fritzing επιτυγχάνουμε ακριβώς αυτό καθώς και ένα τελικό σχέδιο με τις συνδέσεις το οποίο σε κάποια βλάβη εξαρτήματος θα είναι χρήσιμο για τη σωστή διάγνωση.





Εικόνα 15. Οπτικό σχεδιάγραμμα μέσω του λογισμικού Fritzing

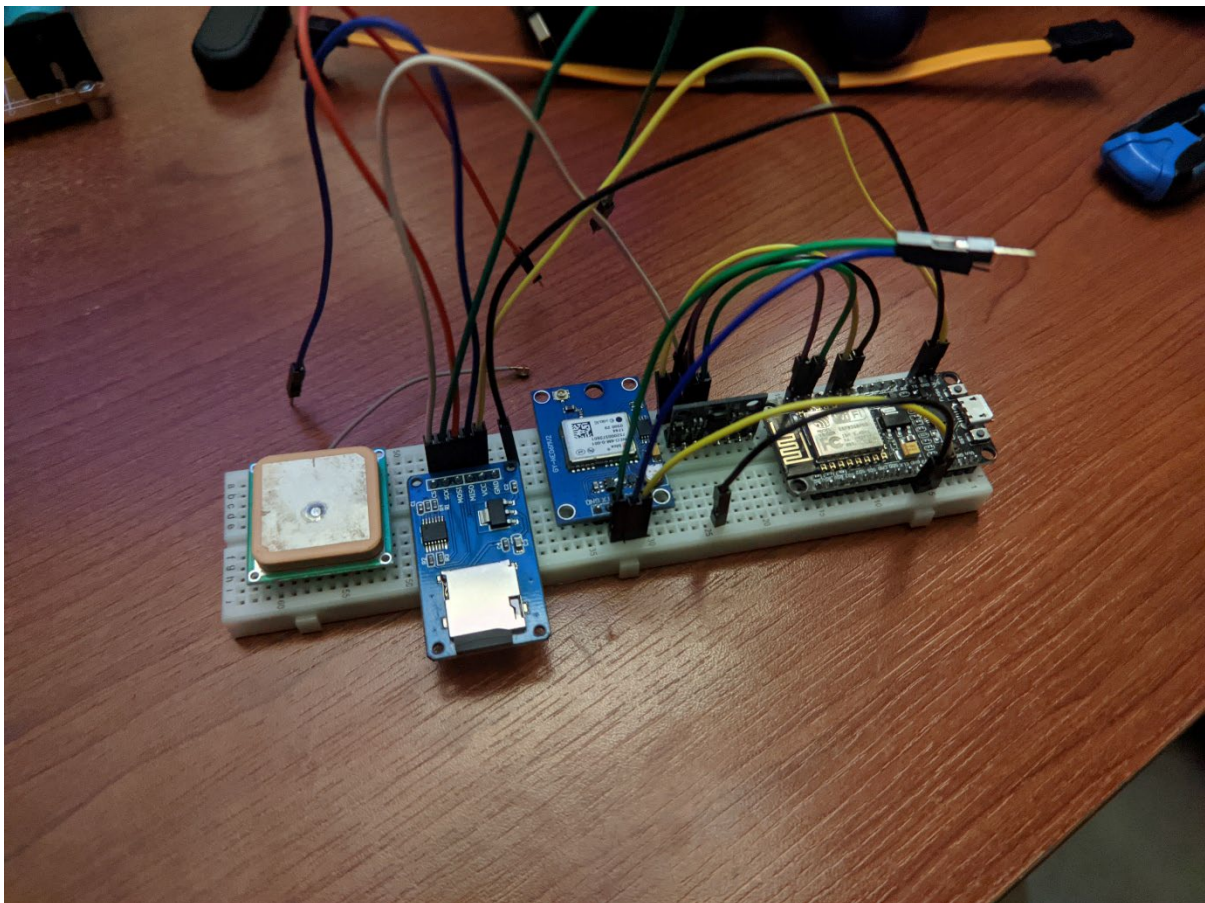
Παρατηρώντας τις συνδέσεις στο σχεδιάγραμμα φαίνεται ότι σχεδόν όλες συνδέσεις των εξαρτημάτων θα είναι από τη μια μεριά του NodeMCU. Αυτό δεν έγινε σκόπιμα, ωστόσο μας διευκολύνει στην οργάνωση των κολλήσεων που θα γίνουν στη συνέχεια (μιας και η τελική πλακέτα θα είναι διάτρητη και όχι ειδική παραγγελία για την εργασία).



Εικόνα 16. Σχηματικός πίνακας κυκλώματος

### 3.2 Εγκατάσταση σε πλακέτα δοκιμών

Αφού έχουμε σχεδιάσει το τελικό κύκλωμα και είμαστε σίγουροι για την ορθή λειτουργία τότε μπορούμε να προχωρήσουμε σε πρόχειρη σύνδεση των εξαρτημάτων σε ένα πλακέτα δοκιμών (breadboard), σύμφωνα με το σχεδιάγραμμα που δημιουργήσαμε. Το NodeMCU module είναι συνδεδεμένο στον υπολογιστή άρα λαμβάνει ρεύμα μέσω της θύρας USB, καθώς και τα περιφερειακά εξαρτήματα που συνδέονται σε αυτό. Με αυτόν τον τρόπο μπορώ να παρακολουθώ μέσω serial monitor τα δεδομένα που επεξεργάζεται συνεχώς, όπως και να αποσυνδέω τους αισθητήρες από το ρεύμα όταν θέλω να κάνω αλλαγές στον κώδικα ή την συνδεσμολογία.



Εικόνα 17. Εγκατάσταση των εξαρτημάτων σε πλακέτα δοκιμών

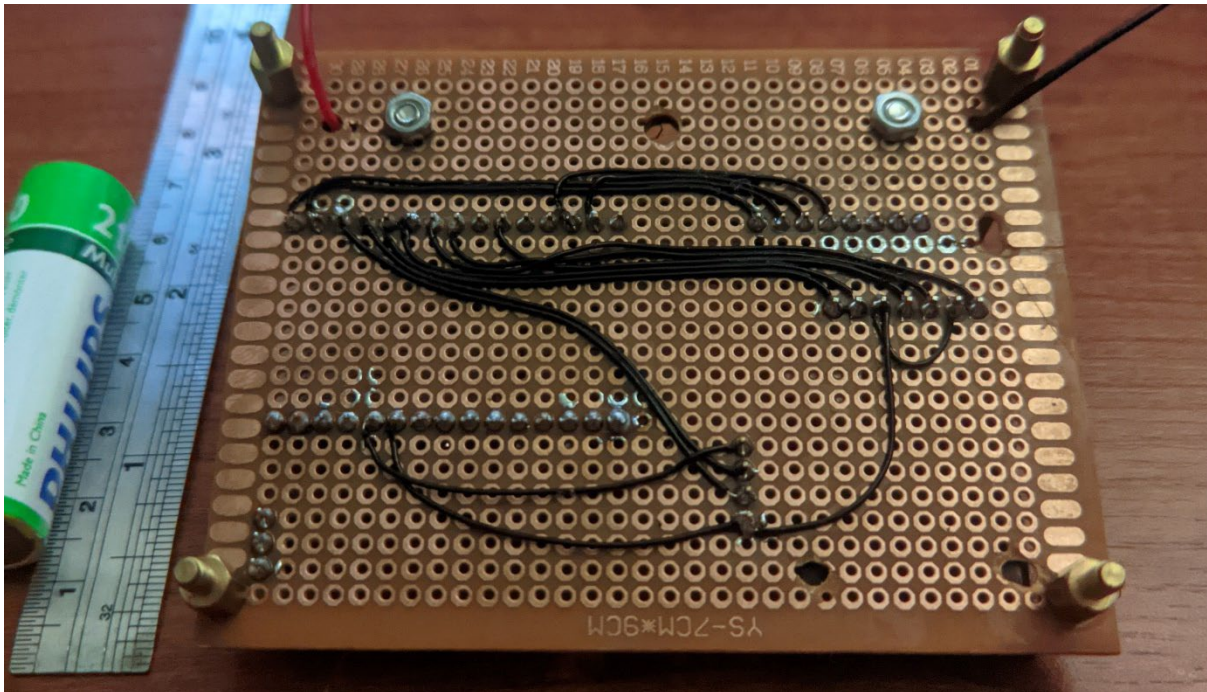
Σε αυτό το στάδιο έγιναν όλες οι δοκιμές των εξαρτημάτων, δηλαδή εάν λειτουργούσαν σωστά μόνα τους και στη συνέχεια ως σύνολο στο κύκλωμα. Αντίστοιχα ο κώδικας γράφτηκε ξεχωριστά για τον κάθε αισθητήρα ως δοκιμή για την ορθή τους λειτουργία και στην συνέχεια ενώθηκαν στο τελικό αρχείο κάνοντας τις απαραίτητες διορθώσεις. Όταν ο κώδικας έφτασε



στο σημείο να ανιχνεύει όλα τα εξαρτήματα επιτυχώς και να εκτελεί τις λειτουργίες που επιθυμούσα, μπορούσα να προχωρήσω στο επόμενο βήμα.

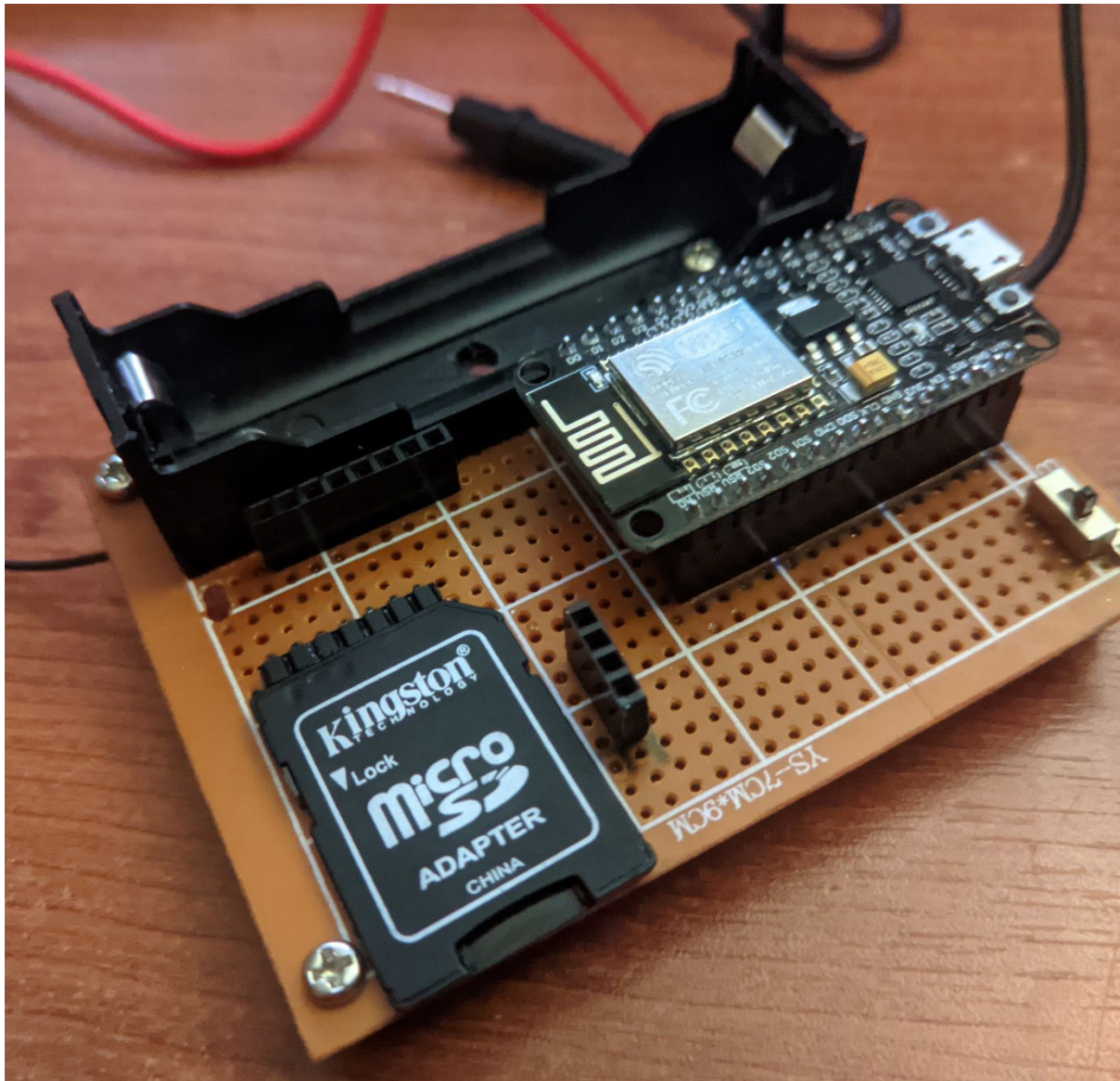
### 3.3 Πρωτότυπη πλακέτα για εξωτερικές δοκιμές

Εφόσον ήμουν σίγουρος για τη συνδεσμολογία, έπρεπε στη συνέχεια να κάνω δοκιμή του κυκλώματος στην πράξη, δηλαδή να κάνω δοκιμαστικές διαδρομές με όχημα και αποσφαλμάτωση στον κώδικα μου. Έτσι λοιπόν ξεκίνησα την κατασκευή μιας διάτρητης πλακέτας με μόνιμες ηλεκτρικές συνδέσεις (perfboard). Χρησιμοποιήθηκε καλώδιο μικρής διατομής ώστε να δημιουργήσω συνδέσεις (traces) μεταξύ του NodeMCU και των εξαρτημάτων, καθώς και θηλυκοί ακροδέκτες πλακέτας για την προσωρινή εγκατάσταση των εξαρτημάτων.



Εικόνα 18. Δημιουργία συνδέσεων με καλώδιο μικρής διατομής

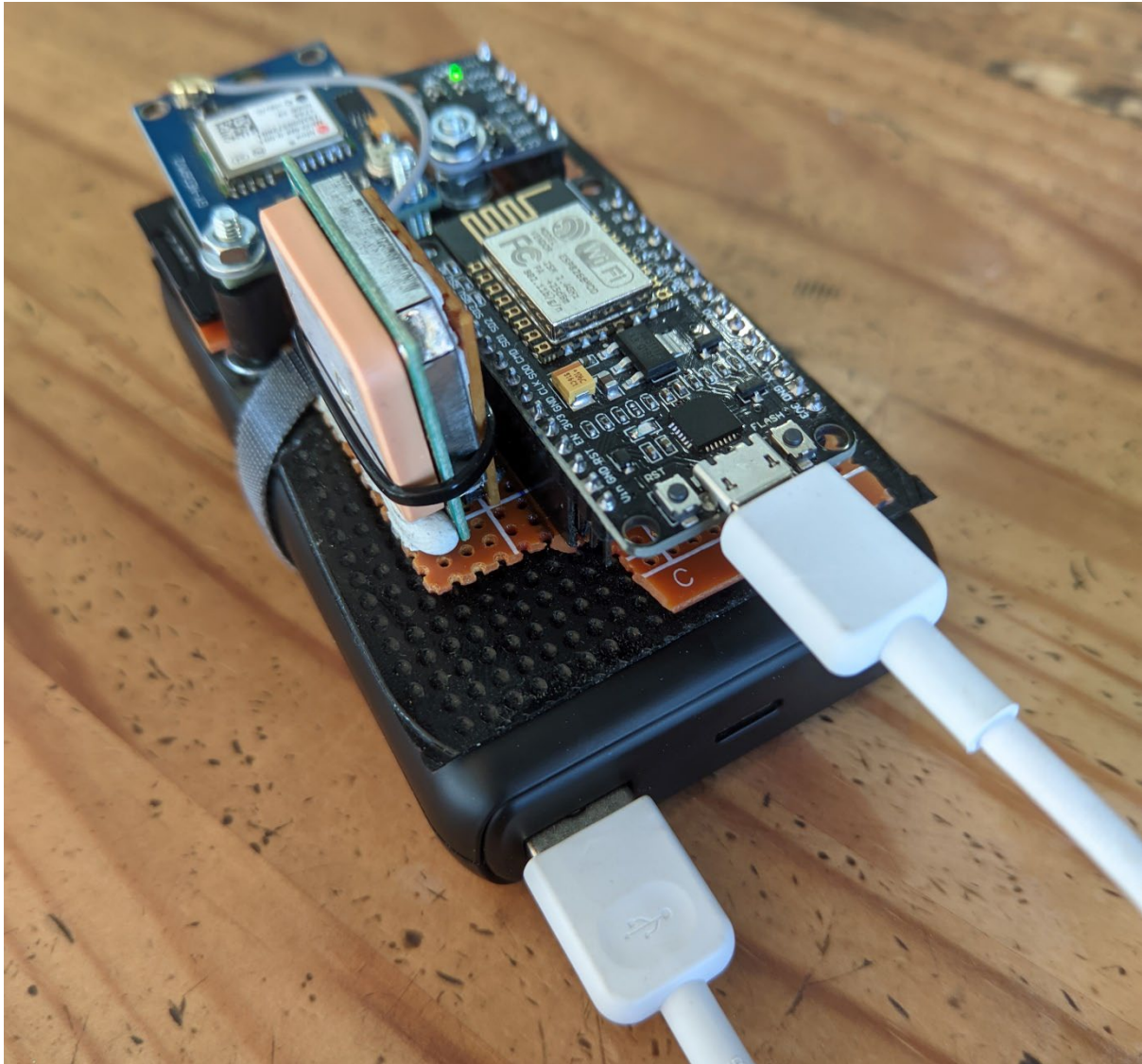
Επίσης τοποθετήθηκε και η θήκη της μπαταρίας πάνω στην πλακέτα αλλά δεν συνδέθηκε. Ήθελα να παρατηρήσω τον χώρο που θα καταλαμβάνει και πως θα είναι ο πιο αποτελεσματικός τρόπος τοποθέτησης των εξαρτημάτων βάσει του διαθέσιμου χώρου.



Εικόνα 19. Τα εξαρτήματα κουμπώνουν στους θηλυκούς ακροδέκτες για δοκιμές

Με την ολοκλήρωση ξεκίνησαν οι πρώτες δοκιμές σωστής λειτουργίας. Η πλακέτα τροφοδοτείται από ένα power bank και καταλαμβάνει αρκετό χώρο. Τα αποτελέσματα είναι ικανοποιητικά, ωστόσο πρέπει να δημιουργηθεί μια βάση ώστε να εφαρμόζεται στο όχημα και οι δονήσεις τους να συντονίζονται και να έχουμε ακριβείς μετρήσεις.

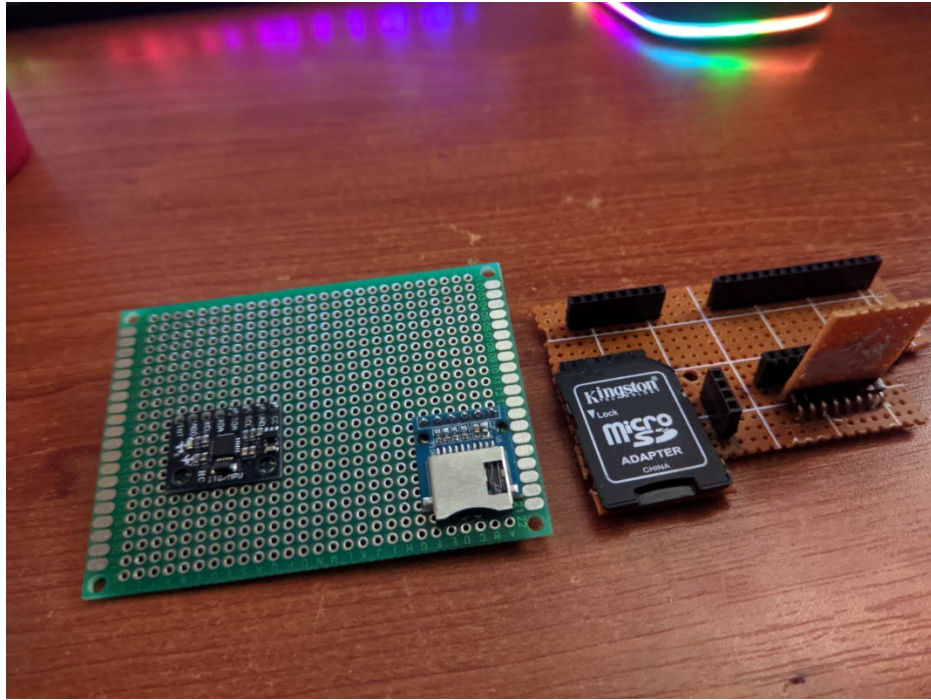




*Εικόνα 20. Η προσωρινή πλακέτα είναι στο στάδιο των δοκιμών και αποσφαλμάτωσης*

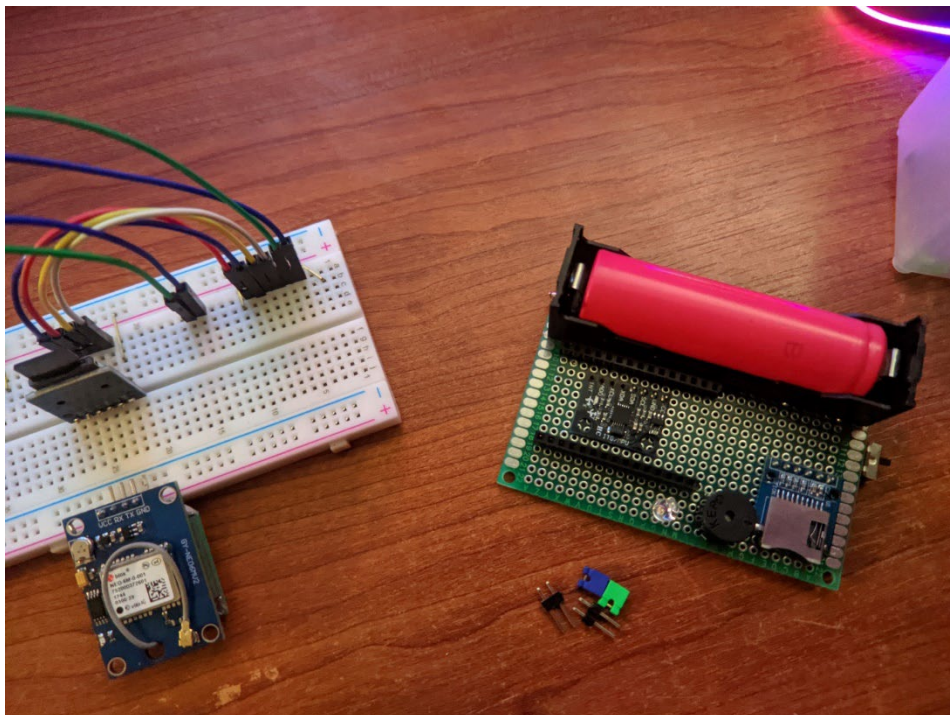
### **3.4 Τελική πλακέτα κυκλώματος**

Οι δοκιμές ολοκληρώθηκαν, ο κώδικας δουλεύει ικανοποιητικά, οπότε μπορώ να προχωρήσω στη δημιουργία της τελικής πλακέτας. Χρησιμοποίησα καλύτερης ποιότητας διάτρητη πλακέτα και τροποποίησα τη θέση των εξαρτημάτων ώστε να καταλαμβάνουν λιγότερο χώρο.



Εικόνα 21. Η τελική διάτρητη πλακέτα, τοποθετήθηκαν τα πρώτα εξαρτήματα

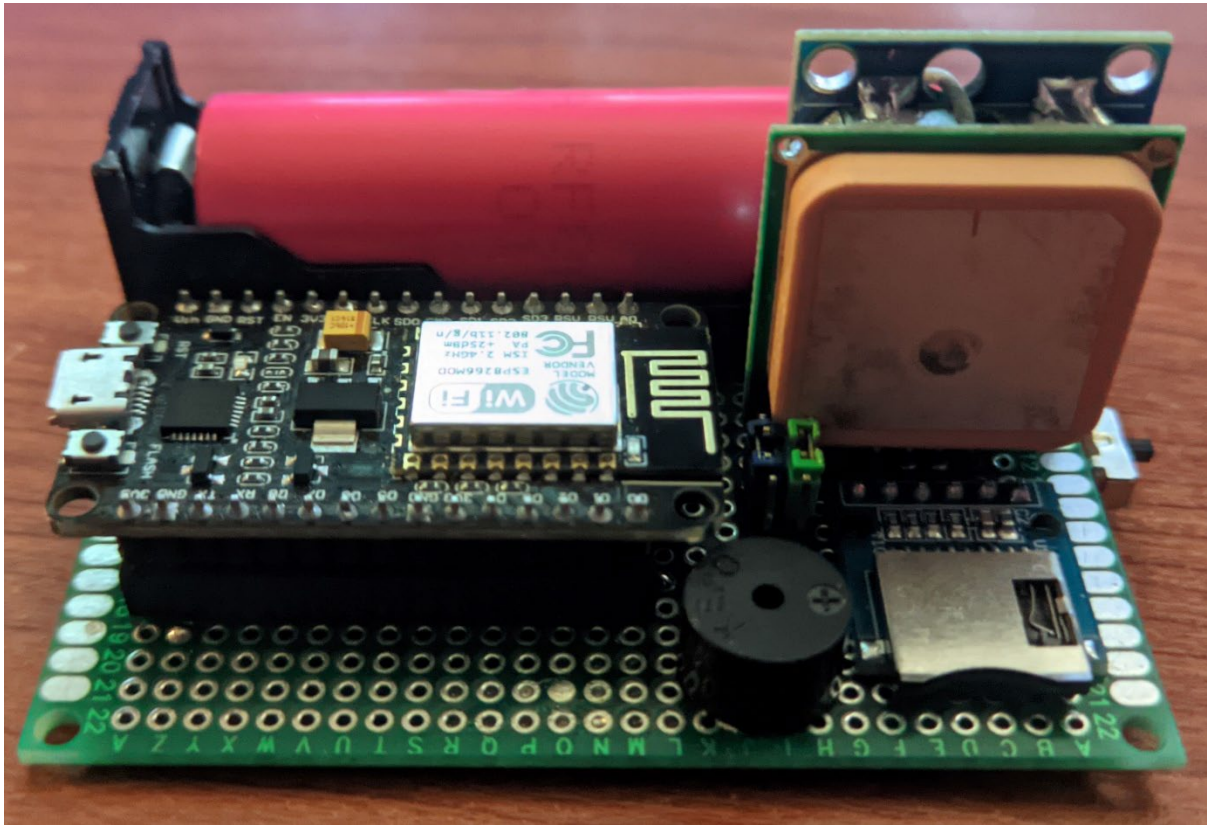
Το MPU-6050 τοποθετήθηκε κάτω από το NodeMCU, το οποίο συνδέεται στην πλακέτα στους θηλυκούς ακροδέκτες που εγκατέστησα. Επέλεξα αυτό τον τρόπο ώστε να μπορώ να αφαιρέσω το NodeMCU σε περίπτωση που θέλω να αλλάξω τον κώδικα.



Εικόνα 22. Τοποθετήθηκαν οι θηλυκοί ακροδέκτες και η μπαταρία στο κύκλωμα

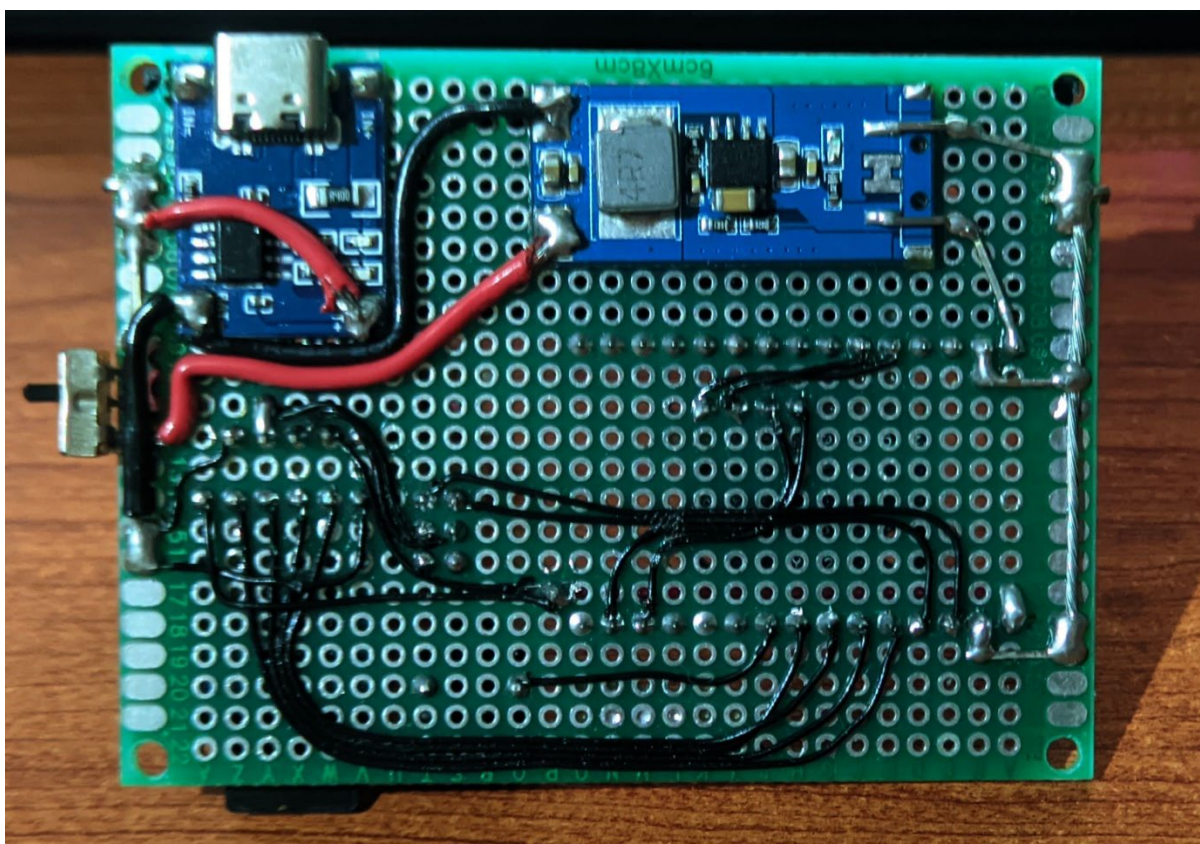


Για τη διαδικασία της αποσφαλμάτωσης τοποθέτησα ένα ηχείο τύπου buzzer και ένα LED για εύκολη εύρεση προβλημάτων κατά τη διάρκεια εκτέλεσης του κώδικα. Στη συνέχεια αποσυνδέθηκαν από το κύκλωμα ωστόσο άφησα το buzzer στην πλακέτα για μελλοντική χρήση.



Εικόνα 23. Η πλακέτα ολοκληρωμένη και έτοιμη για χρήση

Στην πίσω όψη βρίσκουμε όλες τις απαραίτητες συνδέσεις για τη σωστή λειτουργία του κυκλώματος. Το TP4056 αναλαμβάνει την φόρτιση μπαταρίας, ο ενισχυτής τάσης μετατρέπει το ρεύμα της μπαταρίας σε σταθερή τιμή 5 Volt και στη συνέχεια τροφοδοτεί το κύκλωμα μέσω της θύρας Vin του NodeMCU. Ανάμεσά τους έχει τοποθετηθεί ένας διακόπτης on-off για έλεγχο της τροφοδοσίας.



Εικόνα 24. Πίσω όψη της ολοκληρωμένης πλακέτας

## 4. Ανάλυση λειτουργίας κυκλώματος

### 4.1 Σύντομη περιγραφή λειτουργίας

Ξεκινάμε με την παροχή ρεύματος στην συσκευή, χρησιμοποιώντας ένα διακόπτη on-off. Κατά την εκκίνηση γίνεται αρχικοποίηση των αισθητήρων και της κάρτας μνήμης. Στη συνέχεια γίνεται έλεγχος εάν είμαστε στη βάση και μπορούμε να συνδεθούμε στο διαδίκτυο μέσω Wi-Fi. Αν αυτό είναι εφικτό τότε το ESP8266 συνδέεται στην πλατφόρμα ThingSpeak και ανεβάζει συνεχώς τα δεδομένα που είναι αποθηκευμένα στην κάρτα μνήμης μέχρι το πέρας τους. Εφόσον ολοκληρωθεί η διαδικασία ή δεν υπάρχουν δεδομένα για καταγραφή τότε γίνεται διαγραφή του αρχείου από τη microSD, έτσι ώστε να μην υπάρχουν διπλές εισαγωγές στη βάση δεδομένων. Το επόμενο κομμάτι του κώδικα τρέχει συνεχώς μέχρι να απενεργοποιηθεί

η συσκευή. Κατά τη φάση αυτή το MPU-6050 καταγράφει συνεχώς την επιτάχυνση στους τρεις άξονες και υπολογίζει το συνολικό μέγεθος επιτάχυνσης. Εάν αυτή η τιμή δεν είναι ανάμεσα στα όρια που έχουμε θέσει τότε αυτό σημαίνει ότι βρέθηκε ανωμαλία στο οδόστρωμα οπότε ξεκινάει η διαδικασία καταγραφής. Εάν το σήμα το GPS έχει κλειδώσει ξεκινάει να αποθηκεύει σε μεταβλητές τα εξής: ημερομηνία, ώρα, γεωγραφικό μήκος και πλάτος, καθώς και την ταχύτητα. Ακολουθεί μια μετατροπή των μεταβλητών σε σειρές χαρακτήρων, άνοιγμα ή δημιουργία του αρχείου της κάρτας μνήμης και εισαγωγή των δεδομένων σε μια σειρά η οποία αποτελεί στιγμιότυπο. Αφού ολοκληρωθεί η εγγραφή κλείνει το αρχείο και συνεχίζει ο κώδικας να τρέχει σε επανάληψη.

## 4.2 Ανάλυση κώδικα

Σε αυτό το κομμάτι θα γίνει ανάλυση του κώδικα σε 5 μέρη, βιβλιοθήκες, global μεταβλητές, void setup() , void loop() και συναρτήσεις. Στον κώδικα υπάρχουν ακόμη σειρές που χρησιμοποιήθηκαν κατά την διαδικασία της αποσφαλμάτωσης, δεν αφαιρέθηκαν σε περίπτωση που θέλουμε να κάνουμε κάποιον έλεγχο ή να προσθέσουμε κάποιο υλικό (π.χ. οθόνη για προβολή των μηνυμάτων ή ηχείο buzzer).

### 4.2.1 Βιβλιοθήκες

Οι βιβλιοθήκες που χρησιμοποιήθηκαν είναι οι εξής:

- **TinyGPSPlus**  
Αποτελεί μια πλήρως προσαρμόσιμη βιβλιοθήκη ανάλυσης της ροής δεδομένων NMEA που παρέχουν οι συσκευές GPS. Όπως τον προκάτοχο του, τη βιβλιοθήκη TinyGPS, παρέχει εύκολες μεθόδους εξαγωγής θέσης, ημερομηνίας, ώρας, υψόμετρου, συντεταγμένων και ταχύτητας από συσκευές GPS του εμπορίου. Επιπλέον είναι πιο εύκολη στη χρήση της και είναι ικανή να εξάγει αυθαίρετα δεδομένα από τις οποιαδήποτε πρόταση NMEA, ακόμη και από τις ιδιότητες.
- **Adafruit\_MPU6050**  
Βιβλιοθήκη της Adafruit, επικοινωνεί με το MPU-6050 και μας επιτρέπει να εξάγουμε δεδομένα από τον αισθητήρα.
- **Adafruit\_Sensor**  
Άλλη μια χρήσιμη βιβλιοθήκη της, απαραίτητη όταν χρησιμοποιούμε βιβλιοθήκες της Adafruit για επικοινωνία με αισθητήρες. Επιτρέπει την εύκολη ενσωμάτωση και χρήση ποικίλων αισθητήρων μέσω ενός κοινού API, κάνοντας την ανάπτυξη προγραμμάτων πιο απλή και ευέλικτη. Βασικός της στόχος είναι η παροχή μιας ενιαίας διεπαφής για αισθητήρες διαφορετικού τύπου, έτσι ώστε οι προγραμματιστές να μπορούν να χρησιμοποιούν διάφορους αισθητήρες χωρίς να χρειάζεται να μάθουν καινούριες βιβλιοθήκες ξεχωριστά.
- **ThingSpeak**

Χρησιμοποιείται για την αποστολή δεδομένων στην πλατφόρμα ThingSpeak μέσω API.

- **SPI**  
Αυτή η βιβλιοθήκη μας επιτρέπει να επικοινωνήσουμε με συσκευές που υποστηρίζουν τον δίαυλο επικοινωνίας SPI, με τον μικροελεκτή μας. Είναι βασικό κομμάτι της βιβλιοθήκης Arduino και έρχεται εγκατεστημένη με τον λογισμικό Arduino IDE.
- **SD**  
Ενεργοποιεί την ανάγνωση και εγγραφή σε κάρτες μνήμης SD. Δουλεύει σε συνδυασμό με την βιβλιοθήκη SPI, διότι οι κάρτες μνήμης επικοινωνούν με τον μικροελεκτή με αυτό το δίαυλο επικοινωνίας.
- **Wire**  
Αυτή η βιβλιοθήκη μας επιτρέπει να επικοινωνήσουμε με συσκευές που υποστηρίζουν τον δίαυλο επικοινωνίας I2C/TWI.
- **ESP8266WiFi**  
Μας επιτρέπει να αξιοποιήσουμε πλήρως τις δυνατότητες ασύρματου δικτύου που μας προσφέρει το ESP8266.
- **Secrets**  
Αρχείο header που δημιουργήσαμε, χρησιμοποιείται για την αποθήκευση ευαίσθητων πληροφοριών, όπως το όνομα και ο κωδικός του Wi-Fi και τα διαπιστευτήρια σύνδεσης με το TeamSpeak. Αυτές οι πληροφορίες δεν θέλουμε να φαίνονται στον κώδικα και μπορούν εύκολα να τροποποιηθούν μέσω του αρχείου secrets.h .

#### 4.2.2 Global μεταβλητές

Κατά κύριο λόγο χρησιμοποιήθηκαν global μεταβλητές διότι επιτρέπουν την μετάβαση των τιμών τους σε όλες τις συναρτήσεις που δημιουργήθηκαν. Αρχικά έπρεπε να δημιουργήσω μεταβλητές για τα διαπιστευτήρια του Wi-Fi και του ThingSpeak , καθώς και αντικείμενα για να καλώ λειτουργίες από τους αισθητήρες και τις βιβλιοθήκες. Επιπλέον δημιούργησα τις σταθερές scl και sdl , που χρησιμοποιώ για σύνδεση του MPU-6050, και έδωσα ως τιμές τους ακροδέκτες του NodeMCU που είχε γίνει η σύνδεση. Μετέπειτα έγραψα το ελάχιστο και το μέγιστο όριο τιμής για τον συνολικό μέγεθος επιτάχυνσης που υπολογίζει το MPU-6050. Τέλος , συμπλήρωσα τις μεταβλητές όπου τον Neo-6M GPS αποθηκεύει προσωρινά τις τιμές, τις επεξεργάζεται και στη συνέχεια τις αποθηκεύει στην κάρτα μνήμης.



```

1  #include <TinyGPSPlus.h>
2  #include <SPI.h>
3  #include <SD.h>
4  #include <ESP8266WiFi.h>
5  #include <Wire.h>
6  #include <Adafruit_MPU6050.h>
7  #include <Adafruit_Sensor.h>
8  #include <ThingSpeak.h>
9  #include "secrets.h"
10
11 //WiFi credentials
12 char ssid[] = SECRET_SSID;
13 char password[] = SECRET_PASS;
14 unsigned long channelID = SECRET_CH_ID;
15 const char* apiKey = SECRET_WRITE_APIKEY;
16
17 TinyGPSPlus gps;
18 Adafruit_MPU6050 mpu;
19 WiFiClient client;
20
21 const uint8_t scl = D1;
22 const uint8_t sda = D2;
23 const int chipSelect = D8;
24 sensors_event_t a, g, temp;
25 const float minThreshold = 8;
26 const float maxThreshold = 12;
27 //const int buzzerPin = 14;
28 int speed, mergedDate, mergedTime;
29 float latitude, longitude, totalAccel;
30 String mergedDate_str, mergedTime_str, lat_str, lng_str, speed_str;

```

Εικόνα 25. Οι βιβλιοθήκες, global μεταβλητές και οι κλάσεις που δημιουργήθηκαν

### 4.2.3 void setup()

Το πρώτο κομμάτι του κώδικα είναι να αρχικοποιήσουμε τους αισθητήρες και να εκτελέσουμε λειτουργίες που επιθυμούμε να γίνουν στην αρχή ή μια φορά. Ο κώδικας που εκτελείται είναι ο εξής:

```

32 void setup() {
33   Serial.begin(9600); //Neo 6M default baudrate is 9600, other numbers make the module unresponsive
34   Wire.begin(sda, scl);
35   initializeMPU6050();
36   initializeSDCard();
37   connectToWiFi();
38   if (WiFi.status() == WL_CONNECTED){
39     //Serial.println("Wifi check pass");
40     ThingSpeak.begin(client);
41     uploadDataToThingSpeak();
42   }
43 }

```

Εικόνα 26. Η συνάρτηση αρχικοποίησης void setup

- `Serial.begin(9600)`: Θέτει τον ρυθμό δεδομένων το δευτερόλεπτο για τη σειριακή μετάδοση δεδομένων. Το GPS αρχικοποιείται και επικοινωνεί με το ESP8266 στο ρυθμό που θέσαμε, δοκιμάστηκαν πιο υψηλές ταχύτητες αλλά εμφάνιζε σκουπίδια (garbage)
- `Wire.begin(sda, scl)`: Ενεργοποιεί την επικοινωνία μέσω I2C.
- `initializeMPU6050()`: Συνάρτηση, όπου αρχικοποιείται το MPU-6050, καθώς και γίνεται συνεχώς έλεγχος εάν είναι συνδεδεμένο στο σύστημα (`!mpu.begin()` μέσα σε επανάληψη `while`). Εάν δεν ανιχνεύεται ο κώδικας μένει στάσιμος και δεν συνεχίζει η εκτέλεση.
- `initializeSDCard()`: Αρχικοποίηση και έλεγχος για την ύπαρξη κάρτας μνήμης στο σύστημα, χωρίς την ανίχνευση της ο κώδικας σταματάει. Μόλις εντοπιστεί επιτυχώς κάρτα μνήμης, τότε ο κώδικας προχωράει παρακάτω.
- `connectToWiFi()`: Προσπαθεί να συνδεθεί στο ασύρματο δίκτυο που έχουμε ορίσει. Αν δεν τα καταφέρει μέσα στον προκαθορισμένο χρόνο τότε τελειώνει η εκτέλεση και προχωράει στο `void loop`. Ειδικά αν συνδεθεί με επιτυχία τότε καλείται η τελευταία συνάρτηση του `void setup`.
- `uploadDataToThingSpeak()`: Σε αυτή τη συνάρτηση ανεβάζουμε τα δεδομένα που έχουμε συλλέξει στην πλατφόρμα ThingSpeak. Αρχικά ελέγχει εάν υπάρχει το αρχείο στην κάρτα μνήμης και το ανοίγει, διαβάζοντας μία γραμμή κάθε φορά. Ο τύπος του αρχείου είναι `.csv`, δηλαδή `comma separated value`, οπότε η συνάρτηση χωρίζει τη σειρά χαρακτήρων σε μικρότερες, όπου υπάρχει κόμμα. Έτσι έχουμε τα πεδία που έχουμε ορίσει στην πλατφόρμα τοποθετημένα στη σειρά που θέλουμε να ανέβουν. Στη συνέχεια γίνεται η εγγραφή των πεδίων στο κανάλι μας, με χρήση των διαπιστευτηρίων που ορίσαμε, με μια μικρή καθυστέρηση κάθε φορά για να μην ξεπεράσουμε τα όρια που έχει θέσει το ThingSpeak στο ανέβασμα πληροφοριών. Αυτή η διαδικασία συνεχίζεται μέχρι να διαβαστεί και να αποσταλεί και η τελευταία σειρά του αρχείου. Αφού ολοκληρωθεί επιτυχώς η διαδικασία το αρχείο «`Road_data.csv`» διαγράφεται από τη μνήμη και ο κώδικας το `void setup` φτάνει στο τέλος της εκτέλεσης.

```

void initializeMPU6050() {
  while (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
    delay(10);
  }
  Serial.println("MPU6050 Found!");
  mpu.setAccelerometerRange(MPU6050_RANGE_2_G);
}
void initializeSDCard() {
  if (!SD.begin(chipSelect)) {
    Serial.println("Card not found.");
    while (1); // Stop execution if SD card is not found
  }
  Serial.println("Card initialized.");
}
void connectToWiFi() {
  int wificount = 0;
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED && wificount < 10) {
    delay(500);
    wificount++;
  }
  Serial.println("");
  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("WiFi Connected.");
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());
  }
  else
    Serial.println("WiFi connection failed.");
}
}

```

Εικόνα 27. Οι συναρτήσεις που καλούνται κατά την εκτέλεση του void setup

```

void uploadDataToThingSpeak() {
  if (SD.exists("Road_data.csv")) {
    File dataFile = SD.open("Road_data.csv");
    String line;

    while (dataFile.available()) {
      line = dataFile.readStringUntil('\n');

      int comma1 = line.indexOf(',');
      int comma2 = line.indexOf(',', comma1 + 1);
      int comma3 = line.indexOf(',', comma2 + 1);
      int comma4 = line.indexOf(',', comma3 + 1);
      int comma5 = line.indexOf(',', comma4 + 1);

      mergedDate_str = line.substring(0, comma1);
      mergedTime_str = line.substring(comma1 + 1, comma2);
      lat_str = line.substring(comma2 + 1, comma3);
      lng_str = line.substring(comma3 + 1, comma4);
      speed_str = line.substring(comma4 + 1, comma5);
      totalAccel = line.substring(comma5 + 1).toFloat();

      ThingSpeak.setField(1, mergedDate);
      ThingSpeak.setField(2, mergedTime);
      ThingSpeak.setField(3, lat_str);
      ThingSpeak.setField(4, lng_str);
      ThingSpeak.setField(5, speed_str);
      ThingSpeak.setField(6, totalAccel);

      int status = ThingSpeak.writeFields(channelID, apiKey);
      delay(1000); // Delay between each upload to avoid exceeding ThingSpeak rate limits

      if (status == 200) {
        Serial.println("Data uploaded to ThingSpeak successfully");
      }
      else {
        Serial.print("Data upload to ThingSpeak failed. Status code: ");
        Serial.println(status);
      }
    }
    dataFile.close();
    Serial.println("Data uploaded to ThingSpeak");
    SD.remove("Road_data.csv");
    Serial.println("Data file deleted");
  }
  else
  Serial.println("Error opening file...");
}

```

Εικόνα 28. Η συνάρτηση uploadDataToThingSpeak

## 4.2.4 void loop()

```
void loop() {
  readAndProcessAccelerometerData();
  if (isAnomalyDetected()) {
    captureGPSData();
    if (gps.location.isValid())
      saveDataToSDCard();
  }
  delay(100);
}
```

Εικόνα 29. Η συνάρτηση επανάληψης void loop

Το δεύτερο κομμάτι του κώδικα τρέχει συνεχώς σε επανάληψη μέχρι να απενεργοποιήσουμε τη συσκευή. Αποτελείται από τις εξής συναρτήσεις:

- `readAndProcessAccelerometerData()`: Συνάρτηση η οποία επικοινωνεί με το MPU-6050, καταγράφει την επιτάχυνση στους 3 άξονες και υπολογίζει το συνολικό μέγεθος επιτάχυνσης.
- `isAnomalyDetected()`: Στη συνέχεια γίνεται έλεγχος για τον εντοπισμό ανωμαλίας, δηλαδή αν η τιμή που υπολογίστηκε προηγουμένως είναι εντός των ορίων που έχουμε θέσει. Σε περίπτωση που είναι εντάξει ο δρόμος, τότε ο κώδικας δεν σταματάει και συνεχίζει την καταγραφή της επιτάχυνσης εκ νέου.
- `captureGPSData()`: Εάν ο έλεγχος που πραγματοποιήθηκε είναι σωστός, αυτό σημαίνει ότι εντοπίστηκε τιμή εκτός ορίων άρα συναντήθηκε κάποια βλάβη στο οδόστρωμα. Αυτό έχει σαν αποτέλεσμα να κληθεί η συνάρτηση που αρχικά ελέγχει ότι το Neo-6M GPS είναι διαθέσιμο, το σήμα έχει κλειδώσει και έπειτα καταγράφει τα δεδομένα που επιθυμούμε σε μεταβλητές. Τέλος, εκτελεί μια μετατροπή των μεταβλητών σε τύπο σειράς χαρακτήρων (string) για να μπορούμε να τα κάνουμε εγγραφή στο αρχείο.
- `saveDataToSDCard()`: Τελευταία συνάρτηση, η οποία ανοίγει το αρχείο «Road\_data.csv» ή το δημιουργεί εάν είναι η πρώτη εγγραφή. Καταγράφει τις μεταβλητές που δημιουργήσαμε με τα δεδομένα του Neo-6M GPS καθώς και το συνολικό μέγεθος επιτάχυνσης. Όλα τα δεδομένα είναι σε μια σειρά, χωρίζονται με κόμμα και με την ολοκλήρωση το αρχείο αλλάζει στην επόμενη σειρά και είναι έτοιμο να δεχτεί την επόμενη εγγραφή.

```

void readAndProcessAccelerometerData() {
    mpu.getEvent(&a, &g, &temp);

    // Calculate the total acceleration magnitude
    totalAccel = sqrt(pow(a.acceleration.x, 2) + pow(a.acceleration.y, 2) + pow(a.acceleration.z, 2));

    Serial.print("Accel X: ");
    Serial.print(a.acceleration.x);
    Serial.print(" Y: ");
    Serial.print(a.acceleration.y);
    Serial.print(" Z: ");
    Serial.print(a.acceleration.z);
    Serial.print(" Total Accel: ");
    Serial.println(totalAccel);
}

bool isAnomalyDetected() {
    return (totalAccel < minThreshold || totalAccel > maxThreshold);
}

void captureGPSData() {
    while (Serial.available() > 0) {
        if (gps.encode(Serial.read())) {
            if (gps.location.isValid()) {
                latitude = gps.location.lat();
                lat_str = String(latitude, 6);
                longitude = gps.location.lng();
                lng_str = String(longitude, 6);
                speed = gps.speed.kmph();
                speed_str = String(speed, 4);
                if (gps.date.isValid()) {
                    mergedDate = (gps.date.year() * 10000) + (gps.date.month() * 100) + gps.date.day();
                    mergedDate_str = String(mergedDate);
                }
                if (gps.time.isValid()) {
                    mergedTime = (gps.time.hour() * 10000) + (gps.time.minute() * 100) + gps.time.second();
                    mergedTime_str = String(mergedTime);
                }
            }
            else
                Serial.println("GPS not available");
        }
    }
}

void saveDataToSDCard() {
    File dataFile = SD.open(("Road_data.csv"), FILE_WRITE);
    if (dataFile) {
        String dataString = mergedDate_str + "," + mergedTime_str + "," + lat_str + "," + lng_str + "," + speed_str + "," + totalAccel;
        dataFile.println(dataString);
        dataFile.close();
        Serial.println(dataString);
        delay(1000);
    }
}

```

Εικόνα 30. Οι συναρτήσεις που καλούνται μέσα από την void loop

## 5. Αποτελέσματα

Το πρόγραμμα ανεβάζει όλες τις μετρήσεις σε μια πλατφόρμα της επιλογής μου, συγκεκριμένα στο ThingSpeak. Υπάρχουν πολλές εναλλακτικές (όπως π.χ. χρήση των Google Cloud IoT, AWS IoT, Microsoft Azure IoT κτλ.), ωστόσο επέλεξα αυτή λόγω φιλικότητας προς τον χρήστη, είναι δωρεάν και δεν έχει συλλογή προσωπικών στοιχείων του χρήστη. Στην αρχική

σελίδα εμφανίζονται όλα τα πεδία που έχουμε δημιουργήσει, ωστόσο από μόνα τους δεν μπορούμε να αντλήσουμε εύκολα τις πληροφορίες που επιθυμούμε.

## Bump GPS Logger

Channel ID: ██████████  
Author: johnko1  
Access: Private

A thesis project that detects and collects road anomalies data

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: 7 months ago  
Last entry: 2 minutes ago  
Entries: 173



Εικόνα 31. Αρχική σελίδα του καναλιού στο ThingSpeak



Χρησιμοποιώντας την οπτικοποίηση MATLAB, που είναι ένα από τα εργαλεία που μας παρέχει η πλατφόρμα, μπορούμε να απεικονίσουμε τα δεδομένα που έχουμε συλλέξει όπως επιθυμούμε. Εγώ, ως κομμάτι της εργασίας, τοποθέτησα τις συντεταγμένες πάνω σε χάρτη ώστε να δείξω ότι λειτουργεί και ότι τα αποτελέσματα που καταγράφει είναι σωστά.



Εικόνα 32. Χάρτης που απεικονίζει σημεία του δρόμου που χρειάζονται προσοχή

Επέλεξα κομμάτι της πόλης που γνώριζα ότι έχουν προβλήματα οι δρόμοι λόγω έργων και το ολοκληρωμένο κατάφερε να τα εντοπίσει. Μπορούμε να επιλέξουμε τον αριθμό των δεδομένων που εμφανίζονται στον χάρτη, είτε σαν σύνολο στοιχείων ή ημερομηνίας. Αυτό σημαίνει ότι η απεικόνιση στον χάρτη μπορεί να ελέγχεται και να παρατηρούμε τυχόν



προβλήματα που προκύπτουν μέρα με τη μέρα, οι επισκευές που ολοκληρώθηκαν με επιτυχία. Συνοψίζοντας, έχοντας όλα τα δεδομένα αποθηκευμένα στην πλατφόρμα, έχουμε εύκολη πρόσβαση μέσω API για να τα προβάλουμε όπως και όπου θέλουμε, για παράδειγμα σε κεντρική σελίδα του δήμου.

## **6. Μελλοντικά σχέδια**

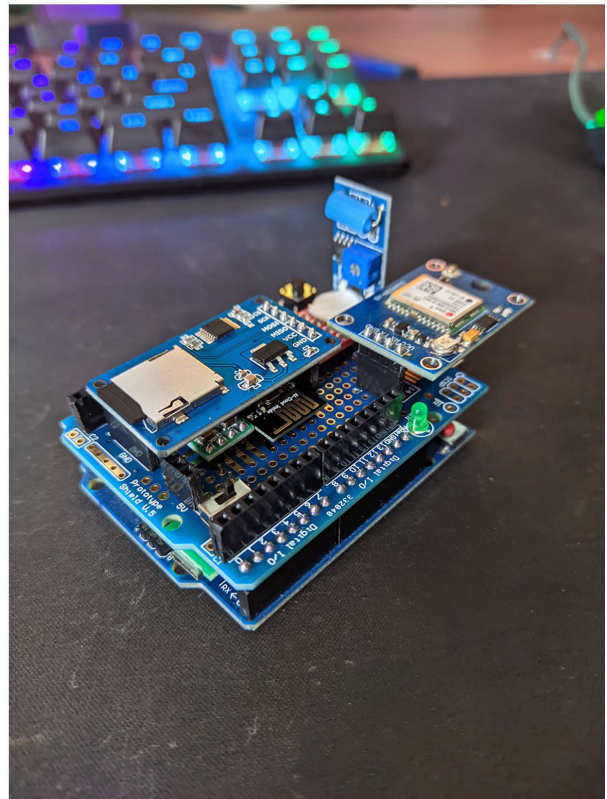
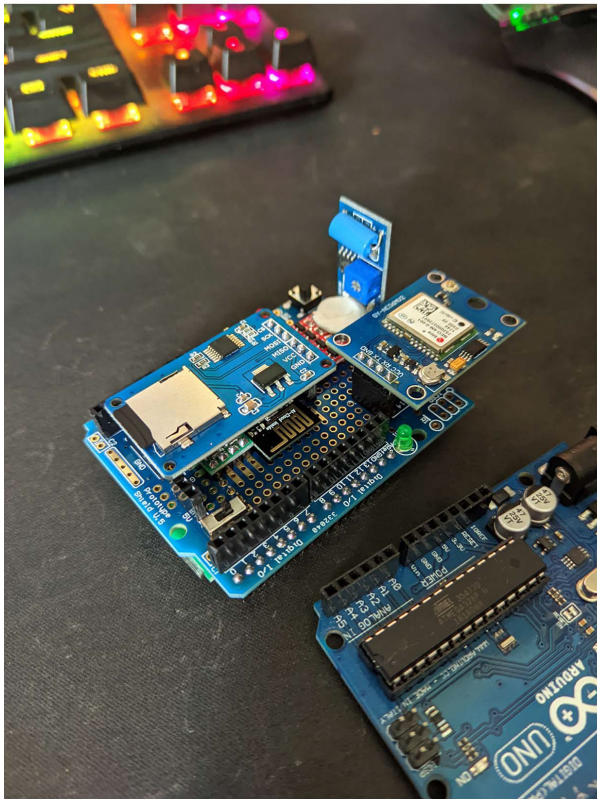
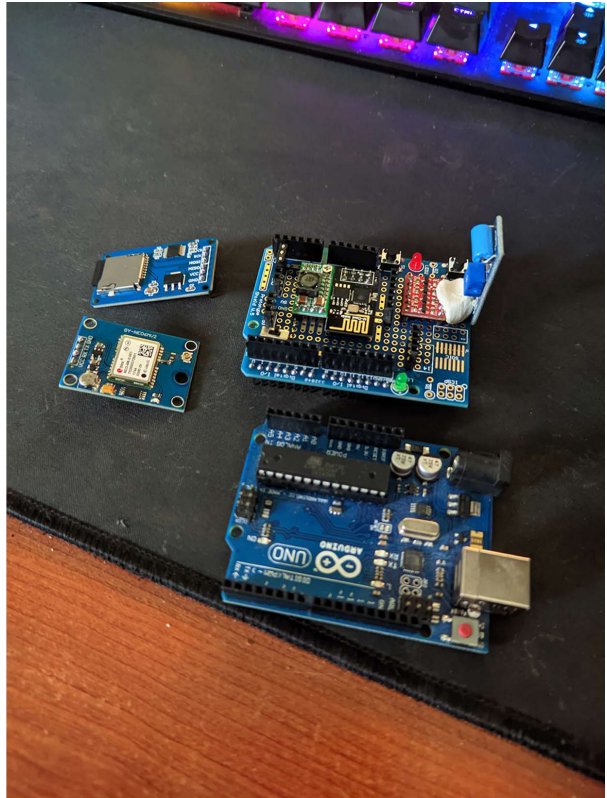
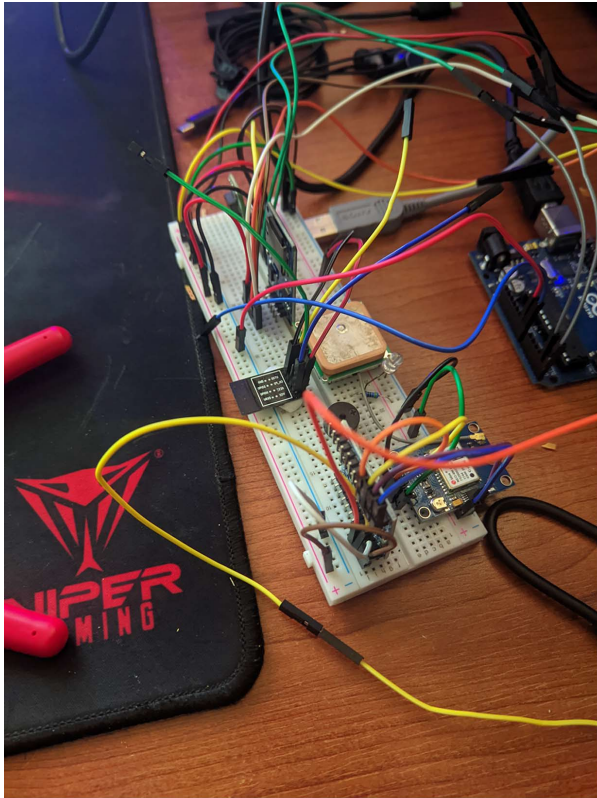
Αρχικά το ESP8266 module που χρησιμοποιώ είναι εγκατεστημένο στο NodeMCU, με ενσωματωμένο ρυθμιστή τάσης και προγραμματιστή ώστε να μπορώ με ευκολία να ανεβάσω στο τσιπ αλλαγές στον κώδικα χωρίς τη χρήση εξωτερικού υλικού. Μελλοντικά μπορεί να χρησιμοποιηθεί ένα εξάρτημα ES8266 χωρίς την πλακέτα ανάπτυξης για να μειωθεί το μέγεθος του τελικού προϊόντος και να αφαιρεθούν εξαρτήματα που χρειάζονται μόνο για τη φάση της ανάπτυξης. Για την τροφοδοσία της συσκευής θα πρέπει να βρεθεί εξάρτημα σταθερής τάσης 3.3 Volt, το οποίο να είναι σε χαμηλή έχοντας γνώμονα την πιθανή μαζική παραγωγή. Οι αναβαθμίσεις σε υλικό είναι εφικτές και εύκολες με μικρές αλλαγές στον κώδικα. Για παράδειγμα να αναβαθμιστεί το επιταχυνσιόμετρο ή το GPS με καλύτερα μοντέλα μεγαλύτερης ακρίβειας.

Όσον αφορά το λογισμικό, οι αλλαγές και ο εμπλουτισμός του κώδικα με επιπλέον λειτουργίες είναι εφικτός, ανάλογα με τις απαιτήσεις που έχουμε. Τα δεδομένα που συλλέγουμε θα μπορούσαμε να τα ανεβάζουμε σε ένα Server στη βάση μας, ώστε να γίνονταν ο έλεγχος τους τοπικά και να επιλέξουμε αν θέλουμε να τα προβάλουμε στο διαδίκτυο απευθείας.

## **7. Επίλογος**

Συνοψίζοντας, η παρούσα εργασία προσπαθεί να δώσει λύση σε ένα πρόβλημα των περισσότερων ελληνικών πόλεων και των δήμων κατ' επέκταση, αυτό της εντόπισης και συντήρησης του κακού οδοστρώματος. Το τελικό ολοκληρωμένο μπορεί να εγκατασταθεί σε οχήματα του δήμου όπου κινούνται στην πλειοψηφία των δρόμων καθημερινά. Η συσκευή θα εντοπίζει τις ανωμαλίες του οδοστρώματος και όταν το όχημα θα επιστρέφει στη βάση του τότε θα συνδέεται μέσω Ίντερνετ σε μια βάση δεδομένων και θα καταχωρεί τα νέα δεδομένα.

Από κει και πέρα είναι στο χέρι του εκάστοτε δήμου να προβεί στις κατάλληλες ενέργειες ώστε να διορθωθούν εγκαίρως οι βλάβες.



Εικόνα 33. Έκδοση 0.1, κατασκευασμένη γύρω από το Arduino Uno

## Βιβλιογραφίες

- [1] ESP 8266, ιστοσελίδα : <https://www.espressif.com/en/products/socs/esp8266>
- [2] Arduino IDE, ιστοσελίδα : <https://docs.arduino.cc/software/ide-v2>
- [3] ThingSpeak IoT, ιστοσελίδα: <https://thingspeak.com/>
- [4] NodeMCU πλακέτα ανάπτυξης, ιστοσελίδα : <https://en.wikipedia.org/wiki/NodeMCU>
- [5] Ublox Neo-6M , ιστοσελίδα: <https://www.u-blox.com/en/product/neo-6-series>
- [6] MPU-6050 , ιστοσελίδα: <https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/>
- [7] Λογισμικό Fritzing, ιστοσελίδα: <https://fritzing.org/>
- [8] Βιβλιοθήκη TinyGPSPlus, ιστοσελίδα: <https://github.com/mikalhart/TinyGPSPlus>
- [9] Βιβλιοθήκη Adafruit\_MPU6050, ιστοσελίδα: [https://github.com/adafruit/Adafruit\\_MPU6050](https://github.com/adafruit/Adafruit_MPU6050)
- [10] Βιβλιοθήκη Adafruit\_Sensor, ιστοσελίδα: [https://github.com/adafruit/Adafruit\\_Sensor](https://github.com/adafruit/Adafruit_Sensor)
- [11] Βιβλιοθήκη SPI, ιστοσελίδα: <https://www.arduino.cc/reference/en/language/functions/communication/spi/>
- [12] Βιβλιοθήκη SD, ιστοσελίδα: <https://www.arduino.cc/reference/en/libraries/sd/>
- [13] Βιβλιοθήκη Wire, ιστοσελίδα: <https://www.arduino.cc/reference/en/language/functions/communication/wire/>
- [14] Βιβλιοθήκη ESP8266WIFI, ιστοσελίδα: <https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html>

## Απόδοση ξενόγλωσσων όρων

<b>Ξενόγλωσση</b>	<b>Απόδοση</b>
Bugs	Σφάλματα
String	Σειρά χαρακτήρων
Online	Διαδικτυακά
Server	Διακομιστής
Task	Εργασία
Test	Δοκιμή
Internet of Things	Διαδίκτυο των Πραγμάτων
Message Queuing Telemetry Transport	Μεταφορά τηλεμετρίας σε ουρά μηνυμάτων
Hypertext Transfer Protocol	Πρωτόκολλο μεταφοράς υπερκειμένου
Application Programming Interface	Διεπαφή προγραμματισμού εφαρμογών
Global	Παγκόσμιος
Comma separated value	Τιμές χωρισμένες με κόμμα
Secret	Μυστικό