



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Έλεγχος κίνησης τετρακίνητου οχήματος μέσω μικροελεγκτή

Ανταράκης Δημήτριος, Α.Μ. 43821
Ματσαϊδώνης Ελπιδοφόρος, Α.Μ. 43859

Εισηγητές: Δρ. Βογιατζής Ιωάννης, Καθηγητής
Ζαχαρένια Γαροφαλάκη, Ε.Δι.Π.



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Έλεγχος κίνησης τετρακίνητου οχήματος μέσω μικροελεγκτή

Ανταράκης Δημήτριος Α.Μ. 43821
Ματσαϊδώνης Ελπιδοφόρος 43859

Τριμελής εξεταστική επιτροπή

1. **Επιβλέπων καθηγητής:** **Ιωάννης Βογιατζής**, Καθηγητής Πανεπιστημίου Δυτικής Αττικής
2. **Μέλος:** **Ζαχαρένια Γαροφαλάκη**, μέλος Ε.Δι.Π. Πανεπιστημίου Δυτικής Αττικής
3. **Μέλος:** **Δημήτριος Καλλέργης**, Λέκτορας Εφ. Πανεπιστημίου Δυτικής Αττικής

Αθήνα, Οκτώβριος 2021

Δήλωση περί μη λογοκλοπής

Δηλώνουμε ότι είμαστε οι συγγραφείς της παρούσας εργασίας με τίτλο:

«Έλεγχος κίνησης τετρακίνητου οχήματος μέσω μικροελεγκτή»

και ότι κάθε βοήθεια την οποία είχαμε για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες κάναμε χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς, είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνουμε ότι αυτή η εργασία έχει συγγραφεί από εμάς αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μας, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μας ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μας.

Επιθυμούμε την απαγόρευση πρόσβασης στο πλήρες κείμενο της εργασίας μας μέχρι 31.10.2022 και έπειτα από αίτηση μας στη Βιβλιοθήκη και έγκριση του επιβλέποντα καθηγητή.

Ανταράκης Δημήτρης
Αριθμός Μητρώου : 43821



(Υπογραφή)

Ματσαϊδώνης Ελπιδοφόρος
Αριθμός Μητρώου : 43859



(Υπογραφή)

Ημερομηνία : Οκτώβριος 2021

ΠΕΡΙΛΗΨΗ

Στόχος αυτής της διπλωματικής εργασίας είναι η κατασκευή, η σχεδίαση και η κατανόηση ενός αυτόνομου ρομποτικού τετρακίνητου οχήματος με χρήση του μικροελεγκτή Arduino UNO και του προγράμματος Arduino IDE, το οποίο θα κινείται αυτόνομα στον χώρο. Το όχημα εκτελεί μία διαδρομή από ένα σημείο A με τελικό προορισμό B σύμφωνα με τις εκάστοτε γεωγραφικές συντεταγμένες που θα του δοθούν. Αυτό επιτυγχάνεται με την βοήθεια ενός GPS συστήματος και μίας ψηφιακής μαγνητικής πυξίδας, με σκοπό η διαδρομή αυτή να εκτελεσθεί σε όσο δυνατό λιγότερο χρόνο, με λιγότερες παύσεις και με μικρότερη απόκλιση από τις συντεταγμένες τερματισμού. Επίσης, το όχημα θα εκτελεί και δεύτερη λειτουργία, όπου με εφαρμογή από το κινητό μέσω σύνδεσης Wi-Fi και μέσω του Διαδικτύου των Πραγμάτων (IoT) θα μπορεί να κινείται σε οκτώ διαφορετικές κατευθύνσεις(πάνω, κάτω, δεξιά, αριστερά και τις 4 διαγώνιες τους) ως τηλεκατευθυνόμενο με χρήση joystick, καθώς θα έχει και την δυνατότητα μείωσης ή αύξησης της ταχύτητας.

ABSTRACT

Our aim in this project is the construction design and understanding of an autonomous robotic four-wheel drive vehicle using Arduino UNO microcontroller and Arduino IDE program, which will move independently in space. The vehicle performs a route from a spot A with final destination B according to the respective geographical coordinate that will be given to it. This task can be accomplished with the help of a GPS system and a digital magnetic compass, so this route can be executed in as little time as possible, with as less pauses and less deviation from the coordinates. Also, the vehicle can execute a second function, which can move in 8 different directions (forward, backward, right, left and the 4 diagonals of them) like remote control with joystick with the help of a phone application connecting with Wi-Fi and via IoT and it has the capability of reducing and increasing its speed.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα θέλαμε να ευχαριστήσουμε θερμά τους δύο υπεύθυνους καθηγητές της διπλωματικής μας, τον κ. Ιωάννη Βογιατζή και την κα. Ζαχαρένια Γαροφαλάκη, για την καθοδήγηση τους, τις συμβουλές τους και την γενική τους βοήθεια σε ακαδημαϊκό και επαγγελματικό επίπεδο, αλλά και τον πρώην υπεύθυνο μας τον κ. Ιωάννη Έλληνα, ο οποίος λόγω συνταξιοδότησης δεν μπόρεσε να παραμείνει υπεύθυνος μας. Τέλος, θα θέλαμε να ευχαριστήσουμε τις οικογένειες μας και τους φίλους μας για την στήριξή τους.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	0
ABSTRACT	Error! Bookmark not defined.
ΚΕΦΑΛΑΙΟ 1ο	9
1.1 ΕΙΣΑΓΩΓΗ.....	9
1.2 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΤΟΥ ARDUINO.....	10
1.3 ΑΝΑΣΚΟΠΗΣΗ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ.....	12
ΚΕΦΑΛΑΙΟ 2ο	15
2.1 ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ.....	15
2.1.1 Αυτόνομο όχημα	15
2.1.2 Μικροελεγκτής	18
2.1.3 GPS (Global Positioning System).....	19
2.1.4 Ηλεκτρονική πυξίδα	20
2.1.5 Wi-Fi	20
2.1.6 Τηλεκατευθυνόμενο Wi-Fi όχημα.....	21
2.1.7 Βιβλιοθήκες.....	21
2.1.8 Πλατφόρμες για δημιουργία εφαρμογών	22
2.1.9 Blynk	22
2.1.10 Internet of Things (IoT)	24
ΚΕΦΑΛΑΙΟ 3ο	27
3.1 ΕΞΟΠΛΙΣΜΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ (ΕΡΓΑΛΕΙΑ).....	27
3.1.1 Γλώσσα προγραμματισμού C++	27
3.1.2 Arduino UNO R3.....	27
3.1.3 L298n Οδηγός DC motor για Arduino	30
3.1.4 DC Motor-Ηλεκτροκινητήρας	31
3.1.5 Σερβοκινητήρας	33
3.1.6 NEO 6M GPS	34
3.1.7 Πυξίδα QMC5883L	35
3.1.8 ESP 8266.....	36
3.1.9 NodeMCU	37
3.1.10 LEDs	38
3.2 ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ (ΑΥΤΟΝΟΜΟ ΟΧΗΜΑ)	41
3.2.1 Εισαγωγή στο πρόγραμμα.....	41
3.2.2 Βιβλιοθήκες.....	41
3.2.3 Περιγραφή βιβλιοθηκών.....	41

3.2.4 Κώδικας	42
3.3 ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΕΞΥΠΝΗ ΣΥΣΚΕΥΗ.....	55
3.3.1 Εισαγωγή στην πλατφόρμα Blynk	55
3.3.2 Joystick	59
3.3.3 Βιβλιοθήκες.....	61
3.3.4 Περιγραφή βιβλιοθηκών.....	61
3.3.5 Κώδικας	62
ΚΕΦΑΛΑΙΟ 4ο	69
4.1 ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ	69
4.1.1 Υλικά	69
4.1.2 Συναρμολόγηση του Τετράτροχου Arduino οχήματος	72
4.1.3 Συνδεσμολογία του τετράτροχου Arduino οχήματος.....	82
4.1.4 Η τελική εικόνα του οχήματος.....	84
ΚΕΦΑΛΑΙΟ 5ο	87
5.1 ΑΠΟΤΕΛΕΣΜΑΤΑ – ΠΡΟΒΛΗΜΑΤΑ ΠΟΥ ΑΝΤΙΜΕΤΩΠΙΣΑΜΕ	87
5.1.1 Αποτελέσματα.....	87
5.1.2 Προβλήματα που αντιμετωπίσαμε.....	89
ΚΕΦΑΛΑΙΟ 6ο	91
6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ	91
ΚΕΦΑΛΑΙΟ 7ο	93
7.1 ΒΙΒΛΙΟΓΡΑΦΙΑ.....	93

ΠΕΡΙΕΧΟΜΕΝΑ ΕΙΚΟΝΩΝ

Εικόνα 1 Στιγμιότυπο του λογισμικού του Arduino.....	10
Εικόνα 2 Αυτόνομο όχημα και συστήματα υποβοήθησης (Radar, LIDAR, Camera) [4].....	17
Εικόνα 3 Απεικόνιση GPS [8].....	20
Εικόνα 4 Wi-Fi Alliance [11]	21
Εικόνα 5 Τα στοιχεία της πλατφόρμας Blynk [14]	23
Εικόνα 6 Επίπεδα IoT	24
Εικόνα 7 Διαδίκτυο των πραγμάτων. Σύνδεση από παντού. [16]	25
Εικόνα 8 Arduino UNO R3 [1]	28
Εικόνα 9 Ακροδέκτες (Pins) Arduino UNO [2].....	30
Εικόνα 10 L298n DC motor [18].....	31
Εικόνα 11 DC Κινητήρας.....	32
Εικόνα 12 Σερβοκινητήρας SG90 (9G)	33
Εικόνα 13 GPS Module for Arduino [21]	34
Εικόνα 14 Πυξίδα QMC5883L.....	35
Εικόνα 15 ESP-12F.....	36
Εικόνα 16 NodeMCU development kit	38
Εικόνα 17 LEDs σε διάφορα χρώματα	40
Εικόνα 18 Blynk home screen.....	55
Εικόνα 19 Blynk widget box	56
Εικόνα 20 widgets (LCD, Slider, Joystick, Button).....	57
Εικόνα 21 Blynk η εφαρμογή της εργασίας	58
Εικόνα 22 Blynk Project settings.....	59
Εικόνα 23 η λογική πίσω από το joystick.....	60
Εικόνα 24 Εξάρτημα μότορας	72
Εικόνα 25 Εξάρτημα μότορας	72
Εικόνα 26 Εξαρτήματα μότορας-σασί	73
Εικόνα 27 Εξαρτήματα μότορες-σασί.....	73
Εικόνα 28 Εξαρτήματα L298	74
Εικόνα 29 L298	74
Εικόνα 30 Εξαρτήματα σασί-L298	75
Εικόνα 31 Εξαρτήματα IR TM	75
Εικόνα 32 Εξαρτήματα	76
Εικόνα 33 Εξαρτήματα	76
Εικόνα 34 Εξαρτήματα Servo.....	77
Εικόνα 35 Εξαρτήματα ultra-sonic sensor.....	77
Εικόνα 36 Εξαρτήματα Arduino.....	78
Εικόνα 37 Εξαρτήματα Arduino.....	78
Εικόνα 38 Εξαρτήματα GPS	79
Εικόνα 39 Εξαρτήματα μπαταρία	79
Εικόνα 40 Εξαρτήματα Compass.....	80
Εικόνα 41 Breadboard power supply module	81
Εικόνα 42 NodeMCU module.....	81
Εικόνα 43 white and red LED.....	82
Εικόνα 44 το όχημα μας (μπροστά)	85

Εικόνα 45 το όχημα μας (πίσω) 86

ΠΕΡΙΕΧΟΜΕΝΑ ΠΙΝΑΚΩΝ

Πίνακας 3.1	Χαρακτηριστικά Μικροελεγκτή [1]	28
Πίνακας 3.2	L298N Διαμόρφωση των ακροδεκτών[9]:	30
Πίνακας 3.3	Χαρακτηριστικά DC Motor	32
Πίνακας 3.4	QMC5883L Διαμόρφωση PIN [22]	35
Πίνακας 3.5	Δήλωση Βιβλιοθηκών	42
Πίνακας 3.6	Δήλωση Global Μεταβλητών	43
Πίνακας 3.7	Αρχικοποιήσεις Αντικειμένων και εξαρτημάτων	44
Πίνακας 3.8	Συνάρτηση setup()	45
Πίνακας 3.9	Βρόγχος επανάληψης κυρίου προγράμματος	45
Πίνακας 3.10	Συναρτήσεις Μετατροπής και στρογγυλοποίησης Δεδομένων	46
Πίνακας 3.11	Συνάρτηση λήψης δεδομένων από το GPS	47
Πίνακας 3.12	Συνάρτηση Υπολογισμού Απόστασης Τελικού προορισμού	49
Πίνακας 3.13	Συνάρτηση υπολογισμού κατεύθυνσης οχήματος	50
Πίνακας 3.14	Συνάρτηση υπολογισμού κατεύθυνσης από την πυξίδα	50
Πίνακας 3.15	Συνάρτηση ορισμού πορείας οχήματος	51
Πίνακας 3.16	Συνάρτηση τερματισμού των Μοτέρ	53
Πίνακας 3.17	Συνάρτηση κίνησης οχήματος ευθεία	53
Πίνακας 3.18	Συνάρτηση κίνησης οχήματος αριστερά	53
Πίνακας 3.19	Συνάρτηση κίνησης του οχήματος δεξιά	54
Πίνακας 3.20	Δήλωση Βιβλιοθηκών	62
Πίνακας 3.21	Δήλωση μεταβλητών και αρχικοποιήσεις	62
Πίνακας 3.22	Βρόγχος επανάληψης κυρίου προγράμματος	64
Πίνακας 3.23	Διεργασίες της Βιβλιοθήκης Blynk	66
Πίνακας 4.1	Υλικά για την Εργασία	69
Πίνακας 5.1	Μετρήσεις Οχήματος σε ανώμαλο έδαφος	88
Πίνακας 5.2	Μετρήσεις Οχήματος σε ομαλό έδαφος	88

ΚΕΦΑΛΑΙΟ 1ο

1.1 ΕΙΣΑΓΩΓΗ

Η μετακίνηση ενός αυτόνομου οχήματος, δηλαδή ένα όχημα το οποίο μπορεί να αντιλαμβάνεται το περιβάλλον του και να πλοηγείται χωρίς ανθρώπινη παρέμβαση, από ένα σημείο A σε ένα σημείο B συντεταγμένων με τη χρήση Global Positioning System (GPS) επιφέρει πολλές δυσκολίες. Κάποιες από αυτές τις δυσκολίες είναι η επικοινωνία με απομακρυσμένους δορυφόρους για την λήψη της σωστής τοποθεσίας του οχήματος. Αυτή η επικοινωνία μπορεί να παρεμποδίζεται, εξαιτίας ατμοσφαιρικών συνθηκών, αλλαγών στην τροχιά των δορυφόρων, παρεμβολές (π.χ. από κάποιο κτήριο, τοίχο) και ύπαρξη αντανεκλαστικών επιφανειών κοντά στον δέκτη. Επίσης, μία άλλη δυσκολία είναι ο υπολογισμός της απόστασης του τελικού προορισμού, με πιθανή απόκλιση μερικών εκατοστών ή μέτρων εξαιτίας των σφαλμάτων που αναφέρθηκαν παραπάνω. Τέλος, σημαντικό ρόλο στην εκτέλεση της σωστής μετακίνησης του αυτόνομου οχήματος παίζει και η ομαλότητα ή μη της διαδρομής (έδαφος, μικρά εμπόδια, πέτρες) σε συνδυασμό με τον τύπο του οχήματος.

Ένας άλλος τρόπος για την μετακίνηση ενός οχήματος, είναι ο τηλεκατευθυνόμενος, δηλαδή ο χρήστης της εφαρμογής, η οποία επικοινωνεί με το όχημα να έχει τον πλήρη έλεγχο της κίνησης του οχήματος. Ένας τρόπος επικοινωνίας μεταξύ εφαρμογής και οχήματος, ώστε να επιτευχθεί η τηλεκατευθυνόμενη λειτουργία είναι με χρήση Wi-Fi, όπου η σύνδεση γίνεται μέσω του NodeMCU μικροελεγκτή, δηλαδή ενός τύπου επεξεργαστή, που διαθέτει πολλά ενσωματωμένα υποσυστήματα. Η διασύνδεση μεταξύ NodeMCU και χρήστη γίνεται με την χρήση της πλατφόρμας Blynk, η οποία δίνει την δυνατότητα στον χρήστη να ελέγξει το υλικό από απόσταση (εφαρμογή) και έτσι η κίνηση δεν γίνεται αυτόματα αλλά όπως εκείνος επιθυμεί.

Τα προβλήματα που προσπαθήσαμε να επιλύσουμε είναι α) η ελαχιστοποίηση των σφαλμάτων της κίνησης και θέσης του οχήματος, όπως η προσέγγιση του τελικού προορισμού με τη μικρότερη δυνατή απόκλιση μέτρων ή εκατοστών, β) η κατά το δυνατόν ομαλότερη πορεία του οχήματος χωρίς παρεκκλίσεις και διακοπές και γ) η εύρεση των κατάλληλων εργαλείων για την επίτευξη της επικοινωνίας μεταξύ οχήματος και κινητής έξυπνης συσκευής μέσω του διαδικτύου, με την λιγότερη δυνατή καθυστέρηση.

1.2 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΤΟΥ ARDUINO

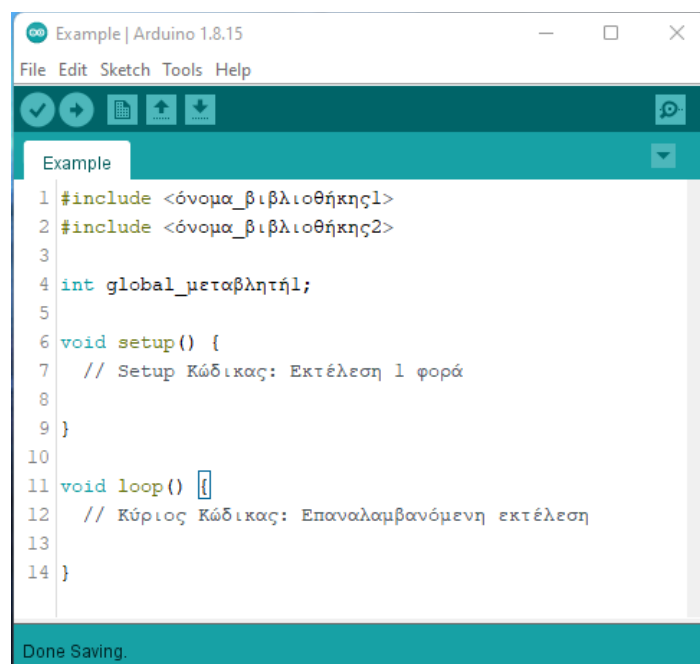
Το Arduino είναι ένας μικροελεγκτής μονής πλακέτας, δηλαδή μια μητρική πλακέτα ανοικτού κώδικα με μικροελεγκτή ενσωματωμένο με εισόδους και εξόδους. Προγραμματίζεται σε γλώσσα Processing, η οποία είναι ένας συνδυασμός γλώσσας C και C++ και είναι μία μεταγλωττιζόμενη γλώσσα πολλαπλών παραδειγμάτων με τύπους. Επίσης αποτελείται από ένα σύνολο βιβλιοθηκών ονομαζόμενο ως Wiring, δηλαδή μια συλλογή από έτοιμα υποπρογράμματα που χρησιμοποιούνται για την ανάπτυξη κώδικα.

Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μια εφαρμογή γραμμένη σε γλώσσα Java και περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα. [1]

Η σύνταξη του προγράμματος χωρίζεται σε 4 κομμάτια [1]:

1. Ορισμός Βιβλιοθηκών και Global Μεταβλητών, δηλαδή μεταβλητών για χρήση από όλο τον κώδικα.
2. Συνάρτηση setup(): μία συνάρτηση που τρέχει μία φορά στην αρχή του προγράμματος η οποία αρχικοποιεί τις ρυθμίσεις και τις παραμέτρους των βιβλιοθηκών και των εξωτερικών εξαρτημάτων που έχουν προστεθεί.
3. Συνάρτηση loop(): Η κύρια συνάρτηση του προγράμματος όπου επανεκτελείται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί.
4. Τυχών συναρτήσεις που είτε προσθέτει ο χρήστης είτε συναρτήσεις που είναι ενσωματωμένες σε κάποια βιβλιοθήκη και χρειάζονται δήλωση παραμέτρων ώστε να εκτελεστούν.

Χρησιμοποιείται για ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων και συνδέεται με υπολογιστή μέσω προγραμμάτων. [1]



```
Example | Arduino 1.8.15
File Edit Sketch Tools Help
Example
1 #include <όνομα_βιβλιοθήκης1>
2 #include <όνομα_βιβλιοθήκης2>
3
4 int global_μεταβλητή1;
5
6 void setup() {
7   // Setup Κώδικας: Εκτέλεση 1 φορά
8
9 }
10
11 void loop() {
12   // Κύριος Κώδικας: Επαναλαμβανόμενη εκτέλεση
13
14 }
Done Saving.
```

Εικόνα 1 Στιγμιότυπο του λογισμικού του Arduino

Το 2005 δημιουργήθηκε ένα σχέδιο για να φτιαχτεί μία φθηνότερη συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές. Οι ιδρυτές Massimo Banzi και David Cueartielles το ονόμασαν Arduino και ξεκίνησαν να παράγουν τις πλακέτες.

Μια πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη. Η πλακέτα περιλαμβάνει έναν γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz. Η πλακέτα είναι σχεδιασμένη έτσι ώστε να μην χρειάζεται εξωτερική πλακέτα προγραμματισμού. Τέλος, ο πίνακας Arduino παρέχει 14 ψηφιακούς I/O ακροδέκτες από τους οποίους οι 6 υποστηρίζουν παλμομετρική διαμόρφωση σήματος (PWM). Επίσης προσφέρει 6 αναλογικούς ακροδέκτες εισόδου. [1]

Τα Πλεονεκτήματα του Μικροελεγκτή Arduino [2]:

➤ **Χαμηλό κόστος**

Μια πλακέτα Arduino έχει πολύ προσιτή τιμή και είναι ευρέως διαθέσιμη για αγορά. Επίσης υπάρχουν διάφορα Arduino Starter Kits που εκτός από την πλακέτα Arduino, διαθέτουν και διάφορα εξαρτήματα χρήσιμα για τους νέους χρήστες, ώστε να ξεκινήσουν βασικά projects.

➤ **Απλότητα**

Η απλότητα του Arduino μπορεί να προσφέρει στον χρήστη μεγάλη διευκόλυνση, ο οποίος θα καταφέρει σε μικρό χρονικό διάστημα να υλοποιήσει την πρώτη του κατασκευή. Είναι ένα εργαλείο κατάλληλο για την απόκτηση ηλεκτρονικών και μηχανικών δεξιοτήτων.

➤ **Διαπλατφορμική Δομή (Cross-platform structure)**

Το πρόγραμμα διαμόρφωσης και προγραμματισμού (IDE) του Arduino είναι συμβατό με πληθώρα λειτουργικών συστημάτων, προσφέροντας έτσι σε όλους τους χρήστες την δυνατότητα να το χρησιμοποιήσουν.

➤ **Μεγάλο εύρος εκδόσεων και τύποι μοντέλων**

Η πλακέτα Arduino είναι μια οικογένεια από πλακέτες μικροελεγκτών και κάθε έκδοση έχει διαφορετικές δυνατότητες, συνδέσεις (pins) καθώς και μέγεθος, έτσι ώστε να καλύπτει διαφορετικές ανάγκες.

➤ **Πλακέτες επέκτασης (Shields)**

Οι πλακέτες επέκτασης είναι ένα δυνατό εργαλείο της οικογένειας Arduino οι οποίες προσφέρουν ενίσχυση και νέες δυνατότητες στην υπάρχουσα πλακέτα. Είναι ολοκληρωμένες πλακέτες κατάλληλα σχεδιασμένες ώστε να εφαρμόζονται στο Arduino. Μερικές από τις πιο δημοφιλείς πλακέτες επέκτασης του Arduino είναι:

1. Wi-Fi Shield: δίνει την δυνατότητα στο Arduino να αποκτήσει πρόσβαση στο Διαδίκτυο μέσω ασύρματης σύνδεσης Wi-Fi
2. Ethernet Shield: δίνει την δυνατότητα στο Arduino να αποκτήσει πρόσβαση στο Διαδίκτυο μέσω καλωδίου Ethernet (RJ-45)
3. GPS Shield: Προσδίδει στο Arduino δυνατότητες GPS όπως ο εντοπισμός θέσης, πλοήγηση κ.α.
4. LCD Shield: Προσθέτει οθόνη LCD στο Arduino.

5. Motor Shield: Επιτρέπει την εύκολη οδήγηση Μοτέρ από το Arduino.

➤ **Μεγάλη κοινότητα υποστήριξης**

Επειδή μεγάλο πλήθος ανθρώπων ασχολείται με την οικογένεια του Arduino και στον λογισμικό (προγράμματα, βιβλιοθήκες) αλλά και στον υλικό τομέα του (συνδεσμολογία, εξαρτήματα), έχει ως αποτέλεσμα να έχουν δημιουργηθεί πολλές ιστοσελίδες και forums τα οποία μπορούν να προσφέρουν ιδέες για ολοκληρωμένες υλοποιήσεις καθώς επίσης και λίστα ερωτήσεων και απαντήσεων σε θέματα λειτουργίας του Arduino.

➤ **Ανοικτού κώδικα Λογισμικό και Hardware**

Οι χρήστες μέσα από τις βιβλιοθήκες C++ και το Arduino IDE μπορούν να γράψουν τον δικό τους κώδικα και να τον μοιραστούν με άλλους χρήστες. Επίσης τα σχέδια των πλατφορμών είναι ανοικτού κώδικα και οι χρήστες μπορούν να επεκτείνουν και να αναβαθμίσουν τις πλατφόρμες, ώστε να τις φέρουν στα μέτρα τους. [3]

1.3 ΑΝΑΣΚΟΠΗΣΗ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Για την εργασία αυτή μελετήσαμε και ερευνήσαμε την υπάρχουσα βιβλιογραφία ως βάσεις πάνω στο τρέχων θέμα. Έπειτα, παραγγείλαμε τα κατάλληλα εξαρτήματα που χρειαζόμασταν για την υλοποίηση της εργασίας.. Αφού προμηθευτήκαμε το Arduino, τους ρότορες μια μπαταρία λιθίου ισχύος για να μπορεί να αντέχει τους ρότορες , το NodeMCU, την πυξίδα κ.α., ξεκινήσαμε να συναρμολογούμε σιγά σιγά το αυτόνομο τετράτροχο όχημα πάνω στον σκελετό, μελετώντας τα χαρακτηριστικά των εξαρτημάτων για την σωστή λειτουργία τους. Μετά, κάναμε τους απαραίτητους ελέγχους στα παραπάνω εξαρτήματα και προχωρήσαμε στην ανάπτυξη του κύριου κώδικα όπου κληθήκαμε να επιλύσουμε τυχών προβλήματα.

Ένα από αυτά τα προβλήματα είναι για παράδειγμα η ρύθμιση της ταχύτητας των μοτέρ μέσω του L298n οδηγητή, καθώς θα πρέπει να ληφθεί υπόψιν ο τύπος του οδοστρώματος, το βάρος του οχήματος καθώς και η απόκλιση στην κατασκευή των ίδιων των μοτέρ.

Προβλήματα επίσης αντιμετωπίσαμε και στο υλικό κομμάτι της εργασίας, για παράδειγμα η πρώτη μπαταρία που δοκιμάσαμε δεν μας ικανοποίησε από θέμα απόδοσης, με αποτέλεσμα να την αντικαταστήσουμε με μπαταρία λιθίου μεγαλύτερης ισχύος και χωρητικότητας χωρίς το μεγαλύτερο φυσικό μέγεθος της να εμποδίζει την σωστή λειτουργία του οχήματος.

Έπειτα από την επίλυση των παραπάνω προβλημάτων, επιτύχαμε να το προγραμματίσουμε ώστε να μετακινείται από ένα σημείο Α σε ένα σημείο Β και επικεντρωθήκαμε στη βελτίωση του κώδικα έτσι ώστε να μειωθεί ο χρόνος, οι διακοπές καθώς και οι αποκλίσεις αυτής της διαδρομής.

Για την τηλεκατευθυνόμενη κίνηση του οχήματος, επιλέξαμε την κατάλληλη εφαρμογή για την επικοινωνία με το NodeMCU. Καταλήξαμε στην εφαρμογή Blynk, την οποία προτιμήσαμε γιατί παρείχε τις απαραίτητες και κατάλληλες δυνατότητες

που χρειαζόμασταν όπως τα κατάλληλα γραφικά στοιχεία για την επίτευξη της λειτουργίας και φιλικό περιβάλλον προς τον χρήστη. Τέλος, παρέχει δωρεάν εξυπηρετητή για έλεγχο και επικοινωνία μέσω του Διαδικτύου, κάτι το οποίο δεν θα μπορούσε να υλοποιηθεί εύκολα με μία εφαρμογή που θα δημιουργούσαμε από το μηδέν. Έπειτα, θα έπρεπε να δημιουργήσουμε την επικοινωνία μεταξύ της εφαρμογής και του οχήματος. Καταλήξαμε στην λύση του NodeMCU, το οποίο προσφέρει δικό του περιβάλλον προγραμματισμού, παραπάνω εισόδους και εξόδους, καθώς και παραμετροποιήσεις στις ρυθμίσεις του.

Στο 2^ο κεφάλαιο της εργασίας, ασχοληθήκαμε με το θεωρητικό υπόβαθρο σε ορισμούς και θεωρίες. Στο 3^ο κεφάλαιο αναφέρουμε τα εργαλεία, μεθόδους, εξαρτήματα και λογισμικά που χρησιμοποιήσαμε καθώς επίσης αναλύουμε και τον κώδικα που γράψαμε για να λειτουργήσει σωστά το όχημα και στις δύο λειτουργίες που χρειάστηκε να εκτελέσουμε. Στο 4^ο κεφάλαιο, παρουσιάζεται η υλοποίηση, δηλαδή η συνδεσμολογία του υλικού και η τελική του εικόνα. Στο 5^ο κεφάλαιο, αναφέρουμε τα αποτελέσματα της εργασίας αλλά και προβλήματα που επιλύσαμε και στο 6^ο κεφάλαιο αναφέρουμε τα συμπεράσματα αλλά και μελλοντικά σχέδια για την βελτίωση και αναβάθμιση της εφαρμογής και του οχήματος.

ΚΕΦΑΛΑΙΟ 2ο

2.1 ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

2.1.1 Αυτόνομο όχημα

Αυτόνομο είναι ένα όχημα το οποίο με την βοήθεια αισθητήρων και άλλων εργαλείων μπορεί να αντιλαμβάνεται το περιβάλλον του και πλοηγείται χωρίς ανθρώπινη παρέμβαση. Ενσωματώνει τεχνολογίες όπως:

- Radar: Είναι ένα σύστημα ανίχνευσης που χρησιμοποιεί ραδιοκύματα για να καθορίσει την γωνία, το εύρος ή την ταχύτητα των αντικειμένων.
- GPS: Είναι ένα δορυφορικό σύστημα πλοήγησης που παρέχει πληροφορίες γεωγραφικής τοποθεσίας και χρόνου.
- Lidar: Είναι μία μέθοδος τηλεπισκόπησης που χρησιμοποιεί το φως με τη μορφή παλμικού λέιζερ για τη μέτρηση εύρους (μεταβλητές αποστάσεις) έως τη Γη. Αυτή η τεχνολογία χρησιμοποιείται για τη σάρωση δρόμων και κτιρίων.
- Κάμερα: Η κάμερα επιτρέπει στα οχήματα να αναγνωρίζουν αντικείμενα και ανθρώπους σε πραγματικό χρόνο, χάρη στην πρόοδο της όρασης υπολογιστών.

Επίσης, ένα αυτόνομο όχημα διαθέτει προηγμένο λογισμικό που συνδυάζει τις διαθέσιμες πληροφορίες, ώστε να εντοπίζει και να αποφεύγει τα πιθανά εμπόδια και να βρίσκει την πιο σωστή διαδρομή.

Το πρώτο πράγμα που χρειάζεται το αυτόνομο όχημα είναι να έχει την δυνατότητα να ανιχνεύσει τη θέση του στο χώρο. Για να γίνει αυτό, ένα αυτόνομο όχημα πρέπει να έχει έναν χάρτη HDM (High Definition map - χάρτη υψηλής ευκρίνειας) ο οποίος περιλαμβάνει πολλά δεδομένα σχετικά με το δρόμο και το περιβάλλον, όπως ανίχνευση εμποδίων, ανθρώπων, άλλων αυτοκινήτων, καθώς και οδικών σημάτων δηλαδή γραμμές δρόμων, πινακίδες κ.α. τα οποία αντλεί από τα συστήματα που αναλύσαμε παραπάνω. Επίσης, το αυτόνομο όχημα χρειάζεται ένα σύστημα εκτίμησης της κατάστασης της κίνησής του. Το σύστημα εκτίμησης κατάστασης, με βάσει τα δεδομένα από τους αισθητήρες του οχήματος, θα δημιουργήσει μια γεωγραφική θέση για το όχημα εντός του χάρτη υψηλής ευκρίνειας. Μετά, θα λαμβάνει πληροφορίες από κάθε είδους διαφορετικούς αισθητήρες του οχήματος και θα ενημερώνει την γεωγραφική του θέση ανάλογα με τις πληροφορίες που λαμβάνει. Διαφορετικές καταστάσεις μπορεί να ευνοούν χρήση διαφορετικών αισθητήρων. Για παράδειγμα, εάν το όχημα βρίσκεται μέσα σε ένα κτήριο, το σήμα GPS ενδέχεται να μην είναι αξιόπιστο και το σύστημα εκτίμησης κατάστασης να χρειαστεί να βασιστεί σε άλλους αισθητήρες όπως το Lidar, το ραντάρ και την κίνηση των ελαστικών για την ενημέρωση της γεωγραφικής θέσης του οχήματος. [4]

Πειράματα έχουν πραγματοποιηθεί σε αυτόνομα συστήματα οδήγησης τουλάχιστον από την δεκαετία του 1920. Οι δοκιμές ξεκίνησαν τη δεκαετία του 1950. Το πρώτο ημιαυτόνομο αυτοκίνητο αναπτύχθηκε το 1977, από το ιαπωνικό εργαστήριο Tsukuba Mechanical Engineering Laboratory, το οποίο απαιτούσε ειδικά σηματοδοτημένους δρόμους. Η ειδική αυτή σηματοδότηση συλλαμβάνονταν από δύο κάμερες στο όχημα και επεξεργάζονταν από έναν αναλογικό υπολογιστή. Τέτοιου τύπου οχήματα κινούνταν με μέγιστη ταχύτητα 30 χιλιόμετρα την ώρα ώστε ο υπολογιστής να προλαβαίνει να επεξεργάζεται τα δεδομένα.. Πλέον σήμερα, έχουν κατασκευαστεί αυτόνομα αυτοκίνητα από εταιρείες όπως η Tesla και η Robocar, η οποία έκανε το παγκόσμιο ρεκόρ Guinness το 2019, παρουσιάζοντας το γρηγορότερο αυτόνομο αυτοκίνητο στον κόσμο με ταχύτητα 282,42 χλμ./Ωρα. [5]

Πλεονεκτήματα χρήσης αυτόνομων αυτοκινήτων [5]

- Μείωση των τροχαίων ατυχημάτων από ανθρώπινο λάθος
- Μείωση μποτιλιαρισμάτων
- Βελτίωση της κυκλοφορίας
- Ξεκούραστο ταξίδι για τον οδηγό
- Μειωμένες έως καθόλου παραβάσεις
- Νέες θέσεις εργασίας γύρω από τα αυτόνομα οχήματα

Μειονεκτήματα χρήσης αυτόνομων αυτοκινήτων [5]

- Κόστος
- Θέμα εμπιστοσύνης και ασφάλειας από τους επιβάτες μέχρι τους πεζούς ή τους άλλους οδηγούς
- Πιθανό πρόβλημα στην συνύπαρξη αυτόνομων και μη αυτόνομων αυτοκινήτων.
- Άνθρωποι που επαγγέλλονται οδηγοί μπορεί να μείνουν χωρίς δουλειά.
- Ανταλλαγή πληροφοριών από ένα αυτόνομο όχημα σε ένα άλλο για τους επιβάτες.

Επίπεδα Αυτοματισμού [5]

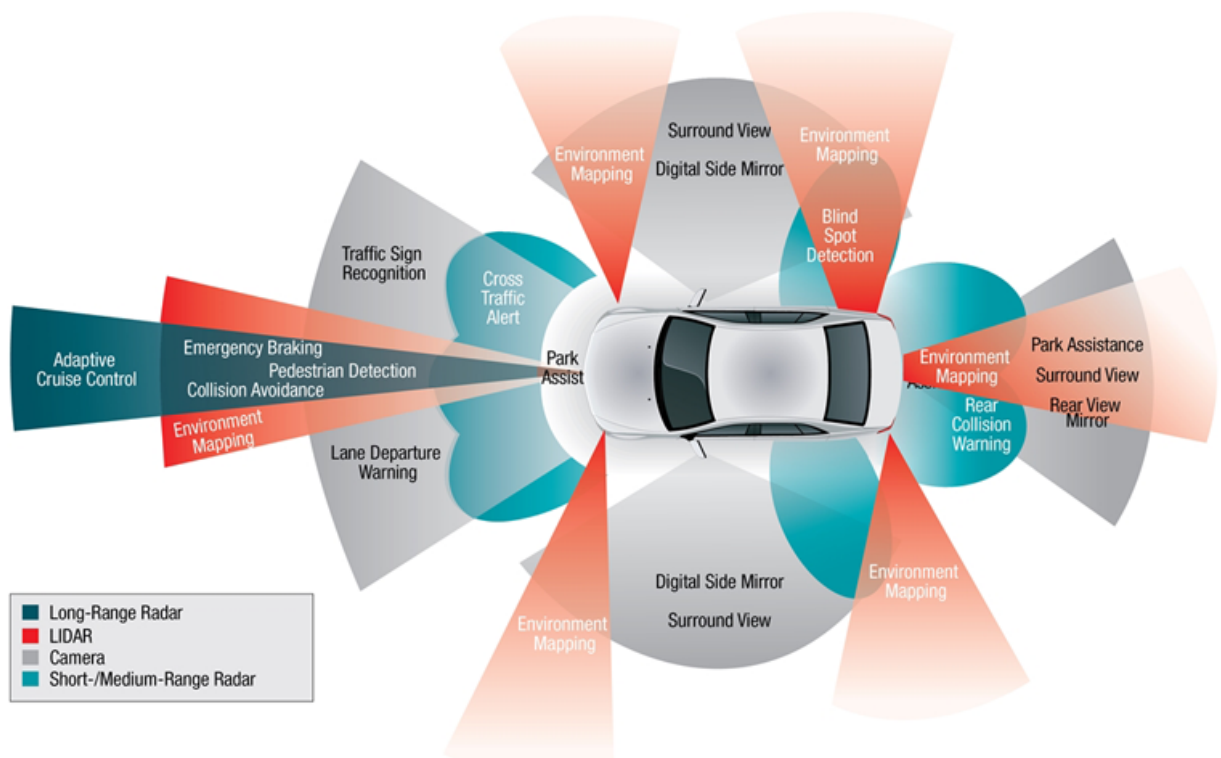
Τα επίπεδα αυτοματισμού ενός οχήματος χωρίζονται σε 2 κατηγορίες:

- 1) Ο Χρήστης-Οδηγός επιβλέπει το περιβάλλον οδήγησης:
 - **Επίπεδο 0 Κανένας Αυτοματισμός:** Ο χρήστης-οδηγός έχει τον πλήρη έλεγχο σε όλες τις πτυχές της οδήγησης.
 - **Επίπεδο 1 Βοήθεια οδηγού:** Εκτέλεση από το σύστημα υποβοήθησης του οδηγού για τον τρόπο οδήγησης, είτε στην κατεύθυνση είτε στην επιτάχυνση/επιβράδυνση, χρησιμοποιώντας πληροφορίες σχετικά με το περιβάλλον οδήγησης.

- **Επίπεδο 2 Μερική Αυτοματοποίηση:** Εκτέλεση από ένα ή περισσότερα συστήματα υποβοήθησης του οδηγού για τον τρόπο οδήγησης, είτε στην κατεύθυνση είτε στην επιτάχυνση/επιβράδυνση, χρησιμοποιώντας πληροφορίες σχετικά με το περιβάλλον οδήγησης.

2) Το αυτόνομο σύστημα οδήγησης επιβλέπει το περιβάλλον οδήγησης :

- **Επίπεδο 3 Αυτοματισμός υπό όρους:** Η απόδοση σχετίζεται με τον τρόπο οδήγησης από ένα αυτοματοποιημένο σύστημα οδήγησης με την προσδοκία ότι ο οδηγός θα ανταποκριθεί κατάλληλα σε αίτημα παρέμβασης.
- **Επίπεδο 4 Υψηλός Αυτοματισμός:** Η απόδοση σχετίζεται με τον τρόπο οδήγησης από ένα αυτοματοποιημένο σύστημα οδήγησης ακόμη και αν ένας οδηγός δεν ανταποκριθεί κατάλληλα σε ένα αίτημα παρέμβασης, το αυτοκίνητο μπορεί να κάνει στην άκρη με ασφάλεια από το σύστημα καθοδήγησης.
- **Επίπεδο 5 Πλήρης Αυτοματισμός:** Η απόδοση σχετίζεται με τον τρόπο οδήγησης από ένα αυτοματοποιημένο σύστημα οδήγησης κάτω από όλες τις οδικές και περιβαλλοντικές συνθήκες που θα μπορούσε να διαχειριστεί και ένας οδηγός.



Εικόνα 2 Αυτόνομο όχημα και συστήματα υποβοήθησης (Radar, LIDAR, Camera) [4]

2.1.2 Μικροελεγκτής

Ο μικροελεγκτής (microcontroller) είναι ένας τύπος επεξεργαστή, ο οποίος μπορεί να λειτουργήσει με ελάχιστα εξωτερικά κομμάτια επειδή διαθέτει πολλά ενσωματωμένα υποσυστήματα, όπως έναν ή περισσότερους επεξεργαστές, μαζί με μνήμη και προγραμματιζόμενες περιφερειακές συσκευές εισόδου-εξόδου. Χρησιμοποιείται σε όλα τα ενσωματωμένα συστήματα μεσαίου και χαμηλού κόστους.

Το ολοκληρωμένο αυτό κύκλωμα περιέχει την Λογική και Αριθμητική Μονάδα (ALU – Arithmetic Logic Unit), καταχωρητές, μία προσωρινή μνήμη RAM υψηλής ταχύτητας και τον ελεγκτή μνήμης.

Μερικοί από τους πιο γνωστούς κατασκευαστές μικροελεγκτών είναι οι [6]:

- Atmel
- ARM
- Epson
- Hitachi
- NEC
- Toshiba
- Microchip κ.α.

Υπάρχουν αρκετές κατηγορίες μικροελεγκτών, όπως [6]:

- Μικροελεγκτές (4-bit ή 8-bit) χαμηλού κόστους, γενικής χρήσης και με μικρό αριθμό αποδεκτών, όπως οι μικροελεγκτές των σειρών PIC (Microchip), AVR (Atmel) και 8051 (Intel, Atmel)
- Μικροελεγκτές (8-bit, 16-bit, 32-bit) χαμηλού κόστους, γενικής χρήσης με μέτριο ή μεγάλο αριθμό αποδεκτών, με μεγάλο αριθμό κοινών περιφερειακών και μετατροπείς αναλογικού-σε-ψηφιακό και το αντίστροφο.
- Μικροελεγκτές εξειδικευμένων εφαρμογών, που ενσωματώνουν κάποιο εξειδικευμένο πρωτόκολλο επικοινωνίας το οποίο υλοποιείται στο υλικό. Ο τύπος αυτός μικροελεγκτών χρησιμοποιείται π.χ. σε μόντεμ.
- Μικροελεγκτές (32-bit) μέσου κόστους, γενικής χρήσης, με μεγάλο αριθμό αποδεκτών, με υψηλή αυτάρκεια περιφερειακών, μεγάλες δυνατότητες μνήμης και ταχύτητα εκτέλεσης εντολών.

Οι πιο διαδεδομένες γλώσσες προγραμματισμού των μικροελεγκτών είναι η C, η C++ και η Assembly, με την τελευταία να προτιμάται σε περιπτώσεις λογισμικού που απαιτείται να έχει μικρό μέγεθος και μεγάλη ταχύτητα εκτέλεσης. Οι μικροελεγκτές προγραμματίζονταν αρχικά μόνο στη γλώσσα Assembly, αλλά διάφορες γλώσσες υψηλού επιπέδου, όπως η C, η Python και η JavaScript, χρησιμοποιούνται εξίσου σήμερα σε μικροελεγκτές και ενσωματωμένα συστήματα.

Ορισμένοι μικροελεγκτές διαθέτουν περιβάλλοντα ανάπτυξης εφαρμογών. Διατίθενται επίσης προσομοιωτές, οι οποίοι επιτρέπουν στον προγραμματιστή να αναλύσει ποια θα ήταν η συμπεριφορά του μικροελεγκτή και του προγράμματος του εάν χρησιμοποιούσε την πραγματική πλακέτα. [6]

2.1.3 GPS (Global Positioning System)

Το GPS (Global Positioning System) είναι παγκόσμιο σύστημα εντοπισμού γεωγραφικής θέσης, ακίνητου ή κινούμενου χρήστη, το οποίο βασίζεται σε ένα "πλέγμα" 24 δορυφόρων της Γης, εφοδιασμένων με ειδικές συσκευές εντοπισμού, τους "πομποδέκτες GPS". Οι πομποδέκτες GPS παρέχουν ακριβείς πληροφορίες για τις συντεταγμένες θέσης ενός χρήστη, το υψόμετρό του, την ταχύτητα και την κατεύθυνση της κίνησης του. Επίσης, σε συνδυασμό με λογισμικό χαρτογράφησης, οι πληροφορίες αυτές μπορούν να απεικονιστούν γραφικά.

Το σύστημα GPS ξεκίνησε από το Υπουργείο Άμυνας των ΗΠΑ και ονομάστηκε Navigation Signal Timing and Ranging Global Positioning System (NAVSTAR) GPS. Το δορυφορικό αυτό σύστημα ρυθμίζεται μέχρι και σήμερα από τη Βάση Πολεμικής Αεροπορίας Σρίβερ (Schriever). [8]

Υπάρχουν 5 κύριες χρήσεις του GPS [8]:

1. Τοποθεσία (Προσδιορισμός θέσης)
2. Πλοήγηση (Μετάβαση από την μία τοποθεσία στην άλλη)
3. Παρακολούθηση (Παρακολούθηση αντικειμένου ή προσωπικής κίνησης)
4. Χρονισμός (Δυνατότητα λήψης ακριβών μετρήσεων χρόνου)
5. Χαρτογράφηση (Δημιουργία χαρτών του κόσμου)

Η χρήση του GPS είναι σημαντική σε πολλές εφαρμογές όπως [8]:

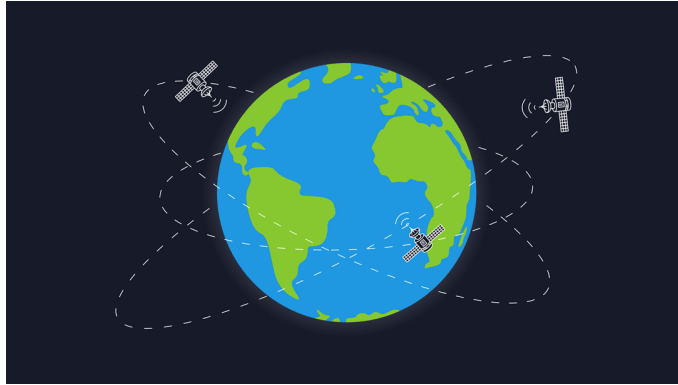
- Σε περίπτωση έκτακτης ανάγκης
- Ψυχαγωγία
- Υγεία και φυσική κατάσταση
- Κατασκευή, εξόρυξη και μεταφορά εκτός δρόμου
- Μεταφορά
- Γεωργία
- Αυτόνομα Οχήματα
- Στρατιωτικές Υπηρεσίες
- Κινητές Επικοινωνίες
- Ασφάλεια
- Αλιεία κ.α.

Η ακρίβεια της συσκευής πομποδέκτη GPS εξαρτάται από πολλές μεταβλητές, όπως τον αριθμό των διαθέσιμων δορυφόρων με τους οποίους επικοινωνεί, τις συνθήκες στην ιονόσφαιρα, όπως σπινθηρισμοί, ηλεκτρομαγνητικά πεδία και ιονίζουσες ακτίνες από τον ήλιο, το αστικό περιβάλλον και άλλα. [8]

Μερικοί παράγοντες που μπορούν να εμποδίσουν την ακρίβεια του GPS περιλαμβάνουν [8]:

- Φυσικά εμπόδια
- Ατμοσφαιρικές επιδράσεις
- Αριθμητικοί λανθασμένοι υπολογισμοί
- Τεχνητή παρέμβαση κ.α.

Η ακρίβεια του υπολογισμού θέσης με GPS είναι υψηλότερη σε ανοιχτούς χώρους χωρίς διπλανά ψηλά κτίρια που μπορεί να δημιουργήσουν παρεμβολές. [8]



Εικόνα 3 Απεικόνιση GPS [8]

2.1.4 Ηλεκτρονική πυξίδα

Οι μικρές πυξίδες που βρίσκονται σε ρολόγια, κινητά τηλέφωνα και άλλες ηλεκτρονικές συσκευές είναι πυξίδες μικροηλεκτρομηχανικών συστημάτων στερεάς κατάστασης, κατασκευασμένες επί το πλείστον από δύο ή τρεις αισθητήρες μαγνητικού πεδίου που παρέχουν δεδομένα για έναν επεξεργαστή. Συνήθως η συσκευή παράγει είτε αναλογικό σήμα είτε ψηφιακό το οποίο είναι ανάλογο με τον προσανατολισμό. Το σήμα αυτό, ερμηνεύεται από έναν μικροεπεξεργαστή ή έναν ελεγκτή. [9]

2.1.5 Wi-Fi

Το Wi-Fi είναι μια οικογένεια τεχνολογιών ασύρματης δικτύωσης, βασισμένη στο πρότυπο IEEE 802.11, που χρησιμοποιείται ευρέως για την δικτύωση συσκευών σε μικρή απόσταση . Οι συσκευές που χρησιμοποιούν τεχνολογίες Wi-Fi είναι οι επιτραπέζιοι και φορητοί υπολογιστές, τα σύγχρονα κινητά τηλέφωνα και tablets, οι έξυπνες τηλεοράσεις, οι εκτυπωτές, οι ψηφιακές κάμερες, τα αυτοκίνητα, τα drones και γενικώς κάθε συσκευή που διαθέτει Wireless Network Interface Card (NIC), δηλαδή ένας ελεγκτής διασύνδεσης δικτύου που συνδέεται σε ασύρματο δίκτυο [10]

Η κάλυψη των συσκευών που είναι συνδεδεμένες στο διαδίκτυο εντός εμβέλειας του Wi-Fi, μπορεί να επεκταθεί από μία πολύ μικρή περιοχή λίγων τετραγωνικών μέτρων έως και πολλά τετραγωνικά χιλιόμετρα αλλά ενδέχεται σε αυτή τη περίπτωση να απαιτεί μια ομάδα σημείων πρόσβασης με επικαλυπτόμενη κάλυψη. Το Wi-Fi είναι εμπορικό σήμα της Συμμαχίας Wi-Fi, η οποία παρέχει πιστοποίηση (Wi-Fi Certified) σε όσα προϊόντα έχουν ολοκληρώσει επιτυχώς τις απαραίτητες δοκιμές, όπως συμβατότητα με το πρότυπο, αλληλεπίδραση με άλλες συσκευές, χαρακτηριστικά που σχετίζονται με την ασφάλεια κ.α. [11]



Εικόνα 4 Wi-Fi Alliance [11]

2.1.6 Τηλεκατευθυνόμενο Wi-Fi όχημα

Τηλεκατευθυνόμενο, λέγεται ένα όχημα το οποίο μέσω μιας συσκευής (χειριστήριο, κινητό τηλέφωνο), μπορεί να δεχτεί οδηγίες και να κινηθεί προς μία συγκεκριμένη κατεύθυνση ή σε έναν συγκεκριμένο προορισμό. Τα πρώτα τηλεκατευθυνόμενα ήταν ενσύρματα συνδεδεμένα με τη συσκευή χειριστηρίου. Αργότερα κατασκευάστηκαν τα ασύρματα τηλεκατευθυνόμενα τα οποία ήταν RC (Radio Controlled), δηλαδή ραδιοελεγχόμενα και ελέγχονταν από απόσταση με χρήση εξειδικευμένου πομπού ή τηλεχειριστηρίου. Ένας διαδεδομένος τρόπος επικοινωνίας μεταξύ οχήματος και τηλεχειριστηρίου είναι μέσω σύνδεσης Wi-Fi όπου το χειριστήριο μπορεί να είναι και ένα κινητό τηλέφωνο.

Ένα όχημα το οποίο έχει πάνω του έναν μικροελεγκτή NodeMCU (Κεφ. 3.1.9) μπορεί να συνδεθεί σε ένα ασύρματο δίκτυο Wi-Fi και μέσω μιας εφαρμογής να γίνει η κατάλληλη επικοινωνία από τον χρήστη και να μετατραπεί σε τηλεκατευθυνόμενο όχημα.

2.1.7 Βιβλιοθήκες

Στην πληροφορική βιβλιοθήκη ονομάζεται μια συλλογή από έτοιμα υποπρογράμματα που χρησιμοποιείται για την ανάπτυξη κώδικα. Η χρήση των βιβλιοθηκών αναπτύχθηκε παράλληλα και για τις ανάγκες του δομημένου προγραμματισμού.

Οι περισσότερες βιβλιοθήκες δεν είναι εκτελέσιμες. Ωστόσο κάποια εκτελέσιμα αρχεία είναι και προγράμματα και βιβλιοθήκες ταυτόχρονα. Τα σύγχρονα λειτουργικά συστήματα χρησιμοποιούν βιβλιοθήκες για την υλοποίηση των υπηρεσιών του συστήματος και έτσι προγραμματιστικός κώδικας που χρησιμοποιείτε από διαφορετικές εφαρμογές παρέχεται από τις βιβλιοθήκες αυτές και δεν χρειάζεται να γραφεί εκ νέου για κάθε νέο πρόγραμμα. [3]

2.1.8 Πλατφόρμες για δημιουργία εφαρμογών

Υπάρχουν πολλές διαθέσιμες πλατφόρμες για τη δημιουργία εφαρμογών για έξυπνο κινητό, το οποίο επικοινωνεί μέσω Wi-fi με ένα Arduino. Κάποιες από αυτές είναι το Android Studio, το Andromo και το Blynk.

Το Android Studio είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον για την ανάπτυξη εφαρμογών αποκλειστικά για το λειτουργικό σύστημα Android. Το Android Studio διατέθηκε για χρήση για πρώτη φορά στις 16 Μαΐου 2013 από την εταιρία Google. Παρέχει εργαλεία, επιλογές ενσωμάτωσης του GitHub και υπογραφής εφαρμογών ProGuard και οδηγούς βασισμένους σε πρότυπα, για τη δημιουργία κοινών σχεδίων και στοιχείων Android καθώς και μία πλούσια διεπαφή επεξεργασίας διάταξης. Το Android Studio παρέχει εικονική συσκευή Android για εξομίωση της λειτουργίας της εφαρμογής και εντοπισμό σφαλμάτων και δεν παρέχει δυνατότητα απενεργοποίησης της λειτουργίας αυτόματης αποθήκευσης. [12]

Το Andromo είναι ένας διαδικτυακός κατασκευαστής εφαρμογών Android για τη δημιουργία εφαρμογών για κινητά και συσκευές Android με αυτόματη συγγραφή κώδικα για τον χρήστη. Σύμφωνα με τους κατασκευαστές του, είναι ένα απλοποιημένο εργαλείο που μπορεί να χρησιμοποιήσει κάθε χρήστης, ακόμα και αυτός που δεν κατέχει τεχνικές δεξιότητες. Υποστηρίζει ανάπτυξη παιχνιδιών, ιστοσελίδες, Podcast, δημιουργία καναλιού στο YouTube και ενσωμάτωση ροής στο Facebook. Έχει σχεδιαστεί κυρίως για τεχνικούς και επιχειρηματίες. Έχει δωρεάν έκδοση και πληρωτέα. Η δωρεάν έκδοση επιτρέπει στους χρήστες να δημιουργούν εφαρμογές πλήρους λειτουργίας, αλλά χωρίς δημιουργία εσόδων [28].

Για την εκπόνηση της εργασίας αυτής χρησιμοποιήθηκε η πλατφόρμα Blynk. Η πλατφόρμα Blynk επιλέχθηκε λόγω της ποικιλίας γραφικών στοιχείων που παρέχει. [12] [13]

2.1.9 Blynk

Η πλατφόρμα Blynk σχεδιάστηκε για το Internet of Things (IoT). Παρέχει δυνατότητες όπως ο έλεγχος του υλικού από απόσταση, η αποθήκευση δεδομένων, καθώς και η εμφάνιση και επεξεργασία δεδομένων αισθητήρων.

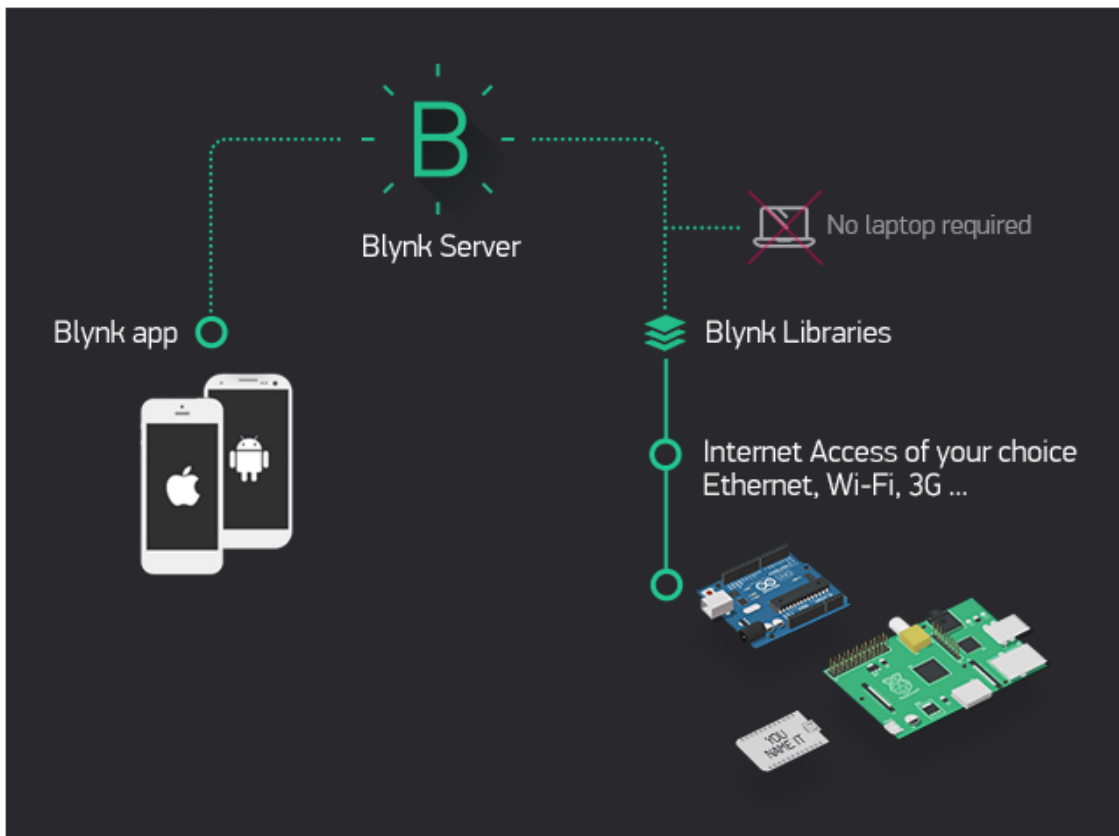
Η πλατφόρμα διαθέτει τέσσερα βασικά στοιχεία [14]:

- Blynk App, που επιτρέπει την δημιουργία διεπαφών με την χρήση διάφορων widgets δηλαδή γραφικών στοιχείων διεπαφής.
- Blynk Libraries, που επιτρέπουν την επικοινωνία με τον διακομιστή της πλατφόρμας Blynk
- Blynk Server, που χειρίζεται τις επικοινωνίες μεταξύ υλικού και κινητού
- Blynk cloud, που είναι η εναλλακτική του εξυπηρετητή Blynk Server, στο υπολογιστικό νέφος

Χαρακτηριστικά της Blynk [14]:

- Παρόμοια API(Application Programming Interface) και UI(User Interface) για τις συσκευές και το υλικό
- Σύνδεση στο Cloud με α) Wi-Fi, β) Bluetooth και γ) BLE(Bluetooth Low Energy), Ethernet, USB(Universal Serial Bus), GSM(Global System for Mobile Communications)
- Εύχρηστα γραφικά στοιχεία (Widgets)
- Χειρισμός εικονικών pin
- Εύκολη προσθήκη νέων λειτουργιών
- Παρακολούθηση δεδομένων ιστορικού
- Επικοινωνία συσκευής με συσκευή μέσω του γραφικού στοιχείου bridge
- Αποστολή μηνυμάτων, ειδοποιήσεις

Η πλατφόρμα Blynk μπορεί να χειριστεί υλικό όπως το Arduino αλλά θα χρειαστεί Ethernet ή Wi-Fi Shield για να επικοινωνήσει με τον χρήστη. [14]



Εικόνα 5 Τα στοιχεία της πλατφόρμας Blynk [14]

2.1.10 Internet of Things (IoT)

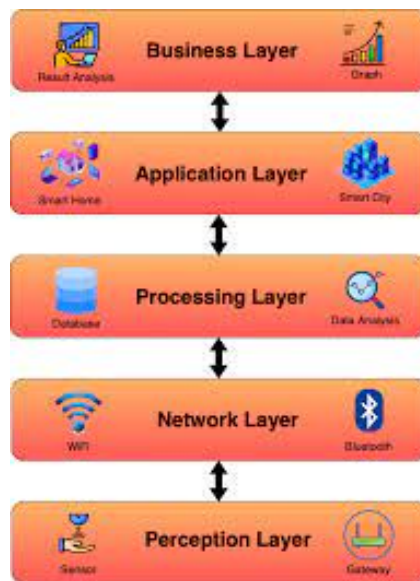
Το Internet of Things (Διαδίκτυο των Πραγμάτων) είναι ένα δίκτυο επικοινωνίας πολλών συσκευών είτε είναι αισθητήρες είτε οικιακές συσκευές είτε άλλες συσκευές. Ο έλεγχος τους γίνεται από έναν υπολογιστή ή ένα κινητό κυρίως. [15]

Οι συσκευές αυτές συνδέονται σε μια πλατφόρμα η οποία συγκεντρώνει δεδομένα από αυτές τις συσκευές τα οποία στη συνέχεια επεξεργάζονται και αξιοποιούνται σε διεργασίες λήψης αποφάσεων. [15]

Παρ' ότι βρίσκεται συνέχεια σε κατάσταση εξέλιξης και βελτίωσης, ένα μεγάλο πρόβλημα είναι η ασφάλεια των δεδομένων που αποθηκεύονται. Για να διασφαλισθεί η προστασία των πολιτών έχουν ανακοινωθεί σχεδόν από όλες τις χώρες νόμοι για την προστασία της ιδιωτικότητας.

Τα πέντε επίπεδα του Διαδικτύου των Πραγμάτων είναι [15]:

1. επίπεδο αντίληψης,
2. μεταφοράς,
3. επεξεργασίας,
4. εφαρμογής και
5. επιχειρήσεων.



Εικόνα 6 Επίπεδα IoT

Τα οφέλη του IoT για τις επιχειρήσεις σχετίζονται με την εκάστοτε εφαρμογή, όπως ευκολία πρόσβασης, καλύτερο αυτοματισμό και καλύτερο έλεγχο. Στόχος είναι οι επιχειρήσεις να έχουν πρόσβαση σε περισσότερα δεδομένα για τα δικά τους προϊόντα και τα δικά τους εσωτερικά συστήματα, και δυνατότητα για αλλαγές και προσαρμογές, ανάλογα με τα δεδομένα αυτά.

Το IoT υπόσχεται να κάνει το περιβάλλον μας, τα σπίτια, τα γραφεία, ακόμα και τα οχήματά μας, πιο «έξυπνα», πιο «μετρήσιμα» και πιο «επικοινωνιακά». Για παράδειγμα, τα συστήματα ασφαλείας στο σπίτι διευκολύνουν την παρακολούθηση για το τι συμβαίνει μέσα και έξω από αυτό. Οι αισθητήρες μπορούν να βοηθήσουν στην μέτρηση θορύβου ή μόλυνσης για το περιβάλλον, αλλά και να επιτρέψουν την απομακρυσμένη διαχείριση και τον έλεγχο των έξυπνων συσκευών ενός σπιτιού και τους εξοπλισμού του. Αυτόνομα αυτοκίνητα και «έξυπνες» πόλεις θα μπορούσαν να αλλάξουν τον τρόπο με τον οποίο κατασκευάζονται και διαχειρίζονται οι δημόσιοι χώροι. [15]



Εικόνα 7 Διαδίκτυο των πραγμάτων. Σύνδεση από παντού. [16]

ΚΕΦΑΛΑΙΟ 3ο

3.1 ΕΞΟΠΛΙΣΜΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ (ΕΡΓΑΛΕΙΑ)

3.1.1 Γλώσσα προγραμματισμού C++

Η C++ είναι μία γενικού σκοπού γλώσσα προγραμματισμού. Θεωρείται μέσου επιπέδου γλώσσα γιατί συνδυάζει γλώσσες χαμηλού και υψηλού επιπέδου. Υποστηρίζει δομημένο, αντικειμενοστραφή και γενικό προγραμματισμό. Είναι μία μεταγλωττιζόμενη γλώσσα πολλαπλών παραδειγμάτων με τύπους. [3]

Αναπτύχθηκε από τον Μπιάρνε Στρούστρουπ το 1979, ως βελτίωση της γλώσσας προγραμματισμού C που προϋπήρχε. Η C++ ορίστηκε παγκοσμίως με το πρότυπο ISO/IEC 14882:1998 το 1998. [3]

Στο βιβλίο *The Design and Evolution of C++* (1994), ο δημιουργός της γλώσσας εξηγεί πως η γλώσσα είναι σχεδιασμένη [17]:

1. ως μία γενικής χρήσης γλώσσας με στατικούς τύπους, που είναι όσο αποτελεσματική είναι και η C,
2. να υποστηρίζει άμεσα και σφαιρικά πολλά είδη προγραμματισμού,
3. να δίνει επιλογές στον προγραμματιστή και να είναι όσο το δυνατόν συμβατή με την C για να γίνει εύκολη η μετάβαση για το χρήστη και για τις προϋπάρχουσες εφαρμογές,
4. να λειτουργεί χωρίς κάποιο εξειδικευμένο προγραμματιστικό περιβάλλον.

Ο λόγος αναφοράς στην C++ είναι ότι το Arduino, που είναι ένας μικροελεγκτής μονής πλακέτας, μπορεί να προγραμματιστεί με τη γλώσσα Wiring, δηλαδή μία παραλλαγή της γλώσσας C++, αξιοποιώντας και ένα σύνολο από βιβλιοθήκες οι οποίες και αυτές είναι υλοποιημένες στην γλώσσα C++. [3]

3.1.2 Arduino UNO R3

Το Arduino UNO είναι μια ανοιχτού κώδικα πλακέτα με μικροελεγκτή υλοποιημένο με τον μικροελεγκτή Microchip AT mega328P [1]. Η πλακέτα ενσωματώνει ένα σύνολο αναλογικών και ψηφιακών ακροδεκτών εισόδου/εξόδου, οι οποίοι μπορούν να χρησιμοποιηθούν για επέκταση του Arduino UNO, μέσω της σύνδεσής τους με άλλες πλακέτες ή άλλα κυκλώματα. Μπορεί να τροφοδοτηθεί είτε μέσω της USB θύρας από άλλο κύκλωμα ή υπολογιστή, είτε αυτόνομα μέσω ηλεκτρικού συσσωρευτή (μπαταρία). [1]

Η λέξη UNO από τα Ιταλικά «έναν» δόθηκε ώστε να σηματοδοτεί την αρχική έκδοση του λογισμικού της Arduino καθώς είναι το πρώτο από την σειρά των USB Arduino. [1]

Πίνακας 3.1 Χαρακτηριστικά Μικροελεγκτή [1]

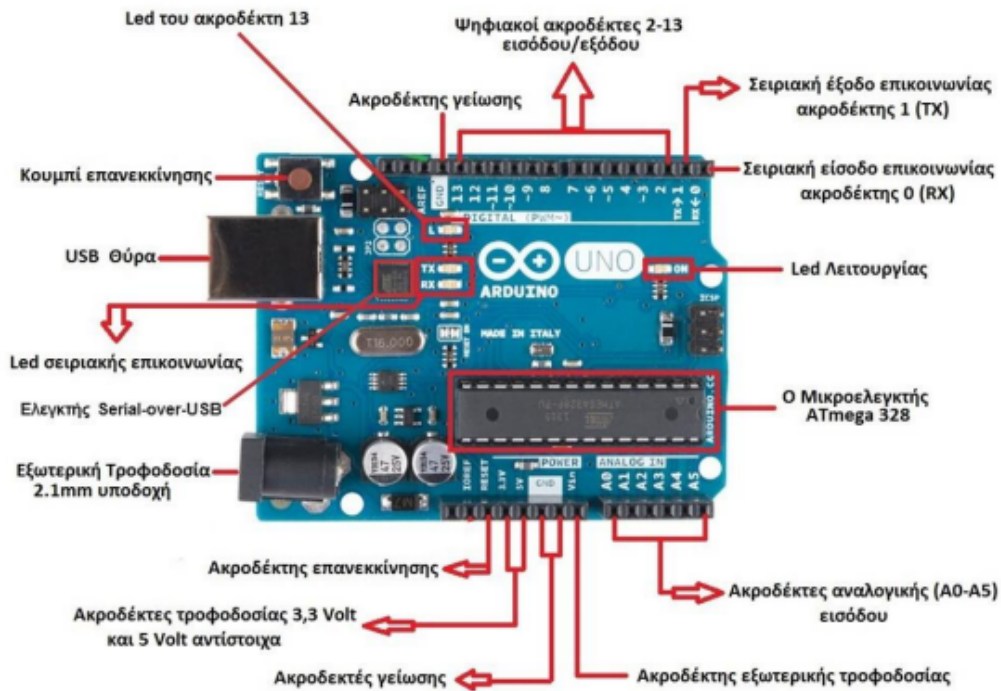
ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ
Μικροελεγκτής : Microchip AT mega328P
Τάση Λειτουργίας: 5 Volts
Τάση Εισόδου: 7 to 20 Volts
Digital I/O Pins: 14 (από τα οποία τα 6 παρέχουν και PWM έξοδο)
UART: 1
I2C: 1
SPPI: 1
Αναλογικά Pins Εισόδου: 6
DC Ρεύμα ανά I/O Pin: 20 mA
DC Ρεύμα ανά 3.3V Pin: 50 mA
Flash Memory: 32 KB από τα οποία τα 0.5 KB χρησιμοποιούνται από τον bootloader
SRAM: 2 KB
EEPROM: 1 KB
Ταχύτητα Clock: 16 MHz
Μήκος: 68.6 mm
Φάρδος: 53.4 mm
Βάρος: 25 g



Εικόνα 8 Arduino UNO R3 [1]

Γενικές λειτουργίες των ακροδεκτών του Arduino UNO [1],[2],[6]:

- LED: Υπάρχει ένα ενσωματωμένο LED που οδηγείται από τον ψηφιακό ακροδέκτη 13. Όταν ο ακροδέκτης έχει υψηλή τιμή, το LED είναι αναμμένο, ενώ όταν ο ακροδέκτης είναι σε χαμηλή τάση, είναι σβηστός.
- VIN: Η τάση εισόδου στην πλακέτα Arduino όταν χρησιμοποιεί εξωτερική πηγή τροφοδοσίας (σε αντίθεση με τα 5 Volt από τη σύνδεση USB ή άλλη ρυθμιζόμενη πηγή ισχύος).
- 5V: Αυτός ο ακροδέκτης εξάγει σταθεροποιημένη 5V DC τάση. Η πλακέτα μπορεί να τροφοδοτηθεί είτε από την πρίζα DC (7 - 20V), την υποδοχή USB (5V) είτε από τον ακροδέκτη VIN της πλακέτας (7-20V). Η παροχή τάσης μέσω των ακίδων 5V ή 3.3V παρακάμπτει τον σταθεροποιητή και μπορεί να προκαλέσει ζημιά στην πλακέτα.
- 3V3: Παρέχεται τροφοδοσία 3,3 Volt από τον ενσωματωμένο σταθεροποιητή. Το μέγιστο ρεύμα που παρέχεται είναι 50 mA.
- GND: Ακροδέκτες γείωσης.
- IOREF: Αυτός ο ακροδέκτης στην πλακέτα Arduino παρέχει την αναφορά τάσης με την οποία λειτουργεί ο μικροελεγκτής. Μια σωστά διαμορφωμένη επέκταση (shield) μπορεί να διαβάσει την τάση του ακροδέκτη IOREF και να επιλέξει την κατάλληλη πηγή ισχύος ή να επιτρέψει στους μεταφραστές τάσης στις εξόδους να λειτουργούν με τα 5V ή 3.3V.
- RESET: Συνήθως χρησιμοποιείται για την προσθήκη κουμπιού επαναφοράς σε Arduino Shields που μπλοκάρουν αυτό που υπάρχει στην πλακέτα.
- Αναλογικοί: Υπάρχουν 6 pins αριθμημένα από το 0 έως το 5 με την σήμανση ANALOG IN τα οποία λειτουργούν ως αναλογικές εισοδοί κάνοντας χρήση του ADC που είναι ενσωματωμένος στον μικροελεγκτή ATmega. Αυτά τα 6 pins μπορούν επίσης να μετατραπούν σε ψηφιακά pins εισόδου/εξόδου αλλά τότε μετονομάζονται σε pins 14-19
- Ψηφιακοί: Υπάρχουν 14 ψηφιακοί ακροδέκτες που οι 2 είναι ακροδέκτες σειριακής επικοινωνίας αριθμημένοι από το 0 έως το 1, αλλά δεν μπορούν να χρησιμοποιηθούν ως ψηφιακές εισοδοί/έξοδοι. Οι υπόλοιποι 12 είναι ακροδέκτες εισόδου/εξόδου, αριθμημένοι από το 2 έως το 13 και λειτουργούν στα 5V.



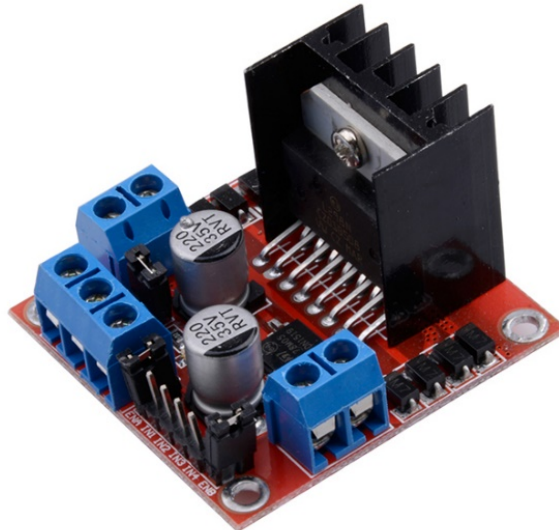
Εικόνα 9 Ακροδέκτες (Pins) Arduino UNO [2]

3.1.3 L298n Οδηγός DC motor για Arduino

Το L298N Motor Driver Module είναι ένας υψηλής ισχύος οδηγός για συνεχούς τάσης(DC) και Βηματικά Μοτέρ. Αυτή η μονάδα αποτελείται από ένα ολοκληρωμένο κύκλωμα L298 και έναν ρυθμιστής τάσης 78M05 στα 5V. Η μονάδα L298N μπορεί να ελέγξει έως και 4 κινητήρες DC ή 2 DC κινητήρες με έλεγχο κατεύθυνσης και ταχύτητας. [18] [6]

Πίνακας 3.2 L298N Διαμόρφωση των ακροδεκτών[9]:

Όνομα	Περιγραφή
IN1 & IN2	Ακροδέκτες εισόδου του κινητήρα A. Χρησιμοποιείται για τον έλεγχο της κατεύθυνσης περιστροφής του κινητήρα A
IN3 & IN4	Ακροδέκτες εισόδου του κινητήρα B. Χρησιμοποιείται για τον έλεγχο της κατεύθυνσης περιστροφής του κινητήρα B
ENA	Ενεργοποιεί το σήμα PWM για τον κινητήρα A
ENB	Ενεργοποιεί το σήμα PWM για τον κινητήρα B
OUT1 & OUT2	Ακροδέκτες εξόδου του κινητήρα A
OUT3 & OUT4	Ακροδέκτες εξόδου του κινητήρα B
12V	Είσοδος 12V από πηγή ισχύος DC
5V	Παρέχει ισχύ για το κύκλωμα λογικής εναλλαγής μέσα στο L298N IC
GND	Ακροδέκτης γείωσης



Εικόνα 10 L298n DC motor [18]

L298 Χαρακτηριστικά & Προδιαγραφές: [18]

- Μοντέλο: L298N 2A
- Chip: Double H Bridge L298N
- Μέγιστη παροχή Ισχύος στα Μοτέρ: 46V
- Μέγιστη παροχή Ρεύματος στα Μοτέρ: 2A
- Λογική Τάση: 5V
- Τάση Τροφοδοσίας: 5-35V
- Ρεύμα Τροφοδοσίας: 2A
- Ρεύμα Λογικού Ακροδέκτη (5V): 0-36mA
- Μέγιστη Ισχύς (W): 25W
- Έλεγχος Ρεύματος για κάθε ρότορα
- Ψύκτρα για καλύτερη απόδοση
- Ένδειξη LED ενεργοποίησης

3.1.4 DC Motor-Ηλεκτροκινητήρας

DC κινητήρα ονομάζουμε μια κατηγορία των ηλεκτρικών μηχανών που μετατρέπει την συνεχή ηλεκτρική ενέργεια (DC-Direct Current) σε μηχανική ενέργεια. Οι πιο κοινοί τύποι DC κινητήρα λειτουργούν με τις δυνάμεις που παράγονται από τα μαγνητικά πεδία. Όλοι σχεδόν οι κλασικοί κινητήρες συνεχούς ρεύματος έχουν κάποιο εσωτερικό μηχανισμό, είτε ηλεκτρομηχανικό ή ηλεκτρονικό, που αλλάζει περιοδικά την κατεύθυνση της ροής του ρεύματος σε ένα μέρος του κινητήρα.

Ο DC κινητήρας ήταν ο πρώτος τύπος που χρησιμοποιήθηκε ευρέως, μιας και μπορούσε να τροφοδοτείται από τα υπάρχοντα συστήματα διανομής ισχύος ρεύματος. Η ταχύτητα ενός κινητήρα DC ελέγχεται σε ένα μεγάλο εύρος στροφών, χρησιμοποιώντας είτε μια μεταβλητή τάση τροφοδοσίας ή την αλλαγή της δύναμης του ρεύματος στις περιελίξεις του. [18]

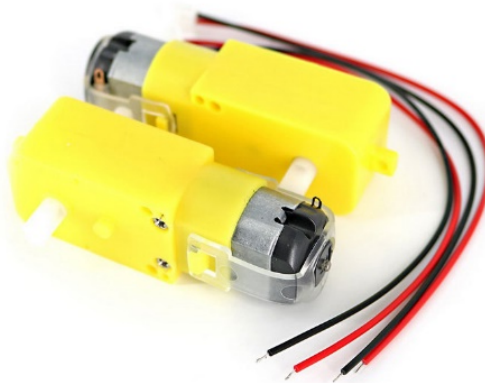
Μικροί κινητήρες συνεχούς ρεύματος χρησιμοποιούνται σε εργαλεία, παιχνίδια και συσκευές. Μεγάλοι κινητήρες DC χρησιμοποιούνται στην κίνηση ηλεκτρικών οχημάτων, ανελκυστήρων κ.α. [18]

Ο Brushless κινητήρας DC χρησιμοποιεί έναν ή περισσότερους μόνιμους μαγνήτες στον άξονα του (ρότορας) και ηλεκτρομαγνήτες στο περίβλημα του (στάτορας). Με την χρήση ενός ελεγκτή κινητήρα μετατρέπεται το DC σε AC. Έτσι με αυτόν τον σχεδιασμό είναι μηχανικά απλούστερο από κάποιον brushed κινητήρα διότι εξαλείφει την επιπλοκή για την περιστροφή του ρότορα. [18] [6]

Τα χαρακτηριστικά των DC Motors που χρησιμοποιούμε είναι [19]:

Πίνακας 3.3 Χαρακτηριστικά DC Motor

Τάση Λειτουργίας	DC 3V	έως	DC 6V
Ρεύμα	100 MA	έως	120MA
Μειωτής στροφών	48:1		
RPM (Στροφές/λεπτό)	100	έως	240
Βάρος	50γρ		
Μέγεθος	70mm*22mm*18mm		
Θόρυβος	<65dB		



Εικόνα 11 DC Κινητήρας

3.1.5 Σερβοκινητήρας

Οι σερβοκινητήρες είναι ειδικά σχεδιασμένοι για εφαρμογές ελέγχου και ρομποτικής. Είναι κινητήρες που μπορούν να γυρίζουν με ακρίβεια έναν άξονα από 0 έως 180 μοίρες. Χρησιμοποιούνται για ακριβή ρύθμιση ταχύτητας σε υψηλές ροπές. Οι σερβοκινητήρες αποτελούνται από έναν κινητήρα, αισθητήρα θέσης, μηχανικό μειωτήρα και εξελιγμένο ελεγκτή. Έχει τρία καλώδια, έναν κόκκινο στην πηγή (+5V), ένα καφέ ή μαύρο στη γείωση (GND) και ένα άσπρο ή κίτρινο ελέγχου που συνδέεται στον ακροδέκτη του Arduino. Έχουν υψηλές δυνατότητες ροπής. [20]

Για τον ευκολότερο έλεγχο του σερβοκινητήρα μέσω Arduino, μπορούμε να εισάγουμε την βιβλιοθήκη 'Servo.h' στο πρόγραμμα μας `#include (<Servo.h>)` και να ορίσουμε μια μεταβλητή (`servo myservo;`).

Ο σερβοκινητήρας στρέφεται 90 μοίρες προς οποιαδήποτε κατεύθυνση, οπότε η μέγιστη κίνηση του είναι 180 μοίρες. Όταν δοθεί εντολή να μετακινηθεί, ο άξονας μετακινείται και διατηρεί την απαιτούμενη θέση.

Οι σερβοκινητήρες χρησιμοποιούνται στη ρομποτική, υπολογιστές, συσκευές αναπαραγωγής CD/DVD, παιχνίδια κ.α. Τα βασικά χαρακτηριστικά κάθε σερβοκινητήρα συνεχούς ή εναλλασσόμενου ρεύματος είναι [20]:

- Η ροπή του κινητήρα που είναι ανάλογη της εφαρμοζόμενης τάσης ελέγχου, όπου αναπτύσσεται από τον ενισχυτή λόγω του σφάλματος στην είσοδο του.
- Η φορά της ροπής που καθορίζεται από την πολικότητα τάσης ελέγχου.



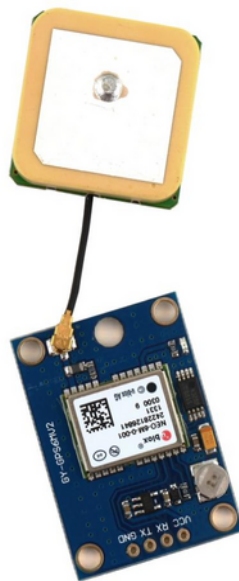
Εικόνα 12 Σερβοκινητήρας SG90 (9G)

3.1.6 NEO 6M GPS

Το εξάρτημα NEO 6M GPS είναι ένας πλήρης δέκτης GPS χαμηλού κόστους, συμβατός με το Arduino UNO. Έχει καλή απόδοση και ενσωματώνει κεραμική κεραία (25x25x4mm), η οποία παρέχει ισχυρή δυνατότητα δορυφορικής αναζήτησης. Με τους δέκτες ισχύος και σήματος μπορεί κάποιος να αντλεί δεδομένα για την κατάσταση της μονάδας. Τα πλεονεκτήματα αυτού του εξαρτήματος είναι η χαμηλή κατανάλωση σε ρεύμα, η μικρή καθυστέρηση για ανίχνευση σήματος, η ευκολία στον προγραμματισμό και τέλος το μικρό του μέγεθος. Χάρη στην εφεδρική μπαταρία δεδομένων, η μονάδα μπορεί να αποθηκεύσει τα δεδομένα όταν απενεργοποιηθεί τυχαία η κύρια τροφοδοσία. [21] Μπορεί να εφαρμοσθεί σε ένα έξυπνο ρομποτικό αυτοκίνητο για αυτόματη επιστροφή ή για κατεύθυνση σε έναν συγκεκριμένο προορισμό, καθιστώντας το έτσι ένα πραγματικά «έξυπνο» όχημα.

NEO 6M GPS MODULE FOR ARDUINO Χαρακτηριστικά [21]:

- Ψυχρός χρόνος εκκίνησης: 38s
- Θερμός χρόνος έναρξης: 1s
- Χρόνος ανασύλληψης σήματος: 0,1 s
- Ευρεία τάση τροφοδοσίας: 3,3 V έως 5 V
- Διεπαφή: UART (Universal Asynchronous Receiver / Transmitter)
- Προεπιλογή Baud rate: 9600bps
- Ενσωματωμένη κεραμική κεραία
- Μικρό μέγεθος: 31 x 24 mm
- Βιβλιοθήκες Arduino: TinyGPS, TinyGPS++



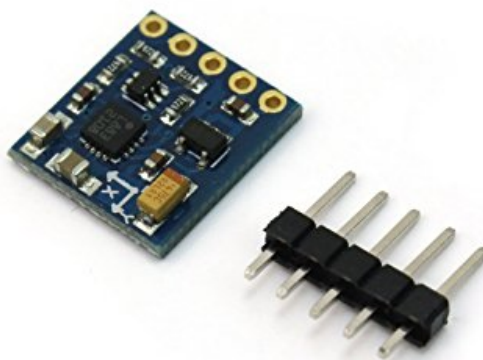
Εικόνα 13 GPS Module for Arduino [21]

3.1.7 Πυξίδα QMC5883L

Το QMC5883L είναι μια πυξίδα και μαγνητόμετρο τριπλού άξονα, η οποία χρησιμοποιεί δίαυλο σειριακής επικοινωνίας I²C (Inter-Integrated Circuit). Είναι ένα εξάρτημα επιφάνειας πολλαπλών ολοκληρωμένων, σχεδιασμένη για μαγνητική ανίχνευση χαμηλού πεδίου με ψηφιακή διεπαφή για εφαρμογές όπως πυξίδα χαμηλού κόστους και μαγνητομετρία. Αυτή η μονάδα αποτελείται από έναν ενσωματωμένο ισοσταθμιστή τάσης που σας επιτρέπει να τροφοδοτείτε τη μονάδα με τάσεις μεταξύ 3.3V και 6V. Το QMC5883L περιλαμβάνει έναν υψηλής ανάλυσης και ανθεκτικό σε μαγνητικά κύματα αισθητήρα HMC118X και έναν ενισχυτή ASIC (Application-Specific Integrated Circuit) που παρέχει ενίσχυση. Ακόμη, περιέχει ένα σύστημα αντιστάθμισης (offset) και ένα αναλογικό σε ψηφιακό μετατροπέα 12 bit που επιτρέπει ακρίβεια πορείας με 1 έως 2 μοίρες απόκλιση. [22]

Πίνακας 3.4 QMC5883L Διαμόρφωση PIN [22]

Όνομα	Περιγραφή
VCC	Το Pin Vcc τροφοδοτεί τη μονάδα, συνήθως με + 5V
GND	Γείωση τροφοδοσίας
SCL	Serial Clock Pin. Συνδέστε αυτόν τον ακροδέκτη στον ακροδέκτη SCL του μικροελεγκτή.
SDA	Serial Data Pin. Συνδέστε αυτόν τον ακροδέκτη στον ακροδέκτη SDA του μικροελεγκτή.
DRDY	Έξοδος Pin για δεδομένα. Αυτός ο ακροδέκτης χρησιμοποιείται για να γνωρίζουμε πότε τα δεδομένα είναι έτοιμα για ανάγνωση



Εικόνα 14 Πυξίδα QMC5883L

Χαρακτηριστικά & προδιαγραφές μονάδας μαγνητόμετρου [22]:

- Τάση λειτουργίας: 3V έως 6V DC
- Διεπαφή I²C
- Ακρίβεια 1-2 μοιρών
- Ενσωματωμένο ADC(Αναλογικό σε ψηφιακό μετατροπέα) 12-bit
- Μέγιστος ρυθμός δεδομένων 160Hz
- Εύρος από -8 έως +8 Gauss
- Δεν χρειάζεται εξωτερικά εξαρτήματα
- Εύκολο στη χρήση με μικροελεγκτές ή ακόμα και με κανονικό ψηφιακό / αναλογικό IC
- Μικρό, φθινό και εύκολα διαθέσιμο

3.1.8 ESP 8266

Το ESP8266 είναι ένα Wi-fi μικροσιπ με δυνατότητα μικροελεγκτή και με υποστήριξη TCP/IP. Παράγεται από την Espressif Systems στην Κίνα. Επιτρέπει στους μικροελεγκτές να συνδεθούν σε δίκτυο Wi-fi 2,4GHz και να κάνουν TCP/IP συνδέσεις. Μπορεί να χρησιμοποιηθεί με ESP-AT εντολές και να δώσει πρόσβαση Wi-Fi σε εξωτερικούς κεντρικούς υπολογιστές MCU(MicroController Unit) ή ως αυτοδύναμο MCU. [23]

Τα καλύτερα ESP8266 Modules για IoT projects είναι [23]:

- NodeMCU Dev Kit
- Adafruit Feather HIUZZAH
- KNEWRON smartWIFI
- WeMo's D1 mini
- Wio Link
- Spark Fun Thing

Ο πομποδέκτης Wi-Fi λειτουργεί σε εύρος συχνοτήτων 2400 έως 2484 MHz στο πρότυπο IEEE 802.11 b/g/n. Η ενέργεια που καταναλώνει είναι από 15μΑ έως 400mA ανάλογα την περίπτωση. Διαθέτει 17 GPIO pins και λειτουργεί στα 3.3 V DC. [23]



Εικόνα 15 ESP-12F

3.1.9 NodeMCU

Το NodeMCU δημιουργήθηκε το 2015, δύο χρόνια μετά το ESP8266. Είναι μία ανοικτού κώδικα πλατφόρμα και ενσωματώνει το ESP8266, ειδικότερα το ESP-12 και αργότερα άρχισε να υποστηρίζει και ESP32 32-bit MCU. Το όνομα συνδυάζει κόμβο (node) και μονάδα μικροελεγκτή (MCU). Χρησιμοποιεί τη γλώσσα Lua, όπως το ESP8266. Επειδή έχει περιορισμένο χώρο στους πόρους που διαθέτει θα πρέπει οι χρήστες να το χειριστούν με βάση τις ανάγκες του project που θέλουν να δημιουργήσουν. [23] [24]

Όσον αφορά το υλικό μέρος, είναι μια πλακέτα κυκλώματος όπου έχει έναν USB ελεγκτή και το MCU που είναι τοποθετημένο σε μια μικρότερου μεγέθους πλακέτα και την κεραία. [23] [24]

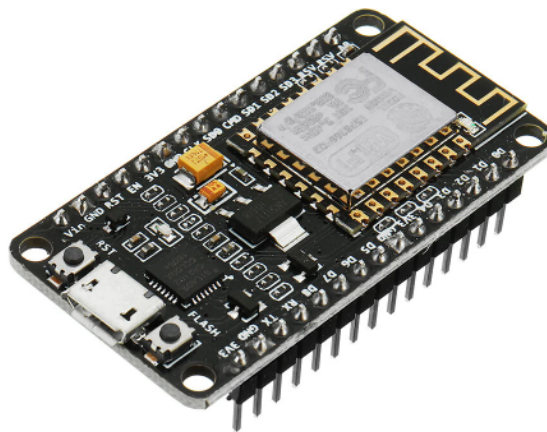
Έχει δύο εκδόσεις την NodeMCU 0.9 και 1.0. η έκδοση 1.0 περιέχει ESP-12E ενώ η 0.9 ESP-12. [24]

Η διαμόρφωση των pinout του NodeMCU είναι [23] [24]:

1. Τροφοδοσία
 - Micro-USB
 - 3.3 V
 - GND
 - Vin
2. Ακροδέκτες Ελέγχου
 - EN
 - RST
3. Αναλογικοί Ακροδέκτες
 - A0 (0-3.3V)
4. SPI Ακροδέκτες
 - SD1
 - SD0
 - CMD
 - CLK
5. GPIO Ακροδέκτες
 - GPIO1 – GPIO16
6. UART Ακροδέκτες
 - TXD0
 - RXD0
 - TXD2
 - RXD2
7. I²C Ακροδέκτες

Οι προδιαγραφές του NodeMCU ESP8266 είναι [23],[24]:

- Τάση λειτουργίας | 3.3V
- Μικροελεγκτής | Tensilica 32-bit RISC CPU Xtensa LX106
- Τάση εισόδου | 7-12 V
- Ψηφιακά Input/output pins | 16
- Αναλογικά Input pins | ADC (1) UARTs (1) SPis (1) I²Cs (1)
- Flash μνήμη | 4MB
- Clock speed | 80MHz
- SRAM(Static Random-Access Memory) | 64 KB



Εικόνα 16 NodeMCU development kit

3.1.10 LEDs

Δίοδος εκπομπής φωτός (LED, Light Emitting Diode) ονομάζεται ένας ημιαγωγός που εκπέμπει φωτεινή ακτινοβολία στενού φάσματος όταν του παρέχεται μία ηλεκτρική τάση κατά τη φορά ορθής πόλωσης. Το χρώμα που εκπέμπεται εξαρτάται από την χημική σύσταση του ημιαγωγικού υλικού που χρησιμοποιείται και μπορεί να είναι ορατό, υπέρυθρο ή υπεριώδες. Τα LED κατατάσσονται σε διάφορες κατηγορίες οι οποίες είναι:

- LED Ετεροεπαφές Υψηλής Εντάσεως
- LED Εκπομπής Άκρου
- LED Εκπομπής Επιφάνειας

Τα πλεονεκτήματα των LEDs είναι [7] [25]:

- Απόδοση : παράγουν περισσότερο φως ανά watt από τις λάμπες πυράκτωσης.
- Χρώμα : εκπέμπουν φως συγκεκριμένου χρώματος χωρίς την χρήση φίλτρων. Είναι πιο αποδοτικά με χαμηλότερο κόστος.
- Μέγεθος : είναι πολύ μικρά και μπορούν να τοποθετηθούν σε πίνακες αποτύπωσης.
- Ψυχρό φως : εκπέμπουν πολύ λίγη θερμότητα σε σχέση με τις κοινές πηγές φωτός.
- Χρόνος ON/OFF : έχουν γρήγορη απόκριση.
- Εστίαση : μπορούν να εστιάσουν σε ένα συγκεκριμένο σημείο.
- Αντίσταση σε κραδασμούς : είναι δύσκολο να υποστούν ζημιά όντας στοιχεία στερεάς κατάστασης.
- Τοξικότητα : δεν περιέχουν υδράργυρο όπως οι λάμπες φθορίου.

Τα μειονεκτήματά τους είναι [7] [25]:

- Εξάρτηση από τη θερμοκρασία : έχει ισχυρή εξάρτηση από τις θερμοκρασιακές συνθήκες που υπάρχουν στον χώρο που βρίσκονται.
- Υψηλό αρχικό κόστος : σήμερα είναι ακριβότερα στην αγορά αλλά αντισταθμίζεται από την χαμηλή κατανάλωση ενέργειας.
- Ποιότητα φωτός : έχουν φάσμα που διαφέρει σημαντικά από αυτό ενός ακτινοβολέα μελανού σώματος όπως ο ήλιος και αυτό σημαίνει ότι τα χρώματα ενός αντικειμένου μπορεί να διαφέρουν.
- Ευαισθησία στην τάση : είναι αρκετά ευαίσθητα στη τάση και κατ' επέκταση στο ρεύμα που τα τροφοδοτεί.
- Μόλυνση από το μπλε : τα μπλε χρώματος LED και αυτά του ψυχρού λευκού είναι πλέον ικανά να εκπέμπουν περισσότερο φως από ότι οι κοινές πηγές φωτός σημαίνει ότι μπορούν να προκαλέσουν υψηλότερη φωτορύπανση από άλλες πηγές φωτός.

Στην εργασία μας χρησιμοποιήσαμε 2 φώτα LED διαμέτρου 10mm, το ένα χρώματος λευκού και το άλλο χρώματος κόκκινου. Για να μπορέσουμε να υπολογίσουμε την αντίσταση που θα έπρεπε να κολλήσουμε μαζί με τα LEDs στο όχημα μας ακολουθήσαμε τον παρακάτω τύπο [26]:

$$\text{Resistor} = (\text{Input Voltage} - \text{LED voltage}) / \text{desired LED current}$$

Για το λευκό LED:

$$R = (3.3V - 3.4V) / 30mA \\ \Rightarrow R = -3.3\Omega$$

Οπότε για το λευκό LED δεν χρειαστήκαμε αντίσταση.

Για το κόκκινο LED:

$$R = (3.3V - 2.5V) / 30mA$$
$$\Rightarrow R = 27\Omega$$

Οπότε για το κόκκινο LED χρησιμοποιήσαμε αντίσταση 33Ω.



Εικόνα 17 LEDs σε διάφορα χρώματα

3.2 ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ (ΑΥΤΟΝΟΜΟ ΟΧΗΜΑ)

3.2.1 Εισαγωγή στο πρόγραμμα

Για την υλοποίηση του προγράμματος(τετράτροχο όχημα που πηγαίνει αυτόματα στις συντεταγμένες που δίνουμε), χρησιμοποιήθηκε η εφαρμογή του Arduino, Arduino IDE. Το πρόγραμμα αρχικοποιεί τις κατάλληλες βιβλιοθήκες και μεταβλητές, όπως της πυξίδας, του GPS, των μοτέρ κ.α. Έπειτα δίνονται οι 2 συντεταγμένες του τελικού προορισμού που πρέπει να κατευθυνθεί το όχημα. Για να φτάσει το όχημα στον τελικό προορισμό γίνονται συνεχώς λήψεις και ενημερώσεις δεδομένων του GPS για τον έλεγχο της θέσης του, καθώς επίσης και συνεχείς υπολογισμοί της κατεύθυνσης με βάση τα δεδομένα που αντλούνται από την πυξίδα και των συντεταγμένων. Τέλος, για την απόκλιση του οχήματος από τον τελικό προορισμό εκτελείται μια συνάρτηση ορισμού κατεύθυνσης οχήματος με υπολογισμό διαφοράς μεταξύ τρέχουσας θέσης του οχήματος με τον τελικό προορισμό

3.2.2 Βιβλιοθήκες

Αρχικά, πριν ξεκινήσει η κατασκευή του κώδικα, προστίθενται οι κατάλληλες βιβλιοθήκες που θα βοηθήσουν στην υλοποίηση του προγράμματος.

Οι βιβλιοθήκες που χρησιμοποιήθηκαν και υπήρχαν ήδη στο πρόγραμμα ήταν οι εξής:

- SoftwareSerial.h
- Wire.h
- Servo.h
- math.h

Οι βιβλιοθήκες που χρησιμοποιήθηκαν και εγκαταστάθηκαν χειροκίνητα ήταν οι εξής:

- TinyGPS.h
- QMC5883L.h
- L298N.h
- NewPing.h

3.2.3 Περιγραφή βιβλιοθηκών

TinyGPS.h : είναι η βιβλιοθήκη που χρησιμοποιείται για τη λειτουργία του Neo 6M GPS module που χρησιμοποιείται στην εργασία και περιέχει λειτουργίες όπως η αρχικοποίηση του GPS, η άντληση των δεδομένων του με την συνάρτηση `get_position()`, η κωδικοποίηση δεδομένων `encode()` κ.α.

QMC5883L.h : είναι η βιβλιοθήκη που χρειάζεται για την λειτουργία της πυξίδας/γυροσκοπίου που χρησιμοποιείται στην εργασία και έχει συναρτήσεις όπως η αρχικοποίηση των δεδομένων `init()`, η άντληση των δεδομένων `read()` και ο υπολογισμός της κατεύθυνσης `heading()`

L298N.h : είναι η βιβλιοθήκη που χρησιμοποιείται για την επικοινωνία της πλακέτας κίνησης των μοτέρ με το Arduino. Περιέχει λειτουργίες όπως η κίνηση και ο έλεγχος διαφορετικών σετ μοτέρ `forward()`, `stop()`, `backward`, η δήλωση ταχύτητας του κάθε μότορα `setSpeed()` κ.α.

NewPing.h : είναι η βιβλιοθήκη που χρησιμοποιείται για την λειτουργία του ultrasonic (υπερηχητικού) αισθητήρα της εργασίας και ορίζει την ελάχιστη και μέγιστη απόσταση από τον αισθητήρα και διαβάζει το σήμα για τυχόν εμπόδιο.

SoftwareSerial.h : είναι η ενσωματωμένη βιβλιοθήκη του Arduino για την σειριακή επικοινωνία του Arduino με τον υπολογιστή μέσω της θύρας USB. [1]

Wire.h : είναι η ενσωματωμένη βιβλιοθήκη του Arduino για την επικοινωνία με I²C/TWI(Two-Wire Interface) συσκευές μέσω των SDA (Serial DAta - γραμμή δεδομένων) και SCL (Serial CLock - γραμμή παλμού). [1]

Servo.h : είναι η ενσωματωμένη βιβλιοθήκη του Arduino για την κίνηση του σερβοκινητήρα που γυρνάει τον ultrasonic αισθητήρα. [1]

math.h : είναι η ενσωματωμένη βιβλιοθήκη του Arduino, η οποία περιέχει μαθηματικές εξισώσεις, ώστε να γίνονται περίπλοκες μαθηματικές πράξεις μέσα στον κώδικα όπως για παράδειγμα τα δεκαδικά νούμερα των συντεταγμένων και οι μετατροπές αυτών. [1]

3.2.4 Κώδικας

- Για την δημιουργία του κώδικα, αρχικά δηλώθηκαν οι βιβλιοθήκες που αναφέρθηκαν παραπάνω.

Πίνακας 3.5 Δήλωση Βιβλιοθηκών

```
#include <SoftwareSerial.h> //Επικοινωνία με GPS module
#include <Wire.h>           //Wire connection library (I/O) Χρήση για motor driver
#include <Servo.h>          //Servo Motor library
#include <math.h>           //Χρήση για μαθηματικές πράξεις του GPS

#include <TinyGPS.h>        //Βιβλιοθήκη GPS module
#include <QMC5883L.h>       //Βιβλιοθήκη Ψηφιακής Πυξίδας
#include <L298N.h>         //Βιβλιοθήκη Motor Driver
#include <NewPing.h>       //add Ultrasonic sensor library
```

- Έπειτα, δηλώθηκαν κάποιες απαραίτητες global μεταβλητές, όπως τις μεταβλητές της πυξίδας, του υπολογισμού απόστασης και μια μεταβλητή γενικής χρήσης για τα μοτέρ. Τέλος, δίνονται και οι δύο συντεταγμένες του τελικού προορισμού που πρέπει να κατευθυνθεί το όχημα.

Πίνακας 3.6 Δήλωση Global Μεταβλητών

```
//Μεταβλητές Πυξίδας
int QMC5883LAddress = 0x0D; //Η διεύθυνση του I2C της πυξίδας σε 16δικό
int slaveAddress; //Μεταβλητή για αποθήκευση της διεύθυνσης της πυξίδας
μετά από ολίσθηση
int x,y,z; //Μεταβλητές για αποθήκευση των x,y,z αξόνων από την πυξίδα
int cm; //μεταβλητή για αποθήκευση του
int x4=0; //Μεταβλητή γενικής χρήσης στις πράξεις υπολογισμού
κατεύθυνσης

//Μεταβλητές υπολογισμού απόστασης
byte headingData[2];
int i; //Μεταβλητή γενικής σε ελέγχους επανάληψης For
float headingcompass; //Μεταβλητή για τον υπολογισμό και αποθήκευση της
κατεύθυνσης από την πυξίδα σε radians
float headingValue; //Μεταβλητή για αποθήκευση των δεδομένων κατεύθυνσης της
πυξίδας
float declinationAngle = 4.36; //Μαγνητική Γωνία απόκλισης της Γης στην περιοχή
float heading=0; //Μεταβλητή για υπολογισμό της κατεύθυνσης του οχήματος με
βάση τις συντεταγμένες
//gps Μεταβλητές
float flat, flon; //Μεταβλητές για αποθήκευση συντεταγμένων από το GPS
float flat1=flat; // flat1 = τρέχον γεωγραφικό πλάτος. flat είναι δεδομένο από το
gps.
float flon1=flon; // flon1 = τρέχον γεωγραφικό μήκος. είναι δεδομένο από το gps.
float dist_calc=0; //Μεταβλητή υπολογισμού απόστασης από τον στόχο
float dist_calc2=0; // Μεταβλητή υπολογισμού απόστασης από τον στόχο
float diflat=0; // Μεταβλητή για πράξεις υπολογισμού με τις
συντεταγμένες
float diflon=0; // Μεταβλητή για πράξεις υπολογισμού με τις συντεταγμένες
//εισαγωγή ενός σημείου γεωγραφικού πλάτους. Αυτό θα είναι το σημείο-στόχος
μας.
float x2lat=38.090304; // Μεταβλητές ορισμού των συντεταγμένων προορισμού
float x2lon=23.795089; //
```

- Μετά, έγιναν οι αρχικοποιήσεις των αντικειμένων του κώδικα καθώς και δήλωση constant μεταβλητών, όπως η σύνδεση με το GPS με την TinyGPS και SoftwareSerial βιβλιοθήκη, την σύνδεση με την πυξίδα με την QMC5883L βιβλιοθήκη. Επίσης, ορίστηκαν κάποιες απαραίτητες αποστάσεις, βιβλιοθήκες και οι ακροδέκτες των μοτέρ. [27], [28]

Πίνακας 3.7 Αρχικοποιήσεις Αντικειμένων και εξαρτημάτων

```

//Αρχικοποίηση και Σύνδεση με το gps με την TinyGPS και SoftwareSerial
βιβλιοθήκη
TinyGPS gps; //δημιουργία αντικειμένου GPS τύπου TinyGPS
SoftwareSerial nss(9, 8); //Ορισμός σειριακής επικοινωνίας του GPS με τα
pins 9(SDA) και 8(SCL)

void gpstdump(TinyGPS &gps); //κάλεσμα συνάρτησης gpstdump
bool feedgps(); //κάλεσμα συνάρτησης feedgps
void printFloat(double f, int digits = 2); //εκτύπωση δεδομένων για debugging

//Αρχικοποίηση και Σύνδεση με την Πυξίδα με την QMC5883L
βιβλιοθήκηQMC5883L compass; //δημιουργία αντικειμένου compass τύπου
QMC5883L

//Ορισμός pins των Μοτέρ
#define ENA 10 //Ορισμός constant μεταβλητής του ακροδέκτη 10 του
Arduino ως ενεργοποίηση του σερβο μοτέρ A
#define ENB 11 //Ορισμός constant μεταβλητής του ακροδέκτη 11 του
Arduino ως ενεργοποίηση του σερβο μοτέρ B
#define pinA1 5 //Ορισμός constant μεταβλητής του ακροδέκτη 5 του Arduino
για έλεγχο του μοτέρ A1
#define pinA2 6 //Ορισμός constant μεταβλητής του ακροδέκτη 6 του Arduino
για έλεγχο του μοτέρ A2
#define pinB1 12 //Ορισμός constant μεταβλητής του ακροδέκτη 12 του
Arduino για έλεγχο του μοτέρ B1
#define pinB2 13 //Ορισμός constant μεταβλητής του ακροδέκτη 13 του
Arduino για έλεγχο του μοτέρ B2

//Ρύθμιση Βιβλιοθήκης L298N για την σύνδεση με τα μοτέρ
L298N motorA(ENA, pinA1, pinA2); //δημιουργία αντικειμένου MotorA τύπου
L298N για σερβο μοτέρ A με τους αντίστοιχους ακροδέκτες
L298N motorB(ENB, pinB1, pinB2); //δημιουργία αντικειμένου MotorB τύπου
L298N για σερβο μοτέρ B με τους αντίστοιχους ακροδέκτες

```

- Συνάρτηση `setup()` όπου γίνεται αρχικοποίηση και σύνδεση των εξαρτημάτων της εργασίας, όπως η εκκίνηση των σειριακών επικοινωνιών, η ενεργοποίηση της πυξίδας και ο ορισμός της slave διεύθυνσης της πυξίδας.

Πίνακας 3.8 Συνάρτηση `setup()`

```
//Συνάρτηση Setup για Αρχικοποίηση και Σύνδεση των εξαρτημάτων
void setup()
{
    slaveAddress = QMC5883LAddress >> 1; // Ορίζει την slave διεύθυνση της
    πυξίδας ως 1bit ολίσθηση δεξιά
    Wire.begin(); //Εκκίνηση της βιβλιοθήκης Wire και είσοδος στο I2C δίαυλο
    Serial.begin(115200); // Ορίζει το ρυθμό δεδομένων σε bits ανά δευτερόλεπτο
    (baud) για σειριακή μετάδοση δεδομένων
    nss.begin(9600); //Ορίζει το ρυθμό δεδομένων σε bits ανά δευτερόλεπτο
    (baud)
    compass.init(); //Αρχικοποίηση πυξίδας
}
```

- Βρόγχος επανάληψης κυρίου προγράμματος όπου αρχικά ξεκινάει η επανάληψη για την ενημέρωση των δεδομένων του GPS μέσω της συνάρτησης `feedgps()` με χρόνο επανάληψης κάθε $\frac{1}{4}$ του δευτερολέπτου. Έπειτα αν υπάρχουν καινούργια δεδομένα καλείται η συνάρτηση `gpsdump()` για άντληση των συντεταγμένων. Μετά καλούνται οι συναρτήσεις `headingcalc()` και `CompassHeading()` για υπολογισμό της κατεύθυνσης με βάση τις αρχικές και τελικές συντεταγμένες και με βάση τα δεδομένα της πυξίδας αντίστοιχα. Τέλος καλείται η συνάρτηση `Direction` για υπολογισμό της πορείας του οχήματος με βάση τα παραπάνω δεδομένα και ορίζεται και η απόκλιση του οχήματος από τον τελικό προορισμό.

Πίνακας 3.9 Βρόγχος επανάληψης κυρίου προγράμματος

```
//Βρόγχος επανάληψης λήψης δεδομένων GPS
void loop()
{
    bool newdata = false; //Δεν έχουμε νέα δεδομένα στην αρχή
    unsigned long start = millis(); //ορισμός χρόνου που ξεκινάει ο κώδικας(σε ms)

    // Ενημέρωση δεδομένων κάθε 1/4 του δευτερολέπτου
    while (millis() - start < 250)
    {
        if (feedgps())
            newdata = true; //έλεγχος αν υπάρχουν καινούργια δεδομένα
    }
}
```

```

if (newdata) //αν υπάρχουν καινούργια δεδομένα προχωράμε:
{
    gpsdump(gps);           //
    Serial.println("-----"); //
    Serial.println();       // ==> Ενημέρωση Θέσης οχήματος
    Headingcalc();         //
    CompassHeading();      //
    motorA.setSpeed(150);  //αλλαγή ταχύτητας
    motorB.setSpeed(165);
    Direction();
}

if(dist_calc<2){ //Ορισμός Απόκλισης σε μέτρα από τον προορισμό
    done(); //Τερματισμός αν είμαστε στις συντεταγμένες-στόχο
}
}

```

- Συνάρτηση μετατροπής και στρογγυλοποίησης δεκαδικών αριθμών. Αρχικά γίνεται έλεγχος για τυχόν αρνητικούς αριθμούς από τις πράξεις και τους μετατρέπει σε θετικούς. Έπειτα με τις κατάλληλες μετατροπές γίνεται στρογγυλοποίηση των αριθμών ώστε η εκτύπωση για παράδειγμα του 1.999 να εμφανίζεται ως "2.00". Τέλος γίνεται εξαγωγή του ακέραιου και δεκαδικού μέρους του αριθμού σε ξεχωριστές μεταβλητές.

Πίνακας 3.10 Συναρτήσεις Μετατροπής και στρογγυλοποίησης Δεδομένων

```

//Συνάρτηση μετατροπής και στρογγυλοποίησης δεκαδικών αριθμών
void printFloat(double number, int digits)
{
    // Χειρισμός αρνητικών αριθμών
    if (number < 0.0)
    {
        Serial.print('-');
        number = -number;
    }

    // Σωστή στρογγυλοποίηση ώστε το print(1.999, 2) να εμφανίζεται "2.00"
    double rounding = 0.5;
    for (uint8_t i=0; i<digits; ++i)
        rounding /= 10.0;

    number += rounding;
}

```

```

// Εξαγωγή του ακέραιου μέρους του αριθμού και εκτύπωση
unsigned long int_part = (unsigned long)number;
double remainder = number - (double)int_part;
Serial.print(int_part);

// Εκτύπωση του δεκαδικού αν απομένουν ψηφία
if (digits > 0)
  Serial.print(".");

// Εξαγωγή από το υπόλοιπο ένα προς ένα
while (digits-- > 0)
{
  remainder *= 10.0;
  int toPrint = int(remainder);
  Serial.print(toPrint);
  remainder -= toPrint;
}
}

```

- Συνάρτηση λήψης θέσης και εκτύπωσης δεδομένων του GPS και συνάρτηση κλήσης και κωδικοποίησης GPS. Αρχικά ορίζονται οι μεταβλητές τύπου long για τα δεδομένα του GPS, όπως οι συντεταγμένες, η ώρα και η ημερομηνία. Έπειτα γίνεται λήψη θέσης όταν το γεωγραφικό πλάτος και μήκος είναι τύπου long και οι τιμές είναι πολλαπλασιασμένες κατά 10.000 μέσω της συνάρτησης get_position(). Μετά, γίνεται κωδικοποίηση των δεδομένων του GPS με την συνάρτηση feedgps(). Τέλος, γίνεται λήψη θέσης όταν το γεωγραφικό πλάτος και μήκος είναι τύπου float και επιστρέφονται οι πραγματικές τιμές μέσω της συνάρτησης f_get_position().

Πίνακας 3.11 Συνάρτηση λήψης δεδομένων από το GPS

```

//Συνάρτηση λήψης και εκτύπωσης δεδομένων του GPS
void gpsdump(TinyGPS &gps)
{
  Serial.println("Acquired Data");
  Serial.println("-----");
  long lat, lon;

  unsigned long age, date, time, chars;
  unsigned short sentences, failed;

  gps.get_position(&lat, &lon, &age); //Λήψη θέσης όταν το γεωγραφικό
  πλάτος και μήκος είναι τύπου long και οι τιμές είναι πολλαπλασιασμένες κατά
  10.000

```

```

Serial.print("Lat/Long(10^-5 deg): "); Serial.print(lat); Serial.print(", ");
Serial.print(lon); //εκτύπωση δεδομένων
Serial.print(" Fix age: "); Serial.print(age); Serial.println("ms.");
//

feedgps(); //Κωδικοποίηση δεδομένων GPS

gps.f_get_position(&flat, &flon, &age); //Λήψη θέσης όταν το γεωγραφικό
πλάτος και μήκος είναι τύπου float και επιστρέφονται οι πραγματικές τιμές
Serial.print("Lat/Long(float): "); printFloat(flat, 5); Serial.print(", ");
printFloat(flou, 5); //εκτύπωση δεδομένων
Serial.print(" Fix age: "); Serial.print(age); Serial.println("ms.");
//

feedgps();

gps.stats(&chars, &sentences, &failed); //Λήψη στατιστικών δεδομένων του
gps
Serial.print("Stats: characters: "); Serial.print(chars); //
Serial.print(" sentences: "); Serial.print(sentences); //Εκτύπωση
στατιστικών δεδομένων του GPS
Serial.print(" failed checksum: "); Serial.println(failed); //
Serial.println("-----");
//
Serial.println();
distance(); //Κλήση συνάρτησης υπολογισμού απόστασης
}

//Συνάρτηση κλήσης και κωδικοποίησης gps
bool feedgps()
{
while (nss.available()) //έλεγχος επικοινωνίας με το GPS
{
if (gps.encode(nss.read())) //κωδικοποίηση δεδομένων GPS μέσω του
διαύλου της NewSoftwareSerial
return true;
}
return false;
}
}

```


- Συνάρτηση υπολογισμού απόστασης από την τρέχουσα τοποθεσία. Κάνει μετατροπή σε radians με την συνάρτηση radians() και έπειτα μετατροπή σε μέτρα για τον υπολογισμό της απόστασης των συντεταγμένων του οχήματος από τις συντεταγμένες τερματισμού.

Πίνακας 3.12 Συνάρτηση Υπολογισμού Απόστασης Τελικού προορισμού

```
//Συνάρτηση υπολογισμού απόστασης
void distance(){

//----- Υπολογισμός Απόστασης. Υπολογίζει την απόσταση από την
τρέχουσα τοποθεσία -----
    diflat=radians(x2lat-flat1); //μετατροπή σε radians
    flat1=radians(flat1); //μετατροπή τρέχον γεωγραφικού πλάτους σε
radians
    x2lat=radians(x2lat); //μετατροπή γεωγραφικού πλάτους στόχου σε
radians
    diflon=radians((x2lon)-(flon1)); //μετατροπή διαφοράς απόστασης
συντεταγμένων σε radians

    dist_calc = (sin(diflat/2.0)*sin(diflat/2.0)); //
    dist_calc2= cos(flat1);
        //
    dist_calc2*=cos(x2lat);
        //
    dist_calc2*=sin(diflon/2.0);
        //
    dist_calc2*=sin(diflon/2.0);
        // Μετατροπή σε μέτρα
    dist_calc +=dist_calc2;
        //
    dist_calc=(2*atan2(sqrt(dist_calc),sqrt(1.0-dist_calc))); //
    dist_calc*=6371000.0;
        //
    Serial.println("distance"); //-----
    Serial.println(dist_calc); //εκτύπωση απόστασης(σε μέτρα)
}
}
```

- Συνάρτηση υπολογισμού κατεύθυνσης με βάση τις γεωγραφικές συντεταγμένες και τις συντεταγμένες τερματισμού και εκτύπωση αποτελέσματος. Αρχικά γίνεται μετατροπή των συντεταγμένων σε radians με την συνάρτηση radians() και μετά μετατροπή από radians σε μοίρες και αποθηκεύεται σε μεταβλητή ακεραίου. Αν το αποτέλεσμα των μοιρών είναι αρνητικό προστίθενται 360 μοίρες ώστε να γίνει θετικό και εκτυπώνεται.

Πίνακας 3.13 Συνάρτηση υπολογισμού κατεύθυνσης οχήματος

```
//Συνάρτηση υπολογισμού κατεύθυνσης
void Headingcalc(){

    flon1 = radians(flon1); //μετατροπή σε radians
    x2lon = radians(x2lon); //-----
    heading = atan2(sin(x2lon-flon1)*cos(x2lat),cos(flat1)*sin(x2lat)-
sin(flat1)*cos(x2lat)*cos(x2lon-flon1)),2*3.1415926535; //
    heading = heading*180/3.1415926535;

                                                    // Μετατροπή από radians σε
Μοίρες
    int head =heading; //Μεταφορά σε μεταβλητή integer

    if(head<0){
        heading+=360; //Αν το αποτέλεσμα είναι αρνητικό τότε πρόσθεσε 360
        μοίρες ώστε να γίνει θετικό
    }
    Serial.println("heading:");
    Serial.println(heading); // Εκτύπωση αποτελέσματος.
}
```

- Συνάρτηση υπολογισμού κατεύθυνσης της πυξίδας όπου αντλούνται δεδομένα από την πυξίδα και με συνυπολογισμό της μαγνητικής γωνίας απόκλισης της Γης στην περιοχή, υπολογίζεται η μαγνητική κατεύθυνση του οχήματος ώστε να συγκριθεί με τη γεωγραφική κατεύθυνση για τη σωστή πορεία του οχήματος. Τέλος, γίνεται εκτύπωση των αποτελεσμάτων

Πίνακας 3.14 Συνάρτηση υπολογισμού κατεύθυνσης από την πυξίδα

```
void CompassHeading(){

    compass.read(&x,&y,&z); //Αντληση δεδομένων από την πυξίδα
    headingValue = atan2(y, x); //Εφαπτομένη τόξου των αξόνων x και y
    headingValue += declinationAngle; //Συνυπολογισμός μαγνητικής Γωνίας
    απόκλισης της Γης στην περιοχή
```

```

// Διόρθωση για τις αρνητικές τιμές.
if(headingValue < 0)
    headingValue += 2*PI;

// Έλεγχος ορίου τιμής σε 2π μετά την προσθήκη Γωνίας απόκλισης.
if(headingValue > 2*PI)
    headingValue -= 2*PI;

// Μετατροπή rad σε Μοίρες για ευαναγνωσιμότητα.
headingcompass = headingValue * 180/M_PI;

Serial.println("current heading(compass):"); //Εκτύπωση αποτελέσματος.
Serial.println(headingcompass);           //
}

```

- Συνάρτηση ορισμού κατεύθυνσης οχήματος όπου υπολογίζεται η διαφορά μεταξύ τρέχουσας θέσης του οχήματος με τον προορισμό και αποθήκευση της απόλυτης τιμής. Έπειτα γίνεται ο κατάλληλος έλεγχος των μοιρών ώστε να αλλάξει η πορεία του οχήματος σύμφωνα με την απόκλιση από την σωστή κατεύθυνση τερματισμού. Επίσης, γίνονται πράξεις για την βελτίωση της πορείας του οχήματος και μείωση των διακοπών/σταματημάτων του και τέλος, εκτελούνται οι συναρτήσεις πορείας.

Πίνακας 3.15 Συνάρτηση ορισμού πορείας οχήματος

```

//Συνάρτηση ορισμού κατεύθυνσης οχήματος
void Direction(){
    x4=heading-headingcompass; //Υπολογισμός της διαφοράς μεταξύ της
    τρέχουσας κατεύθυνσης με την κατεύθυνση προορισμού
    float absolute= abs(x4); //Απόλυτη τιμή για ευαναγνωσιμότητα της απόκλισης
    if(absolute>180){
        absolute-=180; //Μετατροπή από 0-360 μοίρες σε 0 - 180
    }
    Serial.println(absolute); //εκτύπωση

    //Καθορισμός της πορείας του οχήματος σύμφωνα με την απόκλιση από την
    σωστή κατεύθυνση.
    if(x4>=-180){ //Αν η απόκλιση είναι μεταξύ -180 και 0 μοίρες
        if(x4<=0){
            turn=8; //Ορίζει turn =8 το οποίο σημαίνει "δεξιά"
            Serial.println("right1"); //εκτύπωση
        }
    }
    if(x4<-180){ //Αν η απόκλιση είναι μικρότερη των -180 μοιρών
        turn=5; //Ορίζει turn=5 το οποίο σημαίνει "αριστερά"
        Serial.println("left1"); //εκτύπωση
    }
    if(x4>=0){ //Αν η απόκλιση είναι μεταξύ 0 και 180 μοίρες

```

```

    if(x4<180){
        turn=5; //Ορίζει turn=5 το οποίο σημαίνει "αριστερά"
        Serial.println("left2"); //εκτύπωση
    }
}
if(x4>=180){ //Αν η απόκλιση είναι μεγαλύτερη ή ίση των 180 μοιρών
    turn=8; //Ορίζει turn =8 το οποίο σημαίνει "δεξιά"
    Serial.println("Right2"); //εκτύπωση

}

//Πράξεις για βελτίωση της ομαλής πορείας του οχήματος και μείωση των
αχρειαστων διακοπών της κίνησης αυτού
float hd = headingcompass; //μεταφορά κατεύθυνσης πυξίδας σε άλλη
μεταβλητή
float range; //μεταβλητή εύρους κίνησης του οχήματος σε ευθεία
range = hd - heading; //Υπολογισμός εύρους κίνησης του οχήματος σε
ευθεία
//Αλλαγή εύρους κίνησης ανάλογα με την απόσταση από τον στόχο
if(dist_calc<2){ //Αν η απόσταση από τον στόχο είναι μικρότερη των 2 μέτρων
τότε:
    if(-7 <= range && range <= 7){ //Ορισμός εύρους σε 7 μέτρα ανά
πλευρά(αριστερά-δεξιά)
        turn=3; //Ορίζει turn=3 το οποίο σημαίνει "ευθεία"
    }
}
}
else{
    if(-4 <= range && range <= 4){ //Ορισμός εύρους σε 4 μέτρα ανά
πλευρά(αριστερά-δεξιά)
        turn=3; //Ορίζει turn=3 το οποίο σημαίνει "ευθεία"
    }
}
}

//Εκτέλεση συναρτήσεων πορείας
if(turn==3){
    goStraight();
}
else if(turn==8){
    RightTurn();
}
else{
    LeftTurn();
}
}
}

```

- Συνάρτηση τερματισμού του οχήματος, όπου ακινητοποιούνται τα σέτ A και B των μοτέρ.

Πίνακας 3.16 Συνάρτηση τερματισμού των Μοτέρ

```
void done(){
  motorA.stop();
  motorB.stop();
  done();
}
```

- Συνάρτηση κίνησης ευθεία του οχήματος, όπου κινούνται τα σέτ A και B των μοτέρ μπροστά.

Πίνακας 3.17 Συνάρτηση κίνησης οχήματος ευθεία

```
void goStraight(){
  motorA.forward();
  motorB.forward();
}
```

- Συνάρτηση κίνησης του οχήματος προς τα αριστερά με επανυπολογισμό και έλεγχο της κατεύθυνσης, όπου αφού ξαναγίνει έλεγχος της κατεύθυνσης του οχήματος κινούνται το σέτ μοτέρ A πίσω και το σέτ B των μοτέρ μπροστά.

Πίνακας 3.18 Συνάρτηση κίνησης οχήματος αριστερά

```
void LeftTurn(){
  if(headingcompass+2>heading){ //
    if(headingcompass-2<heading){ //
      motorA.stop(); //
      motorB.stop(); //
      return; //
    } //
  } // Επανυπολογισμός και έλεγχος κατεύθυνσης σε περίπτωση
  υπερστροφής
  x4=headingcompass-heading; //
  if(x4>=-180){ //
    if(x4<=0){ //
      return; //
    } //
  }else if(x4>=180){ //
    return;
  }else{
    motorA.setSpeed(150); //Μείωση ταχύτητας
    motorB.setSpeed(150);
    motorA.backward(); // Στρίψιμο Αριστερά
    motorB.forward(); // --/--
    CompassHeading();
    LeftTurn();
  }
}
```

- Συνάρτηση κίνησης του οχήματος προς τα δεξιά με επανυπολογισμό και έλεγχο της κατεύθυνσης, όπου αφού ξαναγίνει έλεγχος της κατεύθυνσης του οχήματος κινούνται το σετ μοτέρ A μπροστά και το σετ B των μοτέρ πίσω.

Πίνακας 3.19 Συνάρτηση κίνησης του οχήματος δεξιά

```

//Συνάρτηση Κίνησης Δεξιά
void RightTurn(){
  //Επανυπολογισμός και έλεγχος κατεύθυνσης σε περίπτωση υπερστροφής
  if(headingcompass+2>heading){
    if(headingcompass-2<heading){
      motorA.stop();
      motorB.stop();
      return;
    }
  }
  x4=headingcompass-heading;
  if(x4<-180){
    return;
  }else if(x4>=0){
    if(x4<180){
      return;
    }
  }else{
    motorA.setSpeed(150); //Μείωση ταχύτητας
    motorB.setSpeed(150);
    motorA.forward(); //Στρίψιμο Δεξιά
    motorB.backward(); // --/--
    CompassHeading();
    RightTurn();
  }
}

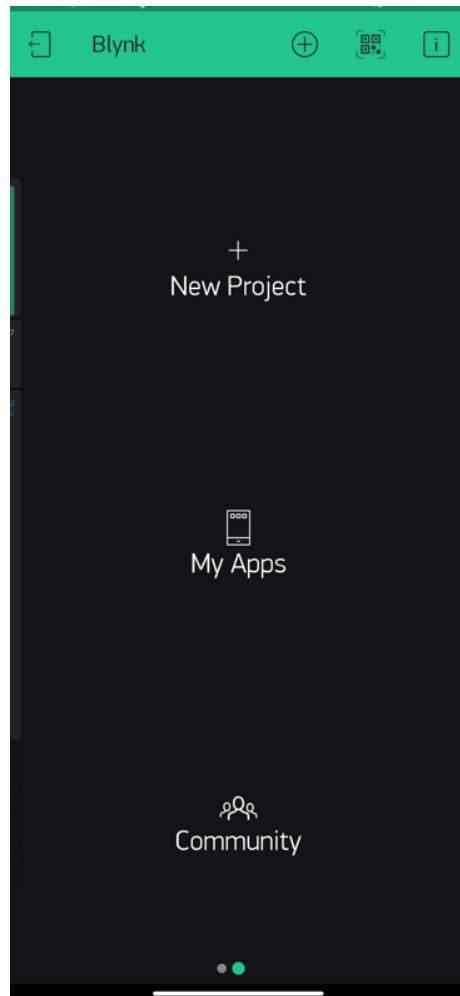
```

3.3 ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΕΞΥΠΝΗ ΣΥΣΚΕΥΗ

3.3.1 Εισαγωγή στην πλατφόρμα Blynk

Για την υλοποίηση του προγράμματος (τετράτροχο όχημα που προχωράει τηλεκατευθυνόμενα μέσω εφαρμογής), έγινε χρήση και εγκατάσταση της εφαρμογής του Arduino, Arduino IDE και της εφαρμογής Blynk για έξυπνες συσκευές. Για το Arduino έγινε αναφορά στην προηγούμενη λειτουργία, σε αυτήν όμως χρησιμοποιήθηκε το NodeMCU ως μικροελεγκτής για να περαστεί ο κώδικας.

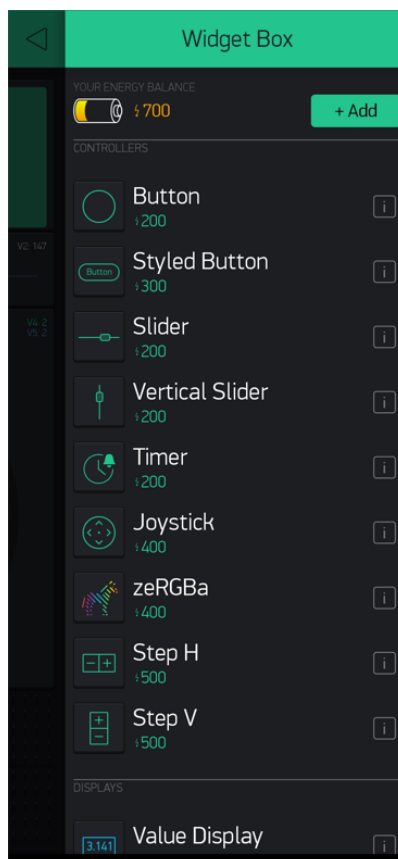
Όσον αφορά την εφαρμογή Blynk, η οποία εγκαταστάθηκε στο έξυπνο κινητό και μέσω Wi-Fi δίνονται εντολές στο τετράτροχο όχημα. Η εφαρμογή παρέχεται δωρεάν από το επίσημο site της <https://Blynk.io/>. Με την έναρξη της εφαρμογής και αφού ολοκληρωθεί η δημιουργία λογαριασμού του χρήστη θα εμφανισθεί το παρακάτω περιβάλλον [14] :



Εικόνα 18 Blynk home screen

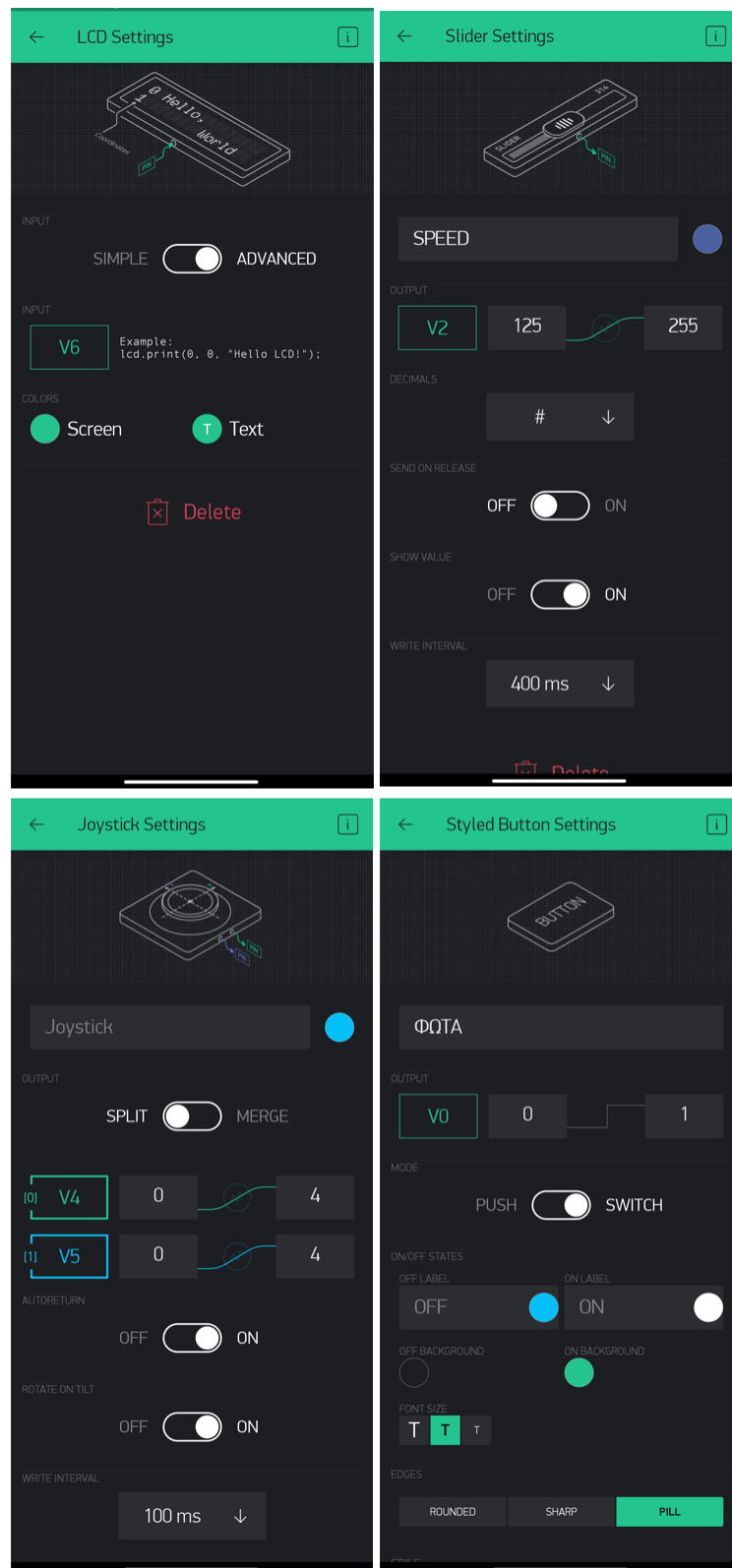
Έπειτα, πατώντας στο “New Project” θα δημιουργηθεί ένα καινούργιο project για τον σχεδιασμό της εφαρμογής. Μετά, θα πρέπει να γίνει επιλογή της συσκευής που θα χρησιμοποιηθεί, στην περίπτωση αυτής της εργασίας εφόσον θα χρησιμοποιηθεί το NodeMCU θα ορισθεί στην εφαρμογή η αντίστοιχη επιλογή.

Στην εφαρμογή θα τοποθετηθεί ένα joystick, το οποίο κινώντας το στις 8 διαφορετικές θέσεις (πάνω, κάτω, δεξιά, αριστερά, πάνω-δεξιά, πάνω-αριστερά, κάτω-δεξιά, κάτω-αριστερά) θα δίνει με τον κατάλληλο κώδικα, εντολή στο όχημα να κινηθεί αντίστοιχα. επίσης, θα χρειαστεί ένας ρυθμιστής έντασης ταχύτητας των μοτέρ όπου οποιαδήποτε στιγμή θα μπορεί ο χρήστης να την μειώσει ή να την αυξήσει σέρνοντας το κουμπί (slider) της εφαρμογής προς τα αριστερά (μείωση ταχύτητας) είτε προς τα δεξιά (αύξηση ταχύτητας). Έπειτα, θα προστεθεί ένα κουμπί το οποίο θα έχει τις επιλογές on/off και με το κάθε πάτημα του κουμπιού θα εναλλάσσονται με σκοπό να ανάβει ή να σβήνει ένα led λευκού χρώματος που έχει συνδεθεί στο όχημα. Τέλος, θα προστεθεί και μία LCD οθόνη η οποία θα δείχνει την IP του NodeMCU που είναι εγκατεστημένο στο όχημα. Για να προστεθούν αυτά τα 4 γραφικά στοιχεία στην εφαρμογή θα επιλεγθεί το σύμβολο πάνω δεξιά που μοιάζει σαν σταυρός μέσα σε κύκλο και θα μας πάει στην επιλογή Widget Box και εκεί υπάρχουν και τα 4 γραφικά στοιχεία που χρειάζονται, όπου εκτός από αυτά παρέχει και μεγάλη πληθώρα από διάφορα γραφικά στοιχεία όπως Timer, Step H, Step V, GPS sensor, Light Sensor κ.α. τα οποία δεν θα χρειαστούν για τη συγκεκριμένη εφαρμογή. [14]



Εικόνα 19 Blynk widget box

Τα 4 γραφικά στοιχεία είναι α) Joystick για την λειτουργία με το joystick για τις 8 διαφορετικές κινήσεις, β) Styled Button για το LED, γ) LCD για την LCD οθόνη για την IP και δ) Slider για το κουμπί της αυξομείωσης της ταχύτητας του οχήματος.



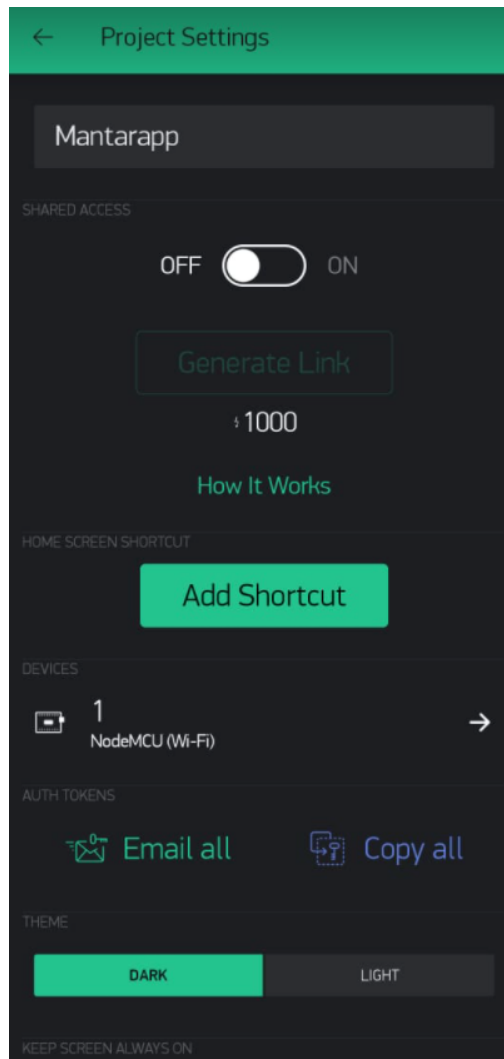
Εικόνα 20 widgets (LCD, Slider, Joystick, Button)

Μέσα στις ρυθμίσεις του κάθε γραφικού στοιχείου μπορούν να ορισθούν τα virtual pins (εικονικοί ακροδέκτες), ώστε να στέλνονται και να λαμβάνονται δεδομένα, είτε γενικά για να επικοινωνεί με το NodeMCU μέσω αυτών και διάφορες άλλες ρυθμίσεις όπως το ελάχιστο και το μέγιστο της έντασης της ταχύτητας για τον αυξομειωτή ταχύτητας του οχήματος μέσω του slider. Αφού προστεθούν τα παραπάνω γραφικά στοιχεία το γραφικό περιβάλλον της εφαρμογής θα είναι σαν την παρακάτω εικόνα:



Εικόνα 21 Blynk η εφαρμογή της εργασίας

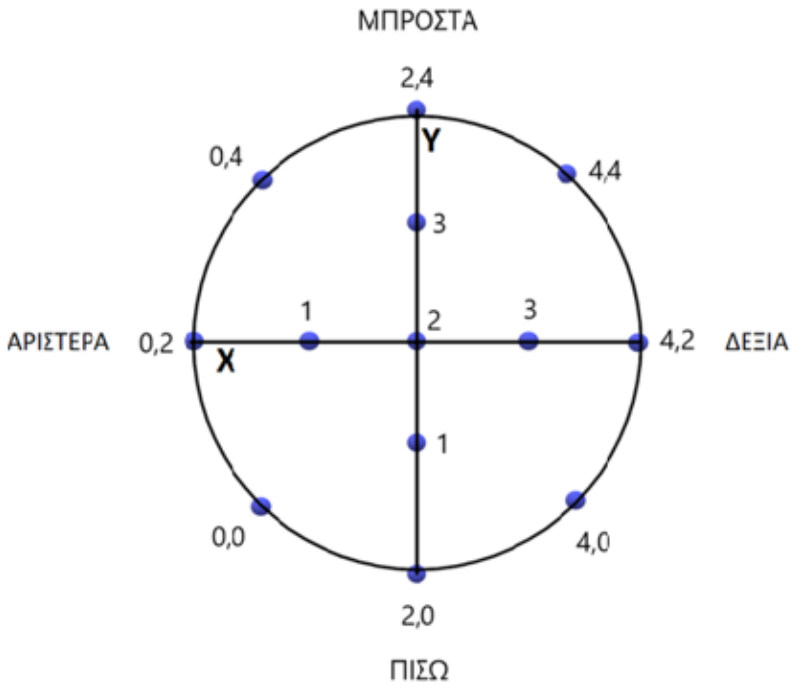
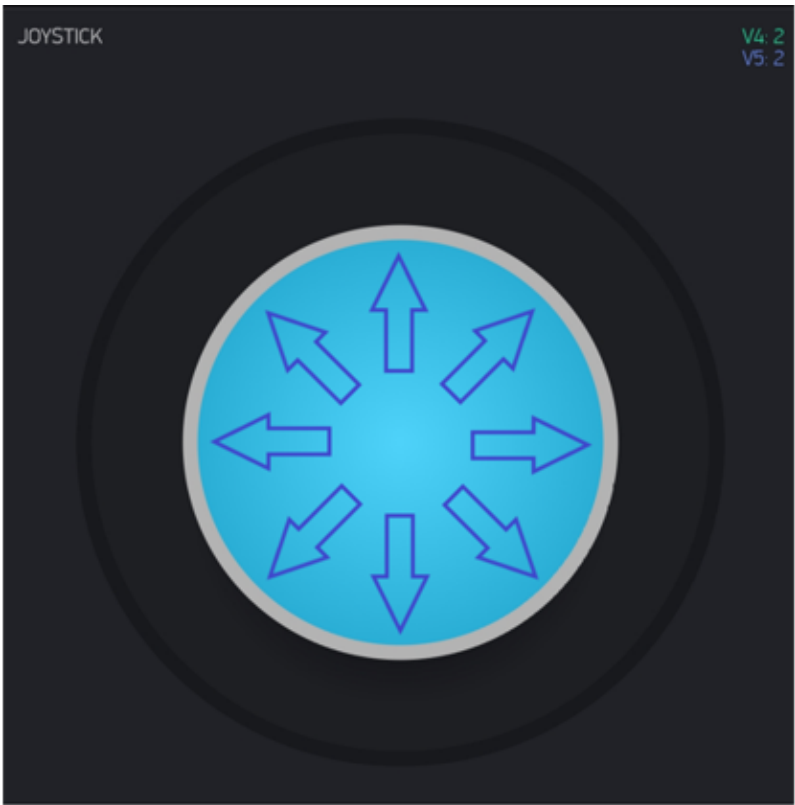
Τέλος, επιλέγοντας το σύμβολο “γρανάζι” πάνω, μας οδηγεί στις ρυθμίσεις της εφαρμογής και από εκεί μπορεί να δημιουργηθεί συντόμευση της εφαρμογής στην έξυπνη κινητή συσκευή και να αποκτήσουμε το authentication token (κλειδί) το οποίο είναι μοναδικό, έτσι ώστε να συνδεθεί η εφαρμογή με το NodeMCU. Φυσικά, εφόσον η επικοινωνία μεταξύ τους θα γίνει μέσω Wi-Fi, είναι απαραίτητη προϋπόθεση να έχει πρόσβαση ενεργή στο διαδίκτυο η κινητή έξυπνη συσκευή. [14]



Εικόνα 22 Blynk Project settings

3.3.2 Joystick

Για να είναι εφικτό το όχημα να εκτελέσει τις οκτώ κινήσεις που απαιτούνται μέσω του Joystick από την εφαρμογή, θα πρέπει πρώτα να οριστούν οι κατάλληλες τιμές στο joystick. Έπειτα, πρέπει να μεταφραστούν οι τιμές αυτές στους δύο άξονες (X, Y) στον κώδικα. Για τον άξονα X από αριστερά προς τα δεξιά τοποθετήθηκαν οι τιμές από 0 ως 4, δηλαδή αριστερά η τιμή 0, στο κέντρο η τιμή 2. Στα δεξιά ορίστηκε η τιμή 4 και ενδιάμεσα τους αντίστοιχα οι τιμές 1 και 3. Έτσι λοιπόν και στον άξονα Y αντίστοιχα από κάτω (πίσω) προς τα πάνω (μπροστά) τοποθετήθηκαν οι ίδιες τιμές. Οπότε το σχέδιο του joystick θα μοιάζει σαν την παρακάτω εικόνα.



Εικόνα 23 η λογική πίσω από το joystick

Οπότε, για την σωστή εκτέλεση των οκτώ κινήσεων θα πρέπει να οριστούν οι σωστές τιμές των δύο αξόνων. Δηλαδή για:

- Μπροστά : 2(X), 4(Y)
- Πίσω : 2(X), 0(Y)
- Δεξιά : 4(X), 2(Y)
- Αριστερά : 0(X), 2(Y)
- Μπροστά-δεξιά : 4(X), 4(Y)
- Μπροστά-αριστερά : 0(X), 4(Y)
- Πίσω-δεξιά : 4(X), 0(Y)
- Πίσω-αριστερά : 0(X), 0(Y)

3.3.3 Βιβλιοθήκες

Αρχικά, πριν την κατασκευή του κώδικα, πρέπει να προστεθούν οι κατάλληλες βιβλιοθήκες που θα βοηθήσουν στην υλοποίηση του προγράμματος. Κάποιες βιβλιοθήκες ήταν ήδη εγκατεστημένες στο Arduino πρόγραμμα και κάποιες άλλες χρειάστηκε να εγκατασταθούν χειροκίνητα από το Διαδίκτυο. [29] [14]

Οι βιβλιοθήκες που χρησιμοποιήθηκαν και υπήρχαν ήδη στο πρόγραμμα ήταν οι εξής:

- SoftwareSerial.h
- SPI.h

Οι βιβλιοθήκες που χρησιμοποιήθηκαν και εγκαταστάθηκαν χειροκίνητα ήταν οι εξής:

- SimpleTimer.h
- BlynkSimpleEsp8266.h

3.3.4 Περιγραφή βιβλιοθηκών

SoftwareSerial.h : Αναφορά στο υποκεφάλαιο 3.2.3

SPI.h : Ενσωματωμένη βιβλιοθήκη του Arduino που είναι ένα πρωτόκολλο ταυτόχρονης επικοινωνίας σειριακών δεδομένων που χρησιμοποιείται από μικροελεγκτές, για γρήγορη επικοινωνία με μία ή περισσότερες περιφερειακές συσκευές. [1]

SimpleTimer.h : βιβλιοθήκη για το χειρισμό και την επεξεργασία του πραγματικού χρόνου μέσα στον κώδικα. [1]

BlynkSimpleEsp8266.h : βιβλιοθήκη που χρησιμοποιείται για την διασύνδεση μιας ESP8266 συσκευής με τη πλατφόρμα Blynk και την αποστολή και λήψη δεδομένων μεταξύ τους. [14]

3.3.5 Κώδικας

- Για την δημιουργία του κώδικα, αρχικά δηλώθηκαν οι βιβλιοθήκες που αναφέρθηκαν παραπάνω. [14]

Πίνακας 3.20 Δήλωση Βιβλιοθηκών

```
#include <SoftwareSerial.h> //Επικοινωνία με GPS module
#include <L298NX2.h> //Βιβλιοθήκη Motor Driver
#include <SPI.h> //Βιβλιοθήκη ταυτόχρονης σειριακής επικοινωνίας
#include <SimpleTimer.h> //Βιβλιοθήκη για χρήση του πραγματικού χρόνου
#include <BlynkSimpleEsp8266.h> //Βιβλιοθήκη Blynk για επικοινωνία με το
οικοσύστημα
```

- Έπειτα, δηλώθηκαν οι απαραίτητες global και constant μεταβλητές. Επίσης έγιναν οι απαραίτητες αρχικοποιήσεις, όπως μεταβλητές για τους εικονικούς ακροδέκτες v0-5, ακροδέκτες για τα μοτέρ, για τα LEDs, την LCD οθόνη της εφαρμογής και για την σύνδεση του Blynk. [14] [30] [31] [32]

Πίνακας 3.21 Δήλωση μεταβλητών και αρχικοποιήσεις

```
#define BLYNK_PRINT Serial //Ορισμός Εκτύπωσης κατάστασης Blynk στη
σειριακή

int pinValue0; //
int pinValue1; //
int pinValue2; // ==> Μεταβλητές για τα εικονικά Pin V0-5 της εφαρμογής
int pinValue3; //
int pinValue4; //
int pinValue5; //

char auth[] = "V27x1mNNrHx1NTJatLsDy5Y3fTxnMN7V"; //Πιστοποιητικό Κλειδί
για επικοινωνία με τον Blynk Server

const char* ssid = "FORTHNET2"; //
const char* password = "2110198327"; //Στοιχεία της Ασύρματης Σύνδεσης του
NodeMCU

bool Connected2Blynk = false; //Λογική μεταβλητή για έλεγχο Σύνδεσης του
Blynk

SimpleTimer timer; //Αρχικοποίηση Μεταβλητής Χρόνου

WidgetLCD lcd(V6); //Δήλωση του εικονικού Pin της LCD στην Εφαρμογή

// Αυτή η Συνάρτηση στέλνει κάθε 1 δευτερόλεπτο την IP του NodeMCU στην LCD
οθόνη της Εφαρμογής
void myLCDEvent()
```

```

{

lcd.print(0, 0, WiFi.localIP().toString()); //Εκτύπωση της IP ως συμβολοσειρά στην
LCD
}

int firstVal, secondVal,thirdVal; //Μεταβλητές δεδομένων του joystick και του slider
ταχύτητας

int ledPin = D7; //Μεταβλητή ορισμού Pin του Λευκού(Μπροστά) LED
int ledPin2 = D8; //Μεταβλητή ορισμού Pin του Κόκκινου(Πίσω) LED

//Ορισμός pins των Μοτέρ
const unsigned int ENA = D1; //Pin Ενεργοποίησης του Σετ1 Μοτέρ
const unsigned int pinA1 = D3; //Pin Μοτέρ 1
const unsigned int pinA2 = D4; //Pin Μοτέρ 2
const unsigned int pinB1 = D5; //Pin Μοτέρ 3
const unsigned int pinB2 = D6; //Pin Μοτέρ 4
const unsigned int ENB = D2; //Pin Ενεργοποίησης του Σετ2 Μοτέρ

//Πύθμιση Βιβλιοθήκης L298NX για την σύνδεση με τα μοτέρ
L298NX2 motors(ENA, pinA1, pinA2, ENB, pinB1, pinB2);

//Συνάρτηση Setup για Αρχικοποίηση και Σύνδεση των εξαρτημάτων
void setup(){

Serial.begin(9600); // Ορίζει το ρυθμό δεδομένων σε bits ανά δευτερόλεπτο (baud)
για σειριακή μετάδοση δεδομένων
Blynk.begin(auth, ssid, password); //Αρχικοποίηση Σύνδεσης του NodeMCU στο
Δίκτυο και Σύνδεσης με το Blynk
timer.setInterval(1000L,myLCDEvent); //Ορισμός καθυστέρησης σε ms της
αποστολής δεδομένων στην LCD

        Serial.print("Use this URL to connect: "); //
        Serial.print("http://"); //====> Εκτύπωση της
IP στην Σειριακή Σύνδεση
        Serial.print(WiFi.localIP()); //
        Serial.println(""); //

motors.setSpeed(125); //Αρχικοποίηση της ταχύτητας των Μοτέρ

pinMode(ledPin, OUTPUT); //Ορισμός Pin του Led1(Λευκού) ως Έξοδο
pinMode(ledPin2, OUTPUT); //Ορισμός Pin του Led1(Κόκκινου) ως Έξοδο
digitalWrite(ledPin, LOW); //Αρχικοποίηση του Led1(Λευκού) ως Σβηστό
digitalWrite(ledPin2, LOW); //Αρχικοποίηση του Led2(Κόκκινου) ως Σβηστό
}

```

- Μετά, γίνεται ένας έλεγχος για την επικοινωνία του NodeMCU με την εφαρμογή Blynk με την συνάρτηση `connected()`. Αν η επικοινωνία είναι επιτυχημένη γίνονται οι κατάλληλοι έλεγχοι για τις τιμές του joystick και αντίστοιχα ορίζονται οι εντολές για την κίνηση του οχήματος στις οκτώ κατευθύνσεις που έχουν δηλωθεί. Σε κάποιες κινήσεις του οχήματος μειώθηκε η ταχύτητα κάποιων μοτέρ για την πιο ομαλή κίνηση του. Επίσης γίνεται έλεγχος για το κουμπί LED της εφαρμογής και αντίστοιχα ανάβει το λευκό LED. Τέλος το κόκκινο LED ανάβει όταν το όχημα κάνει κάποια κίνηση προς τα πίσω. [14] [32] [33] [34]

Πίνακας 3.22 Βρόγχος επανάληψης κυρίου προγράμματος

```
//Συνάρτηση Επανάληψης του Arduino για εκτέλεση του Βασικού Κώδικα
void loop(){

if(Blynk.connected() == false ){ //Έλεγχος Εσφαλμένης Σύνδεσης με το Blynk
  Serial.println("Blynk Not Connected!"); //Μήνυμα Σφάλματος
  Blynk.connect(); //Προσπάθεια Επανασύνδεσης
  delay(100); // Καθυστέρηση 100ms
}else if(Blynk.connected() == true){ //Έλεγχος Επιτυχημένης Σύνδεσης με το Blynk
  Blynk.run(); //Εκτέλεση Διεργασιών Blynk
  timer.run(); //Αρχειοποίηση του Blynk Timer

firstVal = pinValue4; //Αποθήκευση της Τιμής του εικονικού Pin V4 ως Μεταβλητή
joystick κίνησης Αριστερά-Δεξιά
secondVal = pinValue5; //Αποθήκευση της Τιμής του εικονικού Pin V5 ως
Μεταβλητή joystick κίνησης Μπροστά-Πίσω
thirdVal = pinValue2; //Αποθήκευση της Τιμής του εικονικού Pin V2 ως Μεταβλητή
ταχύτητας

  Serial.println(firstVal); //
  Serial.println(secondVal); // Εκτύπωση των Μεταβλητών στην Σειριακή
  Serial.println(thirdVal); //

          if(pinValue0 == 1){ //Έλεγχος τιμής του εικονικού Pin
V0(Κουμπί) για την ενεργοποίηση του Λευκού Led
            digitalWrite(ledPin2, HIGH);
          }else{ //Έλεγχος τιμής του εικονικού Pin
V0(Κουμπί) για την απενεργοποίηση του Λευκού Led
            digitalWrite(ledPin2, LOW);
          }

if(firstVal == 2 && secondVal == 4) //Έλεγχος Τιμών joystick για Κίνηση Μπροστά
{
  motors.setSpeed(thirdVal); //Ορισμός Ταχύτητας όλων των Μοτέρ σύμφωνα
με το Slider της Εφαρμογής
  motors.forward(); //Κίνηση Μοτέρ Μπροστά
  digitalWrite(ledPin, LOW); //Ορισμός Κόκκινου Led ως Σβηστό
}else if(firstVal > 2 && secondVal > 2) //Έλεγχος Τιμών joystick για Κίνηση
Μπροστά-Δεξιά
{
```



```

    motors.setSpeedA(thirdVal/3); //Ορισμός Ταχύτητας Δεξιών Μοτέρ στο 1/3 της
    τιμής του Slider της Εφαρμογής
    motors.setSpeedB(thirdVal); //Ορισμός Ταχύτητας Αριστερών Μοτέρ
    σύμφωνα με το Slider της Εφαρμογής
    motors.forward(); //Κίνηση Μοτέρ Μπροστά
    digitalWrite(ledPin, LOW); //Ορισμός Κόκκινου Led ως Σβηστό
} else if (firstVal == 4 && secondVal == 2) //Ελεγχος Τιμών joystick για Κίνηση
Δεξιά
{
    motors.setSpeedA(0); //Ορισμός Ταχύτητας Δεξιών Μοτέρ ως
    Ακίνητοποιημένα
    motors.setSpeedB(thirdVal); //Ορισμός Ταχύτητας Αριστερών Μοτέρ σύμφωνα
    με το Slider της Εφαρμογής
    motors.forward(); //Κίνηση Μοτέρ Μπροστά
    digitalWrite(ledPin, LOW); //Ορισμός Κόκκινου Led ως Σβηστό
} else if (firstVal == 0 && secondVal == 2) //Ελεγχος Τιμών joystick για Κίνηση
Αριστερά
{
    motors.setSpeedA(thirdVal); //Ορισμός Ταχύτητας Δεξιών Μοτέρ σύμφωνα με
    το Slider της Εφαρμογής
    motors.setSpeedB(0); //Ορισμός Ταχύτητας Αριστερών Μοτέρ ως
    Ακίνητοποιημένα
    motors.forward(); //Κίνηση Μοτέρ Μπροστά
    digitalWrite(ledPin, LOW); //Ορισμός Κόκκινου Led ως Σβηστό
} else if (firstVal < 2 && secondVal > 2) //Ελεγχος Τιμών joystick για Κίνηση
Μπροστά-Αριστερά
{
    motors.setSpeedA(thirdVal); //Ορισμός Ταχύτητας Δεξιών Μοτέρ σύμφωνα
    με το Slider της Εφαρμογής
    motors.setSpeedB(thirdVal/3); //Ορισμός Ταχύτητας Αριστερών Μοτέρ στο 1/3
    της τιμής του Slider της Εφαρμογής
    motors.forward(); //Κίνηση Μοτέρ Μπροστά
    digitalWrite(ledPin, LOW); //Ορισμός Κόκκινου Led ως Σβηστό
} else if (firstVal == 2 && secondVal == 2) //Ελεγχος Τιμών joystick για
Ακίνητοποίηση
{
    motors.stop(); //Ορισμός Μοτέρ ως Ακίνητοποιημένα
    digitalWrite(ledPin, LOW); //Ορισμός Κόκκινου Led ως Σβηστό
} else if (firstVal == 2 && secondVal == 0) //Ελεγχος Τιμών joystick για Κίνηση
Πίσω
{
    motors.setSpeed(thirdVal); //Ορισμός Ταχύτητας όλων των Μοτέρ σύμφωνα
    με το Slider της Εφαρμογής
    motors.backward(); //Κίνηση Μοτέρ Πίσω
    digitalWrite(ledPin, HIGH); //Ορισμός Κόκκινου Led ως Αναμμένο
} else if (firstVal > 2 && secondVal < 2) //Ελεγχος Τιμών joystick για Κίνηση Πίσω-
Δεξιά
{
    motors.setSpeedA(thirdVal/3); //Ορισμός Ταχύτητας Δεξιών Μοτέρ στο 1/3 της
    τιμής του Slider της Εφαρμογής
    motors.setSpeedB(thirdVal); //Ορισμός Ταχύτητας Αριστερών Μοτέρ
    σύμφωνα με το Slider της Εφαρμογής
    motors.backward(); //Κίνηση Μοτέρ Πίσω
    digitalWrite(ledPin, HIGH); //Ορισμός Κόκκινου Led ως Αναμμένο

```

```

}else if(firstVal < 2 && secondVal < 2)//Ελεγχος Τιμών joystick για Κίνηση Πίσω-
Αριστερά
{
  motors.setSpeedA(thirdVal); //Ορισμός Ταχύτητας Δεξιών Μοτέρ
  σύμφωνα με το Slider της Εφαρμογής
  motors.setSpeedB(thirdVal/3);//Ορισμός Ταχύτητας Αριστερών Μοτέρ στο 1/3
  της τιμής του Slider της Εφαρμογής
  motors.backward(); //Κίνηση Μοτέρ Πίσω
  digitalWrite(ledPin, HIGH);//Ορισμός Κόκκινου Led ως Αναμμένο
}
}
}
//Τέλος Επανάληψης

```

- Τέλος, χρησιμοποιώντας τις ενσωματωμένες διεργασίες της βιβλιοθήκης Blynk, διαβάζονται και αποθηκεύονται τα δεδομένα των εικονικών ακροδεκτών της εφαρμογής. Επίσης εκτυπώνονται στην σειριακή. [14]

Πίνακας 3.23 Διεργασίες της Βιβλιοθήκης Blynk

```

//Διεργασία Blynk για Διάβασμα δεδομένων του εικονικού Pin V0
BLYNK_WRITE(V0)
{
  pinValue0 = param.asInt(); //Μετατροπή και αποθήκευση της εισερχόμενης τιμής
  του V0 ως Ακέραιος
  Serial.print("V0 cords x value is: "); // Εκτύπωση Τιμής στην Σειριακή
  Serial.println(pinValue0); //
}

//Διεργασία Blynk για Διάβασμα δεδομένων του εικονικού Pin V2
BLYNK_WRITE(V2)
{
  pinValue2 = param.asInt(); //Μετατροπή και αποθήκευση της εισερχόμενης τιμής
  του V2 ως Ακέραιος
  Serial.print("V0 cords x value is: "); // Εκτύπωση Τιμής στην Σειριακή
  Serial.println(pinValue2); //
}

//Διεργασία Blynk για Διάβασμα δεδομένων του εικονικού Pin V3
BLYNK_WRITE(V3)
{
  pinValue3 = param.asInt(); //Μετατροπή και αποθήκευση της εισερχόμενης τιμής
  του V3 ως Ακέραιος
  Serial.print("V2 slider value is: "); // Εκτύπωση Τιμής στην Σειριακή
  Serial.println(pinValue3); //
}

//Διεργασία Blynk για Διάβασμα δεδομένων του εικονικού Pin V4
BLYNK_WRITE(V4)
{
  pinValue4 = param.asInt(); //Μετατροπή και αποθήκευση της εισερχόμενης τιμής
  του V4 ως Ακέραιος
}

```

```
Serial.print("V4 left-right value is: "); // Εκτύπωση Τιμής στην Σειριακή
Serial.println(pinValue4);           //
}

//Διεργασία Blynk για Διάβασμα δεδομένων του εικονικού Pin V5
BLYNK_WRITE(V5)
{
  pinValue5 = param.asInt(); //Μετατροπή και αποθήκευση της εισερχόμενης τιμής
του V5 ως Ακέραιος
  Serial.print("V5 forward-backward value is: "); // Εκτύπωση Τιμής στην Σειριακή
  Serial.println(pinValue5);           //
}
```





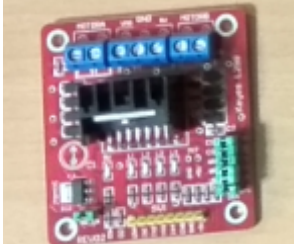
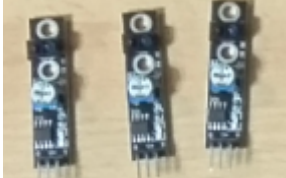

ΚΕΦΑΛΑΙΟ 4ο


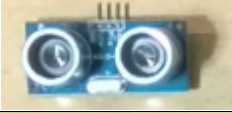

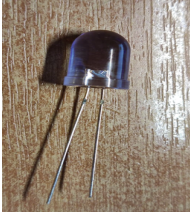


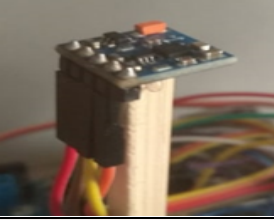


4.1 ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ





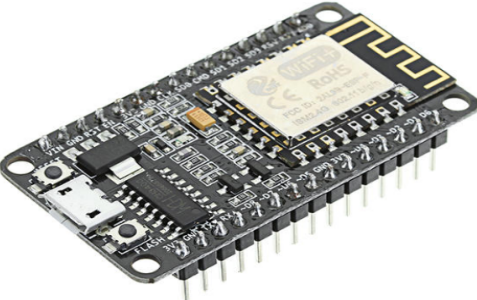
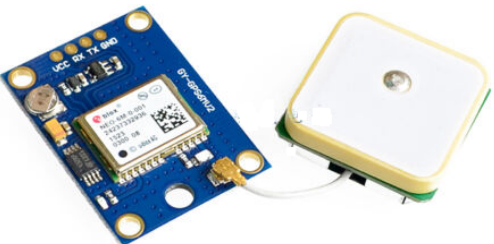
4.1.1 Υλικά

Για την δημιουργία του τετράτροχου Arduino οχήματος χρειάστηκαν τα παρακάτω υλικά:

Πίνακας 24.1 Υλικά για την Εργασία

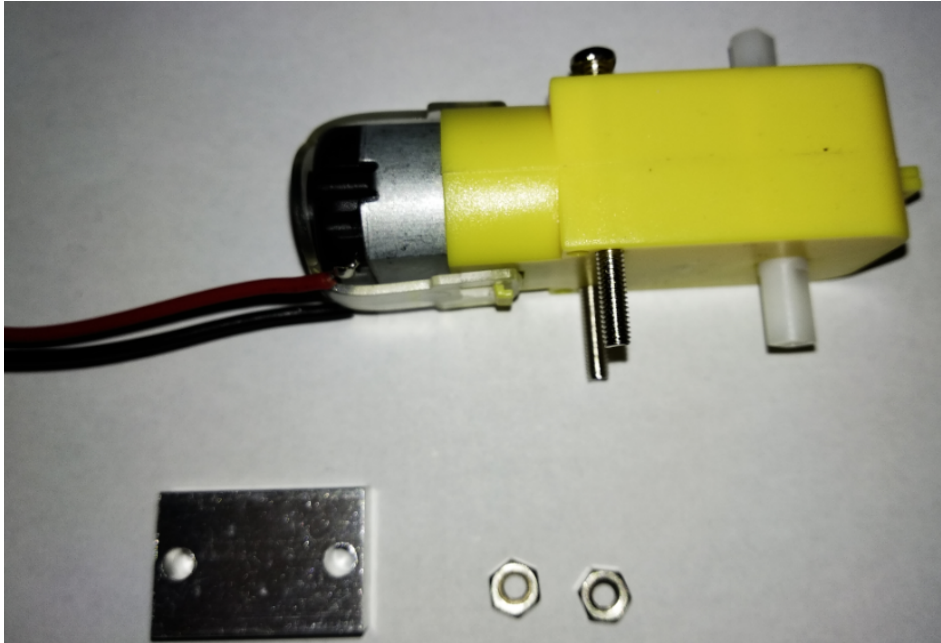
ΟΝΟΜΑ ΥΛΙΚΟΥ	ΠΟΣΟΤΗΤΑ	ΕΙΚΟΝΑ
Arduino UNO R3	1	
Arduino Sensor Shield v5.0	1	
μίνι ράστερ με αυτοκόλλητη πλάτη	1	
καλώδιο USB typeA to typeB - δεδομένων/φόρτισης	1	
Motor Controller Shield - Dual H-Bridge (L298)	1	
Triple IR Line Tracking Module	1	

Διάφορα (παξιμάδια, βίδες κλπ.)	10+	
Ultrasonic Range Finder Module	1	
Σερβοκινητήρας (9G)	1	
Led διάφανο 10mm λευκού χρώματος	1	
Led diffused 10mm κόκκινου χρώματος	1	
Breadboard Power Supply Module 3.3V/5v	1	
Πυξίδα QMC5883L	1	
Βάση για τον Σερβοκινητήρα	1	
καλώδια 10cm DuPont Female to Female Jumper Wire	20-30	

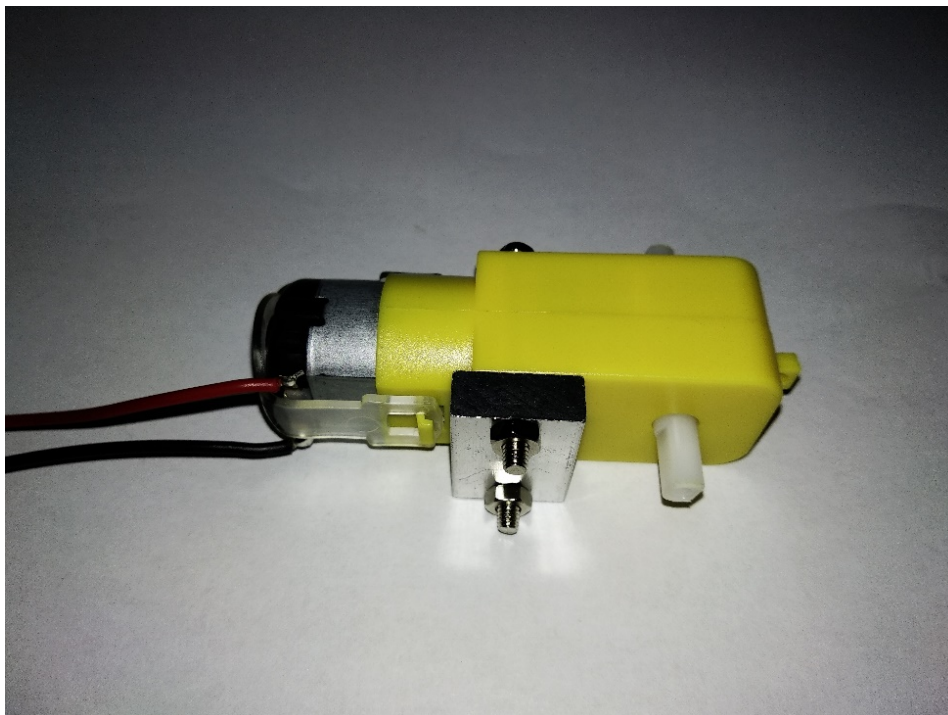
<p>Σασί αυτοκινήτου – βάση για το Arduino</p>	<p>2</p>	
<p>Μοτέρ</p>	<p>4</p>	
<p>Λάστιχα</p>	<p>4</p>	
<p>12.800mA 12V Μπαταρία</p>	<p>1</p>	
<p>ESP8266 ESP-12F WIFI Development Board</p>	<p>1</p>	
<p>GPS NEO-6M Module</p>	<p>1</p>	

4.1.2 Συναρμολόγηση του Τετράτροχου Arduino οχήματος

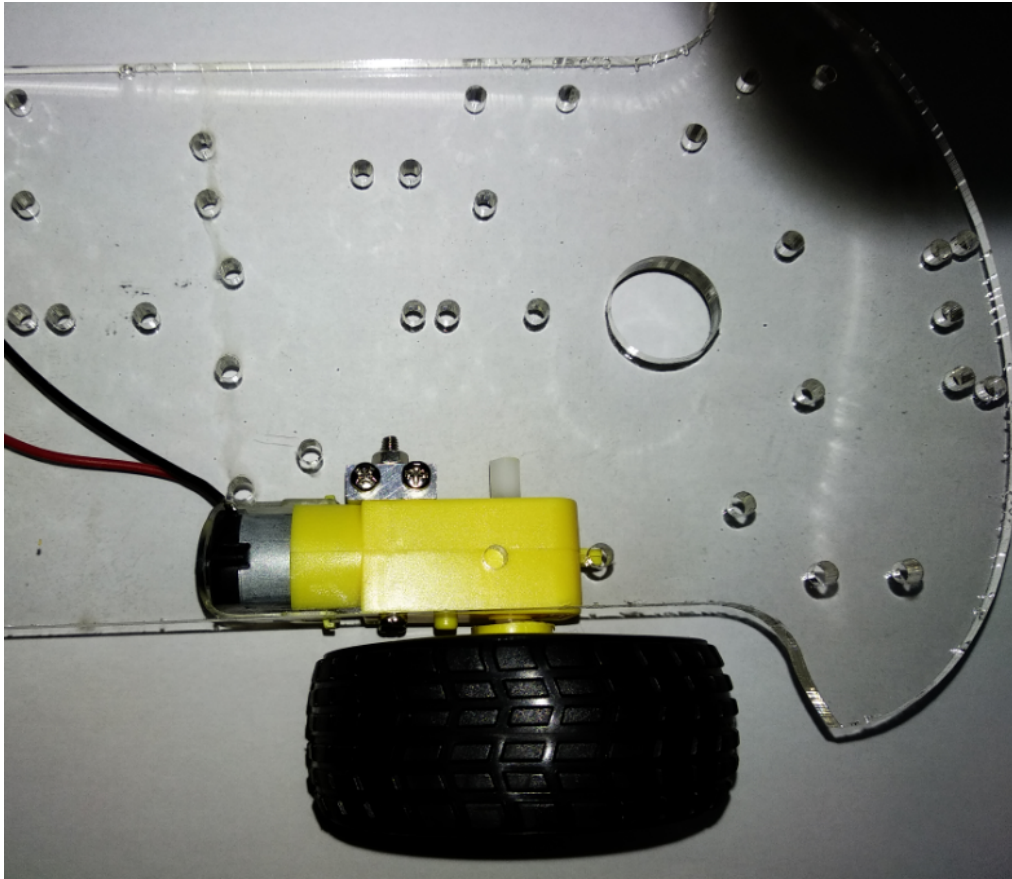
Αρχικά, βιδώθηκαν και τα 4 μοτέρ επάνω στο κάτω σασί και έπειτα τοποθετήθηκαν και οι 4 ρόδες, μία σε κάθε μοτέρ ώστε να δημιουργηθεί η αρχική βάση του τετράτροχου οχήματος.



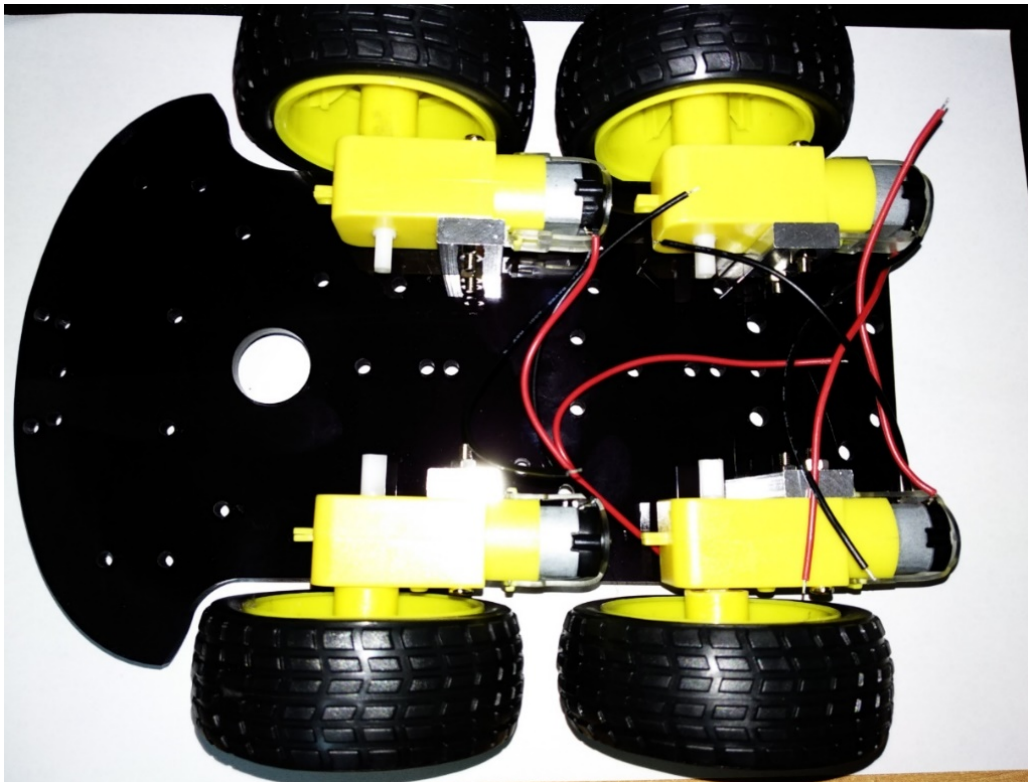
Εικόνα 24 Εξάρτημα μότορας



Εικόνα 25 Εξάρτημα μότορας

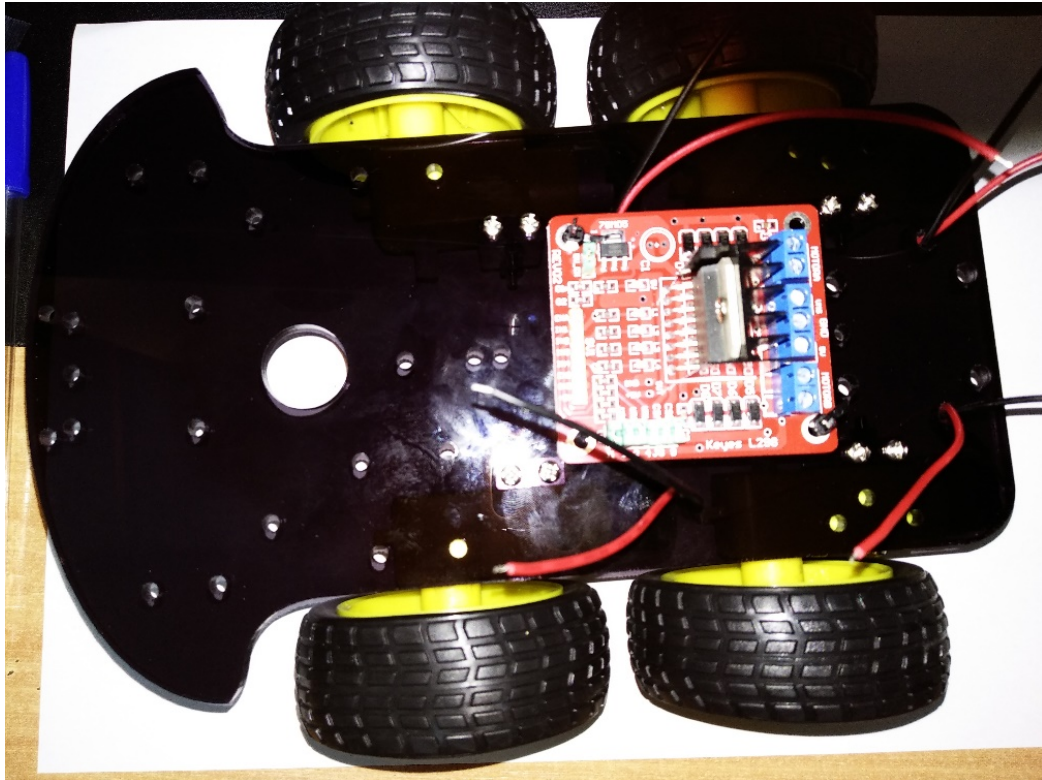


Εικόνα 26 Εξαρτήματα μώτορας-σασί

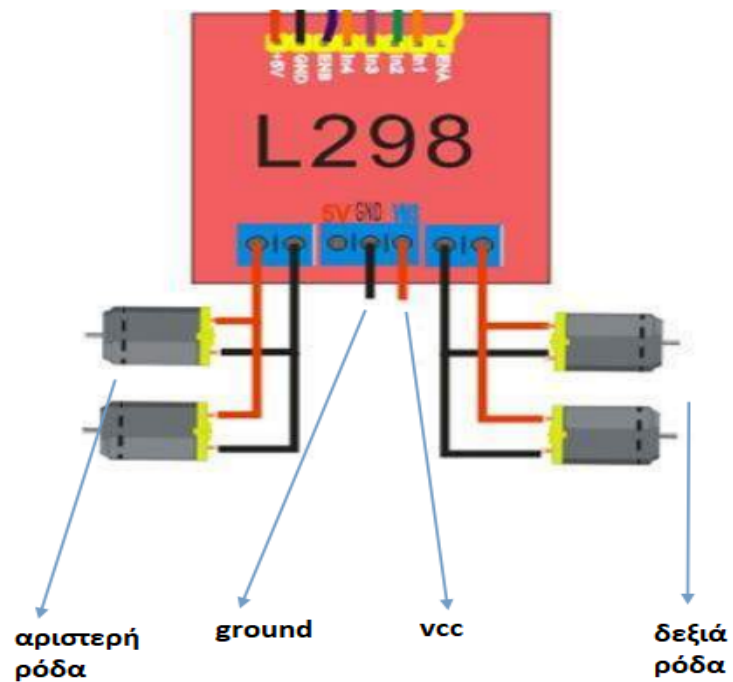


Εικόνα 27 Εξαρτήματα μώτορες-σασί

Έπειτα, περάστηκαν οι απαραίτητες βίδες στο σασί στα κατάλληλα σημεία για να στερεωθεί το L298 motor driver. [35] [19] [36]

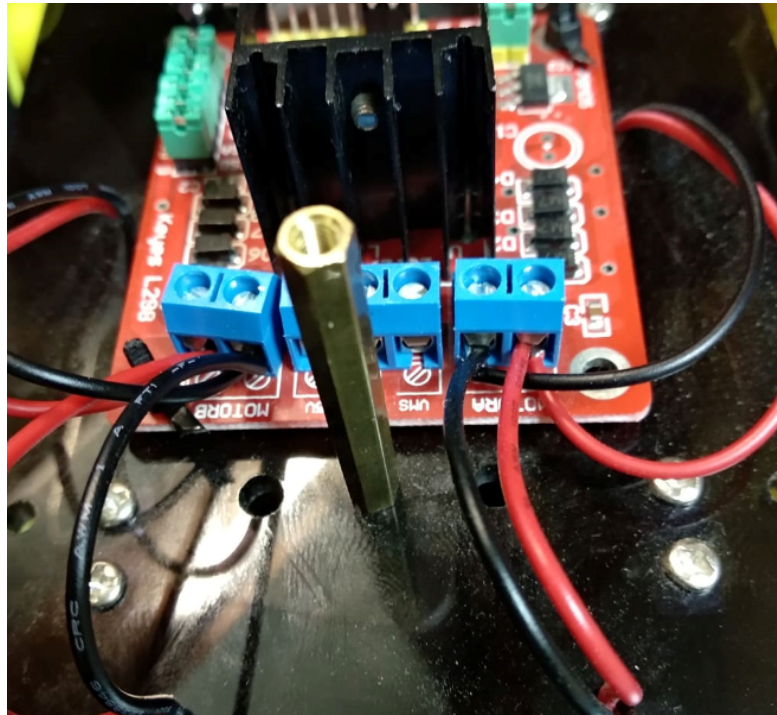


Εικόνα 28 Εξαρτήματα L298



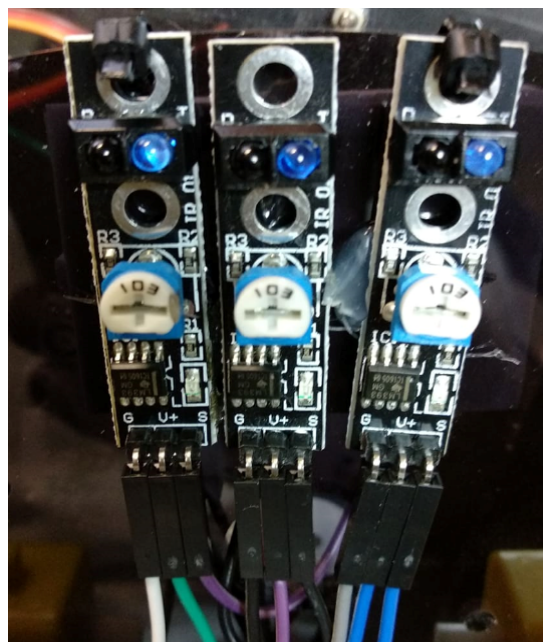
Εικόνα 29 L298

Το μπροστά αριστερά motor με το πίσω αριστερά motor συνδέθηκαν μαζί στο L298 στην κλέμα motor A και το μπροστά δεξιά motor με το πίσω δεξιά motor μαζί στο L298 στην κλέμα motor B.



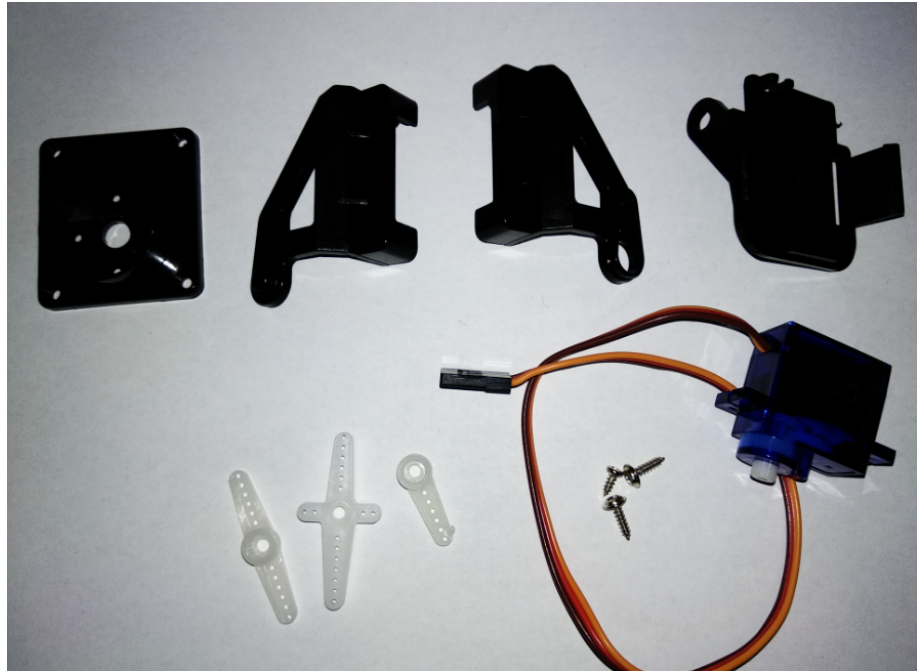
Εικόνα 30 Εξαρτήματα σασί-L298

Μετά, έγινε τοποθέτηση των 3 IR Line Tracking Module στο μπροστά κάτω μέρος στο σασί.



Εικόνα 31 Εξαρτήματα IR TM

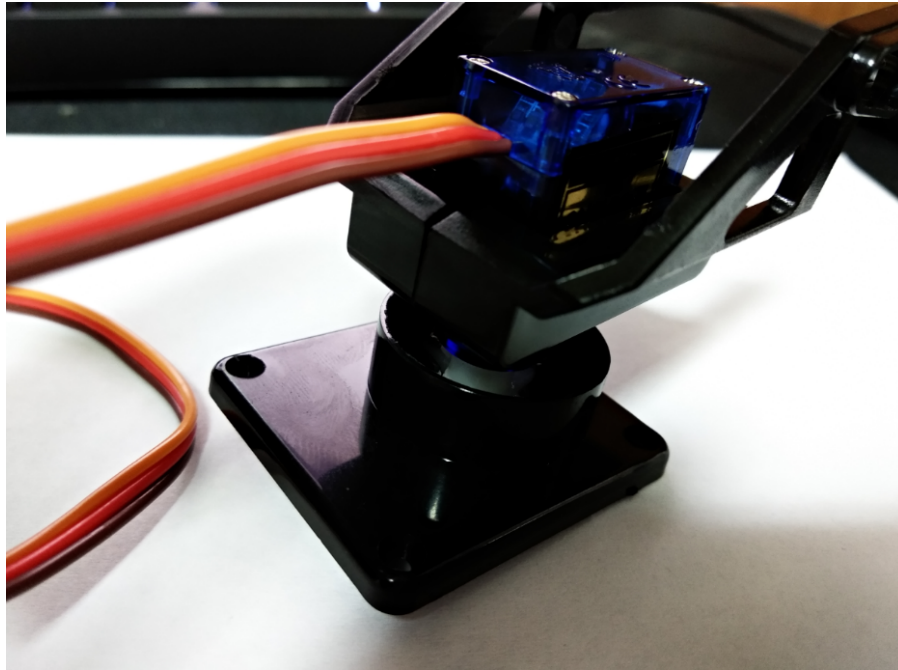
Έπειτα, συναρμολογήθηκε η βάση για τον σερβοκινητήρα και μετά τοποθετήθηκε στην βάση του. [35] [19] [36]



Εικόνα 32 Εξαρτήματα

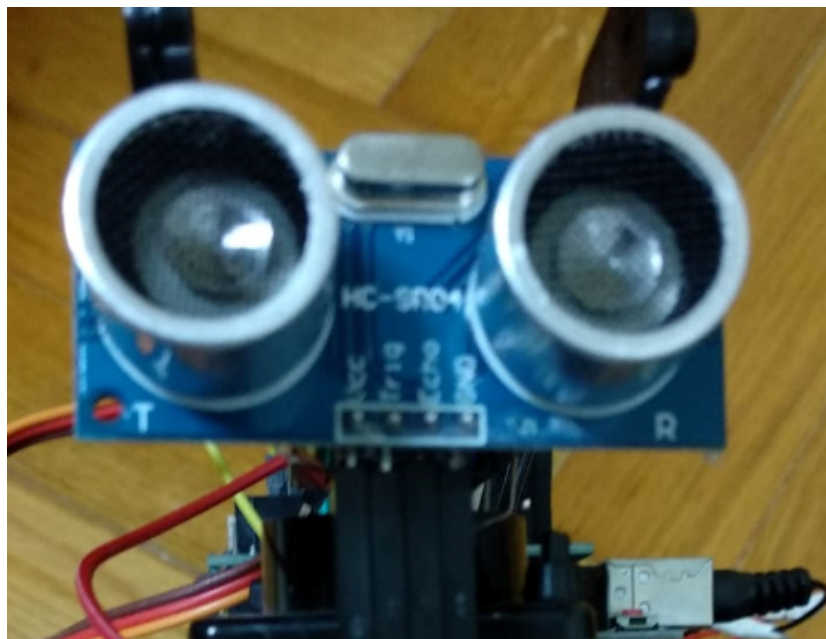


Εικόνα 33 Εξαρτήματα



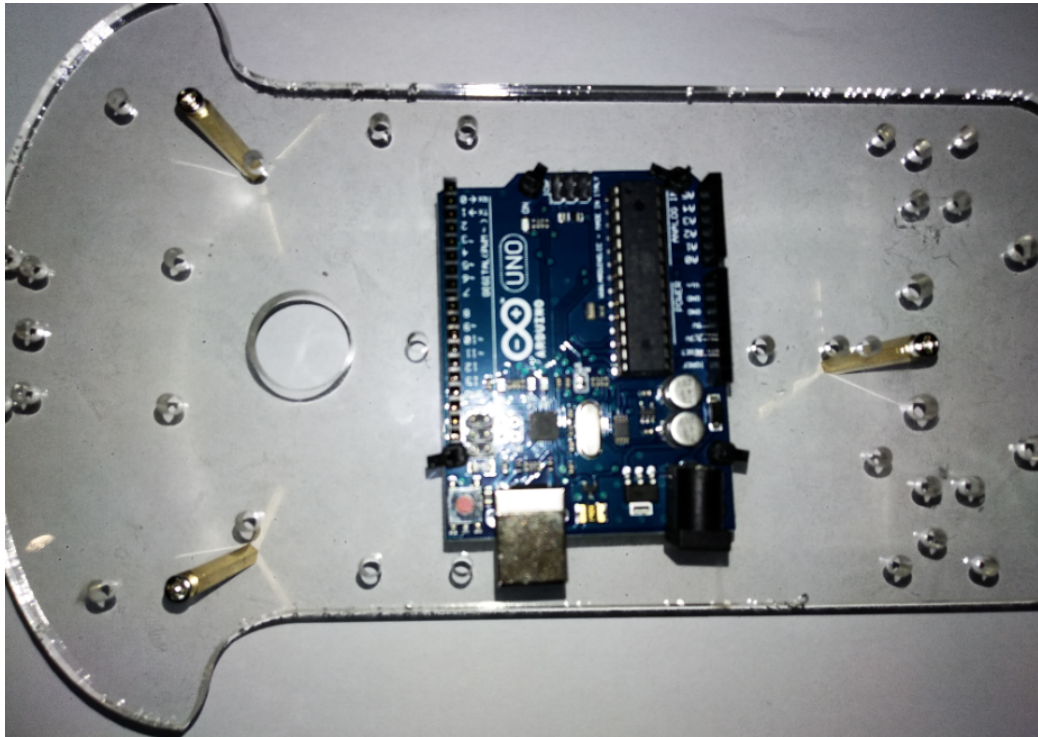
Εικόνα 34 Εξαρτήματα Servo

Μετά, στερεώθηκε πάνω στον σερβοκινητήρα ο Ultrasonic Range Finder Module αισθητήρας και αυτό το σύνολο τοποθετήθηκε στο επάνω σασί. [35] [19] [36]



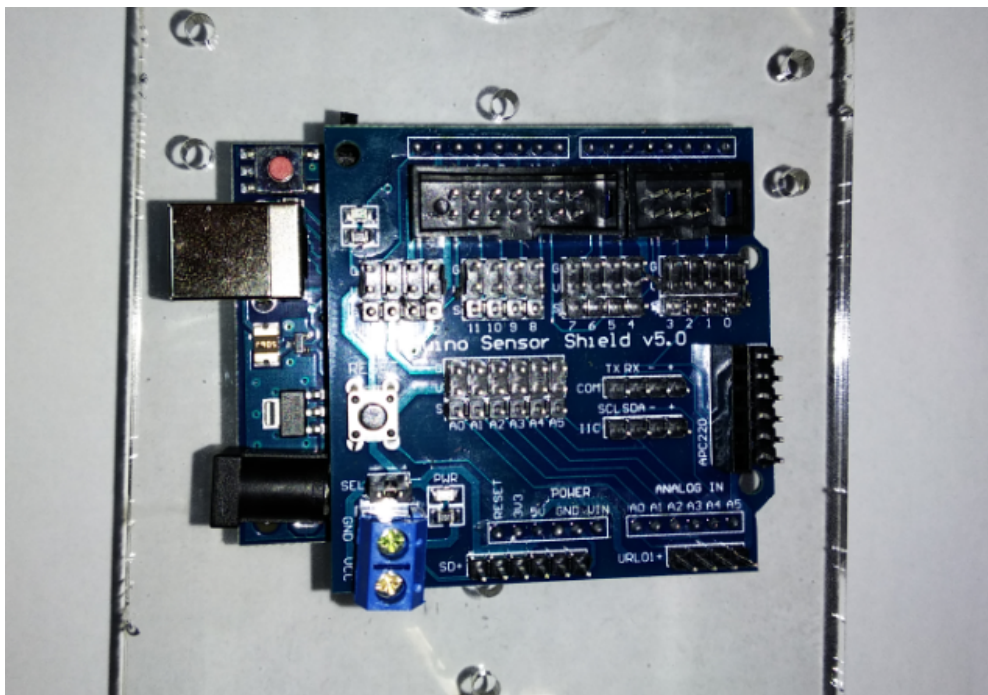
Εικόνα 35 Εξαρτήματα ultra-sonic sensor

Στην συνέχεια, το Arduino Uno R3 μπήκε στο επάνω μέρος του σασί και βιδώθηκαν οι 3 μεγάλοι αποστάτες για να ενωθούν τα σασί μεταξύ τους. [19]



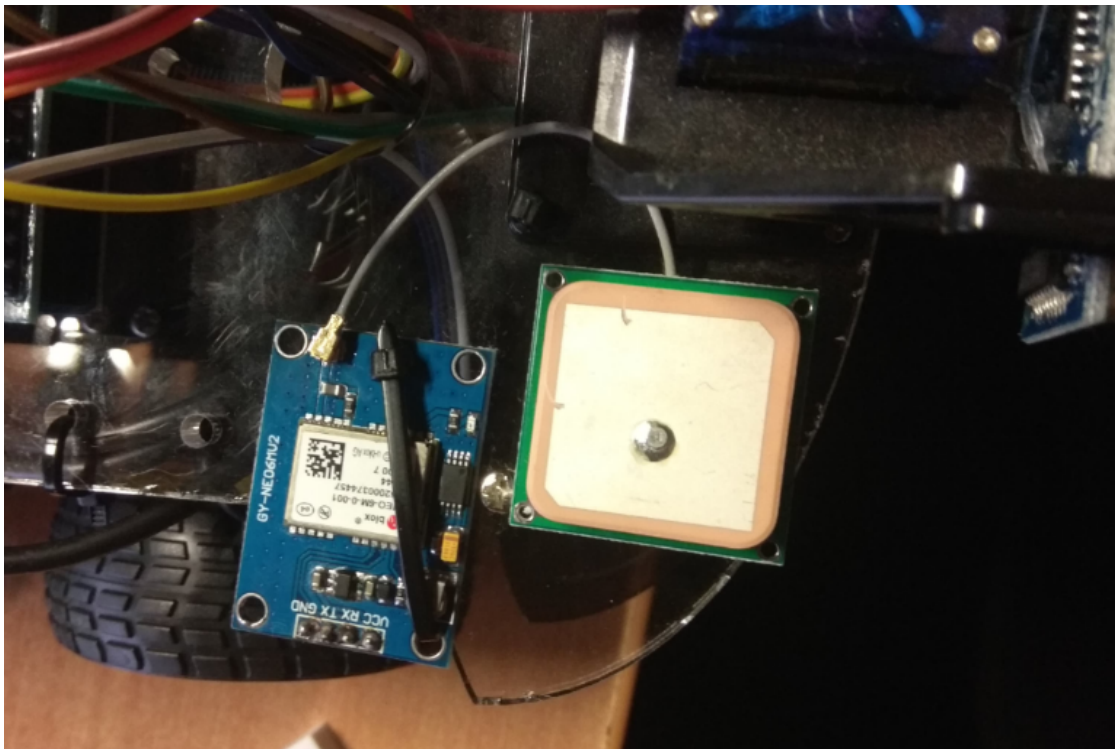
Εικόνα 36 Εξαρτήματα Arduino

Έπειτα, το Arduino Sensor Shield v5.0 ενσωματώθηκε επάνω στο Arduino.



Εικόνα 37 Εξαρτήματα Arduino

Μετάπειτα, το GPS NEO-6M στερεώθηκε στο πάνω σασί καθώς επίσης και η μπαταρία και την συνδέθηκε με έναν θηλυκό connector. [21]

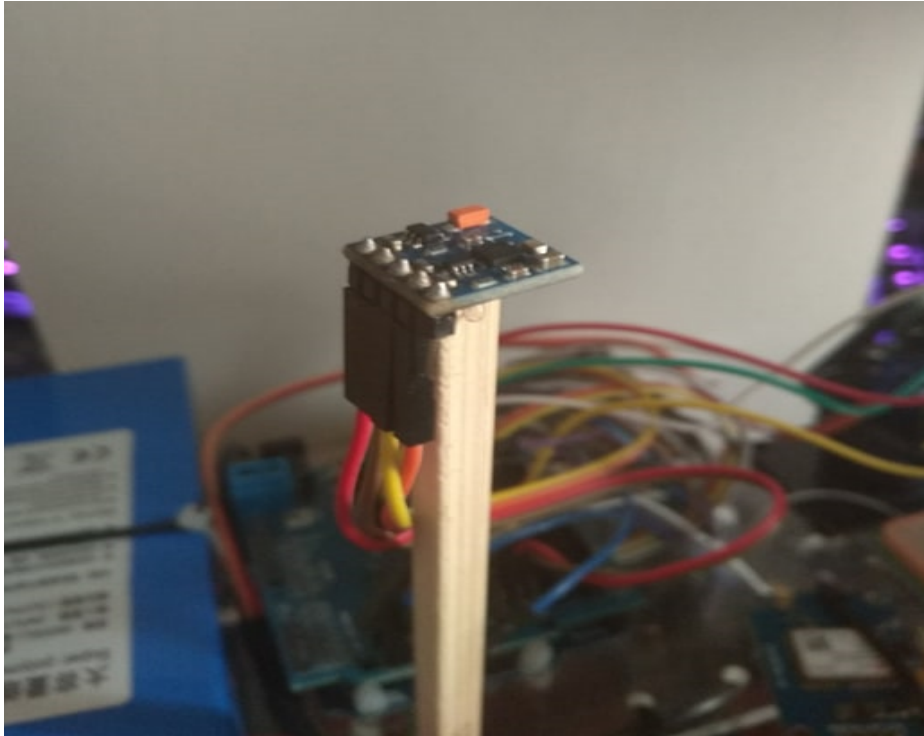


Εικόνα 38 Εξαρτήματα GPS



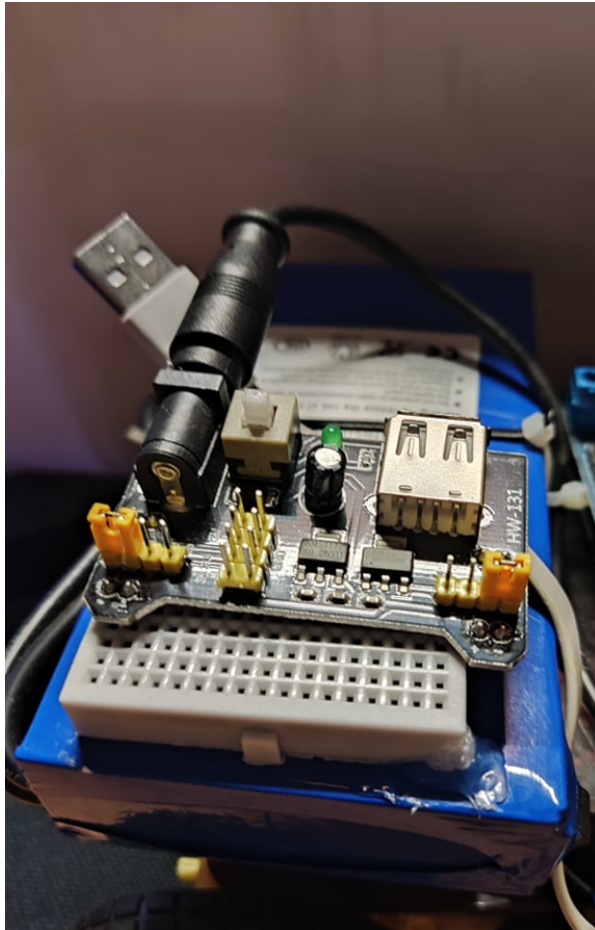
Εικόνα 39 Εξαρτήματα μπαταρία

Στην συνέχεια, έγινε τοποθέτηση της πυξίδας QMC5883L σε ένα ξύλινο στύλο στο πάνω σασί, έτσι ώστε να είναι πιο ψηλά και μακριά από τα υπόλοιπα ηλεκτρονικά μέρη του οχήματος λόγω του ότι είναι ευαίσθητο στις παρεμβολές. Επίσης, τοποθετήθηκε με τον άξονα Y να δείχνει μπροστά.

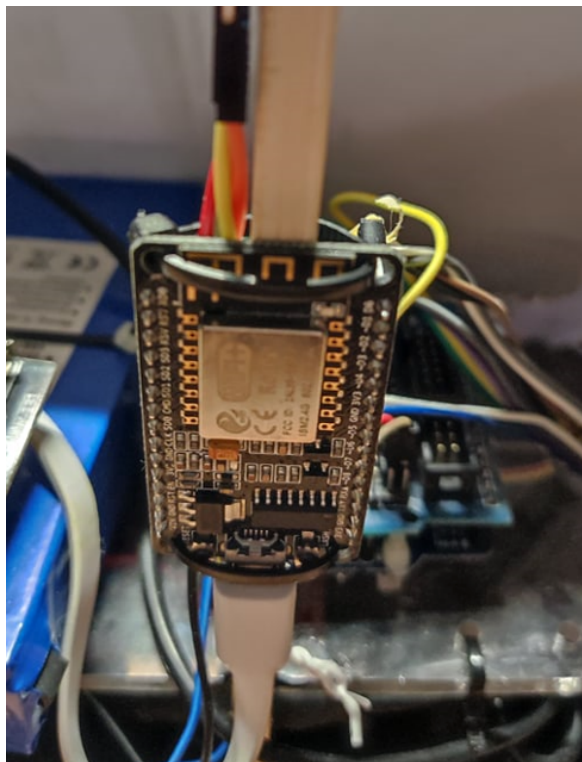


Εικόνα 40 Εξαρτήματα Compass

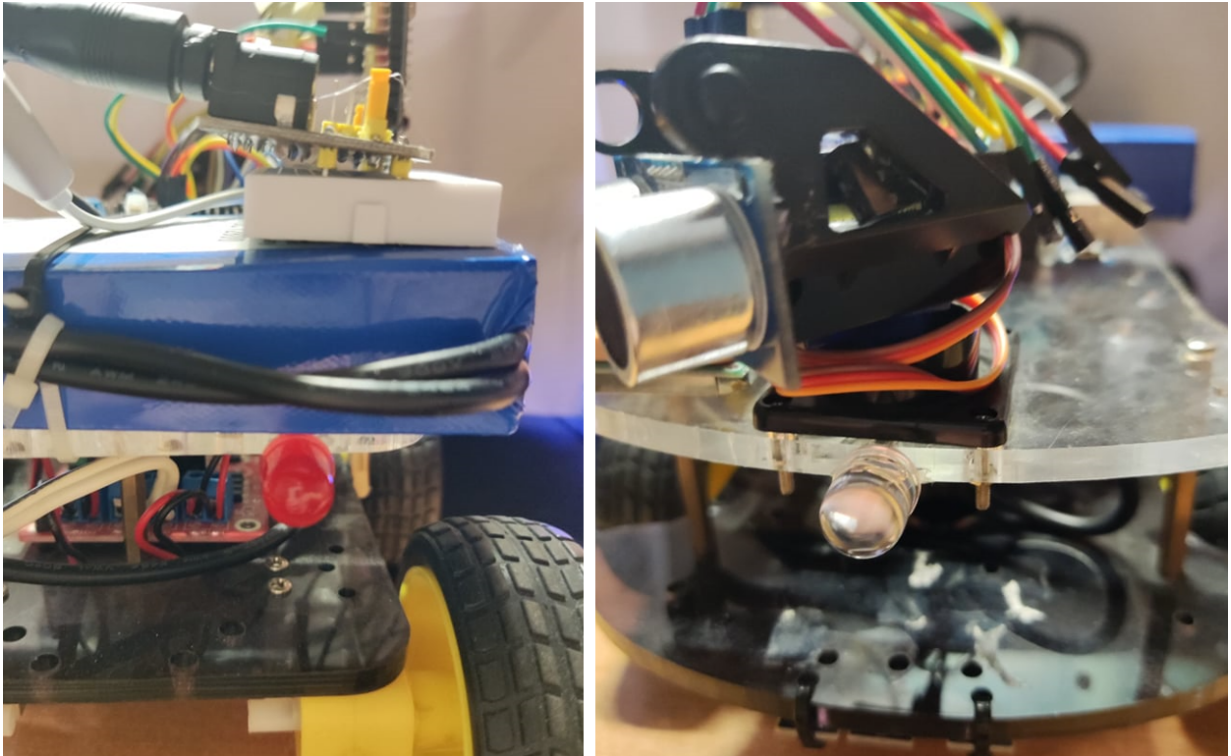
Έπειτα, κολλήθηκε ο NodeMCU ESP8266 WIFI μικροελεγκτής στο πλάγιο μέρος από το ξυλάκι που χρησιμοποιήθηκε για να κολληθεί στην κορυφή του η πυξίδα QMC5883L. Αυτό έγινε για λόγους χωρητικότητας και διευκόλυνσης στην αλλαγή καλωδίων που θα χρειαστεί να πραγματοποιηθούν κατά την εναλλαγή των δύο λειτουργιών που κάνει το όχημα (αυτόνομο με χρήση GPS με το Arduino UNO και τηλεκατευθυνόμενο με χρήση Wi-fi με το NodeMCU). Στην συνέχεια, στερεώθηκε με ζεστή σιλικόνη το Breadboard Power πάνω στην μπαταρία επίσης για λόγους χωρητικότητας, το οποίο θα μας χρειαστεί για την τροφοδοσία του NodeMCU. Τέλος, προστέθηκαν 2 LED φώτα 10mm, ένα λευκό και ένα κόκκινο. Το λευκό το τοποθετήθηκε στο μπροστινό μέρος του οχήματος όπου θα λειτουργεί σαν φως πορείας σε περίπτωση νύχτας για παράδειγμα και το κόκκινου χρώματος LED τοποθετήθηκε στο πίσω μέρος του οχήματος σαν φως όπισθεν. [22] [29] [37] [38]



Εικόνα 41 Breadboard power supply module



Εικόνα 42 NodeMCU module



Εικόνα 43 white and red LED

4.1.3 Συνδεσμολογία του τετράτροχου Arduino οχήματος

○ TRACKING MODULE

Σύνδεση του κάθε tracking module ακροδέκτη G, στο ground του Arduino
 Όπου V+ στο 5v του Arduino και
 Όπου S στους ακροδέκτες 2, 4 και 7 του Arduino.

○ SERVO

Σύνδεση του καλωδίου Data στον ακροδέκτη 3 του Arduino
 Καλώδιο Ground στο Ground του Arduino και
 Καλώδιο Τροφοδοσίας στα 5V του Arduino .

○ ULTRA SONIC ΑΙΣΘΗΤΗΡΑ

Σύνδεση του Vcc του αισθητήρα με το 5V του Arduino
 Σύνδεση του Ground του αισθητήρα με το Ground του Arduino
 Σύνδεση του Trig του αισθητήρα με τον αναλογικό ακροδέκτη A0 του Arduino
 Σύνδεση του echo του αισθητήρα με τον αναλογικό ακροδέκτη A1 του Arduino

- L298

- Για το αυτόνομο όχημα

- Σύνδεση του inr1 του L298 με τον ακροδέκτη 12 του Arduino

- Σύνδεση του inr2 του L298 με τον ακροδέκτη 13 του Arduino

- Σύνδεση του inr3 του L298 με τον ακροδέκτη 5 του Arduino

- Σύνδεση του inr4 του L298 με τον ακροδέκτη 6 του Arduino

- Σύνδεση του ENABLE A του L298 με τον ακροδέκτη 10 του Arduino

- Σύνδεση του ENABLE B του L298 με τον ακροδέκτη 11 του Arduino

- Για το τηλεκατευθυνόμενο

- Σύνδεση του inr1 του L298 με τον ακροδέκτη D3 του NodeMCU

- Σύνδεση του inr2 του L298 με τον ακροδέκτη D4 του NodeMCU

- Σύνδεση του inr3 του L298 με τον ακροδέκτη D5 του NodeMCU

- Σύνδεση του inr4 του L298 με τον ακροδέκτη D6 του NodeMCU

- Σύνδεση του ENABLE A του L298 με τον ακροδέκτη D1 του NodeMCU

- Σύνδεση του ENABLE B του L298 με τον ακροδέκτη D2 του NodeMCU

- ΜΠΑΤΑΡΙΑ

- Για το αυτόνομο όχημα

- Σύνδεση της μπαταρίας με ένα splitter στο Arduino και στο L298

- Για το τηλεκατευθυνόμενο

- Σύνδεση της μπαταρίας με ένα splitter στο Breadboard power και στο L298

- GPS

- Σύνδεση του Vcc του GPS στα 3.3V του Arduino

- Σύνδεση του Ground του GPS στο Ground του Arduino

- Σύνδεση του RX του GPS στο pin 8 του Arduino

- Σύνδεση του TX του GPS στο pin 9 του Arduino

- ΠΥΞΙΔΑ

- Σύνδεση του Vcc της πυξίδας με το 5V του Arduino

- Σύνδεση του Ground της πυξίδας με το Ground του Arduino

- Σύνδεση του SDA της πυξίδας με τον αναλογικό ακροδέκτη A4 του Arduino

- Σύνδεση του SCL της πυξίδας με τον αναλογικό ακροδέκτη A5 του Arduino

- Breadboard Power Supply Module

- Σύνδεση με ένα micro-USB καλώδιο για να τροφοδοτηθεί το NodeMCU

- LED λευκό

- Σύνδεση της ανόδου του LED στο D8 του NodeMCU

- Σύνδεση της καθόδου του LED στο Ground του NodeMCU

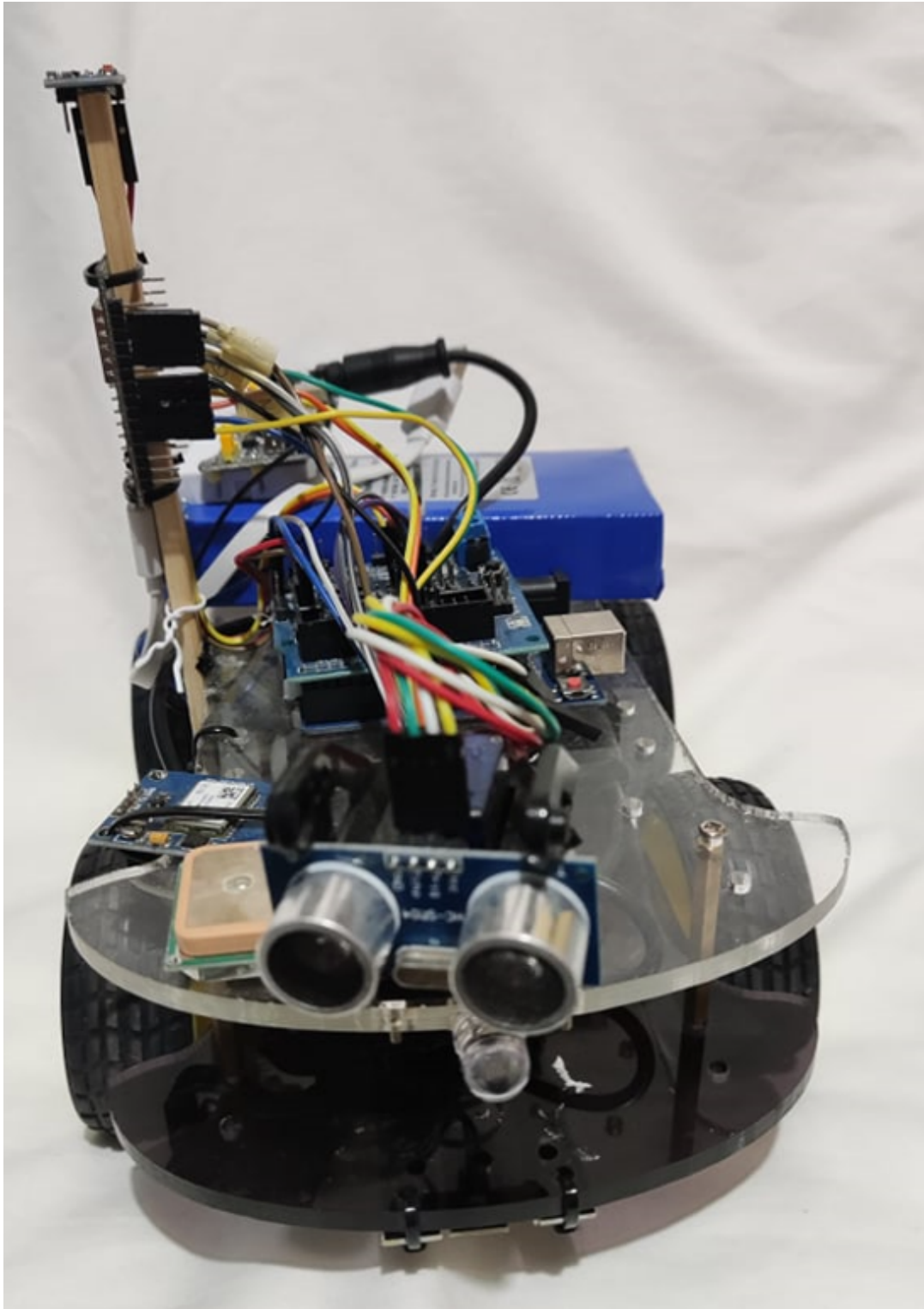
- LED κόκκινο

- Σύνδεση της ανόδου του Led στο D7 του NodeMCU

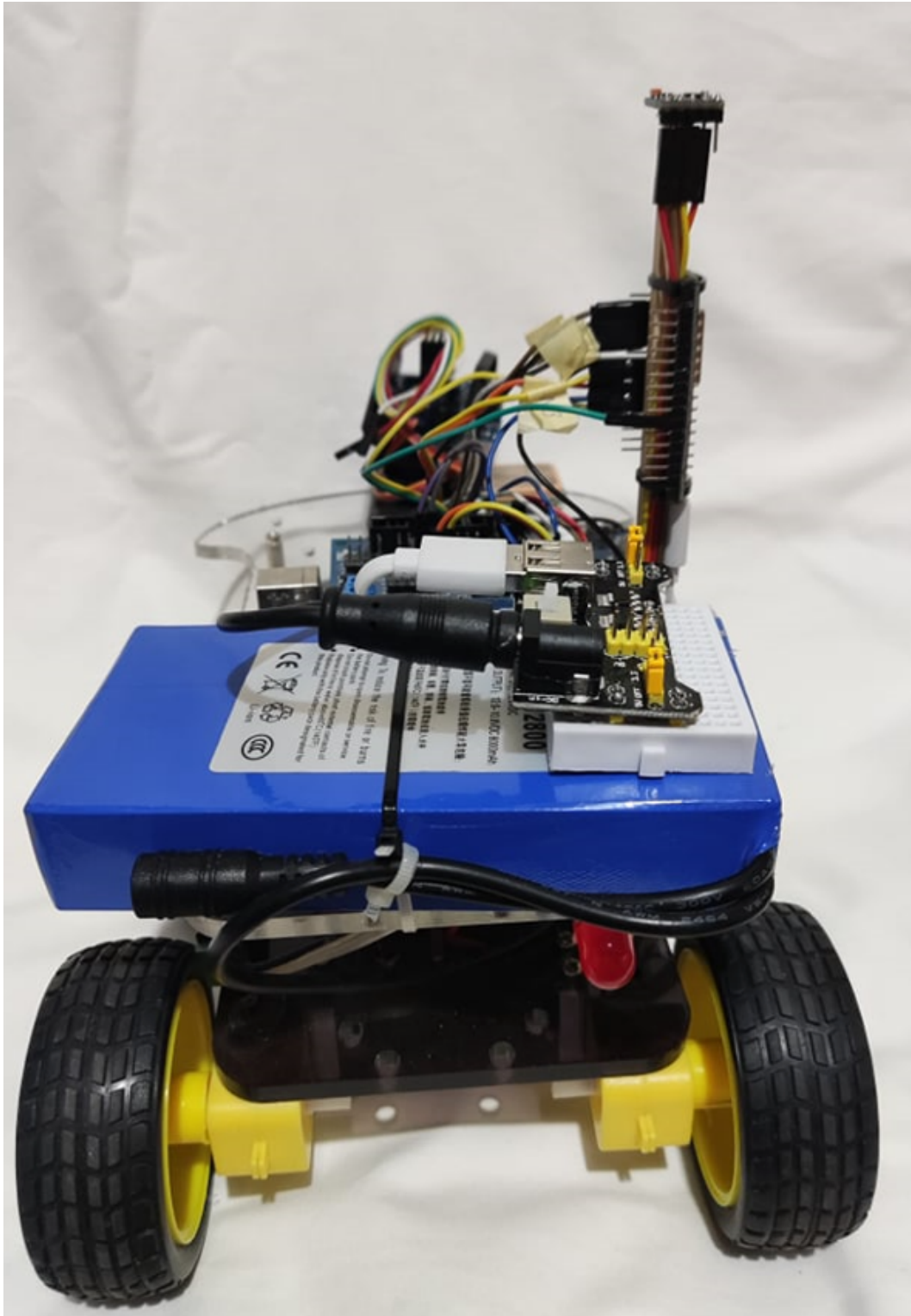
- Σύνδεση της καθόδου του Led στο Ground του NodeMCU

4.1.4 Η τελική εικόνα του οχήματος

Αφού ολοκληρώθηκαν η παραπάνω συναρμολόγηση και η συνδεσμολογία του οχήματος με την χρήση των απαραίτητων υλικών που προαναφέρθηκαν, το όχημα κατέληξε στην τελική του μορφή, όπου πλέον με τον κώδικα που εκτελεί το όχημα μπορεί να κάνει τις δύο απαιτούμενες λειτουργίες που αναφέρθηκαν. Δηλαδή, την αυτόνομη πορεία του σε συντεταγμένες που θα του δοθούν από τον χρήστη και με την βοήθεια του GPS module, της πυξίδας και των υπολογισμών μέσω του Arduino Uno να φτάσει στον προορισμό. Επίσης, την τηλεκατευθυνόμενη λειτουργία που με την βοήθεια του NodeMCU, της πλατφόρμας Blynk και του κώδικα να κινείται από τον χρήστη μέσω της εφαρμογής. Παρακάτω, φαίνεται το αποτέλεσμα, δηλαδή η τελική εμφάνιση του οχήματος.



Εικόνα 44 το όχημα μας (μπροστά)



Εικόνα 45 το όχημα μας (πίσω)

ΚΕΦΑΛΑΙΟ 5ο

5.1 ΑΠΟΤΕΛΕΣΜΑΤΑ – ΠΡΟΒΛΗΜΑΤΑ ΠΟΥ ΑΝΤΙΜΕΤΩΠΙΣΑΜΕ

5.1.1 Αποτελέσματα

Για να καταφέρει το όχημα να κατευθυνθεί από τις τωρινές του συντεταγμένες A, στις συντεταγμένες B που του δώσαμε, με στόχο την λιγότερη απόκλιση του σημείου B (τερματισμός) και να κάνει όσο δυνατόν λιγότερες παύσεις όσο είναι σε κίνηση, πήραμε πολλά δείγματα από διαφορετικές τοποθεσίες κυρίως λόγω του εδάφους (π.χ. σε κάποιες τοποθεσίες υπήρχε είτε ομαλή είτε ανώμαλη άσφαλτος και διασκορπισμένες μικρές πέτρες οπότε υπήρχε μεγαλύτερη τριβή οπότε δυσκολία στην κίνηση των τροχών με αποτέλεσμα να του αλλάζει την πορεία, είτε τοποθεσίες με πιο λείο έδαφος που έκαναν την κίνηση του οχήματος πιο ομαλή και εύκολη). Έπειτα, κάναμε σε κάθε προσπάθεια αλλαγές σε συγκεκριμένα σημεία του κώδικα για να βρούμε την βέλτιστη δυνατή μέτρηση-διαδρομή του οχήματος Arduino μας. Οι αλλαγές που κάναμε ήταν οι εξής :

- `if(dist_calc<2){ //Ορισμός Απόκλισης σε μέτρα από τον προορισμό
done(); //Τερματισμός αν είμαστε στις συντεταγμένες-στόχο`

Η απόκλιση σε μέτρα στον τερματισμό της διαδρομής. Στις μετρήσεις που κάναμε το βάζαμε σε απόκλιση έως 1 ή 2 μέτρα.

- `if(dist_calc<2){ //Αν η απόσταση από τον στόχο είναι μικρότερη των 2
μέτρων τότε:
if(-7 <= range && range <= 7){ //Ορισμός εύρους σε 7 μέτρα ανά
πλευρά(αριστερά-δεξιά)
turn=3; //Ορίζει turn=3 το οποίο σημαίνει "ευθεία"
}
}else{
if(-4 <= range && range <= 4){ //Ορισμός εύρους σε 4 μέτρα ανά
πλευρά(αριστερά-δεξιά)
turn=3; //Ορίζει turn=3 το οποίο σημαίνει "ευθεία"
}
}`

Ανάλογα με την απόσταση από τον στόχο (2 μέτρα), ορίζαμε ένα νεκρό σημείο σε μοίρες (δεξιά – αριστερά), ώστε να βελτιωθεί η πορεία του οχήματος. Π.χ. αν είναι εντός του `disc_calc` το εύρος πάει στις 14 μοίρες (7 δεξιά 7 αριστερά), αλλιώς πάει στις 8 μοίρες (4 δεξιά 4 αριστερά) .

- `while (millis() - start < 250)`

Για λήψη δεδομένων από το GPS ανά ¼ του δευτερολέπτου, ώστε και να μην έχει αλλάξει η πορεία του οχήματος προς την λάθος κατεύθυνση αλλά και για να μην υπάρχουν πολλές παύσεις.

Παρακάτω στον πίνακα βάλουμε κάποιες από τις μετρήσεις που κάναμε στο όχημα στον χώρο της σχολής, σε έδαφος ανώμαλο με άσφαλο σύμφωνα με τις αλλαγές στον κώδικα που εξηγήσαμε παραπάνω.

Πίνακας 25.1 Μετρήσεις Οχήματος σε ανώμαλο έδαφος

dist_calc	If dist_calc	If not dist_calc	Χρόνος ανανέωσης	Σταματήματα	Απόκλιση στον κώδικα	Απόκλιση από τον στόχο
2	8	4	250	11	1	1.5
2	7	4	500	14	1	2.5
2	7	3	500	14	1	2
2	7	3	800	18	1	2.5
2	7	3	50	15	1	2
2	10	5	250	7	1	1.5
2	10	5	500	5	1	1
2	20	8	500	5	1	1.5

Αρχικά παρατηρούμε ότι εξαιτίας του εδάφους (άσφαλος), το όχημα έκανε πολλά σταματήματα κυρίως λόγω της αλλαγής πορείας του λόγω μικρών πετρών και της τριβής των λάστιχων πάνω στο ανώμαλο έδαφος. Μετά βλέπουμε ότι μεγαλώνοντας το εύρος μοιρών όταν έφτανε στα 2 μέτρα απόσταση από τον τερματισμό είχαμε λιγότερα σταματήματα. Η πρασινισμένη γραμμή δηλώνει την καλύτερη μέτρηση που έγινε.

Στον επόμενο πίνακα θα δούμε μετρήσεις σε λείο έδαφος με παρόμοιες αλλαγές στο ίδιο μέρος του κώδικα.

Πίνακας 26 Μετρήσεις Οχήματος σε ομαλό έδαφος

dist_calc	If dist_calc	If not dist_calc	Χρόνος ανανέωσης	Σταματήματα	Απόκλιση στον κώδικα	Απόκλιση από τον στόχο	Χρόνος sec
2	8	4	250	7	1	1	49
2	7	4	500	9	1	1.5	48
2	7	3	500	7	1	0.5	47
2	7	3	800	6	1	1	43
2	7	3	50	5	1	1.5	44
2	7	4	250	3	1	1	44
2	10	5	250	7	1	1	52
2	10	5	500	4	1	1	44
2	20	8	500	6	1	1	46

Εδώ παρατηρούμε ότι λόγω του εδάφους (λείο), το όχημα έκανε λιγότερα σταματήματα από τις προηγούμενες μετρήσεις και το αποτέλεσμα ήταν σαφώς βελτιωμένο και πιο ικανοποιητικό. Η πρασινισμένη γραμμή και εδώ δηλώνει την καλύτερη μέτρηση και πορεία του οχήματος τόσο σε χρόνο όσο και σε σταματήματα και αυτήν τη μέτρηση με αυτά τα νούμερα κρατήσαμε.

Όσον αφορά την τηλεκατευθυνόμενη λειτουργία, αφού επιτύχαμε την επικοινωνία μεταξύ της εφαρμογής Blynk στο smartphone με το NodeMCU στο όχημα, δημιουργήσαμε το joystick με το οποίο θα ελέγχουμε οκτώ διαφορετικές κινήσεις στο όχημα και έπειτα να φτιάξουμε έναν αυξομειωτή ταχύτητας. Παρατηρήσαμε μία μικρή καθυστέρηση (μικρότερες του δευτερολέπτου) στις εντολές που δίναμε στο όχημα από το κινητό οι οποίες γίνονται κυρίως εμφανείς όταν το όχημα κινείται στην μέγιστη του ταχύτητα όπου κυρίως για αυτές τις καθυστερήσεις ευθύνονται είτε η κακή ποιότητα του σήματος Wi-fi είτε στην εμβέλεια αυτού. Η εφαρμογή διαθέτει και μια LCD οθόνη η οποία επιτυχώς δείχνει την IP του NodeMCU και τέλος τα 2 LED φώτα όπου επιτυχώς ανάβουν μόνο εκεί που τους δίνεται η κατάλληλη εντολή.

5.1.2 Προβλήματα που αντιμετωπίσαμε

Για την εκπόνηση της διπλωματικής μας εργασίας αντιμετωπίσαμε αρκετά προβλήματα που καλεστήκαμε να λύσουμε. Αρχικά, στο υλικό κομμάτι χρειάστηκε να αντικαταστήσουμε την μπαταρία του οχήματος διότι δεν έδινε αρκετή ενέργεια ώστε να λειτουργεί σωστά το όχημα και δεν είχε αρκετή διάρκεια. Η νέα μπαταρία έλυσε αυτό το πρόβλημα αλλά επειδή ήταν μεγαλύτερη σε όγκο μας άλλαξε το κέντρο βάρους αλλά με τις κατάλληλες μετατροπές το λύσαμε. Έπειτα, κατασκευάσαμε έναν μικρό ξύλινο στύλο στο όχημα και βάλαμε εκεί την πυξίδα ώστε να είναι σε μεγαλύτερο ύψος για λιγότερες παρεμβολές. Τέλος, οι μότερες και τα λάστιχα που είχαμε δεν ήταν κατάλληλοι για ανώμαλο έδαφος όπως φάνηκε και στον πρώτο πίνακα μετρήσεων αλλά ήταν ικανοποιητικοί για ένα λείο έδαφος.

Όσον αναφορά τον κώδικα, χρειάστηκε να κάνουμε πολλές αλλαγές και προσθήκες ώστε να βελτιωθεί η πορεία του οχήματος. Ορίστηκαν «νεκρά» σημεία, στις μοίρες που κάνει έλεγχο το όχημα. Με αποτέλεσμα, όσο το όχημα πλησιάζει στις συντεταγμένες τερματισμού τόσο μεγαλώνουν οι ανοχές στις μικρές παρεκκλίσεις λόγω εδάφους ή μικρών σφαλμάτων. Για την βελτίωση της απόκλισης του οχήματος από τον τερματισμό διαπιστώθηκε ότι το GPS που χρησιμοποιήθηκε στην εργασία είχε απόκλιση της τάξεως 1,5 έως 2,5 μέτρα. Σύμφωνα με αυτό έγιναν οι κατάλληλες προσθήκες στον κώδικα, έτσι ώστε να παραλείπεται αυτή η απόκλιση με αποτέλεσμα το όχημα να παρεκκλίνει από τον στόχο 0,5-1 μέτρο. Τέλος, επιλέχθηκε κατάλληλο έδαφος για την κίνηση του οχήματος, ώστε να μην παρεκκλίνει από την πορεία του και χρειαστεί ανακατεύθυνση.

Για το τηλεκατευθυνόμενο κομμάτι της εργασίας, στην αρχή είχαμε επιλέξει το ESP8266 Module, όμως επειδή θα χρειαζόμασταν εξωτερική πλακέτα προγραμματισμού, καθώς και να αλλάζουμε συνεχώς την συνδεσμολογία του Arduino κάθε φορά που θα έπρεπε να αλλάξουμε τον κώδικα, καταλήξαμε στο NodeMCU, το οποίο είναι ένα development board όπου φέρει ενσωματωμένο προγραμματιστή και το custom NodeMCU firmware με παραπάνω δυνατότητες. Έπειτα, φάξαμε αρκετά για το ποια εφαρμογή θα χρησιμοποιήσουμε μέχρι να καταλήξουμε στο Blynk, απορρίπτοντας άλλες γνωστές και χρήσιμες εφαρμογές όπως το android studio. Καταλήξαμε στην εφαρμογή Blynk γιατί μας παρείχε πληθώρα από χρήσιμα γραφικά στοιχεία και επειδή προσφέρει διακομιστή γρήγορης

απόκρισης για Internet of Things. Μέσω της εφαρμογής Blynk και των βιβλιοθηκών της για μικροελεγκτές, ο χρήστης μπορεί να χειρίζεται το όχημα με πολύ μικρή καθυστέρηση. Πρόβλημα αντιμετωπίσαμε και στις 4 διαγώνιες κινήσεις του οχήματος με το joystick, αλλά με αρκετές τροποποιήσεις στον κώδικα μας καταφέραμε να τις υλοποιήσουμε σωστά. Τέλος, στο slider της εφαρμογής για την αλλαγή ταχύτητας δεν ορίσαμε τις τιμές από 0 έως 255, αλλά από 125 ως 255, καθώς το 125 είναι το ελάχιστο ώστε να κινείται φυσιολογικά το όχημα.

ΚΕΦΑΛΑΙΟ 6ο

6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

Για τους σκοπούς της εργασίας αυτής, δημιουργήσαμε ένα τετράτροχο όχημα Arduino, υλοποιώντας το σχεδιασμό του οχήματος τόσο στο θέμα του υλικού όσο και στο λογισμικό με σκοπό να έχει δύο λειτουργίες. Πρώτον, να γίνει αυτόνομο με σκοπό να κινηθεί στον χώρο παρέχοντας του μόνο το σημείο τερματισμού με συντεταγμένες. Το όχημα θα προσεγγίζει το σημείο προορισμού με τη μέγιστη δυνατή ακρίβεια. Δεύτερον, να έχει ο χρήστης τον πλήρη έλεγχο του οχήματος με την χρήση της εφαρμογής μέσα από την οποία το όχημα εκτελεί τηλεκατευθυνόμενη λειτουργία και μπορεί ο χρήστης να ορίσει την ταχύτητα που επιθυμεί.

Τα συμπεράσματα που προέκυψαν από την εργασία έπειτα από έρευνα και δοκιμές χωρίζονται σε δύο κομμάτια, ένα για κάθε λειτουργία. Στην πρώτη λειτουργία, για να προσεγγίζει το όχημα με ακρίβεια το σημείο προορισμού έπρεπε να κατανοηθεί όλη η λειτουργία και η τεχνολογία που περιβάλλει ένα σύστημα GPS. Επίσης έπρεπε να ληφθούν υπόψιν τα σφάλματα και οι αποκλίσεις που μπορεί να έχει η επικοινωνία με έναν δορυφόρο GPS αλλά και οι παρεκκλίσεις που μπορεί να υποστεί το όχημα από το έδαφος ή τις συνθήκες του χώρου. Για την δεύτερη λειτουργία, με στόχο το όχημα να έχει όσο τον δυνατόν λιγότερη καθυστέρηση χειρισμού μέσω του διαδικτύου, έπρεπε να εμβαθύνουμε στην λειτουργία του IoT και της φιλοσοφίας του. Ειδικότερα, έπρεπε η πλατφόρμα που θα χρησιμοποιήσουμε και η υλοποίηση του κώδικα να παρέχουν τα κατάλληλα εργαλεία. Για παράδειγμα ο διακομιστής επικοινωνίας της πλατφόρμας Blynk προσφέρει γρήγορη απόκριση, έτσι μειώνεται αρκετά η καθυστέρηση του χειρισμού μέσω της εφαρμογής με αποτέλεσμα το όχημα να πραγματοποιεί μια ομαλή πορεία. Μέσα από τις δύο αυτές διαφορετικές πτυχές της τεχνολογίας καταφέραμε να υλοποιήσουμε ένα όχημα με δύο αντίθετες λειτουργίες, αυτή της αυτονομίας και του χειρισμού μέσω διαδικτύου.

Έχουμε σκοπό να αναπτύξουμε και να αναβαθμίσουμε τις δυνατότητες του οχήματος, για παράδειγμα με την βοήθεια του ultra-sonic αισθητήρα να κάνουμε το όχημα να μπορεί μόνο του να αναγνωρίζει και να αποφεύγει εμπόδια που θα βρει στην πορεία του με σκοπό την αποφυγή σύγκρουσης. Επίσης με τους IR tracker sensors να αναγνωρίζει τις γραμμές του δρόμου. Οι επιπλέον λειτουργίες αυτές θα έχουν ως αποτέλεσμα την επίτευξη ασφαλέστερης και πιο περίπλοκης πορείας. Επίσης, μια χρήσιμη αναβάθμιση θα ήταν να δίνουμε την δυνατότητα στο τηλεκατευθυνόμενο όχημα αντί για 8 διαφορετικές κινήσεις να κινείται σε όλες τις 360 μοίρες του κύκλου του joystick καθώς και να δώσουμε περισσότερες επιλογές στις τιμές του αυξομειωτή ταχύτητας για το χρήστη.

ΚΕΦΑΛΑΙΟ 7ο

7.1 ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Arduino, «Arduino,» Arduino, [Ηλεκτρονικό]. Available: <https://www.arduino.cc/>. [Πρόσβαση Μάρτιος 2019].
- [2] Μ. Ιωάννης, Έλεγχος ρομποτικού οχήματος με Arduino Uno, Πτυχιακή Εργασία, ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ., 2017.
- [3] Ν. Π. Κ. Ε.Μ.Π, Προγραμματισμός με την C++, Αθήνα: Παπασωτηριου, 2018.
- [4] M. Silverio, «How to make a vehicle autonomous,» towardsdatascience.com, 2019. [Ηλεκτρονικό]. Available: <https://towardsdatascience.com/how-to-make-a-vehicle-autonomous-16edf164c30f>. [Πρόσβαση Σεπτέμβριος 2021].
- [5] Wikipedia, «Self Driving Car,» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Self-driving_car.
- [6] Δ. Σ. Αλατσαθιανός, Ανάπτυξη Συστημάτων με Μικροελεγκτές 8051, 2007.
- [7] Π. Η. Γιαννακόπουλος, Ηλεκτρονικά - Δίοδοι και Ενισχυτικά Στοιχεία, 2012.
- [8] G. Team, «What is GPS?,» Geotab, 2020. [Ηλεκτρονικό]. Available: <https://www.geotab.com/blog/what-is-gps/>. [Πρόσβαση Φεβρουάριος 2021].
- [9] Wikipedia, «Compass,» Wikipedia, [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/Compass>. [Πρόσβαση Απρίλιος 2021].
- [10] Wikipedia, «Wireless network interface controller,» 2021. [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Wireless_network_interface_controller. [Πρόσβαση Σεπτέμβριος 2021].
- [11] W.-F. Alliance, «Wi-Fi Alliance,» [Ηλεκτρονικό]. Available: <https://www.wi-fi.org>. [Πρόσβαση Αύγουστος 2021].
- [12] Android, «Meet Android Studio - User Guide,» Google, 2021. [Ηλεκτρονικό]. Available: <https://developer.android.com/studio/intro>. [Πρόσβαση Μάϊος 2021].
- [13] Everybodywiki, «Andromo,» Everybodywiki, Ιούλιος 2020. [Ηλεκτρονικό]. Available: <https://en.everybodywiki.com/Andromo>. [Πρόσβαση Μάϊος 2021].

- [14] Blynk, «Blynk Documentation,» Blynk, 2014. [Ηλεκτρονικό]. Available: <http://docs.blynk.cc/>. [Πρόσβαση Μάιος 2021].
- [15] S. Ranger, «Everything you need to know about the Internet of Things right now,» February 2020. [Ηλεκτρονικό]. Available: <https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now/>. [Πρόσβαση Μάιος 2021].
- [16] Admin, «Τι είναι το Internet of Things,» 2020. [Ηλεκτρονικό]. Available: <https://texnikos-ipologiston.gr/%CF%84%CE%B9-%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9-%CF%84%CE%BF-internet-of-things/>. [Πρόσβαση Σεπτέμβριος 2021].
- [17] B. Stroustrup, Design and Evolution of C++, 1994.
- [18] Wikipedia, «DC motor,» Wikipedia, [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/DC_motor. [Πρόσβαση Νοέμβριος 2020].
- [19] auselectronicsdirect, «4WD Robot Kit Instruction Manual,» [Ηλεκτρονικό]. Available: <https://www.auselectronicsdirect.com.au/assets/files/TA0135%20Instruction%20Manual%20compressed.pdf>. [Πρόσβαση Νοέμβριος 2018].
- [20] D. SAWICZ, «HOBBY SERVO FUNDAMENTALS,» [Ηλεκτρονικό]. Available: <http://www.princeton.edu/~mae412/TEXT/NTRAK2002/292-302.pdf>. [Πρόσβαση Σεπτέμβριος 2020].
- [21] randomnerdtutorials, «Guide to NEO-6M GPS Module with Arduino,» randomnerdtutorials, [Ηλεκτρονικό]. Available: <https://randomnerdtutorials.com/guide-to-neo-6m-gps-module-with-arduino/>. [Πρόσβαση Φεβρουάριος 2019].
- [22] Components101, «HMC5883L Magnetometer Module,» Components101, 2020. [Ηλεκτρονικό]. Available: https://components101.com/ics/hmc5883l-magnetometer-module?fbclid=IwAR2KtRtQ1SaneZT8A5BPk6_w6CKB4sG_SBqDr12IB_ikCQyHWU_AI6EEzFY. [Πρόσβαση Νοέμβριος 2020].
- [23] Components101, «NodeMCU ESP8266,» Components101, 2020. [Ηλεκτρονικό]. Available: <https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet>. [Πρόσβαση Μάιος 2021].
- [24] ReadTheDocs, «NodeMCU Documentation,» [Ηλεκτρονικό]. Available: <https://nodemcu.readthedocs.io>. [Πρόσβαση Νοέμβριος 2020].
- [25] electronics-tutorials.ws, «The Light Emitting Diode,» [Ηλεκτρονικό]. Available: https://www.electronics-tutorials.ws/diode/diode_8.html. [Πρόσβαση Μάιος 2021].

- [26] L. Edman, «Basics: Picking Resistors for LEDs,» evilmadscientist, 29 Αύγουστος 2012. [Ηλεκτρονικό]. Available: <https://www.evilmadscientist.com/2012/resistors-for-leds/>. [Πρόσβαση Ιούνιος 2021].
- [27] Patrickmccabe, «Fundamentals of a GPS guided vehicle,» robotshop, 2010. [Ηλεκτρονικό]. Available: https://www.robotshop.com/community/forum/t/fundamentals-of-a-gps-guided-vehicle/12955?fbclid=IwAR2-4kd86U1ZQqnlYCrpDwHL-w_RyxGgWTahaUTBoliiA5s_F_aValhOns. [Πρόσβαση Μάρτιος 2019].
- [28] lastminuteengineers, «Interface ublox NEO-6M GPS Module with Arduino,» lastminuteengineers, [Ηλεκτρονικό]. Available: https://lastminuteengineers.com/neo6m-gps-arduino-tutorial/?fbclid=IwAR2s9Z9-dZX_6DuZXwlWCUHNaTnisKS-xjyRx25F3QPgJykP8FBHQ_gXGM8. [Πρόσβαση Μάρτιος 2019].
- [29] A. T. G, «Control Arduino Uno Using ESP8266 WiFi Module and Blynk App,» Instructables, 27 Ιούλιος 2018. [Ηλεκτρονικό]. Available: <https://www.instructables.com/Control-Arduino-Uno-Using-ESP8266-WiFi-Module-and-/>. [Πρόσβαση Απρίλιος 2021].
- [30] Circuits4You, «ESP8266 Web Server AP (Access Point),» Circuits4You, 2016. [Ηλεκτρονικό]. Available: <https://circuits4you.com/2016/12/16/esp8266-web-server-ap/>. [Πρόσβαση 2021].
- [31] Last Minute ENGINEERS, «Create A Simple ESP8266 NodeMCU Web Server In Arduino IDE,» Last Minute ENGINEERS, [Ηλεκτρονικό]. Available: <https://lastminuteengineers.com/creating-esp8266-web-server-arduino-ide/>. [Πρόσβαση Μάιος 2021].
- [32] E. Fahad, «Arduino IOT Project: Nodemcu ESP8266 wifi Robot Car “L298N motor driver + Blynk + Joystick”,» electronicclinic, 2019. [Ηλεκτρονικό]. Available: <https://www.electronicclinic.com/arduino-iot-project-nodemcu-esp8266-wifi-robot-car-l298n-motor-driver-blynk-joystick/>. [Πρόσβαση Μάιος 2021].
- [33] CIRCUIT BEST, «How to Make Arduino SmartPhone controlled WiFi car Using NodeMCU Esp8266, L298N Motor Driver,» CIRCUIT BEST, 2020. [Ηλεκτρονικό]. Available: <https://circuitbest.com/arduino-smartphone-controlled-wifi-car-using-nodemcu-esp8266/>. [Πρόσβαση Μάιος 2021].
- [34] Saddam, «WiFi Controlled Robot using Arduino,» circuitdigest, 2016. [Ηλεκτρονικό]. Available: https://circuitdigest.com/microcontroller-projects/arduino-wifi-controlled-robot?fbclid=IwAR3fjOplfkWWWh9pTL_qIC-N-IWcz74GS1mZYY7wBf76m9W7C2alxE4eOhyw. [Πρόσβαση Ιανουάριος 2019].

- [35] ICStation, «instructables,» 2014. [Ηλεκτρονικό]. Available: https://www.instructables.com/Multi-Function-Automatic-Move-Smart-Car-for-Arduin/?fbclid=IwAR2SGMOLCz9X364fIXGXFNjssq1g0Nk0jqyhwSad4Ejflo_gnjshFI3eYFk. [Πρόσβαση Νοέμβριος 2018].
- [36] D. Albright, «How to Build a 4WD Arduino Robot for Beginners,» makeuseof, 28 Δεκέμβριος 2015. [Ηλεκτρονικό]. Available: https://www.makeuseof.com/tag/build-4wd-arduino-robot-beginners/?fbclid=IwAR3cAr2TH_gvbUiBBDCFSzFpDI21-xDTiwgOsc-zWCXUHVywo5OVouh3_EE. [Πρόσβαση Νοέμβριος 2018].
- [37] Alfa(Member), «Control Arduino with WiFi,» Arduino, September 2019. [Ηλεκτρονικό]. Available: <https://create.arduino.cc/projecthub/Raushancpr/control-arduino-with-wifi-9b2347>. [Πρόσβαση Μάιος 2021].
- [38] Chirag_Modi, «Internet Controlled LED Using NodeMCU,» 6 Οκτώβριος 2017. [Ηλεκτρονικό]. Available: <https://www.instructables.com/Internet-Controlled-LED-Using-NodeMCU/>. [Πρόσβαση Μάιος 2021].
- [39] Κ. Καλοβρέκτης και Ν. Κατέβας, Αισθητήρες Μέτρησης και Ελέγχου , ΤΕΧΝΟΛΟΓΙΑ ΜΕΤΡΗΣΕΩΝ, Σχεδίαση & Ανάπτυξη Συστημάτων Μετρήσεων & Συστημάτων Συλλογής Δεδομένων 2η Έκδοση, ΤΖΙΟΛΑΣ, 2019.
- [40] Components101, «L298N Motor Driver Module,» Components101, 2020. [Ηλεκτρονικό]. Available: <https://components101.com/modules/l293n-motor-driver-module?fbclid=IwAR2VdcJeIRoztMfe6X0sXy2AAUqhE0XHxBScmF-FXOBXytMQ0OOZUImJLaE>. [Πρόσβαση Νοέμβριος 2020].