



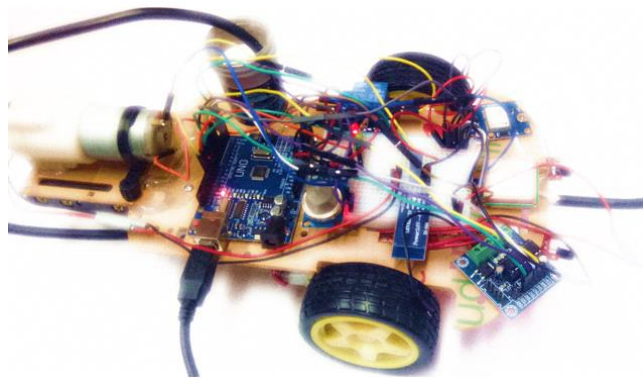
Πανεπιστήμιο Δυτικής Αττικής

Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

## Πτυχιακή Εργασία

*Θέμα:*

«Κατασκευή αυτόνομου πυροσβεστικού οχήματος  
με δυνατότητα ανίχνευσης και κατάσβεσης πύρινης εστίας»



Νικολαρέας Πελοπίδας

Αριθμός Μητρώου: 36962

Επιβλέπων Καθηγητής: Δρ. Ζάχαρης Νικόλαος

Αιγάλεω 2019

## ΠΡΟΛΟΓΟΣ

Η παρούσα πτυχιακή εργασία έχει εκπονηθεί στο Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών του Πανεπιστημίου Δυτικής Αττικής και στόχος της ήταν η κατασκευή ενός πυροσβεστικού οχήματος μικρής κλίμακας το οποίο να μπορεί να ανιχνεύει και να σβήνει μικρή πύρινη εστία.

Η αποστολή επανδρωμένων πυροσβεστικών οχημάτων για την κατάσβεση μεγάλων πύρινων μετώπων ενέχει κινδύνους τόσο για τη σωματική ακεραιότητα αλλά και τη ζωή των πυροσβεστών. Η λύση της δημιουργίας και κατασκευής αυτόνομων μη επανδρωμένων πυροσβεστικών οχημάτων εξασφαλίζει αρχικά ότι δεν θα χαθούν ανθρώπινες ζωές, αλλά και ότι η φωτιά μπορεί να κατασβηστεί αποτελεσματικά χωρίς την ανθρώπινη παρέμβαση, αλλά με τη χρήση τεχνικών μηχανικής όρασης και αισθητήρων καπνού, θερμότητας και έντασης-κατεύθυνσης ανέμου.

Ευχαριστώ θερμά τον κ. Ζάχαρη Νικόλαο τόσο για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα, όσο και για την αμέριστη βοήθεια που μου προσέφερε καθόλη τη διάρκεια εκπόνησης της πτυχιακής μου εργασίας χωρίς την οποία δεν θα ήταν δυνατή η ολοκλήρωση αυτού του έργου.

## ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας πτυχιακής εργασίας η μελέτη των βασικών αρχών σχεδιασμού, ανάπτυξης και λειτουργίας των τεχνικών αυτόματης οδήγησης οχημάτων, αλλά και η κατασκευή ενός μοντέλου μικρής κλίμακας με κύρια χαρακτηριστικά τη δυνατότητα ανίχνευσης και κατάσβεσης μικρής πύρινης εστίας.

Η εργασία αυτή αναπτύσσεται σε έξι κεφάλαια. Στο πρώτο κεφάλαιο γίνεται μία ιστορική αναδρομή στα αυτοκινούμενα οχήματα, καθώς και μία περιγραφή στην ιστορία της εξέλιξής τους.

Στο επόμενο κεφάλαιο περιγράφουμε αναλυτικά τα κύρια τμήματα που συνθέτουν το υλικό του οχήματός μας. Γίνεται λοιπόν μία λεπτομερή αναφορά στην «καρδιά» του μοντέλου μας, τον μικροελεγκτή Arduino Uno, περιγράφοντας τόσο τις δυνατότητές του, όσο και τα δομικά του στοιχεία. Στη συνέχεια αναφερόμαστε στα περιφερειακά τμήματα, δηλαδή τους αισθητήρες καπνού και φωτιάς, τους κινητήρες συνεχούς ρεύματος που είναι υπεύθυνοι για τον έλεγχο κίνησης, το κύκλωμα του bluetooth και τέλος τη μονάδα γεωεντοπισμού θέσης GPS.

Το τρίτο κεφάλαιο ασχολείται με τη διαδικασία που ακολουθήσαμε για την υλοποίηση του μοντέλου αυτόνομου οχήματος, ενώ στο επόμενο κεφάλαιο παραθέτουμε τον κώδικα του προγράμματος, καθώς και την λειτουργία του οχήματος σε αυτόνομη και χειροκίνητη εκδοχή.

Στο πέμπτο κεφάλαιο περιγράφουμε τον αλγόριθμο που ακολουθεί το όχημα μας για την αυτόματη και χειροκίνητη λειτουργία του.

Τέλος, στο έκτο και τελευταίο κεφάλαιο παρουσιάζουμε τα αποτελέσματα της αυτόνομης λειτουργίας, την επικοινωνία με το κινητό, τις διάφορες απαιτήσεις του περιβάλλοντος εργασίας μοντέλου, καθώς και τα σχετικά συμπεράσματα που προκύπτουν.

## ABSTRACT

The purpose of this thesis is to study the basic principles of design, development and operation of automated vehicle driving techniques, as well as the construction of a small scale model with the main features of detecting and extinguishing a small fire hob.

This paper is developed in six chapters. In the first chapter there is a historical retrospective of self-propelled vehicles, as well as a description in the history of their evolution.

In the next chapter we describe in detail the main sections that make up the material of our vehicle. There is a detailed reference to the "heart" of our model, the microcontroller Arduino Uno, describing both its capabilities and its structural elements. Then we refer to the peripheral parts, namely the smoke and fire sensors, the DC motors that are responsible for motion control, the Bluetooth circuit and the GPS positioning unit.

The third chapter deals with the procedure we followed for the implementation of the autonomous vehicle model, while in the next chapter we give the program code, as well as the operation of the vehicle in an autonomous and manual version.

In the fifth chapter we describe the algorithm that follows our vehicle for its automatic and manual operation.

Finally, in the sixth and final chapter we present the results of autonomous operation, communication with the mobile, the various requirements of the model work environment, as well as the resulting conclusions.

## Περιεχόμενα

Κεφάλαιο 1° ΑΥΤΟΚΙΝΟΥΜΕΝΑ ΟΧΗΜΑΤΑ.....	8
1.1 Εισαγωγή.....	8
1.1.1 Ιστορική αναδρομή αυτοκινούμενων οχημάτων .....	8
1.1.2 Αυτοκινούμενα οχήματα στη σύγχρονη εποχή .....	10
Κεφάλαιο 2° Ο ΜΙΚΡΟΕΛΕΓΚΤΗΣ ARDUINO UNO.....	11
2.1 Εισαγωγή στο μικροελεγκτή Arduino-Uno .....	11
2.2 Ιστορική αναδρομή του Arduino.....	12
2.3 Η πλακέτα του Arduino .....	13
2.3.1 Είσοδοι – έξοδοι.....	14
2.3.2 Τροφοδοσία .....	17
2.3.3 Διακόπτες και ενδεικτικές φωτοдиодοι της πλακέτας .....	18
2.4 Το περιβάλλον Arduino IDE (Integrated Development Environment) .....	19
2.5 Γλώσσα προγραμματισμού .....	22
2.6 Bluetooth.....	25
2.6.1 Ιστορία του Bluetooth .....	25
2.6.2 Εφαρμογές του Bluetooth.....	25
2.6.3 Εκδόσεις του Bluetooth.....	25
2.6.4 Δομή του Bluetooth.....	27
2.6.5 Bluetooth Διάταξη (Module) HC-06.....	29
2.7 GPS.....	31
2.7.1 GPS NEO-6M Περιγραφή Λειτουργίας.....	32
2.7.1.1 Πρωτόκολλα και Διεπαφές .....	33
2.7.1.2 Χρονισμός ακριβείας.....	34
2.7.1.3 Διαχείριση ενέργειας .....	36

2.7.1.4 GPS Performance .....	37
2.7.1.5 Absolute maximum ratings .....	38
2.7.1.6 Συνθήκες λειτουργίας .....	39
2.7.1.7 Καταμερισμός εργασίας Pin .....	40
2.7.1.8 Μπλοκ Διάγραμμα .....	41
2.7.2 Τρόπος λειτουργίας .....	42
2.7.3 Λειτουργικά μέρη.....	43
<b>Κεφάλαιο 3° ΚΙΝΗΤΗΡΕΣ ΣΥΝΕΧΟΥΣ ΡΕΥΜΑΤΟΣ.....</b>	<b>45</b>
3.1 Εισαγωγή.....	45
3.1.1 Αρχή λειτουργίας .....	47
3.1.2 Κατηγορίες.....	49
3.2 Κινητήρες DC.....	50
3.2.1 Motor driver .....	51
3.3 DC-κινητήρας.....	52
<b>Κεφάλαιο 4° ΑΝΑΠΤΥΞΗ ΑΛΓΟΡΙΘΜΟΥ.....</b>	<b>54</b>
4.1 Εισαγωγή.....	54
4.2 Περιγραφή – Οδηγίες λειτουργίας.....	54
4.3 Αλγόριθμος Αυτόματης Λειτουργίας .....	58
<b>Κεφάλαιο 5° ΠΕΡΙΓΡΑΦΗ ΚΩΔΙΚΑ – ΣΥΜΠΕΡΑΣΜΑΤΑ - ΑΠΟΤΕΛΕΣΜΑΤΑ.....</b>	<b>62</b>
5.1 Βασικές Λειτουργίες Προγράμματος .....	62
5.1.1 Συνάρτηση pumpControl() .....	63
5.1.2 Συνάρτηση sensorsRead() .....	63
5.1.3 Συνάρτηση bluetoothRead() .....	64
5.1.4 Συνάρτηση gpsRead() .....	65
5.1.5 Συνάρτηση gpsPosition().....	65

5.1.6 Συνάρτηση gpsTime() .....	66
5.1.7 Συνάρτηση speedUp() .....	66
5.1.8 Συνάρτηση speedDown().....	67
5.1.9 Συνάρτηση moveL().....	67
5.1.10 Συνάρτηση moveR().....	67
5.1.11 Συνάρτηση moveF() .....	68
5.1.12 Συνάρτηση moveB().....	68
5.1.13 Συνάρτηση autoMove() .....	68
5.1.14 Κώδικας .....	71
5.2 Περιγραφή Διάταξης.....	76
5.3 Συμπεράσματα.....	78
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ .....</b>	<b>79</b>

# Κεφάλαιο 1<sup>ο</sup> ΑΥΤΟΚΙΝΟΥΜΕΝΑ ΟΧΗΜΑΤΑ

## 1.1 ΕΙΣΑΓΩΓΗ

Τα αυτοκινούμενα οχήματα άρχισαν να εμφανίζονται τις αρχές του 20<sup>ου</sup> αιώνα σε πειραματικό στάδιο. Η λειτουργία τους βασιζόταν σε ηλεκτρονόμους, οι οποίοι τους επέτρεπαν να εκτελούν απλές κινήσεις, ενώ παράλληλα είχαν μεγάλο όγκο και ζύγιζαν πολλά κιλά.

### 1.1.1 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΑΥΤΟΚΙΝΟΥΜΕΝΩΝ ΟΧΗΜΑΤΩΝ

Μία από τις πρώτες προσπάθειες δημιουργίας αυτοκινούμενου οχήματος ήταν η κατασκευή του Electric dog (Σχήμα 1.1) το 1912. Το Electric dog μπορούσε με χρήση φωτοκύτταρου να ανιχνεύσει την ύπαρξη φωτός και να το κατευθύνει προς την πηγή εκείνη.

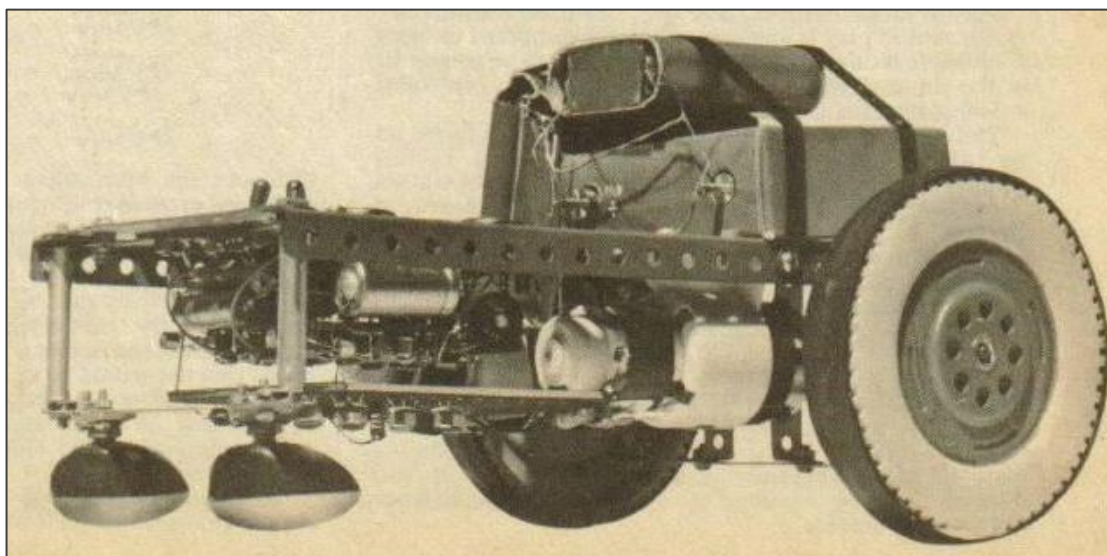


*Σχήμα 1.1 – Electric dog*

Το 1969 κατασκευάστηκε ένα αυτοκινούμενο όχημα με όνομα Emma (Σχήμα 1.2), το οποίο είχε περισσότερες δυνατότητες από τα προηγούμενα μοντέλα του ίδιου είδους, όπως ανίχνευση φωτός από 2

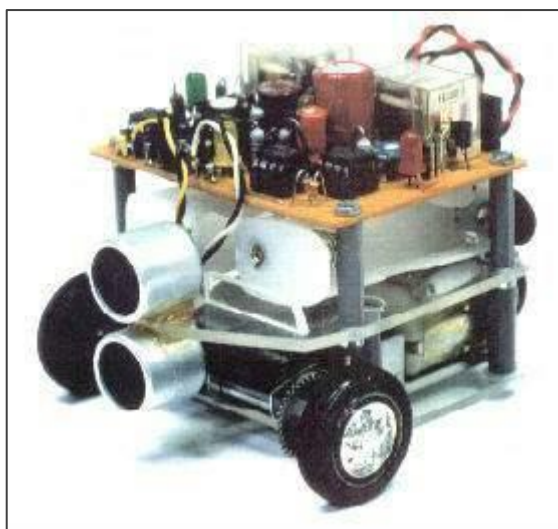


πηγές αλλά και έλεγχο του καταναλισκόμενου φορτίου των κινητήρων, που του επέτρεπε την απενεργοποίηση του.



*Σχήμα 1.2 - Emma*

Το 1980 κατασκευάστηκε το TIMBUG 2 (Σχήμα 1.3), το πρώτο στην ουσία αυτοκινούμενο όχημα που χρησιμοποιούσε αισθητήρες υπερήχων, ενώ μπορούσε να εκτελέσει και όλες τις δυνατές κινήσεις (αριστερά, δεξιά, μπρος, πίσω).



*Σχήμα 1.3 - TIMBUG 2*

Την ίδια εποχή στην Βρετανία δημιουργήθηκαν 2 αυτοκινούμενα οχήματα, το Prowler και το Zeaker, τα οποία ανίχνευαν εμπόδια με αισθητήρες κρούσης και συνδέονταν ενσύρματα με έναν ηλεκτρονικό υπολογιστή. Τέλος, μια ακόμα προσπάθεια έγινε το 1991 στο Πολυτεχνείο της Λοζάνης με το Khepera, το οποίο ήταν από τα πρώτα οχήματα που λειτουργούσαν χωρίς να είναι συνδεδεμένα με ηλεκτρονικό υπολογιστή, αλλά απλά με την εκτέλεση κώδικα που ήταν αποθηκευμένος σ' αυτό.

### 1.1.2 ΑΥΤΟΚΙΝΟΥΜΕΝΑ ΟΧΗΜΑΤΑ ΣΤΗ ΣΥΓΧΡΟΝΗ ΕΠΟΧΗ

Η ανάπτυξη της τεχνολογίας των αυτοκινούμενων οχημάτων από τις αρχές του 20ου αιώνα μέχρι σήμερα είναι αλματώδης. Χωρίς την ύπαρξη αυτοκινούμενων οχημάτων πολλά επιτεύγματα του σύγχρονου πολιτισμού δεν θα ήταν εφικτά, όπως π.χ. η κατάκτηση του διαστήματος, παράλληλα με την εξερεύνηση άλλων πλανητών και ουράνιων σωμάτων. Επίσης συμβάλλουν και στην πρόσβαση σε σημεία του πλανήτη μας που δεν μπορεί να πλησιάσει ο άνθρωπος, όπως μέρη με εξαιρετικά υψηλή ή χαμηλή θερμοκρασία, σπήλαια στα βάθη της Γης, περιοχές που έχουν πληγεί από πυρηνικά ατυχήματα, εμπόλεμες ζώνες κ.ο.κ. .

Η επιστήμη και η τεχνολογία των αυτοκινούμενων οχημάτων αντιμετωπίζει στις μέρες μας μία τεράστια πρόκληση, αυτή της διαχείρισης ενέργειας. Η τάση για ολοένα και περισσότερη χρήση φορητών συσκευών οδηγεί στην ανάγκη και στον τομέα αυτό για έξυπνα οχήματα τα οποία καταναλώνουν όσο το δυνατόν λιγότερη ενέργεια χωρίς αυτό να είναι εις βάρος της λειτουργικότητας τους.

## Κεφάλαιο 2<sup>ο</sup> Ο ΜΙΚΡΟΕΛΕΓΚΤΗΣ ARDUINO UNO

### 2.1 ΕΙΣΑΓΩΓΗ ΣΤΟ ΜΙΚΡΟΕΛΕΓΚΤΗ ARDUINO-UNO

Στο παρόν κεφάλαιο της πτυχιακής εργασίας ασχολούμαστε με το μικροελεγκτή Arduino. Ο μικροελεγκτής Arduino Uno (Σχήμα 2.1) είναι η πιο δημοφιλής αναπτυξιακή πλακέτα της οικογένειας Arduino, με πληθώρα εφαρμογών σε όλο το φάσμα της τεχνολογίας. Αυτός είναι και ο κύριος λόγος που στηρίξαμε την ανάπτυξη του αυτοκινούμενου οχήματος μας σε αυτόν τον μικροελεγκτή.



Σχήμα 2.1 - Η πλακέτα του μικροελεγκτή Arduino

## 2.2 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΤΟΥ ARDUINO

Η αρχή έγινε το 2005 στην Ivrea της Ιταλίας όπου κατασκευάζεται μία συσκευή η οποία είχε την δυνατότητα να ελέγχει και να αλληλεπιδρά σύμφωνα με το περιβάλλον. Σκοπός των κατασκευαστών ήταν αυτή η συσκευή να κοστίζει λιγότερο σε σχέση με άλλες οι οποίες είχαν παρόμοιες δυνατότητες. Η ομάδα των δημιουργών Massimo Banzi, David Cuartielles, Tom Igoe, David Mellis και Gianluca Martino ονόμασε τη συσκευή Arduino («ισχυρός φίλος») από τον Arduin της Ivrea, έναν βασιλιά της Ιταλίας του ενάτου αιώνα όπου κατοικούσε στην ίδια πόλη.

Το Arduino αναπτύχθηκε σύμφωνα με την πλατφόρμα Wiring, μία πτυχιακή εργασία του Hernando Barragan από το Interaction Design Institute Ivrea. Είχε ως στόχο να είναι μία ηλεκτρονική εκδοχή της Processing που θα χρησιμοποιούσε ένα περιβάλλον προγραμματισμού δικό της αλλά θα έμοιαζε σχεδιαστικά και συντακτικά με αυτό της Processing.

Το Arduino αποτελείται από δύο κύρια μέρη, την πλακέτα Arduino το οποίο είναι το τμήμα του υλικού πάνω στο οποίο εργάζεται ο κατασκευαστής όταν πραγματοποιεί μία κατασκευή ενώ το δεύτερο τμήμα είναι το περιβάλλον Arduino IDE, που αποτελεί και το λογισμικό που εκτελείται στον ηλεκτρονικό υπολογιστή. Το IDE χρησιμοποιείται για να δημιουργηθεί ένα μικρό πρόγραμμα στον υπολογιστή (sketch) που φορτώνεται στον μικροελεγκτή της πλακέτας Arduino. Με αυτόν τον τρόπο ο αλγόριθμος που εκτελείται στο sketch οδηγεί το κύκλωμα στην πλακέτα Arduino.

## 2.3 Η ΠΛΑΚΕΤΑ ΤΟΥ ARDUINO

Η πλακέτα Arduino είναι ένα ολοκληρωμένο σύστημα μικρού κυκλώματος που περιέχει ένα μικροελεγκτή. Αυτή η μορφή υπολογιστή είναι κατά πολύ οικονομικότερη από ένα σύγχρονο υπολογιστικό σύστημα και με λιγότερη υπολογιστική ισχύ, αλλά είναι πολύ χρήσιμη για την κατασκευή ηλεκτρονικών συσκευών.

Η ομάδα του Arduino έχει τοποθετήσει σε αυτήν την πλακέτα όλα τα απαραίτητα στοιχεία που απαιτούνται για τον μικροελεγκτή ώστε να μπορεί να λειτουργεί σωστά και να επικοινωνεί με τον υπολογιστή αλλά και με άλλες σειριακές συσκευές.



Σχήμα 2.2 - Η πλακέτα του μικροελεγκτή Arduino και από τις δύο όψεις

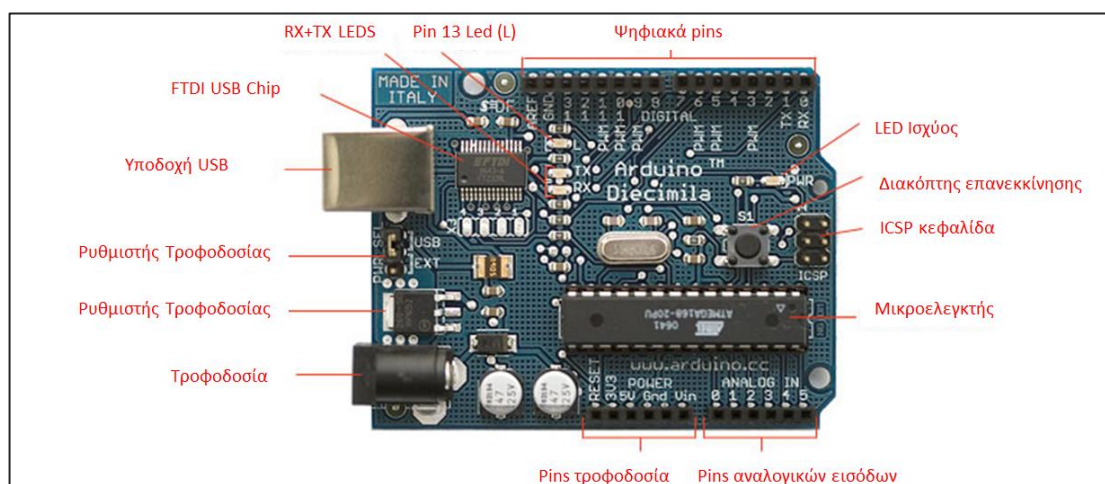
Ειδικότερα μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (ATmega328, ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωση του σε άλλα κυκλώματα. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή χρονισμένο στα 16MHz. Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα bootloader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής.

Γενικά όλες οι πλακέτες είναι προγραμματισμένες μέσω μιας σειριακής σύνδεσης RS-232 αλλά ο τρόπος με τον οποίο αυτό υλοποιείται ποικίλλει ανάλογα με την έκδοση. Οι σειριακές πλακέτες Arduino περιέχουν ένα απλό κύκλωμα αντιστροφής για την μετατροπή ανάμεσα στα σήματα των επίπεδων RS-232 και TTL.

Οι πλακέτες Arduino που κυκλοφορούν σήμερα στην αγορά συμπεριλαμβανόμενης και της Uno, προγραμματίζονται μέσω USB, εφαρμόζοντας ένα chip προσαρμογέα USB-to-Serial όπως το FTDI FT232.

### 2.3.1 ΕΙΣΟΔΟΙ – ΕΞΟΔΟΙ

Η βασική διεπαφή (interface) στο Arduino είναι η σειριακή. Ο μικροελεγκτής ATmega (Σχήμα 2.4) υποστηρίζει σειριακή επικοινωνία, την οποία το Arduino προωθεί μέσα από έναν ελεγκτή Serial-over-USB ώστε να συνδέεται με τον υπολογιστή μέσω USB. Η σύνδεση αυτή χρησιμοποιείται για την μεταφορά των προγραμμάτων που σχεδιάζονται από τον υπολογιστή στο Arduino αλλά και για αμφίδρομη επικοινωνία του Arduino με τον υπολογιστή μέσα από το πρόγραμμα κατά τη διάρκεια της εκτέλεσης.

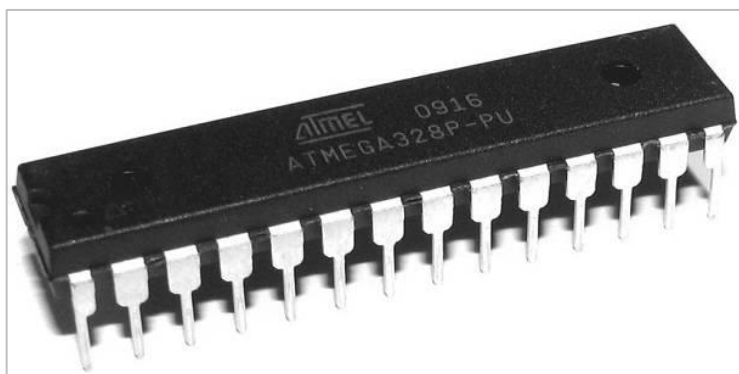


Σχήμα 2.3 - Η πλακέτα του μικροελεγκτή Arduino και με τα στοιχεία που την απαρτίζουν

Επιπλέον, στην πάνω πλευρά του Arduino βρίσκονται 14 θηλυκά pins, αριθμημένα από το 0 έως το 13, που μπορούν να λειτουργήσουν ως ψηφιακές εισοδοι και έξοδοι. Λειτουργούν στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί ρεύμα μέγιστης έντασης 40mA.

Ως ψηφιακή έξοδος, ένα από αυτά τα pins μπορεί να τεθεί από το πρόγραμμα σε κατάσταση HIGH ή LOW, οπότε το Arduino ενεργοποιεί ή όχι την παροχή ρεύματος σ' αυτό το pin. Με αυτόν τον

τρόπο είναι εφικτό να ανάψει και να σβήσει ένα LED που είναι συνδεδεμένο στο συγκεκριμένο pin. Αν πάλι ένα από αυτά τα pins έχει ρυθμιστή ως ψηφιακή είσοδος μέσα από το πρόγραμμα, μπορεί με την κατάλληλη εντολή να διαβαστεί η κατάσταση του (HIGH ή LOW) ανάλογα με το αν η εξωτερική συσκευή που έχει συνδεθεί σε αυτό το pin διοχετεύει ή όχι ρεύμα στο pin.



*Σχήμα 2.4 - Μικροελεγκτής ATMEGA328P-PU*

Μερικά από αυτά τα 14 pin, εκτός από ψηφιακές εισόδους/εξόδους έχουν και δεύτερη λειτουργία. Συγκεκριμένα :

Τα pins 0 και 1 λειτουργούν ως RX και TX της σειριακής όταν το πρόγραμμα ενεργοποιεί την σειριακή θύρα. Έτσι, όταν λόγω χάρη το πρόγραμμα στέλνει δεδομένα στην σειριακή θύρα, αυτά προωθούνται και στην θύρα USB μέσω του ελεγκτή Serial-Over-USB αλλά και στο pin 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή (π.χ. ένα δεύτερο Arduino στο δικό του pin 1). Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμα ενεργοποιηθεί το σειριακό interface τότε οι ψηφιακές εισοδοί/έξοδοι μειώνονται κατά 2.

Τα pins 2 και 3 λειτουργούν και ως εξωτερικές διακοπές (interrupt 0 και 1 αντίστοιχα). Με άλλα λόγια, μπορούν να ρυθμιστούν μέσα από το πρόγραμμα ώστε να λειτουργούν αποκλειστικά ως ψηφιακές εισοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει άμεσα και εκτελείται μια συγκεκριμένη συνάρτηση . Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

Τα pins 3, 5, 6, 9, 10 και 11 μπορούν να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι με το σύστημα διαμόρφωσης εύρους παλμών (PWM-Pulse Width Modulation), δηλαδή το ίδιο σύστημα που

διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων. Έτσι, αν συνδεθεί λόγω χάρη ένα LED σε κάποιο από αυτά τα pins, μπορεί να ελεγχθεί πλήρως η φωτεινότητά του με ανάλυση 8bits (256 καταστάσεις από 0-σβηστό ως 255–πλήρως αναμμένο) αντί να υπάρχει απλά η δυνατότητα να είναι απλά αναμμένο ή σβηστό που παρέχουν οι υπόλοιπες ψηφιακές έξοδοι. Το PWM παίρνει ένα εύρος τιμών από το 0 έως το 255. Με λίγα λόγια το PWM δεν είναι πραγματικά αναλογικό σύστημα, έτσι θέτοντας στην έξοδο την τιμή 127, δεν σημαίνει ότι η έξοδος θα παρέχει 2.5V αντί της κανονικής τιμής των 5V, αλλά ότι θα δίνει έναν παλμό που η τάση του θα εναλλάσσεται με μεγάλη συχνότητα και για ίσα χρονικά διαστήματα μεταξύ των τιμών 0V και 5V με σκοπό η μέση τιμή να ισούται με 2,5V.

Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN, υπάρχει μια ακόμη σειρά από 6 pins, αριθμημένα από το 0 έως το 5. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC (Analog to Digital Converter) που είναι ενσωματωμένο στον μικροελεγκτή. Για παράδειγμα, αν τροφοδοτηθεί ένα από αυτά τα pins με μία τάση η οποία μπορεί να κυμανθεί με ένα ποτενσιόμετρο από 0V ως μία τάση αναφοράς  $V_{ref}$  (η οποία αν δεν γίνει κάποια αλλαγή είναι προρυθμισμένη στα 5V), τότε μέσα από το πρόγραμμα μπορεί να «διαβαστεί» η τιμή της θύρας ως ένας ακέραιος αριθμός χωρητικότητας 10-bits, από το 0 (όταν η τάση στο pin είναι 0V) μέχρι το 1023 (όταν η τάση στο pin είναι 5V) . Η τάση αναφοράς μπορεί να ρυθμιστεί με μία εντολή όπως για παράδειγμα στα 1.1V. Ένας άλλος τρόπος όπου η τάση αναφοράς μπορεί να δηλωθεί από τον προγραμματιστή είναι τροφοδοτώντας με μία εξωτερική τάση αναφοράς τη θύρα με την σήμανση AREF (βλ. Σχήμα 2.3). Έτσι, αν τροφοδοτηθεί η θύρα AREF με 3.3 v και στην συνέχεια εκτελεσθεί η εντολή να διαβαστεί κάποιο pin αναλογικής εισόδου στο οποίο εφαρμόζετε τάση 1.65V , το Arduino θα επιστρέψει την τιμή 512. Τέλος, καθένα από τα 6 αυτά pins, με κατάλληλη εντολή μέσα από το πρόγραμμα μπορεί να μετατραπεί σε ψηφιακό pin εισόδου/εξόδου όπως τα 14 που βρίσκονται στην απέναντι πλευρά και τα οποία περιεγράφηκαν πριν. Σε αυτή την περίπτωση τα pins μετονομάζονται από 0 - 5 σε 14 - 19 αντίστοιχα.



### 2.3.2 ΤΡΟΦΟΔΟΣΙΑ

Το Arduino μπορεί να τροφοδοτηθεί με ρεύμα είτε από τον υπολογιστή μέσω της σύνδεσης USB, είτε από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής φισ των 2.1 mm (θετικός πόλος στο κέντρο) και βρίσκεται στην κάτω-αριστερή γωνία του Arduino.



*Σχήμα 2.5 – Περίπτωση τροφοδοσίας Arduino από μπαταρία*

Για να μην υπάρχουν προβλήματα, η εξωτερική τροφοδοσία πρέπει να είναι από 7V ως 12V και μπορεί να προέρχεται από ένα κοινό μετασχηματιστή του εμπορίου, από μπαταρίες ή οποιαδήποτε άλλη πηγή DC.

Δίπλα από τα pins αναλογικής εισόδου, υπάρχει μια ακόμα συστοιχία από 6 pins με την σήμανση POWER.

Η λειτουργία του καθενός έχει ως εξής:

Το πρώτο, με την ένδειξη RESET, όταν γειωθεί (σε οποιοδήποτε από τα 3 pins με την ένδειξη GND που υπάρχουν στο Arduino) έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.

Το δεύτερο, με την ένδειξη 3.3V, μπορεί να τροφοδοτήσει διατάξεις, συσκευές ή αισθητήρες με τάση 3.3V. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial-over-USB και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA.

Το τρίτο, με την ένδειξη 5V, μπορεί να τροφοδοτήσει και αυτό διάφορες διατάξεις, συσκευές ή αισθητήρες με τάση 5V. Ανάλογα με τον τρόπο τροφοδοσίας του ίδιου του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την θύρα USB (που ούτως ή άλλως λειτουργεί στα 5V), είτε από την εξωτερική τροφοδοσία αφού αυτή περάσει από ένα ρυθμιστή τάσης για να την «κατεβάσει» στα 5V.

Το τέταρτο και το πέμπτο pin, με την ένδειξη GND, είναι γειώσεις. Το έκτο και τελευταίο pin, με την ένδειξη Vin έχει διπλό ρόλο. Σε συνδυασμό με το pin γείωσης δίπλα του, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino, στην περίπτωση που δεν βολεύει να χρησιμοποιηθεί η υποδοχή του φισ των 2.1mm . Αν όμως υπάρχει ήδη συνδεδεμένη εξωτερική τροφοδοσία μέσω του φισ, τότε μπορεί να χρησιμοποιηθεί αυτό το pin για να τροφοδοτήσει εξαρτήματα και συσκευές με την πλήρη τάση της εξωτερικής τροφοδοσίας (7-12V), πριν αυτή περάσει από τον ρυθμιστή τάσης όπως γίνεται με το pin των 5V.

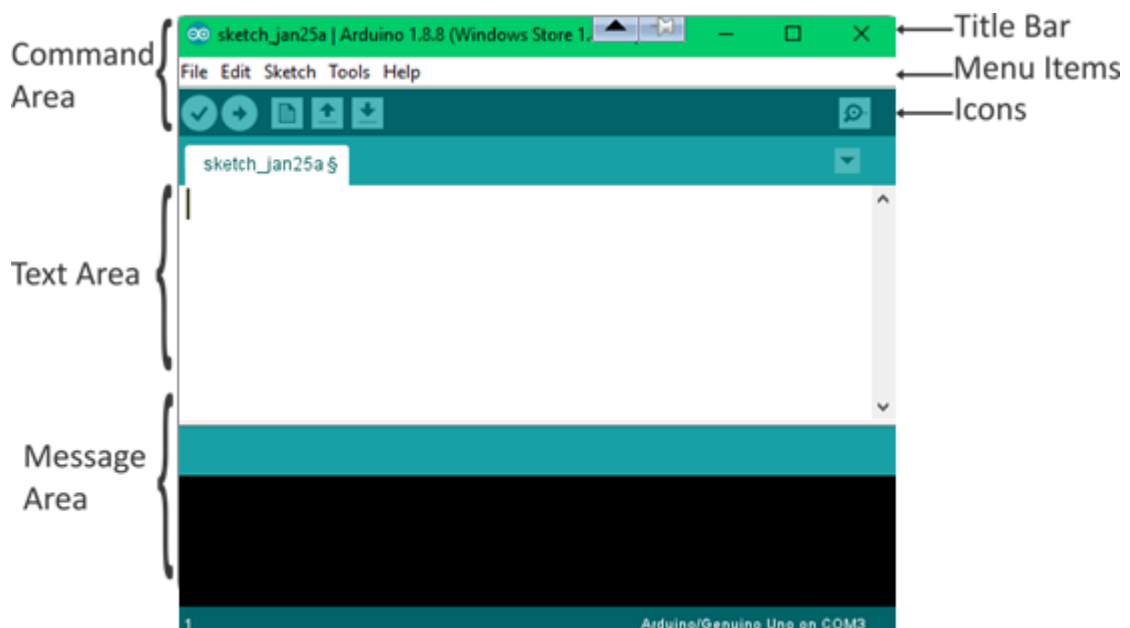
### 2.3.3 ΔΙΑΚΟΠΤΕΣ ΚΑΙ ΕΝΔΕΙΚΤΙΚΕΣ ΦΩΤΟΔΙΟΔΟΙ ΤΗΣ ΠΛΑΚΕΤΑΣ

Στην πλακέτα Arduino υπάρχουν ένας μικροδιακόπτης (microswitch) και τέσσερα μικροσκοπικά LED επιφανειακής στήριξης. Η λειτουργία του διακόπτη (που έχει την σήμανση RESET) είναι για την επανεκκίνηση του συστήματος και του ενός LED με την σήμανση POWER για την παροχή σωστής τροφοδοσίας. Τα δύο LEDs με τις σημάνσεις TX και RX, χρησιμοποιούνται ως ένδειξη λειτουργίας της σειριακής μέσω USB. Τα LEDs αυτά ελέγχονται από τον ελεγκτή Serial-over-USB και συνεπώς δεν λειτουργούν όταν η σειριακή επικοινωνία γίνεται αποκλειστικά μέσω των ψηφιακών pin 0 και 1. Τέλος, υπάρχει το LED με την σήμανση L, με βασική λειτουργία του LED στην πλακέτα Arduino είναι για να αναβοσβήνει συνήθως για δοκιμαστικό σκοπό. Οι κατασκευαστές του σκέφτηκαν να ενσωματώσουν ένα LED στην πλακέτα, το οποίο σύνδεσαν στο ψηφιακό pin 13. Έτσι, ακόμα και αν δεν έχει συνδεθεί τίποτα πάνω στο φυσικό pin 13, αναθέτοντάς του την τιμή HIGH μέσα από το πρόγραμμά, θα ανάψει το ενσωματωμένο LED.

## 2.4 ΤΟ ΠΕΡΙΒΑΛΛΟΝ ARDUINO IDE (INTEGRATED DEVELOPMENT ENVIRONMENT)

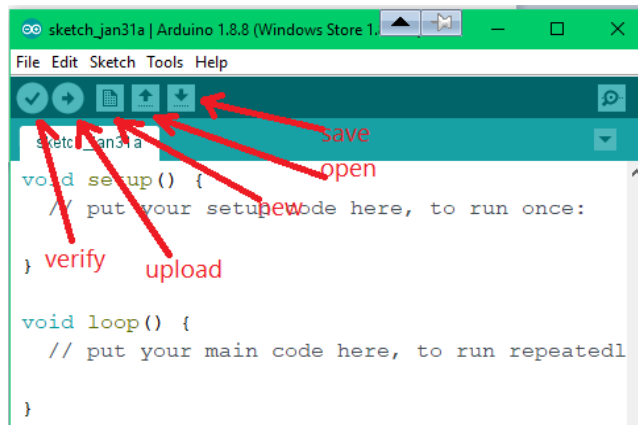
Το Arduino IDE (τελευταία έκδοση v.1.8.8 τη στιγμή συγγραφής της παρούσας πτυχιακής εργασίας) είναι ένα πρόγραμμα βασισμένο σε Java και συγκεκριμένα παρέχει: ένα πρακτικό περιβάλλον για την συγγραφή των προγραμμάτων σας (τα οποία ονομάζονται sketch στην ορολογία του Arduino) με συντακτική χρωματική σήμανση, αρκετά έτοιμα παραδείγματα, μερικές έτοιμες βιβλιοθήκες για προέκταση της γλώσσας και τον εύκολο χειρισμό μέσα από τον κώδικα εξαρτημάτων που συνδέονται στο Arduino, τον compiler για την μεταγλώττιση των sketch, ένα serial monitor που την παρακολούθηση της επικοινωνίας της σειριακής (USB), αναλαμβάνει επίσης την αποστολή αλφαριθμητικών στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για το debugging των sketch καθώς και τη μεταφόρτωση μεταγλωττισμένων sketch στο Arduino.

Το Arduino IDE είναι αρκετά απλούστερο σε αντίθεση με άλλα περιβάλλοντα ανάπτυξης λογισμικού όπως το Eclipse, το Xcode και το Visual Studio. Κυρίως αποτελείται από έναν κειμενογράφο (editor), έναν μεταφραστή (compiler), ένα φορτωτή (loader) και ένα παράθυρο σειριακής παρακολούθησης (serial monitor). Δεν περιέχει προχωρημένες λειτουργίες όπως debugging ή code completion, αλλά δίνει μόνο τη δυνατότητα για απλές ρυθμίσεις (στο υπόμενου preferences).



Σχήμα 2.6 - Περιβάλλον Arduino IDE

Στο σχήμα που ακολουθεί φαίνεται η εικόνα της εργαλειοθήκης (toolbar) του Arduino IDE και η αναλυτική λειτουργία των κουμπιών της που δίνει άμεση πρόσβαση στις λειτουργίες που χρειάζονται περισσότερο για την ανάπτυξη των εφαρμογών του χρήστη.



Σχήμα 2.7 - Το toolbar του Arduino IDE

Με το κουμπί Verify ο χρήστης μπορεί να μεταγλωττίσει τον κώδικα που βρίσκεται εκείνη την στιγμή στον editor. Το κουμπί Verify εκτός από τον συντακτικό έλεγχο, αφού μεταγλωττιστεί ο κώδικας στη συνέχεια τον μετατρέπει σε μορφή κατάλληλη για να «φορτωθεί» στον μικροελεγκτή του Arduino. Το κουμπί New δημιουργεί ένα καινούριο πρόγραμμα διαγράφοντας οτιδήποτε υπάρχει στον editor. Πριν όμως πραγματοποιήσει αυτήν την ενέργεια, δίνει στον χρήστη την ευκαιρία να αποθηκεύσει το υπάρχον sketch (πρόγραμμα) στην περίπτωση που έχει κάνει κάποιες αλλαγές σε αυτό. Με το κουμπί Open μπορεί ο χρήστης να ανοίξει ένα υπάρχον πρόγραμμα από το σύστημα του. Το κουμπί Save αποθηκεύει τις αλλαγές που έχουν γίνει στο πρόγραμμα που επεξεργάστηκε στον Editor. Το κουμπί Upload όπως και το κουμπί Verify μεταγλωττίζει τον υπάρχον κώδικα στον editor. Με τη διαφορά όμως ότι αφού ελέγξει για τυχόν συντακτικά λάθη και το μετατρέψει σε μορφή κατάλληλη για το Arduino, στη συνέχεια θα τον προωθήσει στην θύρα που έχει επιλέξει ο προγραμματιστής από το μενού Tools → Serial Port ώστε να «φορτωθεί» στον μικροελεγκτή.

Ο υπολογιστής μπορεί να επικοινωνήσει με το Arduino μέσω σειριακής σύνδεσης. Επιλέγοντας το κουμπί Serial Monitor ανοίγει ένα παράθυρο με τίτλο serial monitor όπου επιτρέπει να παρακολουθεί ο προγραμματιστής τα δεδομένα που στέλνονται προς τα Arduino αλλά και τα δεδομένα που στέλνονται από το Arduino προς τον υπολογιστή.

Επιλέγοντας από το μενού το Sketch πέρα από τις λειτουργίες που αναφέρθηκαν παραπάνω για την εντολή Verify/compile, εμφανίζονται και κάποιες άλλες ενδιαφέρουσες λειτουργίες. Η επιλογή Show Sketch folder είναι μία συντόμευση η οποία ανοίγει σε ένα παράθυρο την διεύθυνση στην οποία το λειτουργικό σύστημα αποθηκεύει τα αρχεία των εφαρμογών. Η επιλογή Add file επιτρέπει στον χρήστη να ανοίξει ένα αρχείο το οποίο μπορεί να βρίσκεται οπουδήποτε μέσα στο σύστημα και να το αποθηκεύσει στον ίδιο φάκελο όπου ανήκει και η εφαρμογή.

Τελευταία είναι η επιλογή Import Library. Ο προγραμματιστής έχει την δυνατότητα να εισάγει στο πρόγραμμα που αναπτύσσει οποιαδήποτε βιβλιοθήκη, είτε αυτή είναι του Arduino είτε κάποια που δημιούργησε ο ίδιος.

Δίπλα από το Sketch στο μενού υπάρχει η επιλογή Tools. Σε αυτή την στήλη του μενού ο προγραμματιστής έχει την δυνατότητα να επιλέξει την θύρα (serial port) με την οποία θα επικοινωνήσει ο υπολογιστής με το Arduino, καθώς επίσης και πια συγκεκριμένη έκδοση Arduino έχει (board). Εκτός όμως από αυτές τις δύο βασικές λειτουργίες υπάρχουν και κάποιες άλλες. Το Auto Format το οποίο μορφοποιεί τον κώδικα που βρίσκεται στον editor κατάλληλα για να διαβάζεται ευκολότερα.

Το Archive Sketch το οποίο μετατρέπει την εφαρμογή που αναπτύσσει ο προγραμματιστής σε ένα αρχείο zip και το αποθηκεύει. Ακόμα σε μερικές περιπτώσεις όταν ένα αρχείο ανοιχθεί στο Arduino IDE υπάρχει η περίπτωση να περιέχει χαρακτήρες οι οποίοι δεν είναι ASCII. Έτσι επιλέγοντας το Fix Encoding & Reload ο κώδικας θα ανανεωθεί αντικαθιστώντας τους περίεργους χαρακτήρες σε χαρακτήρες UTF-8 έκδοση. Τέλος η επιλογή Burning Bootloader απευθύνεται σε προχωρημένους χρήστες όπου τους δίνεται η δυνατότητα να χρησιμοποιήσουν την μνήμη που αντιστοιχεί στον bootloader.

## 2.5 ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring, μια παραλλαγή της C/C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega, και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++. Για compiler χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc.

Λόγω της σχέσης της από την C, στην γλώσσα του Arduino χρησιμοποιούνται ουσιαστικά οι ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιων τύπων δεδομένων και τους ίδιους τελεστές όπως και στην C. Πέρα από αυτές όμως, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για την διαχείριση του ειδικού hardware του Arduino. Οι πιο σημαντικές από αυτές επεξηγούνται στον πίνακα που ακολουθεί:

Όρισμα	Είδος	Τύπος	Παράμετροι	Περιγραφή
<b>LOW</b>	Σταθερά	int	-	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
<b>HIGH</b>	Σταθερά	int	-	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
<b>INPUT</b>	Σταθερά	int	-	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
<b>OUTPUT</b>	Σταθερά	int	-	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού true.
<b>pinMode</b>	Εντολή	-	( <i>pin</i> , <i>mode</i> )	Καθορίζει αν το συγκεκριμένο ψηφιακό pin θα είναι pin εισόδου ή pin εξόδου ανάλογα με την τιμή που δίνεται στην παράμετρο <i>mode</i> (INPUT ή OUTPUT αντίστοιχα).
<b>digitalWrite</b>	Εντολή	-	( <i>pin</i> , <i>pinstatus</i> )	Θέτει την κατάσταση <i>pinstatus</i> (HIGH ή LOW) στο συγκεκριμένο ψηφιακό pin .
<b>digitalRead</b>	Συνάρτηση	int	( <i>pin</i> )	Επιστρέφει την κατάσταση του συγκεκριμένου ψηφιακού <i>pin</i> (0 για LOW και 1 για HIGH) εφόσον αυτό είναι pin εισόδου.
<b>analogReference</b>	Εντολή	-	( <i>type</i> )	Δέχεται τις τιμές DEFAULT, INTERNAL ή EXTERNAL στην παράμετρο <i>type</i> για να καθορίσει την τάση αναφοράς (V <sub>ει</sub> ) των αναλογικών εισόδων (5V, 1.1V ή η

				εξωτερική τάση με την οποία τροφοδοτείται το pin AREF αντίστοιχα)
<b>analogRead</b>	Συνάρτηση	int	(pin)	Επιστρέφει έναν ακέραιο από 0 έως 1023, ανάλογα με την τάση που τροφοδοτείται το συγκεκριμένο pin αναλογικής εισόδου στην κλίμακα 0 ως Vref.
<b>analogWrite</b>	Εντολή	-	(pin, value)	Θέτει το συγκεκριμένο ψηφιακό pin σε κατάσταση ψευδοαναλογικής εξόδου (PWM). Η παράμετρος value καθορίζει το πλάτος του παλμού σε σχέση με την περίοδο του παραγόμενου σήματος στην κλίμακα από 0 ως 255 (π.χ. με value 127, το πλάτος του παλμού είναι ίσο με μισή περίοδο).
<b>millis</b>	Συνάρτηση	unsigned long	()	Μετρητής που επιστρέφει το χρονικό διάστημα σε ms από την στιγμή που άρχισε η εκτέλεση του προγράμματος. Λάβετε υπόψη ότι λόγω του τύπου μεταβλητής (unsigned long δηλ. 32bit) θα γίνει overflow σε 2 <sup>32</sup> ms δηλαδή περίπου σε 50 μέρες, οπότε ο μετρητής θα ξεκινήσει πάλι από το μηδέν.
<b>delay</b>	Εντολή	-	(time)	Σταματά προσωρινά την ροή του προγράμματος για time ms. Η παράμετρος time είναι unsigned long (από 0 ως 2 <sup>32</sup> ). Σημειώστε ότι παρά την προσωρινή παύση, συναρτήσεις των οποίων η εκτέλεση ενεργοποιείται από interrupt θα εκτελεστούν κανονικά κατά την διάρκεια μιας delay.
<b>attachInterrupt</b>	Εντολή	-	(interrupt, function, triggermode)	Θέτει σε λειτουργία το συγκεκριμένο interrupt, ώστε να ενεργοποιεί την συνάρτηση function, κάθε φορά που ικανοποιείται η συνθήκη που ορίζεται από την παράμετρο triggermode: <ul style="list-style-type: none"> <li>• LOW (ενεργοποίηση όταν η κατάσταση του pin που αντιστοιχεί στο συγκεκριμένο interrupt γίνει LOW)</li> <li>• RISING (όταν από LOW γίνει HIGH)</li> <li>• FALLING (όταν από HIGH γίνει LOW)</li> </ul>

				• CHANGE (όταν αλλάξει κατάσταση γενικά)
<b>detachInterrupt</b>	Εντολή	-	<i>(interrupt)</i>	Απενεργοποιεί το συγκεκριμένο <i>interrupt</i> .
<b>noInterrupts</b>	Εντολή	-	()	Σταματά προσωρινά την λειτουργία όλων των <i>interrupt</i>
<b>interrupts</b>	Εντολή	-	()	Επαναφέρει την λειτουργία των <i>interrupt</i> που διακόπηκε προσωρινά από μια εντολή <i>noInterrupts</i> .
<b>Serial.begin</b>	Μέθοδος κλάσης	-	<i>(datarate)</i>	Θέτει τον ρυθμό μεταφοράς δεδομένων του σειριακού interface (σε baud)
<b>Serial.println</b>	Μέθοδος κλάσης	-	<i>(data)</i>	Διοχετεύει τα δεδομένα <i>data</i> για αποστολή μέσω του σειριακού interface. Η παράμετρος <i>data</i> μπορεί να είναι είτε αριθμός είτε αλφαριθμητικό.

Επιπλέον, στην γλώσσα του Arduino κάθε πρόγραμμα αποτελείται από δύο βασικές ρουτίνες , ώστε να έχει την γενική δομή:

```
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Σχημα 2.8 - Οι δυο βασικές ρουτίνες του Arduino IDE

Η βασική ρουτίνα `setup()` εκτελείται μια φορά μόνο κατά την εκκίνηση του προγράμματος , ενώ η βασική ρουτίνα `loop()` περιέχει τον βασικό κορμό του προγράμματος και η εκτέλεσή της επαναλαμβάνεται συνέχεια σαν ένας βρόγχος `while (true)`.



## 2.6 BLUETOOTH

### 2.6.1 ΙΣΤΟΡΙΑ ΤΟΥ BLUETOOTH

Το Bluetooth κατασκευάστηκε το 1998 από μία ομάδα εταιριών (Special Interest Group, SIG), με επικεφαλής την εταιρία L.M.Ericsson, ενώ συμμετείχαν και οι εταιρίες Nokia, IBM και Toshiba.

Το Bluetooth ανήκει στην κατηγορία των δικτύων προσωπικής περιοχής, γνωστά ως PAN(Personal Area Network) και έχει ως στόχο την ασύρματη διασύνδεση μεταξύ υπολογιστικών και επικοινωνιακών συσκευών.

### 2.6.2 ΕΦΑΡΜΟΓΕΣ ΤΟΥ BLUETOOTH

Το Bluetooth προσφέρει μια πληθώρα εφαρμογών, μερικές εκ των οποίων είναι οι εξής:

- Ασύρματη διασύνδεση επιτραπέζιων υπολογιστών (desktop) και φορητών υπολογιστών(laptop).
- Δυνατότητα μεταφοράς αρχείων (φωτογραφίες, αρχεία mp3) μεταξύ συμβατών κινητών συσκευών.
- Ασύρματα ακουστικά σε κινητά τηλέφωνα.
- Ασύρματες συσκευές όπως πληκτρολόγιο, εκτυπωτές, σαρωτές, ποντίκι, φαξ.
- Bluetooth Car Kit: Δυνατότητα των κινητών τηλεφώνων να εκτελούν κάποιες λειτουργίες του αυτοκινήτου, ενώ υπάρχει και αντίστοιχο σύστημα για κράνη μηχανής.

### 2.6.3 ΕΚΔΟΣΕΙΣ ΤΟΥ BLUETOOTH

Από την πρώτη εμφάνιση του Bluetooth μέχρι και σήμερα έχουν βελτιστοποιηθεί πολλά χαρακτηριστικά του, τα οποία ενσωματώνονται στις παρακάτω εκδόσεις:



*Σχήμα 2.8 – Λογότυπο Bluetooth*

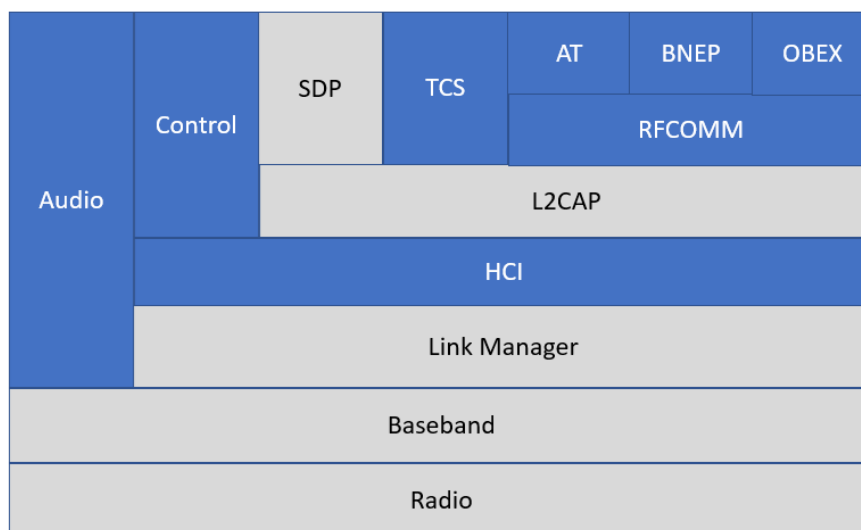
Έτος	Έκδοση
1999	1
2002	1.1
2003	1.2
2004	2
2007	2.1
2009	3
2010	4
2013	4.1
2014	4.2
2016	5

*Πίνακας Εκδόσεων Bluetooth*

Η πρώτη έκδοση Bluetooth που κυκλοφόρησε ήταν η 1.0 τον Ιούλιο του 1999 με αρκετά προβλήματα κυρίως στην επικοινωνιακή ζεύξη συσκευών που φαινομενικά ήταν συμβατές. Αυτά τα προβλήματα λύθηκαν με την βελτιωμένη έκδοση 1.1 (2002) η οποία με τη σειρά της ήρθε να προσθέσει το χαρακτηριστικό της κρυπτογραφίας αλλά και της μέτρησης ισχύος του λαμβανόμενου σήματος. Το 2003 κυκλοφόρησε η επόμενη έκδοση 1.2 με αρκετές καινοτομίες όπως η αναπήδηση συχνότητας για την αποφυγή παρεμβολών έτσι έγινε δυνατή η μεταφορά δεδομένων σε μεγαλύτερη ταχύτητα. Η έκδοση 2.0 κυκλοφόρησε το 2004 και είναι και η έκδοση που ενσωματώνεται στο module Bluetooth HC-05 της πτυχιακής μας εργασίας, σε αυτή την έκδοση συναντάμε για πρώτη φορά το EDR (Enhanced Data Rate) το οποίο επιτρέπει σε πολλές συζευγμένες συσκευές να χρησιμοποιούν το διαθέσιμο εύρος ζώνης. Στη διάρκεια συγγραφής της παρούσας πτυχιακής εργασίας κυκλοφορεί η έκδοση 5.0 με ταχύτητες μεταφοράς δεδομένων της τάξης των 2 Mbps και μεγαλύτερη κάλυψη απόστασης. Σημαντική επίσης είναι και η διαχείριση ενέργειας που γίνεται στην έκδοση αυτή ώστε να μπορεί να έχει μεγαλύτερη διάρκεια ζωής η μπαταρία των φορητών συσκευών που χρησιμοποιούν το Bluetooth.

#### 2.6.4 ΔΟΜΗ ΤΟΥ BLUETOOTH

Το Bluetooth αποτελείται από δύο μέρη: τον πυρήνα του Bluetooth και τα προφίλ του. Ο πυρήνας περιλαμβάνει μια στοίβα επιπέδων (Σχήμα 2.9), με σκοπό τη δημιουργία κοινού επιπέδου στη σύνδεση των δεδομένων και στο φυσικό επίπεδο για τις διάφορες εφαρμογές. Επίσης διασυνδέει πρωτόκολλα υψηλού επιπέδου τα οποία μπορούν να επικοινωνούν με το Bluetooth.



Σχήμα 2.9 – Δομή επιπέδων πυρήνα Bluetooth

Στον πίνακα που ακολουθεί περιγράφουμε συνοπτικά τα τμήματα από τα οποία αποτελείται ο πυρήνας του Bluetooth.

Επίπεδο	Περιγραφή
Radio	ασύρματη ραδιοζεύξη για την αποστολή και λήψη των δεδομένων
BaseBand	διαμόρφωση πλαισίων(frames), έλεγχος ροής, συγχρονισμός και έλεγχος συνδέσεων συσκευών bluetooth
Link Management	εκτέλεση πρωτοκόλλου διαχείρισης σύνδεσης
L2CAP	δυνατότητα πολύπλεξης πρωτοκόλλων, λειτουργίες κατάτμησης και επανασυναρμολόγηση/αφαίρεση ομάδων
SDP	ενημέρωση συσκευής bluetooth για ενημερώσεις και λήψη πληροφοριών για τις γειτονικές συσκευές
RFCOMM	εκτέλεση πρωτοκόλλου προσομείωσης ελέγχου σειριακής γραμμής RS-232 και σηματοδοσίας δεδομένων
TCS	καθορισμός διαδικασιών σηματοδοσίας ελέγχου κλήσεων
HCI	παροχή πρόσβασης στις δυνατότητες υλικού bluetooth

Το Bluetooth είναι στην ουσία ένα βιομηχανικό πρότυπο που χρησιμεύει στη δημιουργία ασύρματων προσωπικών δικτύων υπολογιστών (Wireless Personal Area Networks, WPAN). Επιτρέπει την μικροκυματική μικρής εμβέλειας ασύρματη ζεύξη μεταξύ ψηφιακών συσκευών, δημιουργώντας ένα πρωτόκολλο επικοινωνίας το οποίο παρέχει προτυποποιημένη, ασύρματη επικοινωνία ανάμεσα σε συσκευές τύπου: κινητά τηλέφωνα, τηλεοράσεις, φορητοί υπολογιστές, tablets, προσωπικοί υπολογιστές, εκτυπωτές, καθώς και ψηφιακές φωτογραφικές μηχανές ή ψηφιακές κάμερες, μέσω μιας ασφαλούς, φθηνής και παγκοσμίως διαθέσιμης χωρίς ειδική άδεια ραδιοσυχνότητας μικρής εμβέλειας.

Τεχνικά το Bluetooth ανήκει στο φυσικό επίπεδο και είναι ένα πρωτόκολλο ασύρματης δικτύωσης, υποεπίπεδο MAC και, προαιρετικά, υποεπίπεδο LLC. Οι προδιαγραφές του Bluetooth καθορίζουν την «ασύρματη» τεχνολογία χαμηλού κόστους και χαμηλής ισχύος, που εξαλείφει τα καλώδια μεταξύ των κινητών συσκευών και επιτρέπει τη διασύνδεσή τους. Το Bluetooth λειτουργεί στο

«αδέσμευτο» φάσμα συχνοτήτων των 2,45 GHz (ISM, Industrial, Scientific and Medical) ώστε οι συσκευές που το ενσωματώνουν να μπορούν να λειτουργήσουν σε οποιοδήποτε σημείο του πλανήτη. Για να περιοριστούν στο ελάχιστο οι παρεμβολές από παρεμφερείς συσκευές, το Bluetooth εκμεταλλεύεται τη full-duplex επικοινωνία, καθώς και την αναπήδηση συχνότητας - frequency hopping - (έως και 1600 hops ανά δευτερόλεπτο). Το Bluetooth επιτρέπει τις απευθείας συνδέσεις από συσκευή προς συσκευή (point to point), καθώς και την ταυτόχρονη σύνδεση έως και 7 (slave) συσκευών (και 1 ακόμη που λειτουργεί ως κεντρική - master) με τη χρήση μιας μοναδικής συχνότητας.

Τις προδιαγραφές της συγκεκριμένης τεχνολογίας, ανέπτυξε και υποστηρίζει το Bluetooth Special Interest Group, ενώ η τελευταία «δημόσια» έκδοση τους είναι η 2.0. Αξιοσημείωτο γνώρισμα της τεχνολογίας Bluetooth είναι η δυνατότητα αναβάθμισης και επέκτασής της, ώστε να μπορεί να ενσωματωθεί σε νέα προϊόντα. Το Bluetooth SIG ονομάζει αυτές τις επεκτάσεις «Προφίλ» (profiles) και ήδη έχει παρουσιάσει αρκετά για διάφορες «αγορές» (εκτύπωσης, φωτογραφίας, αυτοκίνητο, δικτύωσης κ.α.).

#### 2.6.5 BLUETOOTH ΔΙΑΤΑΞΗ (MODULE) HC-06

Ένα ρομποτικό σύστημα που διαθέτει τέσσερις τροχούς έχει τεράστιο εύρος δυνατοτήτων, όπως π.χ. η δυνατότητα σύνδεσης από μακρινή απόσταση με διάφορων ειδών προγράμματα, όπως επίσης με κινητά τηλέφωνα, υπολογιστές και άλλες συσκευές επικοινωνίας. Με αυτόν τον τρόπο μπορούμε να επεξεργαζόμαστε τα δεδομένα που λαμβάνουμε από το σύστημά μας, ενώ παράλληλα μπορούμε και να το ελέγξουμε χειροκίνητα εξ 'αποστάσεως. Αλλάζοντας την διάταξη του Bluetooth σε HC-06 module, μπορούμε να χειριστούμε το όχημά μας εξ 'αποστάσεως. Το HC-06 είναι ένας οδηγός που χρησιμοποιούμε σε Bluetooth με σειριακές θύρες και στηρίζεται στο πασίγνωστο μοντέλο επικοινωνίας Master-Slave. Ένα Master module είναι συμβατό με -παραπάνω από μία- συσκευές που βρίσκονται σε

λειτουργία Slave. Επειδή έχουμε μόνο ένα Bluetooth ως master, θα χρησιμοποιούμε το μοντέλο HC-06 σε ρόλο slave. Ο Arduino Uno παρέχει στο Bluetooth τάση 5 Vcc.



Σχήμα 2.10 – Bluetooth module HC-06

Περιέχει ακόμα άλλα 3 pins :

- το GND που πηγαίνει στη γείωση,
- το pin TXD που πηγαίνει στον Πομπό (Transmitter) και
- το pin RXD πηγαίνει στον δεκτή (Receiver).

Η συνδεσμολογία του βαίνει ως εξής:

- το TXD πηγαίνει στο ποδαράκι 0 (Receiver),
- το pin RXD πηγαίνει στο ποδαράκι 1 (Transmitter).

## 2.7 GPS

Από την αρχή της παρουσίας του, το ανθρώπινο είδος είχε την ανάγκη να ταξιδεύει είτε από περιέργεια αλλά κυρίως για την εύρεση τροφής. Συνεπώς είναι έμφυτη η ανάγκη να γνωρίζει διαρκώς τη θέση του, τον προορισμό του αλλά κυρίως δρόμο της επιστροφής προς την αφετηρία του. Τα πρώτα στοιχεία που χρησιμοποίησε ο άνθρωπος ήταν η θέση του Ήλιου στον ορίζοντα, η θέση των αστεριών στον νυχτερινό ουρανό και στη συνέχεια διάφορα όργανα που κατασκεύασε όπως ο αστρολάβος και τη συνέχεια η πυξίδα.

Η πρώτη καινοτομία στο χώρο του εντοπισμού θέσεως έγινε από το σύστημα ραδιοεντοπισμού LORAN. Το LORAN (Long Range Navigation) ξεκίνησε τη λειτουργία του στις αρχές της δεκαετίας του 50 και τα λειτουργικά του μέρη ήταν τέσσερις πομποί, οι οποίοι κάλυπταν μία περιοχή περίπου 500 μιλίων, αυτή η λειτουργική μονάδα ονομαζόταν αλυσίδα.

Η εκάστοτε αλυσίδα του συστήματος εκπέμπει ραδιοκύματα σε μία μοναδική συχνότητα στην οποία συντονίζεται πομπός και δέκτης για να μετρήσει την απόσταση ο δέκτης τη μεταξύ τους απόσταση. Η μόνη λειτουργία που έπρεπε να κάνει ο χρήστης κατά τη διάρκεια της διαδρομής του, ήταν ο συντονισμός του με τον δέκτη του στη συχνότητα που εξέπεμπε η κάθε αλυσίδα που κάλυπτε την περιοχή στην οποία βρισκόταν.

Στη συνέχεια η ευκολία τοποθέτησης δορυφόρων σε τροχιά γύρω από τη γη επέτρεψε τη μετάβαση από ένα σύστημα επίγειων πομπών σε ένα σύστημα με δορυφορικούς πομπούς.

Το Παγκόσμιο Σύστημα Εντοπισμού Θέσης (Global Positioning System- GPS), αποτελείται από ένα "πλέγμα" 24 συνολικά δορυφόρων που έχουν τεθεί σε τροχιά γύρω από τη Γη, πάνω στους οποίους είναι συνδεδεμένες ειδικές συσκευές, γνωστές ως δέκτες GPS. Οι δέκτες αυτοί παρέχουν στον χρήστη πληροφορίες με μεγάλη ακρίβεια για τη θέση του, την ταχύτητά του, το υψόμετρό του, ή ακόμα και την κατεύθυνση της κίνησής του. Η πληροφορία της θέσης συχνά χρησιμοποιείται από λογισμικά πλοήγησης ώστε να απεικονίζουν τη διαδρομή από ένα σημείο-αφετηρία έως το τελικό προορισμό. Η κεντρική ιδέα για τη δημιουργία του συστήματος GPS προήλθε από τον στρατό των Ηνωμένων Πολιτειών της Αμερικής, με σκοπό τον προσδιορισμό μεμονωμένων στρατιωτικών μονάδων με ακρίβεια αρχικά της τάξεων των δέκα μέτρων.

### 2.7.1 GPS NEO-6M ΠΕΡΙΓΡΑΦΗ ΛΕΙΤΟΥΡΓΙΑΣ

Ο δέκτης που χρησιμοποιήσαμε στα πλαίσια της παρούσας πτυχιακής εργασίας είναι ο GPS NEO-6m. Ο δέκτης αυτός ανήκει στη σειρά NEO-6 η οποία περιλαμβάνει μια οικογένεια αυτόνομων δεκτών GPS με χαρακτηριστικό γνώρισμα την ενσωμάτωση της μηχανής u-blox 6 και ταυτόχρονα τη διατήρηση του κόστους του σε χαμηλά επίπεδα. Η χρήση των δεκτών της οικογένειας NEO-6 ενδείκνυται για φορητές συσκευές οι οποίες απαιτούν χαμηλή κατανάλωση ενέργειας και μικρό μέγεθος κατασκευής, καθώς οι διάσταση των δεκτών είναι μόλις 16x12,2x2,4mm.



Σχήμα 2.11 – module GPS NEO-6M

Η u-blox 6 ενσωματώνει 50 κανάλια η χρήση των οποίων ελαχιστοποιεί το χρόνο πρώτου κλειδώματος (Time-To-First-Fix TTFX) σε χρόνο κάτω του ενός δευτερολέπτου. Ο μηχανισμός αναζήτησης και εύρεσης δορυφορικού σήματος βασίζεται σε περίπου δύο εκατομμύρια συσχετιστές σήματος οι οποίοι διατηρούν σταθερή την επικοινωνία με το δορυφορικό σήμα. Ο καινοτόμος σχεδιασμός και τεχνολογία καταστέλλει τις πηγές θορύβου και μετριάξει τα φαινόμενα πολλαπλών



διαδρομών, παρέχοντας στους δέκτες GPS NEO-6 εξαιρετική απόδοση πλοήγησης, ακόμη και στα πιο απαιτητικά περιβάλλοντα.

#### 2.7.1.1 ΠΡΩΤΟΚΟΛΛΑ ΚΑΙ ΔΙΕΠΑΦΕΣ

Πρωτόκολλα	Τύπος
<b>NMEA</b>	Input/output, ASCII, 0183, 2.3 (συμβατό με 3.0)
<b>UBX</b>	Input/output, binary, u-blox proprietary
<b>RTCM</b>	Input, 2.3

Όλα τα αναφερόμενα πρωτόκολλα είναι διαθέσιμα σε UART, USB και DDC. Οι δέκτες NEO-6 ενσωματώνουν διεπαφή UART για σειριακή επικοινωνία ή εναλλακτικά μία διεπαφή USB 2.0 FS μέγιστου διαύλου 12Mbps. Η αντίσταση έλξης στο USB\_DP είναι ενσωματωμένη για να σηματοδοτήσει μια συσκευή πλήρους ταχύτητας στον κεντρικό υπολογιστή. Η καρπίτσα VDDUSB παρέχει τη διεπαφή USB. Το u-blox παρέχει πιστοποιημένο πρόγραμμα οδήγησης USB πιστοποιημένο από Microsoft® για λειτουργικά συστήματα Windows XP, Windows Vista και Windows 7. Η διεπαφή SPI επιτρέπει τη σύνδεση εξωτερικών συσκευών με σειριακή διασύνδεση. Η διεπαφή μπορεί να λειτουργήσει σε λειτουργία master ή slave. Στη master λειτουργία, ένα σήμα επιλογής τσιπ είναι διαθέσιμο για την επιλογή εξωτερικών slaves. Στη λειτουργία slave ένα σήμα επιλογής ενός τσιπ επιτρέπει την επικοινωνία με τον κεντρικό υπολογιστή. Το μέγιστο εύρος ζώνης είναι 100kbit / s.

### 2.7.1.2 ΧΡΟΝΙΣΜΟΣ ΑΚΡΙΒΕΙΑΣ

*Λειτουργία χρόνου:* Το NEO-6T παρέχει μια ειδική λειτουργία χρόνου για την παροχή υψηλότερης ακρίβειας χρονισμού. Το NEO-6T έχει σχεδιαστεί για χρήση με σταθερές διατάξεις κεραίας. Η Λειτουργία Χρόνου διαθέτει τρεις διαφορετικές ρυθμίσεις που περιγράφονται στον Πίνακα 3: Απενεργοποιημένος, Έλεγχος-Είσοδος και Σταθερή Λειτουργία. Για τη βέλτιστη απόδοση εισάγετε τη θέση της κεραίας (όταν είναι γνωστή) συνιστάται ως πιθανή πηγή σφαλμάτων θα μειωθεί.

<b>Disabled Mode</b>	<b>τυπική λειτουργία PVT</b>
<b>Survey-In Mode</b>	Ο δέκτης GPS υπολογίζει τη μέση θέση σε μια εκτεταμένη χρονική περίοδο έως ότου επιτευχθεί μια προκαθορισμένη μέγιστη τυπική απόκλιση. Στη συνέχεια, ο δέκτης θα οριστεί αυτόματα σε Fixed Mode και θα ενεργοποιηθούν οι λειτουργίες χρονισμού
<b>Fixed Mode</b>	Σε αυτή τη λειτουργία, λαμβάνεται μια σταθερή θέση 3D και γνωστή τυπική απόκλιση και ενεργοποιούνται οι λειτουργίες χρονισμού. Η Fixed Mode μπορεί είτε να ενεργοποιηθεί απευθείας με τροφοδοσία προκαθορισμένων συντεταγμένων θέσης (ECEF- Earth Centered Earth Fixed format) ή πραγματοποιώντας μια Survey-In. Στην Fixed Mode, εξαλείφονται τα σφάλματα χρονισμού στο σήμα TIMEPULSE τα οποία διαφορετικά προκύπτουν από σφάλματα τοποθέτησης, ενώ παρέχεται και υποστήριξη μέσω δορυφόρου.

*Συχνότητα αναφοράς και παραγωγή παλμοσειράς:* Το NEO-6T έρχεται με έξοδο χρονικής ώσης, η οποία μπορεί να ρυθμιστεί από 0,25 Hz έως 10 MHz. Η χρονική ώθηση μπορεί είτε να χρησιμοποιηθεί για συγχρονισμό χρόνου (δηλαδή 1 παλμό ανά δευτερόλεπτο) είτε ως συχνότητα αναφοράς στην περιοχή MHz. Μια χρονική ώθηση στην περιοχή MHz παρέχει εξαιρετική μακροχρόνια ακρίβεια και σταθερότητα συχνότητας.

*Χρονική ακρίβεια:* Το NEO-6T μπορεί να χρησιμοποιηθεί για ακριβείς μετρήσεις χρόνου με ανάλυση μικρότερου του δευτερολέπτου χρησιμοποιώντας την εξωτερική διακοπή (EXTINT0). Οι ανερχόμενες και οι πτέρυγες άκρες αυτών των σημάτων χρονολογούνται στο χρόνο GPS και UTC και υπολογίζονται. Η λειτουργία "Χρονική σήμανση" μπορεί να ενεργοποιηθεί με το μήνυμα UBX-CFG-TM2.

*Κεραία:* Το module NEO-6 είναι σχεδιασμένο για χρήση με παθητικές και ενεργές κεραίες.

Παράμετροι	Προσδιορισμός						
Τύπος κεραίας	Παθητικές και Ενεργές κεραίες						
Προτάσεις Ενεργών Κεραίων	<table> <tr> <td>Ελάχιστο κέρδος απώλειας σήματος στο καλώδιο RF)</td> <td>15 dB (για αντιστάθμιση</td> </tr> <tr> <td>Μέγιστο κέρδος</td> <td>50 dB</td> </tr> <tr> <td>Μέγιστο μέγεθος θορύβου</td> <td>1,5 dB</td> </tr> </table>	Ελάχιστο κέρδος απώλειας σήματος στο καλώδιο RF)	15 dB (για αντιστάθμιση	Μέγιστο κέρδος	50 dB	Μέγιστο μέγεθος θορύβου	1,5 dB
Ελάχιστο κέρδος απώλειας σήματος στο καλώδιο RF)	15 dB (για αντιστάθμιση						
Μέγιστο κέρδος	50 dB						
Μέγιστο μέγεθος θορύβου	1,5 dB						

### 2.7.1.3 ΔΙΑΧΕΙΡΙΣΗ ΕΝΕΡΓΕΙΑΣ

Οι δέκτες u-blox υποστηρίζουν διαφορετικές λειτουργίες ισχύος. Αυτοί οι τρόποι λειτουργίας αντιπροσωπεύουν στρατηγικές για τον έλεγχο των κινητήρων απόκτησης και παρακολούθησης προκειμένου να επιτευχθεί η καλύτερη δυνατή απόδοση ή καλή απόδοση με μειωμένη κατανάλωση ενέργειας.

Για περισσότερες πληροφορίες σχετικά με τις στρατηγικές διαχείρισης ενέργειας, ανατρέξτε στην περιγραφή u-blox 6 Receiver, συμπεριλαμβανομένης της προδιαγραφής πρωτοκόλλου.

*Λειτουργία μέγιστης απόδοσης:* Κατά τη διάρκεια μιας ψυχρής εκκίνησης, ένας δέκτης στη λειτουργία μέγιστης απόδοσης (Maximum Performance Mode - MPM) εκπέμπει συνεχώς τη μηχανή απόκτησης για να αναζητήσει όλους τους δορυφόρους. Μόλις ο δέκτης έχει σταθεροποίηση θέσης (ή εάν υπάρχουν διαθέσιμες πληροφορίες προ-τοποθέτησης θέσης), ο κινητήρας εξαγοράς συνεχίζει να χρησιμοποιείται για την αναζήτηση όλων των ορατών δορυφόρων που δεν παρακολουθούνται.

*Οικολογική Λειτουργία:* Κατά τη διάρκεια μιας ψυχρής εκκίνησης, ένας δέκτης στο Eco Mode λειτουργεί ακριβώς όπως στη λειτουργία μέγιστης απόδοσης. Μόλις υπολογιστεί μια θέση και παρακολουθείται ένας επαρκής αριθμός δορυφόρων, ο κινητήρας εξαγοράς απενεργοποιείται με αποτέλεσμα σημαντική εξοικονόμηση ενέργειας. Ο κινητήρας παρακολούθησης παρακολουθεί συνεχώς τους δορυφόρους που έχουν αποκτήσει και αποκτά άλλους διαθέσιμους ή αναδυόμενους δορυφόρους.

*Λειτουργία εξοικονόμησης ενέργειας:* Η λειτουργία εξοικονόμησης ενέργειας (Power Save Mode-PSM) επιτρέπει τη μείωση της κατανάλωσης ενέργειας του συστήματος, ενεργοποιώντας και απενεργοποιώντας επιλεκτικά τμήματα του δέκτη.

### 2.7.1.4 GPS PERFORMANCE

Parameter		Specification		
Receiver type	50 Channels GPS L1 frequency, C/A Code SBAS: WAAS, EGNOS, MSAS			
Time-To-First-Fix		<i>NEO-6G/Q/T</i>	<i>NEO-6M/V</i>	<i>NEO-6P</i>
	Cold Start	26 s	27 s	32 s
	Warm Start	26s	27 s	32 s
	Hot Start	1 s	1 s	1 s
	Aided Starts	1 s	<3 s	<3 s
Sensitivity		<i>NEO-6G/Q/T</i>	<i>NEO-6M/V</i>	<i>NEO-6P</i>
	Tracking & Navigation	-162 dBm	-161 dBm	-160 dBm
	Reacquisition	-160 dBm	-160 dBm	-160 dBm
	Cold Start (without aiding)	-148 dBm	-147 dBm	-146 dBm
	Hot Start	-147 dBm	-146 dBm	-145 dBm
Maximum Navigation update rate		<i>NEO-6G/Q/M/T</i>	<i>NEO-6P/V</i>	
		5 Hz	1 Hz	
Horizontal position accuracy	GPS	2.5 m		
	SBAS	2.0 m		
	SBAS+PPP	< 1 m (2D, R50)		
	SBAS+PPP	< 2 m (3D, R50)		
Configurable Timepulse frequency range		<i>NEO-6G/Q/M/P/V</i>	<i>NEO-6T</i>	
		0.25 Hz to 1 kHz	0.25 Hz to 10 MHz	
Accuracy for Timepulse signal	RMS	30 ns		
	99%	<60 ns		
	Granularity	21 ns		
	Compensated	15 ns		
Velocity accuracy		0.1m/s		
Heading accuracy		0.5 degrees		
Operational Limits	Dynamics	≤4 g		
	Altitude	50,000 m		
	Altitude	500 m/s		

### 2.7.1.5 ABSOLUTE MAXIMUM RATINGS

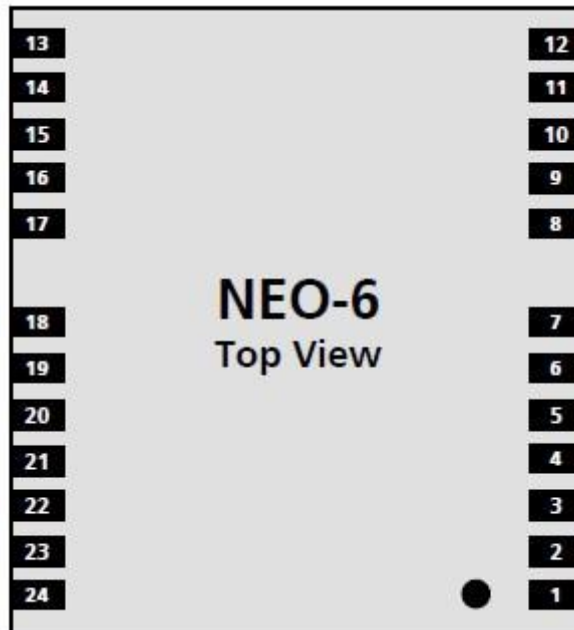
Parameter	Symbol	Module	Min	Max	Units	Condition
Power supply voltage	VCC	NEO-6G	-0.5	2.0	V	
		NEO-6Q, 6M, 6P, 6V, 6T	-0.5	3.6	V	
Backup battery voltage	V_BCKP	All	-0.5	3.6	V	
USB supply voltage	VDDUSB	All	-0.5	3.6	V	
Input pin voltage	V <sub>in</sub>	All	-0.5	3.6	V	
	V <sub>in_usb</sub>	All	-0.5	$V_{DDU_{SB}}$	V	
DC current through any digital I/O pin (except supplies)	I <sub>pin</sub>	All		10	mA	
VCC_RF output current	ICC_RF	All		100	mA	
Input power at RF_IN	Pr <sub>fin</sub>	NEO-6Q, 6M, 6G, 6V, 6T		15	dBm	source impedance=50, continuous wave
		NEO-6P		5	dBm	
Storage temperature	T <sub>stg</sub>	All	-40	-85	°C	

### 2.7.1.6 ΣΥΝΘΗΚΕΣ ΛΕΙΤΟΥΡΓΙΑΣ

Όλες οι προδιαγραφές καταγράφονται σε συνθήκες θερμοκρασίας δωματίου 25°C.

Parameter	Symbol	Module	Min	Typ	Max	Units	Condition
Power supply voltage	VCC	NEO-6G	1.75	1.8	1.95	V	
		NEO-6Q/M NEO-6P/V/T	2.7	3.0	3.6	V	
Supply voltage USB	VDDUSB	All	3.0	3.3	3.6	V	
Backup battery voltage	V_BCKP	All	1.4	22	3.6	μA	V_BCKP = 1.8 V, VCC = 0V
Backup battery current	I_BCKP	All			VCC	V	
Input pin voltage range	V <sub>in</sub>	All	0		0.2*VCC	V	
Digital IO Pin Low level input voltage	V <sub>il</sub>	All	0		VCC	V	
Digital IO Pin High level input voltage	V <sub>ih</sub>	All	0.7*VCC		0.4	V	I <sub>ol</sub> =4mA
Digital IO Pin Low level output voltage	V <sub>ol</sub>	All				V	I <sub>oh</sub> =4mA
Digital IO Pin High level output voltage	V <sub>oh</sub>	All	VCC - 0.4				
USB_DM, USB_DP	V <sub>inU</sub>	All	Compatible with USB with 22 Ohms series resistance				
VCC_RF voltage	VCC_RF	All	VCC-0.1			V	
VCC_RF output current	ICC_RF	All	50			mA	
Antenna gain	G <sub>ant</sub>	All	50			dB	
Receiver Chain Noise Figure	NF <sub>tot</sub>	All	3.0			dB	
Operating temperature	Topr	All	-40		85	°C	

### 2.7.1.7 ΚΑΤΑΜΕΡΙΣΜΟΣ ΕΡΓΑΣΙΑΣ PIN



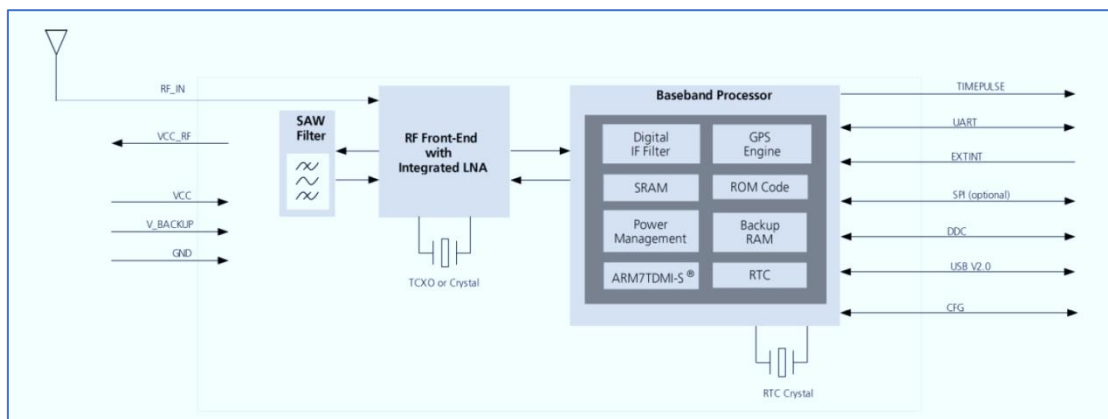
Σχήμα 2.12 – Άνω όψη του GPS NEO-6M

No	Module	Name	I/O	Description
1	All	Reserved	I	Reserved
2	All	SS_N	I	SPI Slave Select
3	All	TIMEPULSE	O	Timepulse (1PPS)
4	All	EXTINT0	I	External Interrupt Pin
5	All	USB_DM	I/O	USB Data
6	All	USB_DP	I/O	USB Data
7	All	VDDUSB	I	USB Supply
8	All	Reserved	O	See Hardware Integration Manual
				Pin 8 and 9 must be connected together
9	All	VCC_RF	O	Output Voltage RF section
				Pin 8 and 9 must be connected together
10	All	GND	I	Ground
11	All	RF_IN		GPS signal input
12	All	GND	I	Ground
13	All	GND	I	Ground



14	All	MOSI/CFG_COM0	I/O	SPI MOSI / Configuration Pin.
				Leave open if not used.
15	All	MISO/CFG_COM1	I	SPI MISO / Configuration Pin.
				Leave open if not used.
16	All	CFG_GPS0/SCK	I	Power Mode Configuration Pin / SPI Clock.
				Leave open if not used.
17	All	Reserved	I	Reserved
18	All	SDA2	I/O	DDC Data
19	All	SCL2	I/O	DDC Clock
20	All	TxD1	O	Serial Port 1
21	All	RxD1	I	Serial Port
22	All	V_BCKP	I	Backup voltage supply
23	All	VCC	I	Supply voltage
24	All	GND	I	Ground

### 2.7.1.8 ΜΠΛΟΚ ΔΙΑΓΡΑΜΜΑ



Σχήμα 2.12 – Μπλοκ διάγραμμα GPS NEO-6M

## 2.7.2 ΤΡΟΠΟΣ ΛΕΙΤΟΥΡΓΙΑΣ

Το σύστημα GPS μετράει την απόσταση μεταξύ ενός δέκτη και του πλησιέστερου δορυφόρου, ωστόσο το σήμα του και μόνο δεν φτάνει για τον εντοπισμό της θέσης, αφού το σήμα ενός δορυφόρου απλώς φτιάχνει πάνω στη Γη έναν νοητό κύκλο.

Ο χρήστης ενός δέκτη GPS μπορεί να βρίσκεται σε οποιοδήποτε σημείο πάνω στη διάμετρο αυτού του κύκλου, κάτι φυσικά που δεν καθορίζει με οποιαδήποτε ακρίβεια το στίγμα του. Όταν λάβουμε σήμα ενός δεύτερου δορυφόρου, εμφανίζεται ένας ακόμα κύκλος, και κάποια τμήματά του συμπίπτουν με τον πρώτο κύκλο. Η θέση μας πάνω στη Γη βρίσκεται σε κάποιο από τα σημεία όπου ενώνονται οι δύο νοητοί κύκλοι, ωστόσο πρόκειται και πάλι για μια αόριστη προσέγγιση. Με το σήμα ενός τρίτου δορυφόρου δημιουργείται ένας τρίτος κύκλος, ο οποίος και θα συμπίπτει με τους άλλους δύο, αυξάνοντας την ακρίβεια της θέσης μας. Το σημείο τομής των τριών κύκλων έχει τώρα πάρα πολύ μικρή έκταση και ακριβώς σε αυτό το σημείο βρισκόμαστε εμείς.

Παρότι η σύνδεση ενός δέκτη με τρεις δορυφόρους επαρκεί προκειμένου να λάβουμε τις απαραίτητες πληροφορίες που χρειαζόμαστε, για μεγαλύτερη ακρίβεια και αποφυγή τυχών σφαλμάτων ή ακόμα και αποκλίσεων, ένας δέκτης GPS πάντα θα συνδέεται με 5 δορυφόρους, δημιουργώντας κάθε φορά ισάριθμους νοητούς κύκλους. Με αυτόν τον τρόπο η πιθανότητα να εντοπίσουμε λάθος το στίγμα μας κάθε φορά κυριολεκτικά εκμηδενίζεται.

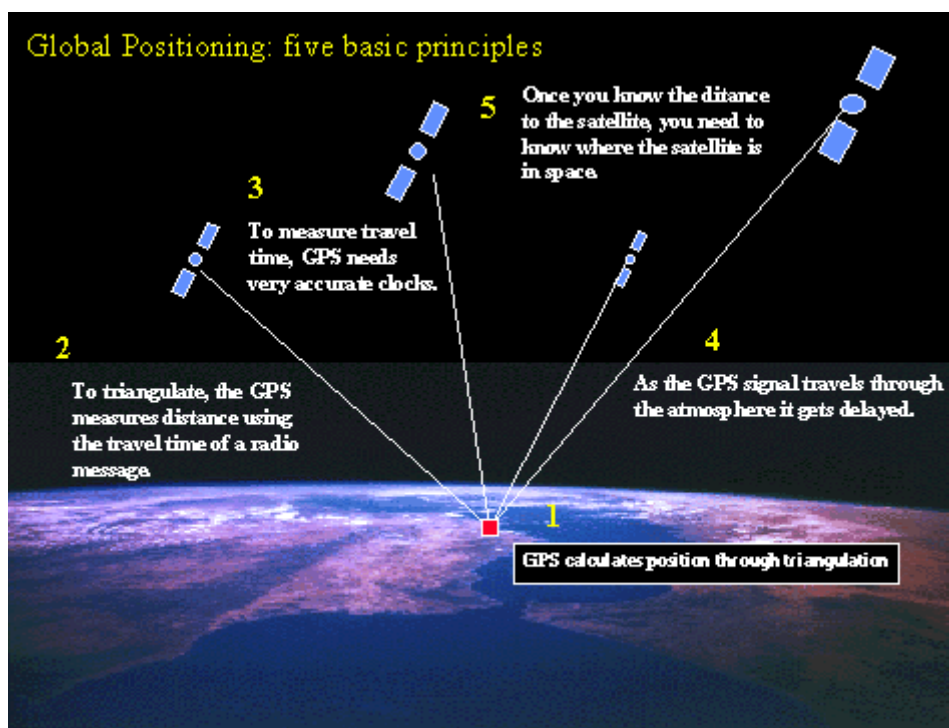
Επειδή το σήμα μεταξύ του κάθε δέκτη GPS και εκάστοτε δορυφόρου είναι πρακτικά ένα ηλεκτρομαγνητικό κύμα, δηλαδή ταξιδεύει με την ταχύτητα του φωτός (300.000 χιλιόμετρα/ δευτ/πτο). Κατά συνέπεια, το σήμα κάθε φορά θα χρειάζεται μεταξύ 7 μέχρι 10 εκατοστών του δευτερολέπτου για να πάει από τον δέκτη στον δορυφόρο και το αντίστροφο.

Στο σημείο τομής των 5 κύκλων θα βρίσκεται η ακριβής θέση μας, την οποία το GPS αναπαριστά με τη εξής μορφή: γεωγραφικό μήκος, γεωγραφικό πλάτος και υψόμετρο. Επειδή σφάλμα της τάξεως ακόμα και του ενός δεκάτου του δευτερολέπτου της επικοινωνίας του δέκτη με το δορυφόρο είναι αρκετό για να μας δώσει ανακριβείς πληροφορίες, σε περίπτωση παρέκκλισης στο χρονόμετρο ενός δορυφόρου αυτός βγαίνει άμεσα εκτός λειτουργίας και στη συνέχεια αντικαθίσταται με κάποιον άλλο από τους διαθέσιμους εφεδρικούς δορυφόρους, αποκαθιστώντας το σφάλμα.

### 2.7.3 ΛΕΙΤΟΥΡΓΙΚΑ ΜΕΡΗ

Το GPS κατασκευάστηκε από την εταιρεία Rockwell International και εκτοξεύτηκε από το ακρωτήριο Canaveral στην πολιτεία Φλόριντα των Ηνωμένων Πολιτειών Αμερικής, ενώ τροφοδοτείται μέσω των ηλιακών συλλεκτών που είναι ενσωματωμένοι στην επιφάνειά του. Το σύστημα GPS καλύπτει τις ανάγκες για εντοπισμό θέσης σε παγκόσμια κλίμακα, λόγω αυτού του μεγάλου εύρους κάλυψης χρειάζεται να διαιρεθεί σε μικρότερα τμήματα, προκειμένου να εκτελούνται όλες οι λειτουργίες του, καθώς και να γίνεται καλύτερος έλεγχος και συντονισμός. Τα τρία τμήματα από τα οποία αποτελείται είναι τα: δορυφορικό τμήμα, τμήμα επίγειου ελέγχου και τμήμα τελικού χρήστη.

Το δορυφορικό τμήμα περιλαμβάνει συνολικά 31 δορυφόρους από τους οποίους πρέπει να είναι ενεργοί οι 24 κατά το 95% του χρόνου λειτουργίας τους. Οι δορυφόροι αυτοί καλύπτουν ομοιόμορφα ολόκληρη την υδρόγειο σε ύψος περίπου 20.200 km, εκτελώντας έτσι δύο γήινες περιστροφές στη διάρκεια της ημέρας.



Σχήμα 2.13 – Αρχή λειτουργίας συστήματος GPS

Οι δορυφόροι αντιμετωπίζουν πολύ συχνά προβλήματα στη λειτουργία τους. Τέτοιου είδους προβλήματα είναι οι ανεπιθύμητες και επικίνδυνες μεταβολές του ύψους της τροχιάς τους, προβλήματα που αφορούν την αδιάλειπτη παροχή ηλεκτρικής ενέργειας στα κυκλώματα τους, αλλά σφάλματα που εμφανίζονται στο σύστημα χρονομέτρησης τους. Η αντιμετώπιση και επίλυση αυτών των καταστάσεων είναι απαραίτητη ώστε να φτάνει σωστά η πληροφορία στον τελικό χρήστη του συστήματος.

Το τμήμα επίγειου ελέγχου περιλαμβάνει ένα επανδρωμένο και τέσσερα μη-επανδρωμένα κέντρα, με τον κεντρικό σταθμό να βρίσκεται στο Κολοράντο των Ηνωμένων Πολιτειών Αμερικής, και τα υπόλοιπα στη νήσο Χαβάη στον Ανατολικό Ειρηνικό Ωκεανό, στη νήσο Ascension στον Ατλαντικό Ωκεανό, στο σταθμό Diego Garcia στον Ινδικό Ωκεανό και στην Kwajalein στο Δυτικό Ειρηνικό Ωκεανό.

Το τρίτο και τελευταίο τμήμα στην αλυσίδα των τμημάτων του GPS είναι το τμήμα τελικού χρήστη. Το τμήμα αυτό περιλαμβάνει όλες τις τερματικές διατάξεις λήψης σήματος GPS.

## Κεφάλαιο 3<sup>ο</sup> ΚΙΝΗΤΗΡΕΣ ΣΥΝΕΧΟΥΣ ΡΕΥΜΑΤΟΣ

### 3.1 ΕΙΣΑΓΩΓΗ

Οι ηλεκτρικοί κινητήρες αποτελούν μια κατηγορία στρεφόμενων ηλεκτρικών μηχανών. Το βασικό στοιχείο μιας στρεφόμενης ηλεκτρικής μηχανής είναι η μετατροπή ενέργειας από ηλεκτρική σε μηχανική μορφή και αντίστροφα. Υπάρχουν τρεις τρόποι λειτουργίας των ηλεκτρικών μηχανών: η λειτουργία τους ως κινητήρες, ως γεννήτριες και ως πέδες.

Στην ηλεκτροτεχνία οι κινητήρες και οι γεννήτριες ρεύματος είναι μηχανήματα αντίστοιχης δομής και αντίστροφης λειτουργίας. Σε μια γεννήτρια συνεχούς ρεύματος η μηχανική ενέργεια μετατρέπεται σε ηλεκτρική και αυτό εκφράζεται με την παραγωγή ηλεκτρικού ρεύματος, ενώ στους κινητήρες το ηλεκτρικό ρεύμα μετατρέπεται σε μηχανική ενέργεια.

Η λειτουργία τόσο των ηλεκτρικών γεννητριών όσο και των ηλεκτρικών κινητήρων στηρίζεται στη ηλεκτρομαγνητική επαγωγή. Με βάση λοιπόν το φαινόμενο της επαγωγής όταν ένας αγωγός (δηλαδή ένα αγώγιμο ηλεκτρικά υλικό) κινείται μέσα σε ένα μαγνητικό πεδίο τότε μέσα στον αγωγό αναπτύσσεται ηλεκτρεγερτική δύναμη από επαγωγή δηλαδή ηλεκτρικό δυναμικό το οποίο είναι και το αίτιο εμφάνισης ηλεκτρικού ρεύματος στον αγωγό. Στους κινητήρες αντίστοιχα αξιοποιείται ένα άλλο φαινόμενο της ηλεκτρομαγνητικής επαγωγής.

Όταν ένας αγωγός που διαρρέεται από ηλεκτρικό ρεύμα βρεθεί μέσα σε ένα μαγνητικό πεδίο τότε στον αγωγό αυτό ασκείται από το μαγνητικό πεδίο μια δύναμη που τείνει να τον κινήσει. Η δύναμη αυτή είναι ανάλογη με την ένταση του ηλεκτρικού ρεύματος, την ένταση του μαγνητικού πεδίου, αλλά και το μήκος του αγωγού. Σημειώνεται πως η φορά της ασκούμενης στον αγωγό δύναμης αντιστρέφεται είτε αν αλλάξει η φορά του ρεύματος, είτε αν αντιστραφεί η πολικότητα του μαγνητικού πεδίου.

Ο ηλεκτρικός κινητήρας παράγει κίνηση, εφαρμόζοντας ηλεκτρικό ρεύμα σε πηνία τα οποία βρίσκονται μέσα σε μαγνητικό πεδίο. Ο ενιαίος ηλεκτρικός κινητήρας χρησιμοποιεί συνεχές ή εναλλασσόμενο ρεύμα.

Ένας ηλεκτρικός κινητήρας αποτελείται από τα εξής μέρη:

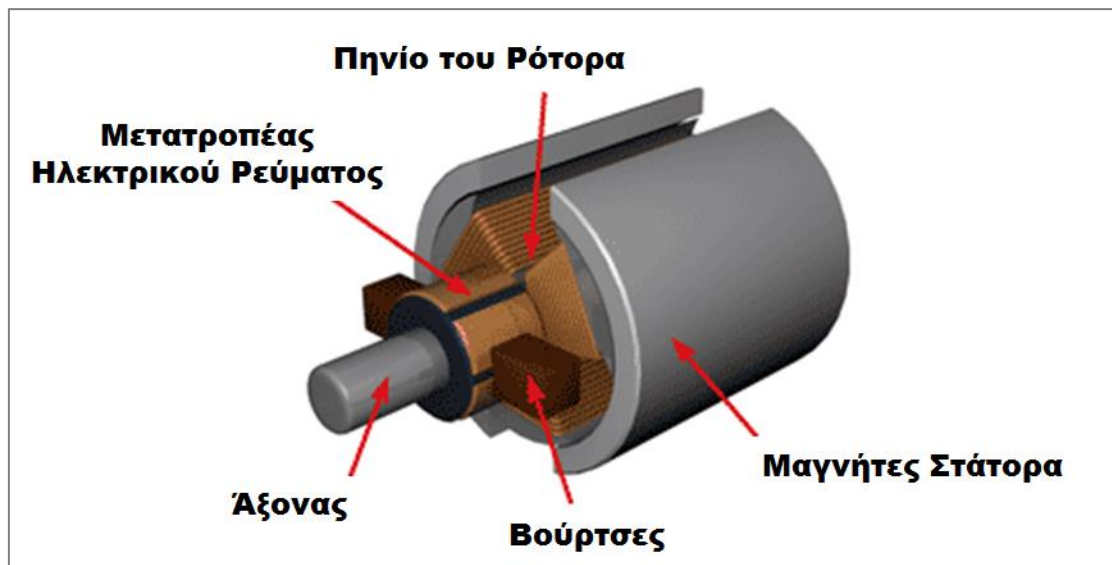
**Στάτορας:** Ο στάτορας αποτελείται από πηνία που τροφοδοτούνται με ηλεκτρικό ρεύμα. Αυτό παράγει το μαγνητικό πεδίο, που ασκεί δύναμη στρέψης στα πηνία του ρότορα.

**Ρότορας:** Ο ρότορας αποτελείται από πολλά πηνία. Καθώς περιστρέφεται, κάθε πηνίο διαρρέεται εναλλάξ με ρεύμα από τις ψήκτρες.

**Ψήκτρα:** Οι ψήκτρες από γραφίτη εξασφαλίζουν την ηλεκτρική σύνδεση του εναλλάκτη.

**Εναλλάκτης:** Ο εναλλάκτης (ή μεταλλάκτης) διαθέτει μια σειρά μεταλλικών συνδέσεων πάνω σε έναν κύλινδρο. Οι ψήκτρες παρέχουν ρεύμα σε κάθε σύνδεση, εναλλάξ, καθώς ο εναλλάκτης περιστρέφεται.

Ένας ηλεκτρομαγνήτης, ο στάτορας, παράγει το μαγνητικό πεδίο. Ο ρότορας (οπλισμός) αποτελείται από διαφορετικά συρμάτινα πηνία. Οι ψήκτρες τροφοδοτούν το ένα πηνίο με ρεύμα. Το μαγνητικό του πεδίο αλληλεπιδρά με αυτό του στάτορα και ο ρότορας στρέφεται ελαφρά. Επειδή κάθε πηνίο τροφοδοτείται με γρήγορη διαδοχή, ο ρότορας περιστρέφεται.



Σχήμα 3.1 - Τα εσωτερικά Μέρη ενός DC κινητήρα

### 3.1.1 ΑΡΧΗ ΛΕΙΤΟΥΡΓΙΑΣ

Κατά τη λειτουργία μιας ηλεκτρικής μηχανής σαν κινητήρα παρέχεται ηλεκτρική ενέργεια στο κύριο τύλιγμα της μηχανής, που λέγεται τύλιγμα τυμπάνου και αποδίδεται μηχανική ενέργεια εξόδου πάνω σε μια περιστρεφόμενη άτρακτο. Μια εξωτερικά επιβαλλόμενη ηλεκτρική τάση  $V$  οδηγεί ένα ρεύμα  $I$  μέσα στο τύλιγμα τυμπάνου ενάντια σε μια εσωτερικά επαγόμενη αντιηλεκτρεγερτική δύναμη  $e$ . Το τύλιγμα τυμπάνου γίνεται έτσι ικανό να απορροφά ηλεκτρική ενέργεια με ρυθμό  $e * I$ . Το πεδίο ζεύξεως ασκεί μια στιγμιαία ηλεκτρομαγνητική ροπή  $T_{πεδ}$  πάνω στο περιστρεφόμενο μέλος της μηχανής που λέγεται δρομέας (ρότορας). Αν ο δρομέας στρέφεται με γωνιακή ταχύτητα  $\omega$  τότε η στιγμιαία ισχύς εξόδου που αποδίδεται στην άτρακτο θα είναι  $T_{πεδ} * \omega$ . Η εξωτερικά εφαρμοζόμενη ροπή φορτίου  $T_{εξ}$  δρα με φορά αντίστροφη από εκείνη της περιστροφής και καθιστά το φορτίο ικανό ν' απορροφά μηχανική ενέργεια. Ισχύει προφανώς ότι:

$$T_{πεδ} - T_{εξ} = J * \frac{d\omega}{dt}$$

Όπου  $J$  είναι η ροπή αδρανείας του δρομέα και του μηχανικού του φορτίου ( η κίνηση του οποίου είναι ο τελικός σκοπός της ύπαρξης του κινητήρα ). Όταν  $T_{πεδ} = T_{εξ}$  τότε  $d\omega/dt = 0$  και η μηχανή περιστρέφεται με σταθερή γωνιακή ταχύτητα. Στη μόνιμη κατάσταση λειτουργίας έχουμε ότι:

$$(\omega * T_{πεδ})_{μέση\ τιμή} = (e * i)_{μέση\ τιμή}$$

Δεδομένου ότι στο τύλιγμα τυμπάνου αναπτύσσεται μια ηλεκτρεγερτική δύναμη(ΗΕΔ), χρειαζόμαστε ένα μαγνητικό πεδίο διεγέρσεως που συνήθως (εκτός από την περίπτωση μόνιμου μαγνήτη στις μικρές μηχανές), στην πράξη παρέχεται από τυλίγματα διεγέρσεως ή τυλίγματα πεδίου. Στη μόνιμη λειτουργία των συνήθων μηχανών της πράξης, οι οποίες έχουν χωριστά τροφοδοτούμενο τύλιγμα διέγερσης, η μέση τιμή της ισχύος που τροφοδοτεί το τύλιγμα αυτό δαπανάται υπό μορφή θερμότητας. Κατά τη λειτουργία μιας ηλεκτρικής μηχανής ως γεννήτριας παρέχεται ηλεκτρική ενέργεια

στην άτρακτο της ηλεκτρικής μηχανής από μια πρωτεύουσα κινητήρια μηχανή και ηλεκτρική ενέργεια εξόδου είναι διαθέσιμη στους ακροδέκτες του τυλίγματος τυμπάνου. Η πρωτεύουσα κινητήρια μηχανή μπορεί να είναι για παράδειγμα, είτε ένας ατμοστρόβιλος σ' ένα θερμικό σταθμό παραγωγής, είτε ένας υδροστρόβιλος σε έναν υδροηλεκτρικό σταθμό παραγωγής, είτε μια εμβολοφόρα μηχανή εσωτερικής καύσεως σ' ένα απομονωμένο ηλεκτροπαραγωγό ζεύγος. Τέλος κατά τη λειτουργία μιας ηλεκτρικής μηχανής σαν πέδης, η μηχανή τροφοδοτείται και με μηχανική και με ηλεκτρική ενέργεια. Η ολική ενέργεια εισόδου χάνεται μέσα στη μηχανή με μορφή απωλειών και έτσι η μηχανή λειτουργεί σαν πέδη και φρενάρι.

Η λειτουργία του κινητήρα βασίζεται και στη μαγνητική δύναμη μεταξύ ρευματοφόρων αγωγών. Η αλληλεπίδραση μεταξύ των ρευμάτων των αγωγών έχει σαν αποτέλεσμα την εμφάνιση μαγνητικής δύναμης και επομένως και ροπής. Με αυτόν τον τρόπο προκαλείται η περιστροφή του άξονα του κινητήρα. Η αποδιδόμενη ισχύς  $P$  ενός κινητήρα (το έργο που παράγεται από τον στρόφαλο (άξονα), ανά μονάδα χρόνου) ισούται με το γινόμενο της ροπής  $\tau$  επί της γωνιακής ταχύτητας  $\omega$  του άξονα, δηλαδή:

$$P = \omega * \tau$$

Ο κάθε κινητήρας μπορεί να αποδώσει έως μία δεδομένη μέγιστη ισχύ, η οποία και ορίζεται από τις τεχνικές προδιαγραφές του. Έτσι όταν το motor shield στείλει την κατάλληλη συνεχή τάση στον αγωγό του ηλεκτροκινητήρα τότε αυτός θα αρχίζει να περιστρέφεται.

Στην περίπτωση Δυστυχώς οι απλοί ηλεκτροκινητήρες δεν δίνουν απεριόριστες δυνατότητες στο όχημα μας. Αυτό οφείλεται στο ότι οι απλοί ηλεκτροκινητήρες έχουν την δυνατότητα να περιστρέφονται μόνο με μία σταθερή ταχύτητα ανά λεπτό . Σε αντίθεση οι βηματικοί κινητήρες και οι servo κινητήρες χρησιμοποιώντας ειδικές βιβλιοθήκες του arduino όπως είναι η stepper.h και η servo.h έχουν την δυνατότητα να ρυθμίσουν τις στροφές ανά λεπτό με σκοπό τον καλύτερο έλεγχο του οχήματος.

Έτσι για παράδειγμα όταν και οι δύο κινητήρες περιστρέφονται προς την ίδια φορά αλλά με μία διαφορετική ταχύτητα τότε το όχημα θα αναγκαστεί να πραγματοποιήσει μία κίνηση κυκλική και όχι ευθεία, όπου η ακτίνα της θα είναι ανάλογη της διαφοράς των στροφών των κινητήρων. Το μοναδικό μειονέκτημα είναι ότι αυξάνεται κατά πολύ το κόστος της κατασκευής εξαιτίας του ότι οι servo κινητήρες και οι βηματικοί κινητήρες είναι πολύ πιο ακριβοί ενώ απαιτούν και πιο πολύπλοκα συστήματα οδήγησης.



### 3.1.2 ΚΑΤΗΓΟΡΙΕΣ

Οι ηλεκτροκινητήρες διακρίνονται σε δύο βασικές κατηγορίες στους κινητήρες συνεχούς ρεύματος και στους κινητήρες εναλλασσόμενου ρεύματος, οι οποίοι καλύπτουν και την πλειοψηφία των εφαρμογών. Οι κινητήρες εναλλασσόμενου ρεύματος με τη σειρά τους διακρίνονται σε μονοφασικούς και πολυφασικούς. Τόσο οι μονοφασικοί όσο και οι πολυφασικοί κινητήρες διακρίνονται σε σύγχρονους κινητήρες και σε κινητήρες επαγωγής ή ασύγχρονους.

Ένας σύγχρονος κινητήρας αποτελείται από μια σειρά τριών τυλιγμάτων στο στάτορα με ένα απλό στρεφόμενο μέρος. Καθώς το ρεύμα που περνάει από το πηνίο μεταβάλλεται ο κινητήρας εργάζεται ομαλά μόνο στη συχνότητα του ημιτονοειδούς ρεύματος επιτυγχάνοντας μια λειτουργία με σταθερή ταχύτητα από μηδενικό ως πλήρες φορτίο λειτουργίας. Στους ασύγχρονους κινητήρες το ηλεκτρικό ρεύμα χρησιμεύει κυρίως για να επάγει την περιστροφή των τυλιγμάτων παρά για να περιστρέφει ευθέως τον άξονα.

Οι πολυφασικοί κινητήρες εναλλασσόμενου ρεύματος είναι κατά βάση μηχανές σταθερής ταχύτητας, αλλά διαφοροποιούνται ως προς κάποια σχεδιαστικά στοιχεία τους με συνέπεια να διαμορφώνονται τέσσερις βασικές υποκατηγορίες των κινητήρων εναλλασσόμενου ρεύματος. Στον πιο συνήθη τύπο (DESIGN A,B) έχουμε κανονικές τιμές ροπής και ρεύματος εκκίνησης και χαμηλή ολίσθηση. Στον δεύτερο τύπο (DESIGN C) έχουμε υψηλή ροπή εκκίνησης με κανονική ένταση ρεύματος εκκίνησης και χαμηλή ολίσθηση. Στον τρίτο τύπο (DESIGN D) έχουμε επίσης υψηλή ροπή εκκίνησης αλλά χαμηλό ρεύμα εκκίνησης, ενώ η ολίσθηση είναι υψηλή. Στον τέταρτο τύπο (DESIGN F) έχουμε χαμηλή ροπή και ρεύμα εκκίνησης αλλά και χαμηλή ολίσθηση.

Οι κινητήρες συνεχούς ρεύματος διακρίνονται σε δύο βασικές κατηγορίες σε αυτήν όπου το μαγνητικό πεδίο παράγεται από ένα μόνιμο μαγνήτη και αυτούς όπου το πεδίο παράγεται από ένα τύλιγμα διεγέρσεως. Στους κινητήρες της πρώτης κατηγορίας η μαγνητική ροή παραμένει σταθερή σε όλες τις ταχύτητες του κινητήρα και οι χαρακτηριστικές καμπύλες ταχύτητας - ροπής και έντασης ρεύματος - ροπής είναι γραμμικές. Η δεύτερη κατηγορία χωρίζεται σε τρεις υποκατηγορίες. Στους κινητήρες παράλληλης διέγερσης, στους κινητήρες διέγερσης εν σειρά και στους κινητήρες σύνθετης διέγερσης.

Στους κινητήρες παράλληλης διέγερσης το τύλιγμα διέγερσης συνδέεται παράλληλα με το τύλιγμα του επαγωγικού τυμπάνου. Στους κινητήρες διέγερσης εν σειρά το τύλιγμα διέγερσης συνδέεται εν σειρά με το τύλιγμα του επαγωγικού τυμπάνου. Στους κινητήρες σύνθετης διέγερσης κάθε κύριος μαγνητικός πόλος έχει δύο τυλίγματα το παράλληλο τύλιγμα και το τύλιγμα εν σειρά.

### 3.2 ΚΙΝΗΤΗΡΕΣ DC

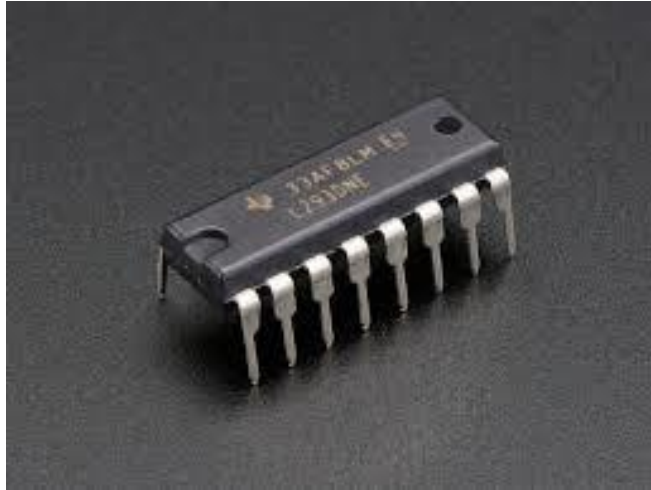
Οι ηλεκτροκινητήρες προορίζονται για χρήση στα 6 V. Σε γενικές γραμμές, αυτά τα είδη των κινητήρων μπορούν να λειτουργήσουν σε τάσεις πάνω και κάτω από αυτήν την ονομαστική τάση. Έτσι λειτουργούν άνετα στην περιοχή τάσης 3 - 9 V. Οι χαμηλότερες τάσεις ενδέχεται να μην είναι πρακτικές, όπως και οι υψηλότερες τάσεις θα μπορούσαν να επηρεάσουν αρνητικά τη διάρκεια ζωής του κινητήρα.

Όσον αφορά λοιπόν την παρούσα εφαρμογή, οι κινητήρες που είναι υπεύθυνοι για την κίνηση των τροχών πρέπει να πληρούν, για τις ανάγκες της εργασίας, τις βασικές προϋποθέσεις. Χρειαζόμαστε συνεχούς περιστροφής κινητήρες, οι οποίοι πρέπει να έχουν τάση λειτουργίας της τάξης των 6V και ροπή ικανή να μεταφέρει το βάρος της κατασκευής χωρίς να υπάρχει μεγάλη επιβάρυνση της μπαταρίας.

Οι κινητήρες που χρησιμοποιήθηκαν για την κίνηση του οχήματος είναι συνεχούς τάσης με κιβώτιο ταχυτήτων. Ο ρόλος του κιβωτίου ταχυτήτων είναι να πολλαπλασιάζει κατά βούληση τη ροπή του κινητήρα, ώστε ανάλογα να επιτυγχάνεται καλύτερη επιτάχυνση ή μεγαλύτερη ταχύτητα κίνησης. Οι συγκεκριμένοι κινητήρες είναι ενδεδειγμένοι ως μηχανισμοί κίνησης τροχών, σε ρομποτικές εφαρμογές όπου απαιτείται η μεταφορά φορτίου με σχετικά μεγάλο βάρος.

### 3.2.1 MOTOR DRIVER

Για την οδήγηση των τεσσάρων κινητήρων του οχήματος, χρησιμοποιήθηκε το ολοκληρωμένο κύκλωμα L293D H-Bridge Motor. Είναι ένας τετραπλός οδηγός υψηλής τάσης (τύπου H) που είναι σχεδιασμένος να παρέχει αμφίδρομα ρεύματα πάνω από 1A και τάσεις από 4.5V ως 36V.

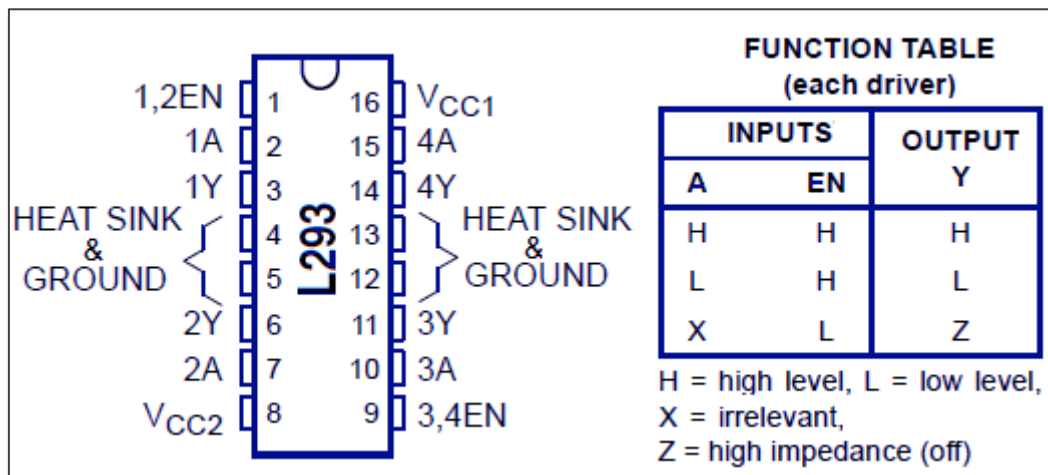


Σχήμα 3.2 - Το ολοκληρωμένο κύκλωμα L293D που χρησιμοποιήσαμε στην κατασκευή

Η συσκευή είναι σχεδιασμένη να μεταφέρει επαγωγικά φορτία σε ρελέ και διπολικούς κινητήρες καθώς και άλλα υψηλής τάσης φορτία σε διάφορες εφαρμογές. Όλες οι εισόδους του είναι συμβατές με TTL (transistor - transistor logic) και χαμηλού επιπέδου λογικής CMOS. Κάθε έξοδος (Y) είναι ένας οδηγός με μια αντίσταση Darlington και μια ψευδό-Darlington πηγή .

Οι οδηγοί ενεργοποιούνται σε ζευγάρια. Δηλαδή οι οδηγοί 1 και 2 ενεργοποιούνται με το 1,2 EN και οι οδηγοί 3 και 4 να ενεργοποιούνται με το 3,4 EN. Όταν η είσοδος είναι HIGH οι αντίστοιχοι οδηγοί ενεργοποιούνται και οι έξοδοι τους δουλεύουν συμφασικά με τις εισόδους τους. Όταν η είσοδος είναι LOW, οι οδηγοί απενεργοποιούνται και μπαίνουν σε κατάσταση υψηλής αντίστασης.

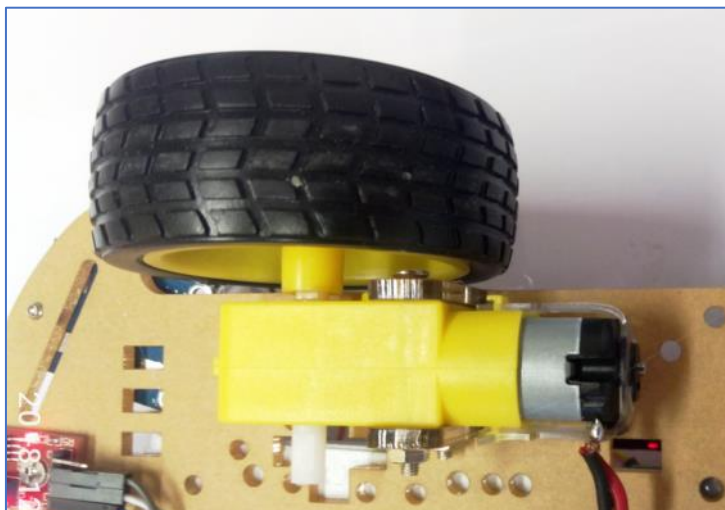
Με τις κατάλληλες εισόδους δεδομένων, κάθε ζευγάρι των οδηγών σχηματίζουν ένα πλήρες H (ή γέφυρα) το οποίο είναι κατάλληλο για την εγκατάσταση κινητήρων. Μια ξεχωριστή παροχή ρεύματος (VCC1) παρέχεται για την λογικές εισόδους και την ελαχιστοποίηση της επαγωγής του ρεύματος. Η παροχή (VCC2) χρησιμοποιείται για τα κυκλώματα εξόδου.



Σχήμα 3.3 - Πίνακας λειτουργίας για κάθε οδηγό του H-Bridge Motor Driver

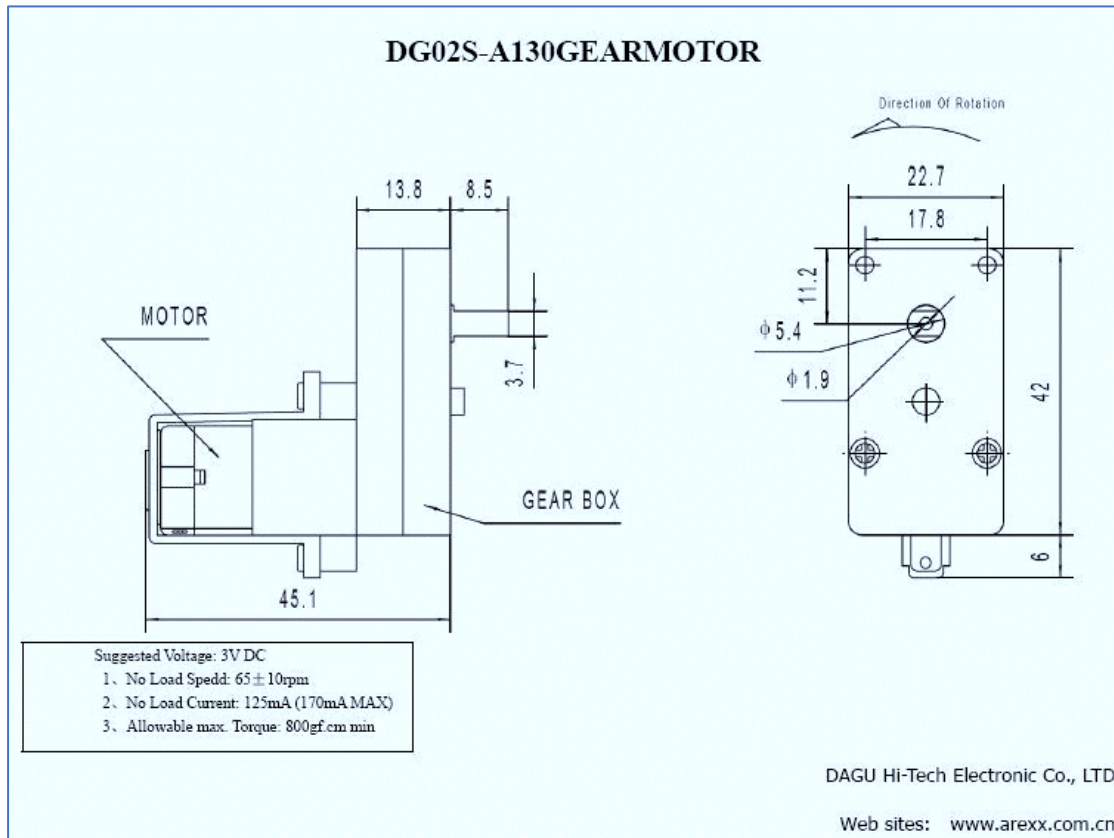
### 3.3 DC-KINHHTHPΑΣ

Στα πλαίσια της παρούσας πτυχιακής εργασίας επιλέξαμε το κινητήρα συνεχούς ρεύματος DG02S-A130GEARMOTOR (Σχήμα 3.4).



Σχήμα 3.4 – Κινητήρας DG02S-A130GEARMOTOR

Κάθε ένας από τους τέσσερις κινητήρες που χρησιμοποιήσαμε απαιτεί τάση τροφοδοσίας από 3 έως 6 Volts και ρεύμα 200mA απουσία φορτίου, με μέγιστη ταχύτητα 65rpm.



Εικόνα 3.5 – Σχηματικό κινητήρα

## Κεφάλαιο 4<sup>ο</sup> ΑΝΑΠΤΥΞΗ ΑΛΓΟΡΙΘΜΟΥ

### 4.1 ΕΙΣΑΓΩΓΗ

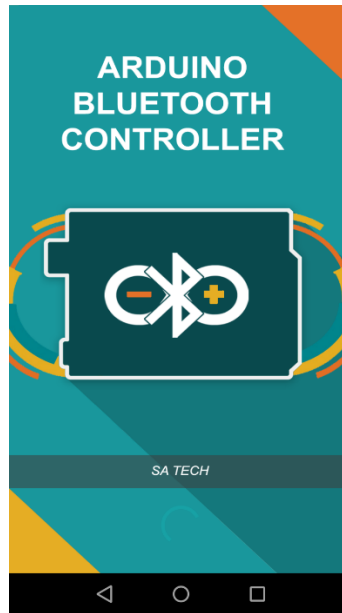
Στο παρόν κεφάλαιο ασχολούμαστε με την περιγραφή του αλγορίθμου που αναπτύξαμε για το μηχανοκίνητο όχημά μας. Ο αλγόριθμος αυτός επιτελεί δύο βασικές και ανεξάρτητες λειτουργίες: την χειροκίνητη λειτουργία (manual mode) και την αυτόματη λειτουργία (auto mode).

Η πρώτη λειτουργία αφορά τον τηλεχειρισμό του οχήματός μας από απομακρυσμένη συσκευή (κινητό τηλέφωνο, φορητός υπολογιστής) μέσω πρωτοκόλλου Bluetooth με σκοπό την χειροκίνητη οδήγησή του στην περιοχή της πύρινης εστίας και την τελική κατάσβεση της μέσω απομακρυσμένης εντολής.

Η δεύτερη λειτουργία ασχολείται με την αυτόματη ανίχνευση της πύρινης εστίας που βρίσκεται στο εγγύς πεδίο του οχήματος και την ακολούθηση πορείας από αυτό με σκοπό την κατάσβεσή της. Σε κάθε περίπτωση, είτε πρόκειται για την αυτόματη είτε για τη χειροκίνητη λειτουργία, το όχημα μας στέλνει δεδομένα της ακριβής θέσης του που λαμβάνει από το δέκτη GPS στην τερματική συσκευή με την οποία είναι σε σύζευξη.

### 4.2 ΠΕΡΙΓΡΑΦΗ – ΟΔΗΓΙΕΣ ΛΕΙΤΟΥΡΓΙΑΣ

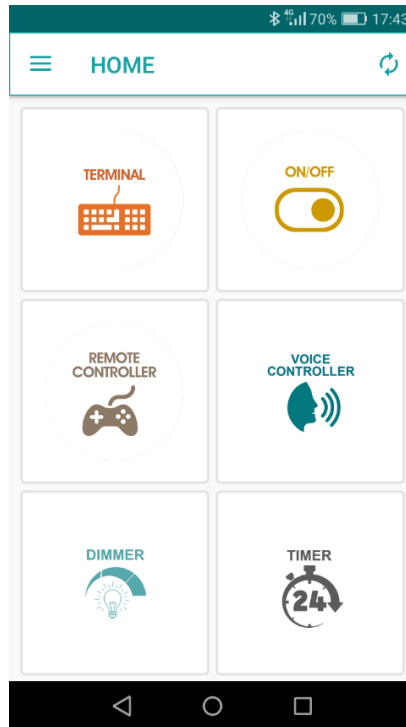
Το αυτόνομο όχημά μας αμέσως μετά την ενεργοποίηση του μεταβαίνει σε κατάσταση χειροκίνητης λειτουργίας. Για όσο χρονικό διάστημα παραμένει σε αυτή την κατάσταση, το ενδεικτικό πράσινο LED παραμένει αναμμένο. Οι πρώτες λειτουργίες που εκτελούνται από το σύστημα είναι η σύνδεση του μέσω Bluetooth με την τερματική μας συσκευή. Στα πλαίσια της παρούσας πτυχιακής εργασίας χρησιμοποιήσαμε ως τερματική συσκευή κινητό τηλέφωνο με λειτουργικό σύστημα Android 6.0, στο οποίο εγκαταστήσαμε την εφαρμογή Arduino Bluetooth Controller (Εικόνα 4.1).



*Εικόνα 4.1 – Αρχική οθόνη εφαρμογής Arduino Bluetooth Controller*

Κατά την εκκίνηση της εφαρμογής γίνεται αυτόματη ζεύξη με το Bluetooth module του οχήματος μας. Στη συνέχεια εμφανίζεται το κεντρικό μενού της εφαρμογής με τις εξής επιλογές (Εικόνα 4.2):

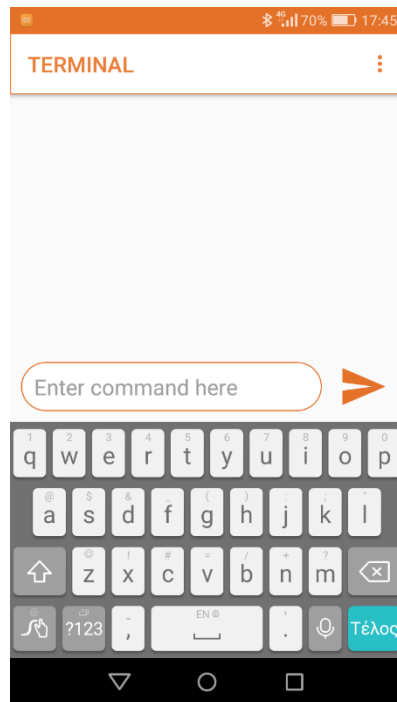
- TERMINAL
- ON/OFF
- REMOTE CONTROLLER
- VOICE CONTROLLER
- DIMMER
- TIMER



*Εικόνα 4.2 – Κεντρική οθόνη εφαρμογής Arduino Bluetooth Controller*

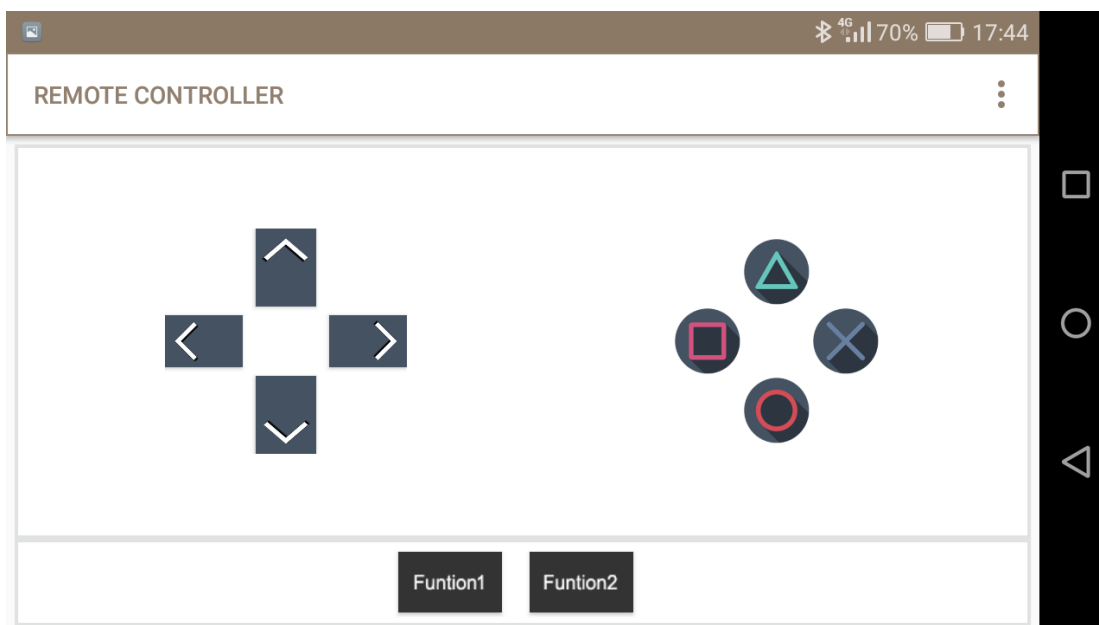
Από τις παραπάνω λειτουργίες χρησιμοποιούμε τις: Terminal (Εικόνα 4.3) για την λήψη των μηνυμάτων θέσης, ημερομηνίας, ώρας, στοιχεία πυκνότητας καπνού, έντασης φωτιάς καθώς και ενημέρωσης για την κατάσταση της συσκευής, Remote Controller για τη χειροκίνητη οδήγηση του οχήματος, για τον καθορισμό της ταχύτητάς του, αλλά και για την ενεργοποίηση του συστήματος πυρόσβεσης.





*Εικόνα 4.3 – Οθόνη τερματικού για επικοινωνία με το όχημα*

Στην επιλογή Remote Controller όπως φαίνεται και στην εικόνα 4.4 υπάρχουν 4 κουμπιά χειρισμού κατεύθυνσης κίνησης (προς τα εμπρός, προς τα πίσω, προς τα αριστερά, προς τα δεξιά), ένα κουμπί με ένα τρίγωνο με το οποίο αυξάνουμε την ταχύτητα του οχήματος, ενώ η μείωση της ταχύτητας γίνεται με το κουμπί που περιέχει τον κύκλο. Η ενεργοποίηση της αντλίας γίνεται για όση ώρα είναι πατημένο του κουμπί με το x. Τέλος, η εναλλαγή της λειτουργίας από αυτόματη σε χειροκίνητη γίνεται με το κουμπί που απεικονίζει ένα τετράγωνο.

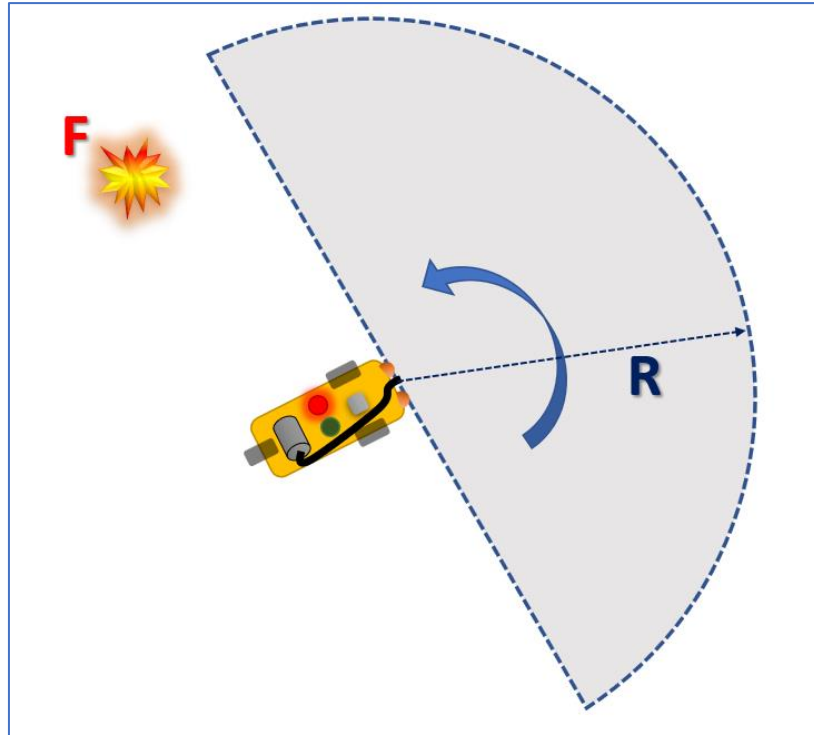


*Εικόνα 4.4 – Στιγμιότυπο οθόνης τηλεχειρισμού*

Η κάθε λειτουργία του οχήματος σηματοδοτείται από ένα LED, συγκεκριμένα κατά την αυτόματη λειτουργία ανάβει το κόκκινο LED, ενώ κατά τη χειροκίνητη απενεργοποιείται το κόκκινο και ανάβει το πράσινο.

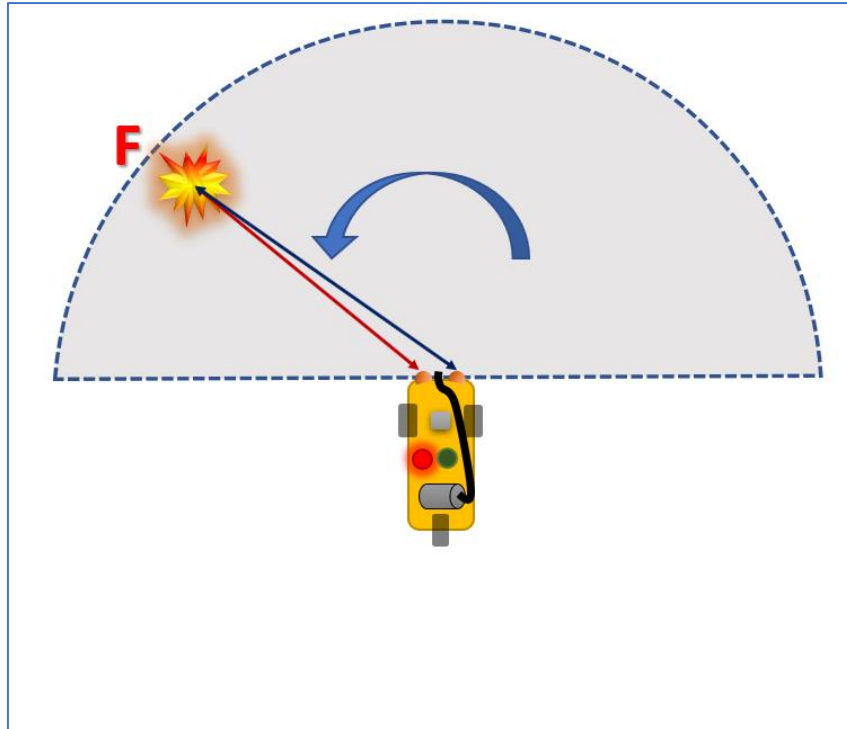
#### 4.3 ΑΛΓΟΡΙΘΜΟΣ ΑΥΤΟΜΑΤΗΣ ΛΕΙΤΟΥΡΓΙΑΣ

Στην παράγραφο αυτή περιγράφουμε τον αλγόριθμο που εκτελεί το όχημα μας για όσο χρονικό διάστημα βρίσκεται στην κατάσταση αυτόματης λειτουργίας. Έχοντας εξοπλιστεί με δύο αισθητήρες φωτιάς στο μπροστινό του τμήμα, έναν αριστερά και έναν δεξιά έχει τη δυνατότητα να ανιχνεύσει πύρινη εστία η οποία βρίσκεται σε μία ενεργό περιοχή ημικυκλίου ακτίνας  $R$  περίπου 50cm μπροστά από τους αισθητήρες όπως φαίνεται στο σχήμα 4.1. Ο αλγόριθμος εκτελεί διαρκώς τη διαδικασία ανίχνευσης ως εξής: αν δεν εντοπίσει τη φωτιά στην ενεργό περιοχή του τότε το όχημα αρχίζει να στρίβει προς τα αριστερά με σταθερό ρυθμό μέχρι που διαπιστωθεί από τους αισθητήρες ότι έχει εντοπιστεί πύρινη εστία (Σχήμα 4.2).

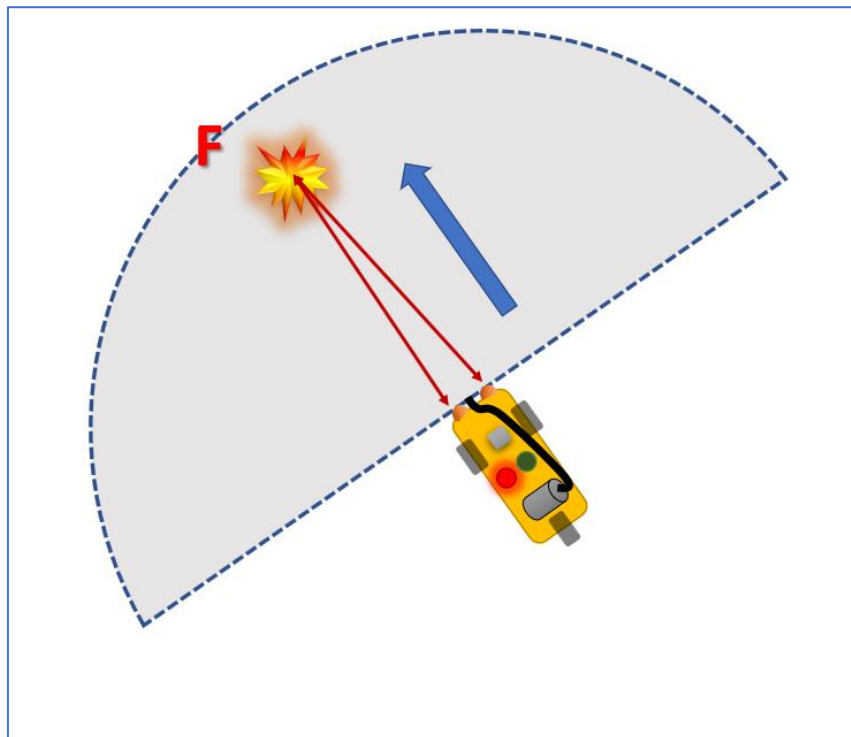


Σχήμα 4.1 – Διαδικασία ανίχνευσης πύρινης εστίας στο χώρο

Μετά τον εντοπισμό, ο αλγόριθμος περνάει στη φάση της οδήγησης του οχήματος προς την πύρινη εστία, φάση η οποία αποτελείται από δύο επιμέρους καταστάσεις. Στην πρώτη κατάσταση δίνεται εντολή στροφής στο όχημα μέχρις ότου οι τιμές που λαμβάνονται από τους δύο αισθητήρες να είναι σχεδόν ίδιες. Στο σχήμα 4.2 βλέπουμε την φάση κατά την οποία το όχημα έχει εντοπίσει τη φωτιά από τους αισθητήρες με τον αριστερό αισθητήρα να λαμβάνει μεγαλύτερη ένταση (κόκκινο βέλος), λόγω ότι βρίσκεται πιο κοντά στην εστία της φωτιάς σε σχέση με το δεξί αισθητήρα (μπλε βέλος). Έτσι, ο αλγόριθμος θα στρίψει το όχημα προς τα αριστερά, μέχρις ότου διαπιστώσει ότι οι τιμές των εντάσεων είναι ίδιες (Σχήμα 4.3).

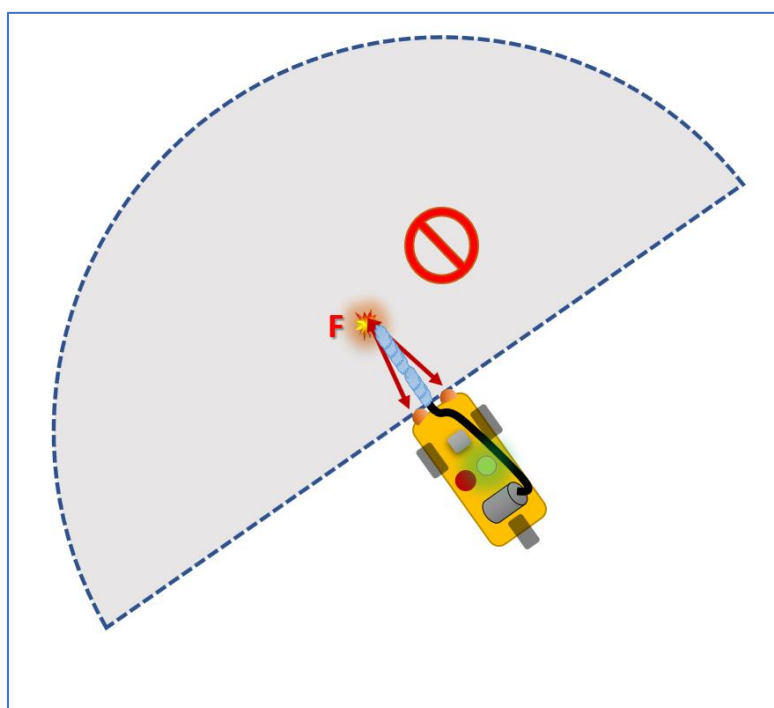


*Σχήμα 4.2-Εντοπισμός πύρινης εστίας, διαδικασία στροφής*



*Σχήμα 4.3- Οδήγηση σε ευθεία πορεία προς την πύρινη εστία*

Λαμβάνοντας πλέον ίδιες τιμές από τους αισθητήρες, ο αλγόριθμος κατευθύνει το πυροσβεστικό όχημα μέχρι το σημείο της φωτιάς σε απόσταση τέτοια ώστε να είναι ασφαλής η θερμοκρασία που αναπτύσσεται στο μπροστινό τμήμα του οχήματος, αλλά και ταυτόχρονα να φτάσει στην πύρινη εστία η υδατινή δέσμη νερού που εκτοξεύεται από την αντλία, όπως παρουσιάζεται και στο σχήμα 4.4. Ο αλγόριθμος ολοκληρώνεται μετά την κατάσβεση της αλλάζοντας τη λειτουργία του οχήματος σε χειροκίνητη.

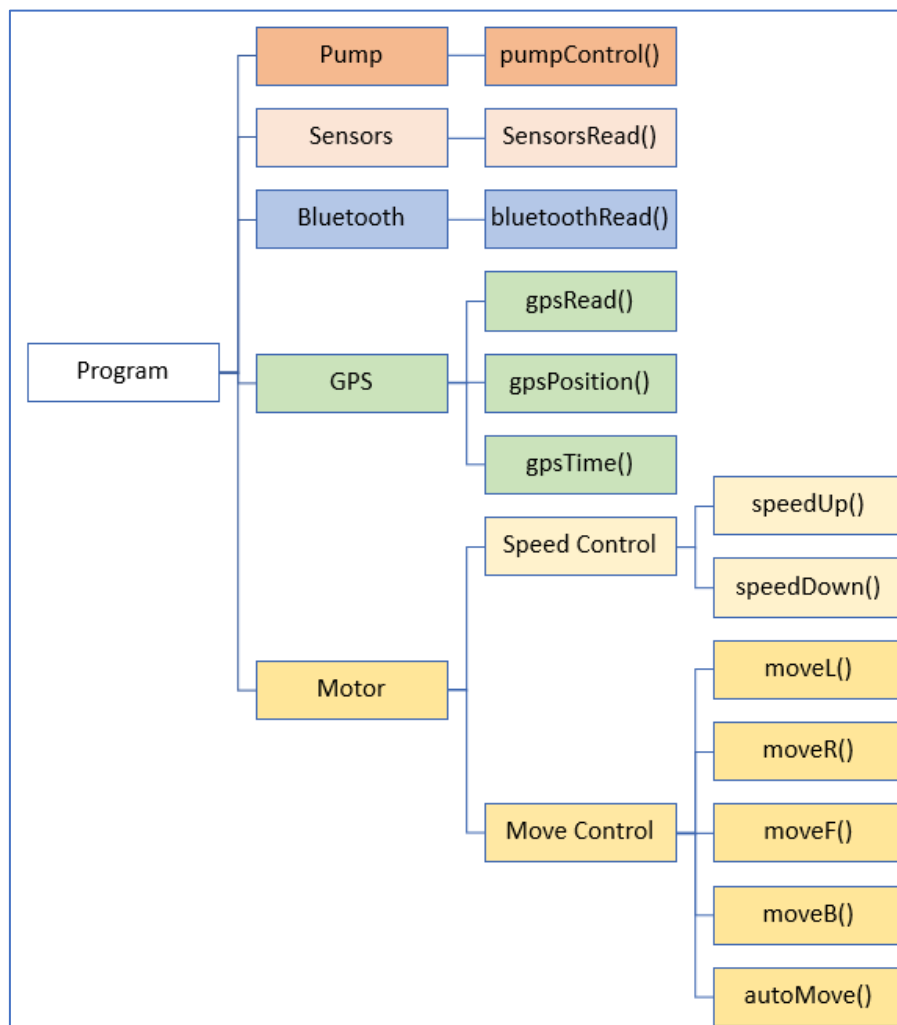


Σχήμα 4.4- Διαδικασία κατάσβεσης–απενεργοποίησης αυτόματης λειτουργίας

## Κεφάλαιο 5<sup>ο</sup> ΠΕΡΙΓΡΑΦΗ ΚΩΔΙΚΑ – ΣΥΜΠΕΡΑΣΜΑΤΑ - ΑΠΟΤΕΛΕΣΜΑΤΑ

### 5.1 ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

Για την απλούστευση της περιγραφής του πηγαίου προγράμματος που αναπτύξαμε στα πλαίσια της παρούσης πτυχιακής εργασίας θα στηριχτούμε στο επόμενο σχήμα (Σχήμα 5.1):



Σχήμα 5.1 – Δομή περιγραφή πηγαίου κώδικα

Όπως φαίνεται και στο σχήμα, σκοπός του προγράμματος είναι να ελέγχει τις πέντε βασικές μονάδες του οχήματος μέσω του μικροελεγκτή Arduino Uno. Αυτές οι μονάδες είναι η αντλία νερού(Pump), οι αισθητήρες φωτιάς και καπνού(Sensors), το Bluetooth, η μονάδα GPS και οι κινητήρες (Motor).

Για τον παραπάνω έλεγχο έχουν πραγματοποιήσει συναρτήσεις οι οποίες όπως απεικονίζονται στο σχήμα συνδέονται με τον έλεγχο της αντίστοιχης μονάδας.

#### 5.1.1 ΣΥΝΑΡΤΗΣΗ pumpControl()

Η συνάρτηση αυτή έχει ως κύριο σκοπό όταν κληθεί να ενεργοποιήσει τους κατάλληλους ακροδέκτες στους οποίους έχει συνδεθεί η αντλία με απώτερο σκοπό τη λειτουργία της για την κατάσβεση της εστίας φωτιάς που έχει ανιχνευτεί.

```
void PumpControl ()
{
  if (switchPump==1)
    digitalWrite (PUMP, 1);
  else
    digitalWrite (PUMP, 0);
}
```

#### 5.1.2 ΣΥΝΑΡΤΗΣΗ sensorsRead()

Η συνάρτηση αυτή αναλαμβάνει τη μέτρηση σε πραγματικό χρόνο των τιμών που λαμβάνουν οι αισθητήρες φωτιάς και καπνού.

```
void SensorsRead ()
{
  Left_Fire=analogRead (LF);
  Right_Fire=analogRead (RF);
  Smoke=analogRead (SM);
}
```

### 5.1.3 ΣΥΝΑΡΤΗΣΗ bluetoothRead()

Για την ανάγνωση των δεδομένων ελέγχου από τη μονάδα Bluetooth χρησιμοποιούμε την παρακάτω συνάρτηση. Η συνάρτηση αυτή έχει την εξής λειτουργικότητα: Κατά την κλήση της ελέγχει εφόσον η αντίστοιχη μονάδα είναι διαθέσιμη και προχωράει στον έλεγχο των δεδομένων που αποστέλλονται σε αυτή υπό τη μορφή ASCII χαρακτήρων. Οι βασικές τις ενέργειες είναι η εναλλαγή της λειτουργίας από αυτόματη σε χειροκίνητη και αντίστροφα, η χειροκίνητη ενεργοποίηση της αντλίας, αλλά και η απομακρυσμένη οδήγηση του οχήματος μας από το κινητό τηλέφωνο ή τον ηλεκτρονικό υπολογιστή.

```
void bluetoothRead()
{
  if (Serial.available())
  {
    BluetoothData=Serial.read();
    if (BluetoothData=='A' || BluetoothData=='a')
    {speedUp();Mode=MANUAL;}
    if (BluetoothData=='D' || BluetoothData=='d')
    {speedDown();Mode=MANUAL;}
    if (BluetoothData=='F') {moveF(del);Mode=MANUAL;}
    if (BluetoothData=='B') {moveB(del);Mode=MANUAL;}
    if (BluetoothData=='L') {moveL(del);Mode=MANUAL;}
    if (BluetoothData=='R') {moveR(del);Mode=MANUAL;}
    if (BluetoothData=='X') {switchPump*=-1;}
    if (BluetoothData=='0') Stop();
    if (BluetoothData=='M')
    {
      quitAuto=0;
      if(change_mode==1)
      {
        change_mode=0;
        if(Mode==AUTO) Mode=MANUAL;
        else if(Mode==MANUAL) Mode=AUTO;
      }
    }
  }
  else
  change_mode=1;
}
}
```



#### 5.1.4 ΣΥΝΑΡΤΗΣΗ gpsRead()

Η συνάρτηση gpsRead αναλαμβάνει τον καθολικό έλεγχο της μονάδας gps. Συγκεκριμένα ελέγχει αν υπάρχει συνδεσιμότητα με το module και εφόσον αυτή είναι εφικτή τότε με συχνότητα 1Hz διαβάζει τις συντεταγμένες θέσεις, την τρέχουσα ώρα και στέλνει τα δεδομένα αυτά προς εκτύπωση στη θύρα του σειριακού τερματικού.

```
void gpsRead()
{
  while(gpsSerial.available())
  { // check for gps data
    if(gps.encode(gpsSerial.read()))// encode gps data
    {

gps.crack_datetime(&year,&month,&day,&hour,&minute,&second,&hundredths);
      if(second!=pre_second) {tick++;pre_second=second;}
      if(tick>=DELAY)
      {
gpsPosition();
gpsTime();
tick=0;
SensorsPrint();
Serial.println();
      }

    }
  }
}
```

#### 5.1.5 ΣΥΝΑΡΤΗΣΗ gpsPosition()

Η gpsPosition(), καλεί την ενσωματωμένη συνάρτηση της βιβλιοθήκης του NEO-6M gps.get\_position(float \*lat, float \*lon) η οποία επιστρέφει στις μεταβλητές lat,lon το γεωγραφικό μήκος και πλάτος της θέσης του οχήματός μας, με ακρίβεια τεσσάρων (4) δεκαδικών ψηφίων.

```
void gpsPosition()
{
      gps.f_get_position(&lat,&lon); // get latitude and longitude
Serial.print("Lat/Lon:");
Serial.print(lat,4);
Serial.print(",");
      Serial.println(lon,4);
}
```

### 5.1.6 ΣΥΝΑΡΤΗΣΗ gpsTime()

Σημαντικό στοιχείο στη λειτουργία της μονάδας GSP είναι η ανάγνωση της τρέχουσας ώρας από τους δορυφόρους. Για τη λειτουργία αυτή πραγματώσαμε τη συνάρτηση gpsTime() η οποία κατά την κλήση της διαβάζει τις τιμές των στοιχείων Ημέρα, Μήνας, Έτος, Ώρα, Λεπτά και Δευτερόλεπτα εκχωρώντας τις τιμές αυτές στις κατάλληλες μεταβλητές. Επιπλέον στοιχείο της λειτουργίας της είναι η διόρθωση που κάνουμε για τη ζώνη ώρας που ανήκει η χώρα μας.

```
void gpsTime ()
{
    int CorrectTime=+3;
    Serial.print("Date:");
    if(day<10) Serial.print("0");
    Serial.print(day, DEC);
    Serial.print("/");
    if(month<10) Serial.print("0");
    Serial.print(month, DEC); Serial.print("/"); Serial.print(year);
    Serial.print(" Time:");
    if(hour+CorrectTime<10) Serial.print("0");
    Serial.print(hour+CorrectTime, DEC); Serial.print(":");
    if(minute<10) Serial.print("0");
    Serial.print(minute, DEC); Serial.print(":");
    if(second<10) Serial.print("0");
    Serial.println(second, DEC);
}
```

### 5.1.7 ΣΥΝΑΡΤΗΣΗ speedUp()

Για να αυξήσουμε την ταχύτητα του οχήματός μας καλούμε την συνάρτηση speedUp(). Για την ταχύτητα έχουμε ορίζει μία κλίμακα από το 1 έως και το 5, με το 5 να αντιστοιχεί στη μέγιστη δυνατή κάθε φορά. Προεπιλεγμένη τιμή της ταχύτητας κατά την εκκίνηση του κώδικα είναι η τιμή 3.

```
void speedUp ()
{
    if(Speed<5) Speed++;
    setVolt();
    Serial.print("Current Speed is:");
    Serial.println(Speed);
}
```

### 5.1.8 ΣΥΝΑΡΤΗΣΗ speedDown()

Όταν θέλουμε να μειώσουμε την ταχύτητα του οχήματός μας, καλούμε την συνάρτηση speedDown().

```
void speedDown ()
{
  if(Speed>1) Speed--;
  setVolt();
  Serial.print("Current Speed is:");
  Serial.println(Speed);
}
```

### 5.1.9 ΣΥΝΑΡΤΗΣΗ moveL()

Προκειμένου να πετύχουμε αριστερόστροφη κίνηση, καλούμε την συνάρτηση moveL(). Η συνάρτηση αυτή ενεργοποιεί τους δύο κινητήρες ώστε να πετύχει την εξής στροφή στους τροχούς: Ο αριστερός τροχός να κινηθεί προς τα πίσω και ο δεξιός τροχός να κινηθεί προς τα εμπρός.

```
void moveL(int del)
{

  analogWrite(IN2,0);
  analogWrite(IN3,0);
  analogWrite(IN1,Volt);
  analogWrite(IN4,Volt);
  delay(del);

}
```

### 5.1.10 ΣΥΝΑΡΤΗΣΗ moveR()

Αντίστοιχη λειτουργία επιτελεί η συνάρτηση δεξιάς στροφής moveR().

```
void moveR(int del)
{

  analogWrite(IN1,0);
  analogWrite(IN4,0);
  analogWrite(IN2,Volt);
  analogWrite(IN3,Volt);
  delay(del);

}
```

### 5.1.11 ΣΥΝΑΡΤΗΣΗ moveF()

Για εμπρόσθια κίνηση καλούμε τη συνάρτηση moveF().

```
void moveF(int del)
{

  analogWrite(IN1,0);
  analogWrite(IN2,Volt);
  analogWrite(IN3,0);
  analogWrite(IN4,Volt);
  delay(del);

}
```

### 5.1.12 ΣΥΝΑΡΤΗΣΗ moveB()

Όταν πάλι επιθυμούμε οπίσθια κίνηση, καλούμε την συνάρτηση moveB().

```
void moveB(int del)
{

  analogWrite(IN1,Volt);
  analogWrite(IN2,0);
  analogWrite(IN3,Volt);
  analogWrite(IN4,0);
  delay(del);

}
```

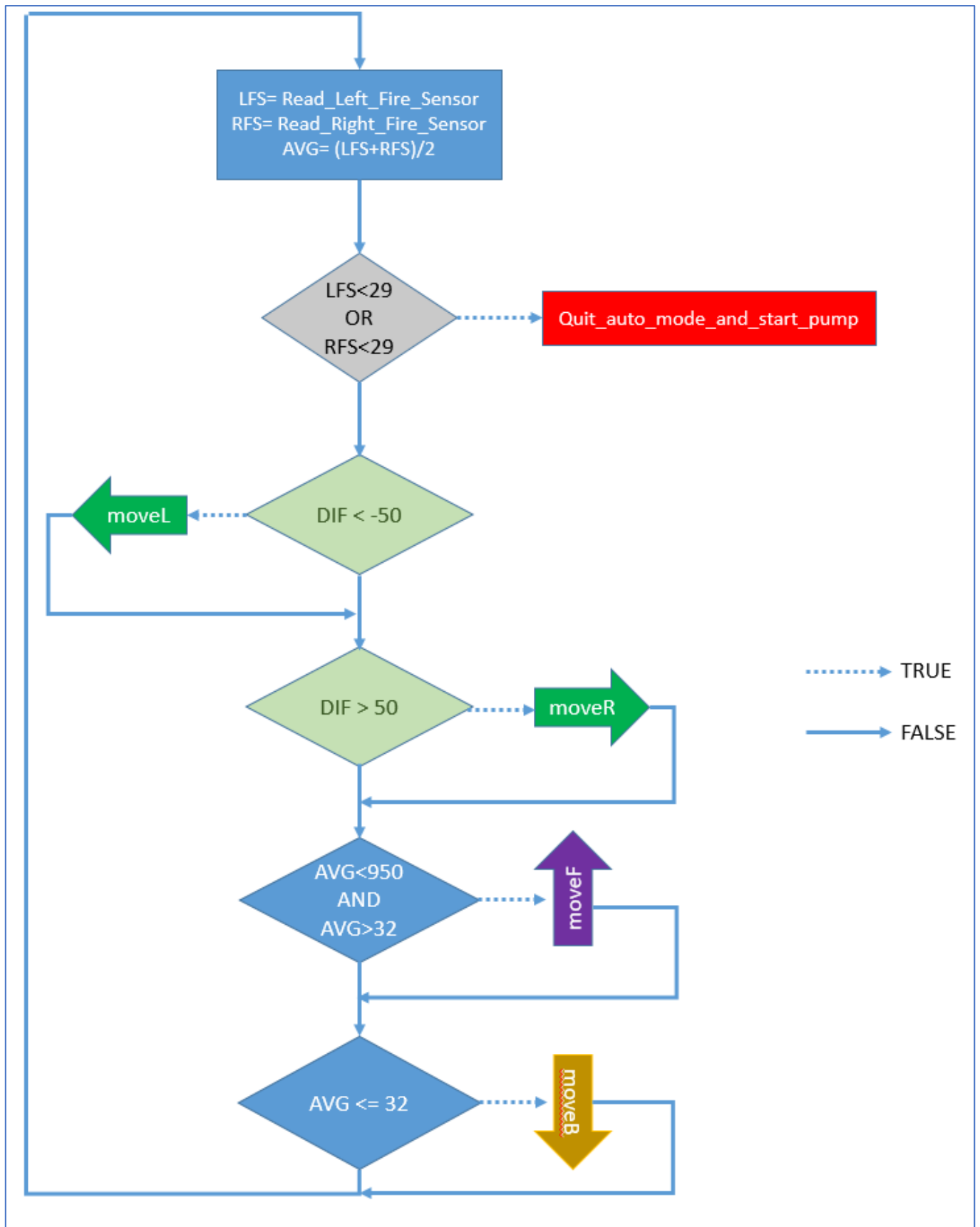
### 5.1.13 ΣΥΝΑΡΤΗΣΗ autoMove()

Η συνάρτηση η οποία αναλαμβάνει να πραγματοποιήσει την αυτόματη αναζήτηση της πύρινης εστίας από το όχημα είναι η autoMove(). Η λειτουργία του αλγορίθμου που εκτελεί φαίνεται στο σχήμα 5.2.

```
void autoMove()
{
  int dd=3;
  int dif=Left_Fire-Right_Fire;

  float ave=(Left_Fire+Right_Fire)*0.5;
  if(Left_Fire<29 || Right_Fire<29) quitAuto=1;
  if(dif<-50)
```

```
{
  moveL(dd);
  Stop();
}
else if(dif>50)
{
  moveR(dd);
  Stop();
}
else
  if(ave<950 && ave>32)
  { moveF(dd);
    Stop();
  }
  else if(ave<=32)
  {
    moveB(dd);
    Stop();
  }
}
```



Σχήμα 5.2. Διάγραμμα ροής συνάρτησης autoMove().

### 5.1.14 ΚΩΔΙΚΑΣ

```
#include <SoftwareSerial.h> // import the serial library
#include <TinyGPS.h>
#define IN1 5
#define IN2 6
#define IN3 7
#define IN4 8
#define SM 2
#define RF 5
#define LF 1
#define DELAY 1
#define ALed 9
#define MLed 10
#define MANUAL 1
#define AUTO 2
#define PUMP 11
SoftwareSerial gpsSerial(3,4); //rx,tx 3,4
TinyGPS gps;
int Speed=1,del=0,Volt=200,quitAuto=0;
int BluetoothData; // the data given from Computer
float lat = -1,lon = -1;
int year;
byte month, day, hour, minute, second, hundredths;
int pre_second=-10,tick=0;
int Left_Fire,Right_Fire,Smoke;
int Mode=MANUAL,change_mode=1;
int switchPump=1;
String latitude, longitude;
void setup()
{
  gpsSerial.begin(9600);
  Serial.begin(9600);
  Serial.println("System is UP!!!");
  setupPINS();
}

void gpsRead()
{
  while(gpsSerial.available())
  { // check for gps data
    if(gps.encode(gpsSerial.read())) // encode gps data
    {
      gps.crack_datetime(&year,&month,&day,&hour,&minute,&second,&hundredths);
      if(second!=pre_second) {tick++;pre_second=second;}
      if(tick>=DELAY)
      {
        gpsPosition();
        gpsTime();
        tick=0;
        SensorsPrint();
        Serial.println();
      }
    }
  }
}
```

```

void gpsPosition()
{
    gps.f_get_position(&lat,&lon); // get latitude and longitude
    Serial.print("Lat/Lon:");Serial.print(lat,4);Serial.print(",");
    Serial.println(lon,4);
}
void gpsTime()
{
    int CorrectTime=+3;
    Serial.print("Date:");
    if(day<10) Serial.print("0");
    Serial.print(day, DEC);
    Serial.print("/");
    if(month<10) Serial.print("0");
    Serial.print(month, DEC); Serial.print("/"); Serial.print(year);
    Serial.print(" Time:");
    if(hour+CorrectTime<10) Serial.print("0");
    Serial.print(hour+CorrectTime, DEC); Serial.print(":");
    if(minute<10) Serial.print("0");
    Serial.print(minute, DEC); Serial.print(":");
    if(second<10) Serial.print("0");
    Serial.println(second, DEC);
}

void bluetoothRead()
{
    if (Serial.available())
    {
        BluetoothData=Serial.read();
        if (BluetoothData=='A' || BluetoothData=='a') {speedUp();Mode=MANUAL;}
        if (BluetoothData=='D' || BluetoothData=='d') {speedDown();Mode=MANUAL;}
        if (BluetoothData=='F') {moveF(del);Mode=MANUAL;}
        if (BluetoothData=='B') {moveB(del);Mode=MANUAL;}
        if (BluetoothData=='L') {moveL(del);Mode=MANUAL;}
        if (BluetoothData=='R') {moveR(del);Mode=MANUAL;}
        if (BluetoothData=='X') {switchPump*=-1;}
        if (BluetoothData=='0') Stop();
        if (BluetoothData=='M')
        {
            quitAuto=0;
            if(change_mode==1)
            {
                change_mode=0;
                if(Mode==AUTO) Mode=MANUAL;
                else if(Mode==MANUAL) Mode=AUTO;
            }
        }
        else
            change_mode=1;
    }
}
//=====LOOP=====
void loop()
{
    SensorsRead();
    bluetoothRead();
    gpsRead();
    PumpControl();
}

```



```

A_M_Led();
if (Mode==AUTO && quitAuto==0) autoMove();

}

void speedUp()
{
  if (Speed<5) Speed++;
  setVolt();
  Serial.print("Current Speed is:");
  Serial.println(Speed);
}

void A_M_Led()
{
  if (Mode==AUTO)
  {
    digitalWrite(ALed,1);
    digitalWrite(MLed,0);
  }
  else
  {
    digitalWrite(ALed,0);
    digitalWrite(MLed,1);
  }
}

void speedDown()
{
  if (Speed>1) Speed--;
  setVolt();
  Serial.print("Current Speed is:");
  Serial.println(Speed);
}

void setupPINS()
{
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
  pinMode(IN3,OUTPUT);
  pinMode(IN4,OUTPUT);
  pinMode(ALed,OUTPUT);
  pinMode(MLed,OUTPUT);
  pinMode(PUMP,OUTPUT);
}

void setVolt()
{
  if (Speed==1) Volt=130;
  if (Speed==2) Volt=150;
  if (Speed==3) Volt=180;
  if (Speed==4) Volt=210;
  if (Speed==5) Volt=255;
}

```

```

void moveL(int del)
{
    analogWrite(IN2,0);
    analogWrite(IN3,0);
    analogWrite(IN1,Volt);
    analogWrite(IN4,Volt);
    delay(del);

}

void moveR(int del)
{
    analogWrite(IN1,0);
    analogWrite(IN4,0);
    analogWrite(IN2,Volt);
    analogWrite(IN3,Volt);
    delay(del);

}

void moveF(int del)
{
    analogWrite(IN1,0);
    analogWrite(IN2,Volt);
    analogWrite(IN3,0);
    analogWrite(IN4,Volt);
    delay(del);

}

void moveB(int del)
{
    analogWrite(IN1,Volt);
    analogWrite(IN2,0);
    analogWrite(IN3,Volt);
    analogWrite(IN4,0);
    delay(del);

}

void SensorsRead()
{
    Left_Fire=analogRead(LF);
    Right_Fire=analogRead(RF);
    Smoke=analogRead(SM);

}

void SensorsPrint()
{

```

```

Serial.print("Left/Right/Smoke:");
Serial.print(Left_Fire); Serial.print("/");
Serial.print(Right_Fire); Serial.print("/");
// Serial.print(Left_Fire-Right_Fire); Serial.print("/");
//Serial.print(quitAuto);Serial.print("/");
//Serial.println((Left_Fire+Right_Fire)/2);
Serial.println(Smoke);

}

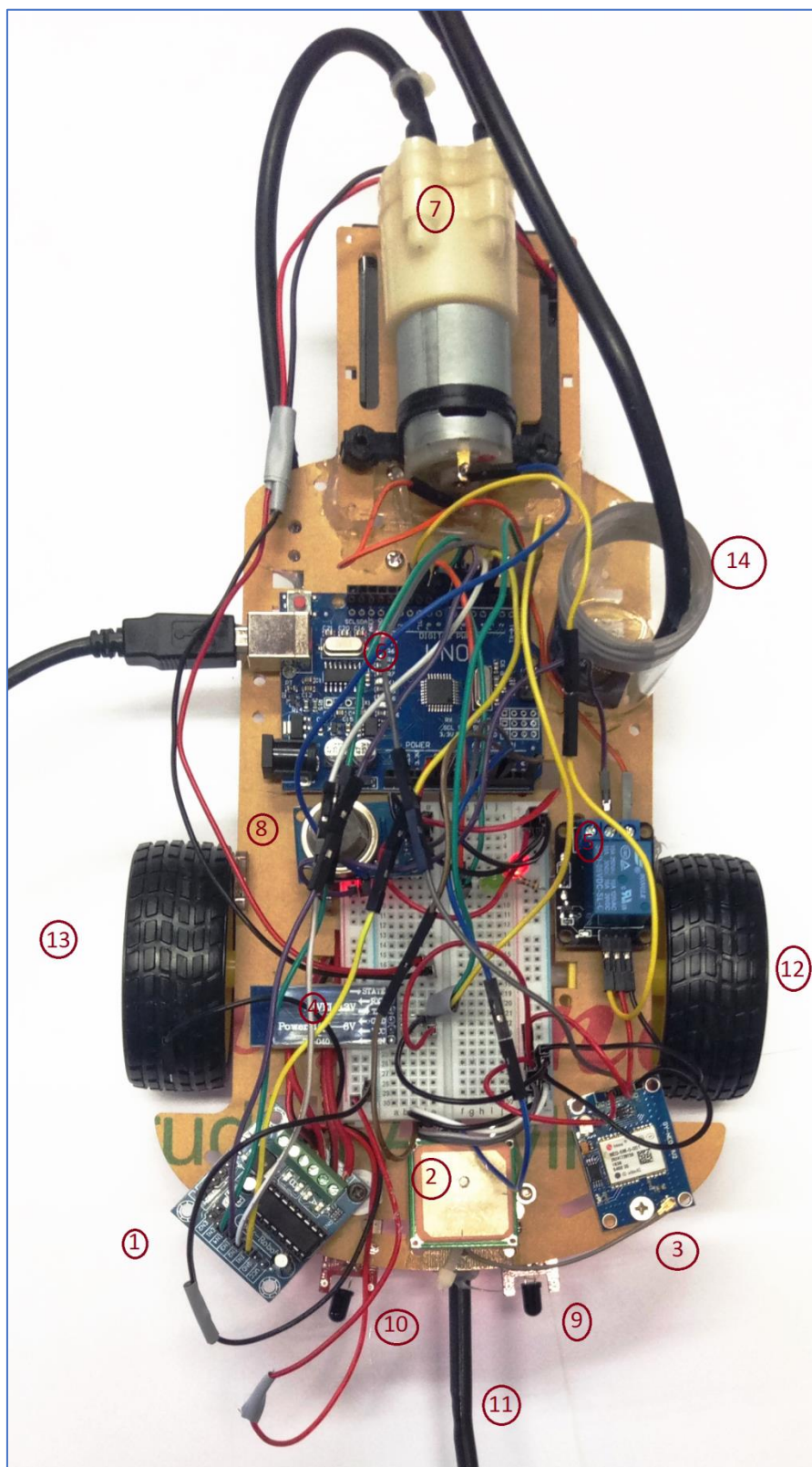
void PumpControl()
{
  if(switchPump==1)
    digitalWrite(PUMP,1);
  else
    digitalWrite(PUMP,0);
}

void Stop()
{
  digitalWrite(IN1,0); digitalWrite(IN2,0); digitalWrite(IN3,0); digitalWrite(IN4,0);
}
//+++++ auto functions |+++++
void autoMove()
{
  int dd=3;
  int dif=Left_Fire-Right_Fire;

  float ave=(Left_Fire+Right_Fire)*0.5;
  if(Left_Fire<29 || Right_Fire<29) quitAuto=1;
  if(dif<-50)
  {
    moveL(dd);
    Stop();
  }
  else if(dif>50)
  {
    moveR(dd);
    Stop();
  }
  else
  if(ave<950 && ave>32)
  { moveF(dd);
    Stop();
  }
  else if(ave<=32)
  {
    moveB(dd);
    Stop();
  }
} //autoMove

```

## 5.2 ΠΕΡΙΓΡΑΦΗ ΔΙΑΤΑΞΗΣ



Εικόνα 5.1-Άνω όψη του οχήματός μας

- 1: Motor drive
- 2: GPS antenna
- 3: GPS Module Neo-6m
- 4: Bluetooth
- 5: Relay
- 6: Μικροελεγκτής Arduino Uno
- 7: Pump motor
- 8: Αισθητήρας καπνού
- 9: Αριστερός αισθητήρας φωτιάς
- 10: Δεξιός αισθητήρας φωτιάς
- 11: Αντλία εκτόξευσης νερού
- 12: Αριστερός τροχός οχήματος
- 13: Δεξιός τροχός οχήματος
- 14: Δοχείο νερού

### 5.3 ΣΥΜΠΕΡΑΣΜΑΤΑ

Με την περάτωση αυτής της πτυχιακής εργασίας, καταλήξαμε σε μερικά σημαντικά συμπεράσματα όσον αφορά το όχημά μας όσον αφορά τις δυνατότητές του, καθώς και τους περιορισμούς που προκύπτουν από την κατασκευή του.

Λόγω της ευαισθησίας του GPS, είναι δύσκολο να πιάσει άμεσα καλό σήμα για τις συντεταγμένες του εντός κτηριακών δομών, καθώς θα πρέπει να είναι κάτω από καθαρό ουρανό για άμεση και αποτελεσματική σύνδεση με τους εν ενεργεία δορυφόρους του συστήματος GPS.

Όσον αφορά το Bluetooth, το όχημα μπορεί να ανιχνεύσει την πύρινη εστία από μόνο του σε απόσταση ελάχιστα μεγαλύτερη του ενός μέτρου. Επίσης, για καλύτερο αποτέλεσμα, βοηθάει αν το πείραμα διεξάγεται σε δωμάτιο χωρίς άμεσο ηλιακό φως καθώς αυτό περιέχει ικανή ποσότητα υπέρυθρης ακτινοβολίας ώστε να δημιουργεί εσφαλμένες μετρήσεις στους αισθητήρες φωτιάς.

Εκτός από το γεγονός ότι χάρη στο πυροσβεστικό μας όχημα δεν θέτουμε σε κίνδυνο ανθρώπινες ζωές, χάρη στην τεχνολογία του, το όχημά μας μας επιστρέφει δεδομένα που υπό άλλες συνθήκες δεν θα ήταν δυνατό να συλλέξουμε ή να εκτιμήσουμε αν ήταν ένας άνθρωπος στο πεδίο δράσης και ερευνών, ενώ σε περίπτωση βλάβης μπορούμε απλά να αντικαταστήσουμε το ελαττωματικό εξάρτημα με ένα καινούργιο και καλύτερο, κάτι που δε μπορούμε να πούμε ότι γίνεται να κάνουμε στην περίπτωση του ανθρώπου.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Ι. Ν. Αβαριτσιώτης, Τεχνολογία αισθητήρων και μικροσυστημάτων, 2003
- [2] Noble, J., Programming interactivity: A Designer's Guide to Processing, Arduino and open Framework, O' Reilly Media, 2009.
- [3] [www.arduino.cc/reference/en/](http://www.arduino.cc/reference/en/)
- [4] [www.arduino.cc/en/Main/ArduinoBoardUno](http://www.arduino.cc/en/Main/ArduinoBoardUno)
- [6] [www.u-blox.com/en/product/neo-6-series](http://www.u-blox.com/en/product/neo-6-series)
- [7] "GPS Guide for Beginners",  
[www.garmin.com/manuals/GPSGuideforBeginners Manual.pdf](http://www.garmin.com/manuals/GPSGuideforBeginners Manual.pdf) , Garmin (2000).
- [8] [www.instructables.com/id/AT-command-mode-of-HC-05-Bluetooth-module/](http://www.instructables.com/id/AT-command-mode-of-HC-05-Bluetooth-module/)
- [9] James Kardach Principle Engineer, "Bluetooth Architecture Overview" Intel Corporation  
Copyright 1998
- [10] Bluetooth SIG, Specification of the Bluetooth System, Feb. 2001

[11] Andreas Goransson, David Cuartielles Ruiz, Professional Android open accessory programming with arduino, Wiley publishing, 2013

[12] [www.arduino.cc/en/Tutorial/DueMotorShieldDC](http://www.arduino.cc/en/Tutorial/DueMotorShieldDC)

[13] "Interfacing Flame Sensor with Arduino to Build a Fire Alarm System", [www.circuitdigest.com/microcontroller-projects/](http://www.circuitdigest.com/microcontroller-projects/)

[14] Bukwoski, R. W. (2010). Techniques For Fire Detection

[15] Electronics Tutorials. (2014). Temperature Sensors. Retrieved October 17, 2016, from [www.electronics-tutorials.ws/io/io\\_3.html](http://www.electronics-tutorials.ws/io/io_3.html)

[16] System Sensor. (2016). System Smoke Detectors Application Guide. Retrieved November 2016, From SystemSensor:  
[www.systemsensor.com/enus/documents/system\\_smoke\\_detectors\\_appguide\\_spag91.pdf](http://www.systemsensor.com/enus/documents/system_smoke_detectors_appguide_spag91.pdf)

[17] [www.instructables.com/id/Arduino-Modules-L298N-Dual-H-Bridge-Motor-Controll/](http://www.instructables.com/id/Arduino-Modules-L298N-Dual-H-Bridge-Motor-Controll/)

[18] [www.societyofrobots.com/](http://www.societyofrobots.com/)