ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε

Πτυχιακή Εργασία: "Ανάπτυξη και υλοποίηση αυτόνομου ρομποτικού οχήματος ελεγχόμενο από μικροελεγκτή"

Φοιτητές: Χιονίδης Παύλος Α.Μ: 4483
Τζαβολάκης Εμμανουήλ Α.Μ:4375

ΧΑΝΙΑ 2016

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ

Ένα από τα βασικότερα προβλήματα στον χώρο των αυτόματων ρομποτικών συστημάτων, είναι αυτό της πλοήγησης. Περιλαμβάνει την χαρτογράφηση του περιβάλλοντα χώρου, τον εντοπισμό της τρέχουσας θέσης και τέλος τον έλεγχο της κίνησης για την ασφαλή μετακίνηση μέσα στον χώρο. Τα τελευταία χρόνια έχει αναπτυχθεί σημαντική ερευνητική δραστηριότητα γύρω από το πρόβλημα της ρομποτικής πλοήγησης και έχει οδηγήσει σε μια πληθώρα εναλλακτικών προσεγγίσεων για διάφορες περιπτώσεις.

Η παρούσα πτυχιακή εργασία έχει ως αντικείμενο την κατασκευή και την πλοήγηση ενός τετράτροχου ρομποτικού συστήματος με σκοπό την κίνηση και συλλογή δεδομένων μέσα σε ένα άγνωστο περιβάλλον. Επίσης το ρομποτικό σύστημα έχει την δυνατότητα επικοινωνίας με κάποιον χρήστη μέσω ηλεκτρονικού υπολογιστή η άλλου συστήματος, με στόχο την επεξεργασία των δεδομένων. Μέσα από την συγκεκριμένη εργασία αναδεικνύεται η χρησιμότητα που μπορούν να έχουν οι εφαρμογές της ρομποτικής σε επίπεδο έρευνας αλλά και στην καθημερινότητα. Η εφαρμογή της κατασκευής μπορεί να αποτελέσει ένα αντιπροσωπευτικό παράδειγμα, αφού με τον κατάλληλο εξοπλισμό θα μπορούσε να μετατραπεί σε ρομποτικό σύστημα συλλογής διαφορετικού τύπου δεδομένων/μετρήσεων μέσα σε ένα περιβάλλον.

Ακόμα, η συγκεκριμένη εργασία συνδυάζει την εφαρμογή γνώσεων από ένα φάσμα επιστημονικών τομέων. Εκτός από τους τομείς της Ηλεκτρονικής και της Ρομποτικής εφαρμόζονται γνώσεις από το πεδίο των Μικροϋπολογιστών, των Συστημάτων Αυτόματου Ελέγχου, καθώς επίσης και της Μηχανικής. Τέλος, η εργασία έχει εκπαιδευτικό χαρακτήρα και σκοπό. Εκτός την κατανόηση των βασικών εννοιών που σχετίζονται με την ρομποτική, και τα συστήματα αυτόματου ελέγχου καθώς και τον προγραμματισμό μικροελεγκτών και ενσωματωμένων συστημάτων, οι στόχοι που καλύπτει η εργασία είναι:

- Η κατανόηση του τρόπου λειτουργίας ενός συστήματος αυτόματου ελέγχου.
- Η ανάπτυξη ενός ενσωματωμένου συστήματος ικανού να ελέγξει όχημα αυτόματης πλοήγησης και συλλογής δεδομένων.
- Η επαφή με τεχνολογίες και διαδικασίες ανάπτυξης και κατασκευής ενός ρομποτικού συστήματος.

ΚΕΦΑΛΑΙΟ 1^{Ο}

Ρομποτική και Ρομποτικά συστήματα

1.1 Εισαγωγή

Η ρομποτική είναι η επιστήμη, το φάσμα της οποίας συνδυάζει τις επιστήμες της Ηλεκτρολογίας/Ηλεκτροτεχνίας, την Φυσική, την Μηχανολογία, την Πληροφορική, τις Τηλεπικοινωνίες. Ακόμα εμπεριέχει την θεωρεία συστημάτων αυτομάτου ελέγχου, την τεχνολογία των αισθητήριων διατάξεων, την επεξεργασία σήματος, την υπολογιστική όραση και τέλος την τεχνητή ζωή και νοημοσύνη. Είναι ο κλάδος του μηχανικού που ασχολείται με την σύλληψη, τον σχεδιασμό, την κατασκευή και την λειτουργία ρομποτικών συστημάτων. Η χρήση των ρομποτικών συστημάτων αποσκοπεί στην αντικατάσταση του ανθρώπου στην εκτέλεση έργου. Η αντικατάσταση αυτή αφορά τόσο στο φυσικό επίπεδο του έργου, όσο και στο επίπεδο λήψης απόφασης.

Σύμφωνα με το **Robot Institute of America**, ως ρομπότ μπορούμε να ορίσουμε έναν μηχανισμό σχεδιασμένο ώστε, μέσω προγραμματιζόμενων κινήσεων να μεταφέρει υλικά, τεμάχια, εργαλεία η ειδικευμένες συσκευές με σκοπό την επιτέλεση ποικιλίας εργασιών. Ένας τέτοιος μηχανισμός περιλαμβάνει συνήθως τις ακόλουθες συνιστώσες:

- Ένα μηχανολογικό υποσύστημα, το οποίο ενσωματώνει τη δυνατότητα του ρομπότ για εκτέλεση έργου. Το υποσύστημα αυτό αποτελείται από μηχανισμούς που επιτρέπουν στο ρομπότ να κινείται όπως αρθρώσεις, συστήματα μετάδοσης κίνησης, επενεργητές /κινητήρες, οδηγούς κλπ.

- Ένα υποσύστημα αίσθησης, μέσω του οποίου το ρομπότ συγκεντρώνει πληροφορίες για την κατάσταση στην οποία βρίσκονται τόσο το ίδιο όσο και το περιβάλλον. Το υποσύστημα αυτό εκτός των άλλων είναι υπεύθυνο για την αποδοχή των εξωτερικών εντολών, την επεξεργασία τους, τη μετάφραση τους σε ηλεκτρική ισχύ που θα δοθεί στους κινητήρες του ρομπότ. Καθώς επίσης και για την παραγωγή σημάτων εξόδου που θα πληροφορούν για την κατάσταση του συστήματος. Στο υποσύστημα αίσθησης περιλαμβάνονται όργανα μετρήσεως, αισθητήρες, ηλεκτρονικά στοιχεία κλπ.

- Ένα σύστημα ελέγχου, το οποίο συνδυάζει κατάλληλα την αίσθηση με τη δράση, έτσι ώστε το ρομπότ να λειτουργεί αποτελεσματικά και με τον επιθυμητό τρόπο. Ο ελεγκτής του ρομπότ επιβλέπεις και συντονίζει ολόκληρο το σύστημα, για την σχεδίαση και υλοποίηση του δε απαιτείται ο συνδυασμός γνώσεων από πολλές γνωστικές περιοχές, όπως είναι ο αυτόματος έλεγχος, η τεχνητή νοημοσύνη, η επιστήμη των υπολογιστών κλπ.

1.2 Ρομποτικά συστήματα στην αρχαιότητα

Αναζητώντας κάνεις τις ρίζες της ρομποτικής, θα οδηγηθεί αρκετά πίσω στην ιστορία της ανθρωπότητας. Πράγματι, η φιλοδοξία του ανθρώπου να δημιουργήσει μηχανές που θα μοιάζουν τόσο στη μορφή όσο και τη λειτουργιά πρωτοσυναντάται στην ελληνική μυθολογία Σύμφωνα με την τελευταία ο τιτάνας Προμηθέας έπλασε την ανθρωπότητα από πηλό. Επιπλέον ο Τάλος, ο μυθικός χάλκινος γίγαντας που κατασκεύασε ο Ήφαιστος για να προστατεύει την Κρήτη από τους εισβολείς, αποτελεί το πρώτο ¨αυτόματο¨ στην ανθρώπινη ιστορία. Ακόμα δημιουργίες του ¨Ηφαίστου ήταν και οι Χρυσοί Βοηθοί, τους οποίους κατασκεύασε για να τον βοηθούν στο εργαστήριο, να τον στηρίζουν για να περπατάει καλύτερα και να επικοινωνεί μαζί τους.

Επίσης ο Όμηρος και ο Πλάτωνας αναφέρουν ότι ο Δαίδαλος (ανάμεσα στις άλλες εντυπωσιακές του κατασκευές) έφτιαξε και κούκλες, για τα παιδία του Μίνωα, που μπορούσαν να μιλάνε και να κινούνται. Λέγεται μάλιστα ότι αναγκάζονταν να τις δένουν για να μην τους φεύγουν μακριά και τις χάνουν. Το ίδιο λέγεται και για τους μηχανικούς ανθρωπόμορφους φύλακες του λαβύρινθου που κινούταν με υδράργυρο.

Ο Έλληνας φιλόσοφος Αριστοτέλης είπε, πως εάν κάθε εργαλείο, όταν του δίνονται οδηγίες (η από δικιά του πρωτοβουλία) μπορεί να εργαστεί μόνο του, τότε δεν θα χρειαζόμασταν σκλάβους η εργάτες. Γενικότερα, οι αρχαίοι είχαν δώσει μεγάλη σημασία στις αυτόματες και αυτόνομες μηχανές. Ένας από τους σημαντικότερους αρχαίους επιστήμονες ήταν ο Ήρωνας ο Αλεξανδρεύς. Ο Ήρωνας ήταν μηχανικός και γεωμέτρης. Έζησε στην Αλεξάνδρεια περίπου το 1^{ο} αιώνα π.Χ. η τον 1^{ο} μ.Χ. Η πιο διάσημη εφεύρεση του είναι η αιολόσφαιρα η ατμοστρόβιλος, η οποία είναι η πρώτη ατμομηχανή στην ιστορία.

Οι κατασκευές και οι αυτοματισμοί του τον κάνουν να βρίσκεται ανάμεσα στις μεγαλύτερες και πιο σημαντικές μορφές της επιστήμης στην αρχαιότητα. Το τεράστιο έργο του αποτελείται από δεκαέξι πραγματείες από τις οποίες έχουν διασωθεί οι δέκα. Οι αυτοματισμοί του Ήρωνα είχαν εφαρμοστεί σε θέατρα, βιομηχανία, θρησκευτικές τελετές κ.α.

Στην σύγχρονη εποχή, η εισαγωγή της έννοιας των ρομπότ έγινε το 1921 από τον Τσέχο θεατρικό συγγραφέα Karel Capek με το θεατρικό έργο “Rossum’s Universal Robots”. Στο τελευταίο ο συγγραφέας φαντάζεται ένα μηχανικό κατασκεύασμα, το οποίο και ονομάζει robot από την ρωσική λέξη robota για την καταναγκαστική εργασία.

1.3 Είδη ρομποτικών συστημάτων

Κατά την πολυετή εξέλιξη της επιστήμης της ρομποτικής προέκυψαν διάφορα είδη ρομποτικών μηχανισμών, οι οποίοι διαφέρουν σημαντικά στη μορφή, αποτελούνται όμως από αντίστοιχα επιμέρους υποσυστήματα, τα οποία έχουν αναφερθεί στο κεφάλαιο 1.2.

Τα σπουδαιότερα είδη ρομποτικών συστημάτων είναι τα παρακάτω:

- **Ρομποτικά συστήματα σταθερής βάσης:** Τα ρομποτικά αυτά συστήματα, αποτελούνται από διαδοχικά στερεά σώματα (σύνδεσμοι) που συνδέονται μέσω αρθρώσεων σχηματίζοντας μια κινηματική αλυσίδα. Η αλυσίδα αυτή

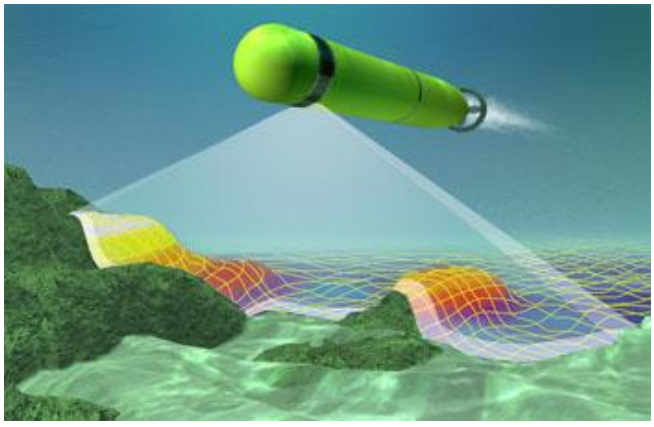
έχει το ένα άκρο της (βάση) σταθερά συνδεδεμένο με κάποιο σημείο του περιβάλλοντος χώρου. Η μορφή αυτή, είναι η παραδοσιακή μορφή ενός βιομηχανικού ρομποτικού βραχίονα και περιλαμβάνει τον βραχίονα, τον καρπό και το εργαλείο. Παράδειγμα του συγκεκριμένου ρομποτικού συστήματος απεικονίζεται στην εικόνα 1.1.

Εικόνα 1.1: Ρομποτικό σύστημα σταθερής βάσης [1].

- **Κινούμενα ρομποτικά συστήματα:** Ως κινητά ρομποτικά συστήματα χαρακτηρίζονται όλα εκείνα τα συστήματα που έχουν τη δυνατότητα να μετακινήσουν όλα τα σημεία του μηχανισμού τους. Η δυνατότητα αυτή προσφέρεται από ειδικά συστήματα προώθησης, τα οποία μπορεί να είναι τροχοί, είτε πολυπλοκότερα, όπως τουρμπίνες, προπέλες, μηχανικά πόδια κ.α. Το ρομποτικό σύστημα της πτυχιακής εργασίας ανήκει σε αυτήν την κατηγορία και συγκεκριμένα στα έντροχα ρομποτικά συστήματα. Τα συγκεκριμένα, λειτουργούν με αρκετά υψηλό βαθμό αυτονομίας. Πιο συγκεκριμένα μπορούν να λειτουργούν χωρίς συνεχή εξωτερική επίβλεψη και είναι ικανά να εκτελούν εργασίες αυτόνομα, δεχόμενα μόνο ορισμένες υψηλού επιπέδου εντολές.
- **Εναέρια ρομποτικά συστήματα:** Πρόκειται για μη επανδρωμένα ιπτάμενα ρομποτικά συστήματα, όπως ελικόπτερα και αεροπλάνα. Τα συγκεκριμένα συστήματα έχουν διαρκώς αυξανόμενες εφαρμογές όμως εξαιτίας της μειωμένης ακόμα σταθερότητας και ασφάλειας στην συμπεριφορά τους χρησιμοποιούνται κυρίως για στρατιωτικούς σκοπούς. Βέβαια με την εξέλιξη της τεχνολογίας και την εφεύρεση των τετρακοπτέρων αλλά και εξακοπτέρων, συστήματα τα οποία στόχευαν στην ασφάλεια την σταθερότητα και την ευκολία διαχείρισης (Quadra copter, hex copter) χρησιμοποιούνται όλο και περισσότερο στην καθημερινότητα για επαγγελματικούς η και προσωπικούς σκοπούς (φωτογραφεία, ερευνητικά, βιντεοσκοπήσεις κ.α.). Στην εικόνα 1.2 απεικονίζεται ένα είδος τετρακοπτέρου.

Εικόνα 1.2: τετρακόπτερο ρομποτικό σύστημα (Quadra Copter). *[2]*

Ρομποτικά συστήματα AUVs: Τα AUVs (Autonomous Underwater Vehicles) είναι πλήρως αυτόνομα συστήματα. Ανήκουν στην κατηγορία των μη επανδρωμένων υποβρύχιων ρομπότ. Για τις ανάγκες τροφοδοσίας χρησιμοποιούνται ειδικές μπαταρίες κάτι που όμως θέτει και περιορισμούς στη λειτουργία τους. Έχουν σχήμα τορπιλών και μπορούν να κινούνται με αρκετά μεγάλη ταχύτητα. Χρησιμοποιούνται σε υποβρύχιες κατασκευές αλλά και εξερευνήσεις.



Εικόνα 1.3:Παραδειγμα AUV ρομποτικού συστήματος *[3]*.

1.4 Ιστορική αναδρομή σε έντροχα ρομποτικά συστήματα

Στο κεφάλαιο αυτό, θα κάνουμε μια αναφορά στην ιστορία των κινούμενων ρομποτικών συστημάτων, η οποία έχει στιγματίσει την πορεία της τεχνολογίας και της επιστήμης από τα μέσα του 20^{ου} αιώνα μέχρι σήμερα. Η εξέλιξη των έντροχων ρομποτικών συστημάτων, προέκυψε με σκοπό την επιτέλεση διάφορων λειτουργιών, από την εκτέλεση απλών καθημερινών εργασιών, μέχρι πολύπλοκες έρευνες.

1939-1945: Εμφανίζονται τα πρώτα κινούμενα ρομποτικά συστήματα κατά την διάρκεια του Β' Παγκόσμιου Πολέμου.

1948-1949: Ο William Grey Walter κατασκευάζει τα ρομπότ Elmer και Elsie, δυο αυτόνομα ρομποτικά συστήματα τα οποία εξερευνούσαν το περιβάλλον τους. Πιο συγκεκριμένα κάθε ένα από τα ρομποτικά συστήματα ήταν εξοπλισμένα με έναν

αισθητήρα φωτός. Έτσι εάν έβρισκαν κάποια πηγή φωτός κινούνταν προς αυτή, αποφεύγοντας παράλληλα τα εμπόδια που βρίσκονταν στο δρόμο τους. Αυτά τα ρομπότ αποδείκνυαν ότι μια πολύπλοκη συμπεριφορά μπορούσε να προκύψει από έναν απλό σχεδιασμό.

1961-1963: Το Johns Hopkins University ανέπτυξε το ρομπότ "Beast". Το ρομποτικό σύστημα Beast χρησιμοποιούσε ένα σόναρ για να μπορέσει να κινηθεί. Όταν οι μπαταρίες του αποφόρτιζαν τότε το ρομπότ θα εντόπιζε μια βάση φόρτισης, με την οποία και θα συνδεόταν.

1970: Το πανεπιστήμιο του Stanford, δημιουργεί ένα κινούμενο ρομπότ το οποίο ήταν ικανό να ακολουθήσει μια άσπρη γραμμή, χρησιμοποιώντας μια κάμερα. Το Cart line follower ήταν ασύρματα συνδεδεμένο με έναν μεγάλο υπολογιστή, ο οποίος πραγματοποιούσε υπολογισμούς.

Την ίδια περίπου περίοδο (1966-1972), το ερευνητικό κέντρο του Stanford κατασκευάζει και πραγματοποιεί έρευνα πάνω σε ένα ρομπότ που ονομάστηκε Shakey, εξαιτίας της σπασμωδικής του κίνησης. Το Shakey ήταν εξοπλισμένο με μια κάμερα, αισθητήρα μέτρησης απόστασης, αισθητήρες επαφής καθώς επίσης και κεραία για ασύρματη επικοινωνία. Ήταν το πρώτο ρομποτικό σύστημα που επιχειρηματολογούσε για τις πράξεις του. Αυτό σημαίνει ότι το ρομπότ δεχόταν πολύ γενικές εντολές και αντιλαμβανόταν τα απαραίτητα βήματα που έπρεπε να κάνει ώστε να φέρει εις πέρας την ζητούμενη εργασία.

Την ίδια χρονολογία αξίζει να αναφερθεί ότι η Σοβιετική Ένωση εξερευνάει την επιφάνεια της Σελήνης, με το Lunokhod 1, ένα σεληνιακό Rover.

1976: Με το πρόγραμμα Viking, η NASA στέλνει δυο μη επανδρωμένα διαστημικά σκάφη στον Άρη.

1980: Η ομάδα του Ernst Dickmanns κατασκευάζει στο Bundeswehr University του Μονάχου, τα πρώτα ρομπότ – αυτοκίνητα, τα οποία είχαν την δυνατότητα να κινηθούν ως και 55 μίλια την ώρα, σε άδειους δρόμους.

1987: Αξίζει να σημειωθεί ότι το Hughes Research Laboratories, κατασκευάζει την πρώτη χαρτογραφημένη και βασιζόμενη σε αισθητήρες, αυτόνομη λειτουργία ενός ρομποτικού οχήματος.

1989: Ο Mark Tilden επινοεί την ρομποτική εταιρεία BEAM (Biology, Electronics, Aesthetics, Mechanics)

1990: Ο Joseph Engelberger, πατέρας του βιομηχανικού ρομποτικού βραχίονα, εργάζεται με συναδέλφους προκειμένου να σχεδιάσει τα πρώτα εμπορικά διαθέσιμα, αυτόνομα κινούμενα ρομπότ για νοσοκομειακή χρήση. Τα συγκεκριμένα κυκλοφορήσαν στην αγορά από την HelpMate. Ακόμα την ίδια περίοδο το υπουργείο Άμυνας των ΗΠΑ επενδύει πάνω στο project MDARS-I, το οποίο βασίζεται στα ρομπότ ασφαλείας εσωτερικών χώρων.

1991: Ο Edo Franzi ο Adre Guignard και ο Francesco Mondada ανέπτυξαν το Khepera, ένα μικρό αυτόνομο κινούμενο ρομπότ, που έχει ως σκοπό ερευνητικές δραστηριότητες. Το συγκεκριμένο project υποστηρίχτηκε από το εργαστήριο του Πολυτεχνείου της Λωζάνης.

1993-1994: Τα Dante I και Dante II αναπτύχθηκαν από το Carnegie Mellon University. Και τα δυο ήταν βαδίζοντα ρομπότ που είχαν ως στόχο την εξερεύνηση ενεργών ηφαιστείων. Την ίδια περίοδο, τα δυο δίδυμα ρομπότ – οχήματα VaMP της Daimler-

Benz και VITA-2 της UNIBwM, ταξίδεψαν με επιβαίνοντες περισσότερα από 1000km/h σε αυτοκινητόδρομο του Παρισιού 3 λωρίδων και με φυσιολογικά υψηλή κίνηση, με ταχύτητες κοντά στα 130 km/h. Τα ρομπότ αυτά επιδείκνυαν αυτόνομη οδήγηση σε ελεύθερες λωρίδες, οδήγηση σε φάλαγγα αυτοκινήτων , καθώς και αλλαγές λωρίδας (αριστερά και δεξιά), με αυτόνομη προσπέραση άλλων αυτοκινήτων.

1995: Ένα από τα ρομπότ – αυτοκίνητα του Ernst Dickmanns ταξίδεψε περισσότερα από 1000 μίλια, από το Μόναχο στην Κοπεγχάγη και επέστρεψε, εν μέσω κίνησης, με ταχύτητες κοντά στα 120 μίλια την ώρα, εκτελώντας περιστασιακά ελιγμούς για να προσπεράσει άλλα αυτοκίνητα. Ενεργής όραση χρησιμοποιήθηκε για να αντιμετώπιση τις ξαφνικές αλλαγές σε ορισμένα σημεία του δρόμου.

1996-1997: Η NASA στέλνει στον Άρη το Mars Pathfinder μαζί με το rover του, το Sojourner, το οποίο είχε ως σκοπό την εξερεύνηση της επιφάνειας του Άρη, δεχόμενο εντολές από την Γη. Το Sojourner ήταν εξοπλισμένο με σύστημα αποφυγής κινδύνων, το οποίο επέτρεπε στο rover να βρίσκει αυτόνομα το δρόμο του μέσα στο άγνωστο έδαφος του Άρη.

1999: Η Sony παρουσιάζει τον Aibo, έναν ρομποτικό σκύλο με δυνατότητα να βλέπει, να βαδίζει και να αλληλοεπιδρά με το περιβάλλον του. Επίσης παρουσιάζεται το τηλεκατευθυνόμενο στρατιωτικό ρομπότ PackBot.

2001: Ξεκινάει το Swarm-bots project. Τα Swarm bots μοιάζουν με αποικίες εντόμων, και αποτελούνται από έναν μεγάλο αριθμό μικρών ανεξάρτητων ρομπότ, τα οποία αλληλοεπιδρούν μεταξύ τους και όλα μαζί εκτελούν πολύπλοκες εργασίες.

2002: Εμφανίζεται το Roomba , το πρώτο οικιακό αυτόνομο κινούμενο ρομπότ που καθαρίζει το πάτωμα.

2003: Η Axxon Robotics αγοράζει την Intellibot, κατασκευάστρια εταιρεία μιας σειράς εμπορικών ρομπότ τα οποία τρίβουν και σκουπίζουν τα πατώματα νοσοκομείων, γραφείων κτλ. Τα ρομπότ περιποίησης δαπέδου από την Intellibot Robotics, λειτουργούν εντελώς αυτόνομα, χαρτογραφώντας το περιβάλλον τους και χρησιμοποιώντας ένα σύνολο αισθητήρων για την αποφυγή εμποδίων κατά την πλοήγηση τους.

2005: Η Boston Dynamics δημιουργεί ένα τετράποδο ρομπότ, προκειμένου να κουβαλάει φορτία, σε έδαφος που είναι πολύ ανώμαλο για οχήματα.

2006: Η Sony σταματάει την κατασκευή του Aibo, και η HelpMate διακόπτει την παραγωγή της, αλλά το χαμηλότερου κόστους, αυτόνομο ρομπότ – βοηθός PatrolBot γίνεται διαθέσιμο, στην συνεχόμενη προσπάθεια να γίνουν τα κινούμενα ρομπότ εμπορικώς βιώσιμα.

2007: Γράφεται ιστορία με το DARPA Urban Grand Challenge, οπού έξι οχήματα ολοκληρώνουν αυτόνομα μια πολύπλοκη πορεία, η οποία περιλάμβανε επανδρωμένα οχήματα και εμπόδια. Την ίδια περίοδο το Seekur, το πρώτο ευρέως διαθέσιμο, μη στρατιωτικό, βοηθητικό ρομπότ εξωτερικών χώρων που έχει την δυνατότητα να ρυμουλκήσει ένα τριών τόνων όχημα από μια θέση παρκινγκ, να το οδηγήσει αυτόνομα σε έναν εσωτερικό χώρο και εν συνεχεία, να καθοδηγηθεί και πάλι προς τα έξω.

2008: Η Boston Dynamics κυκλοφορεί το απόσπασμα ενός βίντεο, για μιας νέας γενιάς BigDog ικανό να βαδίζει σε παγωμένο έδαφος και να ανακτά την ισορροπία του.

2010: Ο διαγωνισμός Multi Autonomous Ground Robotic International Challenge (MAGIC), παρουσιάζει ομάδες αυτόνομων οχημάτων που χαρτογραφούν ένα μεγάλο

αστικό περιβάλλον, που αναγνωρίζουν και παρακολουθούν ανθρώπους, και που αποφεύγουν εχθρικά αντικείμενα.

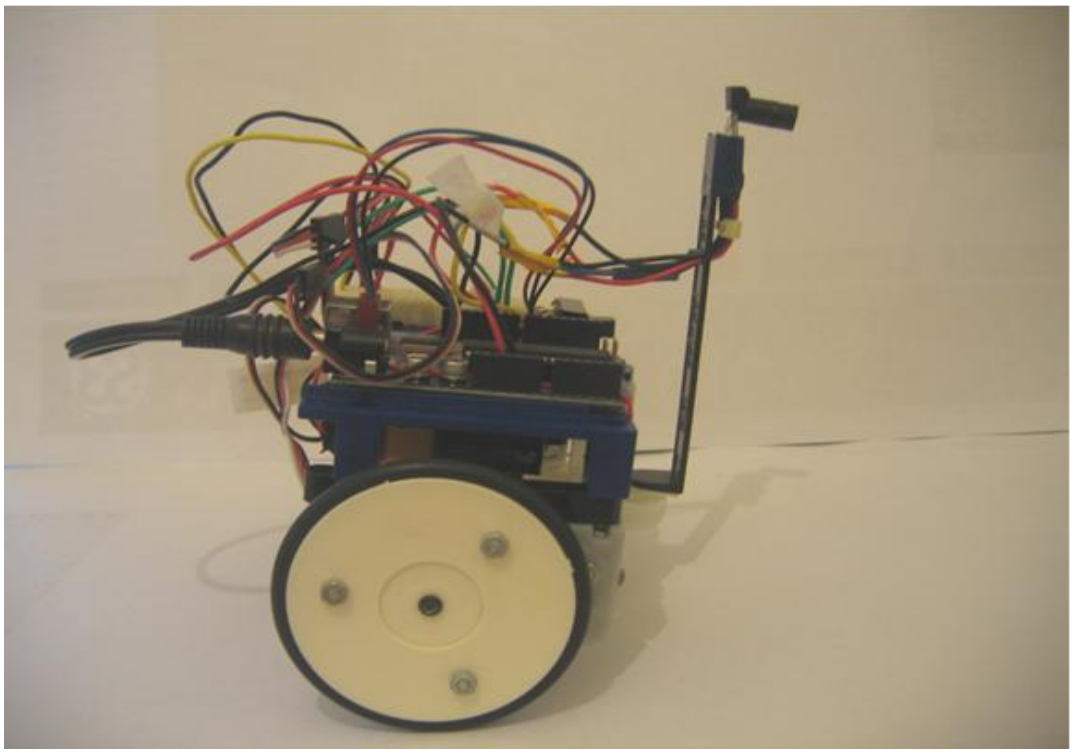
ΚΕΦΑΛΑΙΟ 2ο

Εφαρμογές έντροχων ρομποτικών συστημάτων

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα αναφέρουμε εφαρμογές έντροχων ρομποτικών συστημάτων, καθώς και θα γίνει μια περιληπτική ανάλυση για τον τρόπο κατασκευής και λειτουργίας τους. Όλα τα παραδείγματα, αποτελούν κατασκευές φοιτητών για εκπαιδευτικό σκοπό και έχουν αρκετές ομοιότητες με το ρομποτικό σύστημα της πτυχιακής εργασίας. Παρακάτω ακολουθούν 2 παραδείγματα έντροχων ρομποτικών συστημάτων ενώ στο τελευταίο μέρος του κεφαλαίου θα γίνει μια περιγραφή του τετράτροχου ρομποτικού συστήματος της πτυχιακής εργασίας.

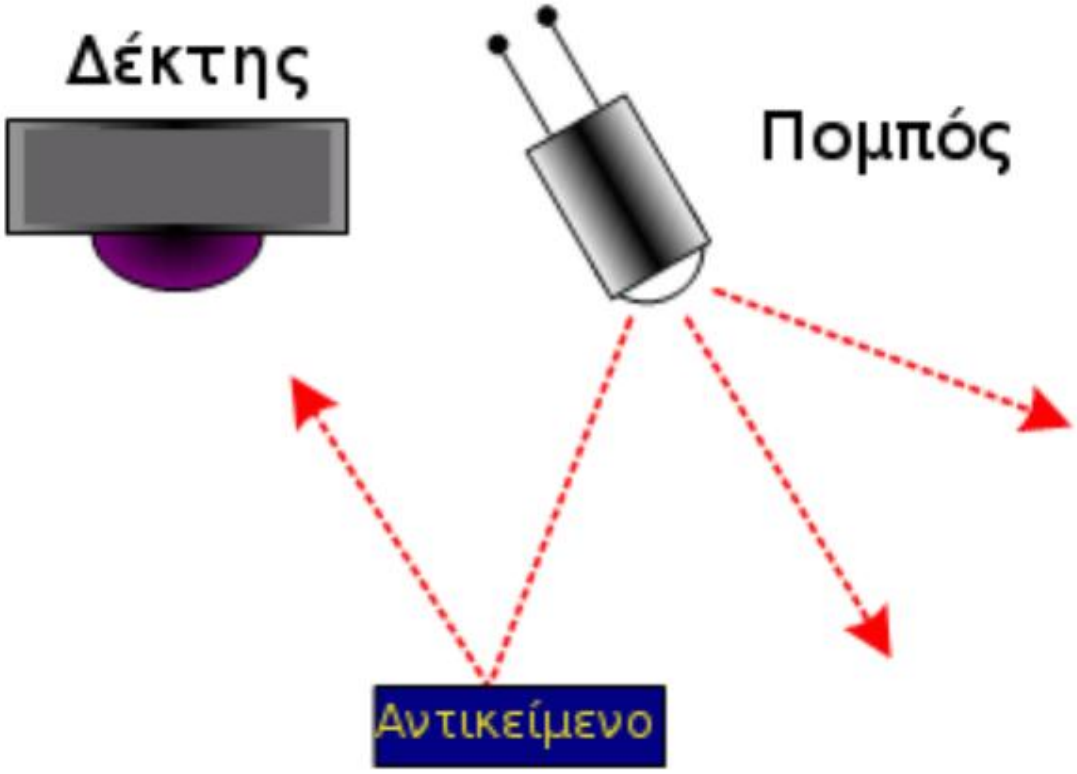
2.2 Εφαρμογή εντροχού ρομποτικού συστήματος ανίχνευσης αντικειμένων.



Εικόνα 2.1: Έντροχο ρομποτικό σύστημα ανίχνευσης αντικειμένων *[4]*.

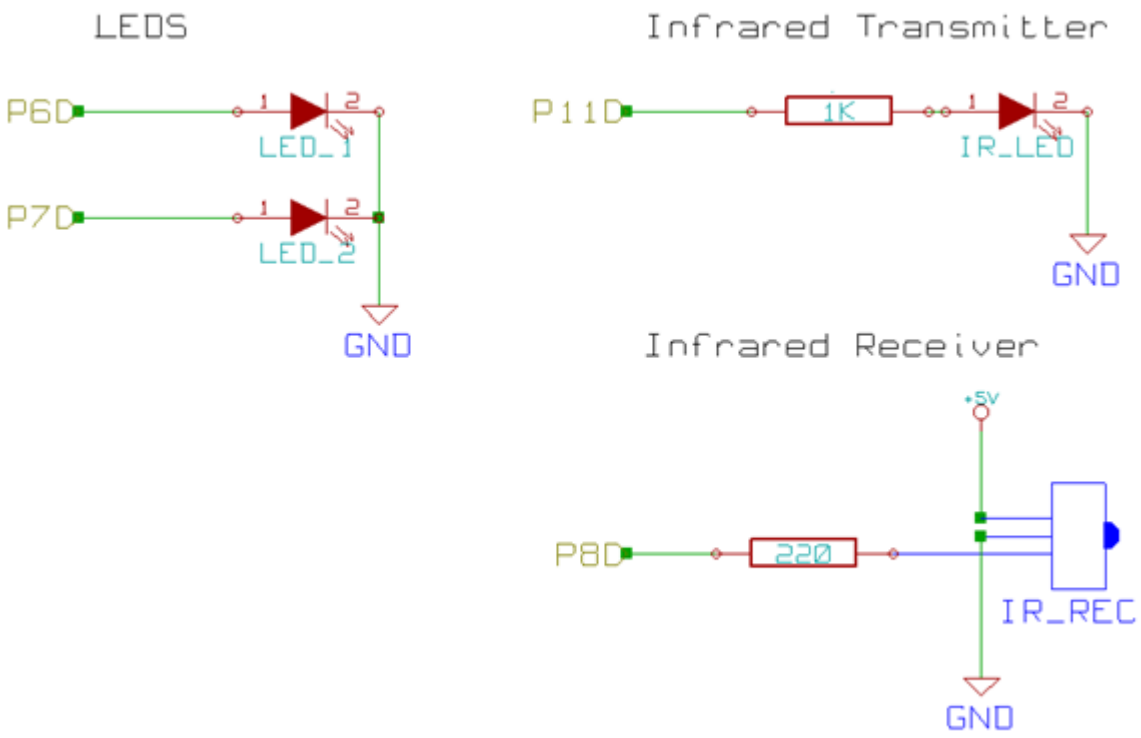
Το σύστημα που θα παρουσιάσουμε είναι ένα σύστημα που έχει την ικανότητα ανίχνευσης αντικειμένων ειδοποιώντας τον χρήστη σε κάθε επιτυχή ανίχνευση. Οι ειδοποιήσεις, έχουν την μορφή μιας LED ένδειξης συγκεκριμένου χρώματος. Στην παραπάνω εικόνα 2.1 απεικονίζεται το συγκεκριμένο ρομποτικό σύστημα. Τα δυο βασικά εξαρτήματα που αποτελούν το ρομποτικό σύστημα είναι ένας πομπός

υπέρυθρων (Parallax IR Transmitter Assembly Kit), καθώς και ένας δέκτης. Η βασική ιδέα είναι να στέλνει συνεχώς παλμούς ο πομπός, ώστε μόλις έχουμε ανάκλαση σε κάποιο αντικείμενο να γίνει αντιληπτό από τον δέκτη. Για να επιτευχθεί το παραπάνω θα πρέπει να συγχρονιστεί ο πομπός και ο δέκτης ώστε να στέλνει συνεχώς κύματα ο πρώτος καθώς και να είναι σε συνεχή αναμονή ώστε να δεχθεί το κύμα (υπέρυθρες) που ανακλάτε πίσω ο δέκτης. Στην εικόνα 2.2 φαίνεται η λειτουργία του πομπού με τον δέκτη, όταν υπάρχει αντικείμενο στον χώρο.



Εικόνα 2.2: Λειτουργία Πομπού – Δέκτη *[5]*.

Στην συγκεκριμένη εφαρμογή χρησιμοποιείτε πομπός ο οποίος μπορεί να στείλει τετραγωνικό παλμό συχνότητας 38,5KHz (38500Hz), καθώς και ένας δέκτης ο οποίος μπορεί να ανιχνεύσει αυτό το σήμα. Για την κατασκευή επιλέχτηκαν εξαρτήματα τα οποία είναι εύκολο να βρεθούν στο διαδίκτυο καθώς και σε οποιοδήποτε κατάστημα ηλεκτρονικών ειδών. Επίσης υπάρχουν και έτοιμες διατάξεις (modules) υπέρυθρων πομποδεκτών, ειδικά σχεδιασμένων για ρομποτικές εφαρμογές. Εγκέφαλος της εφαρμογής είναι το αναπτυξιακό **Arduino Diecimila** οπού συγχρονίζει όλες τις λειτουργίες του συστήματος. Για τις ενδείξεις σχετικά με την ανίχνευση του αντικειμένου χρησιμοποιούνται 2 LED διαφορετικού χώματος (κόκκινο, πράσινο). Ακόμα για την κυκλωματική διάταξη χρησιμοποιούνται 2 αντιστάσεις, μια των 220Ω και μια των 1ΚΩ, αντίστοιχα. Στην εικόνα 2.3 απεικονίζεται η συνδεσμολογία των αντιστάσεων με τον πομποδέκτη.

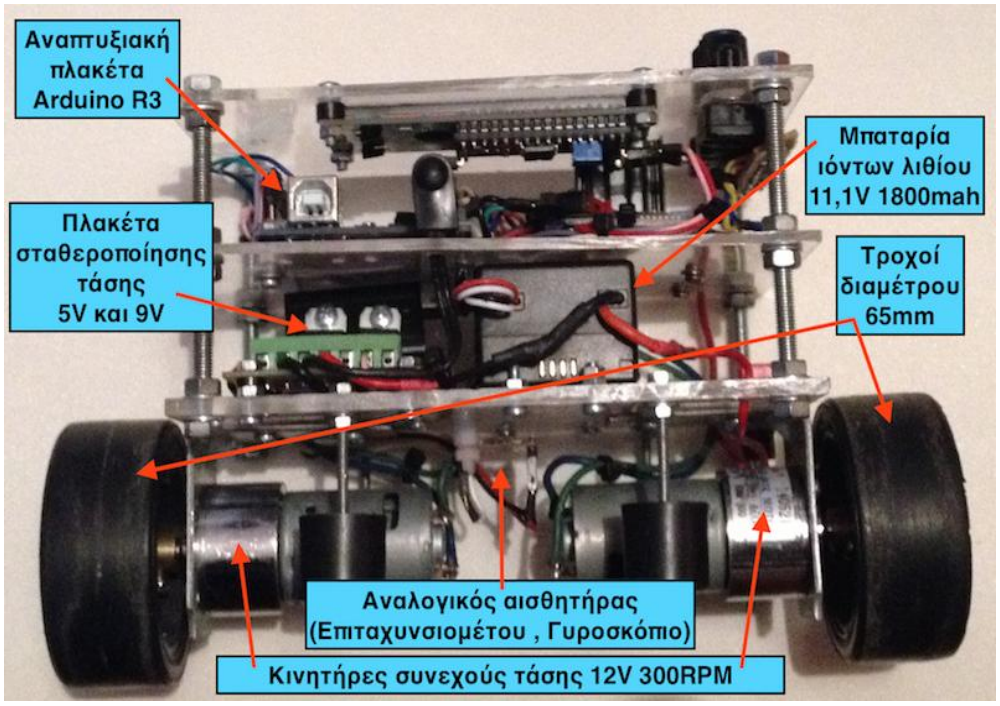


Εικόνα 2.3: Κύκλωμα σύνδεσης των LED και του πομποδέκτη υπέρυθρων *[6]*.

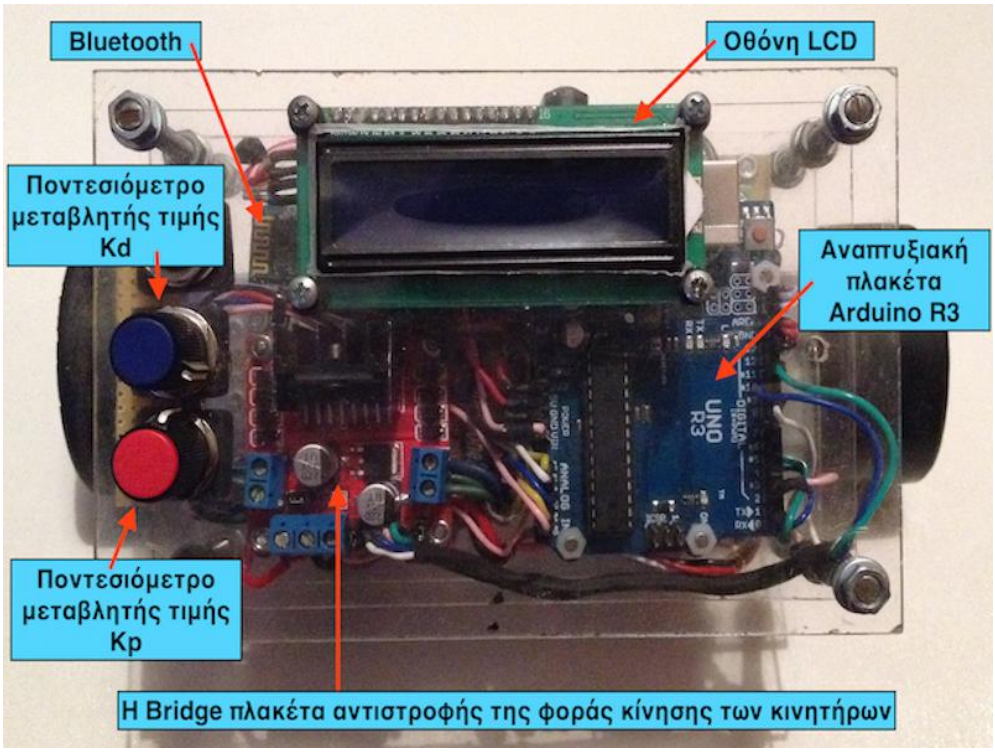
Όλα τα εξαρτήματα της εφαρμογής συνδέονται στο αναπτυξιακό Arduino Diecimila και ελέγχονται από τον μικροελεγκτή Atmega168. Ο προγραμματισμός του συστήματος γίνεται από την πλατφόρμα προγραμματισμού του Arduino. Στο κεφάλαιο 3 θα γίνουν αναλυτικές αναφορές στον μικροελεγκτή Atmega καθώς και στο αναπτυξιακό Arduino.

2.3 Εφαρμογή ρομποτικού συστήματος αυτό-ισορρόπησης

Στην συγκεκριμένη ενότητα θα γίνει η περιγραφή μιας πρωτότυπης ρομποτικής κατασκευής που έχει την ικανότητα να ισορροπεί σε δυο τροχούς ανεξάρτητα από τις διαταραχές που επιδρούν σε αυτή. Η συγκεκριμένη συσκευή αποτελεί ιδανική πλατφόρμα πειραματισμού και κατανόησης θεμάτων που σχετίζονται με την μηχατρονική, την ρομποτική και τον προγραμματισμό ηλεκτρονικών συσκευών. Το εν λόγο ρομποτικό σύστημα απεικονίζεται στις εικόνες 2.4, 2.5.



Εικόνα 2.4: Δίτροχο αυτόνομο ρομποτικό σύστημα αυτό –ισορρόπησης *[7]*.



Εικόνα 2.5: Δίτροχο αυτόνομο ρομποτικό σύστημα αυτό –ισορρόπησης. *[8]*

Το σύστημα αποτελείται από τρία επίπεδα, τα οποία είναι κατασκευασμένα από πλαστικά και οι διαστάσεις τους είναι ακριβώς ίδιες και στα τρία αυτά κομμάτια. Τα πλαστικά αυτά έχουν το κάθε ένα μήκος 155mm, πλάτος 100mm και ύψος 3mm. Χρησιμοποιήθηκε το πλαστικό σαν υλικό κατασκευής διότι είναι ελαφρύ και εύκολο στην επεξεργασία του. Για την ένωση των τριών αυτών πλαστικών επιπέδων, χρησιμοποιήθηκαν τέσσερις βίδες με διάμετρο 5mm καθώς και είκοσι τέσσερα

παξιμάδια και ανάλογες ροδέλες. Από την κάτω μεριά της πρώτης πλαστικής πλατφόρμας τοποθετήθηκαν οι βάσεις στήριξης των κινητήρων οι οποίες είναι κατασκευασμένες από μέταλλο και έχουν την μορφή Γ, όπως φαίνεται και στο σχήμα 2.4. Πάνω στις συγκεκριμένες βάσεις τοποθετήθηκαν οι κινητήρες και στην συνέχεια οι σύνδεσμοι που συνδέουν τον άξονα του κάθε κινητήρα με τον τροχό , καθώς και οι ίδιοι οι τροχοί. Στο μέσο της πρώτης πλατφόρμας τοποθετήθηκαν τα αισθητήρια για την καλύτερη μέτρηση καθώς και τέσσερις άξονες για την στήριξη του ρομποτικού συστήματος, όταν αυτό είναι απενεργοποιημένο. Ακόμα από την επάνω μεριά της ίδιας πλατφόρμας τοποθετήθηκε η πηγή ενέργειας, δηλαδή η μπαταρία, το κύκλωμα σταθεροποίησης τάσης και ο κεντρικός διακόπτης ενεργοποίησης και απενεργοποίησης του ρομποτικού συστήματος. Στο πάνω μέρος της δεύτερης πλατφόρμας τοποθετήθηκε ο οδηγός των κινητήρων η αλλιώς διάταξη H-Bridge (θα γίνει αναλυτική αναφορά στο κεφάλαιο 3). Δίπλα από την διάταξη H-Bridge έχει τοποθετηθεί και το αναπτυξιακό Arduino το οποίο είναι ο εγκέφαλος του συστήματος, καθώς ελέγχει όλες τις λειτουργίες, και τα επιμέρους τμήματα της κατασκευής. Η ασύρματη επικοινωνία Bluetooth έχει τοποθετηθεί δίπλα στο αναπτυξιακό. Τέλος, στην τελευταία πλατφόρμα τοποθετήθηκαν η LCD οθόνη και τα ποτενσιόμετρα που ελέγχουν της παραμέτρους του μικροελεγκτή μέσα στον κώδικα.

Για την ολοκλήρωση της κατασκευής χρειάστηκε αρχικά μια αναπτυξιακή πλακέτα Arduino, πάνω στην οποία είναι ενσωματωμένος ο μικροελεγκτής που προγραμματίζεται για να οδηγήσει όλα τα αισθητήρια αλλά και την διάταξη που οδηγεί τους τροχούς του συστήματος. Συγκεκριμένα χρησιμοποιείται ο μικροελεγκτής Atmega328P-PU, ο οποίος διαθέτει έναν 8 Bit μικροεπεξεργαστή. Το περιβάλλον προγραμματισμού, οπού γράφονται οι αλγόριθμοι και τα προγράμματα ανοίκει στην ίδια εταιρεία Arduino. Για να πέτυχουν την ισορρόπηση του ρομποτικού δίτροχου ήταν απαραίτητη η χρήση αισθητήριων τα οποία θα μπορούν να μετρούν την κλίση και την επιτάχυνση, έτσι ώστε να μπορεί το σύστημα να χρησιμοποιήσει τα δεδομένα και σε συνδυασμό με κατάλληλες στρατηγικές ελέγχου να οδηγεί τους κινητήρες ώστε τελικά να επιτυγχάνεται ισορροπία. Το αισθητήριο που χρησιμοποιήθηκε λοιπόν, ήταν ένα ολοκληρωμένο σύστημα πέντε βαθμών ελευθερίας. Αποτελούμενο από ένα επιταχυνσιόμετρο (ADXL-335) τριών βαθμών ελευθερίας, καθώς και από ένα γυροσκόπιο, δυο βαθμών ελευθερίας (IDG-650). Με την βοήθεια αυτών των αισθητήριών επιτυγχάνεται η παρακολούθηση της κλήσης και της επιτάχυνσης της πλατφόρμας στους τρεις άξονες X,Y και Z. Προκειμένου να οδηγηθούν οι κινητήρες, χρησιμοποιήθηκε μια διάταξη οδήγησης φορτίων H-Bridge, η συγκεκριμένη μπορεί να οδηγήσει 2 μοτέρ ταυτόχρονα.

Το σύστημα διαθέτει την δυνατότητα ασύρματης επικοινωνίας με απομακρυσμένο (H/Y), για διάφορους σκοπούς. Για να επιτευχθεί αυτό χρησιμοποιείται μια Bluetooth διάταξη και πιο συγκεκριμένα το μοντέλο HC-06. Ακόμα για να τροφοδοτηθεί το σύστημα χρησιμοποιείτε μπαταριά τεχνολογίας LiPo (Lithium-ion polymer battery, ιόντων λιθίου). Επειδή όλα τα ηλεκτρονικά μέρη του συστήματος δεν τροφοδοτούνται με την ίδια τάση και η τάση της μπαταρίας κατά την διάρκεια λειτουργιάς της, από πλήρης φορτισμένη μέχρι να αδειάσει δεν είναι σταθερή, κατασκευάστηκε μια πλακέτα σταθεροποίησης τάσης. Η συγκεκριμένη διάταξη τροφοδοτεί ταυτόχρονα την πλακέτα του μικροεπεξεργαστή με τάση 9V και τα υπόλοιπα ηλεκτρονικά μέρη του συστήματος με τάση 5V. Τέλος, η ανάλυση αρκετών από τα εξαρτήματα που χρησιμοποιήθηκαν στην συγκεκριμένη εφαρμογή θα γίνει στο κεφάλαιο 3, καθώς είναι ίδια η παρόμοια με το ρομποτικό σύστημα της πτυχιακής εργασίας.

2.4 Τετράτροχο αυτόνομο ρομποτικό σύστημα αποφυγής εμποδίων και συλλογής δεδομένων από τον περιβάλλοντα χώρο

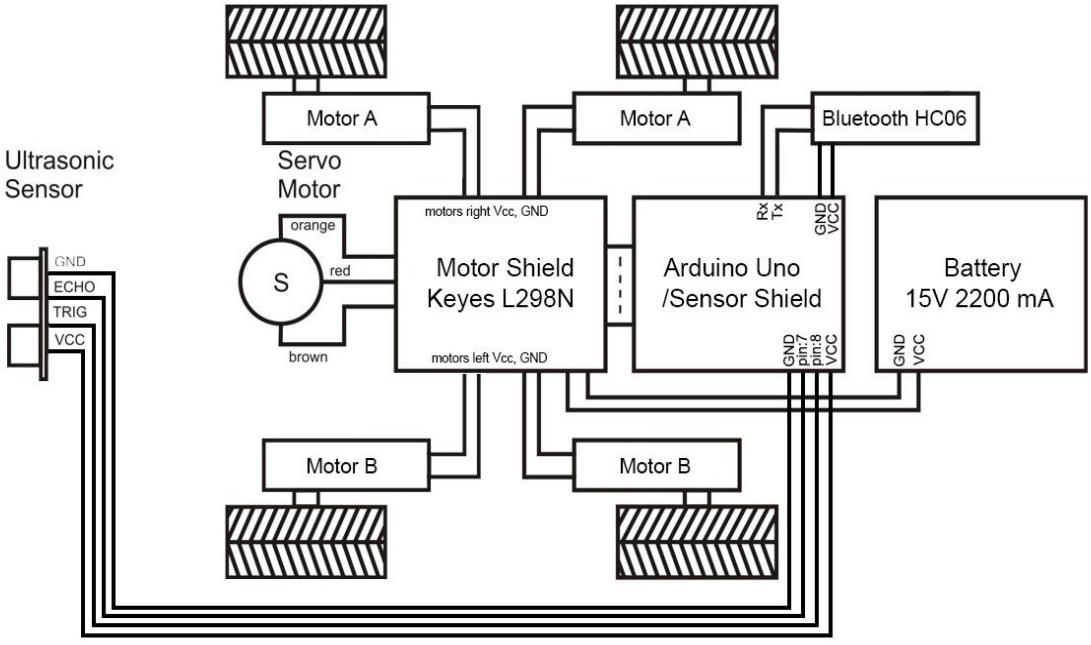
Στην παρούσα πτυχιακή εργασία έγινε έρευνα , μελετήθηκε και κατασκευάστηκε ένα τετράτροχο ρομποτικό σύστημα διαφορικής οδήγησης. Το συγκεκριμένο σύστημα θα έχει την δυνατότητα πλοήγησης μέσα σε έναν περιβάλλοντα χώρο, αποφεύγοντας τα εμπόδια, καθώς και θα είναι ικανό να συλλέγει δεδομένα/πληροφορίες από τον χώρο. Ακόμα είναι ικανό να συνδέεται ασύρματα με τον χρήστη μέσω κάποιου Η/Υ η συστήματος. Την ανάλυση της εφαρμογής την χωρίζουμε σε 3 τμήματα. Το πρώτο τμήμα αναφέρεται στην τεχνική της πρακτικής σχεδίασης, καθώς και στα υλικά που χρησιμοποιήθηκαν, στοχεύοντας στην ποιότητα, την διαθεσιμότητα και την οικονομία. Το δεύτερο τμήμα αναφέρει το προγραμματιστικό μέρος της κατασκευής καθώς το ρομποτικό σύστημα και όλα τα επιμέρους τμήματα του, ελέγχονται από έναν μικροελεγκτή, ο οποίος με την σειρά του είναι υπεύθυνος για τις λειτουργίες του συστήματος. Στο τρίτο τμήμα γίνεται αναφορά στον προγραμματισμό και την επεξεργασία των δεδομένων που συλλέγει το σύστημα μας. Στα επόμενα κεφάλαια θα ακολουθήσει η ανάλυση όλων των τμημάτων της κατασκευής της πτυχιακής εργασίας.

ΚΕΦΑΛΑΙΟ 3^{Ο}

Σχεδιασμός τετράτροχου ρομποτικού συστήματος και ανάλυση τεχνικών και ηλεκτρονικών εξαρτημάτων

3.1 Εισαγωγή

Όπως αναφέραμε στο προηγούμενο κεφάλαιο, η βασική αρχή στην σχεδίαση του ρομποτικού συστήματος, είναι η μη επανδρωμένη κίνηση, αυτόματη πλοήγηση και αποστολή δεδομένων για επεξεργασία (χαρτογράφηση, εντοπισμός), καθώς και τηλεχειρισμός. Για να επιτευχθεί αυτό χρειάστηκε να συγχρονιστούν αισθητήρες, μοτέρ, τα οποία οδηγούνται από κάποιον μικροελεγκτή. Στο κεφάλαιο 3 θα ασχοληθούμε με την ανάλυση των εξαρτημάτων που αποτελούν την κατασκευή. Η έρευνα αγοράς για την απόκτηση των εξαρτημάτων έγινε βάσει οικονομικών κριτηρίων, εστιάζοντας επίσης στην ποιότητα και την διαθεσιμότητα των εξαρτημάτων. Αυτό έγινε γιατί πριν φτάσουμε στην διεκπεραίωση της κατασκευής έγιναν πολλά πειράματα και μετρήσεις, με αποτέλεσμα κάποια εξαρτήματα να καταστρέφονται. Έτσι η άμεση αντικατάσταση και η αντοχή των εξαρτημάτων ήταν αναγκαία. Παρακάτω ακολουθεί η εικόνα 3.1. οπού φαίνεται το διάγραμμα της κατασκευής, της σύνδεσης, καθώς και τα εξαρτήματα που την αποτελούν.



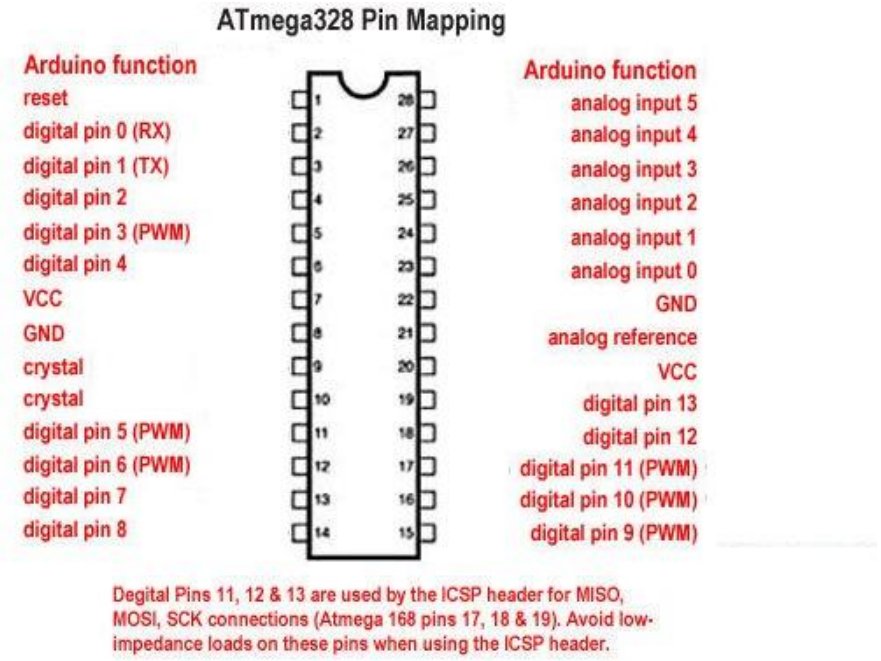
Εικόνα 3.1: Διάγραμμα ρομποτικού συστήματος.

3.2 Κεντρική μονάδα ελέγχου

Το σημαντικότερο κομμάτι στην κατασκευή του τετράτροχου ρομποτικού συστήματος είναι το αναπτυξιακό **ARDUINO UNO.** Εξυπηρετώντας ως κεντρική μονάδα ελέγχου του συστήματος, βασίζεται στον μικροελεγκτή **ATmega328P.** Ιταλικής κατασκευής, το **UNO** στο όνομα του αναπτυξιακού σηματοδοτεί την κυκλοφορία του λογισμικού καθώς και του αναπτυξιακού αφού ήταν το πρώτο με σειριακή σύνδεση (**USB**).

Ξεκινώντας την ανάλυση του αναπτυξιακού θα αναφερθούμε στο μικροελεγκτή **ATmega328P**, το οποίο είναι ενσωματωμένο στην πλακέτα του **ARDUINO UNO.** Το **ATmega328P** είναι ένα υψηλής απόδοσης και χαμηλής ενέργειας AVR μικροελεγκτής των 8 bit. Η τάση λειτουργείας του κυμαίνεται από 1,8**V** ώς 5.5**V**. Ο βαθμός ταχύτητας του είναι 0-20**MHz** στην προαναφερόμενη τάση λειτουργείας. Το **ATmega328P** είναι εκ των προτέρων προγραμματισμένος (Boot loader), ώστε να προγραμματίζεται εύκολα χωρίς την χρήση κάποιου εξωτερικού προγράμματος οδήγησης.

Για την επικοινωνία με υπολογιστή, άλλο αναπτυξιακό η άλλους μικροελεγκτές χρησιμοποιεί το πρωτόκολλο επικοινωνίας **STK500.** Επίσης παρέχει **UARTTTL** σειριακή επικοινωνία (5V), η οποία επιτυγχάνεται με τα ψηφιακά ποδαράκια 0(**RX)** και 1**(TX)** τα οποία βρίσκονται πάνω στο αναπτυξιακό **ARDUINO UNO.** Οι παραπάνω ακροδέκτες είναι ο δέκτης **(ReceiverX)** και ο πομπός **(TransmitterX)** αντίστοιχα. Η συγκεκριμένη **TTL USB** σύνδεση είναι μια συστοιχία καλωδίων (μετατροπή) οπού παρέχει συνδεσιμότητα μεταξύ της **USB** σύνδεσης και της σειριακής σύνδεσης **UART**. Ακόμα, οι αριθμοί 0 και 1 σηματοδοτούν την ορθή σύνδεση των ψηφιακών ακροδεκτών **(PIN)** πάνω στο αναπτυξιακό. Στην συνέχεια στην εικόνα 3.1 ακολουθεί το σχηματικό των ακροδεκτών του μικροελεγκτή.

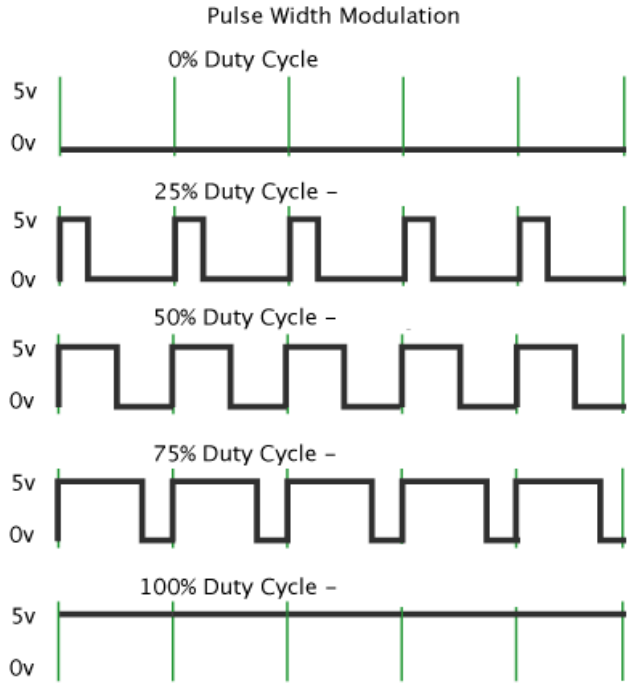


Εικόνα 3.2: Σχηματικό ακροδεκτών ολοκληρωμένου **ATmega328P** *[9]*.

Το ολοκληρωμένο **ATmega328P**, όπως προαναφέραμε είναι ενσωματωμένο στο αναπτυξιακό **ARDUINO UNO**, το οποίο λειτουργεί ως κεντρική μονάδα ελέγχου του

τετράτροχου ρομποτικού συστήματος. Πάνω σε αυτό συνδέονται και συντονίζονται όλα τα επιμέρους τμήματα του συστήματος, (αισθητήρες, DC μοτέρ, σειριακή επικοινωνία ασύρματη επικοινωνία ,σερβομηχανισμοί κ.α.). Τα επιμέρους τμήματα τις κατασκευής θα αναλυθούν μετά την ανάλυση του αναπτυξιακού παρακάτω.

Το **ARDUINO UNO,** αποτελείται από 14 ψηφιακά ¨ποδαράκια¨ **(Pins)** Εισόδου/Εξόδου εκ των οποίων τα 6 μπορούν να χρησιμοποιήσουν διαμόρφωση εύρους παλμών (**Pulse Width Modulation**) ως εξόδοι. Μια **PWM** κυματομορφή αποτελεί μια περιοδική κυματομορφή η οποία έχει δυο τμήματα. Το τμήμα **ON** οπού έχουμε την μέγιστη τιμή και το τμήμα **OFF**, στο οποίο η τιμή είναι μηδέν. Το **ON** τμήμα ονομάζεται **Duty Cycle** και μετριέται σε μονάδες χρόνου (**ms,us** κλπ.), ή σε ποσοστό (%) επι της περιόδου. Αυτή η εναλλαγή μεταξύ **ON** και **OFF** μπορεί να προσομοιώσει ένα εύρος τάσεων, για παράδειγμα 0 ως 5**V.** Στην εικόνα 3.2 φαίνεται η **PWM** κυματομορφή.



Εικόνα 3.3 : PWM κυματομορφή *[10]*.

Στην περίπτωση του τετράτροχου ρομποτικού συστήματος χρησιμοποιούμε τις **PWM** κυματομορφές προκειμένου να ελέγξουμε τις στροφές (ταχύτητα ,κατεύθυνση) των κινητήρων μας. Αυτό το επιτυγχάνουμε ελέγχοντας το ποσοστό της ισχύος που εφαρμόζεται στο φορτίο (κινητήρες).

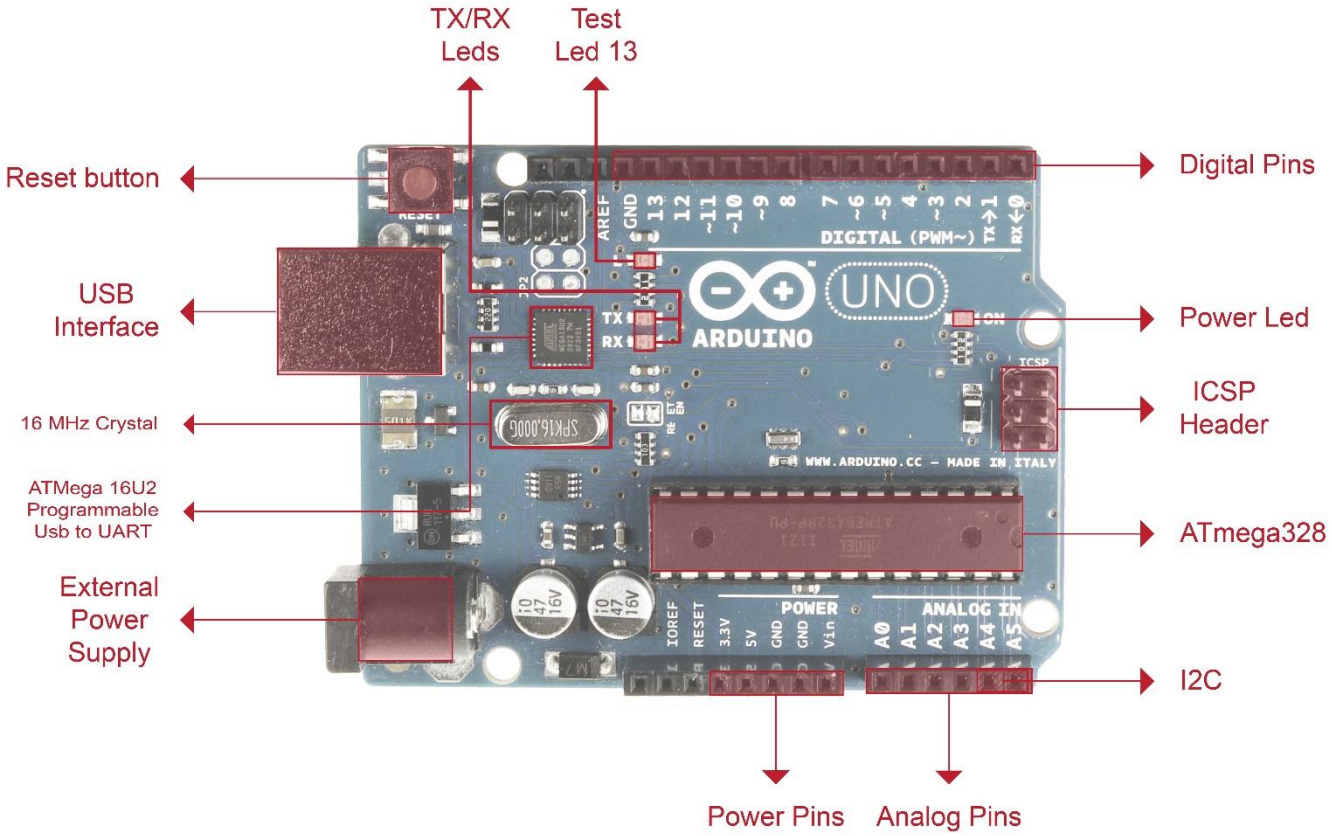
Επίσης ο **UNO** περιέχει 6 αναλογικά ποδαράκια (**Pin**) τα οποία συμπεριφέρονται ως είσοδοι/έξοδοι, ανάλογα με την χρήση που τους κάνουμε. Τα **Pin** βρίσκονται στις θέσεις **A0-A5** του αναπτυξιακού ενώ μπορούν να παρέχουν ανάλυση των 10 **bit.** Όλα τα **Pin** του αναπτυξιακού λειτουργούν στα 5**V** και κάθε ποδαράκι μπορεί να μας παρέχει η να δέχεται 20**mA**, όπως το συνιστά ο κατασκευαστής. Ακόμα, κάθε ποδαράκι έχει και μια εσωτερική PULL_UP αντίσταση των 20-50**KΩ**(**Ohm**). Χρειάζεται να σημειωθεί ότι το μέγιστο ρεύμα δεν θα πρέπει να ξεπερνάει τα 40mA , αλλιώς υπάρχει κίνδυνος καταστροφής του μικροελεγκτή.

Υπάρχει ακόμα ενσωματωμένος στο αναπτυξιακό ένας κρύσταλλος συχνοτήτων χαλαζία στα 16**MHz**. Ο συγκεκριμένος κρύσταλλος χρησιμοποιείται ουσιαστικά ως ταλαντωτής και συντονίζει το ρολόι του μικροελεγκτή. Επίσης το αναπτυξιακό διαθέτει μια θύρα **USB** οπού χρησιμοποιείται για την επικοινωνία του με τον υπολογιστή.

Για να τροφοδοτήσουμε το αναπτυξιακό χρησιμοποιούμε τον υποδοχέα τάσης (**Vin Pin**), το καλώδιο **USB** επικοινωνίας, η κάποια εξωτερική πηγή τροφοδοσίας όπως μπαταρία, AC to DC προσαρμογέα κ.α. Η τάση λειτουργίας του, βάσει κατασκευαστή είναι από 6**V** ως 12**V**, με το όριο τροφοδοσίας να είναι στα 20**V.** Εάν όμως το αναπτυξιακό τροφοδοτηθεί με 20**V**, ο σταθεροποιητής τάσης ο οποίος είναι ενσωματωμένος στον **UNO** θα ανεβάσει υψηλής θερμοκρασίες, με την πιθανότητα να προκληθούν βλάβες στο κύκλωμα. Από την άλλη μεριά εάν τροφοδοτήσουμε με λιγότερα από 6V ενδέχεται η λειτουργία του κυκλώματος να είναι ασταθής. Αξίζει να σημειωθεί ότι στην περίπτωση που τροφοδοτούμε τον **UNO** από μια θύρα **USB** δεν θα μπορέσουμε να αξιοποιήσουμε ολόκληρη την ισχύ, σε φορτία που απαιτούν πολύ ρεύμα. Αυτός είναι και ο λόγος που χρησιμοποιούμε εξωτερική τροφοδοσία. Βέβαια για φορτία μικρά σε ισχύ, η λειτουργία του αναπτυξιακού είναι ορθή.

Το **Pin** τροφοδοσίας με όνομα 3,3**V** μας βγάζει τάση μόνο 3,3**V** ενώ το **Pin, GND** σηματοδοτεί την γείωση. Το **UNO** διαθέτει ένα κουμπί (button), ώστε να κάνει επαναφορά (Reset), του προγράμματος που του έχουμε φορτώσει. Για παράδειγμα στην περίπτωση κάποιου προβλήματος της λειτουργίας του προγράμματος. Επίσης υπάρχουν 2 **LED** όπου το ένα σηματοδοτεί την τροφοδοσία του αναπτυξιακού, ενώ το άλλο μπορούμε να το χρησιμοποιήσουμε για τον έλεγχο της ορθής λειτουργίας του. Επίσης υπάρχουν οι εξωτερικοί διακόπτες **2** και **3** οι οποίοι χρησιμοποιούνται για να ενεργοποιήσουν κάποιο interrupt σε κατάσταση low.

Τα ποδαράκια **10**(**SS**), **11**(**MOSI**), **12**(**MISO**), **13** (**SCK**) περιφερειακή επικοινωνία, (Serial Peripheral interface), τα οποία καλούνται **SPI.** Τέλος για να φορτώνουμε τα προγράμματα στον **UNO** χρησιμοποιούμε μια μνήμη **32 KB Flash Memory** η οποία είναι ενσωματωμένη. Παρακάτω ακολουθεί η Εικόνα 3.3 οπού μας δείχνει ένα αναπτυξιακό **Arduino Uno.**



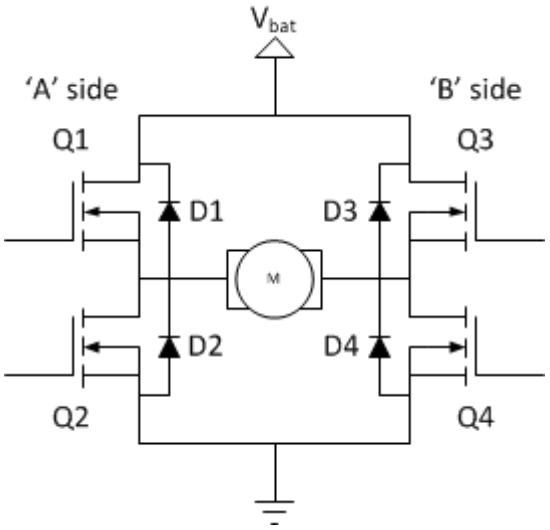
Εικόνα 3.4: Αναπτυξιακό Arduino Uno *[11]*

Επιλέξαμε να εργαστούμε στο συγκεκριμένο ρομποτικό σύστημα με το αναπτυξιακό **Arduino Uno** λόγω των πλεονεκτημάτων του, συγκεκριμένα:

- Επειδή ο σχεδιασμός του **UNO** είναι ανοιχτού τύπου. Έτσι, η συλλογή πληροφοριών σχετικά με την λειτουργία του(προεκτάσεις, δυνατότητες) και ο προγραμματισμός του(αλγόριθμοι, βιβλιοθήκες) ήταν πολύ εύκολη. Ανοιχτού τύπου σχεδιασμός σημαίνει ότι το σχηματικό, το **Datasheet** (Βιβλίο δεδομένων), σχηματικά των αισθητήρων κλπ. είναι εύκολα προσβάσιμα στο διαδίκτυο από ελεύθερους χρήστες.
- Άλλο ένα πλεονέκτημα είναι η εξαιρετική διαχείριση της ενέργειας λόγω του ενσωματωμένου σταθεροποιητή τάσης.
- Η επικοινωνία μέσω **USB** θύρας με τον υπολογιστή μας καθιστά εύκολη την φόρτωση των προγραμμάτων που επιθυμούμε. Επίσης χρησιμοποιώντας την σειριακή θύρα, μπορούμε εύκολα να στέλνουμε και να διαβάζουμε τα δεδομένα που θέλουμε.
- Η αποθήκευση των προγραμμάτων γίνεται στην Onboard μνήμη (**Flash Memory**) των **32KB**
- Τέλος το κοστολόγιο για το αναπτυξιακό και τους αισθητήρες ήταν ιδιέτερα χαμηλό.

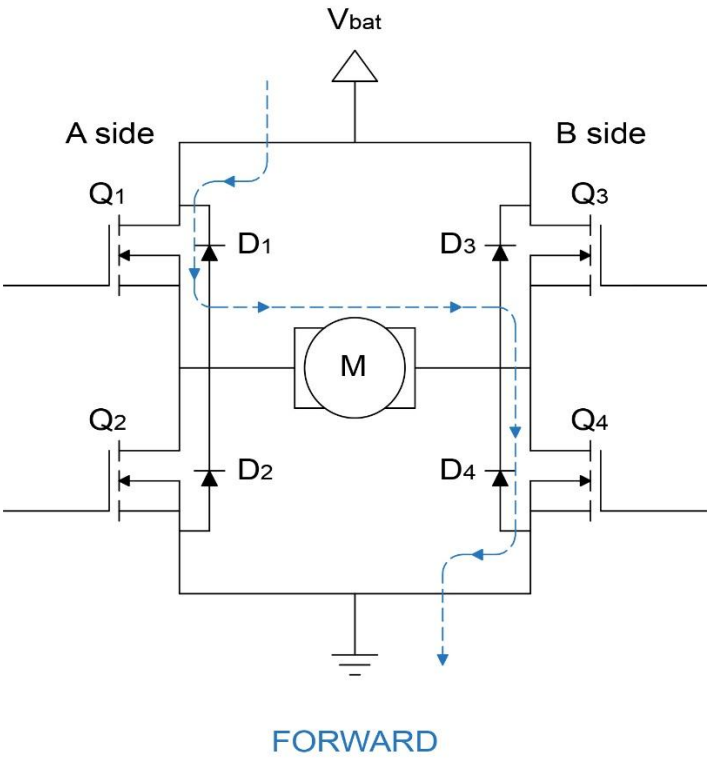
3.3 Motor Shield Keyes LN298N (Οδηγός Μοτέρ/Κινητήρων)

Για να ελέγξουμε τους 4 κινητήρες του ρομποτικού συστήματος χρησιμοποιούμε το **Module** (Διάταξη) **Motor Shield Keyes LN298N.** Το συγκεκριμένο Module, βασίζεται στο τσιπ **LN298N. Το LN298N**, αποτελείται από 2 γέφυρες Η, γνωστές και ως **H-Bridge**. Αυτή η κυκλωματική διάταξη επιτρέπει στο τσιπ να οδηγήσει επαγωγικά φορτία όπως ρελέ, DC motor, κ.α. Για τον λόγο αυτόν η **H-Bridge,** είναι μια ευρέως γνωστή συνδεσμολογία και χρησιμοποιείται σε πολλές ρομποτικές κατασκευές. Έχει πάρει το όνομα της από την διάταξη της καθώς μοιάζει με ¨Η¨, αφού έχει στα αριστερά και δεξιά από ένα ζευγάρι διακοπές, οπού συνήθως είναι **διπολικά**, ή **FET transistor,** ενώ στο κέντρο έχει το φορτίο, όπως στην εικόνα 3.4. Οι δίοδοι που χρησιμοποιούνται για προστασία, είναι συνήθως **Schotky**.

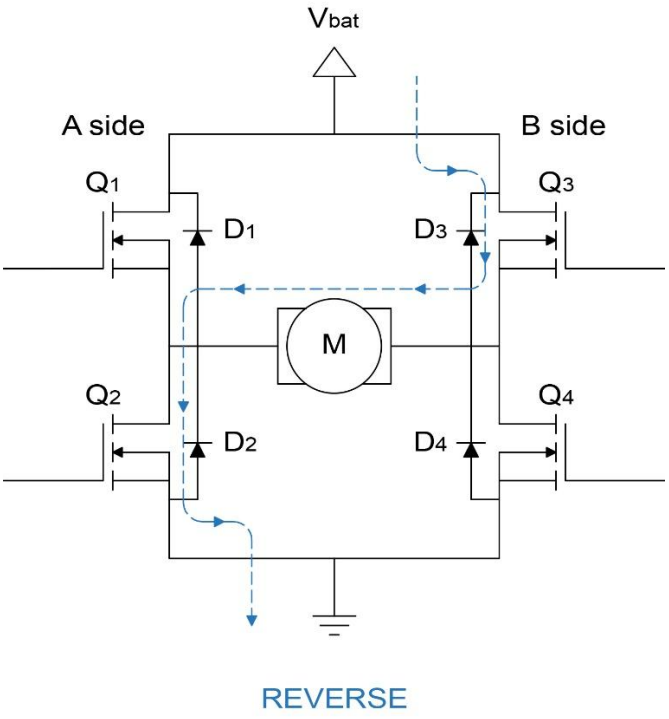


Εικόνα 3.5: Κυκλωματική Διάταξη H-Bridge *[12]*.

Η λειτουργία της είναι απλή, καθώς όταν τα **transistor** της εικόνας 3.5 **Q1** και **Q4** είναι σε ενεργή κατάσταση λειτουργίας, η αριστερή πλευρά του μοτέρ είναι συνδεδεμένη στην τροφοδοσία, ενώ η δεξιά πλευρά στην γείωση. Αυτό έχει ως αποτέλεσμα το μοτέρ μας να κινείται αριστερόστροφα (μπροστινή κίνηση). Στην αντίθετη περίπτωση όταν τα τρανζίστορ **Q2** και **Q3** είναι σε λειτουργία, η δεξιά πλευρά του μοτέρ θα είναι συνδεδεμένη στην τροφοδοσία και η αριστερή στην γείωση. Με αποτέλεσμα την κίνηση του μοτέρ δεξιόστροφα (κίνηση προς τα πίσω). Η κίνηση του ρεύματος για τις δυο περιπτώσεις φαίνεται στις εικόνες 3.6, 3.7 .



Εικόνα 3.6: Πορεία ρεύματος στην αριστερόστροφη κίνηση.



Εικόνα 3.7: Πορεία ρεύματος στην δεξιόστροφη κίνηση.

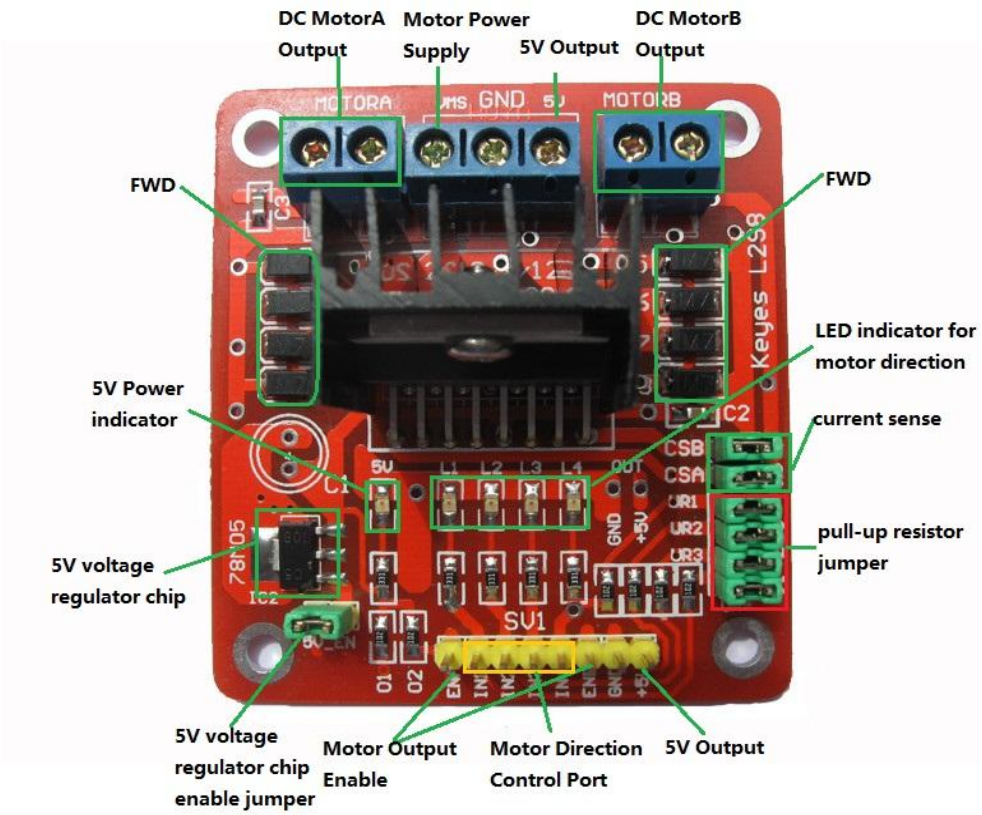
Το Motor Shield συνδέεται με το αναπτυξιακό **Arduino Uno** και συγκεκριμένα χρησιμοποιεί τα ψηφιακά ποδαράκια (PIN) **11,10,9,5**. Μέσω αυτών των Pin στέλνει **PWM** παλμούς ώστε να οδηγήσει τα μοτέρ, ρυθμίζοντας την ταχύτητα και την διεύθυνση. Να σημειωθεί ότι το **Duty Cycle** είναι ανάλογο της ταχύτητας των μοτέρ. Αυξάνοντας το, αυξάνουμε και την ταχύτητα περιστροφής των μοτέρ.

Μπορούμε να τροφοδοτήσουμε το Motor Shield από το Arduino_Uno(5v), αλλά αυτό θα μας οδηγήσει σε μη σωστή λειτουργία των κινητήρων. Αφού δεν υπάρχει αρκετή ισχύς. Για τον λόγο αυτό τροφοδοτούμε το **Motor Shield** με εξωτερική τροφοδοσία (5-12**V**), όπως αναγράφεται με πραγματική τιμή τάσης εσόδου 5-35 **V.**

Συνοψίζοντας το **Motor Shield** περιέχει:

- Δυο διαφορετικές συνδέσεις τροφοδοσίας**.**
- Το τσιπ **LN298N** παρέχει μέγιστο ρεύμα 2A Σε κάθε γέφυρα.
- Pin εξόδου για κάθε κινητήρα, καθώς και Pin (Enable) για τον έλεγχο της κατεύθυνσης των κινητήρων.
- **LED** που χρησιμοποιούνται ως δείκτες ένδειξης για τα 5**V,** καθώς και για την ορθή λειτουργία των μοτέρ.
- Κουμπί **Reset**, οπού στην πραγματικότητα είναι το ίδιο με την πλακέτα του **Arduino Uno** και επαναφέρει το πρόγραμμα που έχουμε φορτώσει, σε περίπτωση σφάλματος.
- Κλέμες **PCB** οπού μας διευκολύνει στην εξωτερική σύνδεση των εξαρτημάτων, στην περίπτωση μας συνδέουμε τα μοτέρ.

Λόγο της κατασκευής του το Motor Shield είναι ένα πολύ εύχρηστο εργαλείο οδήγησης επαγωγικών φορτίων. Συνδέοντας 2 μοτέρ σε κάθε Η-γέφυρα έχουμε την δυνατότητα να οδηγήσουμε και τα τέσσερα μοτέρ του ρομποτικού συστήματος**.** Στην εικόνα 3.8 παρουσιάζεται το **module Motor Shield Keyes LN298N.**



Εικόνα 3.8: Module Motor Shield Keyes LN298N *[13]*.

3.4 Αισθητήρας υπέρηχων HC-SR04 (Ultrasonic Sensor)

Σαν υπέρηχο καθορίζουμε εκείνο το κύμα το οποίο βρίσκεται πάνω από την μέγιστη συχνότητα που μπορεί να ακούσει το ανθρώπινο αυτί. Το οποίο έχει τη δυνατότητα να αντιλαμβάνεται ήχους με συχνότητες από 20**Hz** ως 20**kHz**, ενώ οι υπέρηχοι που αποστέλλει ο αισθητήρας είναι συχνότητας 40**kHz**.

Ο αισθητήρας υπερήχων βασίζεται στο φαινόμενο **Doppler.** Το φαινόμενο **Doppler,** είναι η παρατηρούμενη αλλαγή στην συχνότητα και το μήκος κύματος ενός κύματος από παρατηρητή που βρίσκεται σε σχετική κίνηση με την πηγή των κυμάτων. Οι συγκεκριμένοι αισθητήρες χρησιμοποιούνται σε εφαρμογές που χρειάζεται να είναι γνωστή η απόσταση του αισθητήρα, και γενικότερα της εφαρμογής από ένα πιθανό αντικείμενο/εμπόδιο.

Η λειτουργία των αισθητήρων είναι ίδια με αυτή των σόναρ και των ραντάρ. Εκτιμούν την απόσταση ενός στόχου λαμβάνοντας υπόψη τους την αντανάκλαση ενός ραδιοκύματος η ενός ηχητικού σήματος πάνω στο στόχο. Δημιουργούν υψηλής συχνότητας κύματα και χρησιμοποιώντας επιστρεφόμενο σήμα καθορίζουν την απόσταση ή ακόμα και την ταχύτητα του στόχου. Για να το επιτύχουν αυτό χρησιμοποιούν τον χρόνο που έκανε το σήμα για να καλύψει την απόσταση από τον αισθητήρα στο αντικείμενο και πίσω.

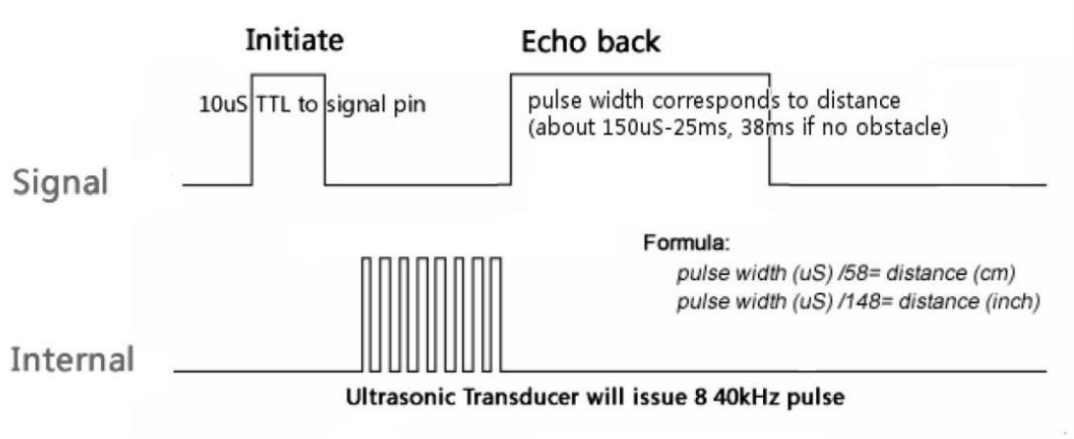
Πιο συγκεκριμένα, από την πηγή του κύματος, γίνεται αποστολή ενός υπερήχου στο χώρο. Όλα τα σώματα έχουν την ιδιότητα να αντανακλούν τους ήχους που προσπίπτουν πάνω τους. Έτσι αν υπάρχει κάποιο αντικείμενο του οποίου η ηχώ του υπερήχου που στάλθηκε, αντανακλάστηκε και επιστέφει προς τον αισθητήρα, γίνετε

αισθητή μέσω ενός μικροφώνου. Γνωρίζοντας το πόσο χρόνο διαρκεί το ταξίδι του υπερήχου, από την μετάδοση του έως και την λήψη του από το μικρόφωνο, μπορούμε να υπολογίσουμε πόση απόσταση διέσχισε ο υπέρηχος και διαιρώντας με το δυο μας επιστέφεται η απόσταση του αισθητήρα από το αντικείμενο.

Ο αισθητήρας υπερήχων **HC-SR04** παρέχει ένα εύρος μέτρησης από 2cm εως 400**cm** με ακρίβεια που αγγίζει τα 3**mm**. Το εξάρτημα αποτελείται από έναν πομπό (**Trigger**) και έναν δέκτη (**Echo**). Επίσης έχει ένα **Pin** για την τάση λειτουργίας του **Vcc** καθώς και ένα για την γείωση, **GND.** Στην εικόνα 3.9 βλέπουμε έναν αισθητήρα **HC-SR04**.

Εικόνα 3.9: Αισθητήρας υπέρηχων HC-SR04 με διαστάσεις (45*20*15mm) *[14]*

Για να λειτουργήσει χρησιμοποιεί παλμούς διάρκειας τουλάχιστον 10**us**. Στέλνει αυτόματα 8 παλμούς συχνότητας 40**kHz** και ανιχνεύει αν υπάρχει παλμός που επέστρεψε. Η απόσταση λαμβάνεται υπόψιν σύμφωνα με το χρόνο του σήματος στην αποστολή και λήψης του σήματος. Εξαρτάται βεβαίως από την ταχύτητα του ήχου που ανέρχεται στα 340**m/s.** Η τάση λειτουργίας του αισθητήρα είναι 5**V** και παρέχεται από τον μικροελεγκτή. Το ρεύμα λειτουργίας είναι 15**mA** και η συχνότητα λειτουργίας είναι 40Hz. Η γωνιά με μέτρησης που επιτυγχάνει είναι οι 30 μοίρες. Στην εικόνα 3.10 παρουσιάζεται ο τρόπος λειτουργίας του εξαρτήματος.



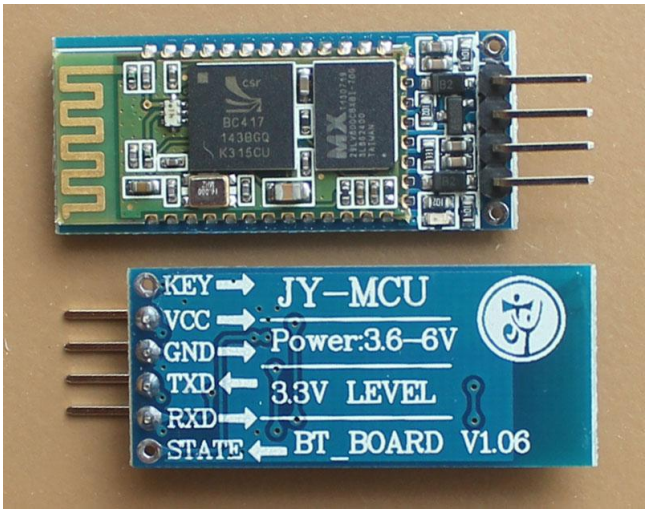
Εικόνα 3.10: Διάγραμμα λειτουργίας του αισθητήρα HC-SR04. *[15]*

3.5 Bluetooth Διάταξη (Module) HC-06

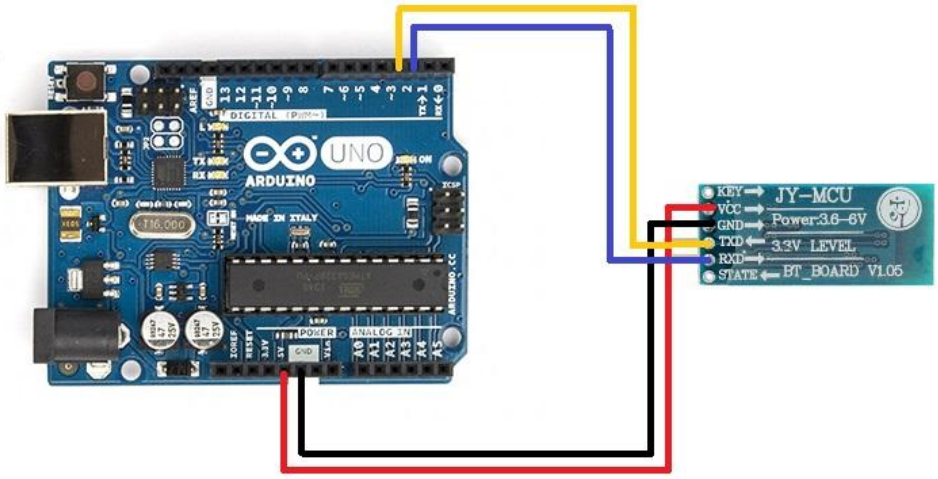
Μια από τις βασικές ιδιότητες του τετράτροχου ρομποτικού συστήματος είναι η απομακρυσμένη σύνδεση του με αλλά προγράμματα καθώς και υπολογιστές. Έτσι μπορούμε να επεξεργαστούμε τα δεδομένα που μας στέλνει το σύστημα καθώς και να το ελέγξουμε από απόσταση. Για να το επιτύχουμε αυτό χρησιμοποιούμε την διάταξη **Bluetooth HC-06 module**. Το **HC-06** είναι ένας σειριακής θύρας Bluetooth οδηγός ο οποίος βασίζεται στο μοντέλο επικοινωνίας **Master-Slave**. Σαν **master** εννοείται αυτός που ελέγχει την ροή των δεδομένων καθώς και τον προορισμό, ενώ ο **slave** είναι αυτός που δέχεται και απαντάει με δεδομένα. Ένα module που είναι ρυθμιζόμενο ως **master** μπορεί να συνδεθεί με περισσότερες από μια συσκευές σε λειτουργία **Slave.**

Το μοντέλο **HC-06** λειτουργεί σε ρόλο slave, έτσι μπορεί να συνδεθεί μόνο με ένα **Bluetooth** που είναι ρυθμιζόμενο ως master. Στην περίπτωση μας χρησιμοποιήσαμε το **Bluetooth** του ηλεκτρονικού υπολογιστή οπού και κάνουμε την επεξεργασία των δεδομένων. Η τάση λειτουργίας του είναι τα 5**Vcc** τα οποία τα παίρνει από τον **Arduino Uno**. Περιέχει άλλα 3 **pin** τα οποία είναι το **GND** για την γείωση, το **pin TXD** για τον Πομπό (**Transmitter**) αλλά και το **pin RXD** για τον δεκτή (**Receiver**).

Η συνδεσμολογία του είναι απλή καθώς το **TXD** συνδέεται στο ποδαράκι 0 (**Receiver**) του αναπτυξιακού, ενώ το **pin RXD** συνδέεται στο ποδαράκι 1 (**Transmitter**). Στην εικόνα 3.11 φαίνεται η διάταξη του **Bluetooth HC-06** καθώς και στην εικόνα 3.12 ο τρόπος σύνδεσης του με τον Arduino Uno.



Εικόνα 3.11: Bluetooth module HC-06 *[16]*.



Εικόνα 3.12: Τρόπος σύνδεσης Bluetooth HC-06 με το αναπτυξιακό Arduino UNO. *[17]*

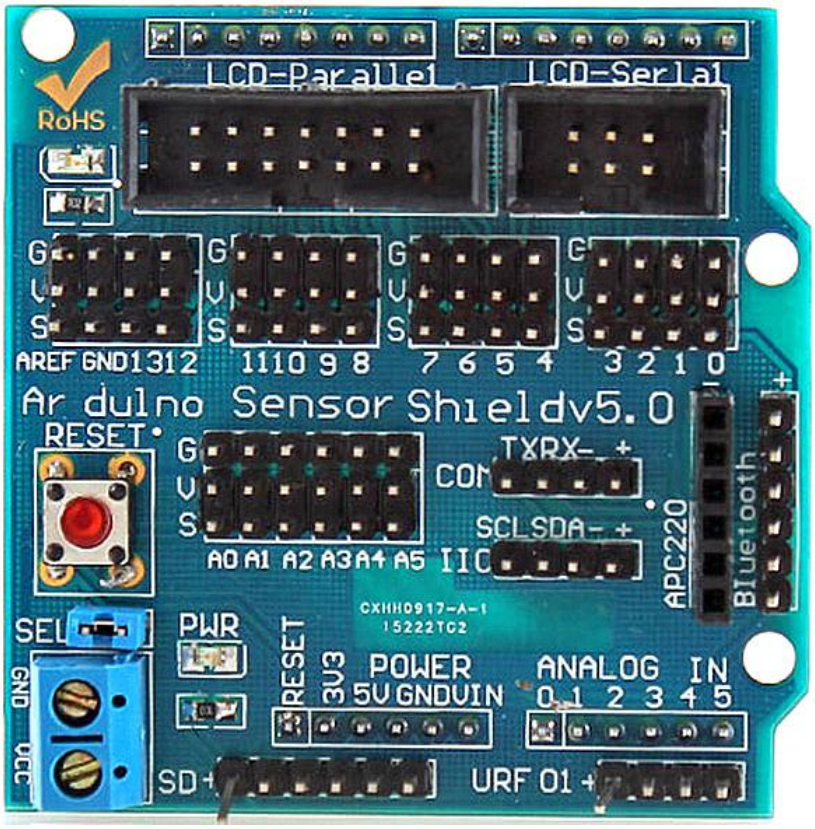
3.6 Arduino Sensor Shield

Το **Arduino Sensor Shield** λειτουργεί ως προέκταση του αναπτυξιακού Arduino Uno. Χρησιμοποιούμε την έκδοση V5, οπού μας επιτρέπει να συνδέσουμε περισσοτέρους αισθητήρες ώστε να λειτουργήσουμε το ρομποτικό σύστημα. Το συγκεκριμένο module ενσωματώνεται στο αναπτυξιακό, UNO και δεν αποτελεί κάποια ενεργή συσκευή. Η λειτουργία του είναι απλή καθώς απλά συνδέει τα **PIN** του Arduino Uno σε περισσότερα PINS. Επίσης τροφοδοτείται από το αναπτυξιακό. Η έκδοση V5, πλεονεκτεί σε σχέση με τις παλιότερες γιατί διαθέτει 2 θύρες επικοινωνίας (σειριακά, I2C) ταυτόχρονα. Συγκεκριμένα το module περιέχει:

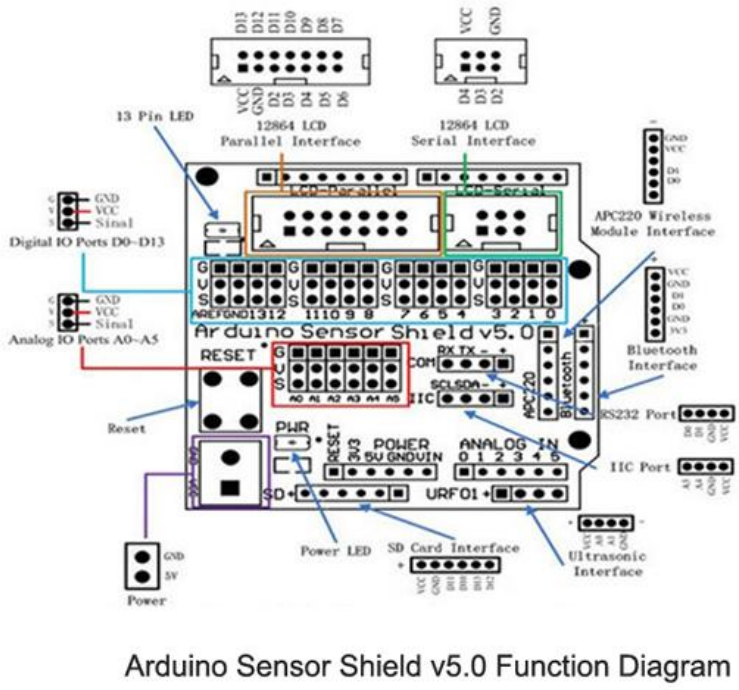
- Ψηφιακούς ακροδέκτες εισόδου/εξόδου (**In**/**Out**). Αριθμημένα από το 0 ως το 13.
- Αναλογικούς ακροδέκτες εισόδου (**Input**) αριθμημένους από **A0** ως **A5**.

- 2 θύρες επικοινωνίας (**Serial, I2C**).
- Ένα Led, που λειτουργεί σαν οδηγός, συνδεδεμένο στο **Pin 13**.
- Κουμπί **RESET** το οποίο έχει την ίδια λειτουργία με του αναπτυξιακού, αφού επαναφέρει το πρόγραμμα, ξεκινώντας το από την αρχή σε περίπτωση βλάβης.
- **Power LED,** το οποίο μας δείχνει την λειτουργία του **Module**.
- **Bluetooth** διεπαφή (**Interface**) ώστε να μπορούμε να συνδέσουμε το Bluetooth module.
- **Serial, I2C** διεπαφή (**Interface**)
- Τροφοδοσία **Power IN**

Στην συνέχεια ακολουθεί στην εικόνα 3.13 το Arduino Sensor Shield V5.



Εικόνα 3.13: Arduino Sensor Shield V5. *[18]*



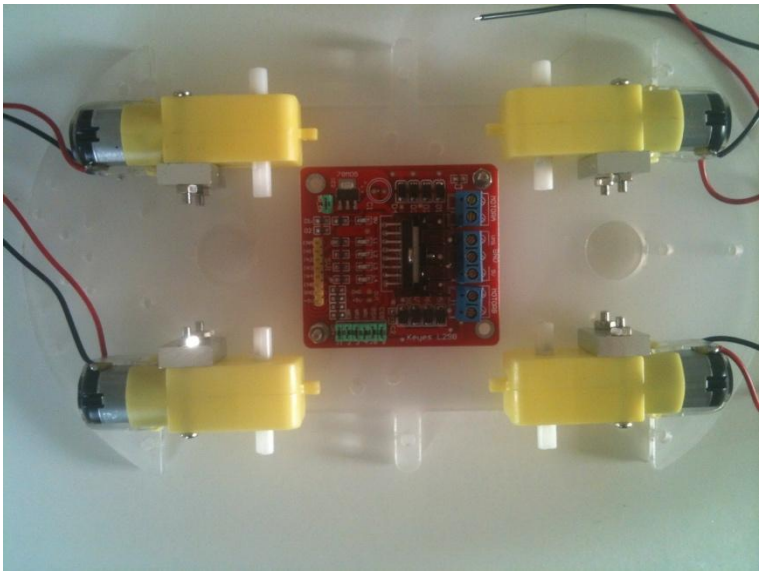
Arduino Sensor Shield v5.0 Function Diagram

Εικόνα 3.14: Σχηματικό Διάγραμμα Arduino Sensor Shield. *[19]*

3.7 Μηχανολογικά μέρη και εξαρτήματα του τετράτροχου ρομποτικού συστήματος

Όπως αναφέραμε και στην εισαγωγή του κεφαλαίου, τα μηχανολογικά μέρη του συστήματος, επιλέχθηκαν βάσει την διαθεσιμότητα τους, την αντοχή τους, και φυσικά το κόστος τους. Στην επιλογή των σασί εστιάσαμε στην αντοχή τους, καθώς θα μπορούσαμε να πούμε ότι λειτουργούν ως σκελετός του αμαξιδίου, αφού πάνω τους ενσωματώνονται όλα τα τμήματα του συστήματος. Για παράδειγμα αισθητήρες, αναπτυξιακό **Arduino Uno,** μπαταρίες, σερβομηχανισμοί. Χρησιμοποιήσαμε 2 σασί γιατί η κατασκευή μας έχει 2 επίπεδα. Στο πρώτο είναι τοποθετημένο το **Motor Shield** και στο δεύτερο το αναπτυξιακό **UNO.** Με αυτό τον τρόπο η καλωδίωση ήταν ευκολότερη και οργανωμένη. Επίσης ο χώρος που μας προσφέρουν τα δυο σασί ήταν ακόμα ένα κριτήριο, καθώς θέλουμε να έχουμε την δυνατότητα μελλοντικής αναβάθμισης του συστήματος μας.

Για λόγους ευκολίας στην συναρμολόγηση επιλέξαμε προκατασκευασμένους κινητήρες ώστε να ενσωματώνονται στις ρόδες. Το υλικό των κινητήρων αλλά και των ροδών είναι το πλαστικό καθώς είναι ελαφρύ και εμείς στοχεύομε στην ευκινησία. Η τάση λειτουργίας των κινητήρων είναι από 5 μέχρι 12**V** και απαιτούν 150**mA.** Στην εικόνα 3.15 που ακολουθεί φαίνεται το σασί του πρώτου επιπέδου με τους 4 κινητήρες. Ενώ στην εικόνα 3.16 φαίνεται ο τύπος της ρόδας που χρησιμοποιήσαμε στο τετράτροχο όχημα..



Εικόνα 3.15: Σασί πρώτου επιπέδου μαζί με τους κινητήρες και το Motor Shield.

Εικόνα 3.16: Κινητήρας μαζί με τροχό του συστήματος.

Ακόμα επιλέξαμε να στερεώσουμε τον αισθητήρα υπέρηχων πάνω σε έναν σερβομηχανισμό που εκτελεί περιοδική κίνηση 180 μοιρών. Δίνοντας στο σύστημα μας την δυνατότητα να σαρώνει σε μεγαλύτερη γωνία για εμπόδια κλπ. Τέλος και για να τροφοδοτήσουμε το σύστημα μας χρησιμοποιήσαμε μια μπαταριά λιθίου **LIRO** των 15**V** και 2200**mA**, δίνοντας την απαραίτητη τάση για να οδηγηθούν όλα τα φορτία με επάρκεια.

ΚΕΦΑΛΑΙΟ 4ο

Προγραμματισμός τετράτροχου ρομποτικού συστήματος

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα εστιάσουμε στον προγραμματισμό του ρομποτικού συστήματος, προκειμένου να οδηγήσουμε και να συντονίσουμε όλες τις λειτουργείες του, καθώς και να μπορέσουμε να επεξεργαστούμε τα δεδομένα που δεχόμαστε από το σύστημα. Θα αναφερθούμε στις πλατφόρμες προγραμματισμού και θα αναλύσουμε τους αλγορίθμους που οδηγούν το σύστημα. Τα προγράμματα που χρησιμοποιήθηκαν είναι το **Arduino**, το πρόγραμμα **Matlab** καθώς και το **BlueSoleil.** Θα χωρίσουμε την δομή του προγραμματισμού σε 3 τμήματα. Το πρώτο τμήμα αναφέρεται στον προγραμματισμό των επιμέρους εξαρτημάτων καθώς και του τετράτροχου ρομποτικού συστήματος, όσων αφορά τις λειτουργείες του. Το δεύτερο τμήμα αναφέρεται στην δημιουργία της ασύρματης επικοινωνίας μεταξύ του συστήματος και του προγράμματος επεξεργασίας δεδομένων **MATLAB** η ακόμα και κάποιο κινητό τηλέφωνο, για την λειτουργεία του τηλεχειρισμού. Τέλος, το τελευταίο τμήμα αναφέρεται στην επεξεργασία των δεδομένων που δεχόμαστε από το σύστημα.

4.2 Προγραμματιστικό περιβάλλον Arduino

Σε αυτήν την ενότητα θα αναλύσουμε το σημαντικότερο ίσως λογισμικό προγραμματισμού που χρησιμοποιήσαμε, το **Arduino.** Η έμφαση δίνεται γιατί στην συγκεκριμένη πλατφόρμα έχουν γραφτεί όλοι οι αλγόριθμοι οδήγησης των εξαρτημάτων (αισθητήρες, μοτέρ, κ.α.), αλλά και το πρόγραμμα που οδηγεί τον ελεγκτή **Arduino Uno,** συντονίζοντας όλες τις λειτουργίες του ρομποτικού συστήματος.

Το συγκεκριμένο λογισμικό είναι ανοιχτού τύπου. Δημιουργώντας έτσι την ευκολία εκμάθησης του προγράμματος αλλά και την εύρεση πληροφοριών, βιβλιοθηκών ακόμα και προγραμμάτων, στο διαδίκτυο από ελευθέρους χρήστες. Στηρίζεται στην ευκολία χρήσης και συνδυασμού του λογισμικού με τα εξαρτήματα που απαρτίζουν τα διάφορα συστήματα. Βασισμένο στην γλώσσα προγραμματισμού **C**/**C++** το λογισμικό του **Arduino** θεωρείται μια γλώσσα υψηλού επίπεδου καθώς είναι αρκετά περιγραφική, με αποτέλεσμα να είναι πιο προσιτή στον προγραμματιστή. Επίσης χρησιμοποιεί βιβλιοθήκες και ρουτίνες, κάτι που καθιστά την δημιουργία προγραμμάτων πιο εύκολη.

Στο κομμάτι του προγραμματισμού, χωρίζεται σε 3 τμήματα. Το 1ο μέρος αναφέρει την **δομή** (**structure**), την οποία επιλέγουμε μέσα στο πρόγραμμα και σχετίζεται με τον τρόπο που γράφεται ο αλγόριθμος. Το 2ο μέρος αναφέρεται στις μεταβλητές που θα χρησιμοποιηθούν καθώς και το είδος τους. Τέλος το 3ο μέρος ονομάζεται ¨**Λειτουργία**¨ (**function**). Σαν **function** ορίζεται ένας έτοιμος αλγόριθμος ο οποίος μπορεί να καλείται στο κυρίως πρόγραμμα και να εκτελεί έναν συγκεκριμένο σκοπό.

4.3 Πρόγραμμα οδήγησης τετράτροχου ρομποτικού συστήματος σε περιβάλλον ARDUINO

Σε αυτή την ενότητα θα αναλυθεί ο αλγόριθμος ο οποίος είναι γραμμένος σε προγραμματιστικό περιβάλλον **Arduino**. Ο συγκεκριμένος αλγόριθμος οδηγεί τους κινητήρες του συστήματος καθώς και τους αισθητήρες, δίνοντας παράλληλα την δυνατότητα του εντοπισμού και της αποφυγής εμποδίων. Παρακάτω ακολουθεί η εικόνα **4.1** στην οποία απεικονίζεται και ο σχετικός αλγόριθμος.

Εικόνα 4.1: Αλγόριθμος κίνησης και αποφυγής εμποδίων σε περιβάλλον ARDUINO.

Ποιο συγκεκριμένα, ξεκινάμε δηλώνοντας τις απαραίτητες μεταβλητές, ώστε το πρόγραμμα να ¨καταλάβει¨ πια Pin (ψηφιακά/αναλογικά ποδαράκια), χρησιμοποιούμε για την οδήγηση των κινητήρων και των αισθητήρων του συστήματος. Ακόμα, μεταβλητές οι οποίες δεν δηλώνουν την χρήση κάποιου Pin αλλά χρησιμοποιούνται μέσα στον αλγόριθμο θα πρέπει να προσδιορίζουν το είδος τους. Για παράδειγμα μεταβλητή που χρησιμοποιείται για κάποιον μετρητή δηλώνεται ως ακέραια, η πραγματική ή ακόμα και χαρακτήρας εφόσον οι τιμές που θα αποθηκεύονται σε αυτήν είναι οι ανάλογες. Οι μεταβλητές που αναφέρονται σε συνδέσεις με το αναπτυξιακό θα πρέπει να προσδιορίζουν την λειτουργεία τους (Έξοδοι η Είσοδοι). Η δομή στην οποία

γίνονται όλες οι αρχικοποιήσεις των μεταβλητών καλείται **SETUP** και εκτελείται μια φορά στην εκκίνηση ενός προγράμματος. Η εντολή μέσα στον αλγόριθμο είναι η **void setup().** Η περιοχή των δηλώσεων φαίνεται στην εικόνα 4.1 με το κόκκινο 3πλαίσιο. Ακόμα, προκειμένου να χρησιμοποιήσουμε κάποια βιβλιοθήκη θα πρέπει να δηλωθεί στην αρχή του προγράμματος, ώστε να μπορέσουμε να την καλέσουμε με το όνομα της, οπού την χρειαστούμε. Στην περίπτωση μας χρησιμοποιούμε την βιβλιοθήκη που οδηγεί τον σερβομηχανισμό, με όνομα **Servo.** Η σχετική εντολή είναι η **#include** ακολουθούμενη από το όνομα της βιβλιοθήκης**.** Η εντολή που προσδιορίζει την λειτουργία των Pin του αναπτυξιακού που χρησιμοποιούνται είναι η **PinMode.** Με την εντολή **digitalWrite** θέτουμε αρχικές τιμές στους κινητήρες μας και στην συγκεκριμένη περίπτωση την τιμή **LOW** ώστε να μην ξεκινήσει το σύστημα μας απότομα. Η εντολή αυτή χρησιμοποιεί PWM παλμούς (*σχετική αναφορά, κεφάλαιο 3*) προκειμένου να οδηγήσει κινητήρες και αισθητήρες. Τέλος, η εντολή **Serial. Begin** θέτει την ταχύτητα της σειριακής επικοινωνίας του αναπτυξιακού με τον υπολογιστή η με κάποιο άλλο λειτουργικό σύστημα. Να σημειωθεί ότι η ταχύτητα της επικοινωνίας μπορεί να διαφέρει από σύστημα σε σύστημα η από τις εφαρμογές που λειτουργούμε**.** Η τιμή μέσα στην παρένθεση αναφέρεται στην ταχύτητα της σύνδεσης.

Αφού λοιπόν γίνουν όλες οι αρχικοποιήσεις, θα συνεχίσουμε στην δομή της επανάληψης η αλλιώς **void loop** όπως είναι και η εντολή στο πρόγραμμα. Σε αυτό το σημείο θα γραφτεί ο κυρίως αλγόριθμος οπού θα επαναλαμβάνεται ατέρμονα. Αυτή είναι και η λειτουργεία της συγκεκριμένης δομής. Ξεκινώντας, ρυθμίζουμε την θέση του σερβομηχανισμού οπού πάνω του ενσωματώνετε ο αισθητήρας υπέρηχων. Μετά από δοκιμές καταλήξαμε ότι η καλύτερη επιλογή για την γωνιά σάρωσης, ξεκινάει από 45 μοίρες καταλήγοντας στις 140. Με την εντολή **for** η οποία είναι μια εντολή επανάληψης καταφέρνουμε να δημιουργήσουμε την κίνηση σάρωσης σε αυτές τις μοίρες. Να σημειωθεί ότι ο σερβομηχανισμός θα πρέπει να επανέρχεται στην αντίθετη κατεύθυνση, οπότε χρησιμοποιούμε την ίδια εντολή αντιστρέφοντας τις τιμές μας. Το κομμάτι του αλγορίθμου που ελέγχει τον σερβομηχανισμό φαίνεται στην εικόνα 4.1 στο πράσινο πλαίσιο. Στην συνέχεια, το πρόγραμμα οδηγεί τον αισθητήρα υπέρηχων ώστε σε συνδυασμό με τον σερβομηχανισμό που τον υποστηρίζει να έχουμε μια επιτυχή σάρωση**.**

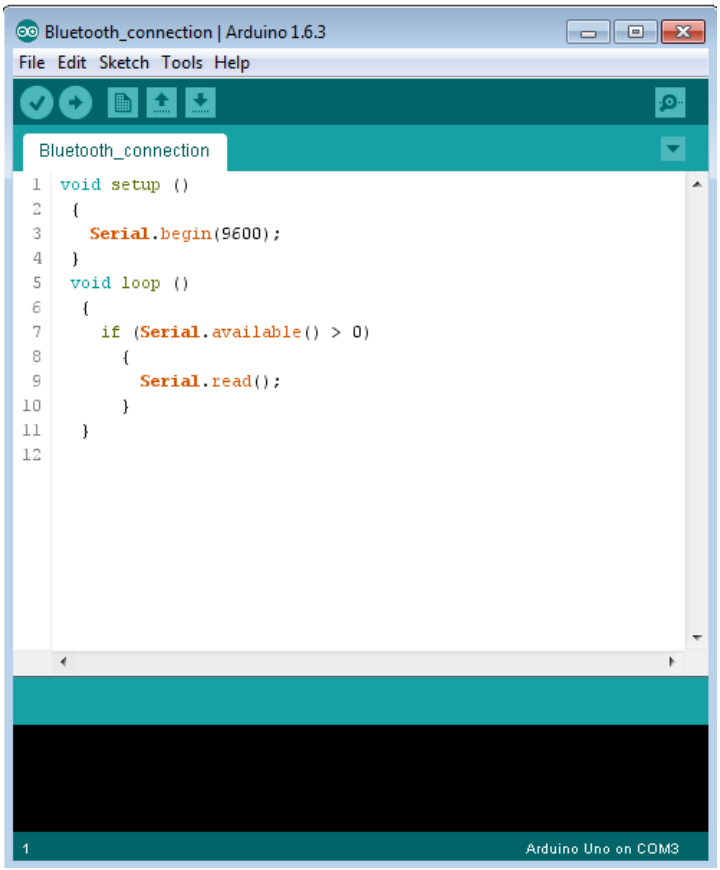
Με την εντολή **digitalWrite** στέλνουμε **PWM** παλμούς οπού ενεργοποιούμε τον αισθητήρα και συγκεκριμένα τον πομπό. Ο πομπός με την σειρά του στέλνει υπέρηχους στο περιβάλλον. Όταν ο υπέρηχος προσκρούσει σε κάποιο αντικείμενο, ένα μέρος του αντανακλάτε πίσω στην πηγή, οπού συλλέγετε από τον δέκτη. Η εντολή που υπολογίζει την απόσταση σε σχέση με τον χρόνο επιστροφής του υπερηχητικού κύματος είναι η **pulseIn.** Ο υπολογισμός τις τιμής αποθηκεύεται στην μεταβλητή **distance** την οποία έχουμε ορίσει σαν πραγματικό αριθμό. Στην συνέχεια με μια αριθμητική πράξη υπολογίζουμε την πραγματική απόσταση.

Το κομμάτι του προγράμματος που οδηγεί αυτήν την διαδικασία φαίνεται στην εικόνα 4.1 στο μπλε πλαίσιο. Η μεταβλητή της απόστασης ελέγχεται από μια συνθήκη οπού επιτρέπει το όχημα να αποφεύγει τα εμπόδια. Έτσι, όταν το όχημα βρεθεί κοντά σε ένα εμπόδιο/αντικείμενο οι κινητήρες παίρνουν εντολή να αλλάξουν κατεύθυνση και να αποφύγουν το εμπόδιο. Η εντολή που δίνει τους παλμούς στους κινητήρες είναι η **analogWrite.** Ρυθμίζοντας τον αριθμό που ακολουθεί την εντολή και το όνομα του κινητήρα που επιθυμούμε, μπορούμε να ελέγξουμε και την ταχύτητα του. Το τελευταίο σκέλος σε αυτόν τον αλγόριθμο είναι η συνθήκη οπού το αναπτυξιακό περιμένει μια τιμή από κάποιον εξωτερικό χρήστη για να στείλει τα δεδομένα στο πρόγραμμα με το

οποίο επικοινωνεί. Στην περίπτωση μας η τιμή αυτή είναι το 1, μόλις λοιπόν το Arduino Uno δεχθεί την τιμή 1 θα στείλει σειριακά τις τιμές της απόστασης καθώς και κάποιας άλλης μεταβλητής που επιθυμούμε, όπως για παράδειγμα την ταχύτητα. Το κομμάτι του αλγορίθμου οπού γίνεται αυτή η διαδικασία, φαίνεται στην εικόνα 4.1 στο μαύρο πλαίσιο.

4.4 Προγραμματισμός ασύρματης επικοινωνίας μέσω Bluetooth

Για να σταλούν τα δεδομένα ασύρματα, χρησιμοποιείται μια ζεύξη μεταξύ δυο διατάξεων **Bluetooth.** Όπως προαναφέραμε τα **Bluetooth** λειτουργούν με μοντέλο Αφέντη- Σκλάβου (**Master-Slave**). Ο προγραμματισμός του Slave, ο οποίος ενσωματώνεται στο τετράτροχο ρομποτικό σύστημα, γίνεται με την πλατφόρμα του **Arduino**. Στην περίπτωση μας ο Slave είναι η κυκλωματική διάταξη HC-06. Ο συγκεκριμένος αλγόριθμος θα πρέπει να ενσωματωθεί σε οποιοδήποτε σημείο της επαναληπτικής δομής μέσα στο κυρίως πρόγραμμα μας. Στην εικόνα 4.2 απεικονίζεται ο σχετικός αλγόριθμος.

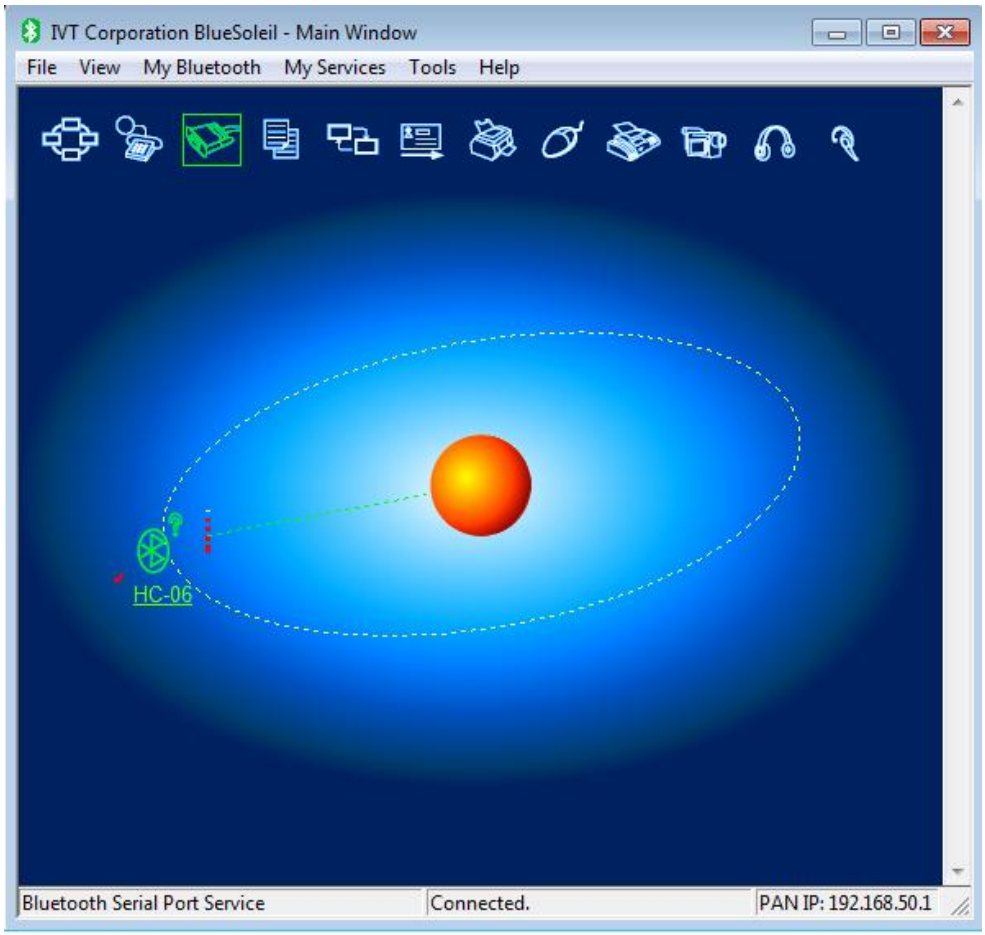


Εικόνα 4.2: Αλγόριθμος σύνδεσης Bluetooth διάταξης (Slave).

Ο παραπάνω αλγόριθμος ρυθμίζει την σειριακή επικοινωνία καθώς και ενεργοποιεί την **Bluetooth** διάταξη σε κατάστασης **Slave.** Για να προγραμματίσουμε τον Slave ξεκινάμε με την εντολή **Serial.begin** οπού θέτουμε και την ταχύτητα της επικοινωνίας, ρυθμίζοντας τον αριθμό της παρένθεσης. Στην συνέχεια φτιάχνουμε μια συνθήκη οπού

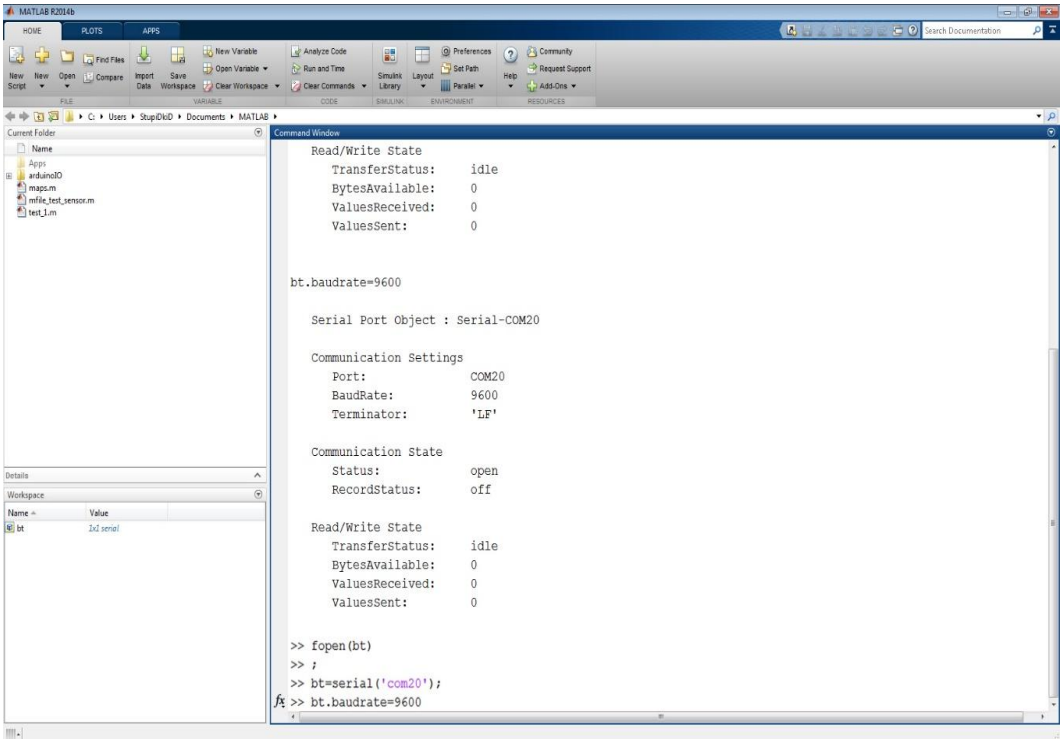
το **Bluetooth** θα αναζητά δεδομένα στην σειριακή θύρα. Έτσι με την εντολή **if** και την **Serial.available** ενεργοποιούμε το **Bluetooth** ώστε να δέχεται/αναζητά δεδομένα. Τέλος για να μπορέσουμε να διαβάσουμε τα δεδομένα χρησιμοποιούμε την εντολή **Serial.read**.

Η δεύτερη διάταξη **Bluetooth** τοποθετείται πάνω στον υπολογιστή οπού γίνεται η επεξεργασία των δεδομένων. Το **module** που χρησιμοποιούμε για τον υπολογιστή ονομάζεται **EDR** (Enchanted Data Transfer), που μας βοηθάει στην ενισχυόμενη μεταφορά δεδομένων. Η διάταξη λειτουργεί σε κατάσταση **Master**. Η ζεύξη των δυο **Bluetooth** γίνεται με την βοήθεια του προγράμματος **BlueSoleil**. Το συγκεκριμένο πρόγραμμα μας επιτρέπει να σαρώσουμε την ευρύτερη περιοχή για **Bluetooth** διατάξεις σε λειτουργεία **Slave**, καθώς και να συνδεθούμε με αυτά. Το **BlueSoleil,** μπορεί να συνδεθεί με σχεδόν όλες της ψηφιακές συσκευές που διαθέτουν Bluetooth σε λειτουργία **Slave** για την ανταλλαγή δεδομένων(κινητά τηλέφωνα, υπολογιστές). Λόγο του ότι η σύνδεση είναι προστατευμένη, χρειάζεται η εισαγωγή κάποιου συνθηματικού-κωδικού. Προκειμένου λοιπόν να προστατευτούν οι συνδέσεις που δημιουργούνται, κάποιες **Bluetooth** διατάξεις έχουν προκαθορισμένους κωδικούς. Ενώ σε άλλες διατάξεις μπορούμε να επιλέξουμε εμείς τον κωδικό για την ζεύξη. Παρακάτω ακολουθεί η εικόνα 4.3 οπού φαίνεται μια επιτυχής σύνδεση μεταξύ των δυο διατάξεων, στην επιφάνεια εργασίας του προγράμματος **BlueSoleil**.



Εικόνα 4.3: Σύνδεση μεταξύ δυο διατάξεων Bluetooth με την χρήση του προγράμματος BlueSoleil.

Το τελευταίο στάδιο της επικοινωνίας του ρομποτικού συστήματος μέσω των **Bluetooth** διατάξεων, είναι η σύνδεση τους με το πρόγραμμα επεξεργασίας δεδομένων, **Matlab**. Για να ολοκληρωθεί η δημιουργία της εικονικής σειριακής σύνδεσης , θα πρέπει να γίνει η κατάλληλη παραμετροποίηση στο πρόγραμμα **Matlab** καθώς και να εκτελεστούν κάποιες εντολές. Στην εικόνα 4.4 φαίνεται το περιβάλλον εργασίας του Matlab καθώς και οι εντολές που χρησιμοποιούμε προκειμένου να θεμελιώσουμε την σύνδεση μεταξύ του αναπτυξιακού, και του προγράμματος.



Εικόνα 4.4: Περιβάλλον εργασίας Matlab. Αρχικοποίηση παραμέτρων για σύνδεση Bluetooth.

Πριν αναλύσουμε τον αλγόριθμο στο περιβάλλον εργασίας του Matlab, θα πρέπει να σημειωθεί ότι, είναι αναγκαίο να εγκαταστήσουμε στις βιβλιοθήκες του προγράμματος, την βιβλιοθήκη **Instrument Toolbox**. Η συγκεκριμένη βιβλιοθήκη έχει όλες τις απαραίτητες ρυθμίσεις σχετικά με την ασύρματη επικοινωνία του προγράμματος, με άλλες συσκευές η προγράμματα.

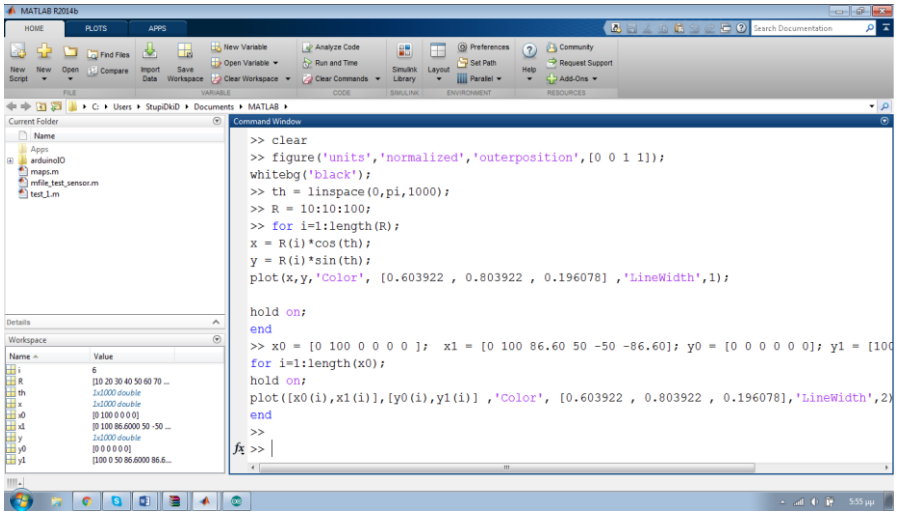
Αφού λοιπόν έχουμε προγραμματίσει το **Bluetooth** του αναπτυξιακού, και καταφέραμε να συνδεθούμε μεσώ του προγράμματος **BlueSoleil** με το **Bluetooth** του υπολογιστή φτιάχνουμε τον αλγόριθμο σύνδεσης στο πρόγραμμα **Matlab**. Ξεκινώντας χρησιμοποιούμε την εντολή **bt-serial('com20')** οπού ρυθμίζουμε την θύρα που θα χρησιμοποιήσουμε για την εικονική σειριακή επικοινωνία. Η θύρα που θα επιλέξουμε θα πρέπει να είναι η ίδια και στην πλατφόρμα του **Arduino** αλλιώς η σύνδεση μας θα είναι ανεπιτυχής. Επίσης με την παραπάνω εντολή έχουμε αποθηκεύσει το στοιχείο μας στην μεταβλητή **bt** (από τα αρχικά Blue Tooth). Στην συνέχεια ρυθμίζουμε την ταχύτητα επικοινωνίας και προσέχουμε να είναι η τιμή της, ίδια με του αλγορίθμου της διάταξης του Bluetooth, οπού έγινε στο πρόγραμμα **Arduino**. Η εντολή που χρησιμοποιούμε είναι η **bt.baudrate=9600**.

Για να ενεργοποιήσουμε την συσκευή του **Bluetooth** μέσα στο πρόγραμμα**,** χρησιμοποιούμε την εντολή **fopen(bt)**, οπού περιμένουμε κάποιες πληροφορίες σχετικά με την κατάσταση της επικοινωνίας, όπως φαίνεται στην εικόνα 4.4. Τέλος έχουμε μια επιτυχή σύνδεση και μπορούμε να ζητήσουμε δεδομένα από το τετράτροχο ρομποτικό σύστημα και το αναπτυξιακό. Εφόσον έχουμε προγραμματίσει το αναπτυξιακό να δέχεται κάποιες λέξεις η χαρακτήρες κλειδιά ώστε να μας στείλει τα απαιτούμενα δεδομένα, ο προγραμματισμός στο περιβάλλον **Matlab** είναι ο εξής: **fwrite(bt,'1'), fscanf(bt).** Με την πρώτη εντολή ζητάμε από το αναπτυξιακό να στείλει τα δεδομένα που έχει. Στην συγκεκριμένη περίπτωση ο χαρακτήρας που περιμένει το Arduino για να απαντήσει είναι το **1**. Ενώ με την **fscanf(bt)** το **Matlab** δέχεται τα δεδομένα από το σύστημα. Ακόμα μπορούμε να αποθηκεύσουμε τα δεδομένα, ώστε να γίνει η απαραίτητη επεξεργασία με την εντολή **data=fscanf(bt)**.

4.5 Προγραμματισμός σε περιβάλλον Matlab με σκοπό την επεξεργασία των δεδομένων

Έχοντας ρυθμίσει την ασύρματη σύνδεση μεταξύ του τετράτροχου ρομποτικού συστήματος και του προγράμματος Matlab , μπορούμε να ξεκινήσουμε την επεξεργασία των δεδομένων. Σε μια πιο γενικευμένη προσεγγίσει θα μπορούσαμε να πούμε ότι οι τρόποι επεξεργασίας των δεδομένων ποικίλουν, ανάλογα με την εφαρμογή. Από το σύστημα μπορούμε να πάρουμε τιμές από όλους τους αισθητήρες καθώς και τους κινητήρες και να τους επεξεργαστούμε ανάλογα. Στην συγκεκριμένη περίπτωση θα χρησιμοποιήσουμε τα δεδομένα από τον αισθητήρα υπέρηχων και θα φτιάξουμε ένα ραντάρ στο πρόγραμμα Matlab. Το ραντάρ (radar) θα μας δείχνει την εκάστοτε θέση του αντικειμένου/εμποδίου, που συναντάει το όχημα σε ένα γράφημα με βαθμονομημένους άξονες.

Ξεκινώντας στο πρώτο κομμάτι του προγράμματος θα δημιουργήσουμε το γράφημα σε μορφή ραντάρ. Στην εικόνα 4.5 απεικονίζεται το πρώτο μέρος του αλγορίθμου οπού δημιουργούμε το γράφημα για την αποτύπωση στίγματος, όταν το όχημα μας εντοπίσει κάποιο εμπόδιο.

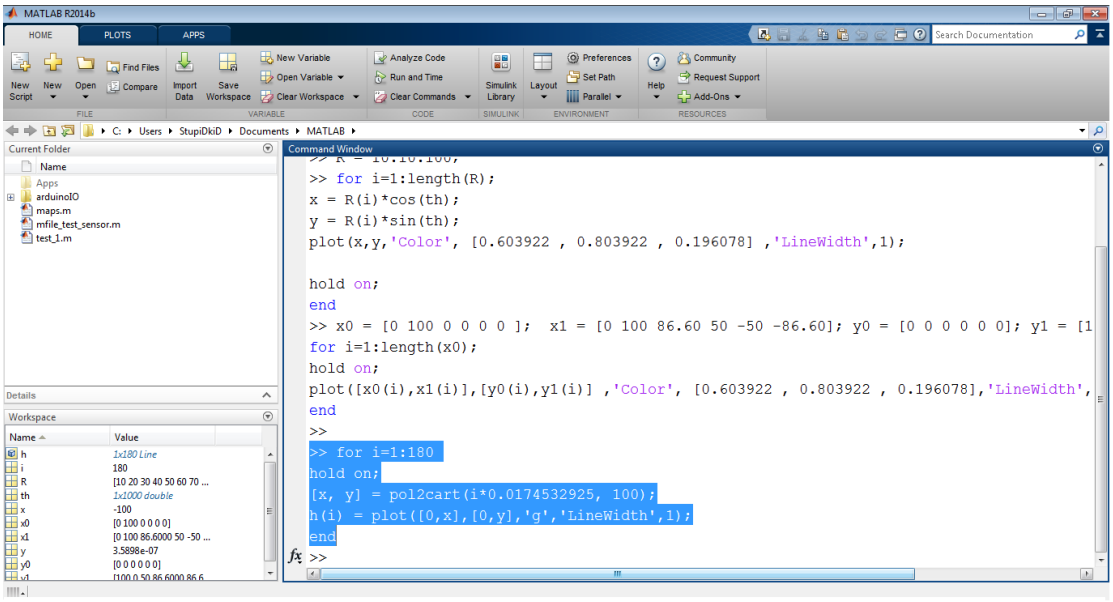


Εικόνα 4.5: Αλγόριθμος δημιουργίας γραφήματος ραντάρ (Radar)**.**

Στο πρώτο μέρος, χρησιμοποιούμε την εντολή **figure('units','normalized', 'outerposition',(0 0 1 1])** καθώς και την εντολή **whitebg('black')**, για να προσαρμόσουμε το γράφημα ανάλογα με τις επιλογές μας (χρώματα φόντο ,στοιχεία κλπ.). Για να δημιουργήσουμε τα δεδομένα της κλίμακας χρησιμοποιούμε τις εντολές **th=linspace(0,pi,1000)**, **R= 10:10:100** (βήμα) και χρησιμοποιώντας μια δομή επανάληψης **for i=1:length[R]** καταφέρνουμε να σχεδιάσουμε στο γράφημα το εύρος των τιμών τις κλίμακας που χρειαζόμαστε.

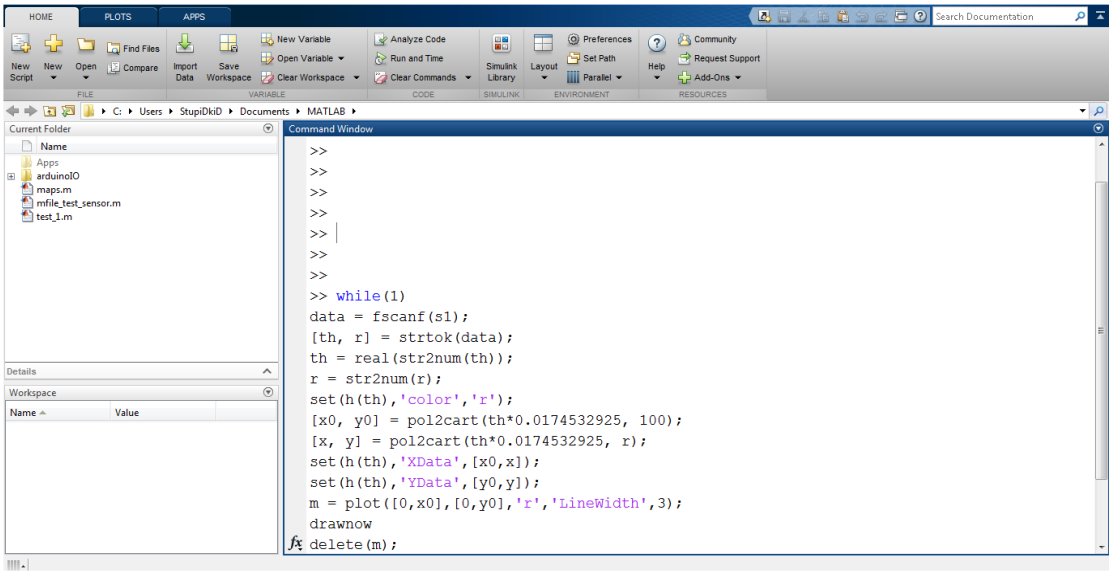
Μέσα στην εντολή της επανάληψης δημιουργούμε τους άξονες x και y με τις εντολές **x= R(i)*cos(**th) και y=R(i)*sin(th). Τέλος με την εντολή plot(x,y) διαμορφώνουμε το κομμάτι του γραφήματος όπως επιθυμούμε και με την εντολή hold on το κρατάμε σε αναμονή.

Στην συνέχεια θα σχεδιάσουμε τα δεδομένα των αξόνων. Θέτοντας τιμές στην μεταβλητή **x0** και δημιουργώντας μια δομή επανάληψης με την εντολή **for i=1:length(x0).** Χρησιμοποιούμε πάλι την εντολή **plot**, μέσα στην εντολή **for** ώστε να διαμορφώσουμε το γράφημα ανάλογα με την επιθυμία μας(για παράδειγμα το χρώμα). Το τελευταίο κομμάτι για την δημιουργία του σόναρ απεικονίζεται στην εικόνα 4.6 οπού δημιουργούμε την περιοχή σάρωσης του συστήματος μας ώστε αυτό να αποτυπώνετε στο γράφημα. Έτσι με τις εντολές **for i=1:180**, **[x,y]=pol2cart** και **h[i] = plot** δημιουργούμε σε μορφή γραφήματος ένα περιβάλλον ίδιο με αυτό ενός σόναρ.



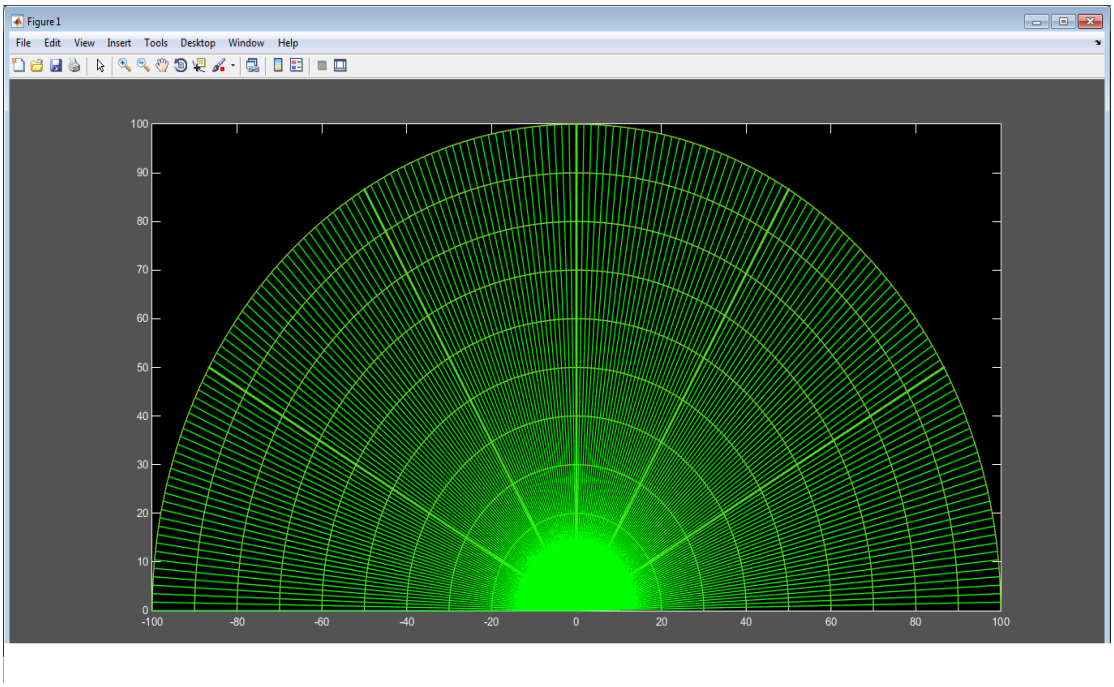
Εικόνα 4.6: Αλγόριθμος δημιουργίας γραφήματος ραντάρ/sonar.

Στο τελευταίο κομμάτι του προγραμματισμού στο περιβάλλον του **Matlab**, θα εστιάσουμε στον αλγόριθμο οπού επεξεργάζεται τα δεδομένα που δέχεται από το σύστημα. Έτσι καταφέρνουμε να δημιουργήσουμε ένα στίγμα στο γράφημα του ραντάρ, κάθε φορά που το ρομποτικό σύστημα εντοπίζει κάποιο εμπόδιο/αντικείμενο. Στην εικόνα 4.7 φαίνεται ο σχετικός αλγόριθμος.



Εικόνα 4.7: Αλγόριθμος δημιουργίας στίγματος, πάνω στο γράφημα του ραντάρ.

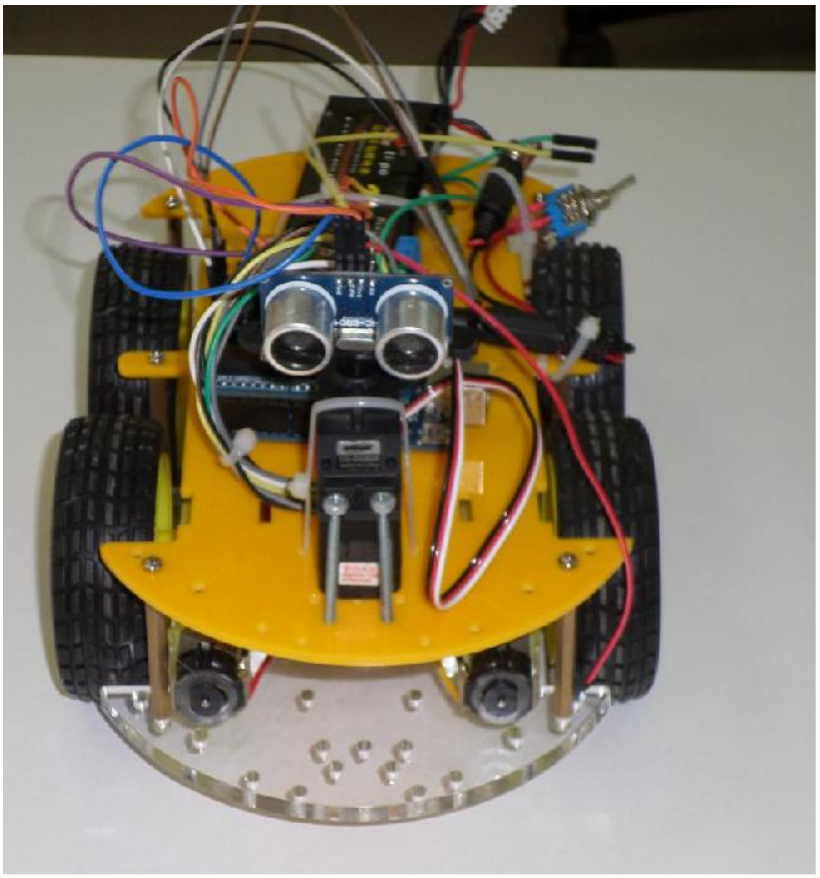
Αφού έχουμε συνδέσει το πρόγραμμα **Matlab** με το σύστημα, όπως έχουμε αναφέρει στην ενότητα 4.4, μπορούμε να δεχτούμε στο πρόγραμμα επεξεργασίας τα απαραίτητα δεδομένα. Η δομή στην οποία θα γραφτεί το πρόγραμμα θα είναι επανάληψης ξεκινώντας με την εντολή **while**. Στην συνέχεια και όπως έχουμε αναφέρει παραπάνω τα δεδομένα που θα δεχόμαστε θα αποθηκεύονται στην μεταβλητή **data** με την εντολή **data=scanf(s1)**. Με τις εντολές που ακολουθούν καταφέρνουμε να επεξεργαστούμε τις εκάστοτε αποστάσεις των αντικειμένων και να τις μετατρέψουμε στην κλίμακα του γραφήματος. Τέλος εμφανίζουμε ένα γράφημα το οποίο έχει μορφή ραντάρ και παίρνει δεδομένα από το ρομποτικό σύστημα σε πραγματικό χρόνο υπολογίζοντας το στίγμα, Στην εικόνα 4.8 φαίνεται το σχετικό διάγραμμα.



Εικόνα 4.8: Γράφημα σε μορφή ραντάρ στο περιβάλλον Matlab.

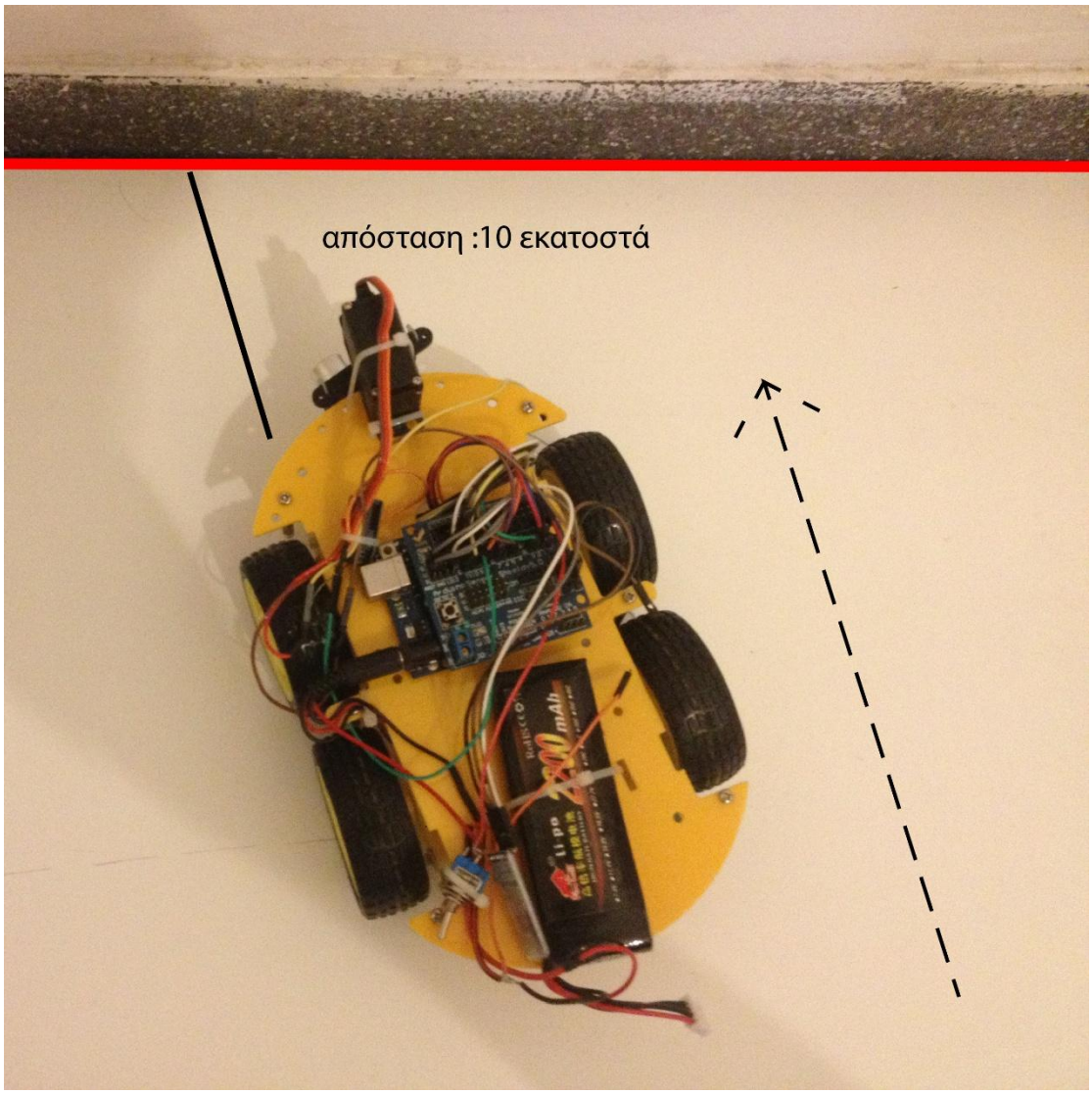
4.6 Πειραματική διαδικασία και έλεγχος ορθής λειτουργίας ρομποτικού συστήματος

Σε αυτήν την ενότητα αναφέρονται τα πειράματα που πραγματοποιήθηκαν προκειμένου να ελεγχθεί η ορθή λειτουργία του συστήματος. Έτσι αφού συναρμολογήθηκε το ρομποτικό σύστημα με τα επιμέρους τμήματα του καθώς και ¨φορτώθηκαν¨ όλοι οι αλγόριθμοι και τα προγράμματα στον μικροελεγκτή, δημιουργήθηκε ένας χώρος οπού το τετράτροχο σύστημα θα μπορούσε να μετακινηθεί, και να αποφύγει τυχόν εμπόδια και αντικείμενα. Παράλληλα το σύστημα είναι συνδεδεμένο ασύρματα με κάποιο υπολογιστικό σύστημα στέλνοντας τα δεδομένα που συλλέγει για επεξεργασία. Στην εικόνα 4.9 απεικονίζεται το τετράτροχο ρομποτικό σύστημα.



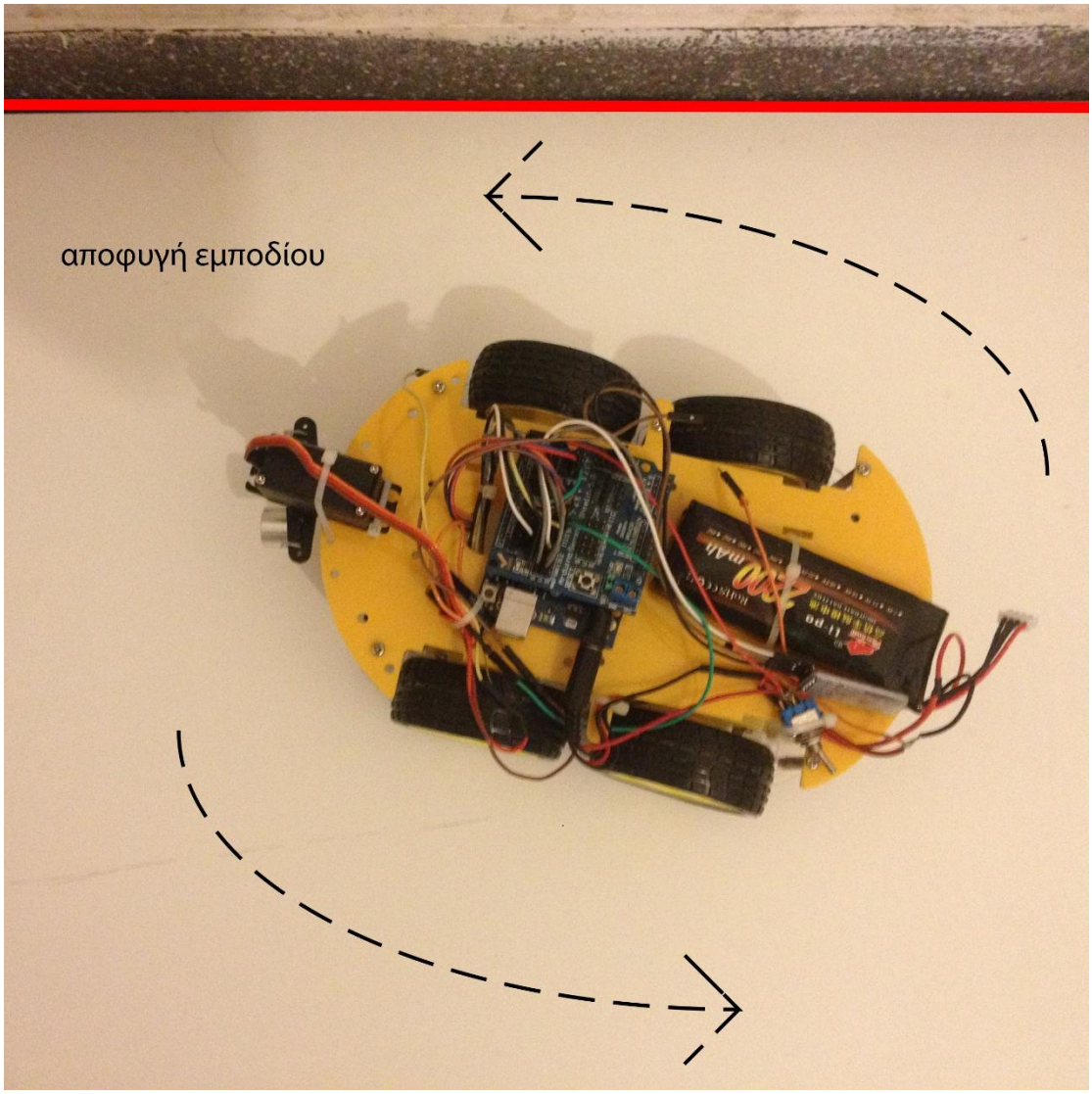
Εικόνα 4.9: Τετράτροχο ρομποτικό σύστημα αυτόματης πλοήγησης

Ξεκινώντας, στην εικόνα 4.10 απεικονίζεται η ευθεία κίνηση του τετράτροχου προς μια κατεύθυνση, προσεγγίζοντας ένα εμπόδιο.



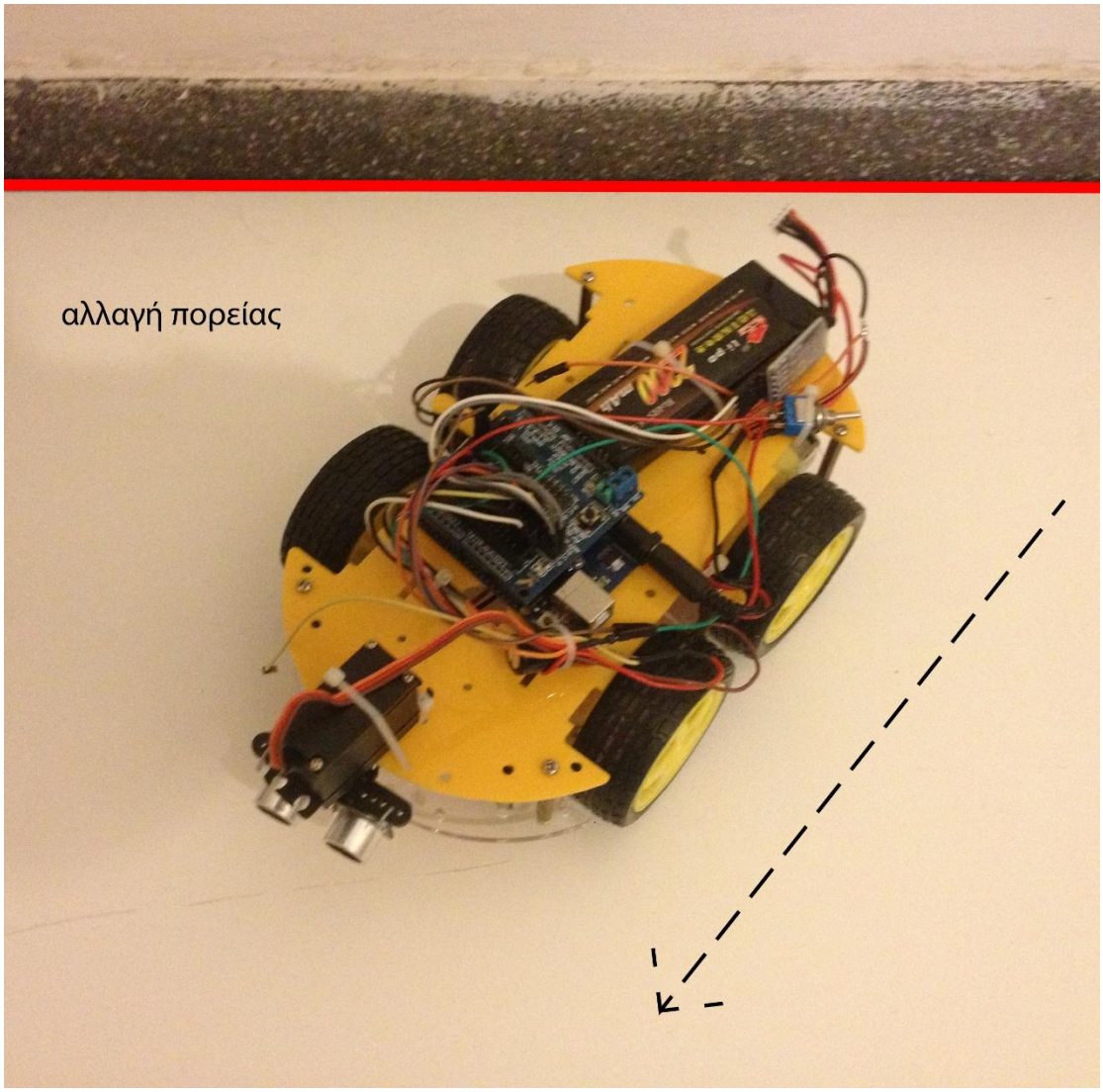
Εικόνα 4.10: Στιγμιότυπα κίνησης και αποφυγής εμποδίων μέσα σε έναν περιβάλλοντα χώρο του τετράτροχου ρομποτικού συστήματος

Στην συνέχεια αφού το ρομποτικό σύστημα πλησιάσει αρκετά το εμπόδιο/αντικείμενο, προκειμένου να αποφύγει, ο μικροελεγκτής δίνει εντολή στους κινητήρες των τροχών να κινηθούν αντίθετα. Η λογική αυτή βασίστηκε στον τρόπο λειτουργίας της ερπύστριας και αυτό γιατί καθιστά την αναστροφή πολύ πιο εύκολη όταν υπάρχει κάποιο εμπόδιο. Στην εικόνα 4.11 απεικονίζεται η κίνηση αναστροφής που κάνει το ρομποτικό σύστημα, αποφεύγοντας το εμπόδιο.



Εικόνα 4.11: Στιγμιότυπα κίνησης και αποφυγής εμποδίων μέσα σε έναν περιβάλλοντα χώρο του τετράτροχου ρομποτικού συστήματος

Τέλος το ρομποτικό σύστημα αφού αλλάξει πορεία, απομακρύνεται από το εμπόδιο/αντικείμενο με ασφάλεια. Στην εικόνα 4.12 απεικονίζεται η στιγμή της απομάκρυνσης του συστήματος.



Εικόνα 4.12: Στιγμιότυπα κίνησης και αποφυγής εμποδίων μέσα σε έναν περιβάλλοντα χώρο του τετράτροχου ρομποτικού συστήματος.

ΚΕΦΑΛΑΙΟ 5^{Ο}

Δυνατότητες επέκτασης-αναβάθμισης του τετράτροχου ρομποτικού συστήματος και συμπεράσματα

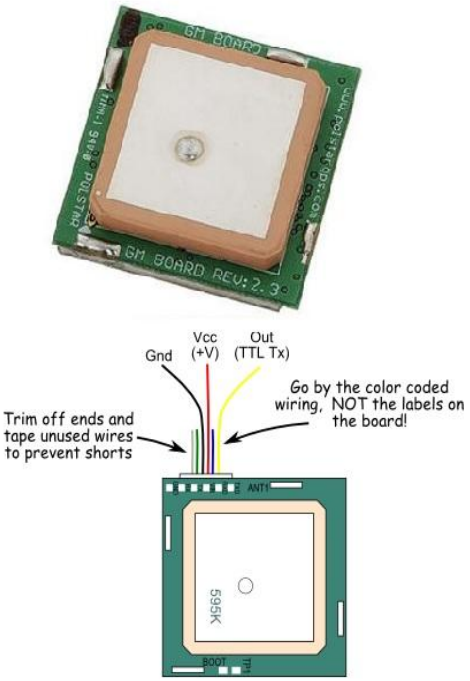
5.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα καταγράψουμε κάποια συμπεράσματα σχετικά με την κατασκευή του τετράτροχου ρομποτικού συστήματος, καθώς και κάποιες μελλοντικές πτυχές ανάπτυξης και υλοποίησης. Στοχεύοντας στην βασική αρχή της σχεδίασης του συστήματος, η οποία είναι η μη επανδρωμένη κίνηση/αυτόματη πλοήγηση και η αποστολή και επεξεργασία των δεδομένων κάναμε μια έρευνα σχετικά με την προσθήκη περισσοτέρων αισθητήρων. Εφόσον ένας από τους στόχους της εργασίας ήταν να αποτυπώσουμε τον περιβάλλοντα χώρο σε μορφή ραντάρ με την χρήση αισθητήρων, αποφασίσαμε πως ο εντοπισμός (localization) με την βοήθεια μιας **GPS** διάταξης θα είναι μια κρίσιμη αναβάθμιση για το ρομποτικό σύστημα.

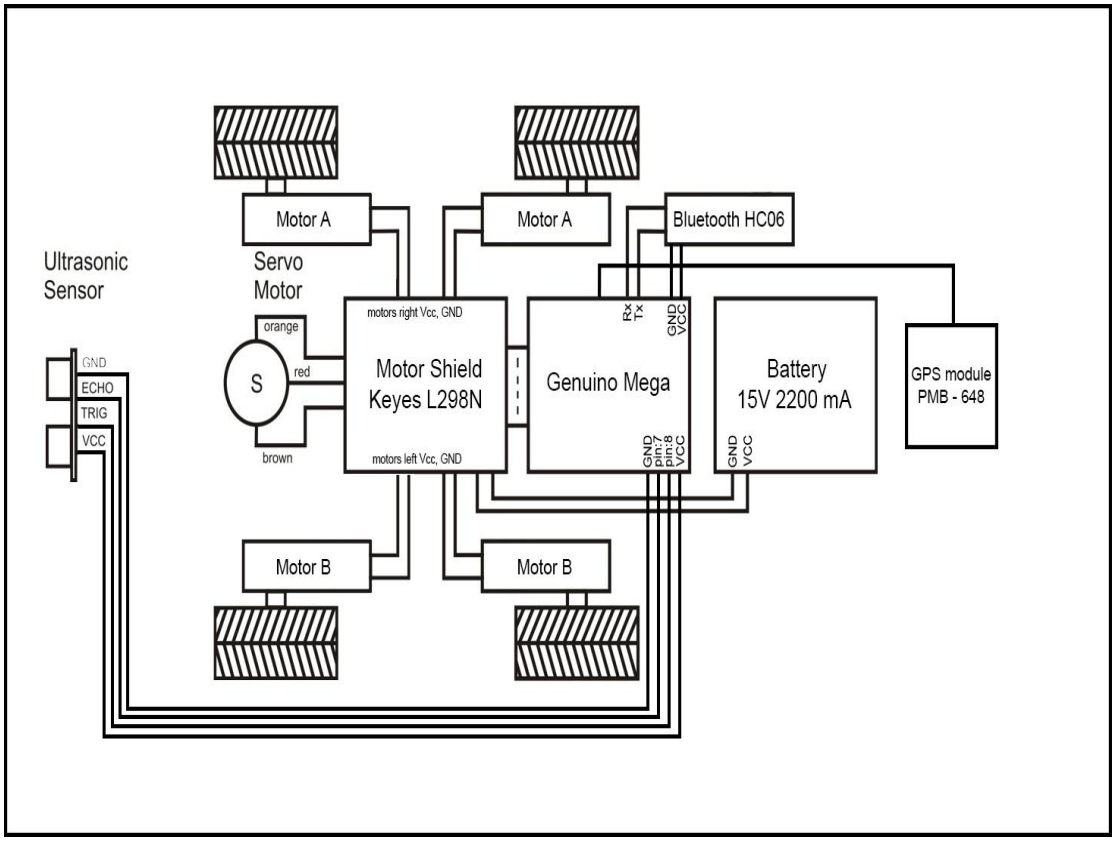
5.2 GPS Διάταξη τύπου PMB-648

Το **GPS** (Global Positioning Satellite) module, με όνομα **PMB-648 GPS,** είναι μια διάταξη μέσω της οποίας μπορούμε να εντοπίσουμε την θέση του τετράτροχου ρομποτικού συστήματος καθώς και να πάρουμε τα απαραίτητα δεδομένα για την επεξεργασία στο πρόγραμμα **Matlab**. Γενικά μια GPS διάταξη μπορεί να μας δώσει μια ποικιλία δεδομένων όπως το γεωγραφικό πλάτος, το γεωγραφικό μήκος, την ώρα, την ταχύτητα, ακόμα και το υψόμετρο. Πιο συγκεκριμένα ο αριθμός των δεδομένων που μπορεί ένα GPS να αποστείλει είναι ανάλογο των δορυφόρων που είναι συνδεδεμένη η διάταξη. Για να μπορέσουμε να πάρουμε το γεωγραφικό πλάτος, μήκος αλλά και ύψος, για να δημιουργήσουμε το στίγμα του ρομποτικού συστήματος οπού φέρει την διάταξη πρέπει το GPS είναι συνδεδεμένο με τουλάχιστον 5 δορυφόρους.

Το **PMB** διαθέτει 20 κανάλια παράλληλου δορυφορικού εντοπισμού, για την γρηγορότερη απόκτηση δεδομένων. Είναι ιδανικός για εφαρμογές στην ρομποτική και τα αυτόνομα συστήματα πλοήγησης, την τηλεμετρία κ.α. Διαθέτει μια ενσωματωμένη κεραία, καθώς και μια επαναφορτιζόμενη μπαταριά για την μνήμη της διάταξης. Το πρωτόκολλο επικοινωνίας που χρησιμοποιεί είναι το **TTL.** Η τάση λειτουργίας του είναι τα 3,3-5**V** και η κατανάλωση του ρεύματος ανέρχεται στα 65 **mA,** σύμφωνα με τον κατασκευαστή. Γενικά το **PMB** χαρακτηρίζεται από την μικρή κατανάλωση ισχύος. Η τροφοδοσία του γίνεται μέσω του αναπτυξιακού **Arduino Uno** οπού συνδέεται. Παρακάτω ακολουθεί η εικόνα 5.1 οπού φαίνεται η διάταξη PMB-648 καθώς και τα ψηφιακά ¨ποδαράκια¨ (**PIN**). Ενώ στην εικόνα 5.2 φαίνεται το σχηματικό του ρομποτικού συστήματος μαζί με το **GPS** module.



Εικόνα 5.1: GPS Module PMB-648 *[20]*.



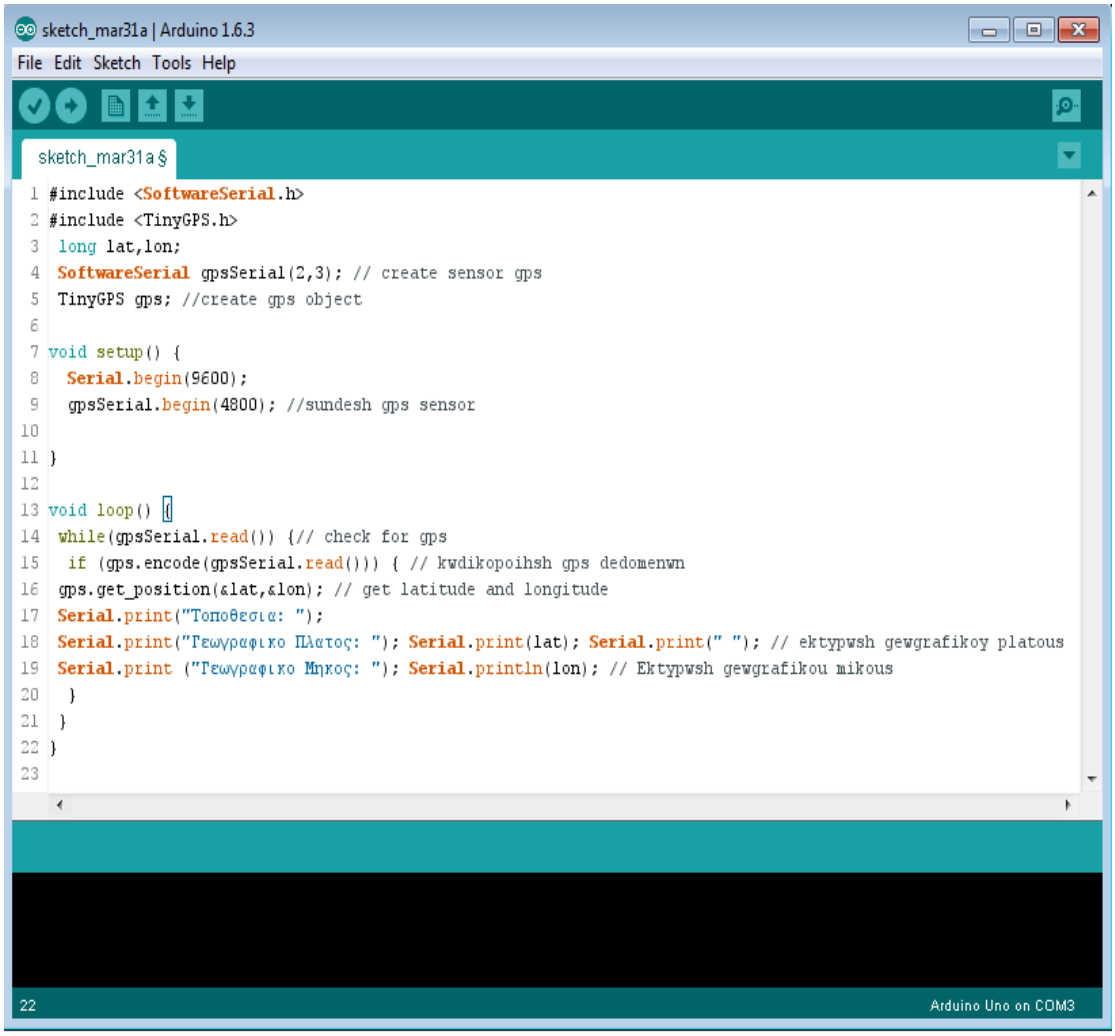
Εικόνα 5.2: Σχηματικό διάγραμμα ρομποτικού συστήματος με GPS διάταξη.

5.3 Προγραμματισμός GPS διάταξης και επεξεργασίας δεδομένων

Στην συγκεκριμένη ενότητα θα αναλύσουμε τον απαιτούμενο προγραμματισμό της **GPS** διάταξης καθώς και των δεδομένων που θα επεξεργαστούμε. Θα χωρίσουμε την ενότητα σε 2 μέρη. Στο πρώτο μέρος θα αναλύσουμε τον προγραμματισμό της διάταξης στο περιβάλλον **Arduino**, ώστε να συγχρονίσουμε/συνδέσουμε το **GPS module** με το αναπτυξιακό **Arduino Uno.** Στο δεύτερο μέρος θα αναλυθεί ο αλγόριθμος επεξεργασίας των δεδομένων, που δεχόμαστε από την **GPS** διάταξη, σε περιβάλλον **Matlab.**

Πριν ξεκινήσουμε, θα πρέπει να αναφερθεί ότι είναι αναγκαίο να έχουμε στις βιβλιοθήκες του προγράμματος Arduino, την βιβλιοθήκη **TinyGPS** ώστε να μπορέσουμε να την χρησιμοποιήσουμε στον κώδικα μας. Αφού βρούμε την συγκεκριμένη βιβλιοθήκη στην σελίδα του Arduino, την αντιγράφουμε στο φάκελο με τις βιβλιοθήκες. Ξεκινώντας, πρέπει να συμπεριλάβουμε τις βιβλιοθήκες που χρησιμοποιούμε στον κώδικα και αυτό γίνεται με την εντολή **#include.** Στην συνέχεια ορίζουμε τις μεταβλητές μας (γεωγραφικό πλάτος και μήκος) καθώς και ενεργοποιούμε τον αισθητήρα ορίζοντας τα ψηφιακά ποδαράκια οπού είναι συνδεδεμένος πάνω στο αναπτυξιακό. Αφού ενεργοποιήσουμε την σειριακή επικοινωνία ώστε να μπορούμε να στέλνουμε και να δεχόμαστε δεδομένα, με την εντολή **Serial.begin**, θέτουμε την ταχύτητα της ασύρματης σύνδεσης, της GPS διάταξης με την εντολή **gpsSerial.begin(4800).** Προγραμματίζοντας μέσα στην δομή επανάληψης **void loop()**, θα χρησιμοποιήσουμε την εντολή **while (gpsSerial.read())** ώστε να ελέγξουμε εάν έχουμε δεδομένα στην διάταξη του **GPS,** για όσο χρόνο η διάταξη μας δίνει δεδομένα.

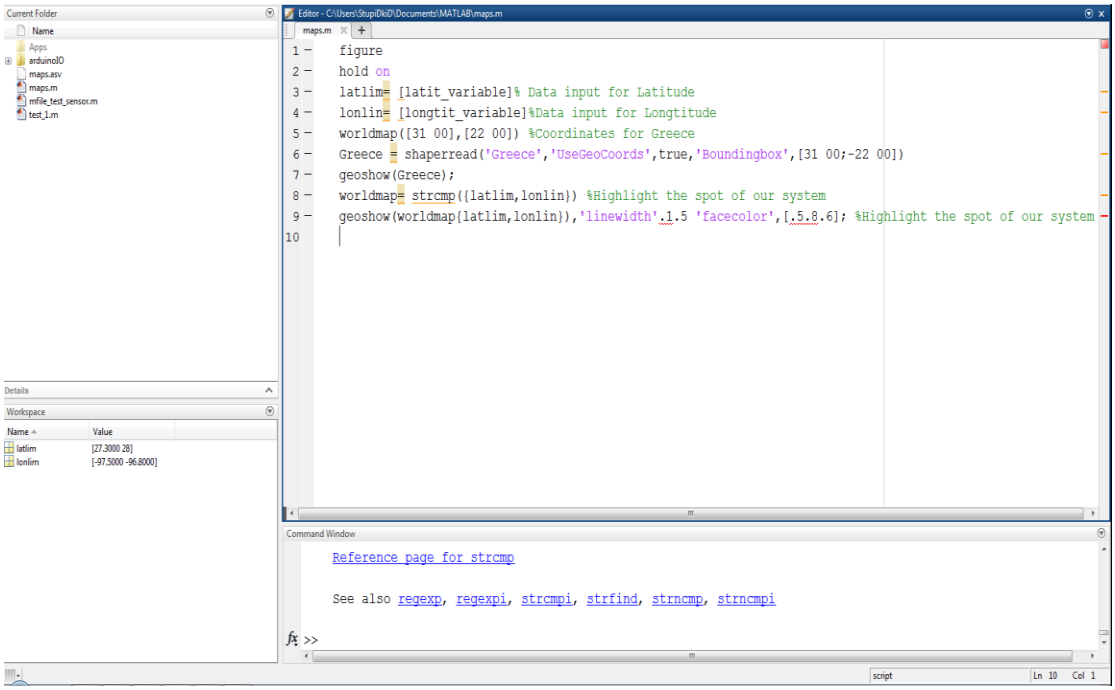
Αφού δεχθούμε τα δεδομένα θα πρέπει να τα κωδικοποιήσουμε ώστε να είναι κατανοητά σε εμάς. Η εντολή που κωδικοποιεί τα δεδομένα που δέχεται το GPS από τους δορυφόρους είναι η **gps.encode.** Στην συνέχεια παίρνουμε τα δεδομένα που χρειαζόμαστε, (στην συγκεκριμένη περίπτωση γεωγραφικό πλάτος και μήκος) και τα εκτυπώνουμε στην σειριακή οθόνη. Η εντολή με την οποία παίρνουμε τα δεδομένα είναι η **gps.get_position**. Ενώ με την εντολή **Serial.Print** εκτυπώνουμε τα δεδομένα. Στην περίπτωση μας το πρόγραμμα περιμένει να πάρει μια εντολή ώστε να στείλει τα δεδομένα στο πρόγραμμα **Matlab** ακριβώς όπως αναλύθηκε στο κεφάλαιο 4. Παρακάτω ακολουθεί η εικόνα 5.3 οπού φαίνεται ο σχετικός αλγόριθμος σε προγραμματιστικό περιβάλλον **Arduino**.



Εικόνα 5.3: Αλγόριθμος οδήγησης GPS διάταξης .

Στο δεύτερο μέρος της ενότητας θα αναλύσουμε την επεξεργασία των δεδομένων που δεχόμαστε από την GPS διάταξη, στο προγραμματιστικό περιβάλλον **Matlab**. Ποιο συγκεκριμένα επεξεργαζόμαστε το γεωγραφικό πλάτος και μήκος και δημιουργούμε ένα γράφημα γεωγραφικού χάρτη οπού θα απεικονίζεται το στίγμα του συστήματος.

Πριν ξεκινήσουμε θα πρέπει να αναφερθεί ότι είναι χρειάζεται να έχουμε εγκατεστημένη στο πρόγραμμα **Matlab** την βιβλιοθήκη ¨Εργαλεία Χαρτογράφησης¨ (**Mapping Toolbox**), ώστε να μπορέσουμε να δημιουργήσουμε χάρτες βασιζομένους σε πραγματικές συντεταγμένες. Αρχικά θα αναφέρουμε ότι αλγόριθμος είναι γραμμένος σε μορφή ρουτίνας (Script) και μπορεί να εκτελείται ατέρμονα. Έτσι είναι ικανός να ζητάει δεδομένα καθώς και να δέχεται συνέχεια, ανανεώνοντας την εικόνα του γραφήματος, με αποτέλεσμα την ¨παρακολούθηση¨ η τον εντοπισμό (localization) του συστήματος μας σε πραγματικό χρόνο. Στην εικόνα 5.4 απεικονίζεται ο αλγόριθμος ανάλυσης των δεδομένων σε προγραμματιστικό περιβάλλον **Matlab.**



Εικόνα 5.4: Αλγόριθμος επεξεργασίας γεωγραφικού χάρτη και στίγματος σε περιβάλλον Matlab.

Ξεκινώντας θα ονομάσουμε την ρουτίνα **maps** και θα την αποθηκεύσουμε στον υπολογιστή. Αφού έχει γίνει η ασύρματη σύνδεση με το σύστημα μας, μέσω Matlab μπορούμε να ζητήσουμε τα απαραίτητα δεδομένα. Ο σχετικός αλγόριθμος της σύνδεσης αναφέρεται στο κεφάλαιο 4. Εφόσον έχουμε δεχτεί τα δεδομένα, δημιουργούμε μια εικόνα με την εντολή **figure** και με την εντολή **hold on** κρατάμε την εικόνα σε αναμονή, ώστε να μπορέσουμε να την επεξεργαστούμε. Στην συνέχεια αποθηκεύουμε τα δεδομένα (γεωγραφικό πλάτος και μήκος) στις μεταβλητές **latlim** και **lonlim** οπού ουσιαστικά είναι τα όρια μας στο χάρτη. Εμφανίζουμε τον χάρτη της Ελλάδας με την βοήθεια της εντολής **worldmap** τοποθετώντας σαν μεταβλητές εισόδου τις συντεταγμένες της Ελλάδας. Με την εντολή **shaperread** οριοθετούμε τα σύνορα της Ελλάδας στο γράφημα και με την εντολή **geoshow** εμφανίζουμε στον χάρτη το αποτέλεσμα.

Τέλος για να εμφανίσουμε την περιοχή του στίγματος χρησιμοποιούμε την εντολή **worldmap=strcmp** εισχωρώντας της μεταβλητές των δεδομένων που έχουμε για το γεωγραφικό πλάτος και μήκος του συστήματος. Για να επεξεργαστούμε την εικόνα ώστε να δώσουμε το στίγμα, θα χρησιμοποιήσουμε την εντολή **geoshow** θέτοντας τις παραμέτρους για τον χρωματισμό της περιοχής του στίγματος και του χάρτη.

5.4 Συμπεράσματα σχετικά με της λειτουργείες του συστήματος

Συμπερασματικά, ο στόχος της εργασίας ήταν η δημιουργία ενός μη επανδρωμένου ρομποτικού συστήματος, αυτόματης κίνησης, σε απομακρυσμένο/εχθρικό περιβάλλον, εκεί που δεν θα μπορεί να φτάσει ο άνθρωπος. Πιο συγκεκριμένα το σύστημα θα είναι ικανό να κινείται σε τέτοιο περιβάλλον συλλέγοντας πληροφορίες και δεδομένα τα οποία θα μπορεί να αποστέλλει σε έναν υπολογιστή για την απαραίτητη επεξεργασία. Δυο από τις κύριες λειτουργίες του όπως έχει αναφερθεί, σχετικά με τα δεδομένα που μπορεί να αποστείλει είναι η χαρτογράφηση (mapping) αλλά και ο εντοπισμός(localization). Αυτό το καθιστά ικανό να επικοινωνεί με κάποιον χρήστη και να αλληλοεπιδρά με αυτόν σε πραγματικό χρόνο. Ένα τέτοιο σύστημα κάτω από κάποιες κατασκευαστικές προϋποθέσεις θα μπορεί να χρησιμοποιηθεί σε ανεύρεση επιζώντων σε κάποιο καταστροφικό σενάριο, για στρατιωτικούς αλλά και ερευνητικούς σκοπούς.

ΚΕΦΑΛΑΙΟ 6^{Ο}

Συμπεράσματα πτυχιακής εργασίας

Όπως αναφέρθηκε στο τελευταίο μέρος του 5^{ου} κεφαλαίου, ένας από τους στόχους της πτυχιακής εργασίας ήταν η δημιουργία ενός τετράτροχου ρομποτικού συστήματος, διαφορικής οδήγησης. Το σύστημα θα είναι ικανό να κινείται αυτόματα και θα έχει την δυνατότητα να συλλέγει και να επεξεργάζεται πληροφορίες/δεδομένα από το περιβάλλον καθώς και να επικοινωνεί με τον χρήστη ασύρματα. Βασίζεται στον μικροελεγκτή **Arduino** ο οποίος είναι υπεύθυνος για τον συγχρονισμό των αισθητήρων καθώς και των λειτουργιών του συστήματος. Προσπαθήσαμε να προσεγγίσουμε το πρόβλημα της ρομποτικής πλοήγησης, αναλύοντας αλγορίθμους οδήγησης του συστήματος. Επίσης αναπτύξαμε αλγορίθμους χαρτογράφησης (mapping) καθώς και εντοπισμού (localization) οπού το σύστημα βρίσκεται σε ένα τυχαίο περιβάλλον και με την βοήθεια αισθητήριων είναι ικανό να το εξερευνήσει και να το αποτυπώσει. Εργαστήκαμε σε περιβάλλον δυο γλωσσών προγραμματισμού, το Arduino αλλά και το Matlab. Το πρώτο είναι υπεύθυνο για τις λειτουργίες του συστήματος, ενώ στο δεύτερο έγινε η επεξεργασία των δεδομένων.

Ο χρόνος εκπόνησης της εργασίας και η ενασχόληση μας με το ρομποτικό σύστημα, αποτέλεσε πηγή γνώσης και μάθησης, τόσο σε θεωρητικό επίπεδο των τεχνικών πλοήγησης αλλά και της επεξεργασίας δεδομένων όσο και στο τεχνικό υπόβαθρο της ανάπτυξης της συγκεκριμένης μελέτης και υλοποίησης. Ακόμα καταφέραμε να εμβαθύνουμε σε γνώσεις σχετικές με την επιστήμη του Ηλεκτρονικού μηχανικού και πιο συγκεκριμένα με ρομποτικά συστήματα και συστήματα αυτόματου ελέγχου. Επίσης η ασχολία μας με πλατφόρμες προγραμματισμού μας βοήθησαν να κατανοήσουμε την λειτουργία των αλγορίθμων, των δομών προγραμματισμού, αλλά και των γλωσσών προγραμματισμού. Από την επαφή μας με τον εξοπλισμό και τις εργασίες/έρευνες σχετικά με το ρομποτικό σύστημα ο τρόπος αντίληψης σχετικά με το αντικείμενο των σπουδών και την εξέλιξη αυτού του επαγγέλματος τουλάχιστον σε προσωπικό επίπεδο διαμορφώθηκε προς το καλύτερο. Τέλος από την επαφή μας με τα ρομποτικά συστήματα και την πιο γενική έρευνα που έγινε γύρω από αυτά αποκομίσαμε χρήσιμες γνώσεις και εμπειρίες σχετικά με την επιθυμητή επαγγελματική μας σταδιοδρομία.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1]: *Ρομποτικό σύστημα σταθερής βάσης.*

http://robotecmed.blogspot.gr

[2]: *Τετρακόπτερο ρομποτικό σύστημα (Quadra copter).*

http://gaspuglia.altervista.org/wp-content/uploads/2013/05/2-4GHz-4CH-RC-Mini-Quadcopter.png

[3]: *Παράδειγμα AUV ρομποτικού συστήματος.*

https://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/D5682F98CBFBC05AC1257497002976E4?OpenDocument

[4]: *Έντροχο ρομποτικό σύστημα ανίχνευσης αντικειμένων.*

http://www.linuxinsider.gr/forum/9250/arduino

[5]: *Λειτουργία πομπού δέκτη.*

http://www.linuxinsider.gr/forum/9250/arduino

[6]:*Κύκλωμα σύνδεσης LED και πομποδέκτη υπέρυθρων.*

http://www.linuxinsider.gr/forum/9250/arduino

[7]: *Δίτροχο αυτόνομο σύστημα αυτό-ισορρόπησης.*

Αρχείο πτυχιακής εργασίας Οδυσσέα Μαυροματάκη.

[8]: Δίτροχο αυτόνομο σύστημα αυτό-ισορρόπησης.

Αρχείο πτυχιακής εργασίας Οδυσσέα Μαυροματάκη.

[9]: Σχηματικό ακροδεκτών ολοκληρωμένου ATmega328P.

https://www.arduino.cc/en/Hacking/PinMapping168

[10]: PWM κυματομορφή.

https://www.arduino.cc/en/Tutorial/PWM

[11]: Αναπτυξιακό Arduino Uno.

http://www.tested.com/tech/robots/456466-know-your-arduino-guide-most-common-boards/

https://www.arduino.cc/en/Main/ArduinoBoardUno

(Διαμορφωμένη εικόνα).

[12]: Κυκλωματική διάταξη H-Bridge.

http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/

[13]: Module Motor Shield Keyes LN298N.

http://www.geekonfire.com/wiki/index.php?title=Dual_H-Bridge_Motor_Driver

[14]: Αισθητήρας υπέρηχων HC-SR04 με διαστάσεις (45*20*15mm).

http://www.scoop.it/t/tecno4/p/4023699536/2014/06/27/sensor-de-ultrasonidos-hc-sr04-piezoelectrico-arduino.

[15]: Διάγραμμα λειτουργίας του αισθητήρα HC-SR04.

http://www.geekfactory.mx/tutoriales/tutoriales-arduino/sensor-ultrasonico-hc-sr04-y-arduino/

[16]: Bluetooth module HC-06.

http://arduinolearning.com/learning/arduino-and-hc-06-bluetooth-example.php

[17]: Τρόπος σύνδεσης Bluetooth HC-06 με το αναπτυξιακό Arduino UNO.

https://helentronica.wordpress.com/2014/06/07/say-your-arduino-how-much-you-love-it/

[18]: Arduino Sensor Shield V5.

http://forum.arduino.cc/index.php?topic=229646.0

[19]: Σχηματικό Διάγραμμα Arduino Sensor Shield.

http://forum.arduino.cc/index.php?topic=229646.0

[20]: GPS Module PMB-648.

https://www.parallax.com/product/28501

Αναφορές web

- https://www.arduino.cc/en/Main/ArduinoBoardUno
- http://www.instructables.com/id/Arduino-Modules-L298N-Dual-H-Bridge-Motor-Controll/
- http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/
- http://www.atmel.com/devices/atmega328p.aspx?tab=parameters
- https://arduino-info.wikispaces.com/SensorShield
- https://mcuoneclipse.com/2013/06/19/using-the-hc-06-bluetooth-module/
- http://www.instructables.com/id/AT-command-mode-of-HC-05-Bluetooth-module/
- http://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/
- http://www.instructables.com/id/Simple-Arduino-and-HC-SR04-Example/
- https://el.wikipedia.org/wiki/%CE%A6%CE%B1%CE%B9%CE%BD%CF%8C%CE%BC%CE%B5%CE%BD%CE%BF_%CE%9D%CF%84%CF%8C%CF%80%CE%BB%CE%B5%CF%81

- https://www.arduino.cc/en/Hacking/PinMapping168
- https://en.wikipedia.org/wiki/Ultrasound
- http://www.mathworks.com/products/instrument/supported/bluetooth.html
- http://courseware.mech.ntua.gr/ml23419/robotics_pdf/intro.pdf
- http://artemis-new.cslab.ece.ntua.gr:8080/jspui/bitstream/123456789/6429/1/DT2012-0200.pdf
- http://akrob.frederick.ac.cy/images/pdf/arxes-rompotikis/EyfiiKinoumenaRompot-1.pdf
- http://www.mathworks.com/videos/importing-geographic-data-and-creating-map-displays-68781.html
- https://www.parallax.com/
- *https://www.arduino.cc/en/Reference/SoftwareSerial*