

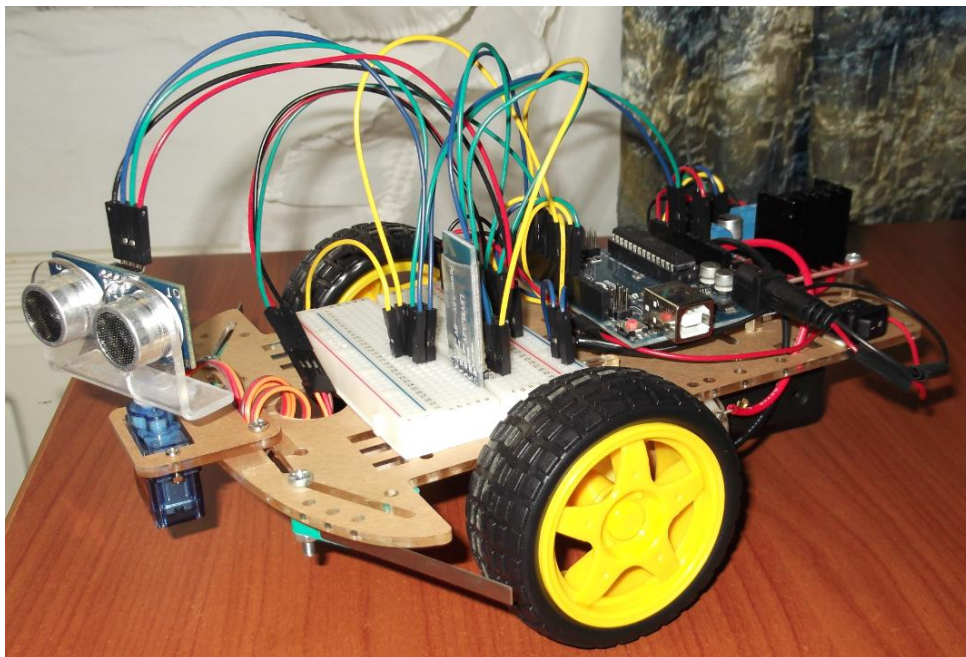


ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΑΥΤΟΜΑΤΙΣΜΟΥ Τ.Ε



ΘΕΜΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

“ΡΟΜΠΟΤΙΚΟ ΟΧΗΜΑ ΑΠΟΦΥΓΗΣ ΕΜΠΟΔΙΩΝ ΜΕ ΧΡΗΣΗ ΥΠΕΡΗΧΟΥ ΑΙΣΘΗΤΗΡΑ”



ΕΙΣΗΓΗΤΡΙΑ ΚΑΘΗΓΗΤΡΙΑ: Δρακάκη Μαρία

ΣΠΟΥΔΑΣΤΕΣ: Κιουρτζής Αναστάσιος

Κηπουρίδης Ορέστης

ΘΕΣΣΑΛΟΝΙΚΗ, ΜΑΙΟΣ 2017

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ	3
ΚΕΦΑΛΑΙΟ 1 ^ο	4
Ρομποτική και Ρομποτικά συστήματα.....	4
1.1 Εισαγωγή.....	4
1.2 Είδη ρομποτικών συστημάτων	5
1.3 Ιστορική αναδρομή σε έντροχα ρομποτικά συστήματα	7
ΚΕΦΑΛΑΙΟ 2 ^ο	10
Θεωρητικό μέρος (ανάλυση τεχνικών και ηλεκτρονικών εξαρτημάτων).....	10
2.1 Κεντρική μονάδα ελέγχου (Arduino UNO).....	10
2.2 L298N Dual H-Bridge Motor Controller module (Οδηγός Μοτέρ/Κινητήρων).....	21
2.3 Αισθητήρας υπερήχων HC-SR04 (Ultrasonic Sensor)	25
2.4 Μικροδιακόπτης επαφής (Microswitch).....	28
2.5 Σερβοκινητήρας (Tower pro 9g SG90).....	30
2.6 Τεχνολογία Bluetooth.....	32
2.7 Bluetooth Διάταξη (Module HC-06).....	38
ΚΕΦΑΛΑΙΟ 3 ^ο	39
Κατασκευαστικό μέρος.....	39
3.1 Εισαγωγή	39
3.2 Υλικά κατασκευής.....	40
3.3 Υλοποίηση κατασκευής	40
ΚΕΦΑΛΑΙΟ 4 ^ο	47
Προγραμματιστικό μέρος	47
4.1 Εισαγωγή.....	47
4.2 Κίνηση του ρομποτικού οχήματος μέσω φωνητικών εντολών.....	54
ΚΕΦΑΛΑΙΟ 5 ^ο	60
Δυνατότητες επέκτασης-αναβάθμισης του έντροχου ρομποτικού συστήματος και συμπεράσματα	60
5.1 Εισαγωγή.....	60
5.2 Συμπεράσματα πτυχιακής εργασίας.....	60
5.3 Βελτιώσεις πτυχιακής εργασίας.....	60
ΒΙΒΛΙΟΓΡΑΦΙΑ-ΔΙΑΔΙΚΤΥΑΚΑ ΑΡΘΡΑ.....	61

ΠΡΟΛΟΓΟΣ

Ένα από τα βασικότερα προβλήματα στον χώρο των αυτόματων ρομποτικών συστημάτων, είναι αυτό της αυτονομίας. Περιλαμβάνει, 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 μελέτη ενός έντροχου ρομποτικού συστήματος με σκοπό την κίνηση και αποφυγή εμποδίων μέσα σε ένα άγνωστο περιβάλλον. Επίσης το ρομποτικό σύστημα έχει την δυνατότητα ασύρματης επικοινωνίας με κάποιον χρήστη μέσω Smartphone ή άλλου συστήματος (π.χ Tablet), με στόχο τον τηλεχειρισμό αλλά και μέσω φωνητικών εντολών. 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 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 αυτομάτου ελέγχου καθώς και τον προγραμματισμό μικροελεγκτών και ενσωματωμένων συστημάτων, οι στόχοι που καλύπτει η εργασία είναι:

- Η κατανόηση του τρόπου λειτουργίας ενός συστήματος αυτομάτου ελέγχου.
- Η ανάπτυξη ενός ενσωματωμένου συστήματος ικανό να ελεγχθεί μέσω ασύρματης επικοινωνίας (Bluetooth).
- 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 θεωρία 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 ή 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, ενεργοποιητές /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 επιβλέπει και συντονίζει ολόκληρο το σύστημα, για την σχεδίαση και υλοποίηση του απαιτείται ο συνδυασμός γνώσεων από πολλές γνωστικές περιοχές, όπως είναι ο αυτόματος έλεγχος, η τεχνητή νοημοσύνη, η επιστήμη των υπολογιστών κλπ.

1.2 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 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 1.2.1.



Εικόνα 1.2.1 Ρομποτικό σύστημα σταθερής βάσης.

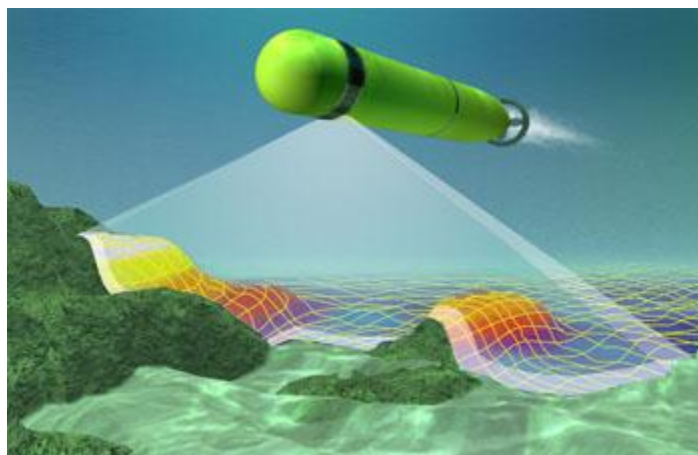
- **Κινούμενα ρομποτικά συστήματα:** Ως κινητά ρομποτικά συστήματα χαρακτηρίζονται όλα εκείνα τα συστήματα που έχουν τη δυνατότητα να μετακινήσουν όλα τα σημεία του μηχανισμού τους. Η δυνατότητα αυτή προσφέρεται από ειδικά συστήματα προώθησης, τα οποία μπορεί να είναι τροχοί, είτε πολυπλοκότερα, όπως 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 ή και προσωπικούς σκοπούς (φωτογραφία, έρευνα, βιντεοσκοπήσεις κ.α.). Στην εικόνα 1.2.2 απεικονίζεται ένα είδος τετρακοπτέρου.



Εικόνα 1.2.2 Τετρακόπτερο ρομποτικό σύστημα (Quadra Copter).

- **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 1.2.3 AUV ρομποτικό σύστημα.

1.3 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 20ου 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 κυκλοφόρησαν 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 αντιμετωπίσει 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 Aaxon 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, όπου 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 2^ο

ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ (Ανάλυση τεχνικών & ηλεκτρονικών εξαρτημάτων)

2.1 Τι είναι το Arduino;

Όπως το περιγράφει ο δημιουργός του, το Arduino είναι μια «ανοικτού κώδικα» πλατφόρμα «πρωτοτυποποίησης» ηλεκτρονικών, βασισμένη σε ευέλικτο και εύκολο στη χρήση hardware και software που προορίζεται για οποιονδήποτε έχει λίγη προγραμματιστική εμπειρία, στοιχειώδεις γνώσεις ηλεκτρονικών και ενδιαφέρεται να δημιουργήσει διαδραστικά αντικείμενα ή περιβάλλοντα.



Εικόνα 2.1.1 Arduino UNO Board.

Στην ουσία, πρόκειται για ένα ηλεκτρονικό κύκλωμα που βασίζεται στον μικροελεγκτή ATmega της Atmel και του οποίου όλα τα σχέδια, καθώς και το software που χρειάζεται για την λειτουργία του, διανέμονται ελεύθερα και δωρεάν ώστε να μπορεί να κατασκευαστεί από τον καθένα (απ' όπου και ο περίεργος -για hardware- χαρακτηρισμός «ανοικτού κώδικα»). Αφού κατασκευαστεί, μπορεί να συμπεριφερθεί σαν ένας μικροσκοπικός υπολογιστής, αφού ο χρήστης μπορεί να συνδέσει επάνω του πολλαπλές μονάδες εισόδου/εξόδου και να προγραμματίσει τον μικροελεγκτή να δέχεται δεδομένα από τις μονάδες εισόδου, να τα επεξεργάζεται και να στέλνει κατάλληλες εντολές στις μονάδες εξόδου.

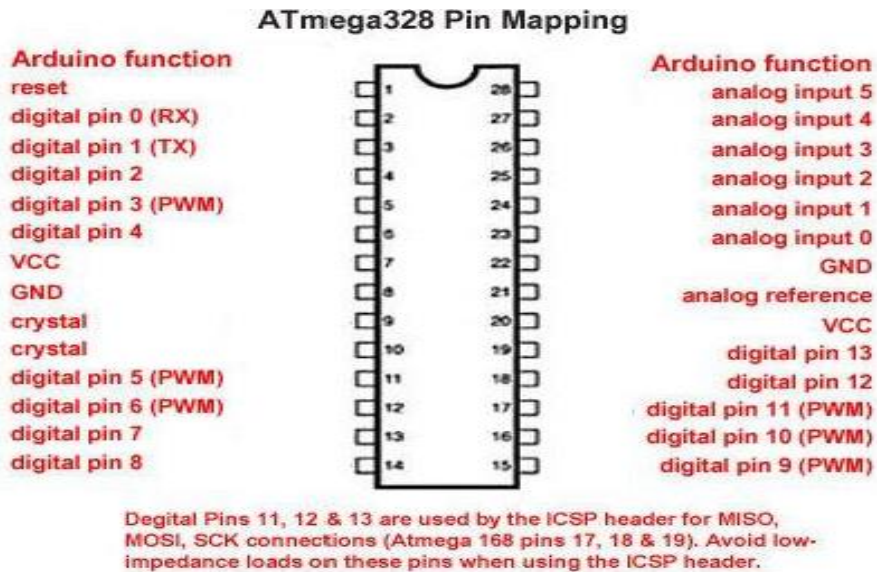
Μάλιστα κάποιος θα μπορούσε να ισχυριστεί - και θα ήταν ένας αρκετά πετυχημένος παραλληλισμός - ότι λειτουργικά το Arduino μοιάζει πολύ με το NXT Brick των Lego Mindstorms NXT. Άλλωστε η ρομποτική είναι μια από τις πολλές εφαρμογές στις οποίες το Arduino διαπρέπει.

Το Arduino βέβαια, δεν είναι ούτε ο μοναδικός, ούτε και ο καλύτερος δυνατός τρόπος για την δημιουργία μιας οποιασδήποτε διαδραστικής ηλεκτρονικής συσκευής. Όμως το κύριο πλεονέκτημά του είναι η τεράστια κοινότητα που το υποστηρίζει και η οποία έχει δημιουργήσει, συντηρεί και επεκτείνει μια ανάλογου μεγέθους online γνωσιακή βάση. Έτσι, παρότι ένας έμπειρος ηλεκτρονικός μπορεί να προτιμήσει διαφορετική πλατφόρμα ή εξαρτήματα ανάλογα με την εφαρμογή που έχει στον νου του, το Arduino, με το εκτενές documentation, καταφέρνει να κερδίσει όλους αυτούς των οποίων οι γνώσεις στα ηλεκτρονικά περιορίζονται στα όσα λίγα έμαθαν στο σχολείο.

Μικροελεγκτής - η καρδιά του Arduino

Το Arduino βασίζεται στον ATmega328, έναν 8-bit RISC μικροελεγκτή, τον οποίο χρονίζει στα 16MHz. Ο ATmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων:

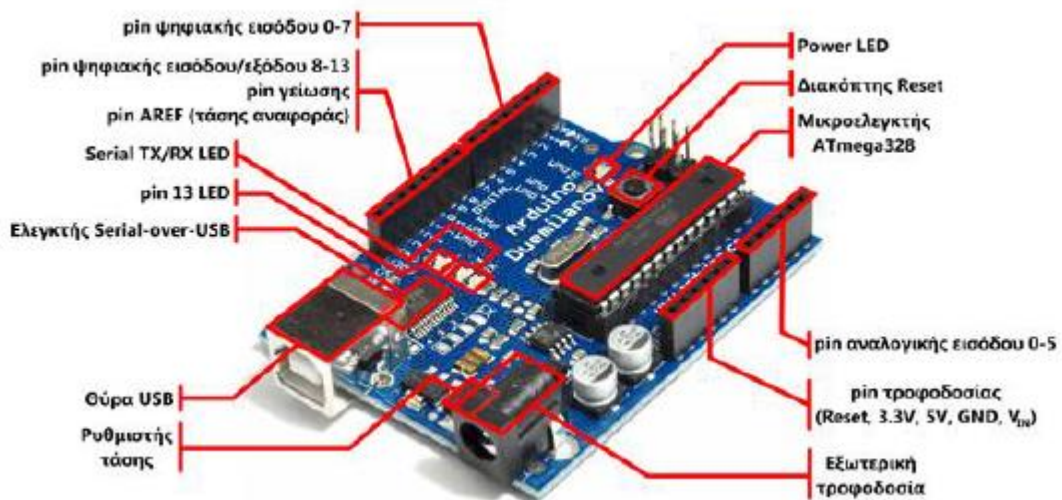
- 2Kb μνήμης SRAM που είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματά σας για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. κατά το runtime. Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή αν γίνει reset.
- 1Kb μνήμης EEPROM η οποία μπορεί να χρησιμοποιηθεί για «ωμή» εγγραφή/ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματά σας κατά το runtime. Σε αντίθεση με την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset οπότε είναι το ανάλογο του σκληρού δίσκου.
- 32Kb μνήμης Flash, από τα οποία τα 2Kb χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware αυτό που στην ορολογία του Arduino ονομάζεται bootloader είναι αναγκαίο για την εγκατάσταση των δικών σας προγραμμάτων στον μικροελεγκτή μέσω της θύρας USB, χωρίς δηλαδή να χρειάζεται εξωτερικός hardware programmer. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή σας. Η μνήμη Flash, όπως και η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset. Επίσης, ενώ η μνήμη Flash υπό κανονικές συνθήκες δεν προορίζεται για χρήση runtime μέσα από τα προγράμματά σας, λόγω της μικρής συνολικής μνήμης που είναι διαθέσιμη σε αυτά (2Kb SRAM + 1Kb EEPROM), έχει σχεδιαστεί μια βιβλιοθήκη που επιτρέπει την χρήση όσου χώρου περισσεύει (30Kb μείον το μέγεθος του προγράμματός σας σε μεταγλωττισμένη μορφή).



Εικόνα 2.1.2 Σχηματικό ακροδεκτών ολοκληρωμένου ATmega328.

Είσοδοι – Έξοδοι

Καταρχήν, το Arduino διαθέτει σειριακό interface. Ο μικροελεγκτής ATmega 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 την ώρα που εκτελείται. Επιπλέον, στην πάνω πλευρά του Arduino βρίσκονται 14 θηλυκά pin, αριθμημένα από 0 ως 13, που μπορούν να λειτουργήσουν ως ψηφιακές εισοδοι και έξοδοι. Λειτουργούν στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί ρεύμα το πολύ 40mA.



Εικόνα 2.1.3 Σχηματικό ακροδεκτών Arduino board.

Ως ψηφιακή έξοδος, ένα από αυτά τα pin μπορεί να τεθεί από το πρόγραμμά σας σε κατάσταση HIGH ή LOW, οπότε το Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι ρεύμα στο συγκεκριμένο pin. Με αυτόν τον τρόπο μπορείτε λόγω χάρη να ανάψετε και να σβήσετε ένα LED που έχετε συνδέσει στο συγκεκριμένο pin. Αν πάλι ρυθμίσετε ένα από αυτά τα pin ως ψηφιακή είσοδο μέσα από το πρόγραμμά σας, μπορείτε με την κατάλληλη εντολή να διαβάσετε την κατάστασή του (HIGH ή LOW) ανάλογα με το αν η εξωτερική συσκευή που έχετε συνδέσει σε αυτό το pin διοχετεύει ή όχι ρεύμα στο pin (με αυτόν τον τρόπο λόγω χάρη μπορείτε να «διαβάζετε» την κατάσταση ενός διακόπτη).

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

Τα pin 0 και 1 λειτουργούν ως RX και TX της σειριακής όταν το πρόγραμμά σας ενεργοποιεί την σειριακή θύρα. Έτσι, όταν λόγω χάρη το πρόγραμμά σας στέλνει δεδομένα στην σειριακή, 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 πρόγραμμά σας ενεργοποιήσετε το σειριακό interface, χάνετε 2 ψηφιακές εισόδους/εξόδους.

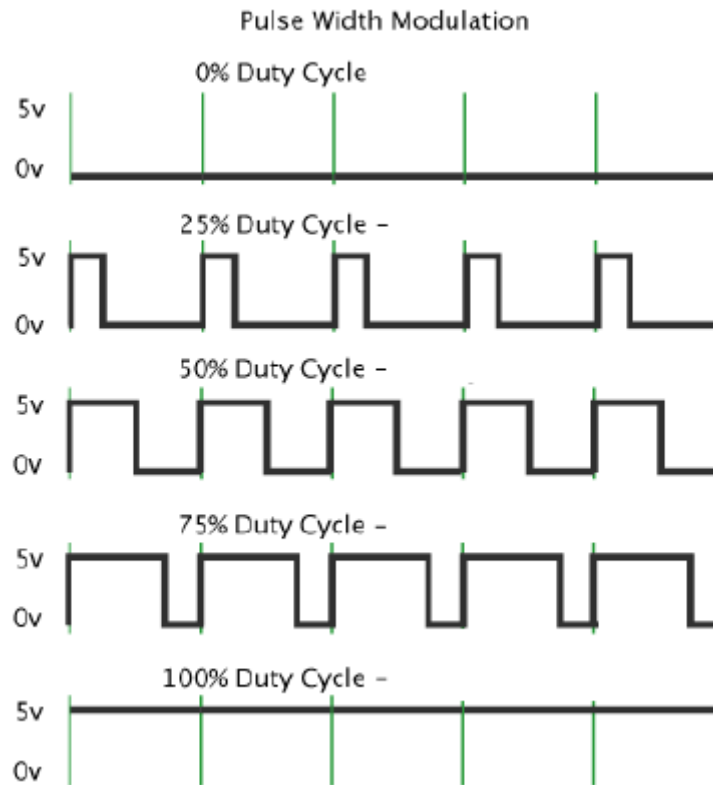
Τα pin 2 και 3 λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Με άλλα λόγια, μπορείτε να τα ρυθμίσετε μέσα από το πρόγραμμά σας ώστε να λειτουργούν αποκλειστικά ως ψηφιακές είσοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει “άμεσα” και εκτελείται μια συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

Το **Arduino UNO**, όπως προαναφέρθηκε αποτελείται από 14 ψηφιακά “ποδαράκια” (Pin) 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,μs** κλπ.), ή σε ποσοστό (%) επι της περιόδου. Αυτή η εναλλαγή μεταξύ **ON** και **OFF** μπορεί να προσομοιώσει ένα εύρος τάσεων, για παράδειγμα 0 ως 5V. Στην Εικόνα 2.1.4 παρακάτω φαίνεται η **PWM** κυματομορφή.

Επίσης, τα συγκεκριμένα pin 3, 5, 6, 9, 10 και 11 είναι αυτά που μπορούν να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι με το σύστημα PWM (Pulse Width Modulation), δηλαδή το ίδιο σύστημα που διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων. Έτσι, μπορείτε να συνδέσετε λόγω χάρη ένα LED σε κάποιο από αυτά τα pin και να ελέγξετε πλήρως την φωτεινότητά του με ανάλυση 8bit (256 καταστάσεις από 0-σβηστό ως 255-πλήρως αναμμένο)

αντί να έχετε απλά την δυνατότητα αναμμένο-σβηστό που παρέχουν οι υπόλοιπες ψηφιακές έξοδοι.

Είναι σημαντικό να καταλάβετε ότι το PWM δεν είναι πραγματικά αναλογικό σύστημα και ότι θέτοντας στην έξοδο την τιμή 127, δεν σημαίνει ότι η έξοδος θα δίνει 2.5V αντί της κανονικής τιμής των 5V, αλλά ότι θα δίνει ένα παλμό που θα εναλλάσσεται με μεγάλη συχνότητα και για ίσους χρόνους μεταξύ των τιμών 0 και 5V.



Εικόνα 2.1.4 PWM κυματομορφή.

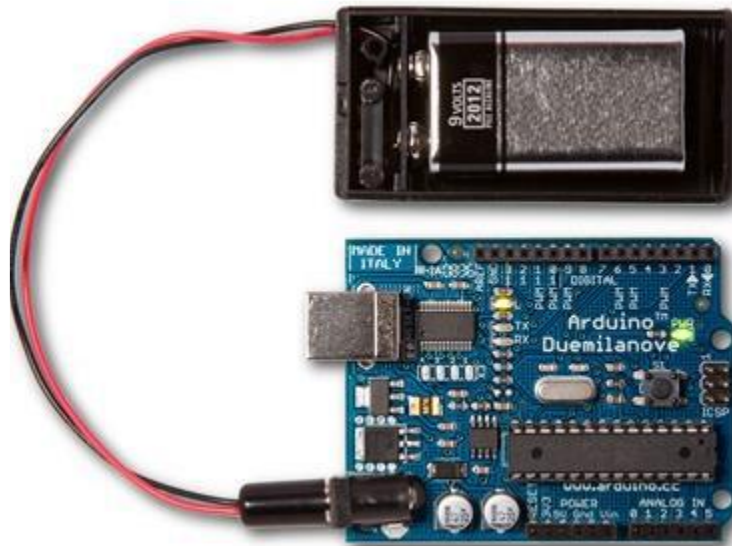
Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN, θα βρείτε μια ακόμη σειρά από 6 pin, αριθμημένα από το 0 ως το 5. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC (Analog to Digital Converter) που είναι ενσωματωμένο στον μικροελεγκτή. Για παράδειγμα, μπορείτε να τροφοδοτήσετε ένα από αυτά με μια τάση την οποία μπορείτε να κυμαίνεται με ένα ποτενσιόμετρο από 0V ως μια τάση αναφοράς V_{ref} η οποία, αν δεν κάνετε κάποια αλλαγή είναι προ ρυθμισμένη στα 5V. Τότε, μέσα από το πρόγραμμά σας μπορείτε να «διαβάσετε» την τιμή του pin ως ένα ακέραιο αριθμό ανάλυσης 10-bit, από 0 (όταν η τάση στο pin είναι 0V) μέχρι 1023 (όταν η τάση στο pin είναι 5V). Η τάση αναφοράς μπορεί να ρυθμιστεί με μια εντολή στο 1.1V, ή σε όποια τάση επιθυμείτε (μεταξύ 2 και 5V) τροφοδοτώντας εξωτερικά με αυτή την τάση το pin με την σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας.

Έτσι, αν τροφοδοτήσετε το pin AREF με 3.3V και στην συνέχεια δοκιμάσετε να διαβάσετε κάποιο pin αναλογικής εισόδου στο οποίο εφαρμόζετε τάση 1.65V, το Arduino θα σας επιστρέψει την τιμή 512.

Τέλος, καθένα από τα 6 αυτά pin, 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 περιγράφηκαν πριν. Σε αυτή την περίπτωση τα pin μετονομάζονται από 0~5 σε 14~19 αντίστοιχα.

Τροφοδοσία

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 2.1mm (θετικός πόλος στο κέντρο) και βρίσκεται στην κάτω-αριστερή γωνία του Arduino.



Εικόνα 2.1.5 Εξωτερική τροφοδοσία Arduino με την βοήθεια 9V μπαταρίας.

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

Δίπλα από τα pin αναλογικής εισόδου, υπάρχει μια ακόμα συστοιχία από 6 pin με την σήμανση POWER. Η λειτουργία του καθενός έχει ως εξής:

Το πρώτο, με την ένδειξη RESET, όταν γειωθεί (σε οποιοδήποτε από τα 3 pin 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 τα εξαρτήματά σας 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 τα εξαρτήματά σας 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 «φέρει» στα 5V.

Το τέταρτο και το πέμπτο pin, με την ένδειξη GND, είναι φυσικά 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 σας βολεύει να χρησιμοποιήσετε την υποδοχή του φics των 2.1mm. Αν όμως έχετε ήδη συνδεδεμένη εξωτερική τροφοδοσία μέσω του φics, μπορείτε να χρησιμοποιήσετε αυτό το pin για να τροφοδοτήσετε εξαρτήματα με την πλήρη τάση της εξωτερικής τροφοδοσίας (7~12V), πριν αυτή περάσει από τον ρυθμιστή τάσης όπως γίνεται με το pin των 5V.

Ενσωματωμένα κουμπιά και LED

Πάνω στην πλακέτα του Arduino υπάρχει ένας διακόπτης micro-switch και 4 μικροσκοπικά LED επιφανειακής στήριξης. Η λειτουργία του διακόπτη (που έχει την σήμανση RESET) και του ενός LED με την σήμανση POWER είναι μάλλον προφανής. Τα δύο LED με τις σημάνσεις TX και RX, χρησιμοποιούνται ως ένδειξη λειτουργίας του σειριακού interface, καθώς ανάβουν όταν το Arduino στέλνει ή λαμβάνει (αντίστοιχα) δεδομένα μέσω USB. Σημειώστε ότι τα LED 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.

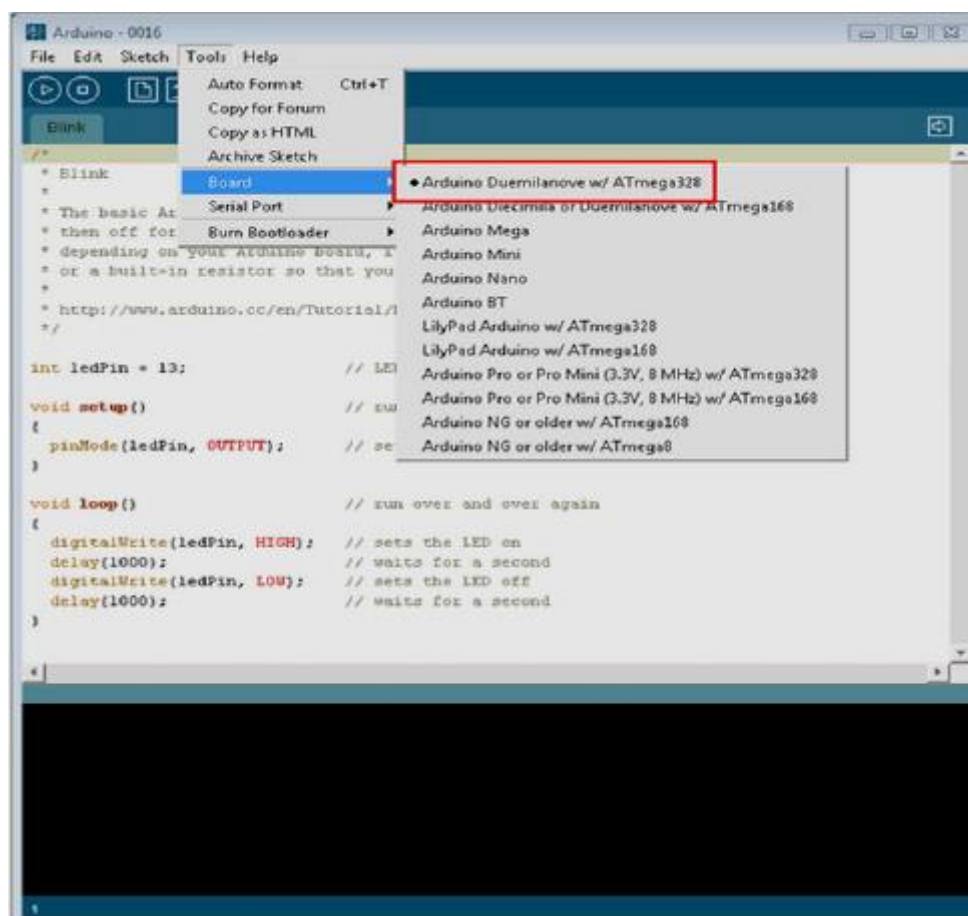
Τέλος, υπάρχει το LED με την σήμανση L. Η βασική δοκιμή λειτουργίας του Arduino είναι να του αναθέσετε να αναβοσβήνει ένα LED (θα το δείτε αυτό στην συνέχεια όταν θα φτιάξετε την πρώτη εφαρμογή σας). Για να μπορείτε να το κάνετε αυτό από την πρώτη στιγμή, χωρίς να συνδέσετε τίποτα πάνω στο Arduino, οι κατασκευαστές του σκέφτηκαν να ενσωματώσουν ένα LED στην πλακέτα, το οποίο σύνδεσαν στο ψηφιακό pin 13. Έτσι, ακόμα και αν δεν έχετε συνδέσει τίποτα πάνω στο φυσικό pin 13, αναθέτοντάς του την τιμή HIGH μέσα από το πρόγραμμά σας, θα ανάψει αυτό το ενσωματωμένο LED.

Arduino IDE και σύνδεση με τον υπολογιστή

Ότι χρειάζεστε για την διαχείριση του Arduino από τον υπολογιστή σας το παρέχει το Arduino IDE, την τελευταία έκδοση του οποίου μπορείτε να κατεβάσετε από το επίσημο site για καθένα από τα τρία δημοφιλέστερα λειτουργικά συστήματα.

Το Arduino IDE είναι βασισμένο σε Java και συγκεκριμένα παρέχει:

- ένα πρακτικό περιβάλλον για την συγγραφή των προγραμμάτων σας (τα οποία ονομάζονται σχέδιο (sketch) στην ορολογία του Arduino) με συντακτική χρωματική σήμανση,
- μερικές έτοιμες βιβλιοθήκες για προέκταση της γλώσσας και για να χειρίζεστε εύκολα μέσα από τον κώδικά σας τα εξαρτήματα που συνδέετε στο Arduino,
- τον compiler για την μεταγλώττιση των sketch σας,
- μια σειριακή οθόνη (serial monitor) που παρακολουθεί τις επικοινωνίες της σειριακής (USB), αναλαμβάνει να στείλει αλφαριθμητικά της επιλογής σας στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για το debugging των sketch σας,
- και την επιλογή να ανεβάσετε το μεταγλωττισμένο sketch στο Arduino.



Εικόνα 2.1.6 Επιλογή κατάλληλου Arduino Board μέσω του περιβάλλοντος Arduino IDE.

Γλώσσα προγραμματισμού

Η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring, μια παραλλαγή 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 καταγωγής της από την C, στην γλώσσα του Arduino μπορείτε να χρησιμοποιήσετε ουσιαστικά τις ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους 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
LOW	Σταθερά	int	-
HIGH	Σταθερά	int	-
INPUT	Σταθερά	int	-
OUTPUT	Σταθερά	int	-
pinMode	Εντολή	-	(pin, mode)
digitalWrite	Εντολή	-	(pin, pinstatus)
digitalRead	Συνάρτηση	int	(pin)
analogReference	Εντολή	-	(type)
analogRead	Συνάρτηση	int	(pin)
analogWrite	Εντολή	-	(pin, value)
millis	Συνάρτηση	unsigned long	()
delay	Εντολή	-	(time)
attachInterrupt	Εντολή	-	(interrupt, function, triggermode)
detachInterrupt	Εντολή	-	(interrupt)
noInterrupts	Εντολή	-	()
interrupts	Εντολή	-	()
Serial.begin	Μέθοδος κλάσης	-	(datarate)
Serial.println	Μέθοδος κλάσης	-	(data)

Εικόνα 2.1.7 Βοηθητικός πίνακας σταθερών, ειδικών εντολών, συναρτήσεων για τον προγραμματισμό.

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

// Ενσωματώσεις βιβλιοθηκών, δηλώσεις μεταβλητών...

```
void setup()
```

```
{
```

```
  // ...
```

```
}
```

```
void loop()
```

```
{
```

```
  // ...
```

```
}
```

// Υπόλοιπες συναρτήσεις...

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)`.

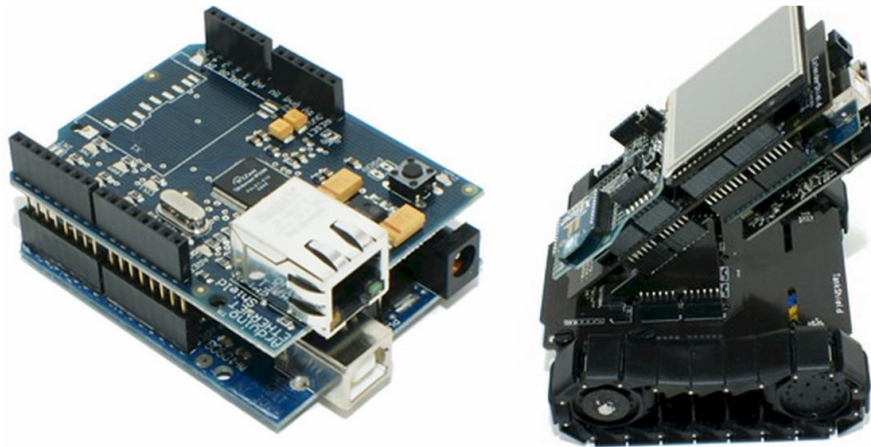
Ασπίδες Arduino (Shield)

Τα shield είναι ολοκληρωμένες πλακέτες οι οποίες είναι σχεδιασμένες έτσι ώστε να κουμπώνουν πάνω στο Arduino προεκτείνοντας την λειτουργικότητά του. Είναι η hardware αντίστοιχη έννοια των plugin, addon και extension που υπάρχουν στο hardware.

Μερικά από τα πιο δημοφιλή shield που κυκλοφορούν στο εμπόριο για το Arduino είναι:

- *Ethernet shield*: Δίνει στο Arduino την δυνατότητα να δικτυωθεί σε ένα LAN ή στο internet μέσω ενός τυπικού καλωδίου Ethernet.
- *WiFi shield*: Όμοιο με το Ethernet shield, χωρίς φυσικά το καλώδιο.
- *Διάφορα shield οθόνης*: Προσθέτουν οθόνη στο Arduino. Κυκλοφορούν από απλές οθόνες τύπου calculator μέχρι OLED touchscreen υψηλής ανάλυσης τύπου iPhone.
- *Wave shield*: Δίνει στο Arduino την δυνατότητα να παίζει ήχους/μουσική από κάρτες SD.
- *GPS shield*: Προσθέτει GPS δυνατότητες στο Arduino (εντοπισμό στίγματος).

- *Διάφορα Motor Shields:* Σας επιτρέπουν να οδηγήσετε εύκολα μοτέρ διάφορων τύπων (απλά DC, servo, stepper κ.λπ.) από το Arduino.
- *ProtoShield:* Προσχεδιασμένη πλακέτα πρωτοτυποποίησης, συμβατή στις διαστάσεις του Arduino και χωρίς εξαρτήματα για να φτιάξετε το δικό σας shield.



Εικόνα 2.1.8 Arduino Shield.

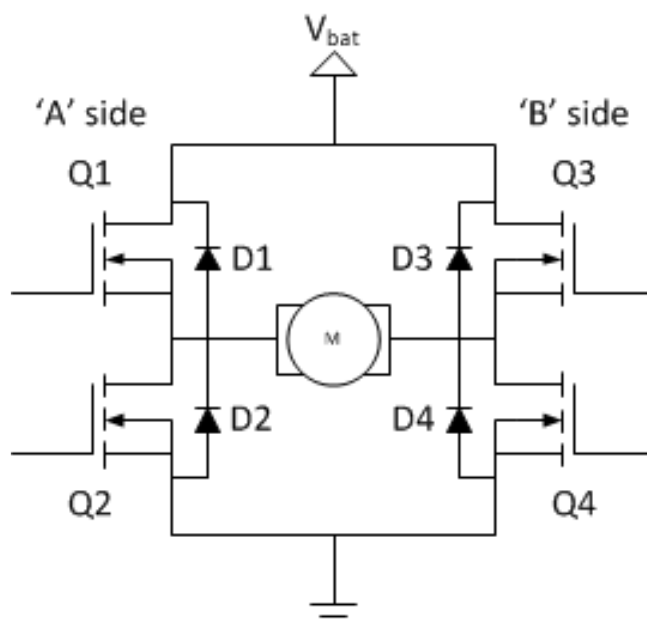
Τα shield είναι σχεδιασμένα ώστε αφού κουμπωθούν πάνω στο Arduino να προωθούν τις υποδοχές του, ώστε να μπορείτε να συνδέσετε επιπλέον τα δικά σας εξαρτήματα ή να κουμπώσετε και επόμενο shield. Φυσικά, το κάθε shield χρησιμοποιεί ορισμένους από τους πόρους συνδεσιμότητας του Arduino και έτσι δεν μπορείτε να συνδέσετε απεριόριστα shield. Μάλιστα κάποια shield μπορεί να μην είναι συμβατά μεταξύ τους γιατί χρησιμοποιούν τα ίδια pin του Arduino για επικοινωνία με αυτό. Επίσης, επειδή κάποια shield δεν προωθούν τις συνδέσεις του Arduino (όπως π.χ. οι οθόνες οι οποίες δεν έχουν νόημα αν τις καλύψετε από πάνω με ένα επόμενο shield), υπάρχουν ειδικά extender shield που κουμπώνουν στο Arduino και δίνουν την δυνατότητα σε δύο άλλα shield να κουμπώσουν πάνω τους, λειτουργώντας σαν πολύπριζα.

Όπως και για το ίδιο το Arduino, το βασικό πλεονέκτημα των shield δεν είναι τόσο το προφανές πλεονέκτημα του έτοιμου hardware όσο ότι συνοδεύονται συνήθως από έτοιμες βιβλιοθήκες που σας επιτρέπουν να προγραμματίζετε τα sketch σας σε high level. Έτσι, λόγω χάρη, δεν χρειάζεται να διαβάσετε datasheet ή να γίνετε ηλεκτρονικός για να συνδέσετε και να λειτουργήσετε ένα GPS module πάνω στο Arduino. Απλά συνδέετε το shield, εγκαθιστάτε τη βιβλιοθήκη που το συνοδεύει και χρησιμοποιείτε μια έτοιμη συνάρτηση -του στυλ getLocation- για να πάρετε το γεωγραφικό στίγμα και να το επεξεργαστείτε περαιτέρω στο sketch σας.

Τα shield σας λύνουν τα χέρια όταν θέλετε να δημιουργήσετε εύκολα ένα πραγματικά πρακτικό project. Αυτός είναι και ο λόγος που δεν συνιστάται η αγορά κάποιας έκδοσης του Arduino που δεν είναι 100% συμβατή με τα shield.

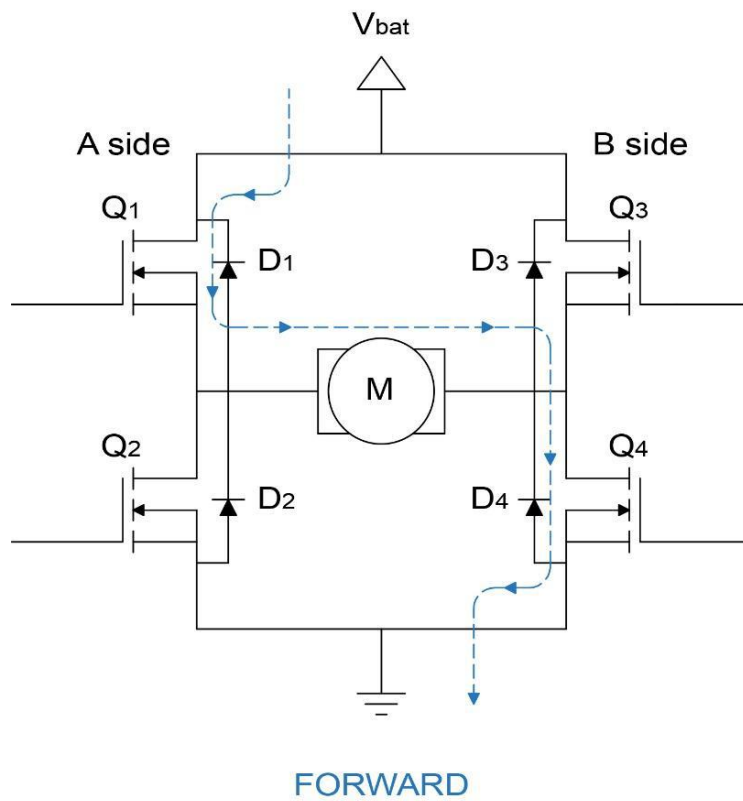
2.2 L298N Dual H- Bridge Motor Controller Module (Οδηγός Μοτέρ/Κινητήρων)

Για να ελέγξουμε τους 2 κινητήρες του ρομποτικού συστήματος χρησιμοποιούμε το L298N Dual Motor Controller. Το συγκεκριμένο Module, βασίζεται στο chip L298N. Το L298N, αποτελείται από 2 γέφυρες H, γνωστές και ως H-Bridge. Αυτή η κυκλωματική διάταξη επιτρέπει στο chip 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 "H", αφού έχει στα αριστερά και δεξιά από ένα ζευγάρι διακοπές, όπου συνήθως είναι διπολικά, ή FET transistor, ενώ στο κέντρο έχει το φορτίο, όπως στην εικόνα 2.2.1. Οι δίοδοι που χρησιμοποιούνται για προστασία, είναι συνήθως Schotky.

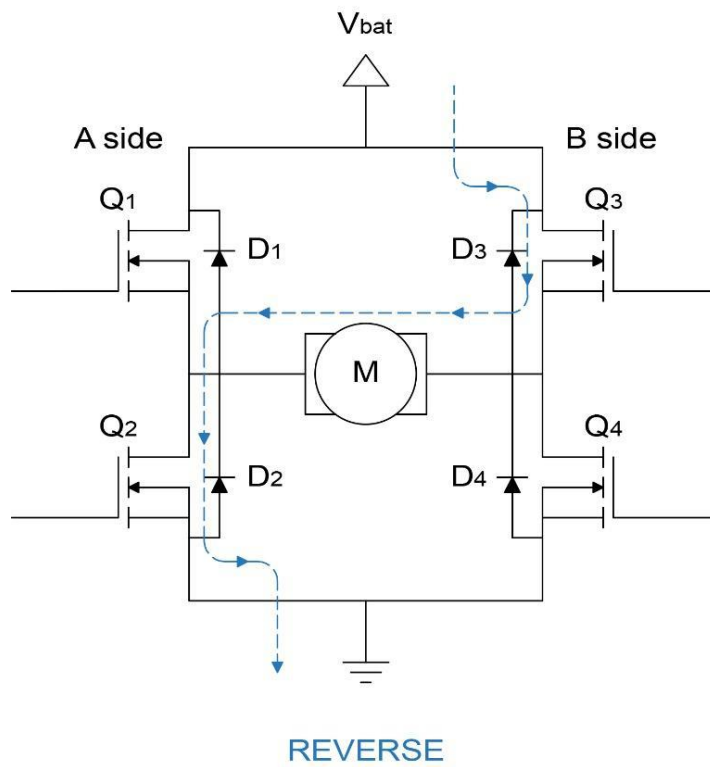


Εικόνα 2.2.1 Κυκλωματική Διάταξη H-Bridge.

Η λειτουργία της είναι απλή, καθώς όταν τα transistor της εικόνας 2.2.1 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 transistor 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 παρακάτω εικόνες :



Εικόνα 2.2.2 Πορεία ρεύματος στην αριστερόστροφη κίνηση.



Εικόνα 2.2.3 Πορεία ρεύματος στην δεξιόστροφη κίνηση.

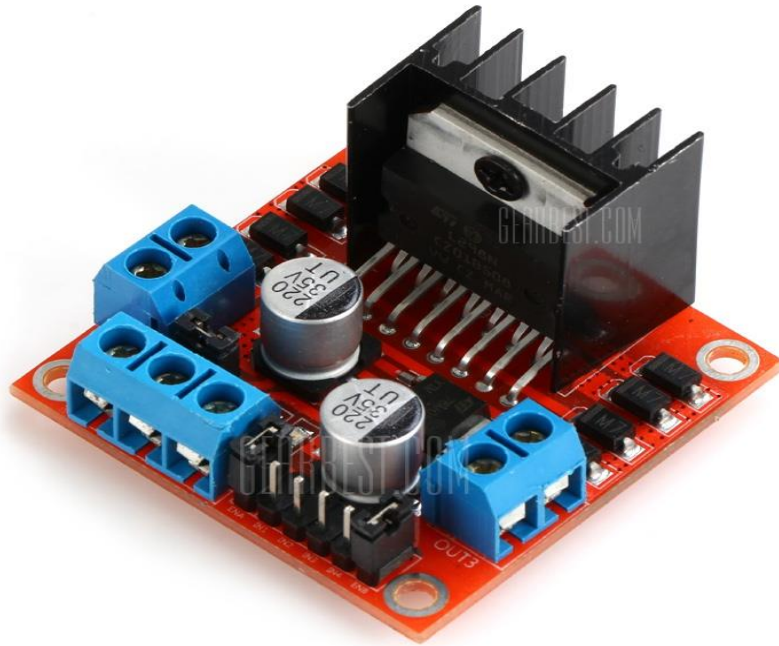
Το L298N Dual Motor Controller συνδέεται με το αναπτυξιακό Arduino UNO και συγκεκριμένα χρησιμοποιεί τα ψηφιακά ποδαράκια (Pin) 11,10,6,5. Μέσω αυτών των Pin στέλνει PWM παλμούς ώστε να οδηγήσει τα μοτέρ, ρυθμίζοντας την ταχύτητα και την διεύθυνση. Να σημειωθεί ότι το Duty Cycle είναι ανάλογο της ταχύτητας των μοτέρ. Αυξάνοντας το, αυξάνουμε και την ταχύτητα περιστροφής των μοτέρ.

Μπορούμε να τροφοδοτήσουμε το L298N Dual Motor Controller 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 Module με εξωτερική τροφοδοσία (5-12V), όπως αναγράφεται με πραγματική τιμή τάσης εισόδου 5-35V.

Συνοψίζοντας το module περιέχει:

- Δυο διαφορετικές συνδέσεις τροφοδοσίας.
- Το chip L298N 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 5V, καθώς και για την ορθή λειτουργία των μοτέρ.
- Κουμπί Reset, όπου στην πραγματικότητα είναι το ίδιο με την πλακέτα του Arduino UNO και επαναφέρει το πρόγραμμα που έχουμε φορτώσει, σε περίπτωση σφάλματος.
- Κλέμες PCB όπου μας διευκολύνει στην εξωτερική σύνδεση των εξαρτημάτων, στην περίπτωση μας συνδέουμε τα μοτέρ.

Λόγο της κατασκευής του το module είναι ένα πολύ εύχρηστο εργαλείο οδήγησης επαγωγικών φορτίων. Συνδέοντας 1 μοτέρ σε κάθε H-γέφυρα έχουμε την δυνατότητα να οδηγήσουμε τα μοτέρ του ρομποτικού συστήματος. Στην εικόνα 2.2.4 παρουσιάζεται το L298N Dual H-Bridge driver motor module.



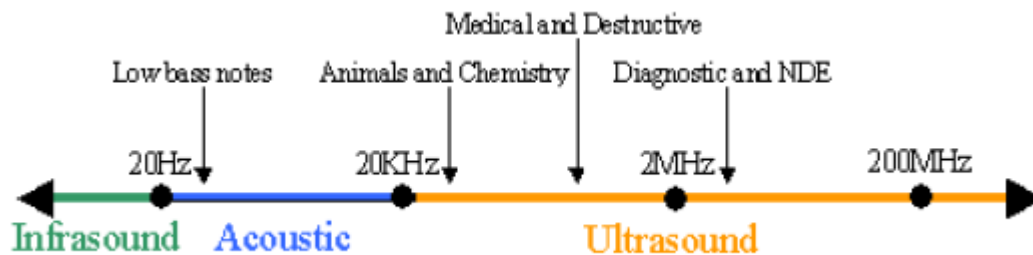
Εικόνα 2.2.4 L298N Dual H-Bridge driver motor module.

Προδιαγραφές L298N Dual H-Bridge driver motor module

- Βάρος προϊόντος: 0,005 kg
- Chip ελέγχου: L298N
- Λογική τάση: 5V
- Τάση οδήγησης: 5V - 35V
- Λογικό ρεύμα: 0mA - 36mA
- Ρεύμα οδήγησης: 2A (MAX ανα γέφυρα)
- Μέγιστη ισχύς: 25W

2.3 Αισθητήρας υπερήχων HC-SR04 (Ultrasonic sensor)

Γενικά, 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 20Hz έως 20KHz, ενώ οι υπέρηχοι που αποστέλλει ο αισθητήρας είναι συχνότητας 40KHz.



Εικόνα 2.3.1 Εύρος συχνοτήτων υπερήχων.

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.



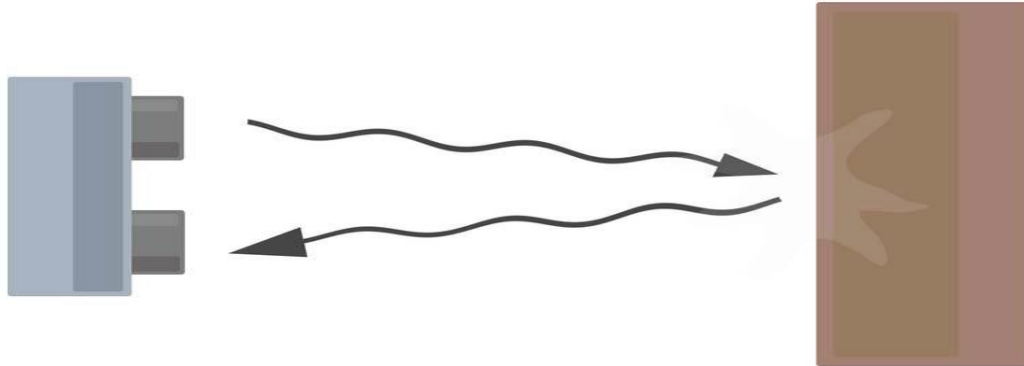
Εικόνα 2.3.2 Σχηματική απεικόνιση του φαινομένου Doppler.

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 ή 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 η αρχή λειτουργίας τους έχει ως εξής: 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 επιστρέφει προς τον αισθητήρα, γίνεται αισθητή μέσω ενός μικροφώνου.



Εικόνα 2.3.3 Αρχή λειτουργίας του αισθητήρα υπερήχων.

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 επιστρέφεται η απόσταση του αισθητήρα από το αντικείμενο.

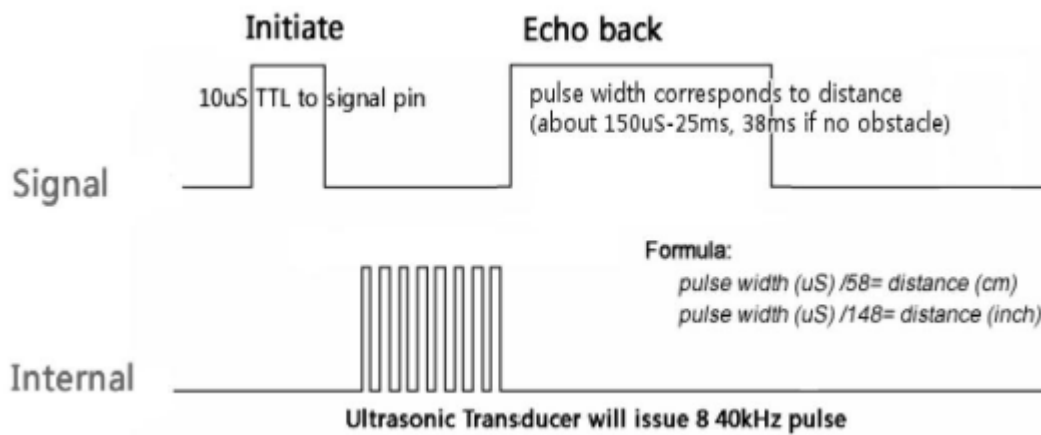
Ο αισθητήρας υπερήχων HC-SR04 παρέχει ένα εύρος μέτρησης από 2cm έως 400cm με ακρίβεια που αγγίζει τα 3mm. Το εξάρτημα αποτελείται από έναν πομπό (Trigger) και έναν δέκτη (Echo). Επίσης έχει ένα Pin για την τάση λειτουργίας του Vcc καθώς και ένα για την γείωση, GND. Στην εικόνα που ακολουθεί βλέπουμε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



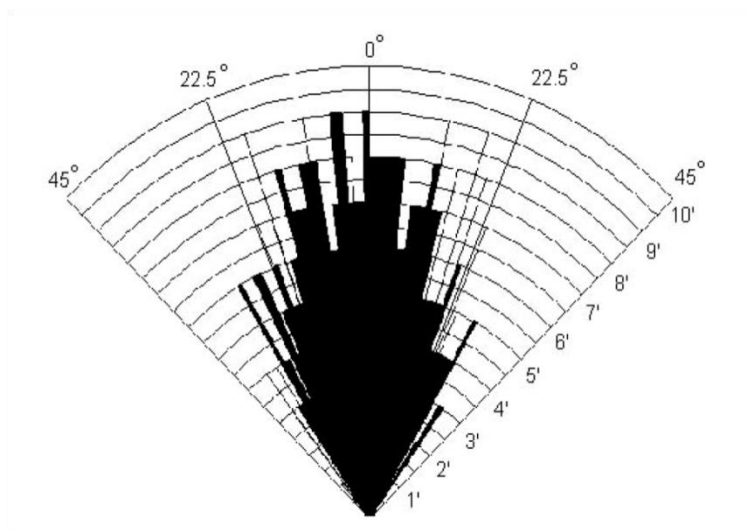
Εικόνα 2.3.4 Αισθητήρας υπερήχων HC-SR04 με διαστάσεις (45*20*15mm).

Για να λειτουργήσει χρησιμοποιεί παλμούς διάρκειας τουλάχιστον 10μs. Στέλνει αυτόματα 8 παλμούς συχνότητας 40KHz 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 340m/s. Η τάση λειτουργίας του αισθητήρα είναι 5V και παρέχεται από τον μικροελεγκτή. Το ρεύμα λειτουργίας είναι 15mA και η συχνότητα λειτουργίας είναι 40Hz.

Πολύ σημαντικό στοιχείο είναι ότι η γωνιά μέτρησης που επιτυγχάνει ο αισθητήρας είναι οι 30° (μοίρες).



Εικόνα 2.3.5 Διάγραμμα λειτουργίας του αισθητήρα HC-SR04.



Εικόνα 2.3.6 Γωνίες λειτουργίας του υπέρηχου αισθητήρα HC-SR04.

2.4 Μικροδιακόπτης επαφής (Microswitch SPDT ON-(ON) - with Big Lever)

Ένας μικροσκοπικός διακόπτης τύπου snap-action, επίσης εμπορικός και συχνά γνωστός ως μικροδιακόπτης, είναι ένας ηλεκτρικός διακόπτης που ενεργοποιείται από πολύ μικρή φυσική δύναμη, μέσω της χρήσης ενός μηχανισμού αιχμής, ο οποίος αποκαλείται και μερικές φορές ένας μηχανισμός "πάνω από κέντρο".

Η εναλλαγή γίνεται αξιόπιστα σε συγκεκριμένες και επαναλαμβανόμενες θέσεις του ενεργοποιητή, κάτι που δεν ισχύει απαραίτητα για άλλους μηχανισμούς. Είναι πολύ συνηθισμένες λόγω του χαμηλού κόστους και της αντοχής τους, περισσότερο από 1 εκατομμύριο κύκλους και έως 10 εκατομμύρια κύκλους για μοντέλα βαρέως τύπου. Αυτή η ανθεκτικότητα αποτελεί φυσική συνέπεια του σχεδιασμού.

Το καθοριστικό χαρακτηριστικό των μικροδιακοπών είναι ότι μια σχετικά μικρή κίνηση στο κουμπί ενεργοποίησης παράγει μια σχετικά μεγάλη κίνηση στις ηλεκτρικές επαφές, η οποία συμβαίνει σε υψηλή ταχύτητα (ανεξάρτητα από την ταχύτητα ενεργοποίησης). Τα πιο επιτυχημένα σχέδια παρουσιάζουν επίσης υστέρηση, πράγμα που σημαίνει ότι μια μικρή αναστροφή του ενεργοποιητή είναι ανεπαρκής για την αντιστροφή των επαφών. Πρέπει να υπάρξει μια σημαντική κίνηση προς την αντίθετη κατεύθυνση. Και τα δύο αυτά χαρακτηριστικά βοηθούν στην επίτευξη μιας καθαρής και αξιόπιστης διακοπής στο κύκλωμα μεταγωγής.



Εικόνα 2.4.1 Σύγκριση μερικών διαφορετικών μικροδιακοπών.

Εφαρμογές μικροδιακόπτη επαφής

Κοινές εφαρμογές μικροδιακοπών περιλαμβάνουν την πόρτα κλειδώματος σε ένα φούρνο μικροκυμάτων, διακόπτες ασφαλείας σε ανελκυστήρες, μηχανήματα αυτόματης πώλησης, καθώς και για τον εντοπισμό εμπλοκών χαρτιού ή άλλες βλάβες στα φωτοτυπικά μηχανήματα. Οι μικροδιακόπτες χρησιμοποιούνται συνήθως σε διακόπτες παραβίασης στις βαλβίδες πύλης σε συστήματα καταιωτιστήρων πυρκαγιάς και σε άλλα συστήματα σωληνώσεων νερού, όπου είναι απαραίτητο να γνωρίζουμε αν μια βαλβίδα έχει ανοιχτεί ή έχει κλείσει.

Οι μικροδιακόπτες χρησιμοποιούνται ευρέως. Μεταξύ των εφαρμογών τους είναι οι συσκευές, τα μηχανήματα, οι βιομηχανικοί έλεγχοι, τα οχήματα και πολλοί άλλοι χώροι για τον έλεγχο των ηλεκτρικών κυκλωμάτων. Συνήθως ταξινομούνται για να μεταφέρουν ρεύμα μόνο στα κυκλώματα ελέγχου, αν και μερικοί διακόπτες μπορούν να χρησιμοποιηθούν απευθείας για τον έλεγχο μικρών κινητήρων, σωληνοειδών, λαμπτήρων ή άλλων συσκευών. Ειδικές εκδόσεις χαμηλής ισχύος μπορούν να ανιχνεύσουν νομίσματα σε μηχανήματα αυτόματης πώλησης ή με πτερύγιο συνδεδεμένο με ροή αέρα.



Εικόνα 2.4.2 Τα εσωτερικά ενός μικροδιακόπτη. Επαφές, από αριστερά προς τα δεξιά, είναι κοινές, κανονικά ανοικτή (NO) και κανονικά κλειστή (NC).

2.5 Σερβοκινητήρας (Tower Pro 9g SG90)

Γενικά οι σερβοκινητήρες είναι AC ή DC κινητήρες που ο έλεγχος τους επιτυγχάνεται μέσω συστημάτων κλειστού βρόχου (closed loop) για γωνίες ελέγχου 0-180° ή για πλήρη περιστροφή. Ένα πλήρη σύστημα σερβομηχανισμού ελέγχου θέσης 0-180° περιλαμβάνει τα παρακάτω τμήματα:

- Κινητήρα AC ή DC,
- Μηχανικό σύστημα ταχύτητας (μειωτήρας),
- Μηχανικό ελεγκτή και
- Γραμμικό αισθητήρα αντίστασης (ποτενσιόμετρο).

Οι σερβοκινητήρες χρησιμοποιούνται σε εφαρμογές όπως ρομποτική, μηχανήματα CNC ή αυτοματοποιημένη κατασκευή.

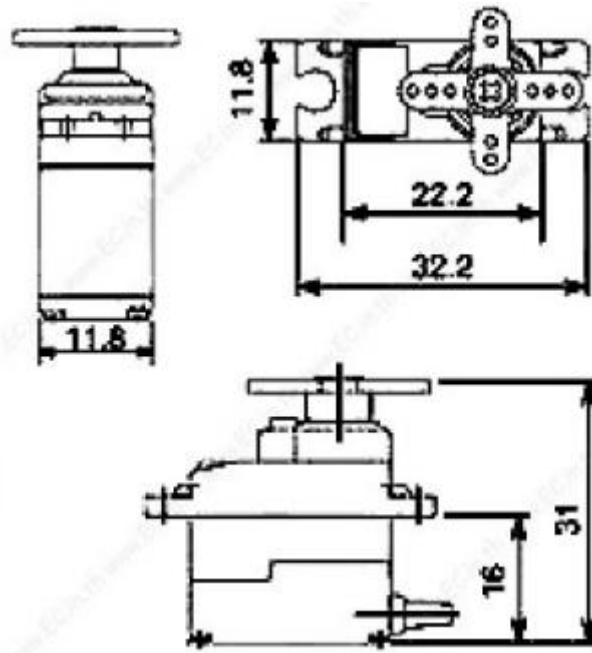
Στις παρακάτω δύο εικόνες έχουμε ένα μικρό σερβοκινητήρα Tower Pro 9g SG90, ο οποίος χρησιμοποιήθηκε και στην παρούσα πτυχιακή εργασία καθώς και το διάγραμμα λειτουργίας του.



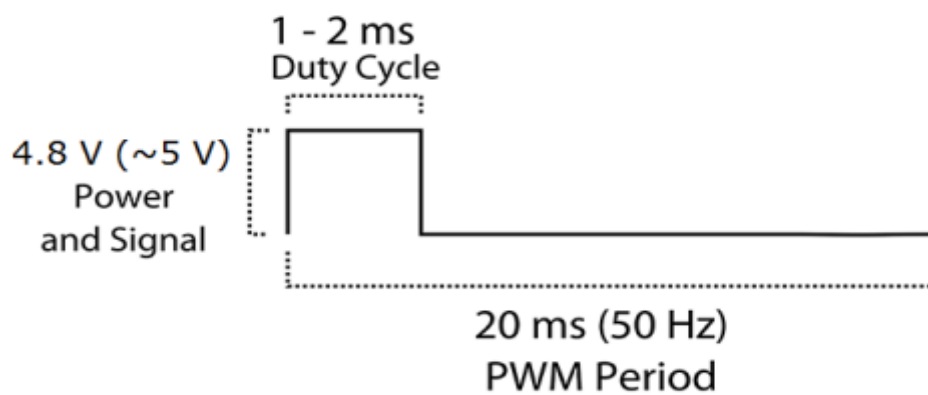
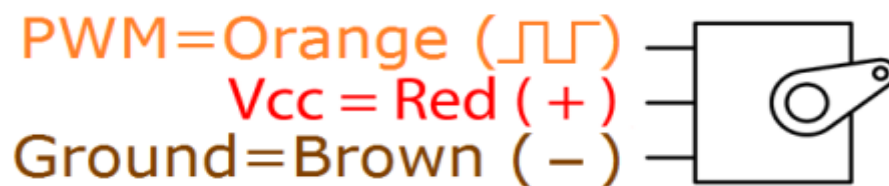
Εικόνα 2.5.1 Microservo Tower Pro 9g SG90.

Προδιαγραφές σερβοκινητήρα (Tower Pro 9g SG90)

- Βάρος: 9 g
- Διαστάσεις: 22,2 x 11,8 x 31 mm (περίπου)
- Ροπή στρέψης: 1,8 kgf · cm
- Ταχύτητα λειτουργίας: 0,1 s / 60 μοίρες
- Τάση λειτουργίας: 4,8 V (~ 5V)
- Εύρος νεκρής ζώνης: 10 μs
- Εύρος θερμοκρασίας: 0 °C - 55 °C



Εικόνα 2.5.2 Μηχανολογικό σχέδιο σερβοκινητήρα Tower Pro 9g SG90.



Εικόνα 2.5.3 Διάγραμμα λειτουργίας Microservo Tower Pro 9g SG90.

2.6 Τεχνολογία Bluetooth

Εισαγωγή

Η ανάγκη για μια ασύρματη ανταλλαγή δεδομένων σε κοντινή απόσταση, οδήγησε στην ιδέα της τεχνολογίας Bluetooth. Είναι ένα ανοικτό τεχνολογικό πρότυπο που επιτρέπει την ανταλλαγή δεδομένων ανάμεσα σε σταθερές και φορητές συσκευές που βρίσκονται σε μικρή απόσταση μεταξύ τους, δημιουργώντας ασφαλή μικρά τοπικά δίκτυα (piconets). Η τεχνολογία αυτή χρησιμοποιεί μικρού μήκους ραδιοσυχνότητες στο ISM (Industrial, Scientific and Medical) εύρος των 2400-2480 MHz. Αρχικός προορισμός του προτύπου Bluetooth ήταν να είναι μια ασύρματη εναλλακτική στα RS-232 καλώδια δεδομένων.

Ιστορική αναδρομή

Το 1994 η εταιρία ERICSSON εκπονεί μια μελέτη για μια χαμηλού κόστους και κατανάλωσης ασύρματη επικοινωνία των κινητών τηλεφώνων με τα εξαρτήματά τους, όπως επίσης και την εξάλειψη των καλωδίων στην σύνδεση των κινητών τηλεφώνων με τις κάρτες των H/Y σε dial up δίκτυα. Το 1998 ξεκινά η συνεργασία των εταιριών ERICSSON, INTEL, IBM, TOSHIBA και NOKIA με στόχο την ανάπτυξη τεχνολογίας που θα επιτρέπει στους χρήστες την εύκολη και ασύρματη επικοινωνία των συσκευών τους. Στις 20 Μαΐου δημιουργείται το Bluetooth SIG (Special Interest Group) με σκοπό τον σχεδιασμό και την υλοποίηση του προτύπου της τεχνολογίας Bluetooth. Από τότε πολλές άλλες εταιρίες έχουν ενταχθεί στο Bluetooth SIG, που πλέον αριθμεί πάνω από 20000 μέλη. Το 1999 δημοσιεύεται η έκδοση 1.0 των προδιαγραφών Bluetooth. Το 2000 κυκλοφορεί το πρώτο κινητό τηλέφωνο με τεχνολογία Bluetooth. Από τότε και στο εξής οι συσκευές που χρησιμοποιούν την τεχνολογία αυτή αυξήθηκαν κατακόρυφα, έχουν βγει στην αγορά τέσσερις εκδόσεις του πρωτοκόλλου Bluetooth με πιο πρόσφατη την έκδοση 4.1 που είναι αναβάθμιση της έκδοσης 4.0 με τεχνολογία χαμηλής ισχύος. Η τεχνολογία Bluetooth έχει εφαρμογές σε όλους τους τομείς, από κινητά τηλέφωνα, υπολογιστές, ιατρικά μηχανήματα και πολλά άλλα. Ένα αρκετά εύχρηστο χαρακτηριστικό της είναι ότι οι μεταγενέστερες εκδόσεις της έχουν σχεδιαστεί έτσι ώστε να έχουν συμβατότητα με εξαρτήματα και συσκευές που χρησιμοποιούν παλαιότερες εκδόσεις. Το 2005 το IEEE 802.15.1 όρισε τις προδιαγραφές, τα χαρακτηριστικά και την αρχιτεκτονική που πρέπει να υφίστανται ώστε να είναι εφικτή μια ασύρματη σύνδεση με τη χρήση της τεχνολογίας Bluetooth. Ενδεικτικά παρατίθενται τα εύρη μεταφοράς δεδομένων των τεσσάρων βασικών εκδόσεων του πρωτοκόλλου:

Έκδοση Bluetooth	Ρυθμός Μετάδοσης Δεδομένων
1.0	768Kbps
2.0	3Mbps
3.0	24Mbps
4.0	24Mbps

Εφαρμογές της τεχνολογίας Bluetooth

Με την τεχνολογική εξέλιξη που υπάρχει στις μέρες μας οι εφαρμογές του πρωτοκόλλου Bluetooth αυξάνονται συνεχώς. Παρέχει ασύρματη και αξιόπιστη επικοινωνία που είναι φιλική προς τον χρήστη και για αυτό το λόγο καταλαμβάνει όλο και πιο πολύ χώρο στην καθημερινή ζωή του ανθρώπου. Ενδεικτικά αναφέρονται κάποιες εφαρμογές όπως:

Ιατρικές συσκευές όπως στηθοσκόπια, πιεσόμετρα, μετρητές γλυκόζης και ζυγαριές που δίνουν την δυνατότητα να μεταφέρονται άμεσα τα δεδομένα των μετρήσεων σε κάποιον υπολογιστή ή κινητό τηλέφωνο και να δημιουργείται εύκολα ένα αρχείο παρακολούθησης του ασθενή.

Ο έλεγχος των λειτουργιών ενός "έξυπνου σπιτιού" μέσω του κινητού τηλεφώνου, όπως τη ρύθμιση της θερμοκρασίας ή του φωτισμού ενός χώρου, τον τηλεχειρισμό συσκευών οικιακής χρήσης κλπ. Η χαμηλού κόστους ασύρματη δικτύωση μεταξύ συσκευών όπως υπολογιστές, κινητά τηλέφωνα, ασύρματα ακουστικά, ηχεία, περιφερειακά υπολογιστών.

Πλεονεκτήματα και μειονεκτήματα της τεχνολογίας Bluetooth

Πλεονεκτήματα: Είναι ασύρματη και αυτόματη. Δε χρειάζονται καλώδια, μετατροπείς και συνδέσεις. Οι συσκευές αναγνωρίζονται και επικοινωνούν μεταξύ τους χωρίς τη βοήθεια του χρήστη, παρά μόνο σε ειδικές περιπτώσεις, π.χ. για την πληκτρολόγηση κωδικού πρόσβασης. Το κόστος κατασκευής του κυκλώματος είναι χαμηλό. Η ραδιοσυχνότητα λειτουργίας είναι μεν ελεγχόμενη, αλλά ελεύθερη. Οι ίδιες συσκευές μπορούν να χρησιμοποιηθούν σε οποιαδήποτε χώρα χωρίς προβλήματα. Η τεχνολογία επιτρέπει τη μετάδοση δεδομένων και φωνής. Οι εφαρμογές μπορούν να διαχειριστούν τα πλεονεκτήματα και των δύο μέσων μετάδοσης. Η ζεύξη είναι αμφίδρομη και περνά μέσα από τοίχους και η αναπήδηση συχνότητας (frequency hopping), ελαχιστοποιεί τον κίνδυνο διακοπής ή παρεμβολών στην επικοινωνία.

Μειονεκτήματα: Σχετικά χαμηλή ταχύτητα μετάδοσης δεδομένων. Η μέγιστη ταχύτητα που μπορεί να έχει μια σύνδεση μεταξύ δύο συσκευών μέσω Bluetooth είναι 1 Mbps, ενώ σε συνδέσεις τύπου Υπερύθρων ή Wi-Fi οι ταχύτητες είναι πιο μεγάλες. Μικρό εύρος λειτουργίας. Συνήθως οι συσκευές με Bluetooth έχουν εμβέλεια περίπου 10 μέτρα, πράγμα που καθιστά περιορισμό ως προς τη χρήση τους. Για παράδειγμα για να χρησιμοποιηθεί ένας εκτυπωτής με σύνδεση Bluetooth θα πρέπει να είναι σε ακτίνα 10 μέτρων από τη συσκευή του χρήστη που θα στείλει το αρχείο για εκτύπωση, ενώ με έναν εκτυπωτή σύνδεσης Wi-Fi η απόσταση είναι αρκετά μεγαλύτερη. Ασφάλεια. Παρά τις μεθόδους ασφαλείας που υπάρχουν, η χρήση ραδιοκυμάτων για μετάδοση των δεδομένων καθιστά λιγότερο ασφαλές ένα δίκτυο Bluetooth σε σχέση με άλλα πιο ασφαλέστερα ασύρματα δίκτυα. Χρήση ενέργειας. Το Bluetooth χρησιμοποιεί την μπαταρία των συσκευών στις οποίες είναι ενεργό, όπως για παράδειγμα κινητά τηλέφωνα, με αποτέλεσμα η μπαταρία να εξαντλείται γρηγορότερα.

Προδιαγραφές τεχνολογίας Bluetooth

Οι προδιαγραφές της τεχνολογίας Bluetooth ορίζουν τις αρχές λειτουργίας της με σκοπό να εξασφαλιστεί η συμβατότητα των συσκευών που τη χρησιμοποιούν από διαφορετικούς κατασκευαστές. Οι βασικές προδιαγραφές είναι οι εξής: Bluetooth Core Specifications, Bluetooth Transport Specifications, Bluetooth Protocols, Bluetooth Profiles.

Ασφάλεια τεχνολογίας Bluetooth

Μέθοδοι ασφάλειας

Αυθεντικοποίηση: Διασφαλίζει την ταυτότητα των συσκευών Bluetooth και επιτυγχάνεται όπως αναφέρθηκε παραπάνω με τη χρήση ενός αποθηκευμένου κλειδιού σύνδεσης ή με την επιβεβαίωση ενός κωδικού (PIN).

Επιβεβαίωση: Επικυρώνει δύο συσκευές βασισμένες σε ένα κοινό κωδικό (passkey) και με αυτόν τον τρόπο δημιουργεί μια σχέση εμπιστοσύνης μεταξύ εκείνων των συσκευών. Το ίδιο passkey πρέπει να εισαχθεί και στις δύο συσκευές. Η διαδικασία αυτή γίνεται μόνο την πρώτη φορά που θα συνδεθούν μεταξύ τους οι συγκεκριμένες συσκευές. Οι συσκευές που δεν έχουν κάποια μέθοδο εισαγωγής κωδικού, όπως τα ακουστικά, έχουν καθορισμένα passkeys.

Έγκριση: Ελέγχει εάν μια συσκευή επιτρέπεται να έχει πρόσβαση σε μια συγκεκριμένη υπηρεσία. Ο χρήστης μπορεί να ορίσει εάν η συσκευή είναι εμπιστεύσιμη ή όχι ώστε να γίνει ή όχι η σύνδεση. Σε αυτή τη μέθοδο απαιτείται πάντα η αυθεντικοποίηση της συσκευής.

Κρυπτογράφηση: Παρέχει προστασία στην επικοινωνία των χρηστών, από υποκλοπές. Το μήκος του κλειδιού κρυπτογράφησης μπορεί να είναι μεταξύ 8 και 128 bits.

Επίπεδα εμπιστοσύνης συσκευής

- Συσκευή “εμπιστοσύνης”: Έχει γίνει η αυθεντικοποίηση αυτής της συσκευής σε παλιότερη σύνδεση, έχει αποθηκεύει ένα κλειδί σύνδεσης και η συσκευή είναι χαρακτηρισμένη ως “εμπιστεύσιμη” στη βάση δεδομένων ασφάλειας της άλλης συσκευής. Έτσι η συσκευή μπορεί να έχει πρόσβαση στις υπηρεσίες Bluetooth χωρίς την αποδοχή του χρήστη.
- “Μη εμπιστεύσιμη” συσκευή: Έχει γίνει η αυθεντικοποίηση αυτής της συσκευής σε παλιότερη σύνδεση, έχει αποθηκεύει ένα κλειδί σύνδεσης, αλλά δεν έχει χαρακτηριστεί “εμπιστεύσιμη”. Η πρόσβαση στις υπηρεσίες απαιτεί την αποδοχή του χρήστη.
- “Άγνωστη” συσκευή: Μια συσκευή θεωρείται άγνωστη όταν δεν υπάρχει καμία πληροφορία ασφαλείας για αυτήν. Αυτή η συσκευή συχνά χαρακτηρίζεται και ως “μη εμπιστεύσιμη”.

Επίπεδο ασφαλείας υπηρεσιών

Αίτηση έγκρισης: Η πρόσβαση γίνεται αυτόματα μόνο για μια “εμπιστεύσιμη” συσκευή ή μιας “μη εμπιστεύσιμης” μετά από μια διαδικασία έγκρισης. Σε αυτό το επίπεδο απαιτείται πάντα η αυθεντικοποίηση της συσκευής.

Αίτηση για αυθεντικοποίηση: Πρέπει να γίνει η επιβεβαίωση της αυθεντικότητας μιας συσκευής πριν αυτή συνδεθεί με την υπηρεσία.

Αίτηση για κρυπτογράφηση: Η σύνδεση πρέπει να κρυπτογραφηθεί προτού να δοθεί πρόσβαση στην υπηρεσία.

Καμία αίτηση: Υπάρχει η πιθανότητα κάποια υπηρεσία να μην χρησιμοποιεί κάποιο από τα προαναφερθέντα επίπεδα, αλλά να έχει τους δικούς της κανόνες ασφαλείας.

Αρχιτεκτονική Bluetooth Core

Η αρχιτεκτονική του πυρήνα του Bluetooth πρότυπου χωρίζεται σε δύο βασικά μέρη: Το Controller Stack (φυσικό μέρος της σύνδεσης, hardware) που περιέχει τα υλικά μέρη και την ραδιοσυχνότητα της σύνδεσης, όπως και τα αντίστοιχα χαμηλού επιπέδου πρωτόκολλα: Bluetooth Radio, Baseband, LMP και το Host Stack (λογισμικό μέρος της σύνδεσης, software): που περιέχει το λειτουργικό σύστημα ή κάποια άλλη εφαρμογή από την οποία δέχεται εντολές για την λειτουργία του, αλλά και τα υψηλότερου επιπέδου πρωτόκολλα: L2CAP, TCS BIN, SDP, RFCOMM. Το Controller Stack επικοινωνεί με το Host Stack μέσω του Host Controller Interface (HCI). Αυτή η διεπαφή παρέχει ένα περιβάλλον εργασίας και δίνει την δυνατότητα ελέγχου τόσο του φυσικού μέρους όσο και του λογισμικού, μέσω εντολών.

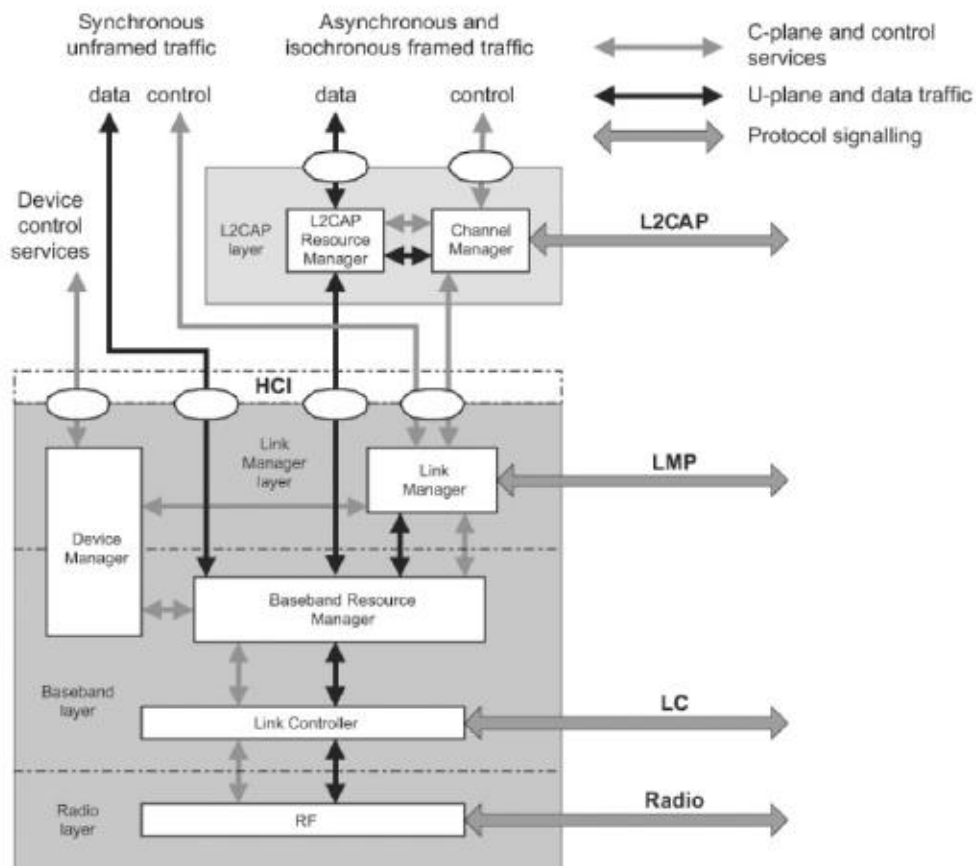


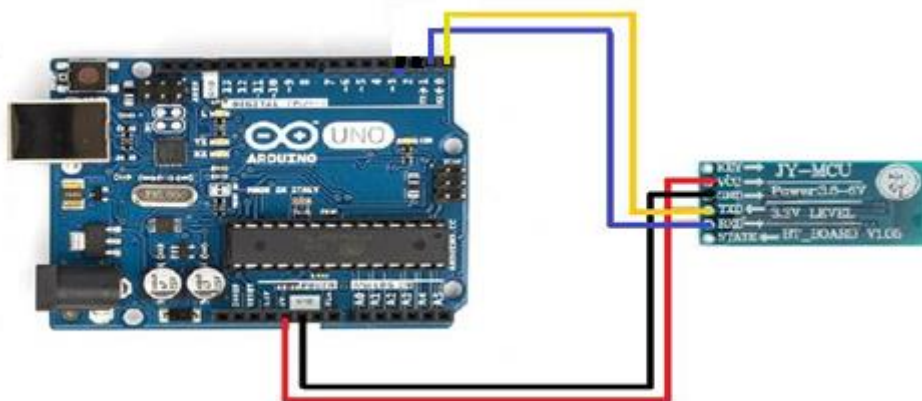
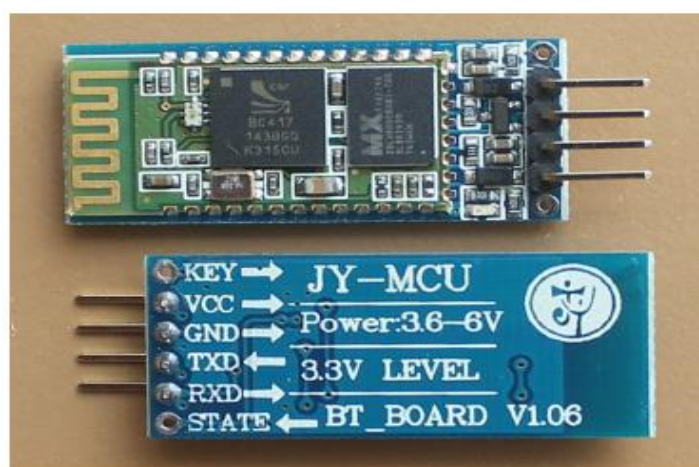
Figure 1—Core system architecture

2.7 Bluetooth Διάταξη (Module HC-06)

Μια από τις βασικές ιδιότητες του έντροχου ρομποτικού συστήματος είναι η απομακρυσμένη σύνδεση του με άλλα προγράμματα καθώς και συσκευές (π.χ Smartphone, Tablet). 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 Smartphone οπότε και κάνουμε την επεξεργασία των δεδομένων. Η τάση λειτουργίας του είναι τα 5Vcc 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 πρώτη εικόνα φαίνεται η διάταξη του Bluetooth HC-06 καθώς και στην δεύτερη εικόνα φαίνεται ο τρόπος σύνδεσης του με το Arduino UNO.

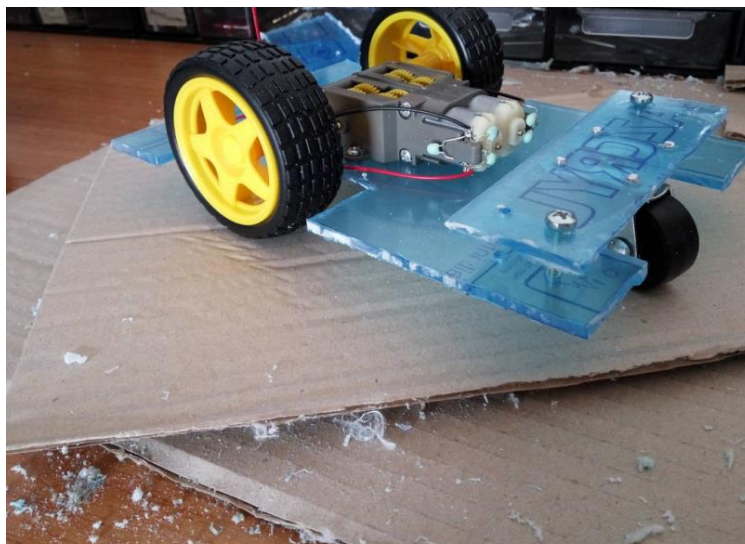


ΚΕΦΑΛΑΙΟ 3^ο

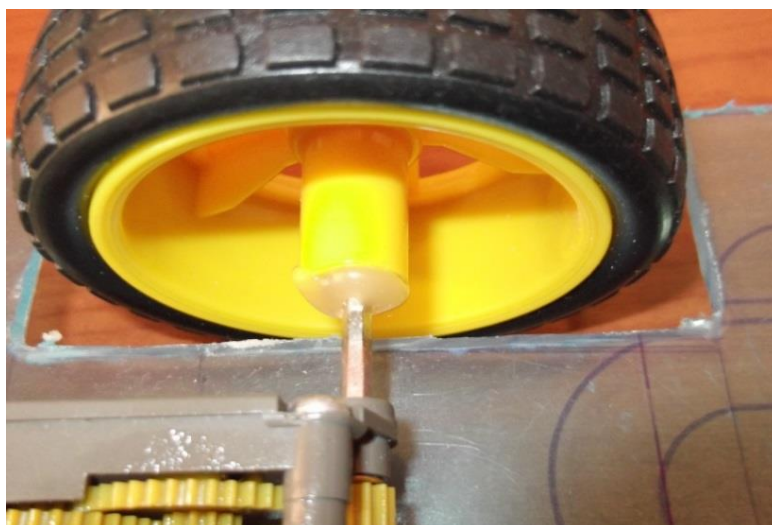
ΚΑΤΑΣΚΕΥΑΣΤΙΚΟ ΜΕΡΟΣ

3.1 Εισαγωγή

Στην παρούσα πτυχιακή εργασία, για την κατασκευή σασί του ρομποτικού οχήματος χρησιμοποιήθηκαν ακριλικές πλάκες (plexiglass) μεγέθους 30x30mm και πάχους 5mm. Αυτό αρχικά μας δημιούργησε το πρώτο από τα πολλά προβλήματα της εργασίας καθώς υπήρχε δυσκολία στην κοπή αλλά και την διαμόρφωση του υλικού. Επίσης, η ασταθής και κακής ποιότητας σασί που χρησιμοποιήθηκε στη συνέχεια αντικαταστάθηκε από την Robot Smart Car 2wd kit ως πιο οικονομική και σταθερή. Ακολουθούν οι παρακάτω εικόνες για την κατανόηση του αρχικού προβλήματος:



Εικόνα 3.1.1 Πρόβλημα κοπής αρχικού σασί ρομποτικού οχήματος.



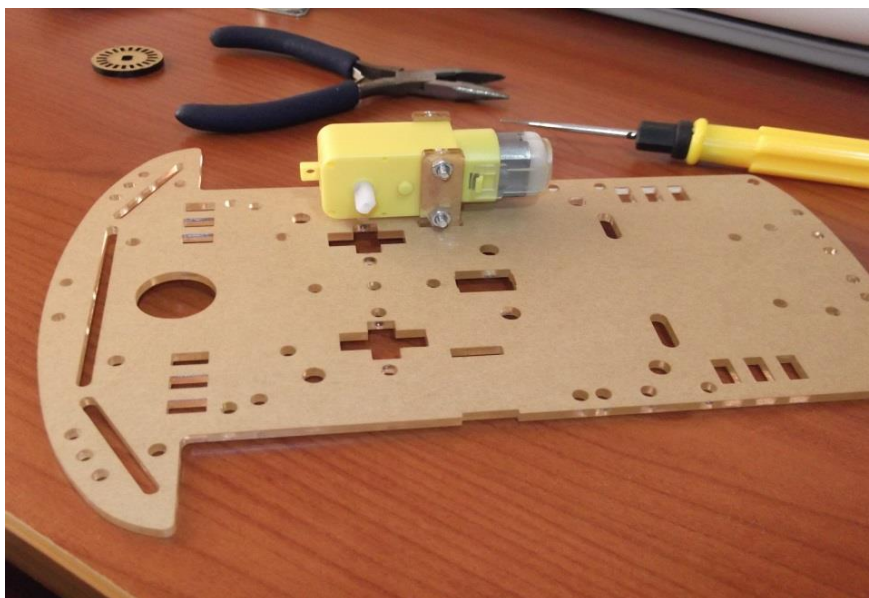
Εικόνα 3.1.2 Αστάθεια τροχού πάνω στον αρχικό άξονα.

3.2 Υλικά Κατασκευής

- Arduino UNO και το καλώδιο usb διασύνδεσης του.
- Μικρό breadboard και καλώδια (male to male και female to male).
- Bluetooth module HC-06.
- Ρομποτικό σασί με 2 τροχούς – κινητήρες και περιστροφικό ροδάκι (360°).
- Αισθητήρας υπερήχων HC-SR04.
- Micro SG90 9g servo-μοτέρ (180°) μαζί με βάση για τον αισθητήρα HC-SR04 και βάση για το servo.
- L298N Motor Driver Module.
- Θήκη για μπαταρίες 18650 3,7V και για 9V μπαταρία.
- Βύσμα για το DC Power Jack του Arduino UNO.
- 2 μικροδιακόπτες "επαφής" .
- Διακόπτης μικρός ON-OFF για την μπαταρία 9V.
- 2 κεραμικούς πυκνωτές 100 nF.
- Αποστάτες - βίδες - παξιμάδια για να τοποθετήσουμε τα εξαρτήματα πάνω στο ρομποτικό σασί.
- Θερμοσλικόνη και πιστόλι θερμοσλικόνης.

3.3 Υλοποίηση κατασκευής

Όπως προαναφέρθηκε στην παράγραφο 3.1 το παλιό ασταθές σασί του ρομποτικού οχήματος αντικαταστάθηκε με το Robot Smart Car 2wd kit, το οποίο περιλαμβάνει μια ακριλική πλάκα 22 x 14.7 cm, δύο τροχούς 65mm, ένα ροδάκι περιστροφής 360° και δύο DC κινητήρες οι οποίοι είχαν τέλεια εφαρμογή με τους τροχούς και με αυτόν τον τρόπο αντιμετωπίστηκε η αστάθεια σε μεγάλο βαθμό.



Εικόνα 3.3.1 DC κινητήρας και ρομποτικό σασί του οχήματος.



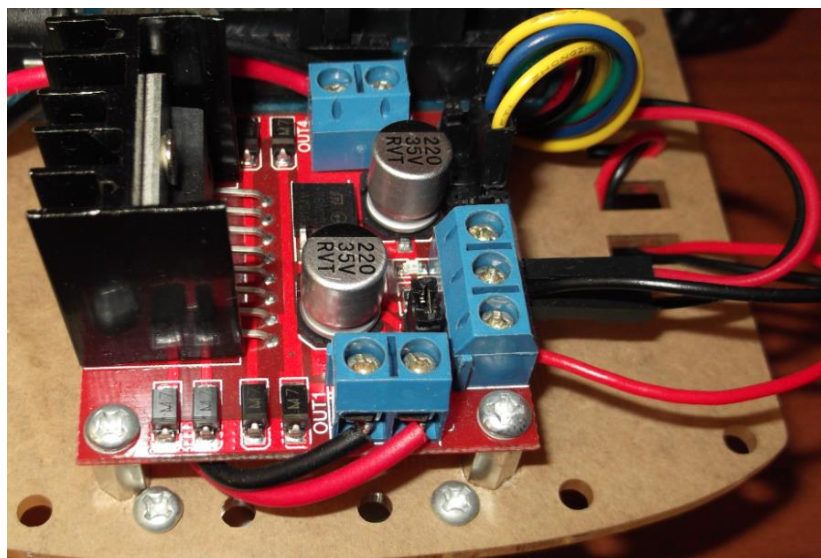
Εικόνα 3.3.2 Κινητήρας μαζί με τον τροχό του ρομποτικού οχήματος.

Για τον έλεγχο των DC κινητήρων χρησιμοποιήθηκε το L298N Motor Driver Module το οποίο αναλύεται πλήρως στο Κεφάλαιο 2.2. Η τοποθέτηση του πάνω στο ρομποτικό σασί έγινε με την αγορά κατάλληλων αποστατών ύψους 10x3mm, το οποίο απαιτούσε επιπλέον άνοιγμα οπών στο σασί.



Εικόνα 3.3.3 Άνοιγμα οπών στο ρομποτικό σασί με την βοήθεια Dremel.

Στην συνέχεια, έγινε η σύνδεση των μοτέρ με το Dual 'H'-Bridge L298N όπου το κάθε μοτέρ είχε την δικιά του γέφυρα και συνδέθηκε στις κατάλληλες κλέμες (OUT1-OUT2) και το άλλο μοτέρ στην άλλη γέφυρα (OUT3-OUT4).

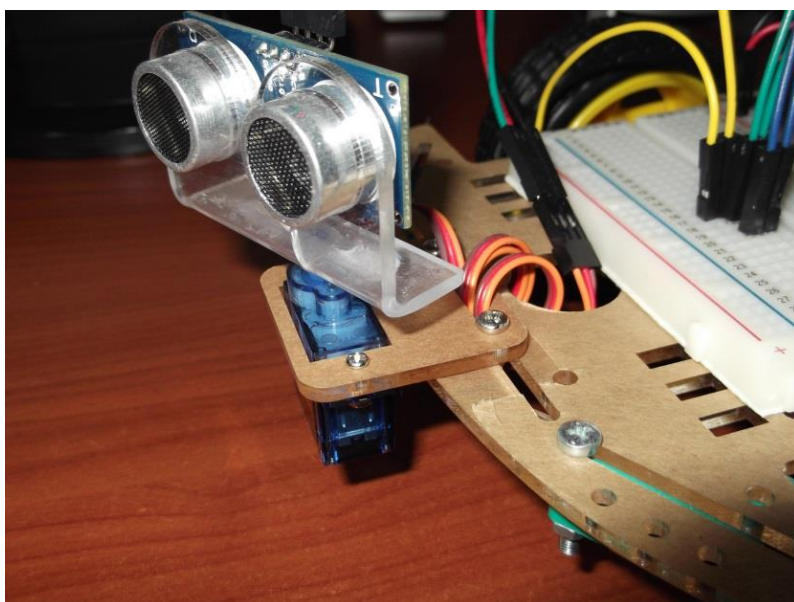


Εικόνα 3.3.4 Συνδέσεις κινητήρων, τροφοδοσίας και σημάτων ελέγχου στο Dual H-Bridge L298N.

Ακριβώς δίπλα από το L298N τοποθετήθηκε η αναπτυξιακή πλακέτα Arduino UNO (άνοιγμα οπών και τοποθέτηση αποστατών), πιο δίπλα κολλήθηκε το mini breadboard επάνω στο ρομποτικό σασί για να του δοθεί τάση και γείωση από το Arduino ώστε να μπορούν να τροφοδοτηθούν τα περιφερειακά μας (ultrasonic, servomotor, Bluetooth). Συνδέθηκε το Arduino UNO επιλέγοντας τέσσερα PWM pin τα οποία οδηγήθηκαν στις εισόδους του L298N με σκοπό τον έλεγχο στροφών των δύο κινητήρων. Για την τροφοδοσία του Arduino επιλέχθηκε μια 9V μπαταρία, μιας που για την λειτουργία του απαιτεί από 7-12V και συνδέθηκε σε ένα διακόπτη ON-OFF δυο επαφών, καθώς και για την ξεχωριστή τροφοδοσία του L298N τοποθετήθηκαν 4xAA επαναφορτιζόμενες μπαταρίες 1.2V. Βέβαια κατόπιν δοκιμών φάνηκε ότι δεν ήταν αρκετές και έτσι μετά έγινε η επιλογή δυο επαναφορτιζόμενων μπαταριών 18650 3.7V 3400mAh οι οποίες έδιναν μεγάλη αυτονομία στο ρομποτικό όχημα. Μετέπειτα έγινε ο έλεγχος λειτουργίας των κινητήρων μέσω προγράμματος στο γνωστό περιβάλλον Arduino IDE.

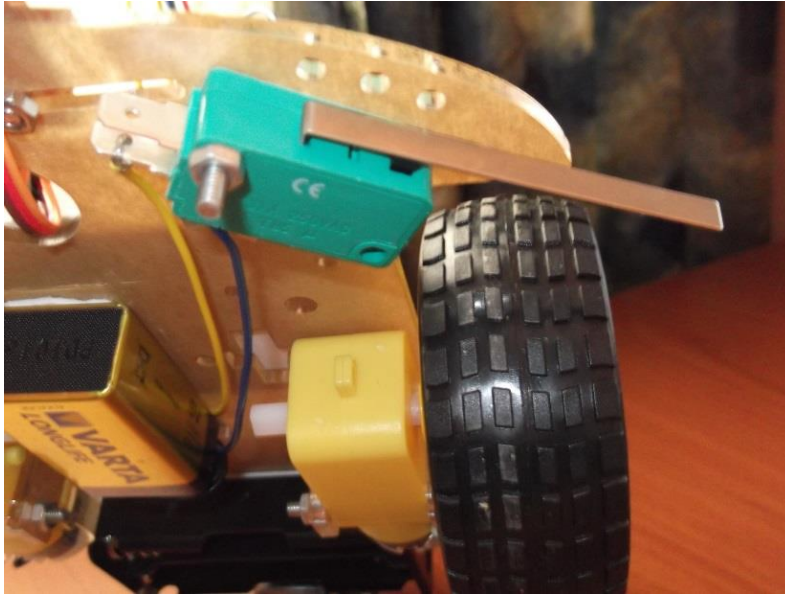
Στο επόμενο στάδιο τοποθετήθηκε με χρήση θερμοσηλικόνης ο σερβοκινητήρας επάνω στο σασί και επάνω του μπήκε η βάση του αισθητήρα υπερήχων. Μετά τροφοδοτήθηκε ο σερβοκινητήρας με ρεύμα και συνδέθηκε με το Arduino με ένα PWM pin για τον έλεγχο των κινήσεων του (0-180°), όπως επίσης έγινε και η τροφοδοσία του αισθητήρα υπερήχων αλλά και η επικοινωνία του με το Arduino μέσω δύο αναλογικών pin αυτού.

Στην συνέχεια προγραμματίστηκε ο μικροελεγκτής μας, με τα επιμέρους περιφερειακά ώστε το ρομποτικό μας όχημα να κινείται αυτόνομα αποφεύγοντας εμπόδια. Από τις πρώτες παρατηρήσεις που έγιναν διαπιστώθηκε ότι ο αισθητήρας ήταν σε μεγάλο ύψος με αποτέλεσμα να μην ανιχνεύει τυχόν εμπόδια. Αποφασίστηκε να γίνει αγορά βάσης για την καλύτερη τοποθέτηση του σερβοκινητήρα ώστε να μειωθεί το ύψος και το ρομποτικό όχημα να μπορεί να ανιχνεύσει περισσότερα εμπόδια στον δρόμο του.



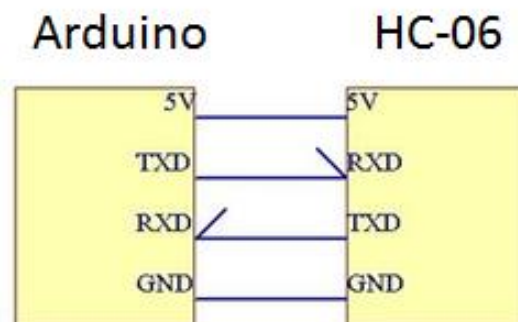
Εικόνα 3.3.5 Η βάση του σερβοκινητήρα μαζί με το Ultrasonic Sensor HC-SR04.

Πέραν αυτού για να καλυφθεί ένα ποσοστό των μη ανιχνεύσιμων γωνιών, χρησιμοποιήθηκαν δύο μικροδιακόπτες επαφής με τέτοιο τρόπο ώστε το άκρο του (σίδηρο) να προεξέχει από τις ρόδες. Ο λόγος είναι όταν το όχημα μας χτυπάει πάνω σε κάποιο αντικείμενο που βρίσκεται κοντά στις ρόδες και δεν έχει γίνει αντιληπτό από τον αισθητήρα, να ενεργοποιούνται οι μικροδιακόπτες και να αλλάζει κατεύθυνση. Ο μικροδιακόπτης έχει τρεις επαφές Common, Normaly Open, Normaly Close. Εμείς χρησιμοποιήσαμε τις δύο επαφές του, το Common συνδέθηκε με την γείωση (GND) του Arduino, ενώ οι επαφές Normaly Open σε δυο ψηφιακά pin του Arduino.



Εικόνα 3.3.6 Μικροδιακόπτης επαφής στην άκρη του οχήματος και κοντά στους τροχούς.

Ένα μεγάλο μέρος της πτυχιακής εργασίας αποτέλεσε το Bluetooth. Μέσω του Bluetooth καταφέραμε να ελέγξουμε την πορεία του ρομποτικού οχήματος με την χρήση Android συσκευής. Χρησιμοποιήσαμε ένα σειριακό Bluetooth module HC-06 το οποίο τοποθετήθηκε πάνω στο breadboard και ξεκίνησε η διαδικασία διασύνδεσης του με το Arduino. Τροφοδοτήθηκε με 5V από το breadboard όπως και GND και η σύνδεση του με το Arduino ήταν σύνδεση αλληλεπίδρασης (το Receive του Bluetooth με το Transmit του Arduino και το αντίστροφο). Ακόμα ένα από τα προβλήματα που συναντήσαμε λόγω άγνοιας ήταν ότι όταν περνάμε πρόγραμμα στο Arduino θα πρέπει τα RXD και TXD του Bluetooth να είναι αποσυνδεδεμένα από το Arduino board.

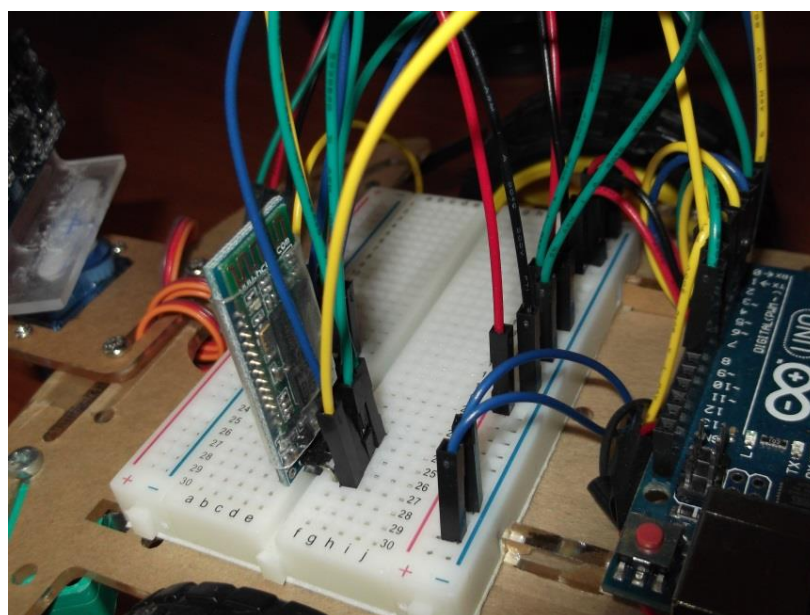


Εικόνα 3.3.7 Διασύνδεση Arduino με το Bluetooth module HC-06.

Για τον προγραμματισμό του Bluetooth module HC-06 χρειάστηκε να περαστεί στον μικροεπεξεργαστή το έτοιμο παράδειγμα "Blink led" από το Arduino IDE αφού πρώτα συνδέθηκε ένα led στα pin 13 (άνοδος) και GND (κάθοδος) αντίστοιχα. Με αυτόν τον τρόπο βεβαιωνόμαστε ότι ο μικροεπεξεργαστής δεν χρησιμοποιεί και αυτός την σειριακή θύρα.

Τα βήματα προγραμματισμού που ακολουθήσαμε ήταν τα εξής:

- Συνδέσαμε το RXD του Bluetooth με το RXD του Arduino και το TXD του Bluetooth με το TXD του Arduino.
- Συνδέσαμε το Vcc του Bluetooth με τα 5V του Arduino (breadboard) και το GND του Bluetooth με το GND του Arduino (breadboard).
- Συνδέσαμε το KEY με μια αντίσταση $\sim 10K\Omega$ στο Vcc του Bluetooth.
- Αφού συνδέσαμε το Arduino με τον Η/Υ ανοίξαμε την επικοινωνία στην σειριακή οθόνη του Arduino IDE.
- Επιλέγουμε ο σωστός ρυθμός μετάδοσης (baud rate) που έχει ήδη το bluetooth module (9600 bps).
- Πληκτρολογήσαμε AT , πήραμε απάντηση OK και είμασταν σε θέση να προγραμματίσουμε το Bluetooth.
- Για αλλαγή ονόματος πληκτρολογήσαμε AT+NAMEκιου_κιου. Το Bluetooth απάντησε OKsetname. Αλλάξαμε και τον κωδικό πρόσβασης χρησιμοποιώντας την εντολή AT+PIN****. Το Bluetooth απάντησε OKsetpin.
- Αποσυνδέσαμε το KEY από την αντίσταση και από το Vcc και το Bluetooth είναι έτοιμο με νέα χαρακτηριστικά.



Εικόνα 3.3.8 Το Bluetooth module HC-06 πάνω στο breadboard και συνδεδεμένο με το Arduino UNO.

Αφού προγραμματίσαμε επιτυχώς το Bluetooth, κατεβάσαμε από το Google Playstore την εφαρμογή ώστε να έχουμε απομακρυσμένο έλεγχο του ρομποτικού οχήματος και στην συνέχεια περάσαμε κώδικα ώστε να βεβαιωθούμε ότι όλα λειτουργούσαν ορθά. Το ρομποτικό όχημα μέσω της εφαρμογής πραγματοποιεί τις εξής κινήσεις στον χώρο:

- Εμπρός (Forward).
- Πίσω (Backward).
- Δεξιά (Right).
- Αριστερά (Left).



Εικόνα 3.3.9 Η εφαρμογή για ασύρματη επικοινωνία μέσω smartphone.

Από την παραπάνω εφαρμογή μας δόθηκε η δυνατότητα τηλεχειρισμού με εμβέλεια περίπου 10m του ρομποτικού οχήματος αλλά και να μπορούμε να το θέσουμε σε αυτόματη λειτουργία (Automatic mode), με το πάτημα ενός κουμπιού (μέσα από την εφαρμογή). Η Android εφαρμογή ονομάζεται "Arduomotive BT Controller" (developer_MichalisVasilakis).

ΚΕΦΑΛΑΙΟ 4^ο

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΜΕΡΟΣ

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα εστιάσουμε στον προγραμματισμό του ρομποτικού συστήματος, προκειμένου να οδηγήσουμε και να συντονίσουμε όλες τις λειτουργίες του. Το προγραμματιστικό περιβάλλον που χρησιμοποιήθηκε είναι το Arduino IDE το οποίο αναλύεται στο Κεφάλαιο 2.1. Αρχικά, δημιουργήθηκε κώδικας ώστε το ρομποτικό μας όχημα να κινείται αυτόνομα στο χώρο και να αποφεύγει τυχών εμπόδια. Έπειτα από πειράματα καταλήξαμε στη δημιουργία κώδικα σε γλώσσα C/C++ για να έχει το όχημα μας εκτός από την αυτόματη λειτουργία (Automatic mode) και χειροκίνητη λειτουργία (Manual mode). Ο συγκεκριμένος κώδικας διαμορφώθηκε σύμφωνα με την έτοιμη εφαρμογή του Google Playstore και συμπεριλήφθηκαν οι έτοιμες βιβλιοθήκες του Arduino IDE (Ultrasonic.h και Servo.h). Ακολουθεί ο κώδικας διπλής λειτουργίας του οχήματος (Dual Function) με αναλυτικά σχόλια:

```
// Arduino Robot Project: Dual function

//Library
#include "Ultrasonic.h"

#include <Servo.h>

//Constants

const int motorR1 = 6;

const int motorR2 = 5;

const int motorL1 = 11;

const int motorL2 = 10;

const int servo = 3;

const int rightSW = 8;

const int leftSW = 7;

Ultrasonic ultrasonic(A0,A1); // Trig, Echo

Servo myservo;
```

```

//Variables

int distance=100;

int rightDistance;

int leftDistance;

int rightSWincomingData;

int leftSWincomingData;

int pos = 90; // for servo, 90 deg looking forward

int incomingData;

char mode = 'm'; // m for manual, a for automatic

int vSpeed = 200; //default pwm value for motors speed

void setup(){

    //Define inputs and outputs

    pinMode(motorR1, OUTPUT);

    pinMode(motorR2, OUTPUT);

    pinMode(motorL1, OUTPUT);

    pinMode(motorL2, OUTPUT);

    pinMode(leftSW, INPUT_PULLUP);

    pinMode(rightSW, INPUT_PULLUP);

    //Init servo and starting position

    myservo.attach(servo);

    myservo.write(pos);

    //Start serial communication

    Serial.begin(9600);

}

```



```

void loop(){
    if(Serial.available() > 0){
        // read the incoming byte:
        incomingData = Serial.read();
        if (incomingData == 'A'){mode = 'a'; }
        if (incomingData == 'M'){mode = 'm';
            stop();
            pos=90;
            myservo.write(pos);
        }
    }
}

//Change speed if incomingData is equal from 0 to 4. Values must be from 0 to 255 (PWM)
if (incomingData == '0'){
    vSpeed=0;}
else if (incomingData == '1'){
    vSpeed=100;}
else if (incomingData == '2'){
    vSpeed=180;}
else if (incomingData == '3'){
    vSpeed=200;}
else if (incomingData == '4'){
    vSpeed=255;}

```

```
//Manual mode
```

```
if (mode == 'm'){  
    if (incomingData=='F'){  
        forward();  
    }  
    else if (incomingData=='B'){  
        backward();  
    }  
    else if (incomingData=='R'){  
        right();  
    }  
    else if (incomingData=='L'){  
        left();  
    }  
    else if (incomingData=='S'){  
        stop();  
    }  
}
```

```
//Automatic mode
```

```
else if (mode == 'a'){  
    distance = ultrasonic.Ranging(CM);  
  
    //If an object detected at 10cm, stop the robot and find a way out  
    if (distance <= 10){  
        stop();  
  
        //Look write  
        for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to  
            180 degrees  
        }  
  
        myservo.write(pos); // tell servo to go to position in variable 'pos'
```

```

delay(5); // waits 5ms for the servo to reach the position
}

//Read new distance from the right side
leftDistance = ultrasonic.Ranging(CM);
delay(100);

//Now look left
                for (pos = 180; pos >= 0; pos -= 1) { // goes from 180
degrees to 0 degrees

myservo.write(pos); // tell servo to go to position in variable 'pos'
delay(5); // waits 5ms for the servo to reach the position
}

//Read new distance from the left side
rightDistance = ultrasonic.Ranging(CM);
delay(100);

pos = 90; // look forward again

myservo.write(pos);

//Finally compare left and right distances and make the best turn decision
if (leftDistance > rightDistance){
    left();

    delay(250); // Change the time to make 90 deg.turn
}

else if (leftDistance < rightDistance){
    right();

    delay(250);
}

else{ //that means that two distances are equal

    backward();

    delay(1000); // Go back for 1 sec

```

```

        left(); // and turn left
        delay(250);
    }
}

//All clear, move forward! And check left and right "contact" switches.
else{
    leftSWincomingData = digitalRead(leftSW);
    rightSWincomingData = digitalRead(rightSW);
    //Check for "contact"
    if (leftSWincomingData == LOW){
        backward();
        delay(500);
        right();
        delay(250);
    }
    if (rightSWincomingData == LOW){
        backward();
        delay(500);
        left();
        delay(250);
    }
    forward();
}
}
}

```

```
//Movement functions
```

```
void forward(){  
    analogWrite(motorR1, vSpeed);  
    analogWrite(motorR2, 0);  
    analogWrite(motorL1, vSpeed);  
    analogWrite(motorL2, 0);  
}
```

```
void backward(){  
    analogWrite(motorR1, 0);  
    analogWrite(motorR2, vSpeed);  
    analogWrite(motorL1, 0);  
    analogWrite(motorL2, vSpeed);  
}
```

```
void right(){  
    analogWrite(motorR1, 0);  
    analogWrite(motorR2, vSpeed);  
    analogWrite(motorL1, vSpeed);  
    analogWrite(motorL2, 0);  
}
```

```
void left(){  
    analogWrite(motorR1, vSpeed);  
    analogWrite(motorR2, 0);  
    analogWrite(motorL1, 0);  
    analogWrite(motorL2, vSpeed);  
}
```

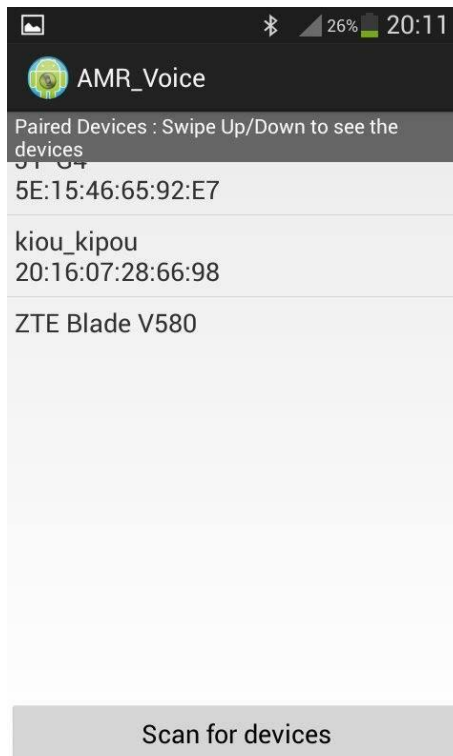
```
void stop(){  
  
    analogWrite(motorR1, 0);  
  
    analogWrite(motorR2, 0);  
  
    analogWrite(motorL1, 0);  
  
    analogWrite(motorL2, 0);  
  
}
```

4.2 Κίνηση του ρομποτικού οχήματος μέσω φωνητικών εντολών

Στο τελευταίο μέρος της πτυχιακής μας εργασίας αφού φτάσαμε σε ικανοποιητικό επίπεδο επεκτείναμε την λειτουργία του ρομποτικού οχήματος βάζοντας και έλεγχο μέσω φωνητικών εντολών. Αυτό έγινε με την βοήθεια μιας δεύτερης Android εφαρμογής για να κινήσουμε το ρομποτικό όχημα. Η εφαρμογή αυτή λέγεται "BT Voice Control for Arduino" και την κατεβάσαμε δωρεάν από το Google Playstore (developer SimpleLabsIN).



Εικόνα 4.2.1 Πρώτη εικόνα εφαρμογής φωνητικών εντολών μέσω Bluetooth.



Εικόνα 4.2.2 Επιλογή του ρομποτικού οχήματος μέσω του ονόματος που προγραμματίστηκε kίου_κίρου.

Η παραπάνω εφαρμογή των φωνητικών εντολών στέλνει σειριακά τις λέξεις στο Arduino. Οι λέξεις ξεκινάνε με '*' και τελειώνουν με '#'. Ο Arduino "καταλαβαίνει" μόνο λατινικούς χαρακτήρες, οι λέξεις για τον έλεγχο του ρομποτικού οχήματος πρέπει να είναι στα Αγγλικά. Οι εντολές που καταλαβαίνει το Arduino είναι:

- Forward - *Forward# (πηγαίνει μπροστά για 1 δευτερόλεπτο)
- Backward - *Backward# (πηγαίνει πίσω για 1 δευτερόλεπτο)
- Left - *Left# (πηγαίνει αριστερά 90°)
- Right - *Right# (πηγαίνει δεξιά 90°)
- Stop - *Stor# (Σταματάει)

Μερικές φορές λόγω προφοράς δεν στέλνεται σωστά μια εντολή, για να αποφευχθεί αυτό το πρόβλημα μπορούμε να χρησιμοποιήσουμε εναλλακτικές λέξεις για παράδειγμα αντί για Backward, να πούμε Reverse.

Παρακάτω, ακολουθεί ο κώδικας λειτουργίας μέσω φωνητικών εντολών του οχήματος (BT Voice Control) με αναλυτικά σχόλια:

```
// Arduino Robot Project: Bluetooth Voice Controlled Function
```

```
//Constants
```

```
const int motorR1 = 6;
```

```
const int motorR2 = 5;
```

```
const int motorL1 = 11;
```

```
const int motorL2 = 10;
```

```
//Variables
```

```
char incomingByte;
```

```
String command;
```

```
void setup()
```

```
{
```

```
    //Define inputs and outputs
```

```
    pinMode(motorR1, OUTPUT);
```

```
    pinMode(motorR2, OUTPUT);
```

```
    pinMode(motorL1, OUTPUT);
```

```
    pinMode(motorL2, OUTPUT);
```

```
    //Start serial communication
```

```
    Serial.begin(9600);
```

```
}
```

```
void loop()
```

```
{
```

```
    while(Serial.available()) {
```

```
        incomingByte = Serial.read();
```

```
        command.concat(incomingByte);
```

```
    }
```


//If voice command is equal with 'forward', start moving forward

```
if (command == "*forward#" || command == "*Forward#") {  
    forward();  
    delay(1000); //forward for 1 sec  
    stop();  
}  
  
else if (command == "*backward#" || command == "*Backward#" || command==  
"*reverse#" || command == "*Reverse#"){  
    backward();  
    delay(1000); //backward for 1 sec  
    stop();  
}  
  
//Turn 90 deg left and stop.  
  
else if (command == "*left#" || command == "*Left#") {  
    left();  
    delay(250); // Turn 90 deg left  
    stop();  
}  
  
else if (command == "*right#" || command == "*Right#" || command == "*write#") {  
    right();  
    delay(250); // Turn 90 deg right  
    stop();  
}  
  
else if (command == "*stop#" || command == "*Stop#") {  
    stop();  
}
```

```
//Wait for new command

if (incomingByte == '#'){
    command="";
}

delay(100); //small delay
}

//Movement functions

void forward(){
    digitalWrite(motorR1, HIGH);
    digitalWrite(motorR2, LOW);
    digitalWrite(motorL1, HIGH);
    digitalWrite(motorL2, LOW);
}

void backward(){
    digitalWrite(motorR1, LOW);
    digitalWrite(motorR2, HIGH);
    digitalWrite(motorL1, LOW);
    digitalWrite(motorL2, HIGH);
}

void right(){
    digitalWrite(motorR1, LOW);
    digitalWrite(motorR2, HIGH);
    digitalWrite(motorL1, HIGH);
    digitalWrite(motorL2, LOW);
}
```

```
void left(){
    digitalWrite(motorR1, HIGH);
    digitalWrite(motorR2, LOW);
    digitalWrite(motorL1, LOW);
    digitalWrite(motorL2, HIGH);
}
void stop(){
    digitalWrite(motorR1, LOW);
    digitalWrite(motorR2, LOW);
    digitalWrite(motorL1, LOW);
    digitalWrite(motorL2, LOW);
}
```

ΚΕΦΑΛΑΙΟ 5^ο

Δυνατότητες επέκτασης - αναβάθμισης του έντροχου ρομποτικού συστήματος και συμπεράσματα

5.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα καταγράψουμε κάποια συμπεράσματα σχετικά με την κατασκευή του έντροχου 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 και με την χρήση περισσότερων αισθητήρων να μπορούμε να συλλέγουμε και να επεξεργαζόμαστε περισσότερα δεδομένα.

5.2 Συμπεράσματα πτυχιακής εργασίας

Η αρχική ιδέα της πτυχιακής εργασίας ξεκίνησε στην κατασκευή ενός μη επανδωμένου έντροχου ρομποτικού συστήματος αυτόματης κίνησης που θα κινείται σε απομακρυσμένο/εχθρικό περιβάλλον εκεί που δεν θα μπορεί να φτάσει ο άνθρωπος. Η πτυχιακή εργασία έφτασε μέχρι ένα πρώιμο στάδιο της αρχικής ιδέας λόγω έλλειψης εμπειρίας. Επιλέχθηκαν οικονομικά αλλά και αξιόπιστα υλικά. Ο χρόνος εκπόνησης της εργασίας και η ενασχόληση μας με το ρομποτικό σύστημα, αποτέλεσε πηγή γνώσης και μάθησης, τόσο σε θεωρητικό επίπεδο των τεχνικών 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 της ηλεκτρονικής και πιο συγκεκριμένα με ρομποτικά συστήματα και συστήματα αυτομάτου ελέγχου. Επίσης η ασχολία μας με πλατφόρμες προγραμματισμού μας βοήθησαν να κατανοήσουμε την λειτουργία των αλγορίθμων, των δομών προγραμματισμού, αλλά και των γλωσσών προγραμματισμού. 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 κάποια project που πρόκειται να ακολουθήσουμε στο μέλλον.

5.3 Βελτιώσεις πτυχιακής εργασίας

Η παρούσα κατασκευή επιδέχεται πολλές βελτιώσεις καθώς και ο σκοπός της είχε να γίνει η βάση για περαιτέρω ανάπτυξη. Ενδεικτικά κάποιες από τις βελτιώσεις/επεκτάσεις της κατασκευής είναι η προσθήκη περισσότερων υπέρηχων αισθητήρων για την καλύτερη πλοήγηση του οχήματος στο χώρο, προσθήκη αισθητηρίου GPS για αποστολή συγκεκριμένων συντεταγμένων, προσθήκη αισθητηρίου θερμοκρασίας, υγρασίας. Θα μπορούσε να προστεθεί μια ρομποτική IP camera (360°) για καταγραφή του χώρου σε ένα άγνωστο περιβάλλον και τέλος τοποθέτηση απλού ρομποτικού βραχίονα με χρήση servo.

ΒΙΒΛΙΟΓΡΑΦΙΑ - ΔΙΑΔΙΚΤΥΑΚΑ ΑΡΘΡΑ

1. Ρομποτικό σύστημα σταθερής βάσης.
<http://robotecmed.blogspot.gr>
2. Ιστορία των ρομποτ.
https://en.wikipedia.org/wiki/History_of_robots
3. Αναπτυξιακό Arduino UNO Board.
<https://www.arduino.cc/en/Main/ArduinoBoardUno>
4. Arduino IDE Software.
<https://www.arduino.cc/en/Main/Software>
5. Datasheet ATmega328P.
http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf
6. Σχηματικό ακροδεκτών ολοκληρωμένου ATmega328P.
<https://www.arduino.cc/en/Hacking/PinMapping168>
7. PWM κυματομορφή.
<https://www.arduino.cc/en/Tutorial/PWM>
8. Κυκλωματική διάταξη Dual H-Bridge.
<http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/>
9. Datasheet L298N and Tutorial.
<https://forum.arduino.cc/index.php?action=dlattach;topic=366727.0;attach=147606>
10. Αισθητήρας υπερήχων HC-SR04 με διαστάσεις (45*20*15mm).
<http://www.micropik.com/PDF/HCSR04.pdf>
11. Διάγραμμα λειτουργίας του αισθητήρα υπερήχων HC-SR04.
<http://www.geekfactory.mx/tutoriales/tutoriales-arduino/sensor-ultrasonico-hc-sr04-γ-arduino/>
12. Ultrasonic sensor HC-SR04 and Arduino tutorial.
<http://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>

13. Υπέρηχος.
<https://en.wikipedia.org/wiki/Ultrasound>
14. Φαινόμενο Doopler.
<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>
15. Μικροδιακόπτης επαφής (Microswitch SPDT ON-(ON) - with Big Lever).
https://en.wikipedia.org/wiki/Miniature_snap-action_switch
16. Σερβοκινητήρας (servomotor).
Βιβλίο "Αισθητήρες Μέτρησης και Ελέγχου" (Κωνσταντίνος Καλοβρέκτης-Νικόλαος Κατέβας).
17. Microservo Tower Pro 9g SG90.
<http://www.micropik.com/PDF/SG90Servo.pdf>
18. Bluetooth Wikipedia.
<http://en.wikipedia.org/wiki/Bluetooth>
19. Ιστορική αναδρομή Bluetooth.
<http://www.bluetooth.com/Pages/History-of-Bluetooth.aspx>
20. Bluetooth module HC-06.
<http://arduinolearning.com/learning/arduino-and-hc-06-bluetooth-example.php>
21. 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/>

