

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΪΑΤΡΙΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ**



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ: “Ανάπτυξη και υλοποίηση ρομποτικού οχήματος με ασύρματο έλεγχο ελεγχόμενο από μικροελεγκτή”

**ΦΟΙΤΗΤΗΣ: ΒΑΣΙΛΗΣ ΚΑΡΑΓΚΟΥΝΗΣ
Α.Μ:15042**

**Επιβλέπων καθηγητής: Παντελεήμων Ασβεστάς,
Αναπληρωτής καθηγητής**

ΑΘΗΝΑ 24/2/2022

Η Τριμελής Εξεταστική Επιτροπή

Ο Επιβλέπων Καθηγητής

Παντελεήμων Ασβεστάς

Δημήτριος Γκλώτσος

Σπυρίδων Κωστόπουλος

Αναπληρωτής καθηγητής

Αναπληρωτής καθηγητής

Επίκουρος καθηγητής

[ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ]

[ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ]

[ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ]

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/η υπογράφων/ουσα ΒΑΣΙΛΕΙΟΣ ΚΑΡΑΓΚΟΥΝΗΣ του ΙΩΑΝΝΗ, με αριθμό μητρώου 15042 φοιτητής/τρια του Τμήματος ΜΗΧΑΝΙΚΩΝ ΒΙΟΙΑΤΡΙΚΗΣ της Σχολής ΜΗΧΑΝΙΚΩΝ του Πανεπιστημίου Δυτικής Αττικής, δηλώνω υπεύθυνα ότι:

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

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου».

Ημερομηνία

24/2/2022

Ο/Η Δηλών/ούσα



Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στο πλαίσιο του προπτυχιακού προγράμματος σπουδών του τμήματος Μηχανικών Βιοιατρικής τεχνολογίας του Πανεπιστημίου Δυτικής Αττικής. Θα ήθελα να εκφράσω την ευγνωμοσύνη μου και να ευχαριστήσω ιδιαίτερα τον επιβλέποντα της εργασίας κύριο Παντελή Ασβεστά, Αναπληρωτή Καθηγητή του τμήματος για την πολύτιμη βοήθεια του, καθώς επίσης και για το χρόνο που μου αφιέρωσε. Ιδιαίτερα επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου και ιδιαίτερα τους γονείς μου για την υποστήριξη τους, την ενθάρρυνση αλλά και τη σημαντική βοήθεια που μου προσέφεραν στο ξεκίνημα της ακαδημαϊκής μου πορείας.

Επιπλέον θα ήθελα να ευχαριστήσω θερμά όλους τους καθηγητές που βοήθησαν να είμαι εδώ που βρίσκομαι σήμερα μοιράζοντας μαζί μου αλλά και με τους υπόλοιπους φοιτητές τις γνώσεις τους.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	8
ΚΕΦΑΛΑΙΟ 1 ^ο	10
Ρομποτική και Ρομποτικά συστήματα.	10
1.1 Εισαγωγή	10
1.2 Ρομποτικά συστήματα στην αρχαιότητα.	11
1.3 Είδη ρομποτικών συστημάτων.	13
1.4 Ιστορική αναδρομή σε έντροχα ρομποτικά συστήματα.	16
ΚΕΦΑΛΑΙΟ 2 ^ο	19
Εφαρμογές έντροχων ρομποτικών συστημάτων.....	19
2.1 Εισαγωγή	19
2.2 Εφαρμογή έντροχού ρομποτικού συστήματος ανίχνευσης αντικειμένων.	19
2.3 Εφαρμογή ρομποτικού συστήματος αυτό-ισορρόπησης.	21
2.4 Τετράτροχο ρομποτικό σύστημα ελεγχόμενο ασύρματα από επιταχυνσιόμετρο με τη βοήθεια πομποδέκτη.	23
ΚΕΦΑΛΑΙΟ 3 ^ο	25
Σχεδιασμός τετράτροχου ρομποτικού συστήματος και ανάλυση τεχνικών και ηλεκτρονικών εξαρτημάτων.	25
3.1 Εισαγωγή	25
3.2 Κεντρική μονάδα ελέγχου αυτοκινήτου	26
3.3 Κεντρική μονάδα ελέγχου τηλεχειρισμού	29
3.4 L293D Motor Driver Shield	32
3.5 MPU-6050.....	36
3.6 RF 433 Μονάδα ραδιοσυχνοτήτων.....	38
3.7 Μηχανολογικά μέρη και εξαρτήματα του τετράτροχου ρομποτικού συστήματος.....	40
3.8 Συναρμολόγηση εξαρτημάτων αμαξίδιου	41
3.9 Συναρμολόγηση εξαρτημάτων τηλεχειριστήριου	44
ΚΕΦΑΛΑΙΟ 4 ^ο	47
Προγραμματισμός τετράτροχου ρομποτικού συστήματος.	47
4.1 Εισαγωγή	47
4.2 Προγραμματισμός.....	47
4.3Πρόγραμματιστικό περιβάλλον ARDUINO	49
4.4 Κώδικες οδήγησης τετράτροχου ρομποτικού συστήματος σε περιβάλλον ARDUINO	49
4.4.1 Κώδικας τετράτροχου.	49
4.4.2 Κώδικας χειριστηρίου.....	52
ΣΥΜΠΕΡΑΣΜΑΤΑ	56
ΒΙΒΛΙΟΓΡΑΦΙΑ	58
ΑΝΑΦΟΡΕΣ	59

Κατάλογος Εικόνων

Εικόνα 1: : Ρομποτικό σύστημα σταθερής βάσης [1].....	14
Εικόνα 2: Τετρακόπτερο ρομποτικό σύστημα (Quadra Copter).	15
Εικόνα 3: Παράδειγμα AUV Ρομποτικού συστήματος [2]	15
Εικόνα 4: Έντροχο ρομποτικό σύστημα ανίχνευσης αντικειμένων	19
Εικόνα 5: Λειτουργία Πομπού – Δέκτη	20
Εικόνα 6: Κύκλωμα σύνδεσης των LED και του πομποδέκτη υπέρυθρων	21
Εικόνα 7: Δίτροχο αυτόνομο ρομποτικό σύστημα αυτό –ισορρόπησης [3].	22
Εικόνα 8: Εικόνα ρομποτικού συστήματος.	25
Εικόνα 9: Εικόνα συστήματος ασύρματου τηλεχειρισμού.....	26
Εικόνα 10: Είσοδοι και έξοδοι της πλακέτας Arduino UNO [4].....	28
Εικόνα 11: Αναπτυξιακό Arduino Nano [5]	29
Εικόνα 12: Πλάκετα Motor Driver Shield [6]	32
Εικόνα 13: Κύκλωμα γέφυρας H Motor Shield [7]	32
Εικόνα 14: Κύκλωμα L293D [8]	33
Εικόνα 15: Λειτουργία ακροδεκτών [9]	33
Εικόνα 16: Ολοκληρωμένο 74HC595 [10]	35
Εικόνα 17: Λειτουργία Pin [11].....	35
Εικόνα 18: MPU-6050 [12]	37
Εικόνα 19: RF 433Mhz [13]	40
Εικόνα 20: Σασί αυτοκινήτου μαζί με τους κινητήρες και τη θήκη μπαταριών [14]	41
Εικόνα 21: Κινητήρας μαζί με τροχό του συστήματος [15]	41
Εικόνα 22: Συναρμολόγηση τροχών και τοποθέτηση πάνω στο σασί του αμαξιδίου	42
Εικόνα 23: Εγκατάσταση Arduino Uno πάνω στο σασί	42
Εικόνα 24: Εφαρμογή L293D Motor Driver Shield πάνω στο Arduino Uno και σύνδεση καλωδίων κινητήρων.	43
Εικόνα 25: Σχηματικό σύνδεσης μονάδας λήψης RF 433Mhz	43
Εικόνα 26: Σύνδεση RF 433Mhz με το Motor Driver Shield	44
Εικόνα 27: Θήκη μπαταρίας 9V και 6V αντίστοιχα για το Driver Motor Shield και το Arduino	44
Εικόνα 28: Arduino Uno.....	45

Εικόνα 29: MPU-6050	45
Εικόνα 30: RF 433Mhz Transmitter	46
Εικόνα 31: Συνδεσμολογία Arduino Nano με το MPU-6050, RF 433Mhz και την τροφοδοσία	46
Εικόνα 32: Χειριστήριο Τετράτροχου.	46
Εικόνα 33: Αλγόριθμος κίνησης Arduino Uno	50
Εικόνα 34: Αλγόριθμος κίνησης Arduino Uno, Συνέχεια	51
Εικόνα 35: Κώδικας χειριστηρίου.	52
Εικόνα 36: Κώδικας χειριστηρίου συνέχεια	53
Εικόνα 37: Κώδικας χειριστηρίου τέλος	53
Εικόνα 38: Συνθήκη με if και βιβλιοθήκη Wire.h	54
Εικόνα 39: Δήλωση μεταβλητών.....	54
Εικόνα 40: Συναρτήσεις στη Void Setup	55
Εικόνα 41: Κίνηση δεξια-αριστερα	55

ΠΕΡΙΛΗΨΗ

Βασικό πρόβλημα στο χώρο των ρομποτικών συστημάτων αποτελεί αυτό της πλοήγησης. Έχει να κάνει με την χαρτογράφηση στον περιβάλλοντα χώρο, τον έλεγχο της μετακίνησης για την ασφαλή κίνηση σε αυτόν και τέλος τον εντοπισμό της θέσης. Η ανάπτυξη της ρομποτικής δραστηριότητας για τη λύση του προβλήματος στη ρομποτική πλοήγηση, τα τελευταία χρόνια έχει οδηγήσει σε διάφορες εναλλακτικές προσεγγίσεις για ποικίλες περιπτώσεις.

Στην τρέχουσα διπλωματική εργασία έχουμε ως αντικείμενο την πλοήγηση και κατασκευή ενός τετράτροχου ρομποτικού συστήματος που έχει σκοπό την κίνηση μέσω ενός ασύρματου χειριστηρίου. Το συγκεκριμένο ρομποτικό σύστημα έχει ως κύριο εξάρτημα το Arduino Uno για το τετράτροχο και το Arduino Nano για το ασύρματο χειριστήριο όπου αποτελούν τις κεντρικές μονάδες για την επεξεργασία και την αποστολή δεδομένων. Στη συγκεκριμένη εργασία βλέπουμε τις εφαρμογές που θα μπορούσαν να έχουν οι κατασκευές ρομποτικής στην καθημερινότητα αλλά και σε επίπεδο έρευνας. Η συγκεκριμένη κατασκευή αποτελεί ένα χαρακτηριστικό παράδειγμα, αφού με την κατάλληλη μετατροπή μπορεί να μετατραπεί σε ρομποτικό σύστημα υποβοήθησης ατόμων με αναπηρίες.

Λέξεις κλειδιά: Arduino Uno, Arduino Nano, τετράτροχο ρομποτικό σύστημα, ασύρματο χειριστήριο.

ABSTRACT

One of the main issues in the field of automatic robotic systems is that of navigation. It includes the mapping of the surrounding area, the detection of the current location and, finally, the regulation of the movement for the safe movement in the area. In recent years, significant research activity has been developed around the issue of robotic navigation that has led to a plethora of alternative approaches for various cases.

The main objective of this dissertation is the construction and navigation of a four-wheeled robotic system that can be moved through a wireless controller. The particular robotic system has as main component the ArduinoUno for the four-wheel controller and the ArduinoNano for the wireless one, which constitute the central units for the processing and transmission of data. Through this dissertation, the usefulness that robotics applications can have at the research level and in everyday life is highlighted. The implementation of the construction can be a representative example, since with the appropriate equipment it could be transformed into a robotic assistance system for people with disabilities.

Key-words: ArduinoUno, ArduinoNano, four-wheel robotic system, wireless controller

ΚΕΦΑΛΑΙΟ 1^ο

ΡΟΜΠΟΤΙΚΗ ΚΑΙ ΡΟΜΠΟΤΙΚΑ ΣΥΣΤΗΜΑΤΑ

1.1 Εισαγωγή

Η ρομποτική είναι η τομή της επιστήμης, της μηχανικής και της τεχνολογίας που παράγει μηχανές, που ονομάζονται ρομπότ και έχει σκοπό να υποκαθιστούν ενέργειες για την εκτέλεση εργασιών που εκτελούνται παραδοσιακά από τα ανθρώπινα όντα. Έχει να κάνει με τον συνδυασμό της Ηλεκτρολογίας με την Μηχανολογία, την Πληροφορική και τις Τηλεπικοινωνίες και περιλαμβάνει τον σχεδιασμό, κατασκευή, λειτουργία και χρήση ρομπότ. Ακόμη πολλές πτυχές της ρομποτικής περιλαμβάνουν τεχνητή νοημοσύνη. Τα ρομπότ μπορούν να προγραμματιστούν και να εφοδιαστούν με αισθητήρες ώστε να αντιγράφουν τις ανθρώπινες αισθήσεις όπως η αφή, η όραση και καταγραφή της θερμοκρασίας. Μερικοί είναι ακόμη ικανοί για απλή λήψη αποφάσεων ενώ ένα ρομπότ σε ανθρώπινη μορφή ονομάζεται Android.

Ένα ρομπότ είναι ένα μηχάνημα - ειδικά ένα προγραμματιζόμενο από έναν υπολογιστή - ικανό να διεκπεραιώνει αυτόματα μια σειρά πολύπλοκων ενεργειών. Τα ρομπότ είναι σχεδιασμένα να ελέγχονται από μια εξωτερική συσκευή ελέγχου ή το στοιχείο ελέγχου μπορεί να είναι ενσωματωμένο μέσα. Τα ρομπότ μπορεί να κατασκευάζονται σύμφωνα με την ανθρώπινη μορφή, αλλά τα περισσότερα ρομπότ αποτελούν μηχανές σχεδιασμένες για να εκτελούν μια εργασία, ανεξάρτητα από την αισθητική τους.

Τα ρομπότ μπορεί να είναι αυτόνομα ή ημι-αυτόνομα και να κυμαίνονται από ανθρωποειδή έως βιομηχανικά ρομπότ, ιατρικά λειτουργικά ρομπότ, ρομπότ υποβοήθησης ασθενών, ρομπότ θεραπείας σκύλου, συλλογικά προγραμματισμένα σμήνη ρομπότ, UAV drone, ακόμη και μικροσκοπικά νανο-ρομπότ. Με τη μίμηση μιας ζωντανής εμφάνισης ή την αυτοματοποίηση των κινήσεων, ένα ρομπότ μπορεί να μεταφέρει μια αίσθηση νοημοσύνης ή σκέψης από μόνη της. Κάθε ρομπότ έχει διαφορετικό επίπεδο αυτονομίας, που κυμαίνεται από bots ελεγχόμενα από τον άνθρωπο που εκτελούν εργασίες που ο άνθρωπος έχει πλήρη έλεγχο έως και αυτόνομα bots που εκτελούν εργασίες χωρίς εξωτερικές επιρροές.

Τα ρομπότ χρειάζονται ηλεκτρικά εξαρτήματα που ελέγχουν και τροφοδοτούν τα μηχανήματα. Ουσιαστικά, απαιτείται ένα ηλεκτρικό ρεύμα (για παράδειγμα μια μπαταρία) για την τροφοδοσία της μεγάλης πλειονότητας των ρομπότ. Σχετικά στενά με την έννοια

του ρομπότ είναι το πεδίο της Συνθετικής Βιολογίας, το οποίο μελετά οντότητες των οποίων η φύση είναι πιο συγκρίσιμη με τα όντα παρά με τις μηχανές.

1.2 Ρομποτικά συστήματα στην αρχαιότητα

Τον Μεσαίωνα, τόσο στην Ευρώπη όσο και στη Μέση Ανατολή, οι αυτοματισμοί ήταν δημοφιλή ως μέρος ρολογιών και θρησκευτικής λατρείας. Ο αραβικός πολυμαθητής Al-Jazari (1136-1206) άφησε κείμενα που περιγράφουν και απεικονίζουν τις διάφορες μηχανικές του συσκευές, συμπεριλαμβανομένου ενός μεγάλου ρολογιού ελέφαντα που κινούνταν και άκουγε την ώρα, ένα μουσικό συγκρότημα ρομπότ και ένα αυτοματοποιημένο ρομπότ-σερβιτόρο που σερβίρει ποτά. Στην Ευρώπη, υπάρχει ένας αυτόματος μοναχός που φιλά τον σταυρό στα χέρια του. Πολλά άλλα αυτόματα δημιουργήθηκαν που έδειξαν κινούμενα ζώα και ανθρωποειδείς μορφές που λειτουργούσαν σε απλά συστήματα έκκεντρων, αλλά τον 18ο αιώνα, τα αυτόματα κατανοήθηκαν αρκετά και η τεχνολογία προχώρησε στο σημείο όπου θα μπορούσαν να γίνουν πολύ πιο περίπλοκα κομμάτια. Ο Γάλλος μηχανικός Jacques de Vaucanson πιστώνεται ότι δημιούργησε το πρώτο επιτυχημένο βιομηχανικό αυτόματο, μια ανθρώπινη φιγούρα που παίζει το φλάουτο. Τα Automata ήταν τόσο δημοφιλή που ταξίδεψαν στην Ευρώπη διασκεδάζοντας αρχηγούς κρατών, όπως ο Frederick the Great και ο Napoleon Bonaparte.

Η Βιομηχανική Επανάσταση και η αυξημένη εστίαση στα μαθηματικά, τη μηχανική και την επιστήμη στην Αγγλία στην εποχή της Βικτώριας προστέθηκαν στην ορμή προς την πραγματική ρομποτική. Ο Charles Babbage (1791-1871) εργάστηκε για να αναπτύξει τα θεμέλια της επιστήμης των υπολογιστών στις αρχές έως τα μέσα του δέκατου ένατου αιώνα, με τα πιο επιτυχημένα έργα του να είναι η μηχανή διαφοράς και η αναλυτική μηχανή. Αν και δεν ολοκληρώθηκε ποτέ λόγω έλλειψης χρημάτων, αυτά τα δύο μηχανήματα παρουσίασαν τα βασικά στοιχεία για μηχανικούς υπολογισμούς. Άλλοι όπως η Ada Lovelace αναγνώρισαν τη μελλοντική δυνατότητα των υπολογιστών να δημιουργούν εικόνες ή να παίζουν μουσική.

Η Automata εξακολούθησε να προσφέρει ψυχαγωγία κατά τη διάρκεια του 19ου αιώνα, αλλά συναίσθημα με αυτήν την περίοδο ήταν η ανάπτυξη μηχανημάτων ατμού και κινητήρων που βοήθησαν να κάνουν την κατασκευή πολύ πιο αποτελεσματική και γρήγορη. Τα εργοστάσια άρχισαν να χρησιμοποιούν μηχανές είτε για να αυξήσουν το φόρτο εργασίας είτε για την ακρίβεια στην παραγωγή πολλών προϊόντων.

Το 1920, ο Karel Capek δημοσίευσε το έργο του R.U.R. (Universal Robots του Rossum), η οποία εισήγαγε τη λέξη «ρομπότ». Λήφθηκε από μια παλιά σλαβική λέξη που σήμαινε κάτι παρόμοιο με «μονότονη ή καταναγκαστική εργασία». Ωστόσο, χρειάστηκαν περισσότερα από τριάντα χρόνια πριν ξεκινήσει να λειτουργεί το πρώτο βιομηχανικό ρομπότ. Στη δεκαετία του 1950, ο George Devol σχεδίασε το Unimate, μια ρομποτική συσκευή βραχίονα που μετέφερε χύτευση σε εργοστάσιο General Motors στο Νιου Τζέρσεϋ, το οποίο άρχισε να λειτουργεί το 1961. Unimation, η εταιρεία Devol ιδρύθηκε με τον ρομποτικό επιχειρηματία Joseph Engelberger, ήταν η πρώτη κατασκευή ρομπότ Εταιρία. Το ρομπότ θεωρήθηκε αρχικά ως περιέργεια. Επίσης στα μέσα της δεκαετίας του 1950, η γερμανική εταιρεία Kuka ανέπτυξε μια αυτοματοποιημένη γραμμή συγκόλλησης για συσκευές, καθώς και μια γραμμή συγκόλλησης πολλαπλών σημείων για τη Volkswagen. Μέχρι το 1968, η Kawasaki είχε αδειοδοτήσει ένα σχέδιο Unimation ενός υδραυλικού ρομπότ και είχε αρχίσει να τα κατασκευάζει. Το 1969, η GM είχε επιτύχει το 90% των συγκολλήσεων σώματος χρησιμοποιώντας Unimates σε ένα από τα εργοστάσιά της. Το 1970, το Πανεπιστήμιο του Στάνφορντ ανέπτυξε το λεγόμενο Standard Arm, όπως είναι ακόμα γνωστό σήμερα, που χρησιμοποιείται για συναρμολόγηση μικρών ανταλλακτικών και ενσωματώνει ανατροφοδότηση αφής και πίεσης.

Ο Kuka, το 1973, είχε εισαγάγει το βραχίονα έξι αξόνων, το οποίο θα γινόταν βιομηχανικό πρότυπο. Ήταν περίπου την ίδια στιγμή που άρχισαν να εμφανίζονται τα πλήρως ηλεκτρικά ρομπότ. Η Cincinnati Milacron παρουσίασε ένα βιομηχανικό ρομπότ ελεγχόμενο από μικροϋπολογιστές για εμπορική χρήση τον ίδιο χρόνο. Καθώς η δεκαετία του '70 κυκλοφόρησε, η δεκαετία είδε πολλά πρώτα: ένα ρομπότ ελεγχόμενο από ηλεκτρικό μικροεπεξεργαστή, ένα ρομπότ εισαγωγής υψηλής ακρίβειας, υψηλότερα ωφέλιμα φορτία, ένα ρομπότ συγκόλλησης με βάση αισθητήρες, το ρομπότ PUMA για συναρμολόγηση μικρών ανταλλακτικών, ανάπτυξη του Selectively Compliant (SCARA) βραχίονα και η μετάβαση από υδραυλικούς σε ηλεκτροκινητήρες σε ρομπότ spot συγκόλλησης.

Όταν έφτασε το 1980, το ίδιο έκανε και μια επίδειξη μηχανικής όρασης στο Πανεπιστήμιο του Ρόουντ Άιλαντ. Η GM θα έθετε τρία ρομπότ για να λειτουργήσει τον επόμενο χρόνο ταξινομώντας χύτευση με μηχανική όραση. Μεγαλύτερη καινοτομία θα έπαιζε τη δεκαετία: την ανάπτυξη μιας γλώσσας προγραμματισμού ρομπότ, ενός ρομπότ άμεσης κίνησης SCARA. Οι ταχύτητες και οι χωρητικότητες ανεβαίνουν επίσης.

Η δεκαετία του 1990 έφερε καινοτομίες στον έλεγχο ρομπότ και τον συγχρονισμό, καθώς και το πρώτο ρομπότ συσκευασίας, το οποίο φόρτωσε κουλούρια. Ένα δίπλωμα ευρεσιτεχνίας στα τέλη της δεκαετίας πρόσθεσε λέιζερ που κατευθύνει απευθείας στον βραχίονα του ρομπότ.

Κατά τη διάρκεια του τελευταίου μισού αιώνα, τα ρομπότ έχουν βρει μια θέση σε άλλους τομείς, όπως παιχνίδια και ψυχαγωγία, στρατιωτικά όπλα, βοηθοί αναζήτησης και διάσωσης, και πολλές άλλες θέσεις εργασίας. Ουσιαστικά, καθώς ο προγραμματισμός και η τεχνολογία βελτιώνονται, τα ρομπότ βρίσκουν τον δρόμο τους σε πολλές θέσεις εργασίας που στο παρελθόν ήταν πολύ επικίνδυνες, θαμπό ή αδύνατες για τους ανθρώπους.

1.3 Είδη ρομποτικών συστημάτων

Η πολυετή εξέλιξη της επιστήμης γύρω από τη ρομποτική είχε ως αποτέλεσμα την δημιουργία διάφορων ρομποτικών μηχανισμών με σημαντικές διαφορές στη μορφή, αποτελούμενα όμως από αντίστοιχα επιμέρους υποσυστήματα.

Τα μεγαλύτερα είδη ρομποτικών συστημάτων είναι τα παρακάτω:

- **Ρομποτικά συστήματα σταθερής βάσης:** Η κατασκευή των συγκεκριμένων ρομποτικών συστημάτων γίνεται με διαδοχικά στερεά σώματα (συνδέσμους) που ενώνονται μέσω αρθρώσεων και δημιουργούν μια κινηματική αλυσίδα. Αύτη η αλυσίδα αποτελείται από ένα άκρο (βασή) που είναι συνδεδεμένο σταθερά σε κάποιο σημείο το οποίο βρίσκεται. Αύτη η μορφή αναπαριστά μια μορφή βιομηχανικού ρομποτικού βραχίονα και αποτελείται από τον καρπό, το βραχίονα και το εργαλείο. Ένα παράδειγμα τέτοιου συστήματος παρουσιάζεται στην εικόνα 1.



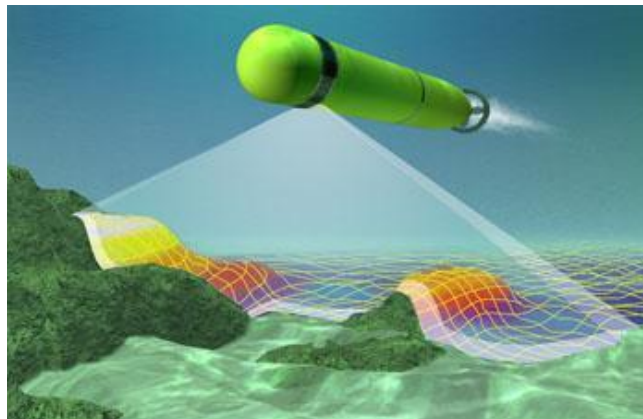
Εικόνα 1: Ρομποτικό σύστημα σταθερής βάσης [1].

- **Κινούμενα ρομποτικά συστήματα:** Ως κινητά ρομποτικά συστήματα ορίζονται όλα τα συστήματα που μπορούν να μετατοπίσουν όλα τα σημεία που το περιλαμβάνουν. Αυτό είναι εφικτό από ειδικά συστήματα προώθησης απλά ή πολυπλοκότερα όπως τροχοί, τουρμπίνες, προπέλες, μηχανικά πόδια κ.α. Το αντικείμενο της διπλωματικής εργασίας ανήκει σε αυτήν την κατηγορία και ειδικότερα στα έντροχα ρομποτικά συστήματα. Πιο συγκεκριμένα έχει τη δυνατότητα λειτουργίας με εξωτερική επίβλεψη εκτελώντας εργασίες δεχόμενα από εντολές.
- **Εναέρια ρομποτικά συστήματα:** Αφορά μη επανδρωμένα ιπτάμενα ρομποτικά συστήματα, όπως αεροπλάνα και ελικόπτερα. Η εφαρμογή των συγκεκριμένων συστημάτων έχουν διαρκώς αυξανόμενους ρυθμούς όμως λόγω της μειωμένης ασφάλειας και σταθερότητας ως προς την συμπεριφορά τους εφαρμόζονται περισσότερο για στρατιωτικούς σκοπούς. Παρόλα αυτά η διαρκής εξέλιξη της τεχνολογίας και η εφεύρεση των τετρακοπτέρων αλλά και εξακοπτέρων, συστήματα όπου έδιναν βάρος στη σταθερότητα την ευκολία διαχείρισης αλλά και την ασφάλεια (Quadra copter, hex copter) βρίσκουν εφαρμογή στην καθημερινότητα όλο και περισσότερο για προσωπικούς ή και επαγγελματικούς σκοπούς (βιντεοσκοπήσεις, φωτογραφία, ερευνητικά, κ.α.). Στο Σχήμα 2 παρουσιάζεται ένα είδος τετρακοπτέρου.



Εικόνα 2: Τετρακόπτερο ρομποτικό σύστημα (Quadra Copter).

- **Ρομποτικά συστήματα AUVs:** Τα AUVs (Autonomous Underwater Vehicles) ανήκουν στην κατηγορία των μη επανδρωμένων υποβρύχιων ρομπότ και είναι πλήρως αυτόνομα ρομποτ. Η τροφοδοσία γίνεται από ειδικές μπαταρίες που όμως δημιουργεί θέματα περιορισμού στη χρήση τους. Το σχήμα τους είναι αντίστοιχο με αυτό της τορπίλης και έχουν τη δυνατότητα κίνησης με αρκετά μεγάλη ταχύτητα. Η χρήση τους αφορά σε υποβρύχιες κατασκευές αλλά και εξερευνήσεις.



Εικόνα 3: Παραδειγμα AUV ρομποτικού συστήματος [2]

1.4 Ιστορική αναδρομή σε έντροχα ρομποτικά συστήματα

Στο κεφάλαιο αυτό, θα γίνει μια αναδρομή στην ιστορία και την εξέλιξη των κινούμενων ρομποτικών συστημάτων, η οποία έχει ξεκινήσει από τα μέσα του 20^{ου} αιώνα και συνεχίζει την εξέλιξη μέχρι σήμερα. Η δημιουργία των έντροχων ρομποτικών συστημάτων έγινε για την επίτευξη διάφορων αναγκών, από την εκτέλεση απλών καθημερινών εργασιών, μέχρι πολύπλοκες έρευνες.

1939-1945: Στην διάρκεια του Β' Παγκόσμιου Πολέμου αρχίζουν να εμφανίζονται τα πρώτα κινούμενα ρομποτικά συστήματα.

1948-1949: Γίνεται κατασκευή από τον William Grey Walter δύο αυτόνομων ρομπότ, το Elmer και το Elsie, τα οποία εξερευνούσαν το περιβάλλον τους. Συγκεκριμένα το κάθε ρομπότ έφερε πάνω του έναν αισθητήρα φωτός με τον οποίο μπορούσαν να ανιχνεύσουν μία πηγή φωτός και να κατευθυνθούν προς αυτή αποφεύγοντας παράλληλα διάφορα εμπόδια που μπορεί να υπήρχαν. Αυτή ήταν η απόδειξη πως από έναν απλό σχεδιασμό μπορούσαν να έχουν μια τόσο πολύπλοκη συμπεριφορά.

1961-1963: Δημιουργήθηκε από το Johns Hopkins University ένα ρομποτικό σύστημα Beast το οποίο χρησιμοποιούσε ένα σόναρ για την κίνηση του και μπόρουσε να εντοπίσει μια βάση φόρτισης, με την οποία και θα συνδεόταν όταν οι μπαταρίες θα αποφορτιζόνταν.

1970: Η εξέλιξη συνεχίζεται από το πανεπιστήμιο του Stanford όπου δημιουργεί ένα κινούμενο ρομπότ με ενσωματωμένη κάμερα το οποίο μπορούσε να ακολουθήσει μια άσπρη γραμμή. Το Cart line follower ήταν ασύρματα συνδεδεμένο με έναν μεγάλο υπολογιστή, ο οποίος πραγματοποιούσε υπολογισμούς.

Τον ίδιο καιρό (1966-1972), στο ερευνητικό κέντρο του πανεπιστημίου Stanford αναπτύσει ένα ρομπότ με την ονομασία Shakey, το οποίο ήταν εξοπλισμένο με έναν αισθητήρα μέτρησης απόστασης, μια κάμερα, αισθητήρες επαφής καθώς επίσης και κεραία για ασύρματη επικοινωνία. Αποτέλεσε το πρώτο ρομποτικό σύστημα που αφορούσε σε ημιαυτόνομη χρήση. Αυτό σημαίνει ότι το ρομπότ μπορούσε να δεχτεί κάποιες πολύ γενικές εντολές και να αντιληφθεί τα απαραίτητα βήματα που έπρεπε να κάνει ώστε να φέρει εις πέρας την ζητούμενη εργασία.

Να αναφερθεί πως την ίδια χρονολογία η Σοβιετική Ένωση εξερευνούσε την επιφάνεια της Σελήνης, με το Lunokhod 1, ένα σεληνιακό Rover.

1976: Γίνεται αποστολή δυο μη επανδρωμένων διαστημικών σκαφών στον Άρη από τη NASA.

1980: Η εξέλιξη στα ρομπότ αυτοκίνητα συνεχίζεται με την ομάδα του Ernst Dickmanns να κατασκευάζει τα πρώτα ρομπότ – αυτοκίνητα, με δυνατότητα κίνησης σε άδειους δρόμους ως και 55 μίλια την ώρα.

1987: Ξεκινάει η κατασκευή από το Hughes Research Laboratories, της πρώτης αυτόνομης λειτουργίας ενός ρομποτικού συστήματος μέσω χαρτογράφησης και βασιζόμενη σε αισθητήρες,

1989: Ιδρύεται η ρομποτική εταιρεία BEAM (Biology, Electronics, Aesthetics, Mechanics) από τον Mark Tilden.

1990: Ο εφευρέτης του βιομηχανικού βραχίονα Joseph Engelberger, προσπαθεί να σχεδιάσει με συναδέλφους τα πρώτα εμπορικά διαθέσιμα, αυτόνομα κινούμενα ρομπότ για νοσοκομειακή χρήση. Η κυκλοφορία τους στην αγορά έγινε από την HelpMate. Επίσης το ίδιο διάστημα το υπουργείο Άμυνας των ΗΠΑ επενδύει πάνω σε ένα project (MDARS-I), το οποίο έχει να κάνει με ρομπότ ασφαλείας εσωτερικών χώρων.

1991: Ο Edo Franzι ο Adre Guignard και ο Francesco Mondada με την βοήθεια του πολυτεχνείου της Λωζάνης ανέπτυξαν το Khepera, ένα αυτόνομο κινούμενο ρομπότ, που δημιουργήθηκε για ερευνητικές δραστηριότητες.

1993-1994: Το Carnegie Mellon University αναπτύσει τα Dante I και Dante II. Και τα δυο ήταν βαδίζοντα ρομπότ και δημιουργήθηκαν για έρευνες ενεργών ηφαιστειών. Την ίδια περίοδο, αναπτύσσονται δυο δίδυμα ρομπότ – οχήματα VaMP της Daimler-Benz και VITA-2 της UNIBwM, τα οποία κατάφεραν να διανύσουν περισσότερα από 1000km/h σε αυτοκινητόδρομο του Παρισιού 3 λωρίδων και με αρκετά υψηλή κίνηση, με ταχύτητες κοντά στα 130 km/h. Τα ρομπότ αυτά ήταν ικανά για αυτόνομη οδήγηση σε όλες τις πιθανές συνθήκες που μπορεί να υπάρξουν κατά την οδήγηση, ελεύθερες λωρίδες, οδήγηση σε φάλαγγα αυτοκινήτων, καθώς και αλλαγές λωρίδας (αριστερά και δεξιά), με αυτόνομη προσπέραση άλλων αυτοκινήτων.

1995: Γίνεται και άλλη προσπάθεια αυτόνομης οδήγησης με ένα ρομπότ αυτοκίνητο του Ernst Dickmanns όπου ταξίδεψε πάνω από 1000 μίλια, σε διάδομή από το Μόναχο στην Κοπεγχάγη με επιστροφή, εν μέσω κίνησης και ταχύτητα γύρω στα 120 μίλια την ώρα, κάνοντας περιστασιακά ελιγμούς για να προσπεράσει άλλα αυτοκίνητα. Ακόμα χρησιμοποιήθηκε ενεργή όραση ώστε να χαρτογραφεί και να αντιμετωπίζει τις ξαφνικές αλλαγές σε ορισμένα σημεία του δρόμου.

1996-1997: Πραγματοποιείται αποστολή από τη NASA στον Άρη για εξερεύνηση της επιφάνειας του Άρη, στέλνοντας το Mars Pathfinder μαζί με το rover του, το Sojourner,

δεχόμενο εντολές από την Γη. Το Sojourner διαθέτει σύστημα αποφυγής κινδύνων και επέτρεπε στο rover να βρίσκει αυτόνομα το δρόμο του μέσα στο άγνωστο έδαφος του Άρη.

1999: Γίνεται παρουσίαση του Aibo από τη Sony, ενός ρομποτικού σκύλου που είχε τη δυνατότητα να βλέπει, να βαδίζει και να αλληλεπιδρά με το περιβάλλον του. Επίσης παρουσιάζεται το τηλεκατευθυνόμενο στρατιωτικό ρομπότ PackBot.

2001: Αρχίζει η ανάπτυξη του Swarm-bots project. Τα Swarm bots παραπέμπουν σε αποικίες εντόμων, αποτελούμενα από έναν αριθμό μικρών ανεξάρτητων ρομπότ, τα οποία συνεργάζονται μεταξύ τους και συμβάλλουν στην εκτέλεση πολύπλοκων εργασιών.

2002: Παρουσιάζεται το πρώτο οικιακό αυτόνομο κινούμενο ρομπότ που καθαρίζει το πάτωμα με την ονομασία Roomba.

2003: Η Intellibot αγοράζεται από την Axxon Robotics, μια εταιρεία που κατασκευάζει ρομπότ που έχουν να κάνουν με την καθαριότητα (τρίβουν και σκουπίζουν τα πατώματα νοσοκομείων, γραφείων κτλ). Τα ρομπότ της Intellibot Robotics, μπορούν να λειτουργούν αυτόνομα, χρησιμοποιώντας ένα σύνολο αισθητήρων για την χαρτογράφηση του περιβάλλον τους και την αποφυγή εμποδίων κατά την πλοήγηση τους.

2005: Ένα τετράποδο ρομπότ δημιουργείται από την Boston Dynamics το οποίο προορίζεται για χρήση σε κακοτράχαλο έδαφος και μεταφορά φορτίων.

2007: Ξεκινάει η τεχνολογική 'επανάσταση' με το DARPA Urban Grand Challenge, όπου έξι με αυτόνομη οδήγηση οχήματα ολοκληρώνουν μια δύσκολη πορεία, με την ενσωμάτωση επανδρωμένων οχημάτων και εμποδίων. Την ίδια περίοδο το Seekur, το πρώτο ευρέως διαθέσιμο, μη στρατιωτικό, βοηθητικό ρομπότ εξωτερικών χώρων, έχει ικανότητα ρυμούλκησης ενός οχήματος τριών τόνων από μια θέση παρκινγκ, να το κατευθύνει αυτόνομα σε έναν εσωτερικό χώρο και στη συνέχεια, να καθοδηγηθεί και πάλι προς τα έξω.

2008: Δημοσιοποιείται απόσπασμα ενός βίντεο από την Boston Dynamics όπου δείχνει ένα νέας γενιάς BigDog με ικανότητα βάδισης σε παγωμένο έδαφος και ανάκτηση της ισορροπίας του.

2010: Στον διαγωνισμό Multi Autonomous Ground Robotic International Challenge (MAGIC), παρουσιάζονται ομάδες αυτόνομων οχημάτων με ικανότητα χαρτογράφησης ενός μεγάλου αστικού περιβάλλοντος, αναγνώριση και παρακολούθηση ανθρώπων, και αποφυγή εχθρικών αντικειμένων.

ΚΕΦΑΛΑΙΟ 2^ο

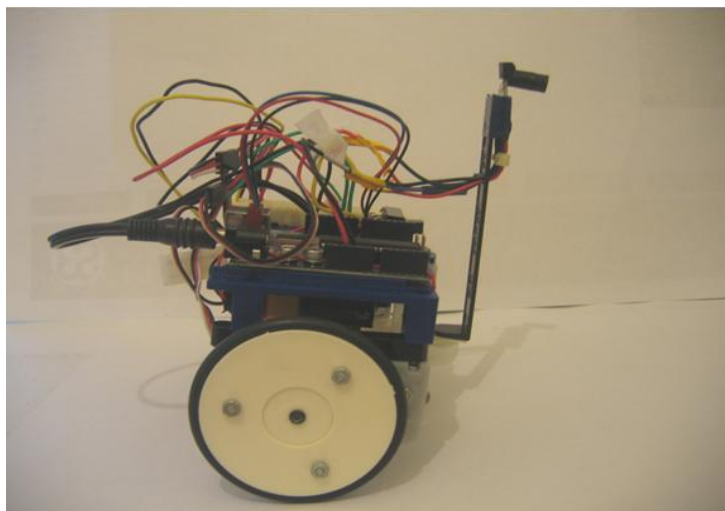
ΕΦΑΡΜΟΓΕΣ ΕΝΤΡΟΧΩΝ ΡΟΜΠΟΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

2.1 Εισαγωγή

Στο κεφάλαιο αυτό θα αναφερθούμε σε έντροχα ρομποτικά συστήματα, καθώς επίσης θα γίνει και μια σύντομη ανάλυση στον τρόπο λειτουργίας τους και κατασκευής τους. Όλες οι παρακάτω εφαρμογές δημιουργήθηκαν από άλλους φοιτητές για εκπαιδευτικούς σκοπούς και παρουσιάζουν αρκετές ομοιότητες με το ρομποτικό της παρούσας διπλωματικής εργασίας. Στις παρακάτω υποενότητες ακολουθούν δύο παραδείγματα ρομποτικών συστημάτων και στο τελευταίο μέρος γίνεται μια περιγραφή του τετράτροχου της συγκεκριμένης διπλωματικής εργασίας.

2.2 Εφαρμογή έντροχού ρομποτικού συστήματος ανίχνευσης αντικειμένων.

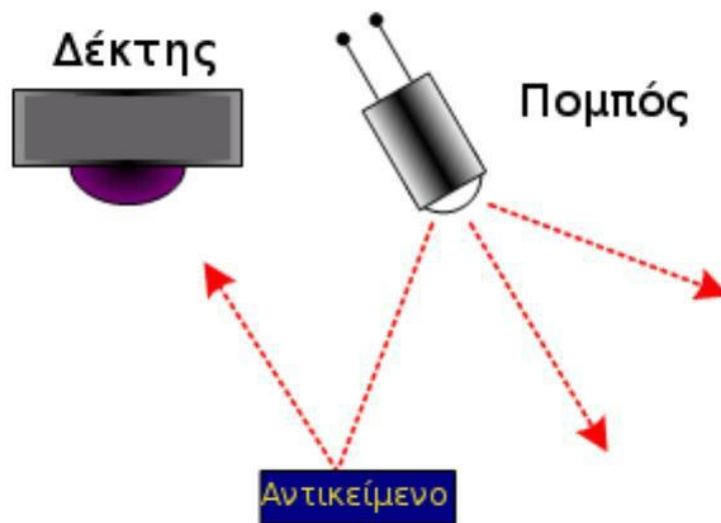
Το παρακάτω σχήμα («Εικόνα 4») απεικονίζει ένα έντροχο ρομποτικό σύστημα ανίχνευσης αντικειμένων, το οποίο ειδοποιεί τον χειριστή του για κάθε αντικείμενο που εντοπίζει. Οι ειδοποιήσεις εμφανίζονται με μια φωτεινή ένδειξη LED καθορισμένου χρώματος για τη συγκεκριμένη λειτουργία.



Εικόνα 4: Έντροχο ρομποτικό σύστημα ανίχνευσης αντικειμένων .

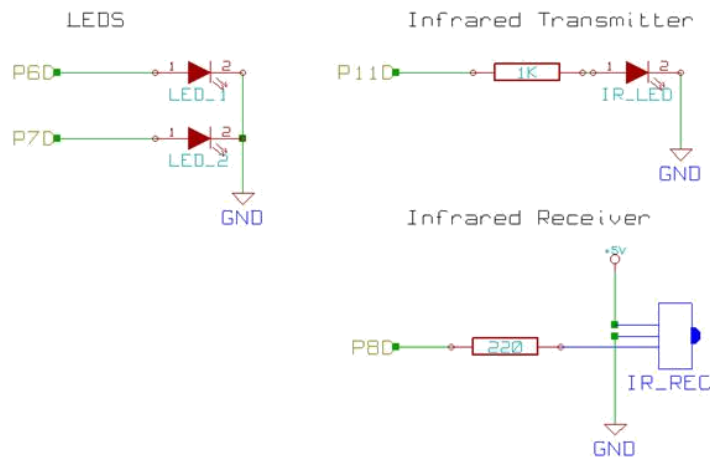
Το έντροχο ρομποτικό σύστημα ανίχνευσης αποτελείται από δύο κύρια εξαρτήματα: έναν πομπό υπέρυθρων (Parallax IR Transmitter Assembly Kit) και έναν δέκτη. Η κύρια ιδέα πίσω από την λειτουργία του ρομποτικού συστήματος είναι η συνεχής αποστολή παλμών από τον

πομπό, με αποτέλεσμα να δημιουργείται ανάκλιση, η οποία ανιχνεύεται στη συνέχεια από τον δέκτη. Προκειμένου να επιτευχθεί αυτό, είναι απαραίτητο να συγχρονιστούν ο πομπός και ο δέκτης, καθώς ο πρώτος (πομπός) πρέπει να εκπέμπει συνεχώς κύματα, και ο δεύτερος (δέκτης) να βρίσκεται σε συνεχή αναμονή για να ανιχνεύσει το κύμα (υπέρυθρες) που ανακλάται. Στην Εικόνα 5 αποτυπώνεται η λειτουργία του πομπού με τον δέκτη, όταν ανιχνεύεται αντικείμενο στον χώρο.



Εικόνα 5: Λειτουργία Πομπού – Δέκτη.

Σε αυτή την περίπτωση, χρησιμοποιείται ένας πομπός ο οποίος έχει δυνατότητα αποστολής ενός τετραγωνικού παλμού συχνότητας 38,5KHz (38500Hz), καθώς και ενός δέκτη ο οποίος έχει δυνατότητα ανίχνευσης του σήματος σε αυτή την συχνότητα. Προκειμένου να κατασκευαστεί αυτό το σύστημα, επιλέχτηκαν εξαρτήματα τα οποία υπάρχουν διαθέσιμα σε καταστήματα πώλησης ηλεκτρικών ειδών και στο διαδίκτυο. Επιπλέον, υπάρχουν διαθέσιμες έτοιμες διατάξεις (modules) υπέρυθρων πομποδεκτών, που είναι ειδικά σχεδιασμένες για ρομποτικές εφαρμογές. Ο έλεγχος της εφαρμογής γίνεται από το αναπτυξιακό **Arduino Diecimila**, το οποίο έχει την ικανότητα να συγχρονίζει όλες τις λειτουργίες του συστήματος. Προκειμένου να αποτυπωθούν οι ενδείξεις όταν ανιχνεύεται ένα αντικείμενο, χρησιμοποιούνται 2 LED διαφορετικού χρώματος, ένα κόκκινο και ένα πράσινο). Παράλληλα, για την διάταξη του κυκλώματος εφαρμόστηκαν 2 αντιστάσεις, μια των 220Ω και μια των 1KΩ, αντίστοιχα. Στην Εικόνα 6 παρουσιάζεται η συνδεσμολογία των αντιστάσεων με τον πομποδέκτη.

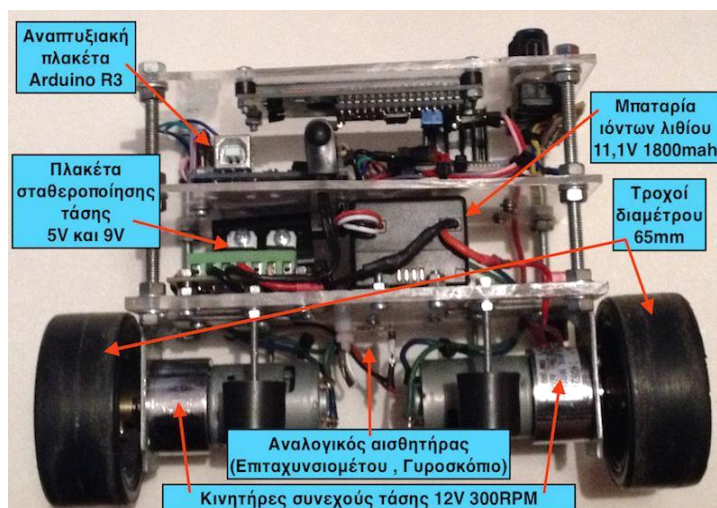


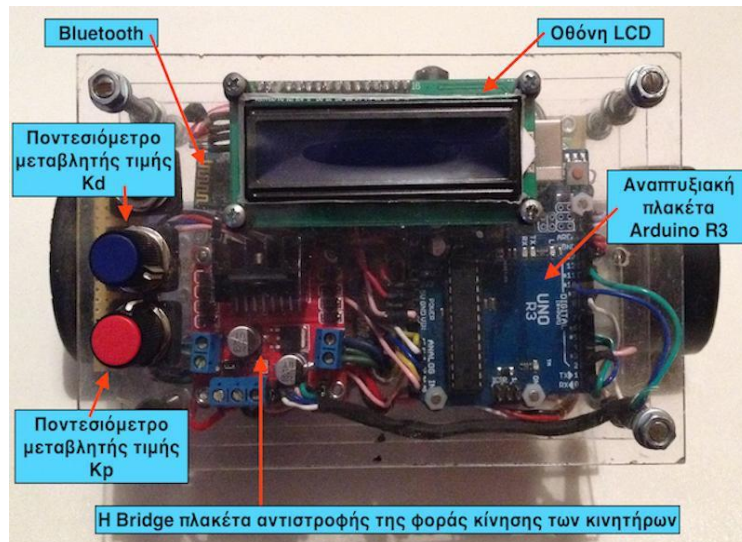
Εικόνα 6: Κύκλωμα σύνδεσης των LED και του πομποδέκτη υπέρυθρων.

Όλα τα εξαρτήματα της εφαρμογής ελέγχονται από τον μικροελεγκτή Atmega168 και συνδέονται στο αναπτυξιακό Arduino Diecimila. Η πλατφόρμα προγραμματισμού του Arduino είναι υπεύθυνη για τον προγραμματισμό του συστήματος. Το κεφάλαιο 3 περιλαμβάνει αναλυτικές αναφορές στον μικροελεγκτή Atmega καθώς και στο αναπτυξιακό Arduino.

2.3 Εφαρμογή ρομποτικού συστήματος αυτό-ισορρόπησης

Σε αυτή την ενότητα θα γίνει παρουσίαση μιας ρομποτικής κατασκευής με ικανότητα αυτοισορρόπησης σε δύο τροχούς χωρίς να επηρεάζεται από διαταραχές που μπορεί να την περιβάλλουν. Η κατασκευή αυτή ενδείκνυται για πλατφόρμα κατανόησης θεμάτων και προγραμματισμού σχετικά με την ρομποτική, τον προγραμματισμό και την μηχανοτρονική ηλεκτρονικών συσκευών. Το συγκεκριμένο σύστημα παρουσιάζεται στις παρακάτω εικόνες.





Εικόνα 7: Δίτροχο αυτόνομο ρομποτικό σύστημα αυτό –ισορρόπησης. [3]

Στο συγκεκριμένο ρομποτικό σύστημα υπάρχουν τρία επίπεδα κατασκευασμένα από το ίδιο υλικό (πλαστικό) και τα κομμάτια του έχουν ακριβώς τις ίδιες διαστάσεις. Το κάθε ένα από αυτά έχει μήκος 15,5cm, πλάτος 10cm και ύψος 0,03cm. Το υλικό κατασκευής είναι πλαστικό και επιλέχθηκε για τους εξής λόγους: είναι ευκολότερο στην επεξεργασία και ελαφρύτερο από άλλα υλικά. Προκειμένου να ενωθούν αυτά τα τρία πλαστικά επίπεδα, χρησιμοποιήθηκαν 24 παξιμάδια με τις αντίστοιχες ροδέλες και τέσσερις βίδες διαμέτρου 5mm. Στην κάτω πλευρά του πρώτου επιπέδου ενσωματώθηκαν οι βάσεις για τη στήριξη των κινητήρων οι οποίες είναι από μέταλλο και έχουν σχήμα Γ, όπως βλέπουμε και στην εικόνα 7. Σε αυτές τις βάσεις ενσωματώθηκαν οι κινητήρες και οι σύνδεσμοι που ενώνουν τον κινητήρα με τον άξονα και τον τροχό. Στη μέση του πρώτου επιπέδου τοποθετήθηκαν οι αισθητήρες για καλύτερες μετρήσεις καθώς και τέσσερις άξονες για να επιτευχθεί η στήριξη του ρομποτικού συστήματος χωρίς αυτό να βρίσκεται σε λειτουργία. Επιπλέον στην πάνω πλευρά του ίδιου επιπέδου τοποθετήθηκε η μπαταρία (πηγή ενέργειας της συσκευής), ο κεντρικός διακόπτης για την ενεργοποίηση και την απενεργοποίηση της συσκευής και ο σταθεροποιητής τάσης. Στο πάνω τμήμα του δεύτερου επιπέδου εφαρμόστηκε η διάταξη H-Bridge ή αλλιώς οδηγός των κινητήρων. Δίπλα από τον οδηγό των κινητήρων τοποθετήθηκε και το αναπτυξιακό Arduino το οποίο αποτελεί την κεντρική μονάδα του συστήματος, αφού συντονίζει και ελέγχει όλα τα μέρη και τις λειτουργίες που αποτελούν την κατασκευή. Η επικοινωνία γίνεται μέσω Bluetooth και είναι τοποθετημένη δίπλα στο αναπτυξιακό. Τέλος στο τελευταίο επίπεδο τοποθετήθηκαν τα ποτενσιόμετρα για τον έλεγχο των παραμέτρων του μικροελεγκτή και η οθόνη LCD.

Προκειμένου να ολοκληρωθεί η κατασκευή του ρομποτικού συστήματος, ήταν απαραίτητη αρχικά μια πλακέτα αναπτυξιακού Arduino. Πάνω στην πλακέτα είναι ενσωματωμένος ο μικροελεγκτής που προγραμματίζεται από τον κατασκευαστή του συστήματος, ώστε να μπορεί να οδηγεί όλα τα αισθητήρια, ενώ ο μικροελεγκτής είναι επίσης υπεύθυνος και για την διάταξη που οδηγεί τους τροχούς που αποτελούν τμήμα του ρομποτικού συστήματος. Πιο συγκεκριμένα, εφαρμόστηκε ο μικροελεγκτής Atmega328P-PU, ο οποίος έχει έναν 8 Bit μικροεπεξεργαστή. Το περιβάλλον προγραμματισμού, στο οποίο αναπτύσσονται οι αλγόριθμοι και τα προγράμματα αποτελούν ιδιοκτησία της εταιρείας Arduino. Προκειμένου να ρυθμίσουμε το ρομποτικό δίτροχο, έτσι ώστε να ισορροπεί, ήταν απαραίτητο να τοποθετηθούν αισθητήρες, για τη μέτρηση της επιτάχυνσης και της κλίσης, έτσι ώστε το ρομποτικό σύστημα να μπορεί να επεξεργάζεται τα δεδομένα και οδηγεί τους κινητήρες σε συνδυασμό με κατάλληλες στρατηγικές, ώστε να εξασφαλίζεται ισορροπία.

Το αισθητήριο που εφαρμόστηκε στο συγκεκριμένο ρομποτικό σύστημα, συνιστά ένα ολοκληρωμένο σύστημα πέντε βαθμών ελευθερίας. Πιο συγκεκριμένα, αποτελείται από ένα ένα γυροσκόπιο, δύο βαθμών ελευθερίας (IDG-650) και ένα επιταχυνσιόμετρο (ADXL-335) τριών βαθμών ελευθερίας. Αυτά τα αισθητήρια είναι υπεύθυνα να «παρακολουθούν» με επιτυχία την επιτάχυνση και την κλίση 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 ρομποτικό σύστημα έχει την ικανότητα να συνδέεται και να επικοινωνεί ασύρματα με απομακρυσμένο Η/Υ), για την επίτευξη διαφόρων λόγων. Προκειμένου να πραγματοποιηθεί η ασύρματη σύνδεση χρησιμοποιείται μια διάταξη Bluetooth και - πιο συγκεκριμένα - το μοντέλο HC-06. Επιπλέον, για να τροφοδοτηθεί με ενέργεια το ρομποτικό σύστημα εφαρμόζεται μια μπαταρία τεχνολογίας LiPo (Lithium-ion polymer battery, ιόντων λιθίου). Δεδομένου ότι δεν λαμβάνουν όλα τα ηλεκτρονικά μέρη του συστήματος την ίδια τάση, και η τάση της μπαταρίας δεν παραμένει σταθερή καθ'όλη την διάρκεια της λειτουργίας της, από πλήρως φορτισμένη μέχρι να εξαντληθεί, κατασκευάστηκε μια πλακέτα σταθεροποίησης της τάσης. Η λειτουργία της συγκεκριμένης διάταξης είναι να τροφοδοτεί με ενέργεια την ίδια στιγμή την πλακέτα του μικροεπεξεργαστή με τάση 9V και τα υπόλοιπα ηλεκτρονικά τμήματα του συστήματος με τάση 5V.

2.4 Τετράτροχο ρομποτικό σύστημα ελεγχόμενο ασύρματα από ένα επιταχυνσιόμετρο με τη βοήθεια πομποδέκτη.

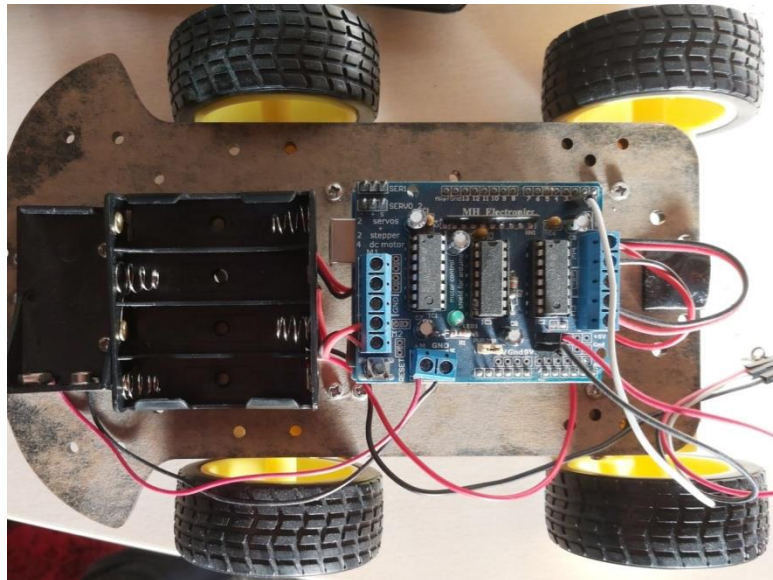
Η παρούσα διπλωματική εργασία αναφέρεται στην έρευνα, τη μελέτη και την κατασκευή ενός τετράτροχου ρομποτικού συστήματος με ασύρματη πλοήγηση. Το συγκεκριμένο τετράτροχο θα έχει τη δυνατότητα πλοήγησης με τη βοήθεια ενός πομποδέκτη RF 433Mhz καθώς και ενός επιταχυνσιόμετρου-γυροσκοπίου. Η ανάλυση της εφαρμογής χωρίζεται σε 3 μέρη. Το πρώτο μέρος έχει να κάνει με τον τρόπο σχεδίασης, αλλά και τα υλικά που ενσωματώθηκαν, με βάση τη διαθεσιμότητα, την ποιότητα και την οικονομία. Το δεύτερο μέρος έχει να κάνει με τον προγραμματισμό της συσκευής καθώς όλα τα επιμέρους εξαρτήματα του ρομποτικού συστήματος ελέγχονται από μικροελεγκτή, ο οποίος είναι υπεύθυνος για την λειτουργία του συστήματος. Το τελευταίο μέρος έχει να κάνει με τα συμπεράσματα που βγάλαμε, τις γνώσεις που αποκομίσαμε αλλά και τις αναβαθμίσεις που θα μπορούσαμε να κάνουμε στο μέλλον.

ΚΕΦΑΛΑΙΟ 3^ο

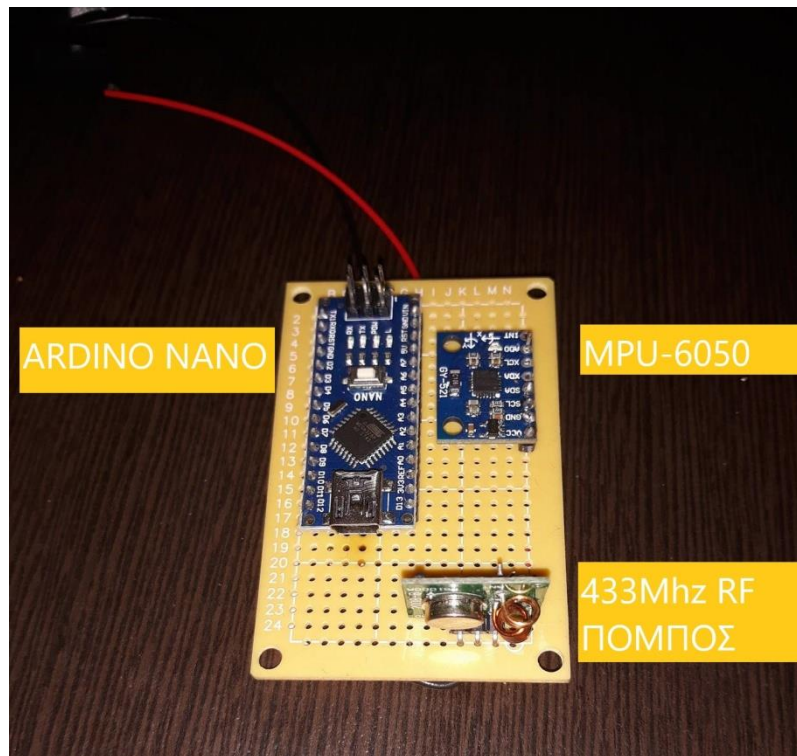
ΣΧΕΔΙΑΣΜΟΣ ΤΕΤΡΑΤΡΟΧΟΥ ΡΟΜΠΟΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΚΑΙ ΑΝΑΛΥΣΗ ΤΕΧΝΙΚΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΕΞΑΡΤΗΜΑΤΩΝ

3.1 Εισαγωγή

Όπως είπαμε και στην ενότητα 2.4 το βασικό στοιχείο στην κατασκευή του ρομποτικού συστήματος είναι ο ασύρματος τηλεχειρισμός μέσω πομποδέκτη και επιταχυνσιόμετρο-γυροσκόπιου. Για να λειτουργήσει αυτό τον κύριο ζήτημα ήταν η άριστη επικοινωνία του μικροελεγκτή με τα μοτέρ και τους αισθητήρες. Σε αυτό το κεφάλαιο θα αναφερθούμε στα εξαρτήματα που απαρτίζουν την κατασκευή. Τα συγκεκριμένα εξαρτήματα επιλέχθηκαν με οικονομικά κριτήρια, καθώς επίσης και με την ποιότητα αλλά και την διαθεσιμότητά τους. Αυτό συνέβη γιατί κατά την δημιουργία της κατασκευής υπήρξαν αρκετές αστοχίες και κάποια εξαρτήματα δεν είχαν την απαιτούμενη αντοχή. Παρακάτω ακολουθούν οι εικόνες 8 και όπου φαίνεται ο τρόπος κατασκευής αλλά και τα παρελκόμενα που την αποτελούν.



Εικόνα 8: Εικόνα ρομποτικού συστήματος.



Εικόνα 9: Εικόνα συστήματος ασύρματου τηλεχειρισμού.

3.2 Κεντρική μονάδα ελέγχου τετράτροχου

Το κομμάτι που είναι υπεύθυνο για την λειτουργία του τετράτροχου ρομπότ είναι το **ARDUINO UNO**. Το Arduino είναι ένα αναπτυξιακό σύστημα λειτουργώντας ως κεντρική μονάδα ελέγχου στην κατασκευή και βασίζεται στον μικροελεγκτή **ATmega328P**. Το **UNO** Ιταλικής κατασκευής σηματοδοτεί μια νέα εποχή με την κυκλοφορία του λογισμικού και του αναπτυξιακού καθώς αποτέλεσε το πρώτο με σειριακή σύνδεση (**USB**).

Ξεκίνησε το 2005 αρχικά ως συσκευή ελέγχου προγραμμάτων για μαθητές η οποία θα αποτελούσε μια οικονομική λύση από άλλα συστήματα που υπήρχαν εκείνη την περίοδο.

Η ονομασία του Arduino όπου δόθηκε από τους ιδρυτές Massimo Banzi και David Cueartielles προήλθε από την περιοχή Ιβρέα, μια κωμόπολη της επαρχίας Τορίνο στη βορειοδυτική Ιταλία όπου ξεκίνησε και η παραγωγή των πλακετών σε ένα μικρό εργοστάσιο στην περιοχή. Στην ίδια περιοχή στεγαζόταν και η Olivetti, μια εταιρία υπολογιστών.

Το Arduino αποτελεί μια παραλλαγή της υπολογιστικής πλατφόρμας Wiring όπου προγραμματίζεται χρησιμοποιώντας μια γλώσσα βασισμένη στο Wiring (σύνταξη και βιβλιοθήκες) και είναι ένα λογισμικό ανοικτού κώδικα, παρόμοια με την C++ με κάποιες παραλλαγές και απλοποιήσεις, καθώς και ένα ολοκληρωμένο περιβάλλον ανάπτυξης (**IDE**).

Η κατασκευή του βασίζεται σε έναν μικροελεγκτή ενσωματωμένο σε μια μητρική πλακέτα, με αναλογικές και ψηφιακές εισόδους και εξόδους. Χρησιμοποιείται για τη δημιουργία διαφόρων 'έξυπνων' συσκευών καθώς επάνω μπορούν να ενσωματωθούν διάφορων ειδών αισθητήρες πίεσης, φωτός, θερμοκρασίας, κινητήρες, οθόνες κ.λ.π. Επίσης τα τελευταία χρόνια χρησιμοποιείται και για εκπαιδευτικούς σκοπούς, για την υλοποίηση διάφορων κατασκευών.

Ουσιαστικά είναι ένα κύκλωμα που βασίζει την λειτουργία του στον μικροελεγκτή ATmega της Atmel καθώς το λογισμικό και τα σχέδια που χρειάζονται για να λειτουργήσει, υπάρχουν στο ίντερνετ δωρεάν και ελεύθερα ώστε να έχει την δυνατότητα κατασκευής από τον καθένα. Συμπεριφέρεται σαν ένας μικρός υπολογιστής, αφού υπάρχει η δυνατότητα από το χρήστη:

- σύνδεσης επάνω σε αυτό πολλαπλών μονάδων εισόδου/εξόδου,
- προγραμματισμού του μικροελεγκτή ώστε να δέχεται δεδομένα από τις μονάδες εισόδου,
- και να μπορεί να στέλνει και να επεξεργάζεται κατάλληλες εντολές για τις μονάδες εξόδου.

Το κύριο πλεονέκτημά του σε σχέση με άλλες διαδραστικές ηλεκτρονικές συσκευές είναι η τεράστια κοινότητα που έχει δημιουργήσει και το υποστηρίζει, συντηρεί και επεκτείνει μια μεγάλη διαδικτυακή γνωστική βάση και με εκτενή τεκμηρίωση που βοηθάει όποιον θέλει να δημιουργήσει έργο με αυτό

Ένα άλλο πλεονέκτημα του μικροελεγκτή είναι οι χαμηλές τιμές στις οποίες προσφέρεται τόσο ο μικροελεγκτής όσο και τα απαραίτητα εξαρτήματα.

I. Μικροελεγκτές

Σαν μικροελεγκτής ορίζεται ένα ολοκληρωμένο κύκλωμα το οποίο μπορεί να προγραμματιστεί καθώς διαθέτει μνήμη, επεξεργαστή κυκλώματα περιφερειακά καθώς επίσης και θύρες για την επικοινωνία με εξωτερικές συσκευές.

Οι κατασκευαστικές εταιρείες κατασκεύασαν τους μικροελεγκτές ώστε ένα σύστημα να έχει μεγαλύτερες δυνατότητες και μικρότερο μέγεθος και έτσι ενσωμάτωσαν όλες τις λειτουργίες ενός υπολογιστή σε ένα ή μερικά ολοκληρωμένα κυκλώματα.

II. Περιγραφή του μικροελεγκτή Arduino

Το **ATmega328P** είναι ένα υψηλής απόδοσης και χαμηλής ενέργειας AVR μικροελεγκτής των 8 bit. Η λειτουργία του γίνεται με τάση από 1,8V έως 5.5V. Ο βαθμός ταχύτητας του είναι 0-20MHz στην προαναφερόμενη τάση λειτουργίας. Το **ATmega328P** είναι εκ των

προτέρων προγραμματισμένος (Boot loader), ώστε να προγραμματίζεται εύκολα χωρίς την χρήση κάποιου εξωτερικού προγράμματος οδήγησης.

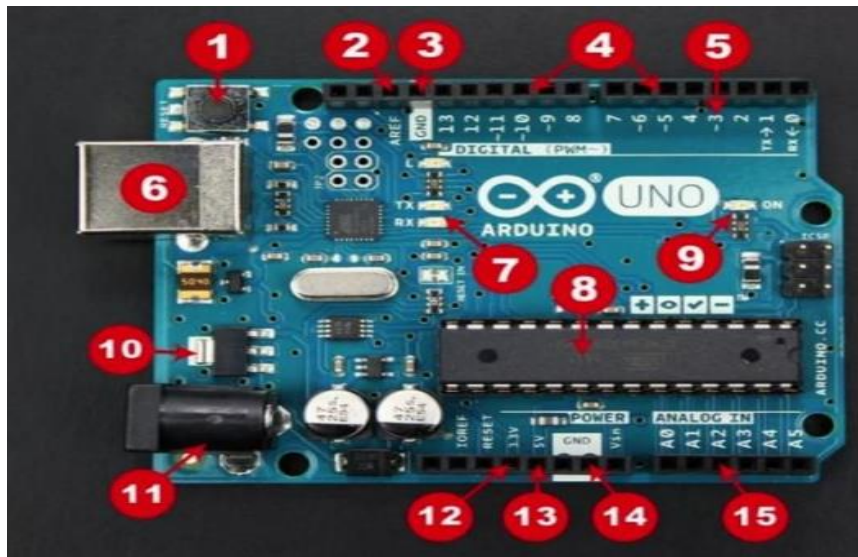
Ο ATmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων:

- 2Kb μνήμης SRAM είναι η μνήμη για την αποθήκευση μεταβλητών των προγραμμάτων που φορτώνονται σε αυτό.
- 1Kb μνήμης EEPROM χρησιμοποιείται για την εγγραφή/ανάγνωση κατά την εκτέλεση των προγραμμάτων.
- 32Kb μνήμης Flash, όπου βρίσκεται το πρόγραμμα που έχει φορτωθεί μέσω της USB θύρας.

Και στις τρεις μνήμες δεν χάνονται τα περιεχόμενα όταν δεν υπάρχει τροφοδοσία ή αν γίνει RESET.

Όπως προαναφέραμε, το ολοκληρωμένο **ATmega328P** αποτελεί την κεντρική μονάδα ελέγχου στο αναπτυξιακό **ARDUINO UNO** και λειτουργεί ως κεντρική μονάδα για τον έλεγχο του τετράτροχου ρομποτικού συστήματος. Όλα τα επιμέρους εξαρτήματα συντονίζονται μέσω της κεντρικής μονάδας, (DC μοτέρ, ασύρματη επικοινωνία ,motor shield κ.α.). Η ανάλυση των επιμέρους εξαρτημάτων θα γίνει μετά την ανάλυση του Arduino Uno παρακάτω.

3. Είσοδοι / Έξοδοι



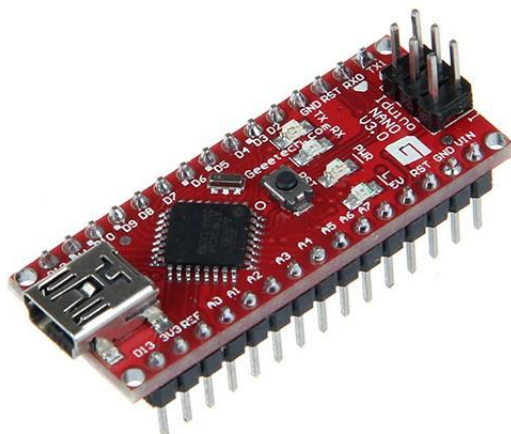
Εικόνα 10: Είσοδοι και έξοδοι της πλακέτας Arduino UNO [4]

1. Κουμπί Reset: Γίνεται επανεκκίνηση της πλακέτας του Arduino και συνεπώς του κώδικα που έχει ανέβει.
2. AREF: Υπάρχει για την ρύθμιση εξωτερικής τάσης αναφοράς .

3. Ακροδέκτης γείωσης.
4. Ψηφιακοί είσοδοι / έξοδοι: Οι ακροδέκτες 0-13 χρησιμοποιούνται ως ψηφιακή είσοδος ή έξοδος.
5. PWM: Είναι ακροδέκτες προσομοιάζουν την αναλογική έξοδο και υπάρχει το σύμβολο (~).
6. Σύνδεση USB: Χρησιμοποιείται για τον προγραμματισμό ή τροφοδοσία..
7. TX / RX: Ένδειξη LED εκπομπής και λήψης δεδομένων.
8. ATMEGA 328P Μικροελεγκτής: Είναι το κύριο εξάρτημα όπου χρησιμοποιείται για την εκτέλεση του κώδικα.
9. Ενδεικτική λυχνία ισχύος: Η λυχνία LED ανάβει όταν η πλακέτα είναι συνδεδεμένη σε πηγή τροφοδοσίας
10. Σταθεροποιητής τάσης.
11. Υποδοχή Jack DC: Χρησιμοποιείται για την τροφοδοσία του Arduino από τροφοδοτικό. 12. 3.3V ακροδέκτης: Παρέχει 3.3V.
13. 5V ακροδέκτης: Παρέχει 5V.
14. Ακροδέκτης γείωσης
15. Αναλογικοί ακροδέκτες: Οι ακροδέκτες αυτοί χρησιμοποιούνται για να διαβάσουν αναλογικό σήμα.

3.3 Κεντρική μονάδα ελέγχου τηλεχειρισμού

Το **Arduino Nano** είναι ουσιαστικά η μικρότερη έκδοση του **Arduino Uno** και είναι βασισμένο στον μικροελεγκτή **ATmega328** (**Arduino Nano 3.0**) ή το **ATmega168** (**Arduino Nano 2.x**). Οι διαφοροποιήσεις μεταξύ τους είναι ως προς το μέγεθος και ενός βύσματος τροφοδοσίας **DC** καθώς και το καλώδιο λειτουργίας **USB Mini-B** αντί για ένα τυπικό. Το **Nano** σχεδιάστηκε και παράγεται από τη **Gravitech**.



Εικόνα 11: Αναπτυξιακό Arduino Nano [5].

Η τροφοδοσία γίνεται μέσω της σύνδεσης **USB Mini-B**, ανεξάρτητης εξωτερικής τροφοδοσίας **6-20V (pin 30)** ή εξωτερικής τροφοδοσίας ισχύος **5V (pin 27)**. Η πηγή ισχύος εφαρμόζεται αυτόματα στην υψηλότερη πηγή τάσης.

Το τσιπ **FTDI FT232RL** στο **Nano** τροφοδοτείται μόνο αν η κάρτα τροφοδοτείται μέσω USB. Ως αποτέλεσμα, όταν εκτελείται εξωτερική (**μη-USB**) ισχύς, η έξοδος **3,3V** (η οποία παρέχεται από το τσιπ **FTDI**) δεν είναι διαθέσιμη και τα **LED RX** και **TX** θα τρεμοπαίζουν αν οι ψηφίδες 0 ή 1 είναι υψηλές.

Ο **ATmega168** είναι σχεδιασμένος με μνήμη **16 KB flash** ώστε να γίνεται η αποθήκευση κώδικα (εκ των οποίων τα **2 KB** αφορούν το **bootloader**). Αντίθετα το **ATmega328** περιέχει **32 KB**, (επίσης με **2 KB** για το **bootloader**). Το **ATmega168** έχει **1 KB SRAM** και **512 bytes EEPROM** (όπου διαβάζεται και να γράφεται με τη βιβλιοθήκη **EEPROM**). το **ATmega328** έχει **2 KB SRAM** και **1 KB EEPROM**.

Το **Arduino Nano** περιέχει 14 ψηφιακές ακίδες όπου χρησιμοποιούνται ως είσοδοι ή έξοδοι χρησιμοποιώντας λειτουργίες **pinMode**, **digitalWrite** και **digitalRead**. Η λειτουργία τους είναι στα **5V**. Κάθε ακροδέκτης παρέχει ή λαμβάνει έως **40 mA** και έχει εσωτερική αντίσταση **pull-up 20-50 kOhms**. Ακόμη, μερικά pin έχουν εξειδικευμένες λειτουργίες: Serial: 0 (**RX**) και 1 (**TX**). Χρησιμοποιείται για τη λήψη σειριακών δεδομένων **TTL (RX)** και εκπομπής (**TX**). Οι ακίδες αυτές συνδέονται με τους αντίστοιχους ακροδέκτες του σειριακού τσιπ **FTDI USB-to-TTL**. Εξωτερικές Διακοπές: 2 και 3. Αυτές οι ακίδες μπορούν να διαμορφωθούν για να προκαλέσουν διακοπή σε χαμηλή τιμή, άνοδο ή πτώση άκρης ή αλλαγή τιμής

PWM: 3, 5, 6, 9, 10 και 11. Έχουν έξοδο **PWM 8-bit** με τη λειτουργία αναλογική έξοδο.
SPI: 10 (**SS**), 11 (**MOSI**), 12 (**MISO**), 13 (**SCK**). Αυτές οι ακίδες υπάρχουν για επικοινωνία SPI, η οποία, παρόλο που υπάρχει, δεν ενσωματώνεται αυτή τη στιγμή στη γλώσσα **Arduino**.

LED: 13. Υπάρχει μια ενσωματωμένη λυχνία **LED** συνδεδεμένη στον ψηφιακό ακροδέκτη 13. Όταν ο ακροδέκτης στέλνει ή λαμβάνει δεδομένα, η λυχνία **LED** είναι αναμμένη, όταν ο ακροδέκτης δεν χρησιμοποιείται, είναι απενεργοποιημένος.

Στο **Nano** υπάρχουν 8 αναλογικές εισόδους, καθένα από τις οποίες παρέχει 10 ψηφία ανάλυσης (δηλαδή 1024 διαφορετικές τιμές). Από προεπιλογή, μετρούν από τη γείωση σε 5 βολτ. Ακόμη, μερικές από τις ακίδες έχουν εξειδικευμένη χρήση:

- **I2C: 4 (SDA) και 5 (SCL).** Υποστηρίζει την επικοινωνία **I2C (TWI)** χρησιμοποιώντας τη βιβλιοθήκη **Wire**.

Υπάρχουν μερικές άλλες ακίδες:

- **AREF.** Είναι για χρήση ως τάση αναφοράς για τις αναλογικές εισόδους.
- **RESET** Χρησιμοποιείται για να κάνει **reset** τον μικροελεγκτή. Συνήθως χρησιμοποιείται για να προσθέσετε ένα κουμπί επαναφοράς σε ασπίδες.

Το **Arduino Nano** μπορεί να προγραμματιστεί με το λογισμικό **Arduino**. Επιλέξτε "**Arduino Diecimila, Duemilanove ή Nano w / ATmega168**" ή "**Arduino Duemilanove ή Nano w / ATmega328**" από το μενού **Tools > Board** (σύμφωνα με τον μικροελεγκτή στο ολοκληρωμένο σας).

Ο μικροελεγκτής **ATmega168** ή **ATmega328** του **Arduino Nano** περιέχει προεγκατεστημένο ένα **bootloader** που επιτρέπει να φορτώσουμε νέο κώδικα χωρίς τη χρήση εξωτερικού προγραμματιστή υλικού. Επικοινωνεί με το αρχικό πρωτόκολλο **STK500**.

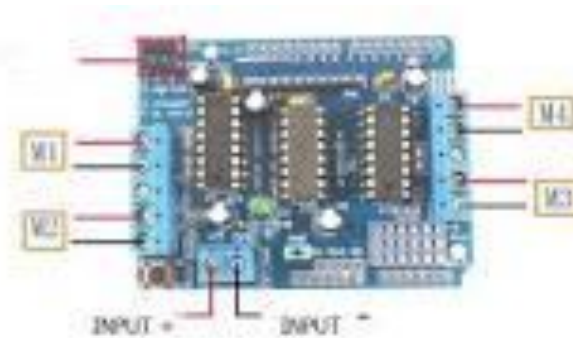
Υπάρχει και η δυνατότητα παράκαμψης του **bootloader** για προγραμματισμό του μικροελεγκτή μέσω της κεφαλίδας **ICSP** (σειριακό προγραμματισμό εντός κυκλώματος).

Αντίθετα, με τη χρήση του κουμπιού επαναφοράς πριν από τη μεταφόρτωση, το **Arduino Nano** επιτρέπει την επαναφορά του από το λογισμικό που μπορεί να εκτελείται σε συνδεδεμένο υπολογιστή. Μία από τις γραμμές ελέγχου ροής υλικού (**DTR**) του **FT232RL** συνδέεται στη γραμμή επαναφοράς του **ATmega168** ή **ATmega328** μέσω ενός πυκνωτή 100 **nanofarad**. Όταν η γραμμή αυτή επαναφοράς πέσει αρκετά επαναφέρει το τσιπ σε μια αρχική κατάσταση. Η δυνατότητα αυτή χρησιμοποιείται από το **Arduino** ώστε να μπορούμε να ανεβάσουμε κώδικα πατώντας απλά το κουμπί φόρτωσης στο περιβάλλον **Arduino**. Αυτό σημαίνει ότι ο **bootloader** μπορεί να έχει μικρότερο χρονικό όριο, καθώς η μείωση του **DTR** συντονίζεται καλύτερα με την έναρξη της φόρτωσης.

Παρόλα αυτά υπάρχουν επιπτώσεις σε αυτή τη ρύθμιση. Κατά τη σύνδεση του **Nano** με υπολογιστή που έχει λειτουργικό **Mac OS X** ή **Linux** κάθε φορά που γίνεται σύνδεση με

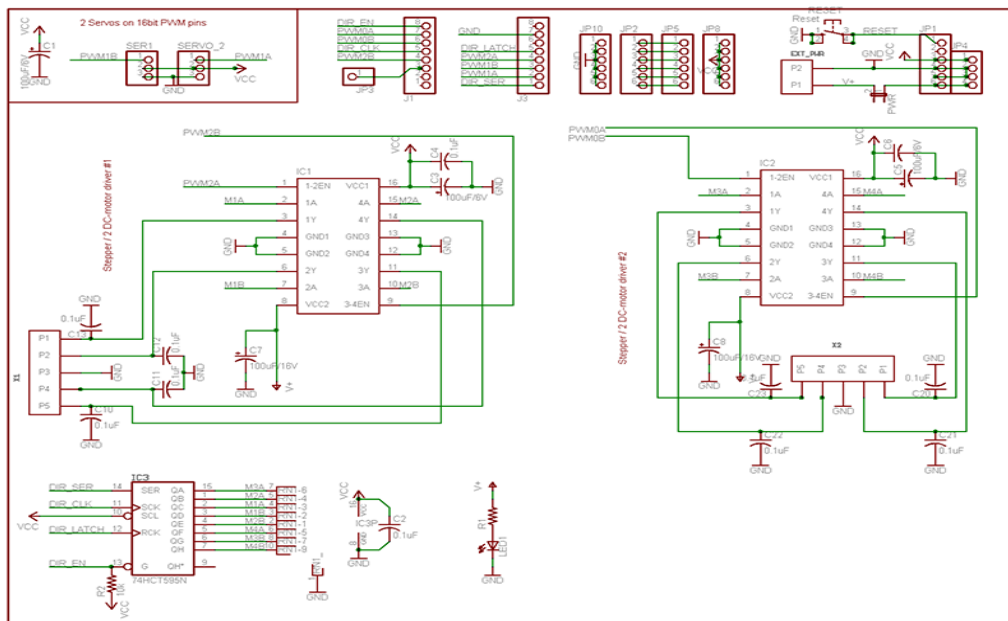
το λογισμικό (μέσω **USB**) γίνεται επαναφορά στις αρχικές του ριθμύσεις. Για περίπου μισό δευτερόλεπτο, ο **bootloader** τρέχει στο **Nano**. Παρόλο που είναι προγραμματισμένο να απορρίπτει τα παραμορφωμένα δεδομένα (δηλ. Οτιδήποτε εκτός από τη μεταφόρτωση νέου κώδικα), παρεμποδίζει την είσοδο των πρώτων **bytes** τα οποία είναι δεδομένα που στέλλονται στην πλακέτα μετά την σύνδεση. Άρα σε μια διαμόρφωση που λαμβάνει δεδομένα κατά την πρώτη εκκίνηση πρέπει να βεβαιωθούμε ότι αναμένει ένα δευτερόλεπτο μετά το άνοιγμα της σύνδεσης με το λογισμικό το οποίο επικοινωνεί και πριν την αποστολή δεδομένων

3.4 L293D Motor Driver Shield



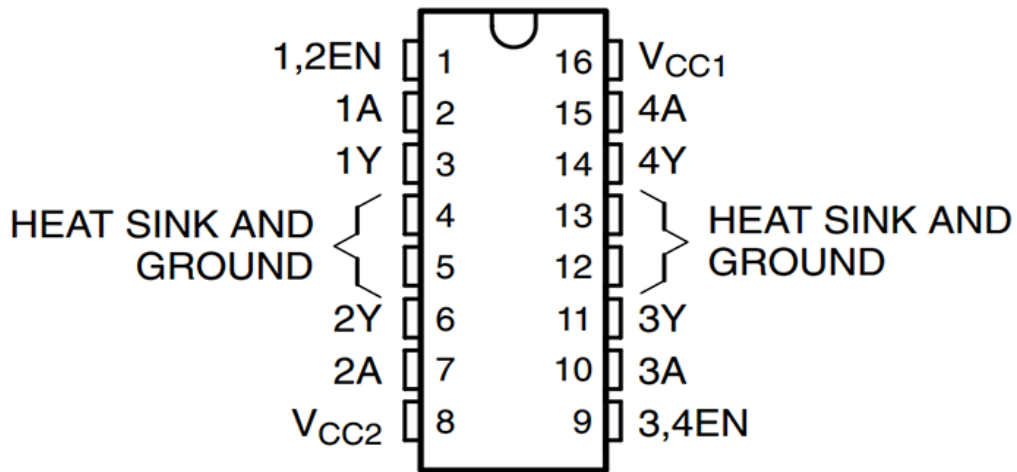
Εικόνα 12: Πλάκετα Motor Driver Shield [6].

Το **SunFounder L293D** μεταφράζεται ως ένας μονολιθικός οδηγός υψηλής τάσης και ρεύματος 4 καναλιών. Με απλά λόγια σημαίνει ότι χρησιμοποιώντας αυτό το τσιπ μπορείτε να συνδέσετε DC κινητήρες και τροφοδοτικά έως **16 Volts**, γι αυτό το λόγο το τσιπ μπορεί να παρέχει ένα μέγιστο ρεύμα **600mA** ανά κανάλι(για μεγάλους κινητήρες), το τσιπ **L293D** είναι επίσης αυτό που είναι γνωστό ως Η -Γέφυρα. Η γέφυρα Η είναι ουσιαστικά ένα ηλεκτρικό κύκλωμα που εφαρμόζει τάση σε ένα φορτίο προς μία έξοδο, π.χ. μοτέρ. Το σχηματικό διάγραμμα έχει ως εξής:

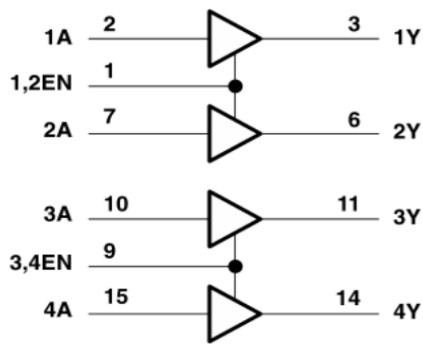


Εικόνα 13: Κύκλωμα γέφυρας H Motor Shield [7].

Το **L293D** είναι τετραπλός οδηγός ημιαγωγού υψηλού ρεύματος. Είναι σχεδιασμένος ώστε να παρέχει σε τάσεις από **4,5 V** έως **36 V** αμφίδρομα ρεύματα κίνησης έως και **600 mA**. Και οι δύο συσκευές έχουν σχεδιαστεί για να κινούν επαγωγικά φορτία όπως, ηλεκτρομαγνητικές βαλβίδες, ηλεκτρονόμους, διπολικούς βηματικούς κινητήρες και dc, καθώς και άλλα φορτία υψηλής τάσης / υψηλού ρεύματος σε κατασκευές θετικής παροχής. Οι εισοδοί είναι όλοι συμβατές με **TTL**. Η κάθε έξοδος αποτελεί ένα πλήρες κύκλωμα κίνησης **totem pole**, με μια πηγή ψευδο-**Darlington** και έναν τρανζίστορ **Darlington**. Η ενεργοποίηση των οδηγών γίνεται ανά ζεύγη, με τους οδηγούς 1 και 2 ενεργοποιημένους από 1,2EN και τους οδηγούς 3 και 4 ενεργοποιημένους από 3,4EN. Οι έξοδοι των προγραμμάτων οδήγησης που είναι συνδεδεμένα είναι ενεργές και σε φάση με τις εισόδους τους όταν η είσοδος ενεργοποίησης είναι υψηλή. Αντίθετα όταν είσοδος ενεργοποίησης είναι χαμηλή, αυτοί οι οδηγοί είναι απενεργοποιημένοι και οι έξοδοι τους είναι απενεργοποιημένοι και στην κατάσταση υψηλής σύνθετης αντίστασης. Με τις κατάλληλες εισόδους δεδομένων, κάθε ζευγάρι οδηγών δημιουργεί έναν αναστρέψιμο κινητήρα πλήρους-H (ή γέφυρας) κατάλληλη για εφαρμογές σωληνοειδών ή κινητήρων.



Εικόνα 14: Κύκλωμα L293D [8].



Logic diagram

FUNCTION TABLE
(each driver)

INPUTS [†]		OUTPUT Y
A	EN	
H	H	H
L	H	L
X	L	Z

H = high level, L = low level, X = irrelevant,
Z = high impedance (off)

[†] In the thermal shutdown mode, the output is in the high-impedance state, regardless of the input levels.

Function Table

Εικόνα 15: Λειτουργία ακροδεκτών [9].

Pin	Name	Function
1	Enable1,2	Enable pin to control 1,2 driver
2	Input 1A	Input to control 1Y
3	Output 1Y	Output,connect to motor
4	GND	Ground and heat sink
5	GND	Ground and heat sink
6	Output 2Y	Output,connect to motor
7	Input 2A	Input to control 2Y
8	Vcc2	Output supply voltage
9	Enable3,4	Enable pin to control 3,4 driver
10	Input 3A	Input to control 3Y
11	Output 3Y	Output,connect to motor
12	GND	Ground and heat sink
13	GND	Ground and heat sink
14	Output 4Y	Output,connect to motor
15	Input 4A	Input to control 4Y
16	Vcc1	Supply voltage(7 max)

Χαρακτηριστικά:

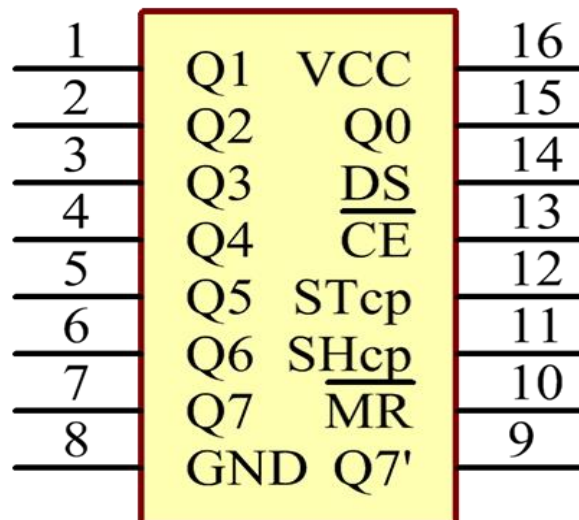
- Ευρεία περιοχή παροχής τάσης: 4,5 V έως 36 V
- Ξεχωριστή εισαγωγή-λογική τροφοδοσία
- Εσωτερική προστασία ESD
- Θερμική διακοπή λειτουργίας
- High-Noise-Immunity Inputs
- Ρεύμα εξόδου 600 mA ανά κανάλι
- Μέγιστο ρεύμα εξόδου 1,2 A ανά κανάλι

Ο **74HC595** περιέχει έναν καταχωρητή μετατόπισης **8 bit** και έναν καταχωρητή αποθήκευσης με παράλληλες εξόδους τριών καταστάσεων. Χρησιμοποιείται για τη μετατροπή της σειριακής εισόδου σε παράλληλη έξοδο, για την αποθήκευση των θυρών **IO** μιας μονάδας MCU. Το **74HC595** χρησιμοποιείται ευρέως για την ένδειξη **LED** πολλαπλών διαδρομών και για την προβολή οθονών πολλαπλών δυαδικών ψηφίων. Με τον όρο "τριφασικό" εννοούμε ότι μπορούμε να ορίσουμε τις ακίδες εξόδου ως υψηλή, χαμηλή ή "υψηλή αντίσταση". Με τη μανδάλωση δεδομένων, η στιγμιαία έξοδος δεν θα επηρεαστεί

κατά τη μετατόπιση. με έξοδο δεδομένων, μπορούμε να καταργήσουμε εύκολα το **74HC595**.



Εικόνα 16: Ολοκληρωμένο 74HC595 [10].



Εικόνα 17: Λειτουργία Pin [11].

No	Name	Function
1	Q0- Q7	8-bit parallel data output pins, able to control 8 LEDs or 8 pins of 7-segment display directly.
2	Q7'	Series output pin, connected to DS of another 74HC595 to connect multiple 74HC595s in series
3	MR	Reset pin, active at low level; here it is directly connected to 5V.

4	SHcp	Time sequence input of shift register. On the rising edge, the data in shift register moves successively one bit i.e. data in Q1 moves to Q2, and so forth. While on the falling edge, the data in shift register remain unchanged.
5	STcp	Time sequence input of storage register. On the rising edge, data in the shift register moves into memory register.
6	OE	Output enable pin, active at low level
7	DS	Serial data input pin
8	VCC	Positive supply voltage
9	GND	Ground

Χαρακτηριστικά:

- 8-Bit Serial-In, παράλληλη-Out Shift
- Ευρεία περιοχή τάσης λειτουργίας από 2 V έως 6 V
- Οι έξοδοι υψηλού ρεύματος 3 καταστάσεων μπορούν να μεταφέρουν έως 15 φορτία LSTTL
- Χαμηλή κατανάλωση ενέργειας, μέγιστη ICC 80 μ A
- Τυπικό $t_{pd} = 13$ ns
- Ρεύμα χαμηλής εισόδου 1
- Το Shift Register έχει άμεση διαγραφή

3.5 MPU-6050

Ο αισθητήρας **InvenSense MPU-6050** αποτελείται από ένα **MEMS** επιταχυνσιόμετρο και ένα γυροσκόπιο **MEMS** σε ενιαίο τσιπ. Έχει υψηλή ακρίβεια, καθώς διαθέτει ψηφιακό υλικό μετατροπής 16 ψηφίων για κάθε κανάλι. Συνεπώς, έχει την δυνατότητα καταγραφής ταυτόχρονα στα κανάλια x, y και z. Ο αισθητήρας συνδέεται με το **Arduino** μέσω του δίαυλου **I2C**. Το **MPU-6050** δεν είναι ακριβό, ειδικά εάν αναλογιστούμε ότι συνδυάζει και επιταχυνσιόμετρο και γυροσκόπιο.



Εικόνα 18: MPU-6050 [12].

ΟΙ πρώτες τιμές του επιταχυνσιόμετρου και του γυροσκοπίου είναι εύκολο να αναγνωστούν. Αρχικά πρέπει να απενεργοποιηθεί η λειτουργία ύπνου και, στη συνέχεια, μπορεί να γίνει η ανάγνωση των καταχωρητών για το επιταχυνσιόμετρο και το γυροσκόπιο. Ακόμη ο αισθητήρας περιέχει ένα **buffer 1024 byte FIFO** όπου μπορούμε να προγραμματίσουμε τις τιμές του αισθητήρα για να τοποθετηθούν στο **buffer FIFO** και το **buffer** να μπορεί να διαβαστεί από το **Arduino**.

Το **buffer FIFO** λειτουργεί μαζί με το σήμα διακοπής έτσι ώστε εάν το **MPU-6050** τοποθετήσει τα δεδομένα στο **buffer FIFO**, δίνει σήμα στο **Arduino** με το σήμα διακοπής έτσι ώστε το **Arduino** να γνωρίζει την ύπαρξη δεδομένων στο **buffer FIFO** που περιμένουν να διαβαστούν.

Η δυνατότητα ελέγχου μιας δεύτερης συσκευής **I2C** είναι λίγο πιο περίπλοκη.

Το **MPU-6050** λειτουργεί πάντα σύμφωνα με τις εντολές του **Arduino** που είναι συνδεδεμένο στους ακροδέκτες **SDA** και **SCL** που είναι συνδεδεμένοι στο δίαυλο **I2C**.

Εκτός από τον κανονικό δίαυλο **I2C**, περιέχει έναν δικό του ελεγκτή **I2C** λειτουργώντας ως κύριος σε ένα δεύτερο (υπο) **-I2C-bus**. Χρησιμοποιεί τους ακροδέκτες **AUX_DA** και **AUX_CL** για το δεύτερο (υπο) **-I2C-bus**.

Μπορεί για παράδειγμα να ελέγχει ένα μαγνητόμετρο οι τιμές του οποίου μπορούν να μεταφερθούν στο **Arduino**. Τα πράγματα γίνονται πολύπλοκα με το "**DMP**".

Ο ψηφιακός επεξεργαστής κινήσεων (**DMP**), ο οποίος ονομάζεται επίσης "Ψηφιακή μονάδα επεξεργασίας κίνησης" έχει τη δυνατότητα προγραμματισμού με firmware και μπορεί να κάνει σύνθετους υπολογισμούς με τις τιμές των αισθητήρων.

Για το **DMP**, η **InvenSense** εφαρμόζει μια πολιτική αποθάρρυνσης, επειδή δεν παρέχει επαρκείς πληροφορίες για τον προγραμματισμό του **DMP**. Ωστόσο, το υλικολογισμικό μπόρεσε να καταγραφεί από μερικούς με την αντίστροφη μηχανική.

Το **DMP** ("**Digital Motion Processor**") κάνει γρήγορους υπολογισμούς απευθείας στο τσιπ μειώνοντας έτσι το φορτίο για τον μικροελεγκτή (όπως το **Arduino**). Μια ακόμη καινοτομία του **DMP** είναι ότι μπορεί να κάνει υπολογισμούς με τις τιμές των αισθητήρων ενός άλλου τσιπ, όπως ένα μαγνητόμετρο συνδεδεμένο με το δεύτερο (υπο) **-I2C-bus**.

Οι τιμές του επιταχυνσιόμετρου και του γυροσκοπίου ονομάζονται "ακατέργαστες" τιμές. Μια άλλη εφαρμογή είναι υπεύθυνη για την ανάκτηση συγκεκριμένων υπολογιζόμενων τιμών από τον αισθητήρα μέσω του **DMP**.

3.6 RF 433 Μονάδα ραδιοσυχνότητας

Μια μονάδα RF (μονάδα ραδιοσυχνότητας) είναι μια ηλεκτρονική συσκευή που λειτουργεί ως συσκευή μετάδοσης ή λήψης ραδιοσημάτων μεταξύ δύο συσκευών. Η χρήση μιας τέτοιας συσκευής είναι συχνά αναγκαία για την επικοινωνία σε ένα ενσωματωμένο σύστημα. Αυτή η ασύρματη επικοινωνία επιτυγχάνεται είτε μέσω οπτικής επικοινωνίας είτε μέσω επικοινωνίας ραδιοσυχνότητας (RF). Η χρήση της μονάδας RF είναι πιο συχνή, καθώς δεν απαιτεί οπτική επαφή. Η μονάδα RF αποτελείται από ένα πομπό και ένα δέκτη. Υπάρχουν διάφορων τύπων και σειρών. Μερικοί από αυτούς έχουν τη δυνατότητα μετάδοσης έως και 150 μέτρα περίπου. Η χρήση των μονάδων RF είναι ευρεία στον ηλεκτρονικό σχεδιασμό καθώς υπάρχει δυσκολία στον σχεδιασμό ραδιοκυμάτων.

Ο σχεδιασμός των ηλεκτρονικών ραδιοσυχνότητας είναι αρκετά πολύπλοκος καθώς υπάρχει μεγάλη ευαισθησία των ραδιοκυκλωμάτων και της ακρίβειας των εξαρτημάτων και των διατάξεων που χρησιμοποιούνται ώστε να επιτευχθεί η λειτουργία σε συγκεκριμένη συχνότητα. Επιπρόσθετα, για την διασφάλιση του μη επηρεασμού αρνητικά των επιδόσεων RF απαιτείται προσεκτική παρακολούθηση της διαδικασίας κατασκευής.

Τέλος, λόγω ακτινοβολούμενων εκπομπών, απαιτούνται δοκιμές συμμόρφωσης και πιστοποίησης από κάποιον οργανισμό συμμόρφωσης όπως το ETSI ή την Ομοσπονδιακή Επιτροπή Επικοινωνιών των ΗΠΑ (FCC). Για αυτό το λόγο, οι μηχανικοί σχεδιασμού σχεδιάζουν πρώτα ένα κύκλωμα για μια εφαρμογή που χρειάζεται την ραδιοεπικοινωνία και έπειτα ενσωματώνει μια προκατασκευασμένη μονάδα ραδιοφώνου από το να προσπαθήσει ένα διακριτό σχέδιο, εξοικονομώντας χρόνο και χρήμα για την ανάπτυξη.

Η χρήση των μονάδων RF γίνεται από μικρό-μεσαίες συσκευές που χρησιμοποιούνται για ανοίγματα θυρών γκαράζ, ασύρματα συστήματα συναγερμού ή παρακολούθησης,

βιομηχανικά τηλεχειριστήρια, έξυπνες εφαρμογές αισθητήρων και ασύρματα συστήματα αυτοματισμού στο σπίτι. Εμφανίζονται συνήθως ως αντικαταστάτες παλαιότερων σχεδίων υπέρυθρης επικοινωνίας, καθώς πλεονεκτούν στο γεγονός ότι δεν απαιτείται λειτουργία γραμμής όρασης.

Υπάρχουν ποικίλες συχνότητες φορέων όπου χρησιμοποιούνται συνήθως σε εμπορικά διαθέσιμες μονάδες RF, συμπεριλαμβανομένων και εκείνων στις βιομηχανικές, επιστημονικές και ιατρικές ζώνες ραδιοσυχνοτήτων (ISM) όπως 433,92 MHz, 915 MHz και 2400 MHz. Λόγω των εθνικών και διεθνών κανονισμών που διέπουν τη χρήση του ραδιοφώνου για επικοινωνία χρησιμοποιούνται οι συχνότητες που είδαμε πιο πάνω. Για τις συσκευές μικρής εμβέλειας υπάρχει η δυνατότητα διαθέσιμων συχνότητων για μη εξουσιοδοτημένους χρήστες, όπως 315 MHz και 868 MHz.

Ο όρος RF module εφαρμόζεται σε διάφορους τύπους, μορφές και μεγέθη μικρών κυκλωμάτων ηλεκτρονικών υποσυστημάτων. Υπάρχει επίσης και η δυνατότητα εφαρμογής σε ενότητες σε μια ποικιλία λειτουργιών και δυνατοτήτων. Τα δομοστοιχεία RF αποτελούνται ουσιαστικά από ένα κύκλωμα τυπωμένου κυκλώματος, ένα κύκλωμα εκπομπής ή λήψης, μια σειριακή διεπαφή και μια κεραία ώστε να επικοινωνεί με τον κεντρικό επεξεργαστή.

Οι πιο συνηθισμένοι, γνωστοί τύποι καλύπτονται εδώ:

- Μονάδα πομπού
- Μονάδα λήψης
- Μονάδα πομποδέκτη
- Σύστημα σε μια μονάδα τσιπ

Στη συγκεκριμένη διπλωματική χρησιμοποιείται μια μονάδα RF 433Mhz. Η συγκεκριμένη μονάδα λήψης RF 433Mhz είναι μια μονάδα ραδιοφωνικού δέκτη με χαμηλή κατανάλωση και υψηλή ευαισθησία. Υπάρχει εσωτερική αποκωδικοποίηση λογισμικού και ενσωματώνει ένα κύκλωμα αποκωδικοποίησης δεδομένων, χωρίς πρόσθετο κύκλωμα για την υλοποίηση της εισόδου σήματος και της εξόδου δεδομένων. Οι προγραμματιστές χρησιμοποιούν αυτή τη μονάδα για την ανάπτυξη λειτουργίας ασύρματης επικοινωνίας μέσω αυτής της μονάδας.

Χαρακτηριστικά:

- Εύρος τάσης εισόδου: 3.3V ~ 5.0V
- Χαμηλή κατανάλωση ισχύος: 5V @ 433.92M, 3.8mA ~ 6mA
- Συχνότητα λειτουργίας: 433,92 Mhz (μπορεί να προσαρμοστεί σε 315M, 390M, 418M κ.λπ.)
- Διαμόρφωση: ASK
- Λειτουργία εξόδου: λειτουργία L (εισαγωγή έγκυρου σήματος, έξοδος αντίστοιχου σήματος, εξαφάνιση έγκυρου σήματος, έξοδος αυτόματης διακοπής)
- Υποστήριξη πολλαπλών ελεγκτών, όπως ελεγκτής με τσιπ PT2260, PT2262, PT2264, EV1527, κ.λπ.
- Καλή καταστολή της ακτινοβολίας τοπικού ταλαντωτή, χωρίς παρεμβολή όταν εκπέμπεται μεμονωμένη μετάδοση και πολλαπλή λήψη



Εικόνα 19: RF 433Mhz [13].

3.7 Μηχανολογικά μέρη και εξαρτήματα του τετράτροχου ρομποτικού συστήματος

Όπως είπαμε και στην αρχή του κεφαλαίου, τα εξαρτήματα του συστήματος, τοποθετήθηκαν με βάση την αντοχή τους, το κόστος τους και την διαθεσιμότητα τους. Η αγορά του σασί έγινε αποκλειστικά με βάση την αντοχή του, καθώς θα λέγαμε ότι αποτελεί το σκελετό του αμαξιδίου, αφού πάνω σε αυτό τοποθετούνται όλα τα επιμέρους εξαρτήματα. Για παράδειγμα Motor Driver Shield, Arduino Uno, δέκτης RF 433Mhz και μπαταρίες. Χρησιμοποιήσαμε ένα σασί για την κατασκευή μας καθώς η πλακέτα του Motor Shield ενσωματώνεται πάνω από το Arduino Uno. Με αυτό τον τρόπο έγινε καλύτερη οργάνωση στην καλωδίωση και στην τοποθέτηση των εξαρτημάτων. Επίσης

λόγω της εξοικονόμησης χώρου υπάρχει η δυνατότητα μελλοντικής αναβάθμισης στο σύστημα.

Για την καλύτερη ενσωμάτωση και ευκολία συναρμολόγησης επιλέχθηκαν προκατασκευασμένοι κινητήρες ώστε να μπορούν να ενσωματώνονται με τις ρόδες. Για λόγους βάρους αλλά και καλύτερης ευκινησίας το υλικό κατασκευής των κινητήρων και τον τροχών που επιλέχθηκε είναι το πλαστικό. Οι κινητήρες έχουν τάση λειτουργίας 5 έως 12V και απαιτούν ρεύμα 150mA. Στην εικόνα 20 που ακολουθεί φαίνεται ένα ανάλογο σασί με τους 4 κινητήρες. Ενώ στην εικόνα 21 βλέπουμε τον τύπο ρόδας που χρησιμοποιήθηκε στο τετράτροχο όχημα.



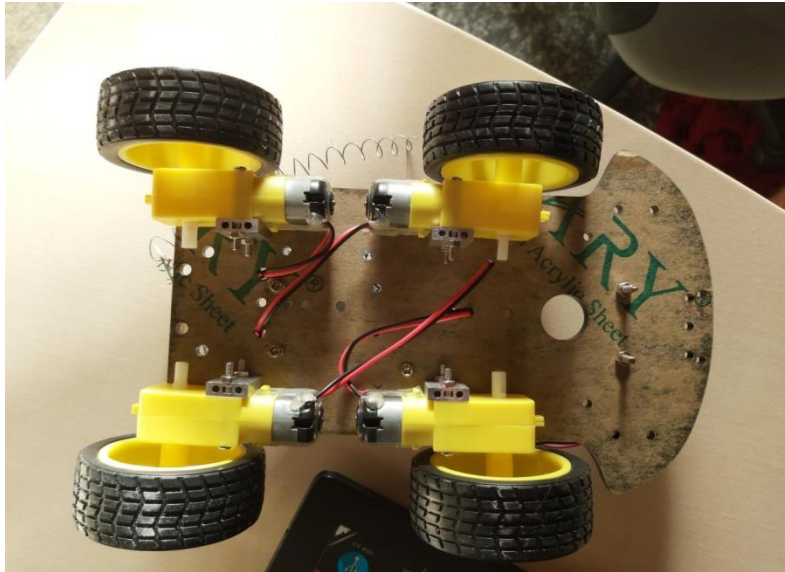
Εικόνα 20: Σασί αυτοκινήτου μαζί με τους κινητήρες και τη θήκη μπαταριών[14].



Εικόνα 21: Κινητήρας μαζί με τροχό του συστήματος[15].

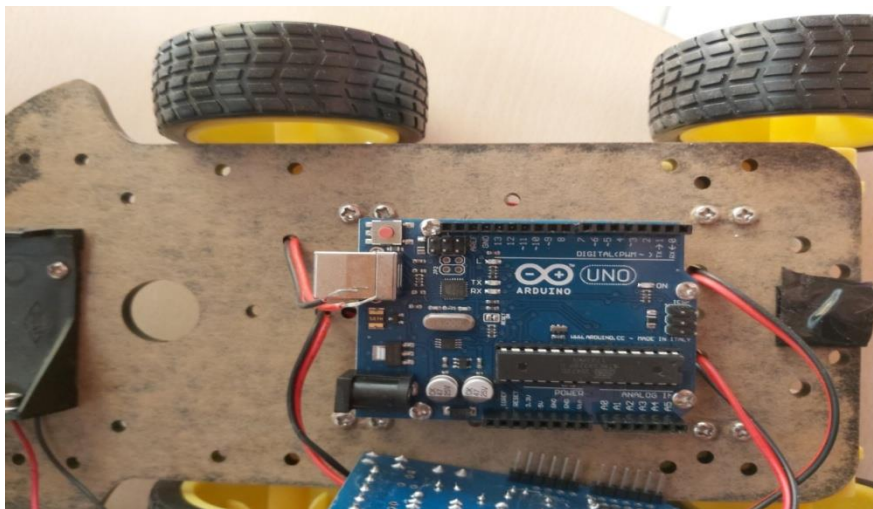
3.8 Συναρμολόγηση εξαρτημάτων αμαξιδίου.

Η συναρμολόγηση των εξαρτημάτων του αμαξιδίου είναι αρκετά εύκολη και απλή. Αρχικά βιδώνουμε επάνω στο σασί τους κινητήρες με τους τροχούς περνώντας τα καλώδια στην άλλη πλευρά ώστε να συνδεθούν επάνω στο **L293D Motor Driver Shield**.



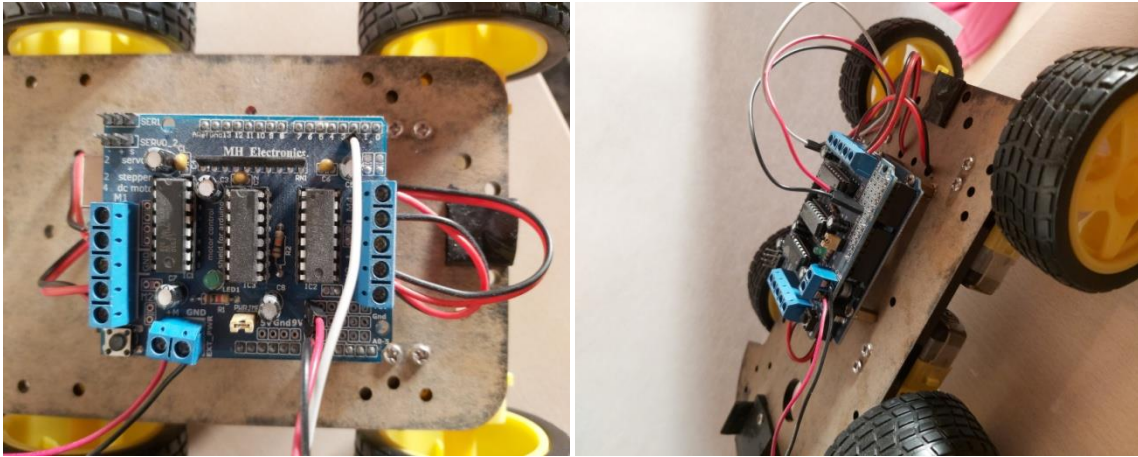
Εικόνα 22: Συναρμολόγηση τροχών και τοποθέτηση πάνω στο σασί του αμαξιδίου.

Στη συνέχεια βιδώνουμε από την άλλη πλευρά το **Arduino Uno** όπου είναι υπεύθυνο για τη σύνδεση όλων των εξαρτημάτων πάνω σε αυτό και την σωστή λειτουργία όλων μέσω του κατάλληλου προγραμματισμού και της σωστής σύνδεσης.



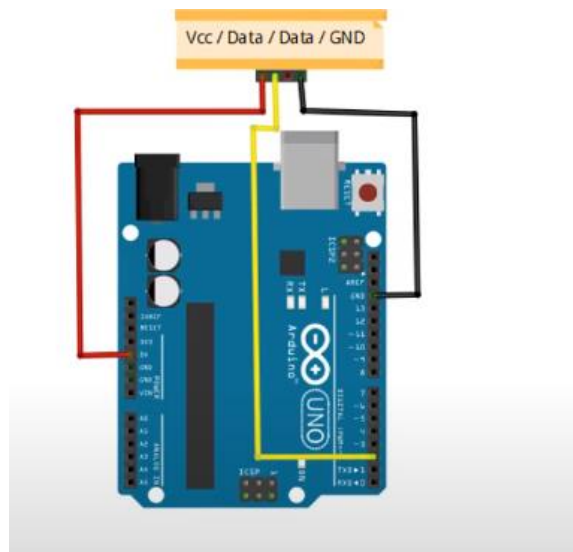
Εικόνα 23: Εγκατάσταση **Arduino Uno** πάνω στο σασί.

Το επόμενο βήμα είναι η προσθήκη του στο **L293D Motor Driver Shield** όπου εφαρμόζει ακριβώς από πάνω από το **Arduino Uno** και είναι υπεύθυνο για τη λειτουργία των κινητήρων του τετράτροχου που συνδέονται πάνω σε αυτό.

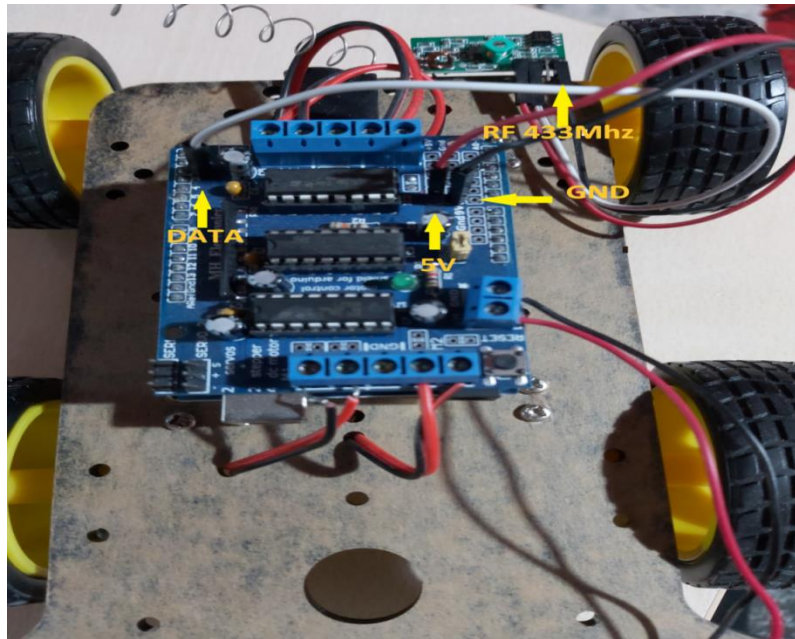


Εικόνα 24: Εφαρμογή **L293D Motor Driver Shield** πάνω στο **Arduino Uno** και σύνδεση καλωδίων κινητήρων.

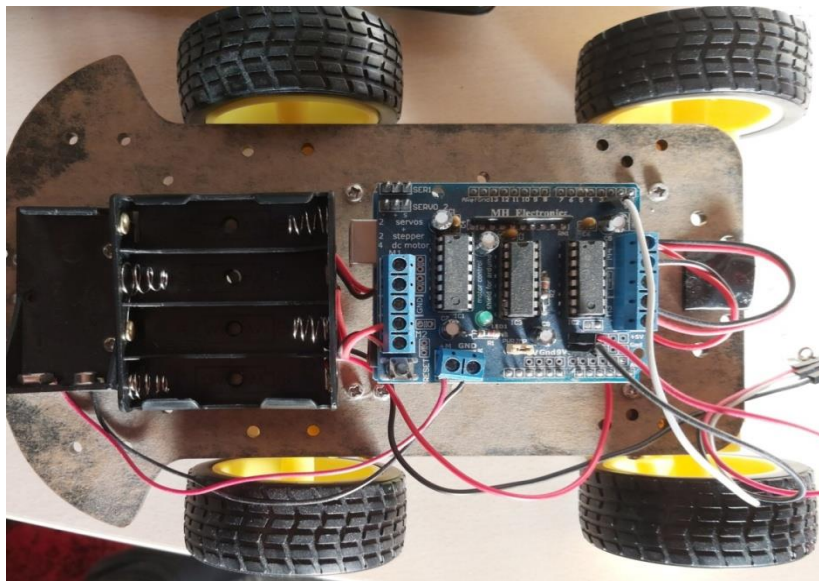
Τέλος για την ολοκλήρωση της συναρμολόγησης του τετράτροχου συνδέουμε τη μονάδα λήψης RF 433Mhz όπου δέχεται τα σήματα που στέλνουμε από το χειριστήριο και προσαρμόζει αναλόγως της κινήσεις του αμαξίδιου. Λόγω του ότι επάνω από το **Arduino Uno** έχουμε εφαρμόσει το **Motor Driver Shield**, η σύνδεση της τροφοδοσίας, της γείωσης αλλά και του Data Pin στην ψηφιακή είσοδο 2 γίνεται επάνω στο **Shield** όπου υπάρχουν προεκτάσεις των εισόδων του **Arduino Uno**. Η λειτουργία του τετράτροχου γίνεται από δύο τροφοδοσίες καθώς στις δοκιμές παρατηρήθηκε ότι η μπαταρία 9V δεν ήταν αρκετή ώστε να τροφοδοτήσει επαρκώς τους τέσσερις κινητήρες και το **Arduino Uno** με τα περιφερειακά του. Άρα εφαρμόσαμε μία τάση 9V στο **Motor Driver Shield** για τη λειτουργία των κινητήρων και μία ακόμη τάση 6V για την λειτουργία του **Arduino**.



Εικόνα 25: Σχηματικό σύνδεσης μονάδας λήψης **RF 433Mhz**.



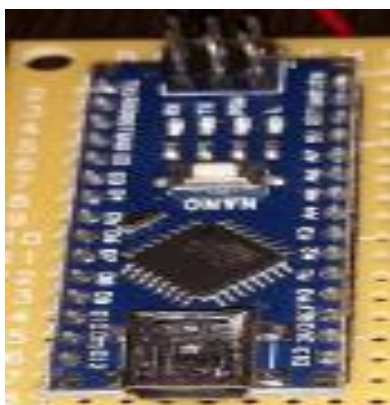
Εικόνα 26: Σύνδεση RF 433Mhz με το Motor Driver Shield.



Εικόνα 27: Θήκη μπαταρίας 9V και 6V αντίστοιχα για το Driver Motor Shield και το Arduino.

3.9 Συναρμολόγηση εξαρτημάτων τηλεχειριστήριου.

Όσον αφορά το τηλεχειριστήριο αποτελείται από τρία κομμάτια. Το βασικότερο εξάρτημα είναι το ARDUINO NANO όπου εκεί σνδέονται όλα τα εξαρτήματα και είναι υπεύθυνο για τον προγραμματισμό και την λειτουργία όλων των μερών του χειριστηρίου. Πάνω σε αυτό συνδέεται επίσης και μια τροφοδοσία 9V για την ηλεκτροδότηση του κυκλώματος. Ο λόγος που χρησιμοποιήθηκε ο συγκεκριμένος μικροελεγκτής είναι λόγω εξοικονόμησης χώρου καθώς έχει το 1/3 μέγεθος του ARDUINO UNO.



Εικόνα 28: Arduino Uno.

Το επόμενο εξάρτημα που χρησιμοποιείται είναι το **MPU-6050** όπου είναι ένας αισθητήρας που περιέχει ένα επιταχυνσιόμετρο και ένα γυροσκόπιο σε ένα ενιαίο τσιπ. Είναι πολύ ακριβής, καθώς ενσωματώνει ψηφιακό υλικό μετατροπής 16 ψηφίων για κάθε κανάλι καταγράφοντας έτσι ταυτόχρονα τα κανάλια x, y και z. Η διασύνδεση με το **Arduino** γίνεται μέσω του δίαυλου **I2C**. Στην παρούσα εργασία χρησιμοποιείται ώστε να αντιλαμβάνεται τις κινήσεις του χεριού και να τις μεταδίδει στο τετράτροχο ώστε να εκτελέσει τις κινήσεις.

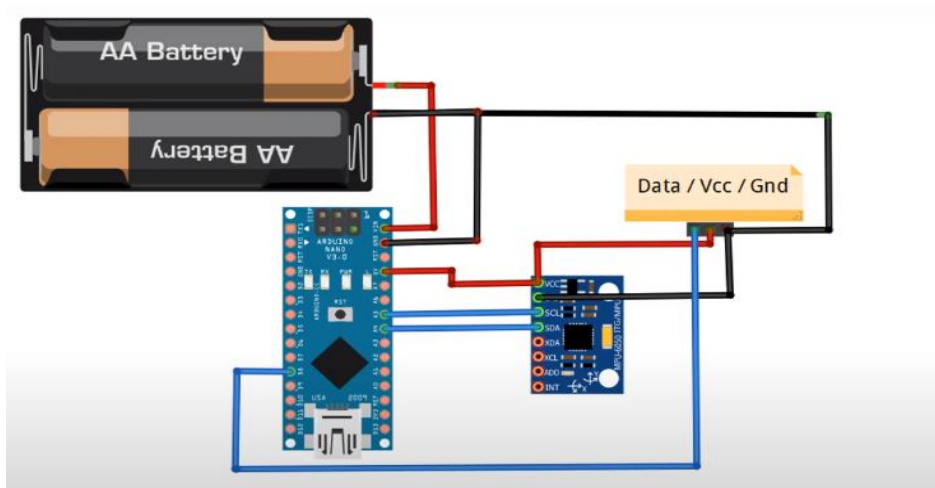


Εικόνα 29: MPU-6050.

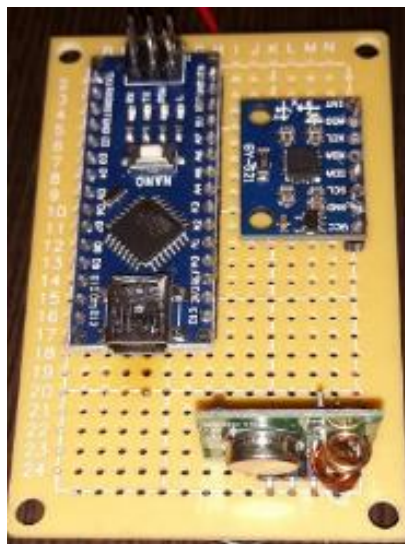
Το τρίτο και τελευταίο κομμάτι του χειριστηρίου αφορά το **RF 433Mhz** και πιο συγκεκριμένα τον πομπό όπου στέλνει τα δεδομένα που λαμβάνει από το MPU-6050 στο **RF 433Mhz receiver** για την εκτέλεση των κατάλληλων κινήσεων.



Εικόνα 30: RF 433Mhz Transmitter.



Εικόνα 31: Συνδεσμολογία Arduino Nano με το MPU-6050, RF 433Mhz και την τροφοδοσία



Εικόνα 32: Χειριστήριο Τετράτροχου.

ΚΕΦΑΛΑΙΟ 4^ο

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΕΤΡΑΤΡΟΧΟΥ ΡΟΜΠΟΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα αναλύσουμε το κομμάτι του προγραμματισμού που συνδέεται με το ρομποτικό σύστημα, ώστε να οδηγήσει και να συντονίσει όλα τα εξαρτήματα, καθώς επίσης και να μπορέσει να γίνει η επεξεργασία των δεδομένων που λαμβάνουμε από το σύστημα. Θα εστιάσουμε στην πλατφόρμα προγραμματισμού και θα αναφερθούμε στους αλγόριθμους που λειτουργούν στο σύστημα. Το πρόγραμμα που χρησιμοποιήθηκε είναι το **Arduino Uno**. Συγκεκριμένα θα αναφερθούμε στην κωδικοποίηση των εξαρτημάτων καθώς και στις λειτουργίες του τετράτροχου ρομποτικού συστήματος.

4.2 Προγραμματισμός

Για πολλά χρόνια, η "τέχνη" της κωδικοποίησης δεν ήταν ευρέως διαδεδομένη. Ωστόσο, αυτό άρχισε να αλλάζει. Τα άτομα που αρχίζουν να μαθαίνουν να κωδικοποιούν αυξάνεται με ραγδαίους ρυθμούς χρόνο με το χρόνο, με εκτιμήσεις περίπου 22,3 εκατομμύρια προγραμματιστών λογισμικού παγκοσμίως, οι οποίοι είναι πρωτοεμφανιζόμενοι στον τομέα του προγραμματισμού χωρίς να σχετίζονται επαγγελματικά με αυτόν.

Ο προγραμματισμός είναι ουσιαστικά η δημιουργία ενός κώδικα υπολογιστή είτε για τη δημιουργία ενός προγράμματος είτε για την επίλυση ενός προβλήματος. Τα προγράμματα δημιουργούνται για την εφαρμογή αλγορίθμων. Οι αλγόριθμοι μπορούν να αναπαρασταθούν ως ψευδοκώδικας ή διάγραμμα ροής και ο προγραμματισμός είναι η μετάφραση αυτών σε πρόγραμμα υπολογιστή.

Όταν δίνουμε εντολή σε έναν υπολογιστή να εκτελέσει μία ενέργεια, αυτό είναι αποτέλεσμα ενός κώδικα για να δείξει στον υπολογιστή πώς να το κάνει. Εφόσον έχει σχεδιαστεί ένας αλγόριθμος, το πρόγραμμα υπολογιστή τον ακολουθεί βήμα προς βήμα, και δείχνει στον υπολογιστή τι ακριβώς πρέπει να κάνει. Ο προγραμματισμός είναι μια τεχνητή γλώσσα που κατανοεί ένας υπολογιστής. Η γλώσσα περιλαμβάνει μια σειρά δηλώσεων που ταιριάζουν μαζί για τον σχηματισμό οδηγιών. Αυτές οι οδηγίες καθοδηγούν τον υπολογιστή τι να κάνει. Υπάρχουν γλώσσες προγραμματισμού διαφορετικές ως προς την περιπλοκότητα και την πολυπλοκότητα. Οι πιο δημοφιλείς εξ αυτών είναι:

- Python

- Java
- C++
- BASIC
- Scratch

Η κάθε γλώσσα προγραμματισμού λειτουργεί με τον δικό της τρόπο. Για παράδειγμα, στην Python ο κώδικας είναι γραμμένος με πεζά, αλλά στο BASIC γράφονται με κεφαλαία γράμματα.

Οι γλώσσες προγραμματισμού είναι σχεδιασμένες έτσι ώστε να μπορεί να τις κατανοήσει και να τις γράψει ο άνθρωπος. Παρόλα αυτά, ο υπολογιστής δεν μπορεί να εκτελεί προγράμματα γραμμένα σε αυτές τις γλώσσες. Οι περισσότερες γλώσσες προγραμματισμού μεταφράζονται πρώτα σε μηχανήμα κώδικα πριν ο υπολογιστής μπορέσει να εκτελέσει τις οδηγίες. Τα προγράμματα περιέχουν δηλώσεις που αναγνωρίζει και κατανοεί η γλώσσα προγραμματισμού. Ένα πρόγραμμα συνδυάζει μία ή περισσότερες δηλώσεις για να σχηματίσει μια οδηγία, όπως οι λέξεις ενώνονται για να σχηματίσουν μια πρόταση. Κάθε δήλωση δίνει μια συγκεκριμένη εντολή στον υπολογιστή να εκτελέσει μια εργασία και οι οδηγίες λένε στον υπολογιστή τι να κάνει.

Δηλώσεις

Διαφορετικές γλώσσες προγραμματισμού χρησιμοποιούν διαφορετικές δηλώσεις. Μερικά από αυτά παρατίθενται σε αυτόν τον πίνακα:

Statement	Purpose
print	Output a message on the screen
input	Get data from the user
if...else	A decision
while	A loop controlled by a decision
for	A loop controlled by a counter

Παραδείγματα

Η ακόλουθη πρόταση ζητά από κάποιον να γράψει ένα μήνυμα σε έναν πίνακα:

"Παρακαλώ γράψτε τις λέξεις" Γεια σου κόσμος! "Στον πίνακα."

Αυτή η πρόταση είναι μια οδηγία, η οποία περιέχει μία μόνο δήλωση. Η δήλωση είναι «γράψτε τις λέξεις». Στο Python (3.x), η ισοδύναμη δήλωση είναι **print**.
`print("Hello world!")`.

4.3 Προγραμματιστικό περιβάλλον Arduino

Στην παρούσα ενότητα θα γίνει ανάλυση του λογισμικού που χρησιμοποιήσαμε για τον προγραμματισμό, το **Arduino**. Η μεγάλη αναφορά γίνεται γιατί σε αυτή την πλατφόρμα γράφτηκαν όλοι οι κώδικες των εξαρτημάτων που υπάρχουν (μοτέρ, αισθητήρες, κ.α), αλλά και ο αλγόριθμος που υπάρχει στον ελεγκτή **Arduino Nano** και **Arduino Uno**, ελέγχοντας όλες τις λειτουργίες του ρομποτικού συστήματος.

Στο συγκεκριμένο λογισμικό λόγω του ότι είναι ανοιχτού τύπου, υπάρχει μεγάλη ευκολία στην εκμάθηση άλλα και την άντληση πληροφοριών του προγράμματος, καθώς επίσης προγραμμάτων και βιβλιοθηκών από ελεύθερους χρήστες στο διαδίκτυο. Δεδομένου ότι βασίζεται σε γλώσσα προγραμματισμού **C/C++** το λογισμικό του χαρακτηρίζεται ως γλώσσα υψηλού επιπέδου αφού είναι πιο προσιτή στον προγραμματιστή και θεωρείται πιο περιγραφική. Επίσης την καθιστά πιο εύκολη στην δημιουργία προγραμμάτων καθώς χρησιμοποιεί βιβλιοθήκες και ρουτίνες.

Το κομμάτι του προγραμματισμού, διαχωρίζεται σε 3 μέρη. Στο πρώτο μέρος αναφερόμαστε στη δομή, η οποία επιλέγεται μέσα από το πρόγραμμα και έχει να κάνει με τον τρόπο δημιουργίας του κώδικα. Στο δεύτερο μέρος γίνεται αναφορά στο είδος των μεταβλητών που θα χρησιμοποιήσουμε και τέλος το τρίτο μέρος έχει να κάνει με τη λειτουργία (function). Function ονομάζεται ένας αλγόριθμος ο οποίος καλείται στο κυρίως πρόγραμμα και εκτελεί έναν σκοπό.

4.4 Κώδικες οδήγησης τετράτροχου ρομποτικού συστήματος σε περιβάλλον ARDUINO

4.4.1 Κώδικας τετράτροχου

Στην υποενότητα αυτή θα γίνει ανάλυση του αλγορίθμου ο οποίος έχει γραφτεί σε προγραμματιστικό περιβάλλον **Arduino**. Ο αλγόριθμος του τετράτροχου χρησιμοποιείται στο **Arduino Uno** για να λαμβάνει πληροφορίες από το χειριστήριο και να οδηγεί τους κινητήρες του συστήματος. Παρακάτω ακολουθούν οι εικόνες 33 και 34 στις οποίες παρουσιάζεται και ο σχετικός αλγόριθμος.

```

void loop()
{
  if (vw_get_message(msg, smsgLen))
  {
    Serial.print("got transmittion");
    motor1.setSpeed(255);
    motor2.setSpeed(255);
    motor3.setSpeed(255);
    motor4.setSpeed(255);
    switch (msg[0])// if msg
    {
      case '0':
      {
        motor1.run(RELEASE);
        motor2.run(RELEASE);
        motor3.run(RELEASE);
        motor4.run(RELEASE);
        break;
      }

      case '1':
      {
        motor1.run(FORWARD);
        motor2.run(FORWARD);
        motor3.run(FORWARD);
        motor4.run(FORWARD);
        break;
      }
    }
  }
}

#include <VirtualWire.h>
#include <AFMotor.h>
byte msg[VW_MAX_MESSAGE_LEN];
byte msgLen = VW_MAX_MESSAGE_LEN;
char forward[1] = {1};

AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);

void setup()
{
  Serial.begin(9600);
  Serial.println("READY.....");
  vw_setup(2048);
  vw_set_rx_pin(2);
  vw_rx_start();
}

```

Εικόνα 33: Αλγόριθμος κίνησης Arduino Uno.

```

case '2':
{
  motor1.setSpeed(150);
  motor2.setSpeed(150);
  motor3.setSpeed(150);
  motor4.setSpeed(180);
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(BACKWARD);
  motor4.run(FORWARD);
  break;
}

case '3':
{
  motor1.setSpeed(180);
  motor2.setSpeed(150);
  motor3.setSpeed(150);
  motor4.setSpeed(150);
  motor1.run(FORWARD);
  motor2.run(BACKWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
  break;
}

case '4':
{
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(BACKWARD);
  motor4.run(BACKWARD);
  break;
}

case '5':
{
  motor3.setSpeed(70);
  motor4.setSpeed(70);
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
  break;
}

case '6':
{
  motor1.setSpeed(70);
  motor2.setSpeed(70);
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
  break;
}
}

```

Εικόνα 34: Αλγόριθμος κίνησης Arduino Uno, Συνέχεια.

Πιο συγκεκριμένα ξεκινάμε δηλώνοντας στην αρχή του προγράμματος τις βιβλιοθήκες που θα χρειαστούμε ώστε να μπορέσουμε να τις καλέσουμε όταν τις χρειαστούμε. Στη δική μας περίπτωση θα χρησιμοποιήσουμε τις βιβλιοθήκες που οδηγεί τα **Motors** με όνομα **AFMotor** και τη βιβλιοθήκη **ViltrualWire** για την επικοινωνία των **RF 433Mhz**. Η σχετική εντολή είναι η **#include** ακολουθούμενη από το όνομα της βιβλιοθήκης. Στη συνέχεια οι εντολές `bytemsg[VW_MAX_MESSAGE_LEN];& byte msgLen=VW_MAX_MESSAGE_LEN;`

Είναι για τη δημιουργία ενός πίνακα μεγέθους **vw_max_message_len** μονοδιάστατος και τη δημιουργία μιας μεταβλητής με τιμή **vw_max_message_len** αντίστοιχα και χρησιμοποιείται στην αρχή τις **void loop()** ώστε να ελέγξει αν το σύστημα είναι εντάξει και να προχωρήσει. Μετά μπαίνουμε στην **void setup()** όπου εκτελείται μία φορά και εκεί καλούμε αρχικά τη συνάρτηση **begin** της class **serial** όπου διαχειρίζεται την επικοινωνία idle με τον H/Y. Οι επόμενες εντολές που βρίσκονται μέσα στην **setup** βρίσκονται για να ξεκινήσουν να μεταδίδονται δεδομένα. Τέλος μπαίνουμε στην **void loop()** όπου αρχικά έχουμε μία **if** όπου ελέγχει αν όλα είναι εντάξει ώστε να προχωρήσει και να εκτελέσει εντολές. Οι εντολές που μπορεί να δεχτεί είναι 7 και είναι **case 1, case 2...** κλπ όπως βλέπουμε και στις εικόνες. Αναλόγως με την εντολή που λαμβάνει ελέγχει ξεχωριστά τον

κάθε **Motor**. Αν για παράδειγμα δώσουμε εντολή να κινηθεί το τετράτροχο μπροστά θα εκτελεστεί η **case 1**. Σε περιπτώσεις όπου βλέπουμε εντολή πχ `motor1.setspeed(150)` είναι για να δώσουμε συγκεκριμένη ταχύτητα στο motor που θέλουμε ώστε να εκτελεστεί μια συγκεκριμένη κίνηση, πχ στροφή δεξιά.

4.4.2 Κώδικας χειριστηρίου

Όπως είχαμε πει υπάρχουν δύο κώδικες, ένας για το τετράτροχο και ένας για το τηλεχειριστήριο οδήγησης. Στην παραπάνω υποενότητα αναλύσαμε τον αλγόριθμο ο οποίος χρησιμοποιήθηκε στο **Arduino Uno** και κατ' επέκταση στο τετράτροχο άρα σε αυτή θα δούμε τον κώδικα που χρησιμοποιήθηκε στο χειριστήριο . Ο συγκεκριμένος αλγόριθμος 'ανέβηκε' στο **Arduino Nano** και χρησιμοποιείται ώστε να λαμβάνει πληροφορίες από το **MPU-6050** ανάλογα με τις κινήσεις που κάνουμε και να στείλει τις πληροφορίες του **RF 433Mhz Transmitter** στο **RF 433Mhz Receiver** και να εκτελεστεί η κίνηση. Παρακάτω ακολουθούν οι εικόνες στις οποίες απεικονίζεται και ο σχετικός κώδικας.

```
#include <VirtualWire.h>
#include "I2Cdev.h"
#include "MPU6050.h"

#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif
MPU6050 accelgyro;
int16_t ax, ay, az;
int16_t gx, gy, gz;

#define OUTPUT_BINARY_ACCELYGRO
#define LED_PIN 13
bool blinkState = false;
char text[5] = "";
byte i = 0;

void setup()
{
  #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
  Wire.begin();//
  #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
  Fastwire::setup(400, true);
  #endif
  Serial.begin(9600);
  accelgyro.initialize();
  Serial.println(accelgyro.testConnection() ? "MPU6050 connection successful" : "MPU6050 connection failed");
  Serial.println("Ready.....");
}
```

Εικόνα 35: Κώδικας χειριστηρίου.

```

vw_set_ptt_inverted(true);
vw_setup(2048);
vw_set_tx_pin(8);
pinMode(LED_PIN, OUTPUT);
}

void loop()
{
  accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz); //
#ifdef OUTPUT_READABLE_ACCELYGYRO
  // display tab-separated accel/gyro x/y/z values
  Serial.print("a/g:\t");
  Serial.print(ax); Serial.print("\t");
  Serial.print(ay); Serial.print("\t");
  Serial.print(az); Serial.print("\t");
  Serial.print(gx); Serial.print("\t");
  Serial.print(gy); Serial.print("\t"); // -32000
  Serial.println(gz);
  //delay(250);
#endif
#ifdef OUTPUT_BINARY_ACCELYGYRO
  |
#endif
}

if ((ay < 9000) and (ay > -9000) and (ax < 12000) and (ax > -3000))
{
  text[i] = '0';
  vw_send((byte *)text, sizeof(text));
  vw_wait_tx();
}

if (ax < -3800)
{
  if (ay < -13000)
  {
    text[i] = '5';
    vw_send((byte *)text, sizeof(text));
    vw_wait_tx();
  }
  else if (ay > 8000)
  {
    text[i] = '6';
    vw_send((byte *)text, sizeof(text));
    vw_wait_tx();
  }
  else
  {
    text[i] = '1'; //tói
    vw_send((byte *)text, sizeof(text));
    vw_wait_tx();
  }
}
}

```

Εικόνα 36: Κώδικας χειριστηρίου συνέχεια.

```

if (ay < -13000)
{
  if (ax < -3800)
  {
    text[i] = '5'; // tói phải
    vw_send((byte *)text, sizeof(text));
    vw_wait_tx();
  }
  else
  {
    text[i] = '2'; // phải
    vw_send((byte *)text, sizeof(text));
    vw_wait_tx();
  }
}

if (ay > 7000)
{
  if (ax < -3800)
  {
    text[i] = '6'; // tói trái
    vw_send((byte *)text, sizeof(text));
    vw_wait_tx();
  }
  else
  {
    text[i] = '3'; // trái
    vw_send((byte *)text, sizeof(text));
  }
}

if (ax < -3800)
{
  text[i] = '6'; // tói trái
  vw_send((byte *)text, sizeof(text));
  vw_wait_tx();
}
else
{
  text[i] = '3'; // trái
  vw_send((byte *)text, sizeof(text));
}

if (ax < -3800)
{
  text[i] = '6'; // tói trái
  vw_send((byte *)text, sizeof(text));
  vw_wait_tx();
}
else
{
  text[i] = '3'; // trái
  vw_send((byte *)text, sizeof(text));
}

if (ax > 14000)
{
  text[i] = '4'; //lúì
  vw_send((byte *)text, sizeof(text)); // gúì tín hiệù ðì
  vw_wait_tx();
}

blinkState = !blinkState;
digitalWrite(LED_PIN, blinkState);
}

```

Εικόνα 37: Κώδικας χειριστηρίου τέλος.

Όπως και στην προηγούμενο κώδικα ξεκινάμε δηλώνοντας στην αρχή του προγράμματος τις βιβλιοθήκες που θα χρειαστούμε ώστε να μπορέσουμε να τις καλέσουμε όταν τις χρειαστούμε. Σε αυτόν τον κώδικα οι βιβλιοθήκες που θα χρησιμοποιήσουμε είναι η βιβλιοθήκη **Wire** για την επικοινωνία των **RF 433Mhz**, η **I2Cdev.h** και η **MPU6050.h** όπου χρησιμοποιούνται για το **MPU-6050** καθώς έχει δύο λειτουργίες, σαν επιταχυνσιόμετρο και σαν γυροσκόπιο . Η σχετική εντολή είναι η **#include** ακολουθούμενη από το όνομα της βιβλιοθήκης. Στην εικόνα 38 από κάτω βλέπουμε μια **if** στην αρχή του κώδικα όπου υπάρχει μια συνθήκη ώστε να μπορούν να τρέξουν οι συναρτήσεις από τη βιβλιοθήκη **Wire.h** και η χρήση της είναι να απενεργοποιεί τη μετάδοση δεδομένων εάν υπάρξει υπέρβαση των byte.

```
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif
```

Εικόνα 38: Συνθήκη με if και βιβλιοθήκη Wire.h.

Τέλος υπάρχει επίσης η δήλωση μεταβλητών το οποίο φαίνεται στην εικόνα 39 για το **MPU-6050** **accelgyro** ώστε μετά μέσα στη **Void loop** αναλόγως τις τιμές που θα παίρνει να στέλνει τα αντίστοιχα δεδομένα και να εκτελείται η αντίστοιχη κίνηση.

```
MPU6050 accelgyro;
int16_t ax, ay, az;
int16_t gx, gy, gz;
```

Εικόνα 39: Δήλωση μεταβλητών.

Μπαίνοντας στην **Void Setup** όπως παρατηρούμε και στην εικόνα 40 εκτελούμε μια σειρά από συναρτήσεις ώστε να ελέγξουμε αν όλα επικοινωνούν μεταξύ τους για να αρχίσει η μεταφορά δεδομένων στο τετράτροχο και να εκτελέσει τις κινήσεις. Η **Void Setup** όπως έχουμε ξαναπεί εκτελείται μία φορά στην αρχή. Γενικά επίσης θα πρέπει να γνωρίζουμε ότι όλοι οι κώδικες τρέχουν κυλιόμενα δηλαδή με τη σειρά από την αρχή μέχρι το τέλος.

```

void setup()
{
  #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    Wire.begin();//
  #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
    Fastwire::setup(400, true);
  #endif
  Serial.begin(9600);// giao tiếp Serial với baudrate 9600
  accelgyro.initialize();
  Serial.println(accelgyro.testConnection() ? "MPU6050 connection successful" : "MPU6050 connection failed");
  Serial.println("Ready.....");
  vw_set_ptt_inverted(true);// yêu cầu cho RF link modules
  vw_setup(2048);// cài đặt tốc độ truyền tin hiệu
  vw_set_tx_pin(8);// cài đặt chân digital để phát tín hiệu
  pinMode(LED_PIN, OUTPUT);
}

```

Εικόνα 40: Συναρτήσεις στη Void Setup.

Εφόσον έχουν εκτελεστεί όλες οι συναρτήσεις και έχει δείξει ότι όλα λειτουργούν σωστά μπαίνουμε στην **Void loop** όπου εκεί μέσω κάποιων if όπως θα δούμε και στην εικόνα 41, αναλόγως τις τιμές που δίνει το **MPU-6050** και κατ' επέκταση εμείς με την κίνηση που κάνουμε, στέλνει κάποιες τιμές και αυτές μεταφράζονται σε κινήσεις που θα κάνει το αμαξίδιο.

```

if (ax < -3800)
{
  if (ay < -13000)
  {
    text[i] = '5';
    vw_send((byte *)text, sizeof(text));
    vw_wait_tx();
  }
  else if (ay > 8000)
  {
    text[i] = '6';
    vw_send((byte *)text, sizeof(text));
    vw_wait_tx();
  }
}

```

Εικόνα 41: Κίνηση δεξιά-αριστερά.

Στην εικόνα 41 βλέπουμε στην πρώτη if όπου λέει πως αν το ay είναι μικρότερο της τιμής -13000 τότε γίνεται η κίνηση δεξιά και αντίστοιχα από κάτω αν ay μεγαλύτερο του 8000 τότε γίνεται η κίνηση αριστερά.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Ο σημαντικότερος στόχος της παρούσας διπλωματικής εργασίας ήταν η κατασκευή ενός τετράτροχου ρομποτικού συστήματος με ασύρματο έλεγχο των κινήσεων . Το σύστημα θα είναι ικανό να κινείται ασύρματα και να εκτελεί κινήσεις από απόσταση. Βασίζεται στον μικροελεγκτή Arduino ο οποίος είναι υπεύθυνος για την λειτουργία του συστήματος και τον συγχρονισμό όλων των αισθητήρων. Εργαστήκαμε σε περιβάλλον μίας γλώσσας προγραμματισμού, το Arduino το οποίο αποτελούνταν από δύο κώδικες έναν για το τετράτροχο όχημα και έναν για το χειριστήριο.

Η ενασχόληση με την εργασία και κατ' επέκταση με το ρομποτικό σύστημα αλλά και ο χρόνος εκπόνησης αποτέλεσε μια σημαντική πηγή μάθησης και γνώσης, τόσο σε θεωρητικό επίπεδο όσο και σε τεχνικό για την ανάπτυξη και την υλοποίηση της συγκεκριμένης μελέτης. Επίσης μπορέσαμε να εντρυφήσουμε σε γνώσεις σχετικές με τα ρομποτικά συστήματα. Ακόμη η επαφή μας με την πλατφόρμα προγραμματισμού συνέβαλαν στην καλύτερη κατανόηση της λειτουργίας του, της δομής που έχει αλλά και της γλώσσας προγραμματισμού. Όλη αυτή η επαφή με την συγκεκριμένη εργασία βοήθησε να αντιληφθούμε καλύτερα τον κλάδο της ρομποτικής που ασχολείται το τμήμα μας, να την φέρουμε στην επιφάνεια και να βοηθήσουμε αρκετούς να προβληματιστούν ευχάριστα σχετικά με την επαγγελματική τους σταδιοδρομία.

Όσον αφορά την εξέλιξη της παρούσας εργασίας, θα μπορούσαν να γίνουν κάποιες διαφοροποιήσεις. Μία από αυτές αφορά στον έλεγχο του τετράτροχου. Μιας και είμαστε πλέον στον κόσμο της τεχνολογίας ο έλεγχος θα μπορούσε να γίνει αντικαθιστώντας το εξάρτημα Rf 433Mhz με ένα Bluetooth module (**HM10**) ώστε ο έλεγχος του τετράτροχου να γίνεται μέσω εφαρμογής του κινητού αντί να χρησιμοποιούμε τη δεύτερη πλακέτα. Με αυτό τον τρόπο δεν χρειαζόμαστε επιπλέον συσκευή για τον έλεγχο του τετράτροχου και γίνεται παράλληλα και εξοικονόμηση χρημάτων

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] https://en.wikipedia.org/wiki/History_of_robots
- [2] ΜΑΥΡΟΜΑΤΑΚΗΣ ΟΔΥΣΣΕΑΣ. ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ: ΜΕΛΕΤΗ, ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ ΡΟΜΠΟΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΑΥΤΟ-ΙΣΟΡΡΟΠΗΣΗΣ. ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΗΣ, ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΕΠΙΣΤΗΜΩΝ, ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε (2015). Σελ 16-17.
- [3] ΑΓΓΕΛΙΚΗ ΜΠΕΛΕΧΑΚΗ. ΕΚΠΑΙΔΕΥΤΙΚΗ ΡΟΜΠΟΤΙΚΗ ΜΕ ARDUINO. (2021) σελ 11-17
- [4] ΠΕΤΡΑΚΗΣ ΦΩΤΙΟΣ, ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ, ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ, ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ, ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ: ΣΥΣΤΗΜΑΤΑ ΣΥΝΑΓΕΡΜΟΥ ΜΕ ARDUINO (2020). Σελ 16-19.
- [5] ARDUINO. HOME > ARDUINO NANO. Ανακτήθηκε από:
<https://store.arduino.cc/products/arduino-nano/?selectedStore=eu>
- [6] ARDUINO. GETTING STARTED > GETTING STARTED WITH THE ARDUINO NANO.
Ανακτήθηκε από: <https://www.arduino.cc/en/Guide/ArduinoNano>
- [7] L293, L293D QUADRUPLE HALF-H DRIVERS (SLRS008C – SEPTEMBER 1986 – REVISED NOVEMBER 2004). Ανακτήθηκε από: http://wiki.sunfounder.cc/images/a/af/L293D_datasheet.pdf
- [8] SN54HC595, SN74HC595, 8-BIT SHIFT REGISTERS WITH 3-STATE OUTPUT REGISTERS (SCLS041G-DECEMBER 1982-RVISED FEBRUARY 2004). Ανακτήθηκε από:
<http://wiki.sunfounder.cc/images/9/9d/74HC595.pdf>
- [9] F. Egan, William (2003). *Practical RF System Design*. Wiley-IEEE Press
Peter Birnie & Fairall, John (2002). RF Solutions Ltd

ΠΗΓΕΣ ΕΙΚΟΝΩΝ

[1] <http://robotecmed.blogspot.gr>

[2]<https://www.kongsberg.com/maritime/products/marine-robotics/autonomous-underwater-vehicles/?OpenDocument>

[3] ΜΑΥΡΟΜΑΤΑΚΗΣ ΟΔΥΣΣΕΑΣ. ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ: ΜΕΛΕΤΗ, ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ ΡΟΜΠΟΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΑΥΤΟ-ΙΣΟΡΡΟΠΗΣΗΣ. ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ, ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΕΠΙΣΤΗΜΩΝ, ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε (2015).

[4] ΠΕΤΡΑΚΗΣ ΦΩΤΙΟΣ, ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ, ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ, ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ, ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ: ΣΥΣΤΗΜΑΤΑ ΣΥΝΑΓΕΡΜΟΥ ΜΕ ARDUINO (2020).

[5] ARDUINO. HOME > ARDUINO NANO. Ανακτήθηκε από:
<https://store.arduino.cc/arduino-nano-every-with-headers>

[6,7,8,9,10,11] http://wiki.sunfounder.cc/index.php?title=L293D_Motor_Driver_Shield

[12] http://wiki.sunfounder.cc/index.php?title=MPU6050_Module

[13] HOME > ROBOTICS > DEVELOPMENT TOOLS > ARDUINO > MODULES FOR ARDUINO > RF LINK TRANSMITTER AND RECEIVER – 433Mhz. Ανακτήθηκε από:

<https://grobotronics.com/rf-link-transmitter-and-receiver-433mhz.html?sl=en>

[14]<https://www.cableworks.gr/ilektronika/kit/robot-car-kit/4wd-smart-robot-car-chassis-kit-for-arduino-with-speed-encoder-for-arduino/>

[15]<https://www.cableworks.gr/ilektronika/arduino-and-microcontrollers/motors/dc/car-robot-plastic-tire-wheel-dc-gear-motor-for-arduino/>