



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

**ΜΕΛΕΤΗ, ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ**  
**ΑΝΑΠΤΥΞΗ ΡΟΜΠΟΤΙΚΟΥ**  
**ΣΥΣΤΗΜΑΤΟΣ ΑΥΤΟ-ΙΣΟΡΡΟΠΗΣΗΣ**

υπό

Μαυροματάκη Οδυσσέα

Χανιά, 2015

Τριμελής Επιτροπή:

Ε. Δοϊτσίδης

Επίκουρος Καθηγητής, Επιβλέπων

Κ. Πετρίδης

Επίκουρος Καθηγητής

Ν. Φραγκιαδάκης

Επίκουρος Καθηγητής

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η σελίδα αυτή μπορεί να παραλειφθεί.

## ΠΕΡΙΛΗΨΗ ΠΤΥΧΙΑΚΗΣ

Σκοπός της παρούσας εργασίας ήταν η ανάπτυξη και υλοποίηση μιας πρωτότυπης ρομποτικής κατασκευής που θα είχε την ικανότητα να ισορροπεί σε δύο τροχούς ανεξάρτητα από τις διαταραχές που επιδρούν σε αυτή. Η συγκεκριμένη συσκευή αποτελεί ιδανική πλατφόρμα πειραματισμού και κατανόησης θεμάτων που σχετίζονται με τη μηχανική, τη ρομποτική και των προγραμματισμό ηλεκτρονικών συσκευών.

Στα πλαίσια της εργασίας έγινε λεπτομερής σχεδιασμός και ανάπτυξη διαφόρων πρωτοτύπων προκειμένου να καταλήξουμε στον τελικό σχεδιασμό. Το τελικό πρωτότυπο είναι αυτόνομο και πλήρως ελεγχόμενο από έναν μικροεπεξεργαστή και έχει τη δυνατότητα εξωτερικής ασύρματης επικοινωνίας με Η/Υ.

Το συγκεκριμένο πρωτότυπο αποτέλεσε τη βάση για να γίνει διερεύνηση της δυνατότητας ανάπτυξης ενός αντίστοιχου οχήματος ικανού να σηκώσει και να μεταφέρει ένα ενήλικα άτομο.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1 .....	1
ΕΙΣΑΓΩΓΗ .....	1
1.1    ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΕΡΓΑΣΙΑΣ.....	1
1.2    ΣΤΟΧΟΙ ΤΗΣ ΕΡΓΑΣΙΑΣ .....	1
ΚΕΦΑΛΑΙΟ 2 .....	2
ΜΕΘΟΔΟΛΟΓΙΑ / ΣΧΕΔΙΑΣΗ .....	2
2.1 Η ΙΣΤΟΡΙΑ ΤΩΝ ΡΟΜΠΟΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΑΥΤΟ-ΙΣΟΡΡΟΠΗΣΗΣ....	2
2.2 ΚΑΤΑΣΚΕΥΗ 1 <sup>ου</sup> ΠΡΩΤΟΤΥΠΟΥ .....	6
2.3 ΚΙΝΗΤΗΡΕΣ .....	7
2.4 ΑΙΣΘΗΤΗΡΙΑ.....	9
2.5 Η-BRIDGE MOTOR CONTROLLER.....	9
2.6 ΣΥΣΚΕΥΗ ΑΣΥΡΜΑΤΗΣ ΕΠΙΚΟΙΝΩΝΙΑΣ (BLUETOOTH).....	10
2.7 ΤΡΟΦΟΔΟΣΙΑ ΣΥΣΤΗΜΑΤΟΣ .....	11
2.8 ΠΛΑΚΕΤΑ ΣΤΑΘΕΡΟΠΟΙΗΣΗΣ ΤΑΣΗΣ .....	12
2.9 LCD ΟΘΟΝΗ .....	13
2.10 ΣΥΝΔΕΣΜΟΣ ΜΕΤΑΞΥ ΑΞΟΝΑ ΚΙΝΗΤΗΡΑ ΚΑΙ ΤΡΟΧΟΥ .....	13
2.11 ΤΡΟΧΟΙ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ.....	14
2.12 ΜΟΝΑΔΑ ΕΛΕΓΧΟΥ ΤΟΥ ΡΟΜΠΟΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ .....	14
2.13 ΤΕΛΙΚΗ ΚΑΤΑΣΚΕΥΗ ΠΡΩΤΟΤΥΠΟΥ ΔΙΤΡΟΧΟΥ ΑΥΤΟΝΟΜΟΥ ΡΟΜΠΟΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ .....	16
ΚΕΦΑΛΑΙΟ 3 .....	19
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ .....	19
3.1 Ο ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΗΣ.....	19
3.2 ΠΡΟΓΡΑΜΜΑΤΙΖΟΝΤΑΣ ΤΟΝ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΗ.....	19
ΚΕΦΑΛΑΙΟ 4 .....	23
ΜΟΝΤΕΛΟΠΟΙΗΣΗ .....	23
4.1 ΦΥΣΙΚΗ ΣΗΜΑΣΙΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ .....	23
4.2 ΦΙΛΤΡΑ ΚΑΙ ΣΥΝΔΙΑΣΜΟΣ ΤΩΝ ΑΙΣΘΗΤΗΡΙΩΝ.....	24
4.3 COMPLEMENTARY ΦΙΛΤΡΟ .....	25

4.4 ΠΕΙΡΑΜΑΤΑ ΓΙΑ ΤΗ ΣΥΜΠΕΡΙΦΟΡΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	27
ΚΕΦΑΛΑΙΟ 5 .....	30
ΣΥΜΠΕΡΑΣΜΑΤΑ/ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	30
5.1 ΣΥΜΠΕΡΑΣΜΑΤΑ .....	30
5.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΦΑΡΜΟΓΕΣ.....	31
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	35
ΠΑΡΑΡΤΗΜΑ.....	36
1.1 ΑΝΑΛΥΣΗ ΤΟΥ ΚΩΔΙΚΑΣ.....	36
1.2 ΚΩΔΙΚΑΣ.....	36

# ΚΕΦΑΛΑΙΟ 1

## ΕΙΣΑΓΩΓΗ

### 1.1 ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΕΡΓΑΣΙΑΣ

Σκοπός της παρούσας εργασίας ήταν η ανάπτυξη και υλοποίηση μιας πρωτότυπης ρομποτικής κατασκευής που θα είχε την ικανότητα να ισορροπεί σε δύο τροχούς ανεξάρτητα από τις διαταραχές που επιδρούν σε αυτή. Η συγκεκριμένη συσκευή αποτελεί ιδανική πλατφόρμα πειραματισμού και κατανόησης θεμάτων που σχετίζονται με τη μηχανική, τη ρομποτική και των προγραμματισμό ηλεκτρονικών συσκευών.

Στα πλαίσια της εργασίας έγινε λεπτομερής σχεδιασμός και ανάπτυξη διαφόρων πρωτοτύπων προκειμένου να καταλήξουμε στον τελικό σχεδιασμό. Το τελικό πρωτότυπο είναι αυτόνομο και πλήρως ελεγχόμενο από έναν μικροεπεξεργαστή και έχει τη δυνατότητα εξωτερικής ασύρματης επικοινωνίας με Η/Υ.

Το συγκεκριμένο πρωτότυπο αποτέλεσε τη βάση για να γίνει διερεύνηση της δυνατότητας ανάπτυξης ενός αντίστοιχου οχήματος ικανού να σηκώσει και να μεταφέρει ένα ενήλικα άτομο.

### 1.2 ΣΤΟΧΟΙ ΤΗΣ ΕΡΓΑΣΙΑΣ

Η εργασία έχει εκπαιδευτικό σκοπό και στοχεύει στην ευκολότερη κατανόηση των βασικών εννοιών που σχετίζονται με τη μηχανική, τα αυτόνομα ρομποτικά οχήματα καθώς και των προγραμματισμό ενσωματωμένων συστημάτων.

Οι στόχοι που καλύπτει η εργασία είναι:

- 1) Η κατανόηση του τρόπου λειτουργίας ενός οχήματος αυτό-ισορρόπησης.
- 2) Η ανάπτυξη ενός ενσωματωμένου συστήματος ικανού να ελέγξει πλήρως ένα όχημα αυτού του τύπου.
- 3) Η ανάπτυξη ενός ελεγκτή πραγματικού χρόνου.
- 4) Η επαφή με τεχνολογία και διαδικασία ανάπτυξης και κατασκευής ενός ρομποτικού συστήματος.

## ΚΕΦΑΛΑΙΟ 2

### ΜΕΘΟΔΟΛΟΓΙΑ / ΣΧΕΔΙΑΣΗ

#### 2.1 Η ΙΣΤΟΡΙΑ ΤΩΝ ΡΟΜΠΟΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΑΥΤΟ-ΙΣΟΡΡΟΠΗΣΗΣ

Τα ρομποτικά συστήματα αυτό-ισορρόπησης έκαναν την εμφάνισή τους στις αρχές του 21<sup>ου</sup> αιώνα για την ακρίβεια εμφανίστηκαν το 2001. Το πρώτο ρομποτικό σύστημα αυτό-ισορρόπησης ονομαζόταν Segway και εφευρέτης του είναι ο Dean Kamen ο οποίος το παρουσίασε στο κοινό το 2001 σε μία αμερικάνικη τηλεοπτική (Good Morning America) του καναλιού ABC. Με αυτή την εφεύρεση ο κύριος Dean Kamen είχε στόχο να προτείνει ένα εναλλακτικό τρόπο κίνησης σε αστικό περιβάλλον. Η ονομασία που προήλθε το όνομά του είναι παράγωγο της λέξης «segue» που σημαίνει «ομαλή μετάβαση από το ένα πράγμα στο άλλο». Στη συγκεκριμένη περίπτωση, σήμαινε την υιοθέτηση του «Segway», που θα μεταφέρει ανθρώπους όλων των ηλικιών σε όλο τον κόσμο ομαλά και ανεμπόδιστα. Το Segway διαθέτει πέντε γυροσκόπια και δύο αισθητήρες κλίσης, ζυγίζει 30 κιλά και μπορούσε να φτάσει τελική ταχύτητα 20.1 χιλιόμετρα την ώρα. Το 2003 διατίθεντο στο κοινό τρις διαφορετικές εκδόσεις και με τα χρόνια η εξέλιξή του άρχισε να παίρνει τεράστιες διαστάσεις [10].

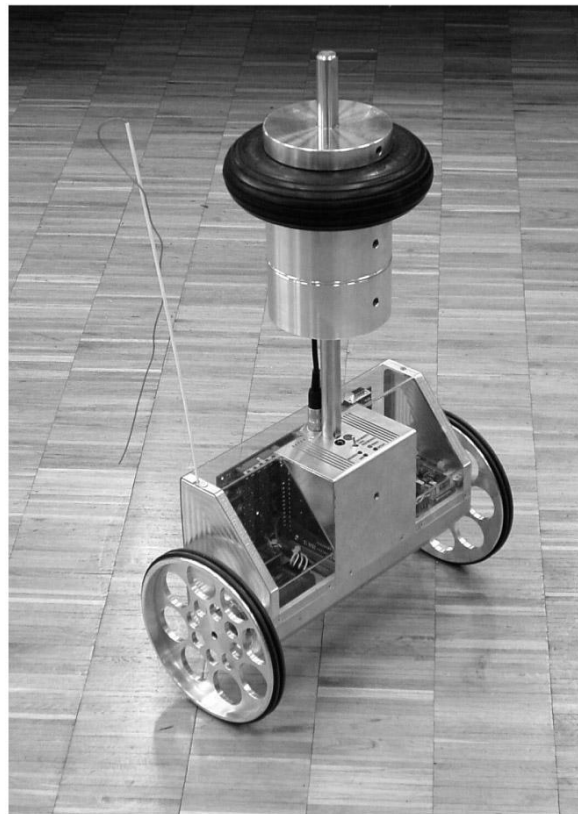


Σχήμα 2.1 : Ρομποτικό σύστημα αυτό-ισορρόπησης (Segway) [1]



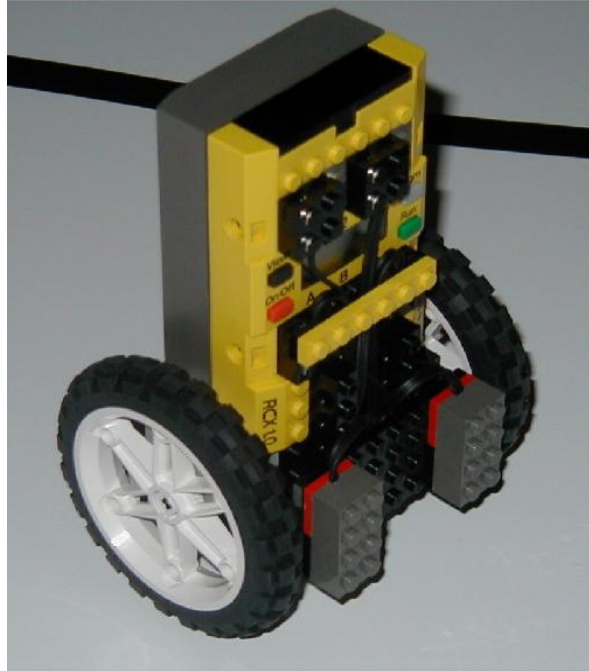
Με βάση τα χαρακτηριστικά και τις δυνατότητες του Segway αναπτύχθηκαν σε μικρότερη κλίμακα τα ρομποτικά συστήματα αυτό-ισορρόπησης που τράβηξαν το ενδιαφέρον πολλών ηλεκτρονικών μηχανικών και ερευνητών. Όλα σχεδόν τα ρομποτικά συστήματα χρησιμοποιούν τις ίδιες αρχές. Κάποια από τα πιο γνωστά ρομποτικά συστήματα αυτό-ισορρόπησης είναι τα: Joe le pendule, LegWay, Nbot και Equibot.

Το Joe le pendule (Grasser, D'Arrigo, Colombi, & Rufer) σχεδιάστηκε το 2002 από τον Felix Grasser με την ομάδα του από το Industrial Electronics Laboratory στο Swiss Federal Institute of Technology (EPFL) στην Lausanne. Το Joe le pendule ρομπότ χρησιμοποιεί ένα επιταχυνσιόμετρο και ένα γυροσκόπιο για να ανιχνεύει την κλίση της γωνιακής ταχύτητας του σώματος του, και δύο καταγραφείς για να μετρούν τη θέση και την ταχύτητα του ρομπότ. Το ρομποτικό αυτό σύστημα παρουσιάζεται στο σχήμα 2.2.



Σχήμα 2.2 : Joe le pendule ρομποτικό σύστημα[1]

Το LegWay (Hassenplug, 2002) κατασκευάστηκε από τον Steve Hassenplug χρησιμοποιώντας τουβλάκια Lego και την πλατφόρμα προγραμματισμού Lego Mindstorms το 2002. Αν και τα υλικά και οι αισθητήρες με τους οποίους κατασκευάστηκε δεν ήταν τα ιδανικά, αποδείχθηκε ότι ένα δίτροχο αυτό-εξισορρόπησης ρομπότ ήταν δυνατόν να κατασκευαστεί χρησιμοποιώντας κατά κύριο λόγο συστατικά από "παιχνίδια". Η λειτουργία του βασίζονταν στη χρήση δύο ηλεκτροπτικών ανιχνευτών απόστασης για την εξισορρόπηση και εντοπισμό γραμμών. Το LegWay παρουσιάζεται στο σχήμα 2.3.



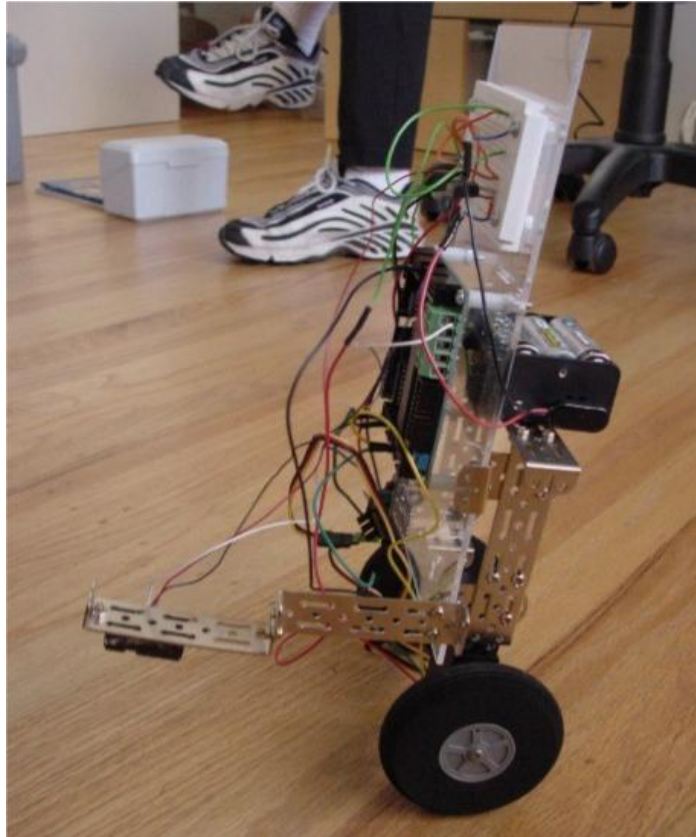
Σχήμα 2.3 : LegWay ρομποτικό σύστημα[1]

Το Nbot (Anderson, 2010) σχεδιάστηκε από τον David P. Anderson, ο διευθυντής του Εργαστηρίου Γεωφυσικής απεικόνισης στο πανεπιστήμιο Southern Methodist στο Ντάλας. Διέθετε ένα μικροεπεξεργαστή HC11, ADXL202 επιταχυνσιόμετρο, γυροσκόπιο, αισθητήρα κλίσης και οπτικούς κωδικοποιητές για τους κινητήρες. Το γυροσκόπιο και επιταχυνσιόμετρο συνδυάζονται με συμπληρωματικά φίλτρα για να ενεργούν ως ένα αισθητήρα αναφοράς (Σχήμα 2.4).



Σχήμα 2.4 : Nbotρομποτικό σύστημα[1]

Το Equibot (Pironi, 2006) του Dan Pironi, το παρουσιάστηκε το 2006. Χρησιμοποιεί ένα αισθητήρα υπερύθρων Sharp GP2D120 για τη μέτρηση της απόστασης από το έδαφος. Με βάση τη συγκεκριμένη μέτρηση ένας μικροεπεξεργαστής atmega32 υπολογίζει τη γωνία κλίσης του ρομπότ, που στη συνέχεια χρησιμοποιείται ως βάση για την εξισορρόπηση της. Το ρομποτικό σύστημα παρουσιάζεται στο σχήμα 2.5.



Σχήμα 2.5 :Equibotρομποτικό σύστημα[1]

Το uBot-4 έχει δύο μπράτσα που το κάθε ένα διαθέτει τέσσερις αρθρώσεις με ενσωματωμένο σύστημα ανάδρασης για την καλύτερη λειτουργία του συστήματος. Ο χώρος εργασίας που μπορεί να κινηθεί το ρομποτικό αυτό σύστημα είναι κατά ένα βαθμό ελευθερίας μεγαλύτερος στα χέρια από κάποια άλλα ρομποτικά συστήματα της ίδιας φιλοσοφίας και διαθέτει μια επιπλέον περιστροφή στο κορμό. Διαπιστώθηκε επίσης ότι η απαίτηση από τους κατασκευαστές του uBot-4 θα είναι η ικανότητα του να επιβιώσει από μια πτώση και στη συνέχεια να μπορεί να σηκωθεί χωρίς να του δοθεί εξωτερική βοήθεια. Προκειμένου να επιβιώσει από πτώση, το uBot-4 σχεδιάστηκε να είναι μηχανικά ισχυρό επειδή θα πρέπει να αντεπεξέλθει το μέγεθος των δυνάμεων κρούσης που αναμενόταν. Προς το παρόν, το πρωτότυπο uBot-4 χρησιμοποιεί κινητήρες που δεν πληρούν τις προδιαγραφές σχεδιασμού του και ως εκ τούτου δεν είναι σε θέση να ανακτήσει τη αρχική του θέση από μια πτώση. Το ρομποτικό αυτό σύστημα παρουσιάζεται στο σχήμα 2.6.

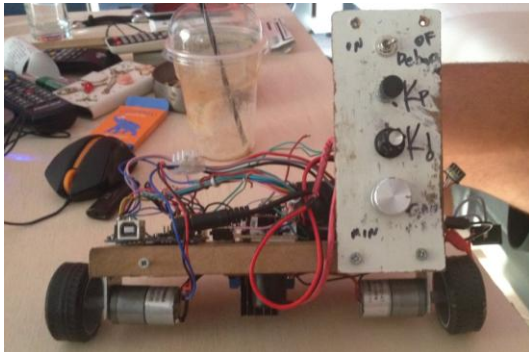


Σχήμα 2.6 :uBot-4 [2]

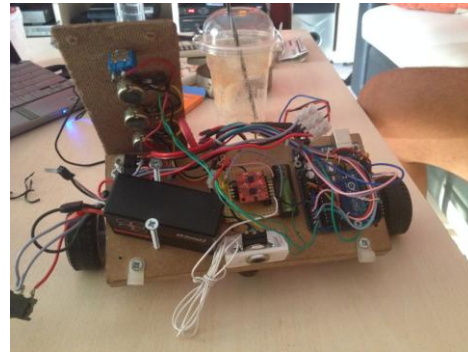
## 2.2 ΚΑΤΑΣΚΕΥΗ 1<sup>ΟΥ</sup> ΠΡΩΤΟΤΥΠΟΥ

Στα πλαίσια της πτυχιακής εργασίας κατασκευάστηκε, σχεδιάστηκε και αναπτύχθηκε πρωτότυπο δίτροχο ρομποτικό σύστημα το οποίο απεικονίζεται στο σχήμα 2.7 α) και 2.7 β). Αρχικά έγινε ένας πρόχειρος σχεδιασμός του ρομποτικού δίτροχου με απλά και εύχρηστα υλικά, διότι θέλαμε να διαπιστώσουμε την ικανότητά του για αυτόματη ισορρόπηση. Χρησιμοποιήθηκε μία πλατφόρμα από ξύλο με μήκος 200mm, πλάτος 95mm και ύψος 28mm στην οποία προσαρμόστηκαν όλα τα ηλεκτρονικά και μηχανικά μέρη του δίτροχου. Στο κάτω μέρος της πλατφόρμας τοποθετήθηκαν οι κινητήρες πάνω σε βάσεις με σχήμα Γ καθώς οι τροχοί και η H-Bridge. Στο επάνω μέρος της τοποθετήθηκαν η μπαταρία, τα αισθητήρια, τα ποτενσιόμετρα για εύκολη ρύθμιση

κάποιων παραμέτρων μέσα στον κώδικα και η πλακέτα ανάπτυξης Arduino R3. Στη συνέχεια το ρομποτικό δίτροχο κατασκευάστηκε από πλαστικό έτσι ώστε να μειωθεί το βάρος του και για να είναι ανθεκτικότερο<sup>1</sup>.



(α)



(β)

Σχήμα 2.7: 1<sup>ο</sup> Πρωτότυπο (α) μπροστά όψη, (β) πίσω όψη

### 2.3 ΚΙΝΗΤΗΡΕΣ

Οι αρχικοί κινητήρες που χρησιμοποιήθηκαν για την υλοποίηση στις εργασίες χρησιμοποιούσαν συνεχή τάση 12V και η ροπή τους ήταν στα 200RPM αλλά δεν χρησιμοποιήθηκαν διότι η ροπή τους δεν μπορούσε να αντεπεξέλθει στο βάρος της κατασκευής και έτσι δεν μπορούσε να ισορροπήσει το δίτροχο ρομποτικό σύστημα. Ο άξονας περιστροφής του κινητήρα έχει διάμετρο στα 4mm.



Σχήμα 2.8 :Κινητήρες 12V 200RPM

<sup>1</sup> Στις επόμενες παραγράφους θα γίνει αναλυτική περιγραφή των ηλεκτρονικών και μηχανικών μερών του οχήματος.



Προκειμένου να ξεπεραστεί το πρόβλημα με τη ροπή των κινητήρων χρησιμοποιήθηκαν οι κινητήρες A401 της εταιρίας CNDF με συνεχή τάση λειτουργίας στα 12V και ροπή στα 300RPM. Ο άξονας περιστροφής του κινητήρα έχει διάμετρο στα 4mm.



Σχήμα 2.9 :Κινητήρας A401 12V 300RPM

Στον τελικό σχεδιασμό χρησιμοποιήθηκαν οι κινητήρες A608 της ίδιας εταιρίας καθώς και οι δύο προηγούμενοι τύπο αποδείχτηκαν ανεπαρκής. Η συνεχής τάση λειτουργίας τους είναι στα 12V, η ροπή τους είναι στα 300RPM και ο άξονας τους έχει την ίδια διάμετρο με την διάμετρο του άξονα σύνδεσης του τροχού δηλαδή η διάμετρος του είναι στα 6mm.



Σχήμα 2.10 : Κινητήρας A608 12V 300RPM

## 2.4 ΑΙΣΘΗΤΗΡΙΑ

Για να επιτύχουμε την ισορρόπηση του ρομποτικού δίτροχου ήταν απαραίτητη η χρήση κατάλληλων αισθητήρων. Τα αισθητήρια αυτά θα έπρεπε να είναι σε θέση να μετρούν την κλίση και την επιτάχυνση, έτσι ώστε να μπορεί το σύστημα να χρησιμοποιήσει αυτή την πληροφορία και σε συνδυασμό με κατάλληλες στρατηγικές ελέγχου να δίνει τις κατάλληλες τιμές στους κινητήρες προκειμένου να γίνεται διόρθωση της κλίσης της πλατφόρμας ώστε τελικά να επιτυγχάνεται ισορροπία. Το αισθητήριο που χρησιμοποιήθηκε ήταν ένα ολοκληρωμένο σύστημα πέντε βαθμών ελευθερίας, αποτελούμενο από: α) ένα επιταχυνσιόμετρο (ADXL-335) τριών βαθμών ελευθερίας και β) ένα γυροσκόπιο (IDG-650) δυο βαθμών ελευθερίας. Με την βοήθεια αυτών των αισθητήρων είναι δυνατόν να επιτευχθεί η παρακολούθηση της κλίσης και η επιτάχυνση της πλατφόρμας στους τρεις άξονες X, Y και Z. Η τάση λειτουργίας των αισθητηρίων κυμαίνεται από 3.3V έως 5V και η επικοινωνία με την μονάδα ελέγχου (μικροεπεξεργαστή) επιτυγχάνεται με τη χρήση των αναλογικών επαφών των αισθητηρίων.



Σχήμα 2.11 : Αναλογικά Αισθητήρια (Επιταχυνσιόμετρο ,Γυροσκόπιο)

## 2.5 H-BRIDGE MOTOR CONTROLLER

Προκειμένου να επιτευχθεί η αντιστροφή της κίνησης των κινητήρων έτσι ώστε η πλατφόρμα να μπορέσει να ισορροπήσει χρησιμοποιήθηκε μια H-Bridge. Η πλακέτα αυτή μπορεί να λειτουργήσει με εύρος τάσης από 5V έως 35V και το μέγιστο ρεύμα το οποίο μπορεί να διαπεράσει το ολοκληρωμένο της κύκλωμα είναι τα 2A. Η συγκεκριμένη H-Bridge πλακέτα μπορεί να οδηγήσει δυο κινητήρες ταυτόχρονα και

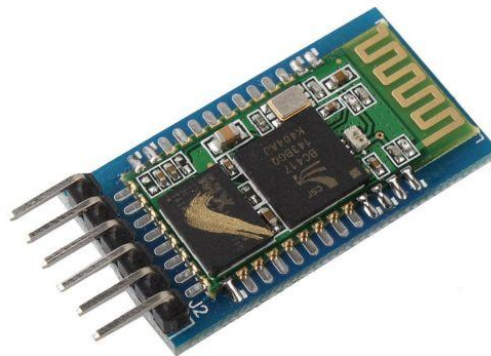
για αυτό το λόγο επιλέχτηκε καθώς το μέγεθος της με βάση τα χαρακτηριστικά της είναι κατάλληλα για την συγκεκριμένη κατασκευή (Σχήμα 2.12).



Σχήμα 2.12 : Ηλεκτρονικό κύκλωμα επιλογής φοράς κίνησης στους κινητήρες

## 2.6 ΣΥΣΚΕΥΗ ΑΣΥΡΜΑΤΗΣ ΕΠΙΚΟΙΝΩΝΙΑΣ (BLUETOOTH)

Το ρομποτικό σύστημα διαθέτει τη δυνατότητα ασύρματης επικοινωνίας με απομακρυσμένο σταθμό βάσης (H/Y), προκειμένου να είναι ευκολότερο ο προγραμματισμός του καθώς και ο έλεγχος του με εξελιγμένες στρατηγικές ελέγχου που δεν μπορεί να υποστηρίξει τοπικά η υπολογιστική μονάδα που φέρει. Η επικοινωνία επιτυγχάνεται μέσω Bluetooth και συγκεκριμένα του μοντέλου HC-06 (Σχήμα 2.13).



Σχήμα 2.13 : Συσκευή ασύρματης μετάδοσης δεδομένων (Bluetooth)



## 2.7 ΤΡΟΦΟΔΟΣΙΑ ΣΥΣΤΗΜΑΤΟΣ

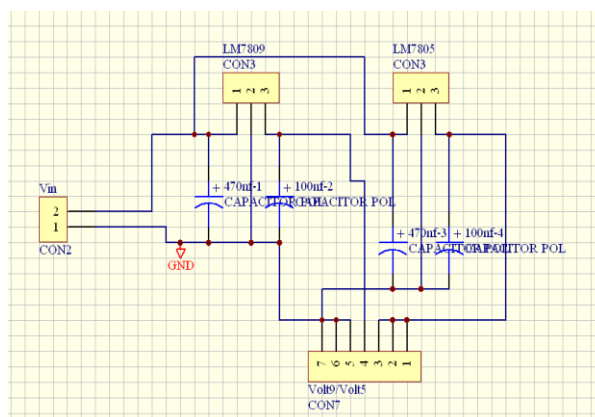
Μετά από έρευνα και μελέτη για την κατάλληλη πηγή για την τροφοδοσία του συστήματος καταλήξαμε στη χρήση μπαταριών τεχνολογίας LiPo (Lithium-ion polymer battery, ιόντων λιθίου). Οι μπαταρίες αυτού του τύπου είναι κατάλληλες για αυτόνομα συστήματα όπως ελικόπτερα, σκάφη, αεροπλάνα, αυτοκίνητα αλλά και άλλου είδους εφαρμογές όπως κινητά τηλέφωνα και Laptops, διότι έχουν την ιδιότητα να αποθηκεύουν μεγάλες ποσότητες ενέργειας σε σχέση με τις κλασικές μπαταρίες και επιπλέον ζυγίζουν 20% λιγότερο. Έγινε επιλογή του συγκεκριμένου τύπου μπαταριών λόγω: α) της εκτεταμένης τους χρήσης για τροφοδοσία μικρών ρομποτικών συστημάτων, του βάρους τους και του μεγέθους τους, β) της δυνατότητας επαναφόρτισης και γ) της δυνατότητας να παρέχουν υψηλά ρεύματα έτσι ώστε οι κινητήρες να μπορούν να αποδώσουν το μέγιστο της ικανότητας τους την στιγμή που θα τους ζητηθεί. Η τάση της μπαταρίας που επιλέγεται είναι στα 11,1V και η χωρητικότητά της είναι στα 1800 mAh. Η επιλογή αυτή έγινε βάση των χαρακτηριστικών της κατανάλωσης των κινητήρων του συστήματος καθώς αποτελούν ένα από τα βασικότερα κριτήρια της κατανάλωσης ισχύος. Ο κάθε ένας κινητήρας καταναλώνει στην κανονική του λειτουργία 100 mAh αλλά αυτό μπορεί να αλλάξει κατά την διάρκεια της λειτουργίας του και να φτάσει μέχρι και τα 500 mAh στιγμιαία ανά κινητήρα. Με βάση τα παραπάνω επιλέγει η χωρητικότητα της μπαταρίας να είναι στα 1800 mAh έτσι ώστε το σύστημά να έχει αρκετή ώρα αυτονομίας. Δεν επιλέγει μεγαλύτερη μπαταρία καθώς θα είχε σαν συνέπεια την αύξηση του βάρους της κατασκευής. Ταυτόλοια ηλεκτρονικά συστήματα απαιτούν μικρή κατανάλωση ισχύος που είναι αμελητέα σε σύγκριση με την κατανάλωση των κινητήρων. Η μπαταρία που τελικά επιλέχθηκε παρουσιάζεται στο σχήμα 2.14.



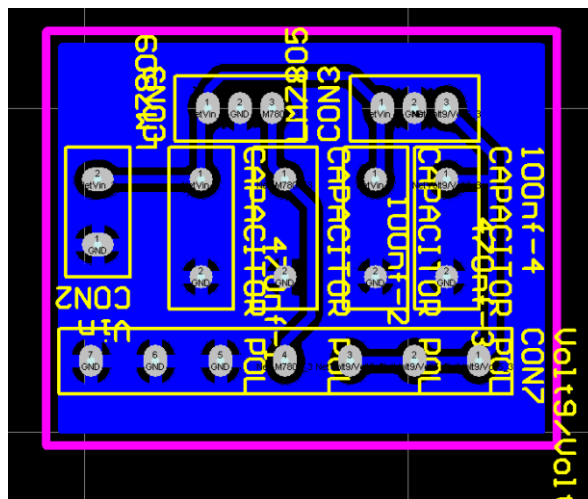
Σχήμα 2.14 : Μπαταρία ιόντων λιθίου 11,1V 1800mah

## 2.8 ΠΛΑΚΕΤΑ ΣΤΑΘΕΡΟΠΟΙΗΣΗΣ ΤΑΣΗΣ

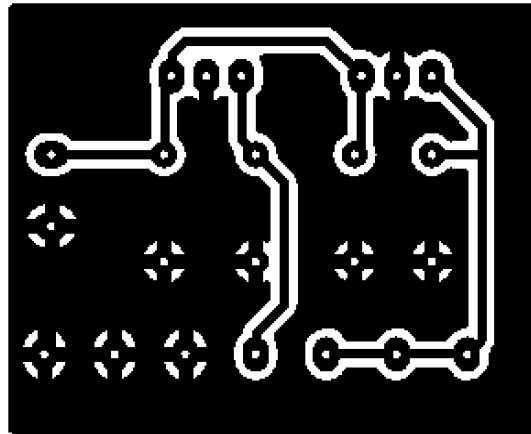
Επειδή όλα τα ηλεκτρονικά μέρη του συστήματος δεν τροφοδοτούνται με την ίδια τάση και η τάση της μπαταρίας κατά την διάρκεια λειτουργίας της, από πλήρης φορτισμένη μέχρι να αδειάσει, δεν είναι σταθερή κατασκευάστηκε πλακέτα σταθεροποίησης τάσης που τροφοδοτεί ταυτόχρονα την πλακέτα του μικροεπεξεργαστή με τάση 9V και τα υπόλοιπα ηλεκτρονικά μέρη του συστήματος με τάση 5V. Όπως φαίνεται στο σχήμα 2.15, 2.16 και 2.17 σχεδιάστηκε και υλοποιήθηκε η πλακέτα σταθεροποίησης τάσης. Για την υλοποίηση της χρειάστηκαν ένας σταθεροποιητής τάσης LM7809 που σταθεροποιεί την τάση στα 9V, ένας επιπλέον σταθεροποιητής τάσης LM7805 που σταθεροποιεί την τάση στα 5V και στους δύο σταθεροποιητές τάσης τοποθετήθηκε ψήκτρα για καλύτερη λειτουργία του ηλεκτρονικού κυκλώματος και για την αποφυγή υπερθέρμανσης του. Στις εισόδους και εξόδους των σταθεροποιητών τοποθετήθηκαν πυκνωτές οι οποίοι βοηθούν στην εξομάλυνση της τάσης για να έχουμε ακόμα καλύτερη σταθεροποίηση. Στις εισόδους χρησιμοποιήθηκαν 470nF πυκνωτές και στις εξόδους 100nF. Επίσης για την σύνδεση των καλωδίων τοποθετήθηκαν κλέμες τυπωμένου κυκλώματος.



Σχήμα 2.15 : Σχηματικό πλακέτας σταθεροποίησης τάσης



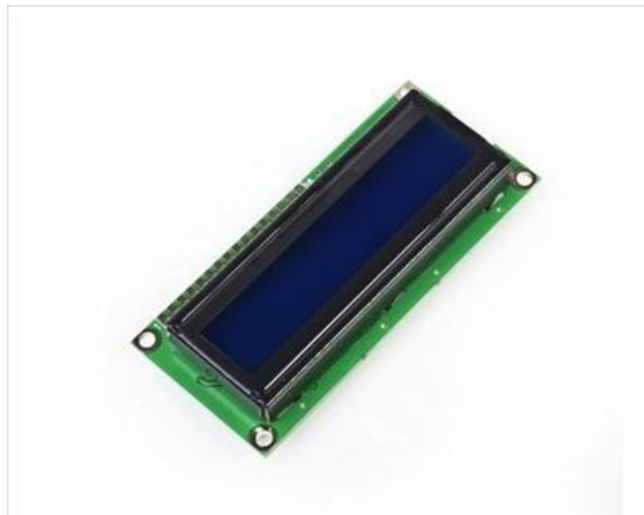
Σχήμα 2.16 : PCB πλακέτας σταθεροποίησης τάσης



Σχήμα 2.17 : PCB πλακέτας σταθεροποίησης τάσης

## 2.9 LCD ΟΘΟΝΗ

Στο ρομποτικό σύστημα έχει προσαρμοστεί μια οθόνη LCD ώστε να είναι δυνατή η παρακολούθηση των διαφόρων λειτουργικών παραμέτρων και η λήψη διορθωτικών κινήσεων από τον προγραμματιστή .



Σχήμα 2.18 : LCD οθόνη 2X16 χαρακτήρων

## 2.10 ΣΥΝΔΕΣΜΟΣ ΜΕΤΑΞΥ ΑΞΟΝΑ ΚΙΝΗΤΗΡΑ ΚΑΙ ΤΡΟΧΟΥ

Για να μπορέσει να συνδεθεί ο άξονας του κάθε κινητήρα με τους τροχούς θα πρέπει να υπάρχει κάποιος συνδετικός κρίκος μεταξύ άξονα και τροχό. Αυτό επιτυγχάνεται με τη χρήση ενός συνδέσμου στον όπου από την μία μεριά έχει υποδοχή 6mm για να τοποθετηθεί ο άξονας του κινητήρα και από την άλλη μεριά μετατρέπεται σε μία εξαγωνική επιφάνεια η οποία ταιριάζει απόλυτα με την αντίστοιχη εξαγωνική εσοχή του τροχού.



Σχήμα 2.19 : Σύνδεσμος μεταξύ άξονα κινητήρα και τροχού

## 2.11 ΤΡΟΧΟΙ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ

Στο τελικό στάδιο της κατασκευής του δίτροχου ρομποτικού συστήματος χρησιμοποιήθηκαν τροχοί διαμέτρου 65mm με εξαγωνική εσοχή έτσι ώστε να μπορούν να συνδεθούν μέσω του συνδέσμου που αναφέρθηκε παραπάνω.



Σχήμα 2.20 : Σύνδεσμος μεταξύ άξονα κινητήρα και τροχού

## 2.12 ΜΟΝΑΔΑ ΕΛΕΓΧΟΥ ΤΟΥ ΡΟΜΠΟΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

Προκειμένου να επιτευχθεί η διαχείριση των διαφορετικών ηλεκτρονικών δομικών μερών του ρομποτικού συστήματος έπρεπε να γίνει η επιλογή της κατάλληλης μονάδας ελέγχου. Μετά από έρευνα επιλέχθηκε κατάλληλος μικροεπεξεργαστής για τις ανάγκες της εργασίας με την βοήθεια του οποίου μπορούμε να υλοποιήσουμε όλες τις απαραίτητες ενέργειες που θα χρειαστούν όπως: α) ο έλεγχος της ταχύτητας περιστροφής και φοράς των κινητήρων συνεχούς τάσης που χρησιμοποιήθηκαν, β) η ασύρματη επικοινωνία του

μικροεπεξεργαστή με απομακρυσμένο ηλεκτρονικό υπολογιστή, γ) η δειγματοληψία από τα αισθητήρια, δ) η προβολή δεδομένων στην οθόνη LCD και ε) η είσοδος δεδομένων από τα ποτενσιόμετρα. Ο μικροεπεξεργαστής που επιλέχτηκε είναι ένας 8-Bit μικροεπεξεργαστή ATmega328P-PU της εταιρίας Atmel διότι καλύπτει όλες τις απαιτήσεις που χρειάζεται για να υλοποιηθεί η πτυχιακή εργασία αλλά επιπλέον έχει την δυνατότητα για ακόμα περαιτέρω αναβαθμίσεις και τροποποιήσεις. Η πλακέτα που περιέχει αυτό τον μικροεπεξεργαστή ονομάζεται Arduino Uno R3 η οποία έχει αναπτυξιακά χαρακτηριστικά δηλαδή μπορεί να δεχτεί εύκολα την προσθήκη περιφερειακών εξαρτημάτων με τη σύνδεση απλών καλωδίων. Τα βασικά χαρακτηριστικά του μικροεπεξεργαστή και της πλακέτας ανάπτυξης που επιλέχτηκαν είναι: 14 ψηφιακές επαφές οι οποίες μπορούν να χρησιμοποιηθούν είτε σαν έξοδο είτε σαν είσοδο (οι 6 από αυτές μπορούν να δώσουν απευθείας PWM σήμα εξόδου), 6 αναλογικές εισόδους ή εξόδους, κρύσταλλος στα 16 MHz, ένα κουμπί reset, μια θήρα προγραμματισμού USB και ένα φισ τροφοδοσίας συνεχούς τάσης από 7 έως 12 Volt. Το σημαντικότερο πλεονέκτημα αυτού του μικροεπεξεργαστή είναι ότι για να προγραμματιστεί χρειάζεται πηγαίο πρόγραμμα Arduino και γλώσσα προγραμματισμού Arduino η οποία είναι μια πολύ φιλική και εύχρηστη γλώσσα προγραμματισμού για τον χρήστη και διατίθεται ελεύθερα στο διαδίκτυο ([www.arduino.cc](http://www.arduino.cc)). Το κόστος κατασκευής της πλακέτας ήταν κατά πολύ μεγαλύτερο από ότι να αγοραστεί [8].

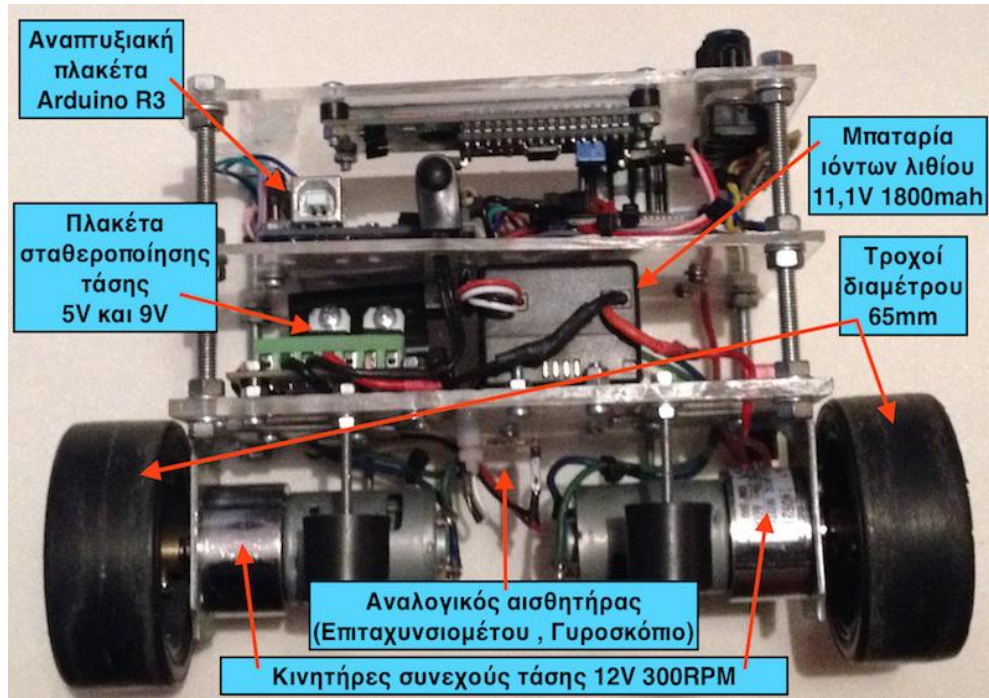


Σχήμα 2.21 : Αναπτυξιακή πλακέτα Arduino Uno

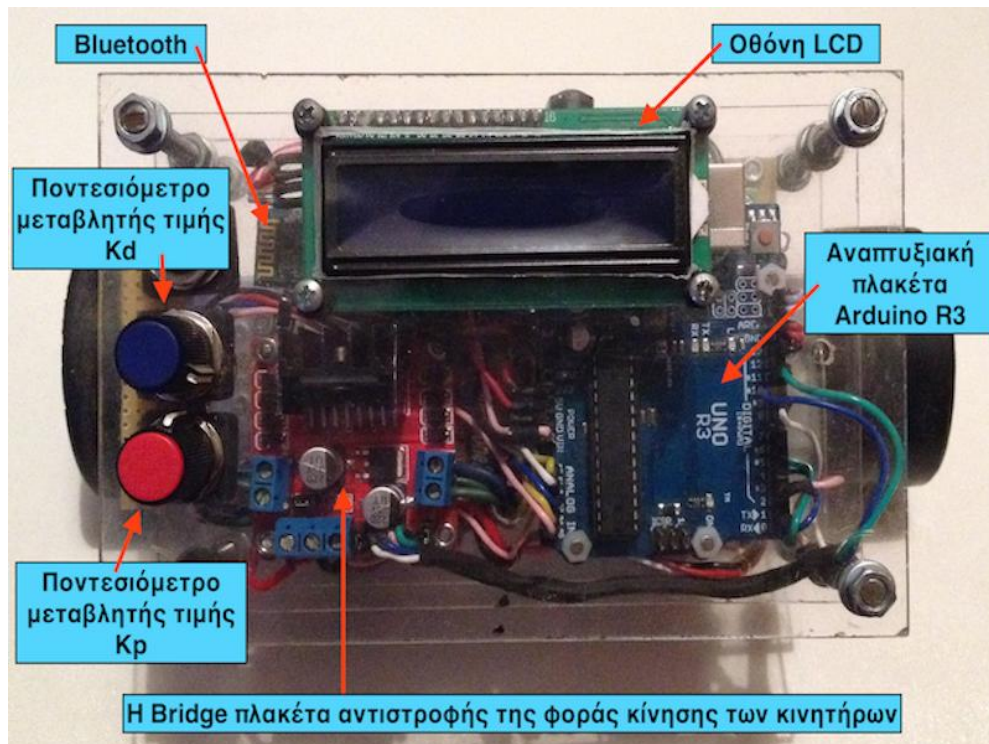
## 2.13 ΤΕΛΙΚΗ ΚΑΤΑΣΚΕΥΗ ΠΡΩΤΟΤΥΠΟΥ ΔΙΤΡΟΧΟΥ ΑΥΤΟΝΟΜΟΥ ΡΟΜΠΟΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

Η τελική μορφή του πρωτότυπου δίτροχου ρομποτικού συστήματος απεικονίζεται στα σχήματα 2.22 και 2.23. Η τελική μορφή και ταξινόμηση των ηλεκτρονικών συστημάτων αλλά και τον μηχανικών μερών πάρθηκαν για τους παρακάτω λόγους : α)καλύτερη ταξινόμηση βάρους, β)απομόνωση των ηλεκτρικών κυκλωμάτων από τα μηχανικά μέρη και γ)εύκολη πρόσβαση για αλλαγές στον κώδικα μέσω της USB θήρας. Η τελική μορφή αποτελείται από τρία επίπεδα τα οποία είναι κατασκευασμένα από πλαστικό και οι διαστάσεις τους είναι ακριβώς ίδιες και στα τρία αυτά κομμάτια. Τα πλαστικά αυτά έχουν το κάθε ένα μήκος 155mm, πλάτος 100mm και ύψος 3mm. Χρησιμοποιήθηκε αυτό το υλικό για την τελική μορφή διότι το πλαστικό είναι ένα ελαφρύ υλικό και μπορεί να το επεξεργαστεί κανείς με εύκολο τρόπο παραδείγματος χάρη μπορεί να τρυπηθεί με ένα απλό τρυπάνι και να κοπεί με εύκολο τρόπο. Για την ένωση των τριών αυτών πλαστικών χρησιμοποιήθηκαν τέσσερις βίδες με διάμετρο 5mm καθώς και εικοσιτέσσερα παξιμάδια και ανάλογες ροδέλες. Στη πρώτη πλαστική πλατφόρμα από την κάτω της μεριά τοποθετήθηκαν οι βάσεις στήριξης των κινητήρων οι οποίες είναι κατασκευασμένες από μέταλλο και έχουν τη μορφή Γ (Σχήμα 2.22). Πάνω σε αυτές τοποθετήθηκαν οι κινητήρες και στη συνέχεια τοποθετήθηκαν οι σύνδεσμοι που συνδέουν τον άξονα του κάθε κινητήρα με τον τροχό και τελικά τοποθετήθηκαν και οι ίδιοι οι τροχοί. Στη πρώτη πλατφόρμα στο μέσο της τοποθετήθηκαν τα αισθητήρια για την καλύτερη μέτρηση καθώς και τέσσερις άξονες για τη στήριξη του ρομποτικού συστήματος όταν αυτό είναι απενεργοποιημένο. Ακόμα στην ίδια πλατφόρμα αλλά από την επάνω της μεριά τοποθετήθηκε η πηγή ενέργειας, δηλαδή η μπαταρία, το κύκλωμα σταθεροποίησης τάσης και ο κεντρικός διακόπτης ενεργοποίησης και απενεργοποίησης του ρομποτικού συστήματος. Στη δεύτερη πλατφόρμα στο επάνω της μέρος τοποθετήθηκε η πλακέτα H-Bridge ή αλλιώς πλακέτα επιλογής φοράς κίνησης των κινητήρων. Επίσης δίπλα από την πλακέτα H-Bridge τοποθετήθηκε η αναπτυξιακή πλακέτα Arduino καθώς και η ασύρματη επικοινωνία Bluetooth. Στη τελευταία πλατφόρμα τοποθετήθηκαν η LCD οθόνη και τα ποτενσιόμετρα τα οποία ελέγχουν κάποιες παραμέτρους μέσα στο κώδικα. Οι τελικές διαστάσεις του ρομποτικού συστήματος με όλα τα συστήματα ελέγχου, τα μηχανικά και τα ηλεκτρικά συστήματα τοποθετημένα επάνω του έχουν μήκος 210mm, πλάτος 100mm, ύψος 160mm και το βάρος του είναι 1040 γραμμάρια.





Σχήμα 2.22 : Τελικό πρωτότυπο δίτροχου αυτόνομου ρομποτικού συστήματος



Σχήμα 2.23 : Τελικό πρωτότυπο δίτροχου αυτόνομου ρομποτικού συστήματος

Το συνολικό κόστος της κατασκευής και τα υλικά που χρησιμοποιήθηκαν παρουσιάζονται στον πίνακα 2.1.

ΥΛΙΚΑ	ΠΟΣΟΤΗΤΑ	ΤΙΜΗ ΑΝΑ ΤΕΜΑΧΙΟ	ΣΥΝΟΛΙΚΗ ΤΙΜΗ
UNO R3 ATmega328P Arduino Compatible Board	1	9,90	9,90
1602 Character HD44780 LCD Display Module	1	2,50	2,50
Wireless Bluetooth HC-06 Serial Backplane RF Transceiver for AVR Arduino RS232	1	4,50	4,50
Board Module Port IIC/I2C/TWI/SPI Serial Interface For Arduino 1602 LCD SS	1	1,70	1,70
1800mAH 11.1V Li-Po Battery	1	18,70	18,70
6mm Shaft Motor Flexible Coupling Coupler	2	4,10	8,20
6mm Shaft 2 Pin Connector Cylindrical Gear Box Motor DC 12V 300RPM	2	16,50	33,00
5DOF module accelerometer and dual-axis gyro instrument IDG650 ADXL335	1	19,50	19,50
Regulator board 5V and 9V	1	5,70	5,70
L298N Stepper Motor Drive Controller Dual H Bridge Module Board	1	2,30	2,30
12mm Hex Joint Star Wheel Rims Arrow Grain Rubber Tires For Electric Car	2	1,80	3,60
ΣΥΝΟΛΟ			109,60

*Πίνακας 2.1 : Υλικά που χρησιμοποιήθηκαν και το κόστος τους*



## ΚΕΦΑΛΑΙΟ 3

### ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

#### 3.1 Ο ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΗΣ

Ο Arduino είναι ένας μικροεπεξεργαστής στον οποίο περιέχονται όλα τα κυκλώματα σε ενιαίο σύστημα, δηλαδή μια απλή αναπτυξιακή πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροεπεξεργαστή με εισόδους/εξόδους, που μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Ο Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, Super Collider. Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προσυναρμολογημένες. Το 2005, ξεκίνησε μια προσπάθεια προκειμένου να φτιαχτεί μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο φθηνή από άλλα πρωτότυπα συστήματα διαθέσιμα εκείνη την περίοδο. Οι ιδρυτές Massimo Banzi και David Cuartielles ονόμασαν το σχέδιο από τον Arduino της Intel και ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρέα κωμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας- την ίδια περιοχή στην οποία στεγαζόταν η εταιρία υπολογιστών Olivetti. Το σχέδιο Arduino είναι μία διακλάδωση της πλατφόρμας Wiring για λογισμικό ανοικτού κώδικα και προγραμματίζεται χρησιμοποιώντας μια γλώσσα βασισμένη στο Wiring (σύνταξη και βιβλιοθήκες), παρόμοια με την C++ με απλοποιήσεις και αλλαγές, καθώς και ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE).

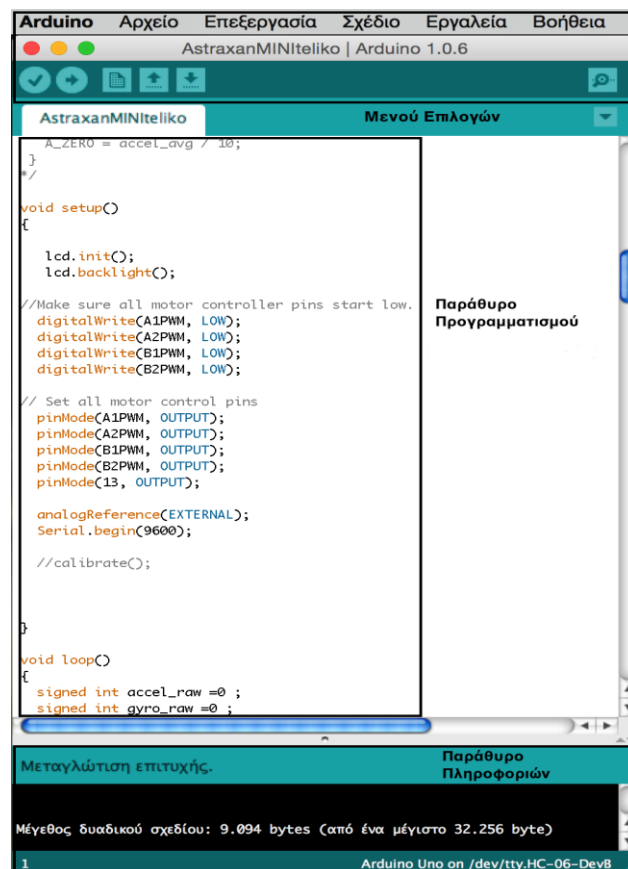
#### 3.2 ΠΡΟΓΡΑΜΜΑΤΙΖΟΝΤΑΣ ΤΟΝ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΗ

Το περιβάλλον προγραμματισμού της αναπτυξιακής πλακέτας είναι πολύ φιλικό προς το χρήστη, παρέχει όλες τις βασικές εντολές και κανόνες του προγραμματισμού που συναντάμε καθώς είναι πολύ κοντά στο τρόπο προγραμματισμού της C++. Κάθε φορά που εκτελείται ένα πρόγραμμα Arduino ακολουθεί κάποιους βασικούς κανόνες και κάποια κριτήρια. Επίσης αποτελείται από τα ακόλουθα βασικά:

- Κάθε πρόγραμμα ξεκινάει συνήθως με δυο κύριες εντολές οι οποίες είναι το `#define` και `#include`. Με την εντολή `#define` καταχωρούμε σε μια σταθερά μια συγκεκριμένη τιμή που θα χρησιμοποιηθεί στο πρόγραμμά μας ενώ με την εντολή `#include` καταχωρούμε κάθε βιβλιοθήκη που θα χρησιμοποιήσουμε στο πρόγραμμά μας, και οι δυο αυτές εντολές τοποθετούνται στην αρχή του προγράμματός μας.

- Το δεύτερο μέρος του προγράμματος είναι η ρουτίνα setup στην οποία ο χρήστης δηλώνει τις τιμές στις εισόδους και εξόδους καθώς και μερικές υπορουτίνες αρχικοποίησης οι οποίες είναι απαραίτητες για τη σωστή λειτουργία του προγράμματος αλλά και των βιβλιοθηκών του .
- Το τρίτο σημαντικό μέρος του προγράμματος είναι η ρουτίνα loop η οποία εκτελείτε επαναλαμβανόμενα μέσα στην οποία βρίσκεται το κυρίως πρόγραμμα και οι υποσυναρτήσεις ή υπ ρουτίνες οι οποίες καλούνται μέσα από το κυρίως πρόγραμμα και εκτελούν διάφορες εργασίες .
- Στο τελευταίο κομμάτι του προγράμματος βρίσκονται οι υπορουτίνες οι οποίες καλούνται από το κυρίως πρόγραμμα και του επιστρέφουν δεδομένα ή τα επιστρέφουν σε άλλες υπορουτίνες .

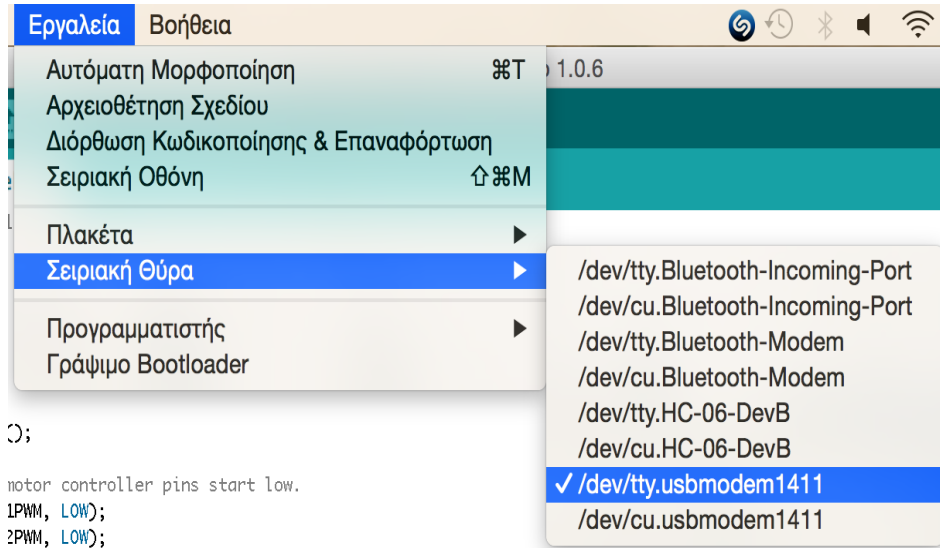
Το περιβάλλον προγραμματισμού Arduino παρουσιάζεται στο σχήμα 4.1 και αποτελείται από τρία βασικά μέρη: α) το μενού επιλογών, β)το παράθυρο προγραμματισμού και γ) το παράθυρο πληροφοριών που μας πληροφορεί για τα σφάλματα που θα υπάρξουν καθώς και για την σωστή μετάβαση του κώδικά στην αναπτυξιακή πλακέτα.



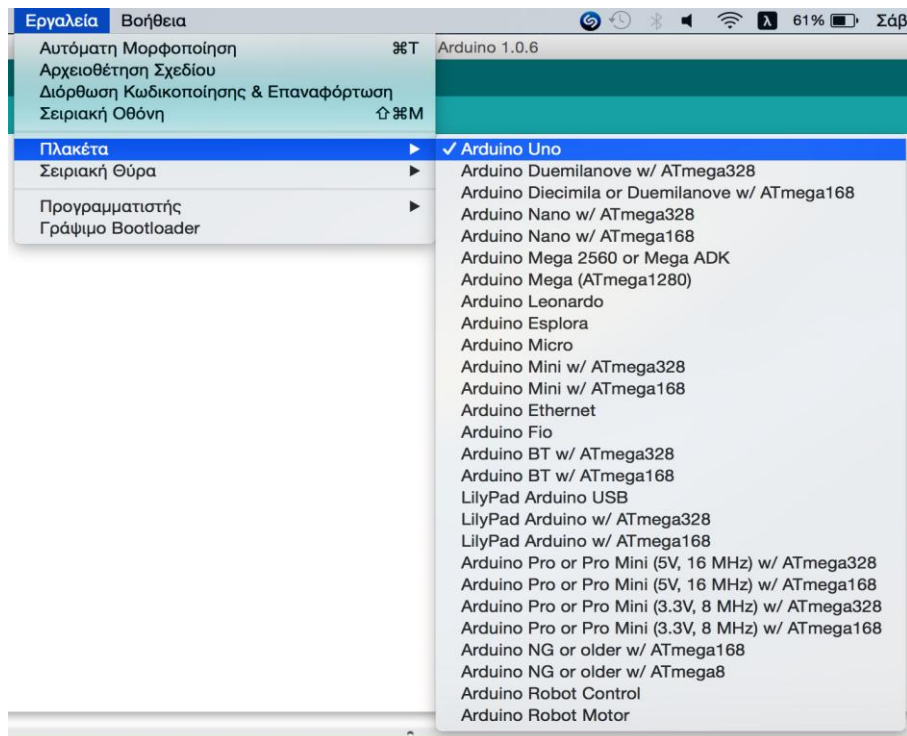
Σχήμα 3.1: Παράθυρο προγραμματισμού Arduino

Από το μενού επιλογών ο χρήστης μπορεί να πραγματοποιήσει βασικές λειτουργίες όπως αποθήκευση του κώδικα ή ανάκτηση του ήδη υπάρχοντος μέσω του «Αρχείου». Με την

επιλογή «Εργαλεία/Πλακέτα» χρήστης μπορεί να επιλέξει ποία αναπτυξιακή πλακέτα θα χρησιμοποιήσει ενώ με την επιλογή «Εργαλεία/Σειριακή θύρα» ο χρήστης έχει την δυνατότητα να ενεργοποιήσει την σειριακή επικοινωνία με μεταξύ του ηλεκτρονικού υπολογιστή και του μικροεπεξεργαστή όπως φαίνεται παρακάτω.

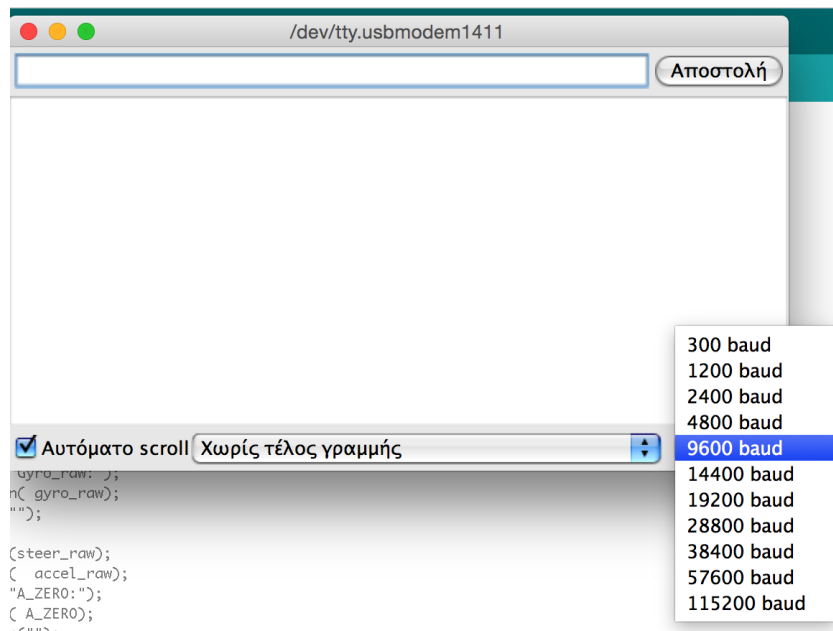


Σχήμα 3.2:Επιλογή σειριακής θύρας



Σχήμα 3.2:Επιλογή αναπτυξιακής πλακέτας Arduino

Ακόμα ο χρήστης μπορεί με την επιλογή «Εργαλεία/Σειριακή Οθόνη» να παρακολουθήσει την σειριακή και να επιλέξει το Baud rate. Αυτές οι επιλογές φαίνονται στο σχήμα 4.3.



Σχήμα 3.3: Παράθυρο παρακολούθησης σειριακής θύρας και επιλογή Baud rate

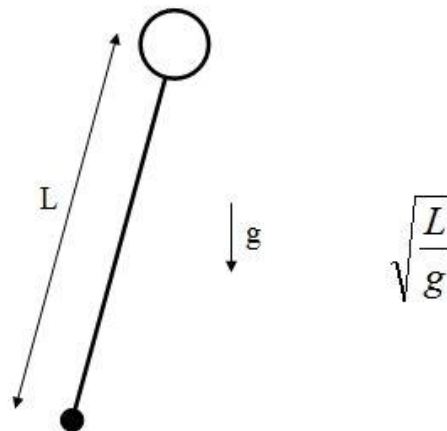
Με το παράθυρο παρακολούθησης σειριακής θύρα ο χρήστης μπορεί να παρατηρεί αποτελέσματα τα οποία στέλνονται από τον μικροεπεξεργαστή ή μπορεί ο χρήστης να στείλει κάποιες πληροφορίες που θα επηρεάσουν τον κώδικα. Αυτό επιτυγχάνεται με το κουμπί «Αποστολή» το οποίο βρίσκεται στο παράθυρο παρακολούθησης της σειριακής θύρα στο πάνω μέρος και δεξιά του παραθύρου. Επιπλέον μετά την δημιουργία και αποθήκευση του κώδικα ο χρήστης θα πρέπει να ελέγξει για τυχόν συντακτικά λάθη τα οποία έχουν γίνει. Αυτό επιτυγχάνεται με την χρήση του συντακτικού ελέγχου που κατέχει το πρόγραμμα το οποίο για να το επιτύχει ο χρήστης θα πρέπει να επιλέξει από το μενού ενεργειών «Σχέδιο/Επαλήθευση-Μεταγλώττιση». Ο χρήστης όμως θα πρέπει να πραγματοποιήσει ακόμα ένα βήμα έτσι ώστε το πρόγραμμα να μεταφερθεί από τον ηλεκτρονικό υπολογιστή στον μικροεπεξεργαστή, θα πρέπει από το μενού επιλογών να επιλέξει «Αρχείο /Φόρτωση». Τελικά με αυτόν τον τρόπο ο χρήστης έχει πραγματοποιήσει την μεταφορά του κώδικα από τον ηλεκτρονικό υπολογιστή στον μικροεπεξεργαστή. Αν όμως κάτι δεν πάει καλά σε μια από τις παραπάνω διαδικασίες τότε θα εμφανιστεί σχετικό μήνυμα στο παράθυρο πληροφοριών στο οποίο αναφέρεται το σφάλμα του πραγματοποιήθηκε και χρήστης θα πρέπει να το λάβει υπόψιν του.

## ΚΕΦΑΛΑΙΟ 4

### ΜΟΝΤΕΛΟΠΟΙΗΣΗ

#### 4.1 ΦΥΣΙΚΗ ΣΗΜΑΣΙΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

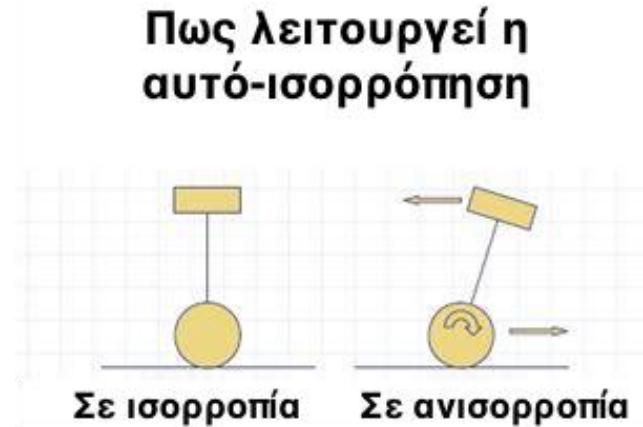
Η πλατφόρμα αυτό-ισορρόπησης μελετήθηκε από φυσικής άποψης και κατόπιν υλοποιήθηκε. Η μέθοδος που χρησιμοποιεί το ρομποτικό σύστημα αυτό είναι η μέθοδος του ανεστραμμένου εκκρεμούς [9]. Ένα ανεστραμμένο εκκρεμές είναι ένα εκκρεμές που έχει το κέντρο της μάζας πάνω από το σημείο περιστροφής του. Οι περισσότερες εφαρμογές περιορίζουν το εκκρεμές σε ένα βαθμό ελευθερίας του σε έναν άξονα περιστροφής. Ενώ ένα κανονικό εκκρεμές είναι σταθερό όταν κρέμεται προς τα κάτω, ένα ανεστραμμένο εκκρεμές είναι εγγενώς ασταθές, και πρέπει να είναι ισοσκελισμένος ενεργά προκειμένου να παραμείνει όρθιο, αυτό μπορεί να γίνει με την εφαρμογή ροπής στο σημείο περιστροφής, στο συγκεκριμένο σύστημα αυτό επιτυγχάνεται με τους κινητήρες, με την μετακίνηση του σημείου περιστροφής οριζοντίως ως μέρος ενός συστήματος ανάδρασης, αλλάζοντας την ταχύτητα περιστροφής της μάζας, αν τοποθετηθεί το εκκρεμές επί ενός άξονα παράλληλο προς τον άξονα περιστροφής έτσι δημιουργεί μια καθαρή ροπή στο εκκρεμές. Μια απλή επίδειξη από τη μετακίνηση του σημείου περιστροφής σε ένα σύστημα ανάδρασης επιτυγχάνεται με την εξισορρόπηση ενός ανάποδου σκουπόξυλο στο άκρο του δακτύλου σου [3]. Το ανεστραμμένο εκκρεμές είναι ένα κλασικό πρόβλημα στη δυναμική όπως φαίνεται και στο σχήμα 4.1 .



Σχήμα 4.1 : Μέθοδος αναστρέφω εκκρεμούς [3]

Για να ισορροπήσει το ρομποτικό σύστημα θα πρέπει η πλατφόρμα να είναι παράλληλη με την επιφάνεια στην οποία βρίσκεται το σύστημα. Αυτό επιτυγχάνεται μέσω των

αισθητήρων οι οποίοι αντιλαμβάνονται τη διαφορά της κλίσης που έχει μετατοπιστεί και μέσω του μικροεπεξεργαστή δίνουν την κατάλληλη ροπή στους κινητήρες έτσι ώστε η πλατφόρμα να έρθει στην αρχική της θέση. Όπως φαίνεται στο σχήμα 4.2 εάν η πλατφόρμα γείρει προς τα δεξιά τότε θα δοθεί στους κινητήρες εντολή να κινηθούν δεξιόστροφα έτσι ώστε η πλατφόρμα να γυρίσει στην αρχική της θέση. Το αντίστοιχο θα συμβεί εάν η πλατφόρμα κινηθεί από την αριστερή μεριά, τότε οι κινητήρες θα στραφούν αριστερόστροφα. Αυτό θα συμβαίνει συνεχώς μέχρι οπότε η πλατφόρμα να σταθεροποιηθεί.



Σχήμα 4.2 : Μέθοδος ισορρόπησης ρομποτικού συστήματος[4]

## 4.2 ΦΙΛΤΡΑ ΚΑΙ ΣΥΝΔΙΑΣΜΟΣ ΤΩΝ ΑΙΣΘΗΤΗΡΙΩΝ

Για να καταφέρουμε να ισορροπήσουμε την πλατφόρμα χρειαζόμαστε κάποια αισθητήρια και συγκεκριμένα ένα επιταχυνσιόμετρο και ένα γυροσκόπιο. Το επιταχυνσιόμετρο δεν μπορεί να δείξει τη διαφορά μεταξύ κινούμενων και ακίνητων τιμών της γωνίας και αυτό δεν ενδείκνυται για γρήγορες μεταβολές του συστήματος. Το γυροσκόπιο από την άλλη μεριά θα παρασυρθεί από την συσσώρευση λαθών, ιδικά με την ακριβή γωνία εκκίνησης και αυτό δεν ενδείκνυται για αργές μεταβολές. Οπότε για την επίτευξη της ισορρόπησης του συστήματος μπορούν να χρησιμοποιηθούν φίλτρα τα οποία συνδυάζουν και τους δύο αισθητήρες έτσι ώστε το σύστημα να ανταποκρίνεται στις γρήγορες αλλά και στις αργές μεταβολές. Τα φίλτρα που μπορούν να χρησιμοποιηθούν είναι είτε το Kalman φίλτρο είτε το Complementary φίλτρο. Το φίλτρο Kalman δεν χρησιμοποιήθηκε διότι για να λειτουργήσει χρειάζεται : α)περίπλοκες μαθηματικές πράξεις οι οποίες για να πραγματοποιηθούν χρειάζονται γνώσεις στην γραμμική άλγεβρα και β) την υποστήριξη από έναν κατά πολύ πιο δυνατό επεξεργαστικό σύστημα. Οπότε η ιδέα αυτού του φίλτρου εγκαταλείφτηκε και στη θέση του επιλέχτηκε το Complementary φίλτρο το οποίο είναι πιο κατανοητό και πιο εύκολο στην υλοποίησή του.

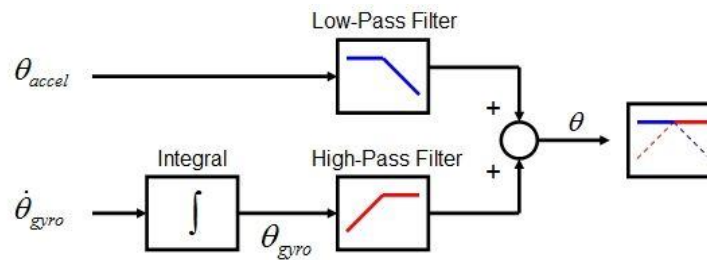
### 4.3 COMPLEMENTARY ΦΙΛΤΡΟ

Με το complementary φίλτρο συνδυάζονται και οι δύο αισθητήρες αρμονικά. Το φίλτρο αυτό χρησιμοποιεί το επιταχυνσιόμετρο για τις στατικές γωνίες, δηλαδή αφήνει να μετρήσουν μόνο οι αργές μεταβολές του συστήματος και αγνοεί τις γρήγορες ενώ χρησιμοποιεί το γυροσκόπιο μόνο για τις γρήγορες μεταβολές στις τιμές της γωνίας και αγνοεί τις αργές μεταβολές. Στο σχήμα 4.3 φαίνεται ξεκάθαρα ότι το φίλτρο του επιταχυνσιομέτρου χρησιμοποιείται για να κόβει τις γρήγορες μεταβολές και το φίλτρο του γυροσκοπίου κόβει τις αργές μεταβολές. Έπειτα συνδυάζονται για να έχει το σύστημα σε όλες τις μεταβολές της γωνίας για την κατάλληλη απόκριση είτε γρήγορες είναι αυτές είτε αργές.



Σχήμα 4.3 : Συνδυασμός φίλτρων του επιταχυνσιομέτρου και του γυροσκοπίου[5]

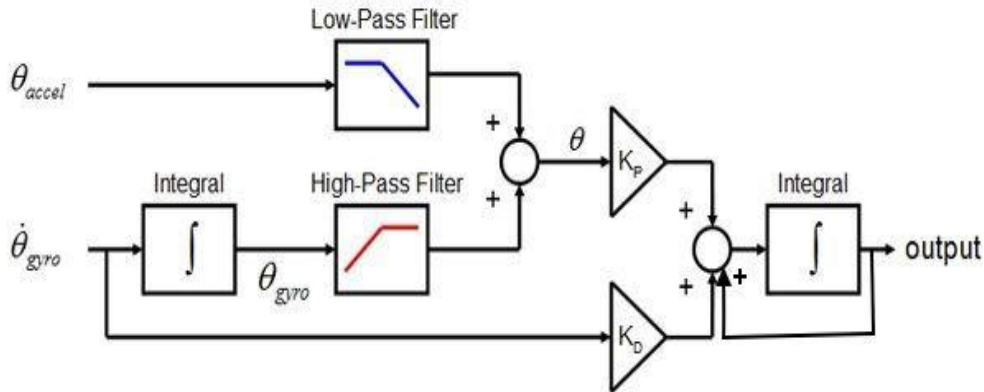
Όπως αναφέρθηκε και παραπάνω το επιταχυνσιόμετρο μετράει δύναμη ανά μάζα ενώ το γυροσκόπιο μετράει την ταχύτητα της περιστροφής. Για να μπορέσει να πραγματοποιηθεί ένα κοινό φίλτρο θα πρέπει να μετατραπεί η έξοδος του γυροσκοπίου και αντί να μετράει ταχύτητα ανά περιστροφή θα πρέπει να μετράει γωνία. Οπότε θα πρέπει να ολοκληρώσουμε την έξοδο του γυροσκοπίου και έπειτα να την φιλτράρουμε με το υψηλοπερατό φίλτρο καθώς και την έξοδο του επιταχυνσιομέτρου θα την φιλτράρουμε με το χαμηλοπερατό φίλτρο όπως φαίνεται στο σχήμα 4.4 και έπειτα μόλις φιλτράρουμε και τις δύο τιμές θα προστεθούν για μια κοινή έξοδο.



Σχήμα 4.4 : Τελικό φίλτρο γωνίας επιταχυνσιομέτρου και γυροσκοπίου[6]

Όμως αυτό δεν είναι αρκετό για την υλοποίηση της συνάρτησης ισορρόπησης του ρομποτικού συστήματος. Επιπλέον θα χρειαστεί ένας αλγόριθμος ελέγχου ανάδρασης που αποφασίζει πραγματικά πώς να διορθώσει την κλίση και να κρατήσει την πλατφόρμα ισορροπημένη. Για αυτό το λόγο θα χρησιμοποιηθεί ένας PD (proportional-derivative) ελεγκτής, ο οποίος είναι υποσύνολο του PID ελέγχου. Αναλογικά (proportional) είναι μια διορθωτική ενέργεια που κλιμακώνεται ανάλογα με τη γωνία με την οποία η πλατφόρμα εκτρέπεται από την αρχική της θέση. Αν η πλατφόρμα κλίνει προς τα εμπρός δύο φορές πιο μακριά, η διορθωτική ενέργεια είναι δύο φορές πιο μεγάλη. Αυτό είναι σαν την «σταθερά του ελατηρίου» σε ένα σύστημα, εφαρμόζοντας μια δύναμη επαναφοράς όπως η πλατφόρμα κινείται μακριά από τον κατακόρυφο άξονα του συστήματος. Παράγωγο (derivative) είναι μια διορθωτική ενέργεια που κλιμακώνεται αναλογικά προς το παράγωγο της γωνίας, ή την γωνιακή τιμή της. Αν η πλατφόρμα πέφτει δύο φορές πιο γρήγορα, η διορθωτική ενέργεια είναι δύο φορές πιο μεγάλη. Αυτό είναι όπως η «σταθερά απόσβεσης» σε ένα σύστημα, εφαρμόζοντας μια δύναμη που ανθίσταται στην περιστροφή σε οποιαδήποτε κατεύθυνση. Μαζί και τα δύο αυτά αποτελούν ένα σύστημα μάζας-ελατηρίου-αμορτισέρ, με εικονικά ελατήρια και αποσβεστήρες. Η σχετική σταθερά του ελατηρίου και σταθερά απόσβεσης επηρεάζουν το πόσο το σύστημα ταλαντώνεται αλλά και πώς διορθώνει τη γωνιακή μετατόπιση. Η τελική τιμή που θα σταλεί στους κινητήρες θα είναι η έξοδος της συνάρτησης ενώ τα KD και KP είναι ρυθμισμένα χειροκίνητα για να έρθει το αποτέλεσμα της ισορρόπησης ή για το τελείως αντίθετο αποτέλεσμα που είναι το σύστημα να βρίσκεται σε αστάθεια. Μία λεπτή γραμμή διαχωρίζει το αποτέλεσμα της εξόδου των κινητήρων η οποία βασίζετε στον ελεγκτή PD μέσω του οποίου εάν υπάρχει συνεχής μετατόπιση της πλατφόρμας από την αρχική θέση που της έχει οριστεί τότε η τιμή της εξόδου θα συνεχίσει να αυξάνετε βάση της τιμής που θα έχει ο ελεγκτής PD. Είναι σαν να προστεθεί ένα επιπλέον αναπόσπαστο μέρος του συστήματος, όπως φαίνεται στο διάγραμμα μπλοκ παρακάτω. Συνήθως στα διαγράμματα ανάδρασης, η διαδρομή ανάδρασης έχει αρνητικό πρόσημο, αλλά στην ουσία εξαρτάται από τον τρόπο που οι κινητήρες του συστήματος έχουν συνδεθεί. Η έξοδος μπορεί να τροποποιηθεί είτε σε λογισμικό επίπεδο (- = αντί + =) είτε σε υλικό επίπεδο. Σε υλικό επίπεδο αρκεί μία μόνο αντιστροφή στη πόλωση των καλωδίων του κινητήρα για να φέρουν το επιθυμητό αποτέλεσμα. Αν ο ένας ή και οι δύο τροχοί αρχίζουν να κινούνται με αντίθετο αποτελεσματότε όπως εξηγήσαμε στη παράγραφο 4.1, συνδέουμε κατάλληλα τα καλώδια στους κινητήρες έτσι ώστε η φορά κίνησης των κινητήρων να είναι κατάλληλη για την ισορρόπηση της πλατφόρμας (σχήμα 4.2) ή όπως αναφέρθηκε παραπάνω αρκεί να αλλάξουμε ένα πρόσημο στο κώδικα. Στο σχήμα 4.5 φαίνεται η φιλοσοφία block διαγράμματος της συνάρτησης μεταφοράς του συστήματος.

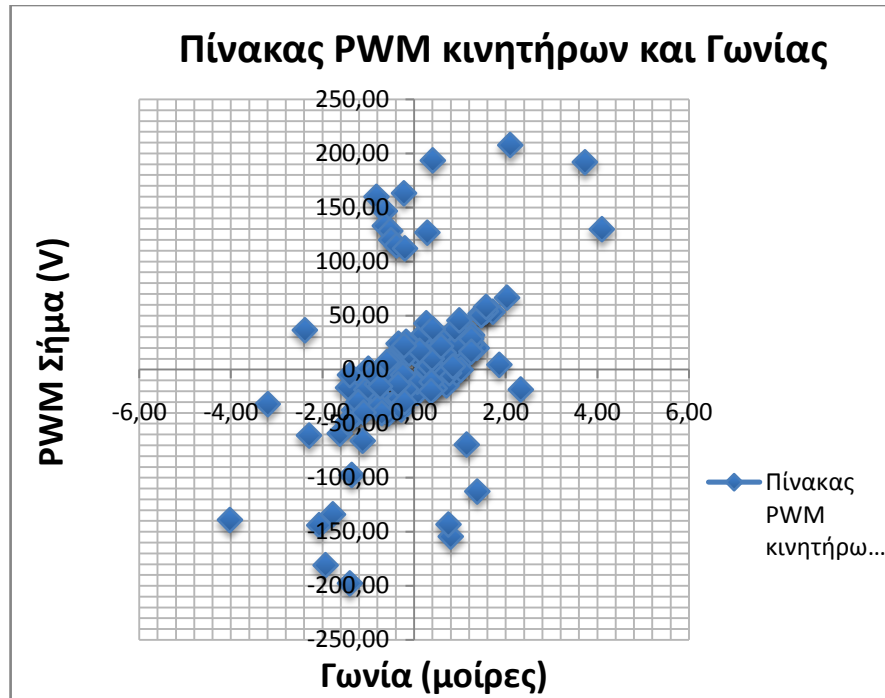




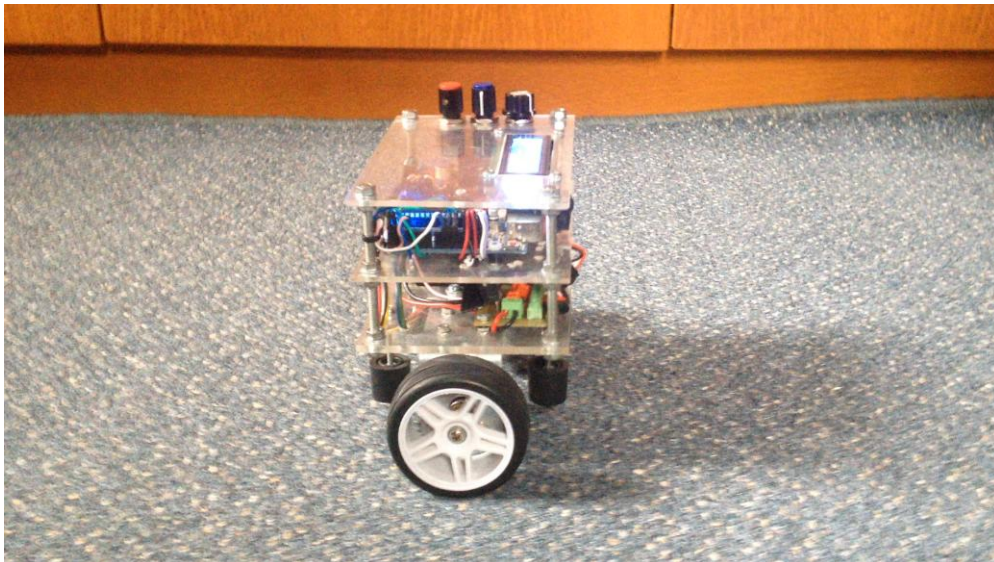
Σχήμα 4.5 : Συνάρτηση μεταφοράς σε διάγραμμα του ρομποτικού συστήματος [7]

#### 4.4 ΠΕΙΡΑΜΑΤΑ ΓΙΑ ΤΗ ΣΥΜΠΕΡΙΦΟΡΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Για να διαπιστωθεί πώς συνδέεται η γωνία κλίσης της πλατφόρμας σε σχέση με τη ταχύτητα με την οποία περιστρέφονται οι κινητήρες, για να μπορέσει η πλατφόρμα να γυρίσει στην αρχική της κατάσταση, δηλαδή να ισορροπήσει, πραγματοποιήθηκαν κάποια πειράματα και λάβαμε κάποια αποτελέσματα που παρουσιάζονται στο σχήμα 4.6. Σε αυτό το σχήμα στο Y άξονα περιλαμβάνονται οι τιμές του PWM το οποίο δίνεται στους κινητήρες για να μπορέσουμε να ρυθμίσουμε την ταχύτητα περιστροφή τους. Ενώ στο X άξονα περιλαμβάνονται οι τιμές της γωνίας της πλατφόρμας. Παρατηρώντας αυτό το διάγραμμα μπορούμε να συμπεράνουμε ότι οι περισσότερες τιμές συγκεντρώνονται κοντά στο σημείο (0,0) που αυτό σημαίνει ότι η πλατφόρμα ισορροπεί. Όταν οι τιμές απομακρύνονται από αυτό το σημείο σημαίνει ότι η πλατφόρμα δέχεται εξωτερικές παρεμβάσεις και έχει μετατοπιστεί από την αρχική της θέση. Το σύστημα προσπαθεί να την επαναφέρει στην αρχική της θέση και ανάλογα με τη μετατόπιση της γωνίας της πλατφόρμας αν αυτή είναι μεγάλη ή είναι μικρή δίνεται το κατάλληλο PWM σήμα στους κινητήρες για να επανέλθει το σύστημα στην αρχική του θέση. Παραδείγματος χάρη εάν η μετατόπιση της πλατφόρμας από την αρχική της θέση είναι μεγάλη και θετική σε σχέση με τον αισθητήρα οι κινητήρες θα λάβουν ένα αρκετά μεγάλο θετικό PWM σήμα από τον κώδικα για να φέρουν το σύστημα σε ισορροπία, ενώ εάν υπάρξει μία αρνητική μετατόπιση τότε θα δοθεί ένα αρνητικό PWM σήμα. Στο σχήμα 4.7 φαίνεται τμήματα από ένα video με το οποίο διακρίνεται ξεκάθαρα αυτό που αναφέρθηκε παραπάνω, δηλαδή όταν το ρομποτικό σύστημα επηρεαστεί από κάποια εξωτερική παρέμβαση (εδώ είναι η επίδραση από το χέρι) αυτό αντιδρά κατάλληλα βάση του κώδικα, των ηλεκτρονικών και μηχανικών μερών του για να επανέλθει στη αρχική του θέση η οποία είναι η ισορρόπηση του.



Σχήμα 4.6 : Διάγραμμα PWM των κινητήρων και της γωνίας της πλατφόρμας



Σχήμα 4.7 : Video ισορρόπησης ρομποτικού συστήματος



(α)

(β)

(γ)



(δ)

(ε)

(ζ)

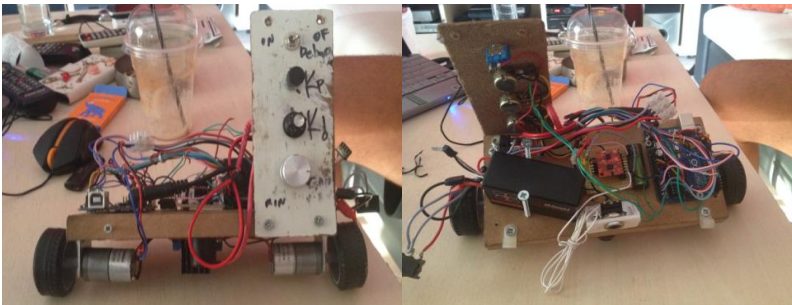
Σχήμα 4.8 : Στιγμιότυπα κίνησης του ρομποτικού συστήματος (α), (β), (γ), (δ), (ε) και (ζ)

## ΚΕΦΑΛΑΙΟ 5

### ΣΥΜΠΕΡΑΣΜΑΤΑ/ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

#### 5.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Κατά την σχεδίαση και υλοποίηση 1<sup>ο</sup> πρωτότυπου δίτροχου ρομποτικού συστήματος αυτό ισορρόπησης όπως φαίνεται στα σχήματα 5.1 (α), (β), (γ) και (δ) είχε κάποια σχεδιαστικά και λειτουργικά προβλήματα τα οποία ήταν : α) η κατανομή του βάρους του 1<sup>ο</sup> πρωτότυπου δεν είχε μοιραστεί ομοιόμορφα με αποτέλεσμα να καταπονείται ο κινητήρας στον οποίο το βάρος του δίτροχου συστήματος συσσωρευόταν, β) η ισχύς των κινητήρων των οποίων ήταν πολύ μικρή σε σχέση με το βάρος της κατασκευής και είχε ως αποτέλεσμα να μην μπορούν να αντεπεξέλθουν στις απαιτήσεις του συστήματος και στις γρήγορες εναλλαγές της κίνησης των κινητήρων, γ) η διάμετρος των τροχών έπρεπε να είναι μεγαλύτερη για ομαλότερη κύλιση και δ) η έλλειψη πλακέτας σταθεροποίησης της τάσης για τα διάφορα ηλεκτρονικά κυκλώματα τα οποία αποτελούν το ρομποτικό σύστημα.



(α)

(β)



(γ)

(δ)

Σχήμα 5.1 : 1<sup>ο</sup> Πρωτότυπο (α) Μπροστά όψη, (β) Πίσω όψη, (γ) Πλαϊνή όψη, (δ) Πλαϊνή όψη



Στο 2<sup>ο</sup> πρωτότυπο που κατασκευάστηκε λήφθηκαν υπόψιν όλα τα παραπάνω συμπεράσματα και προβλήματα για την βελτίωση και την εξέλιξη του ρομποτικού αυτού συστήματος. Για αυτό το λόγο προστέθηκαν κάποια επιπλέον χαρακτηριστικά όπως : α) οθόνη LCD για την ανάγνωση σημαντικών ενδείξεων, β) τοποθέτηση Bluetooth για την ασύρματη μετάδοση δεδομένων σε ηλεκτρονικό υπολογιστή, γ) τοποθέτηση μεταλλικών βάσεων στήριξης των κινητήρων και δ) μεγαλύτεροι τροχοί με μαλακά λάστιχα με αποτέλεσμα τη μείωση των κραδασμών και το σύστημα να έχει καλύτερη απόκριση, δ) τοποθετήθηκαν οι κινητήρες A608 και τα εσωτερικά γρανάζια είχαν σχεδόν αμελητέα ταλάντωση. Το συγκεκριμένο ρομποτικό σύστημα στο μέλλον μπορεί να δεχτεί επιπρόσθετες βελτιώσεις όπως : α) να τοποθετηθεί GPS για να μπορεί να καταγράφει μέσω μίας κάρτας αποθήκευσης τη διαδρομή που πραγματοποίησε, β) να εγκατασταθεί ασύρματος τηλεχειρισμός για απομακρυσμένο έλεγχο από το χρήστη και γ) να τοποθετηθεί μία κάμερα η οποία θα μεταδίδει ασύρματα στο χρήστη δεδομένα εικόνας για τον καλύτερο έλεγχο του συστήματος.

## 5.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΦΑΡΜΟΓΕΣ

Το δίτροχο ρομποτικό σύστημα το οποίο υλοποιήθηκε για ερευνητικούς σκοπούς μπορεί να χρησιμοποιηθεί ως βάση για να κατασκευαστεί και να λειτουργήσει ένα όχημα μεγαλύτερης κλίμακας. Με βάση αυτό τον σχεδιασμό έγιναν κάποιες προσπάθειες ανάπτυξης ενός αντίστοιχου οχήματος ικανού να φέρει φορτίο ίσο με το βάρος ενός ενήλικα ατόμου (σχήμα 5.2). Σε αυτό το σχήμα μπορούμε να διακρίνουμε το μεταλλικό πλαίσιο, τους άξονες και τους τροχούς που είναι ενσωματωμένοι πάνω στον άξονα το οποίο σαν σύνολο μπορεί να μετατραπεί σε ένα δίτροχο όχημα μεταφοράς και μετακίνησης ενός επιβάτη.



Σχήμα 5.2 : Αυτό ισορροπούμενο δίτροχο όχημα

Στο σχήμα 5.3 φαίνεται ο τροχός του δίτροχου οχήματος στον οποίο έχει τοποθετηθεί ένα γρανάζι για να μπορεί μέσω μιας αλυσίδας να μεταδίδεται η κίνηση από τον κινητήρα στο τροχό.

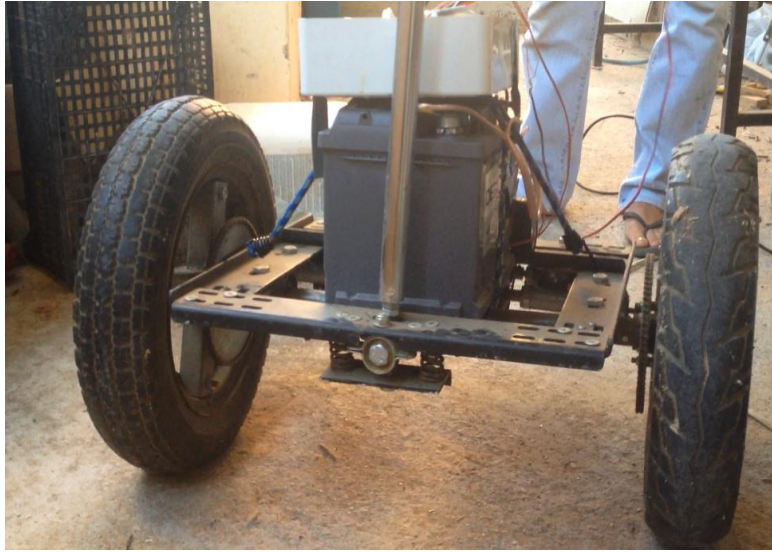


*Σχήμα 5.3 : Τροχός αυτό ισορροπούμενο δίτροχου οχήματος*

Στα σχήματα 5.4 και 5.5 φαίνεται το δίτροχο όχημα να αυτό ισορροπεί καθώς έχουν τοποθετηθεί τα ηλεκτρονικά μέρη του και η πηγή ενέργεια με την οποία τροφοδοτείται το σύστημα.



*Σχήμα 5.4 : Αυτό ισορροπούμενο δίτροχο όχημα*



*Σχήμα 5.5 : Αυτό ισορροπούμενο δίτροχο όχημα*

Στο σχήμα 5.6 το δίτροχο όχημα μπορεί να ισορροπήσει με έναν επιβάτη με επιτυχία. Πάρα τη λειτουργικότητά του, το συγκεκριμένο όχημα προκειμένου να είναι πλήρως λειτουργικό απαιτείται σειρά βελτιώσεων τόσο στον σχεδιασμό του (βρίσκεται ακόμα σε μορφή πρώιμου πρωτοτύπου) όσο και στα ηλεκτρονικά και στο σύστημα κατανομής ενέργειας του.



*Σχήμα 5.6 : Αυτό ισορροπούμενο δίτροχο όχημα*



Στο σχήμα 5.7 (α) και (β) παρουσιάζονται εικόνες από (α) τη προσπάθεια για ισορρόπηση του δίτροχου οχήματος με τη προσθήκη επιβάτη και στο σχήμα 5.7 (β) φαίνεται η χειροκίνητη δοκιμή λειτουργίας των κινητήρων του δίτροχου οχήματος.



(α)

(β)

*Σχήμα 5.7 : (α) Video δοκιμής ισορρόπησης δίτροχου συστήματος, (β) Video δοκιμής λειτουργίας κινητήρων*



## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1][http://www.wpi.edu/Pubs/E-project/Available/E-project-012012-002422/unrestricted/Two-Wheel\\_Self\\_Balancing\\_Robot.pdf](http://www.wpi.edu/Pubs/E-project/Available/E-project-012012-002422/unrestricted/Two-Wheel_Self_Balancing_Robot.pdf)
- [2][http://projects.csail.mit.edu/manipulation/rss06/paper\\_\\_designing\\_a\\_self\\_stabilizing\\_robot\\_for\\_dynamic\\_mobile\\_manipulation\\_\\_deegan.pdf](http://projects.csail.mit.edu/manipulation/rss06/paper__designing_a_self_stabilizing_robot_for_dynamic_mobile_manipulation__deegan.pdf)
- [3]<http://www.instructables.com/id/Segstick/step1/Physics-says-its-easier-to-build-a-full-size-self-/>
- [4]<http://www.instructables.com/id/Self-Balancing-Robot/step2/Physics/>
- [5]<https://www.youtube.com/watch?v=6iSl4WL1PkI#t=10>
- [6]<http://www.instructables.com/id/Segstick/step10/A-very-flattering-filter/>
- [7]<http://www.instructables.com/id/Segstick/step11/The-control-part-of-the-controller/>
- [8]<http://el.wikipedia.org/wiki/Arduino>
- [9][http://en.wikipedia.org/wiki/Inverted\\_pendulum](http://en.wikipedia.org/wiki/Inverted_pendulum)
- [10]<http://praguemonitor.com/2013/05/03/history-segway-pt>

## ΠΑΡΑΡΤΗΜΑ

### 1.1 ΑΝΑΛΥΣΗ ΤΟΥ ΚΩΔΙΚΑΣ

Ένα ρομποτικό σύστημα εκτός το υλικό και κατασκευαστικό κομμάτι αποτελείται και από λογισμικό. Αυτό το κομμάτι είναι ο κώδικας του συστήματος ο οποίος είναι ένα από τα πιο βασικά μέρη του. Στην αρχή του κώδικα τοποθετούμε κάποιες έτοιμες βιβλιοθήκες οι οποίες έχουν γραφτεί από την εταιρία που παρέχει το λογισμικό προγραμματισμού για τη διευκόλυνση του χρήστη. Έπειτα ακολουθεί η δήλωση των μεταβλητών και η αρχικοποίηση κάποιων τιμών που θα χρειαστούν στο κώδικα. Μετά μέσα στο setup ορίζουμε ποία άκρα του μικροεπεξεργαστή θα είναι έξοδοι και ποία θα είναι είσοδοι έτσι ώστε να μπορεί να λαμβάνει από τα αισθητήρια τις πληροφορίες, να τις επεξεργάζεται και έπειτα να τις δίνει στα άκρα της εξόδου. Αμέσως μετά ακολουθεί το κυρίως πρόγραμμα το οποίο αποτελείται από : α) αλγόριθμους οι οποίοι καθορίζουν τη συνάρτηση μεταφοράς του συστήματος, β) καλέσματα υποπρογράμματα τα οποία έχουν κατασκευαστεί στο τέλος του κώδικα και χρησιμοποιούνται μόνο όταν αυτά χρειάζονται από το κυρίως πρόγραμμα και γ) εντολές εκτύπωσης κάποιων σημαντικών τιμών στην οθόνη LCD. Τα υποπρογράμματα είναι μικρά προγράμματα τα οποία έχουν κάποιο συγκεκριμένο σκοπό και καλούνται από το κυρίως πρόγραμμα όποτε είναι απαραίτητα και όσες φορές αυτά χρειάζονται.

### 1.2 ΚΩΔΙΚΑΣ

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

#define A1PWM 9 //motor PWM
#define A2PWM 10 //motor PWM
#define B1PWM 3 //motor PWM
#define B2PWM 11 //motor PWM
#define A_PIN A0 //accelerometer analog input
#define G_PIN A1 //gyro analog input
#define A_GAIN 0.61 //[(degree/LSB)]
#define G_GAIN 1.409 //[(degree/s)/LSB]
#define DT 0.07 //[(s/loop) loop period]
#define A 0.962 //complementary filter constant
```

```

char temp;
float angle = 0.0;    //[degree]
float rate = 0.0;     //[degree/s]
float output = 0.0;   //[LSB] (100% voltage to motor is 255LSB)

int accel_avg = 0;
int gyro_avg = 0;
int G_ZERO=0;
int A_ZERO=0;

float KP;
float KD;
const int Kp = A3;
const int Kd = A2;
int Kp_reading;
int Kd_reading;
unsigned long time;
long last_cycle = 0;
int cycle_time;

void calibrate()
{
    //setup loop to read gyro and accel 10 times
    for (int i = 0; i < 10; i++)
    {
        // read gyro and each time, add the value to the running total
        gyro_avg = gyro_avg + analogRead(G_PIN);
        accel_avg = accel_avg + analogRead(A_PIN);
    }
    // with a sum of 10 readings, divide by 10 to get the average

```

```
G_ZERO = gyro_avg / 10 ;
A_ZERO = accel_avg / 10;
}

void setup()
{
  lcd.init();
  lcd.backlight();

  //Make sure all motor controller pins start low.
  digitalWrite(A1PWM, LOW);
  digitalWrite(A2PWM, LOW);
  digitalWrite(B1PWM, LOW);
  digitalWrite(B2PWM, LOW);

  //Set all motor control pins to outputs.
  pinMode(A1PWM, OUTPUT);
  pinMode(A2PWM, OUTPUT);
  pinMode(B1PWM, OUTPUT);
  pinMode(B2PWM, OUTPUT);
  pinMode(13, OUTPUT);

  analogReference(EXTERNAL);
  Serial.begin(9600);

  calibrate();
}

void loop()
{
  signed int accel_raw =0 ;
```

```
signed int gyro_raw =0 ;
signed int output_left = 0;
signed int output_right = 0;

//signed int steer_raw = 0;
read_pots();
// Loop speed test.
digitalWrite(13, HIGH);

//Read in the raw accelerometer and gyro
//Offset for zero angle/rate.
accel_raw = (signed int) analogRead(A_PIN) - A_ZERO;
gyro_raw = G_ZERO - (signed int) analogRead(G_PIN);

// Scale the gyro to [deg/s].
rate = (float) gyro_raw * G_GAIN;

// Complementarty filter.
angle = A * (angle + rate * DT) + (1 - A) * (float) accel_raw * A_GAIN;

// PD controller.
output += angle * KP + rate * KD;

// Clip as float (to prevent wind-up).
if(output < -255.0)
{
    output = -255.0;
}
if(output > 255.0)
{
    output = 255.0;
```

```
}

// Add/subtract steering and integerize.
output_left = (signed int) output ;/+(float) steer_raw * S_GAIN );
output_right = (signed int) output;/- (float) steer_raw * S_GAIN);

// Clip as integer.
if(output_left < -255)
{
    output_left = -255;
}
if(output_left > 255)
{
    output_left = 255;
}
if(output_right < -255)
{
    output_right = -255;
}
if(output_right > 255)
{
    output_right = 255;
}

// Choose directions and set PWM outputs.
if(output_left>=0)
{
    digitalWrite(A1PWM,LOW);
    analogWrite(A2PWM,output_left);
}
else
```

```
{
  digitalWrite(A2PWM,LOW);
  analogWrite(A1PWM,-output_left);
}

if(output_right>=0)
{
  digitalWrite(B1PWM,LOW);
  analogWrite(B2PWM,output_right);
}
else
{
  digitalWrite(B2PWM,LOW);
  analogWrite(B1PWM,-output_right);
}

// Loop speed test.
digitalWrite(13, LOW);
//LCD print values
lcd.setCursor(0,0);
lcd.print("Kp=");
lcd.println(analogRead(Kp));
lcd.setCursor(7,0 );
lcd.print("KD=");
lcd.println(analogRead(Kd));
lcd.setCursor(0,1 );
lcd.print("G_PIN");
lcd.println(G_ZERO);
lcd.setCursor(8,1 );
lcd.print("A_PIN");
lcd.println( A_ZERO);
```



```
Serial.print(angle);
Serial.print(" ");

Serial.print(output);
Serial.println("");

time_stamp();
}

void read_pots()
{
  Kp_reading = analogRead(Kp); //pin A3
  KP = map(Kp_reading, 0, 1023, 0,50);

  Kd_reading = analogRead(Kd); //pin A4
  KD = map(Kd_reading, 0, 1023, 0,2);

}

void time_stamp()
{
  // check to make sure it has been exactly 70 milliseconds since the last recorded time-
stamp
  while((millis() - last_cycle) < 70)
  {
    delay(1);
  }

  // once the loop cycle reaches 70 mS, reset timer value and proceed
  cycle_time = millis() - last_cycle;
  last_cycle = millis();
}
```