

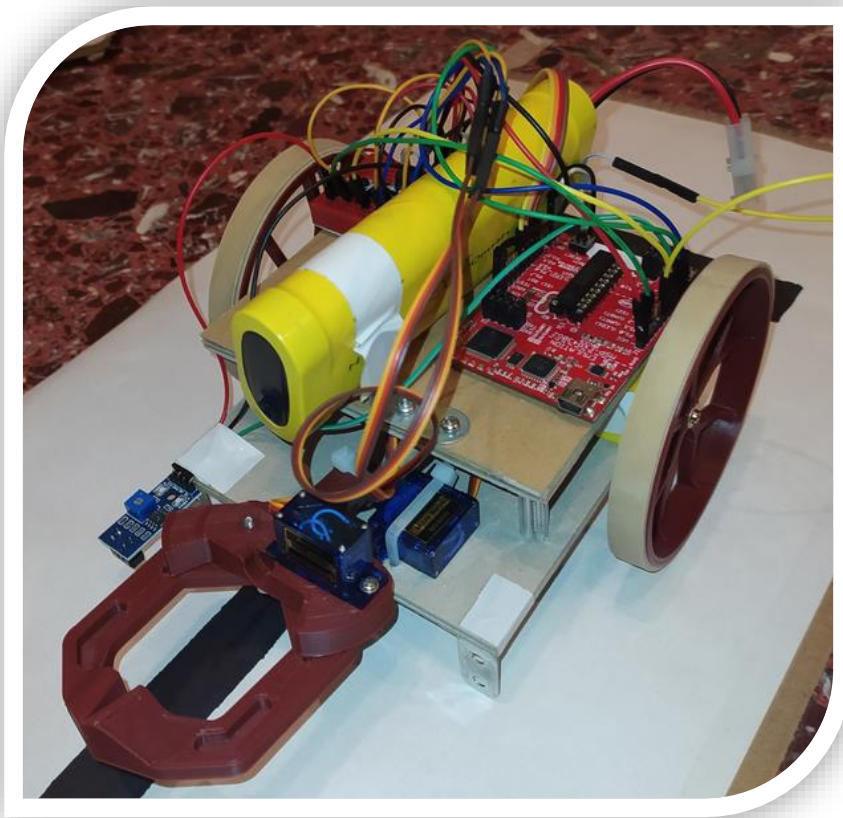


ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

Θεματική Εργασία Ρομποτικής

Ρομποτικό Όχημα με gripper



Κουτουλάκης Γρηγόρης

A.M. 18387224

Ομάδα: ΓΑ-5

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2021-2022

Περιεχόμενα	2
Πρόλογος	3
Περιγραφή-Σκοπός.....	3
Μέρος 1^ο	3
1.1 Θεωρητικό Υπόβαθρο.....	3
1.2 Εφαρμογές.....	3
1.3 PID ελεγκτής.....	3
Μέρος 2^ο	4
2.1 Υλικά Κατασκευής.....	4
2.1.1 Ηλεκτρονικά Υλικά.....	4
2.1.2 Μηχανολογικά Υλικά.....	4
2.1.3 3D Εκτύπωση.....	5
2.2 Κατασκευή Σώματος Οχήματος.....	6
2.3 Συνδεσμολογία και Ηλεκτρονικά Κυκλώματα.....	7
Μέρος 3^ο	8
3.1 Προγραμματισμός Τμημάτων.....	8
3.1.1 Προγραμματισμός Gripper.....	8
3.1.2 Προγραμματισμός Κίνησης.....	9
3.1.3 Προγραμματισμός PID ελεγκτή.....	10
3.2 Ρύθμιση Παραμέτρων.....	10
3.3 Κυρίως Πρόγραμμα και Αλγόριθμος.....	11
Μέρος 4^ο	12
4.1 Περιγραφή Τελικού Αποτελέσματος.....	12
4.2 Προβλήματα και Αντιμετώπιση.....	12
4.3 Προτάσεις Βελτιστοποίησης.....	12
4.4 Συμπεράσματα και Παρατηρήσεις.....	13
Μέρος 5^ο	13
5.1 Πηγές.....	13
5.2 Βιβλιογραφία.....	13

Πρόλογος

Περιγραφή-Σκοπός

Για την εκπόνηση της εργασίας αυτής, κατασκευάστηκε ρομποτικό όχημα, το οποίο με χρήση κατάλληλων αισθητήρων μπορεί να κινηθεί σε ελεγχόμενο χώρο ακολουθώντας μία χαρακτηριστική μαύρη γραμμή. Ακόμη, κάνοντας χρήση gripper, μπορεί να ανυψώνει συγκεκριμένα αντικείμενα και να τα μεταφέρει. Με την χρήση μπαταρίας έχει επαρκή αυτονομία για πολλαπλές διαδρομές.

Ο σκοπός της συγκεκριμένης εφαρμογής είναι:

A) Η ανάκτηση ενός αντικειμένου από τον χώρο.

B) Η επιστροφή του αντικειμένου στην κατάλληλη θέση.

Για την πλοήγηση πάνω στην γραμμή έχει χρησιμοποιηθεί ελεγκτής PID ο οποίος σε συνδυασμό με έναν αισθητήρα φωτεινότητας μπορεί να κινηθεί σε προκαθορισμένο χώρο.

Μέρος 1^ο

1.1 Θεωρητικό Υπόβαθρο

Τροχοφόρα Ρομπότ

Τροχοφόρα ρομπότ ονομάζονται τα ρομπότ τα οποία χρησιμοποιούν τροχούς με κινητήρες για την πλοήγησή τους στον χώρο. Πρόκειται για έναν από τους πιο απλούς μηχανισμούς κίνησης σε σχέση με τα αρθρωτά άκρα και είναι ιδανικά για επιφάνειες με μικρή υψομετρική διαφορά. Επίσης είναι πολύ πιο εύκολος και ο προγραμματισμός τους. Ένα σημαντικό μειονέκτημα που έχουν είναι ότι δεν μπορούν να υπερπηδήσουν εμπόδια όπως μεγάλες πέτρες, σκάλες κ.α. [1]

Grippers

Τα gripper, ή end effectors είναι ειδικοί μηχανισμοί ο οποίοι συνδέονται με το κύριο ρομπότ και έχουν ως σκοπό την αλληλεπίδραση με αντικείμενα τα οποία βρίσκονται στον περιβάλλον χώρο τον οποίο κινούνται. Υπάρχουν πολλά είδη gripper ανάλογα με την δομή και την μορφή του αντικειμένου που θέλουν να διαχειριστούν. Η πολυπλοκότητά τους αυξάνεται με την πολυπλοκότητα της λειτουργίας την οποία θα πρέπει να εκτελέσουν. [2]

1.2 Εφαρμογές

Ο συνδυασμός ενός τροχοφόρου ρομπότ με το κατάλληλο gripper έχουν εφαρμογή σε πολλούς τομείς της καθημερινότητάς του ανθρώπου. Από ρομπότ ταξινόμησης σε αποθήκες μέχρι και σε ρομπότ διάσωσης. Με την τροποποίηση των τροχών για λείο ή ανώμαλο έδαφος είναι δυνατή η πλοήγηση σε πολλαπλές επιφάνειες. Χρησιμοποιώντας ανάλογο gripper οι συνδυασμοί που μπορούν να προκύψουν είναι αμέτρητοι.

1.3 PID ελεγκτής

Ένας PID controller, ή αλλιώς Proportional integral derivative controller, είναι ένα σύστημα ελέγχου το οποίο με χρήση ανάδρασης, ρυθμίζει συνεχώς την έξοδο του συστήματος με σκοπό την σταθεροποίησή της. Ο PID υπολογίζει το σφάλμα που προκύπτει από την τιμή που του έχουμε θέσει και από την τρέχουσα έξοδο. Με τη χρήση αυτού του σφάλματος υπολογίζει το ποσό διόρθωσης με τον αναλογικό όρο (proportional), τον ολοκληρωτικό όρο (integral) και τον διαφορικό όρο (derivative) και το εφαρμόζει στο σύστημα. Η διαδικασία αυτή επαναλαμβάνεται αδιάκοπα. [3]

Μέρος 2^ο

2.1 Υλικά Κατασκευής

Πολλά από τα υλικά τα οποία χρησιμοποιήθηκαν για την κατασκευή του ρομποτικού συστήματος υπήρχαν ήδη στην κατοχή μου από παλαιότερα project, με αποτέλεσμα να διατηρηθεί το κόστος στο χαμηλό ποσό των 20€ περίπου. Τα υλικά αποτελούνται από τα ηλεκτρονικά εξαρτήματα, τα κομμάτια για την κατασκευή του σώματος και μερικά κομμάτια τα οποία εκτυπώθηκαν με χρήση 3D εκτυπωτή. Παρακάτω θα παρουσιαστούν όλα τα υλικά που χρησιμοποιήθηκαν, καθώς και ο σκοπός τους και ο λόγος που επιλέχθηκαν.

2.1.1 Ηλεκτρονικά Υλικά

Τα ηλεκτρονικά που χρησιμοποιήθηκαν είναι τα παρακάτω:

- TI Launchpad MSP430G2553: ο μικροελεγκτής που χρησιμοποιήθηκε για τον προγραμματισμό.
- 2 generic DC motors με ενσωματωμένο gearbox για την κίνηση των δύο τροχών.
- 2 Servo Motors WS-SG90 για τον έλεγχο του gripper.
- QRE1113 analog line sensor ως ο κύριος αισθητήρας για το line following.
- TCRT5000 Infrared Sensor Module ως ο δευτερεύον αισθητήρας για την αναγνώριση τερματικών γραμμών.
- L298N motor driver για τον έλεγχο των DC κινητήρων.
- Mini breadboard για την σύνδεση των εξαρτημάτων.
- L7805CV voltage regulator για την ρύθμιση τάσης στα 5V τα οποία απαιτούνται από άλλα εξαρτήματα.
- LM1086CT-3.3 voltage regulator για την ρύθμιση τάσης στα 3.3V τα οποία απαιτούνται από άλλα εξαρτήματα.
- 4 10μF electrolytic capacitors χρησιμοποιήθηκαν σε συνδυασμό με τα 2 voltage regulators (2 στο καθένα) για την σωστή λειτουργία τους.
- 2 push buttons για την επιλογή προγράμματος.
- Μπαταρία 7.2V 1700mAh για την τροφοδοσία.
- Jumper cables M-M, F-M και F-F για την διασύνδεση των εξαρτημάτων, όπως και για την σύνδεσή τους στον μικροελεγκτή.

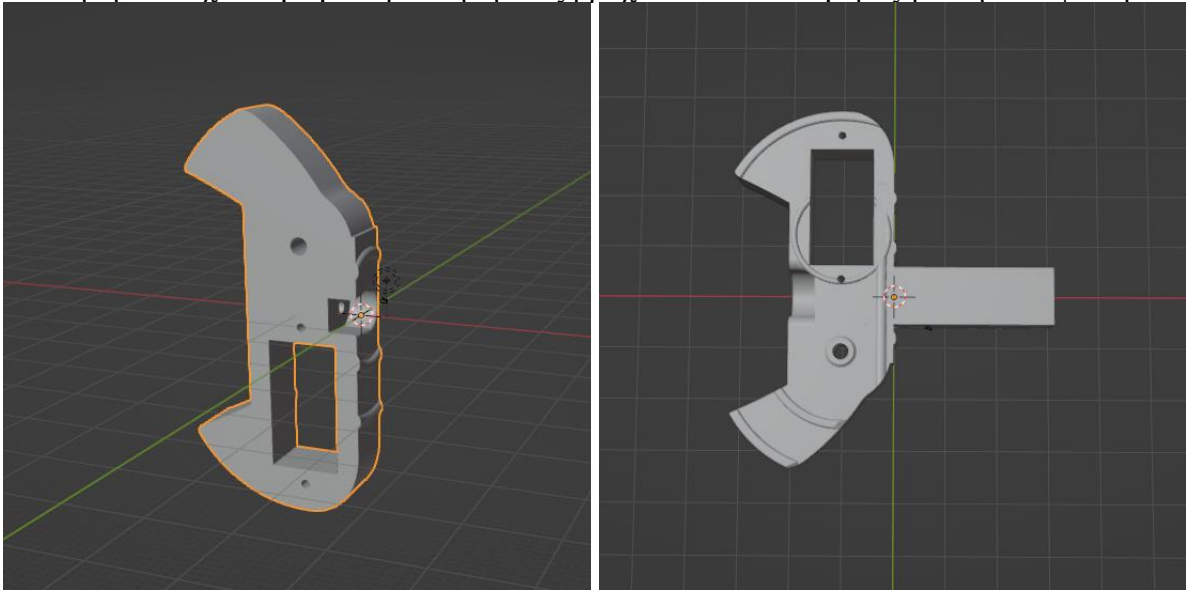
2.1.2 Μηχανολογικά Υλικά

Για την κατασκευή του σκελετού του ρομπότ χρησιμοποιήθηκαν:

- Χοντρά χαρτόνια από τετράδια για την κατασκευή της κύριας βάσης.
- Βίδες διαφόρων μεγεθών για την συγκράτηση τόσο των χαρτονιών μεταξύ τους όσο και των ηλεκτρονικών εξαρτημάτων.
- Κομμάτια από παλιά στυλό για την δημιουργία στηριγμάτων και διαχωριστικών.
- Μεταλλικές γωνίες για τα εμπρόσθια στηρίγματα, όπως και για την στήριξη των DC κινητήρων.
- Zip ties για την συγκράτηση των περισσότερων εξαρτημάτων ή όπου δεν ήταν δυνατή η χρήση βιδών.
- Μονωτική ταινία για τα σημεία όπου ούτε οι βίδες, ούτε τα zip ties δεν μπορούσαν να χρησιμοποιηθούν.

2.1.3 3D Εκτύπωση

Το gripper και οι τροχοί του ρομπότ εκτυπώθηκαν με χρήση 3D εκτυπωτή. Πιο συγκεκριμένα το gripper βρέθηκε έτοιμο σε γνωστή ιστοσελίδα για 3D σχέδια (thingiverse.com βλ.5.1 Πηγές) και τροποποιήθηκε ελάχιστα με μια προσθήκη ενός βραχίονα στο πίσω μέρος για την ανύψωσή του:



Εικόνα 1: Βάση του gripper, πριν (αριστερά) και μετά (δεξιά) την τροποποίηση

Οι τροχοί του οχήματος σχεδιάστηκαν εξολοκλήρου από την αρχή. Σαν βάση της σχεδίασης ήταν τα λάστιχα τα οποία χρησιμοποιήθηκαν, δηλαδή η διάμετρος τους ήταν λίγο μεγαλύτερη από αυτή των λάστιχων για την καλύτερη δυνατή εφαρμογή. Επίσης για την σύνδεσή των τροχών λήφθηκε υπόψη το σχήμα του άξονα των DC κινητήρων και σχεδιάστηκε αντίστοιχο, στα σημεία σύνδεσης των τροχών. Παρακάτω φαίνεται μία φωτορεαλιστική απεικόνιση των εξαρτημάτων που τυπώθηκαν:



Εικόνα 2: Φωτορεαλιστική απεικόνιση των εξαρτημάτων, αριστερά: τα άκρα του gripper, μέση: ένας τροχός και δεξιά η βάση του gripper.

2.2 Κατασκευή Σώματος Οχήματος

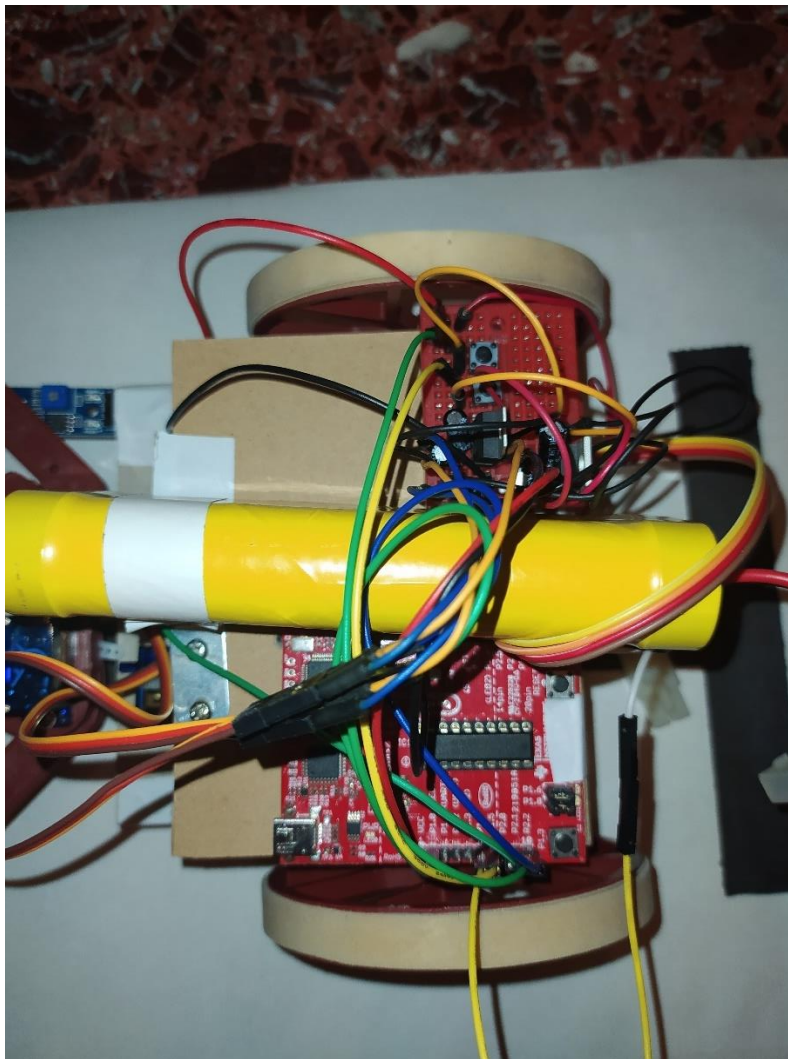
Ένας από τους στόχους της κατασκευής αυτής ήταν η προσπάθεια για το μικρότερο δυνατό μέγεθος του οχήματος, τόσο έτσι ώστε να μπορεί να μεταφερθεί εύκολα, όσο και για να μπορεί να στρίβει και να περιστρέφεται ευκολότερα, μιας και όσο πιο κοντά ήταν τα σημεία επαφής με το έδαφος στο κέντρο βάρους του, τόσο πιο απότομες στροφές θα μπορούσε να κάνει. Ακόμη είχε αποφασιστεί ότι για να επιτευχθεί αυτός ο στόχος έπρεπε να χωριστεί το ρομπότ σε 2 'ορόφους', δηλαδή 2 κύρια χαρτόνια τα οποία θα έμπαιναν το ένα πάνω στο άλλο αφήνοντας κενό ανάμεσα για λοιπά εξαρτήματα.

Έτσι μετά από πολλές δοκιμές ανακατανομής των εξαρτημάτων πάνω στα δύο βασικά χαρτόνια, που θα αποτελούσαν το κύριο σώμα, κατέληξε στη τελική δομή του. Τα βήματα που ακολούθησαν έχουν ως εξής. Αρχικά κατασκευάστηκε το gripper τοποθετώντας τα 2 servomotors στα 3d κομμάτια και με χρήση βιδών συγκρατήθηκαν στην θέση τους. Ακόμη κολλήθηκαν καλώδια πάνω στις επαφές των DC κινητήρων με τέτοιο μήκος ώστε να φτάνουν να συνδεθούν επάνω στο motor driver. Με τη χρήση τρυπανιού χειρός ανοίχθηκαν τρύπες πάνω στα χαρτόνια στα σημεία τα οποία θα συνδεόταν αργότερα στηρίγματα ή και ηλεκτρονικά εξαρτήματα. Πρώτα τοποθετήθηκαν οι DC κινητήρες (με zip ties), οι τροχοί και τα εμπρόσθια στηρίγματα (με βίδες) στο κάτω χαρτόνι ώστε να μπορεί να στέκεται το όχημα χωρίς βοήθεια και μετά τοποθετήθηκε το gripper (με zip ties) και το motor driver (με βίδες). Ακόμη στο κάτω μέρος τοποθετήθηκε και ο αισθητήρας φωτός για το line following (με τη χρήση κομματιού στυλό και βίδες) όπως και ο δεύτερος αισθητήρας στο δεξί μέρος για την αναγνώριση γραμμών τερματισμού.



Εικόνες 3,4,5: Απεικόνιση τελικής κατασκευής στο κάτω μέρος του οχήματος

Ύστερα, τοποθετήθηκαν τέσσερα κομμάτια από στυλό ανάμεσα στους δύο 'ορόφους' ως διαχωριστικά. Στο επάνω μέρος του οχήματος τοποθετήθηκε ο μικροελεγκτής, το mini breadboard και η μπαταρία.



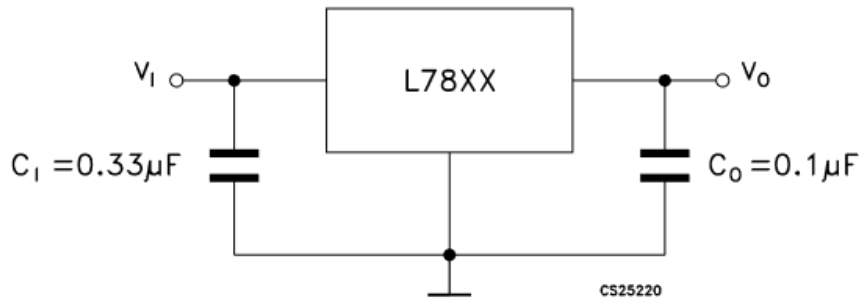
Εικόνα 6: Απεικόνιση τελικής κατασκευής στο επάνω μέρος του οχήματος

Το τελευταίο βήμα που έμεινε ήταν η διασύνδεση όλων των καλωδίων και εξαρτημάτων με την κατάλληλη συνδεσμολογία η οποία θα παρουσιαστεί στην επόμενη ενότητα (βλ. 2.3)

2.3 Συνδεσμολογία και Ηλεκτρονικά Κυκλώματα

Παρακάτω ακολουθεί περιγραφή της συνδεσμολογίας των εξαρτημάτων μεταξύ τους.

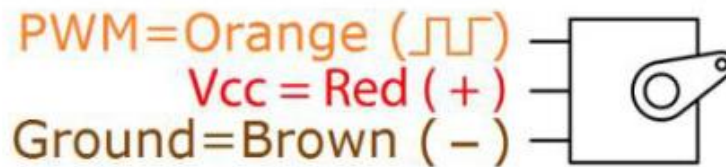
- **Motor Driver:** Στο motor driver συνδέθηκε η μπαταρία στις εισόδους του, οι DC κινητήρες στις αντίστοιχες θύρες του και τα 4 data pins για τον έλεγχό τους στον μικροελεγκτή.
- **Mini breadboard:** Στο breadboard τοποθετήθηκαν τα δύο buttons όπου το ένα τους άκρο συνδέθηκε στο GND και το άλλο σε pin εισόδου στον μικροελεγκτή. Ακόμη, συνδεσμολογήθηκαν οι δύο Voltage Regulators με βάση το παρακάτω σχεδιάγραμμα:



Εικόνα 7: Κύκλωμα των Voltage Regulators

Το V_I συνδέθηκε απευθείας στην μπαταρία και το V_O δίνει την σταθεροποιημένη τάση στα 5V και στα 3.3V αντίστοιχα. Για τους πυκνωτές C_I και C_O χρησιμοποιήθηκαν 10μF.

- Servo Motors: Τα Servo motors συνδέθηκαν με την τάση τροφοδοσίας στα 5V, το GND και το PWM signal pin τους με τον μικροελεγκτή.



Εικόνα 8: Pin layout του SG90 servo motor

- Line Follow Sensors: Οι αισθητήρες αυτοί συνδέθηκαν σε τροφοδοσία των 3.3V και τα data pins τους στον μικροελεγκτή.
- MSP430G2553: Ο μικροελεγκτής τροφοδοτείται από τα 3.3V και όλα τα υπόλοιπα εξαρτήματα συνδέονται πάνω του ώστε να μπορεί να τα ελέγχει.

Μέρος 3^ο

3.1 Προγραμματισμός Τμημάτων

Ο προγραμματισμός έγινε στο περιβάλλον Energia IDE (παρόμοιο με το Arduino IDE). Η διαδικασία προγραμματισμού ήταν η εξής: αρχικά προγραμματίστηκαν συναρτήσεις οι οποίες έλεγχαν κάθε τμήμα του ρομπότ ξεχωριστά (gripper, κίνηση, Line following κλπ.) και με την χρήση αυτών των συναρτήσεων δημιουργήθηκε το κυρίως πρόγραμμα.

3.1.1 Προγραμματισμός Gripper

Για το gripper χρησιμοποιήθηκε η βιβλιοθήκη Servo.h η οποία διευκολύνει τον έλεγχο των servo motors. Οπότε αφού συμπεριλήφθηκε στο πρόγραμμα, υπολογίστηκαν με την μέθοδο trial and error οι παρακάτω τιμές:

```
int SerO = 15;
int SerC = 55;
int SerCC = 60;
int SerU = 125;
int SerD = 100;
```

Εικόνα 9: Σταθερές για την κίνηση των Servo motor

Οι τιμές αυτές δίνονται σε μοίρες και ελέγχουν την θέση που θα πρέπει να βρίσκεται το κάθε servo motor ανάλογα με την δράση που θέλουμε να εκτελέσει το ρομπότ. Πιο αναλυτικά:

- SerO = Servo Open → Οι μοίρες στις οποίες η δαγκάνα του gripper είναι ανοικτή.
- SerC = Servo Closed → Οι μοίρες στις οποίες η δαγκάνα του gripper πρέπει να κλείσει ώστε να κρατάει ένα προκαθορισμένο αντικείμενο με αρκετή δύναμη ώστε να μην γλιστράει κατά την μεταφορά.
- SerCC = Servo Completely Closed → Οι μοίρες στις οποίες η δαγκάνα του gripper είναι κλειστή (χρησιμοποιήθηκε μόνο κατά την διάρκεια του προγραμματισμού και για λόγους debugging).
- SerU = Servo Up → Οι μοίρες στις οποίες όλο το gripper ανυψώνεται.
- SerD = Servo Down → Οι μοίρες στις οποίες το gripper επιστρέφει σε θέση παράλληλη με το έδαφος με σκοπό την συλλογή του αντικειμένου.

Με την χρήση αυτών των τιμών δημιουργήθηκαν 2 συναρτήσεις, η Pick() και η Place(), όπου η πρώτη εκτελεί κινήσεις για την συλλογή και ανύψωση του αντικειμένου, ενώ η δεύτερη την επιστροφή του αντικειμένου στο έδαφος. Ο κώδικάς τους φαίνεται παρακάτω:

```
//Function that Picks the can
void Pick()
{
    for(int ang = SerO; ang < SerC; ang++)
    {
        OCser.write(ang);
        delay(10);
    }
    for(int ang = SerD; ang < SerU; ang++)
    {
        UDser.write(ang);
        delay(10);
    }
}

//Function that Resets can
void Place()
{
    for(int ang = SerU; ang > SerD; ang--)
    {
        UDser.write(ang);
        delay(10);
    }
    for(int ang = SerC; ang > SerO; ang--)
    {
        OCser.write(ang);
        delay(10);
    }
}
```

Εικόνα 10: Κώδικας συναρτήσεων Pick(); (αριστερά) και Place(); (δεξιά)

3.1.2 Προγραμματισμός Κίνησης

Για την ολοκλήρωση του ρεπερτορίου κινήσεων που θα χρειαστεί να εκτελέσει το όχημα, ήταν απαραίτητη η δημιουργία άλλων δύο συναρτήσεων, οι οποίες χρησιμοποιώντας τους DC κινητήρες θα κάνουν το όχημα να οπισθοχωρεί και να περιστρέφεται κατά 180° αντίστοιχα. Οι δύο συναρτήσεις είναι η Bck, η οποία δέχεται σαν είσοδο τον χρόνο σε ms για τον οποίο θα πρέπει να οπισθοχωρεί το όχημα, και η Flip η οποία χρησιμοποιώντας τον αισθητήρα line following περιστρέφει το όχημα γύρω από τον εαυτό του μέχρις ότου ο αισθητήρας να ανιχνεύσει την μαύρη γραμμή και να σταματήσει να κινείται.

```
//Function that moves robot in reverse for specified ms
void Bck(int ms)
{
    //Reverses movement direction
    digitalWrite(LMOTORD,HIGH);
    digitalWrite(RMOTORD,HIGH);
    analogWrite(LMOTOR, 60);
    analogWrite(RMOTOR, 60);
    delay(ms);

    //Resets and stops motors
    digitalWrite(LMOTORD,LOW);
    digitalWrite(RMOTORD,LOW);
    analogWrite(LMOTOR, 0);
    analogWrite(RMOTOR, 0);
}

void Flip()
{
    digitalWrite(LMOTORD,LOW);
    digitalWrite(RMOTORD,HIGH);
    analogWrite(LMOTOR, 200);
    analogWrite(RMOTOR, 255-200);
    delay(100);
    while(analogRead(LFS) < 600);
    digitalWrite(RMOTORD,LOW);
    analogWrite(LMOTOR, 0);
    analogWrite(RMOTOR, 0);
}
```

Εικόνα 11: Κώδικας συναρτήσεων Bck (αριστερά) και Flip(); (δεξιά)

3.1.3 Προγραμματισμός PID ελεγκτή

Η πιο βασική από όλες τις συναρτήσεις είναι αυτή του PID Line Following. Ο αλγόριθμος έχει ως εξής: Όσο ο αισθητήρας δεν ανιχνεύει γραμμή τερματισμού, υπολογίζεται το σφάλμα (error) από την διαφορά τιμή στόχου (600) και της μέτρησης του αισθητήρα line following. Μετά υπολογίζονται τα παρακάτω:

- $P \rightarrow$ Το proportional μέρος του ελεγκτή, το οποίο ισούται με $P = \text{error}$. Άρα, όσο μεγαλύτερο σφάλμα έχουμε, τόσο μεγαλύτερη διόρθωση θα γίνει για να μηδενιστεί το σφάλμα.
- $I \rightarrow$ Το integral μέρος του ελεγκτή, το οποίο ισούται με $I = I + \text{error}$. Με λίγα λόγια το I κρατάει το άθροισμα των σφαλμάτων και με βάση αυτό κάνει την διόρθωση.
- $D \rightarrow$ Το derivative μέρος του ελεγκτή, το οποίο ισούται με $D = \text{error} - \text{lasterror}$. Το lasterror είναι το σφάλμα που είχε προκύψει στην προηγούμενη επανάληψη του αλγορίθμου. Πρακτικά το D μας δίνει την διαφορά σφάλματος που υπάρχει. Αν το σφάλμα μεγαλώσει σε σχέση με πριν, τότε η διόρθωση πρέπει να γίνει με μεγαλύτερη ένταση. Αντίθετα αν το σφάλμα γίνει μικρότερο σημαίνει ότι το όχημα κινείται προς την σωστή διεύθυνση και άρα πρέπει να κάνει ανάλογη διόρθωση ώστε να μην κάνει overshoot τον στόχο.

Αφού υπολογιστούν τα παραπάνω, εκτελείται η εντολή: $\text{sped} = P \cdot K_p + I \cdot K_i + D \cdot K_d$, η οποία προσθέτει τους όρους του ελεγκτή πολλαπλασιάζοντάς τους με τις αντίστοιχες σταθερές (βλ. 3.2 Ρύθμιση Παραμέτρων), δημιουργώντας έτσι την ταχύτητα με την οποία θα πρέπει τελικά να περισταφούν οι κινητήρες ώστε να καταλήξει το όχημα να ακολουθεί την γραμμή.

```
void LineFollow()  
{  
    while(digitalRead(DET) == 0)  
    {  
        error = 600 - analogRead(LFS);  
        P = error;  
        I = I + error;  
        D = error - lasterr;  
        lasterr = error;  
        sped = P*Kp + I*Ki + D*Kd;  
        analogWrite(LMOTOR, bsped - sped);  
        analogWrite(RMOTOR, bsped + sped);  
    }  
    analogWrite(LMOTOR, 0);  
    analogWrite(RMOTOR, 0);  
}
```

Εικόνα 12: Κώδικας συνάρτησης LineFollow()

3.2 Ρύθμιση Παραμέτρων

Για την σωστή λειτουργία του PID ελεγκτή έπρεπε να βρεθούν οι κατάλληλες τιμές K_p , K_i και K_d . Για να επιτευχθεί αυτό ακολουθήθηκε μία διαδικασία trial an error. Αρχικά δοκιμάστηκαν μεταβολές μόνο της μεταβλητής K_p κρατώντας τις άλλες δύο στην τιμή μηδέν (άρα δεν επηρέαζαν το σύστημα). Αφού βρέθηκε μία ικανοποιητική τιμή για το K_p , μετά έγινε διερεύνηση για την τιμή της K_i και τέλος της K_d . Το τελικό αποτέλεσμα που προέκυψε ήταν ένα σύστημα το οποίο είχε καλή ακρίβεια και ταυτόχρονα τις μικρότερες δυνατές ταλαντώσεις.

3.3 Κυρίως Πρόγραμμα και Αλγόριθμος

Το κυρίως πρόγραμμα, με την χρήση των προηγούμενων συναρτήσεων, έχει την εξής μορφή: Με την χρήση ενός if statement, το ρομπότ μένει ακίνητο μέχρις ότου ο χρήστης να πατήσει ένα από τα δύο buttons που βρίσκονται πάνω στο όχημα. Το πρώτο κουμπί εκτελεί το μέρος του κώδικα κατά το οποίο το όχημα ξεκινάει να ακολουθεί την γραμμή και όταν φτάσει σε γραμμή τερματισμού, κάνει pick το αντικείμενο που βρίσκεται σε αυτή τη θέση και το επιστρέφει πίσω στην αφετηρία. Το δεύτερο κουμπί αντίθετα, υποθέτοντας ότι το όχημα συγκρατεί ήδη επάνω του το αντικείμενο, έχει ως σκοπό να πάει να το αφήσει στο σημείο εκείνο από όπου το πήρε (βλ. Εικόνα 14).

```
//Checks For Button Presses    else if(digitalRead(BUT2) == HIGH)
if(digitalRead(BUT1) == HIGH) {
{
    delay(5000);                Pick();
    LineFollow();                LineFollow();
    Pick();                      Place();
    Flip();                      Bck(150);
    delay(500);                  Flip();
    LineFollow();                delay(500);
    Flip();                      LineFollow();
    Place();                      Flip();
}                                }
}
```

Εικόνα 13: Κύριος Κώδικας, περίπτωση 1(αριστερά) και περίπτωση 2(δεξιά)



Εικόνα 14: Εκτέλεση περίπτωσης 1(αριστερά) και περίπτωσης 2(δεξιά)

Ολόκληρος ο κώδικας μπορεί να βρεθεί στο Github (βλ. 5.1 Πηγές).

Μέρος 4^ο

4.1 Περιγραφή Τελικού Αποτελέσματος

Το τελικό αποτέλεσμα που προέκυψε, ήταν και αυτό το οποίο είχε τεθεί ως σκοπός στην αρχή της εκπόνησης της εργασίας. Δηλαδή, η κατασκευή του ρομπότ να έχει το επιθυμητό μέγεθος, στιβαρότητα και αισθητική. Ακόμη ο PID ελεγκτής να έχει τις κατάλληλες για την εφαρμογή σταθερές (K_p , K_i , K_d). Αξίζει να σημειωθεί, ότι με την χρήση των συναρτήσεων που είχαν ετοιμαστεί, η δομή του κύριου κώδικα ήταν πολύ απλή. Πράγμα που σημαίνει ότι σε περίπτωση που κάποιος επιθυμεί να αλλάξει τον χώρο επάνω στον οποίο κινείται το όχημα, ο προγραμματισμός του θα είναι μία εύκολη υπόθεση μιας και οι πιο σημαντικές συναρτήσεις υπάρχουν ήδη. Παρόλα αυτά όμως, παρουσιάστηκαν κάποιες δυσκολίες οι οποίες θα αναλυθούν στην επόμενη ενότητα, όπως και βελτιώσεις που θα μπορούσαν να γίνουν στο μέλλον (βλ. ενότητα 4.3)

4.2 Προβλήματα και Αντιμετώπιση

Όπως και ήταν αναμενόμενο, κατά την διάρκεια της κατασκευής και του προγραμματισμού προέκυψαν διαφόρων ειδών προβλήματα τα περισσότερα εκ των οποίων αντιμετωπίστηκαν, ενώ μερικά άλλα ήταν αιτία αλλαγής του αρχικού πλάνου εξαιτίας της δυσκολίας τους ως προς την επίλυση.

- Μη στιβαρή κατασκευή. Λόγω των υλικών που χρησιμοποιήθηκαν καθώς και οι μέθοδοι για την συγκράτηση των εξαρτημάτων, το όχημα είχε μεγάλη ενδοτικότητα. Λόγω της έλλειψης εμπειρίας στο κατασκευαστικό κομμάτι, δεν μπορούσαν να γίνουν και πολλές αλλαγές για την αντιμετώπιση του συγκεκριμένου προβλήματος.
- Βάρος και τοποθέτηση της μπαταρίας. Το βάρος της συγκεκριμένης μπαταρίας που χρησιμοποιήθηκε ήταν τόσο μεγάλο σε σχέση με το υπόλοιπο όχημα όπου η τοποθέτησή της ήταν ζωτικού χαρακτήρα. Αρχικά είχε τοποθετηθεί επάνω από τον αριστερό τροχό με αποτέλεσμα, όταν το όχημα προσπαθούσε να κινηθεί ευθύγραμμα, ο δεξιά τροχός να ξεκινούσε πιο γρήγορα από τον αριστερό (μιας και δεν είχε μεγάλο βάρος από πάνω του). Το αποτέλεσμα ήταν, το όχημα να προχωρούσε σε τόξο, ανεξέλεγκτο. Αρχικά έγινε προσπάθεια επίλυσης του συγκεκριμένου προβλήματος με κώδικα. Δηλαδή είχε προστεθεί μία παραπάνω εντολή πριν από την εντολή εκκίνησης του δεξιού τροχού η οποία την καθυστερεί έτσι ώστε να προλάβει να ξεκινήσει ο αριστερός και να αποκτήσει την κατάλληλη φόρα. Αυτή η λύση βοήθησε μόνο για πάρα πολύ μικρές αποστάσεις, ενώ στις μεγάλες επικρατούσε το ίδιο πρόβλημα. Έτσι έγινε μετακίνηση της μπαταρίας τη μέση του ρομπότ και το πρόβλημα έπαψε να υπάρχει.
- Φωτισμός δωματίου και ελεγχόμενο περιβάλλον. Οι αισθητήρες που χρησιμοποιήθηκαν για την πλοήγηση, όντας αισθητήρες φωτός, ήταν υπερευαίσθητοι στο φως του ηλίου με αποτέλεσμα, το περιβάλλον στο οποίο γινόταν ο προγραμματισμός να έπρεπε να είναι τεχνητά φωτισμένο.

4.3 Προτάσεις Βελτιστοποίησης

Συμπεριλαμβανομένων των προβλημάτων που έχουν προκύψει, μελλοντικές αλλαγές τόσο και στον σχεδιασμό όσο και στον προγραμματισμό μπορούν να γίνουν.

- Περισσότερες Συναρτήσεις. Με τον προγραμματισμό περισσότερων βασικών συναρτήσεων, θα είναι δυνατή η πλοήγηση σε πιο δύσκολα περιβάλλοντα (με περισσότερες από μία γραμμές για ακολουθία)
- Αύξηση μεγέθους και στιβαρότητας. Με την χρήση διαφορετικών υλικών και καλύτερων τρόπων σύνδεσης, όπως και μία μικρή αύξηση στο μέγεθος, θα βοηθούσαν αρκετά στην στιβαρότητα του οχήματος. Για παράδειγμα, η 3D εκτύπωση του σώματος του ρομπότ και άλλων συνδετικών στοιχείων θα ήταν μία πρώτη προσέγγιση.

- Προσθήκη περισσότερων αισθητήρων. Μιας και το ρομπότ διαθέτει έναν μόνο αισθητήρα ανίχνευσης γραμμής τερματισμού από την δεξιά πλευρά, θα μπορούσε να τοποθετηθεί και ένας από τα αριστερά ώστε να γίνεται δυνατή η ανίχνευση δύο διαδρομών ταυτόχρονα. Ακόμη ένας αισθητήρας απόστασης (Ultrasonic sensor) θα έκανε δυνατή την αποφυγή εμποδίων που πιθανόν να υπάρχουν στο περιβάλλον το οποίο βρίσκεται.
- Αύξηση της μηχανικής δύναμης με χρήση ισχυρότερων κινητήρων. Με την χρήση μεγαλύτερης ισχύος κινητήρων θα υπάρξει σημαντική αύξηση στο φορτίο το οποίο θα μπορεί να κουβαλήσει το όχημα. Ακόμη με πιο ισχυρά servo motors, η ανύψωση αντικειμένων με μεγαλύτερο βάρος θα είναι πολύ ευκολότερη.
- Χρήση καλύτερου είδους μπαταρίας. Με την χρήση μπαταρίας νεότερης τεχνολογίας (Li-ion) μειώνεται κατά πολύ το βάρος όπως και αυξάνεται η αυτονομία, μιας και η πυκνότητα της χωρητικότητας ως προς το βάρος ολοένα και μεγαλώνει.

4.4 Συμπεράσματα και Παρατηρήσεις

Μετά το πέρας της εργασίας αυτής, αποκτήθηκε αρκετή εμπειρία στα ρομποτικά συστήματα. Πιο συγκεκριμένα στο κατασκευαστικό κομμάτι, στην 3D εκτύπωση, στον προγραμματισμό και στα ηλεκτρονικά κυκλώματα. Ακόμη έγινε εξοικείωση με αισθητήρες, κινητήρες και motor drivers καθώς και άλλες παραλλαγές τους που κυκλοφορούν στην αγορά. Όπως αναφέρθηκε και στην ενότητα 4.1, οι στόχοι οι οποίοι είχαν τεθεί στην αρχή πραγματοποιήθηκαν παρά τα προβλήματα που προέκυψαν, και έγινε και μελέτη για μελλοντικές εξελίξεις οι οποίες προέκυψαν μέσα από τις δυσκολίες που είχε το συγκεκριμένο project.

Μέρος 5^ο

5.1 Πηγές

- <https://create.arduino.cc/projecthub/electronicprojects/dc-motor-speed-control-using-arduino-0e5b2b>
- <https://github.com/Gregory6148/Robotics-Class-Project-2022.git>
- <https://www.thingiverse.com/thing:3648782>
- <http://adam-meyer.com/arduino/QRE1113>
- http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf
- <https://datasheet.octopart.com/L7805CV-STMicroelectronics-datasheet-7264666.pdf>
- <https://create.arduino.cc/projecthub/anova9347/line-follower-robot-with-pid-controller-cdedbd>

5.2 Βιβλιογραφία

- [1] https://en.wikibooks.org/wiki/Robotics/Types_of_Robots/Wheeled
- [2] <https://en.wikibooks.org/wiki/Robotics/Components/Grippers>
- [3] https://en.wikipedia.org/wiki/PID_controller