

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΡΟΜΠΟΤΙΚΗ, STEAM ΚΑΙ ΝΕΕΣ ΤΕΧΝΟΛΟΓΙΕΣ
ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ»

Διπλωματική Εργασία

ΚΑΤΑΣΚΕΥΗ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΚΙΝΗΤΟΥ ΡΟΜΠΟΤ
ΑΚΟΛΟΥΘΙΑΣ ΓΡΑΜΜΗΣ ΜΕ ARDUINO

ΕΙΡΗΝΗ ΖΑΡΟΓΙΑΝΝΟΥ Α.Μ. 2022027

Επιβλέπων Καθηγητής
Δρ. Χρήστος Υφούλης, Αναπληρωτής Καθηγητής

Υποβλήθηκε ως απαιτούμενο για την απόκτηση του μεταπτυχιακού διπλώματος
ειδίκευσης Ρομποτική, STEAM και Νέες Τεχνολογίες στην Εκπαίδευση
Θεσσαλονίκη, Ιανουάριος 2024



Η παρούσα Διπλωματική Εργασία καλύπτεται στο σύνολό της νομικά από δημόσια
άδεια πνευματικών δικαιωμάτων CreativeCommons:

Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Παρόμοια Διανομή



Μπορείτε να:

- Μοιραστείτε: αντιγράψετε και αναδιανέμετε το παρόν υλικό με κάθε μέσο και τρόπο
- Προσαρμόστε: αναμείξτε, τροποποιήστε και δημιουργήστε πάνω στο παρόν υλικό

Υπό τους ακόλουθους όρους:

- Αναφορά Δημιουργού: Θα πρέπει να καταχωρίσετε αναφορά στο δημιουργό, με σύνδεσμο της άδειας, και με αναφορά αν έχουν γίνει αλλαγές. Μπορείτε να το κάνετε αυτό με οποιονδήποτε εύλογο τρόπο, αλλά όχι με τρόπο που να υπονοεί ότι ο δημιουργός αποδέχεται το έργο σας ή τη χρήση που εσείς κάνετε.
- Μη Εμπορική Χρήση: Δε μπορείτε να χρησιμοποιήσετε το υλικό για εμπορικούς σκοπούς.
- Παρόμοια Διανομή: Αν αναμείξετε, τροποποιήσετε, ή δημιουργήσετε πάνω στο παρόν υλικό, πρέπει να διανείμετε τις δικές σας συνεισφορές υπό την ίδια άδεια CreativeCommonsόπως και το πρωτότυπο.

Αναλυτικές πληροφορίες νομικού κώδικα στην ηλεκτρονική διεύθυνση:

<https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

Υπεύθυνη Δήλωση

Η παρούσα Διπλωματική Εργασία και τα συμπεράσματά της, σε οποιαδήποτε μορφή, αποτελούν συνιδιοκτησία του Τμήματος Μηχανικών Παραγωγής και Διοίκησης του Διεθνούς Πανεπιστημίου Ελλάδος και του φοιτητή. Οι προαναφερόμενοι διατηρούν το δικαίωμα ανεξάρτητης χρήσης και αναπαραγωγής (τμηματικά ή συνολικά) για διδακτικούς και ερευνητικούς σκοπούς. Σε κάθε περίπτωση πρέπει να αναφέρεται ο τίτλος, ο συγγραφέας, ο επιβλέπων και το τμήμα του ΔιΠαΕ.

Η έγκριση της παρούσας Διπλωματικής Εργασίας από το Τμήμα Μηχανικών Παραγωγής και Διοίκησης δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Η υπογεγραμμένη δηλώνω υπεύθυνα ότι η παρούσα Διπλωματική Εργασία είναι εξ' ολοκλήρου δικό μου έργο και συγγράφηκε ειδικά για τις απαιτήσεις του μεταπτυχιακού προγράμματος σπουδών του Τμήματος Μηχανικών Παραγωγής και Διοίκησης με τίτλο «Ρομποτική, STEAM και νέες Τεχνολογίες στην Εκπαίδευση».

Δηλώνω υπεύθυνα ότι κατά τη συγγραφή ακολούθησα την πρόπουσα ακαδημαϊκή δεοντολογία αποφυγής λογοκλοπής και έχω αποφύγει οποιαδήποτε ενέργεια που συνιστά παράπτωμα λογοκλοπής.

Ζαρόγιαννου Ειρήνη



31/1/2024

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά, από τα βάθη της καρδιάς μου, τον κύριο επιβλέποντα της διπλωματικής μου εργασίας, αναπληρωτή καθηγητή Υφούλη Χρήστο, για την ουσιαστική και εμπειριστατωμένη καθοδήγηση που μου προσέφερε και για τις συμβουλές που μου παρείχε καθ' όλη τη διάρκεια εκπόνησης αυτής.

Επίσης, οφείλω ένα μεγάλο ευχαριστώ στην οικογένειά μου για την αστείρευτη ηθική και ψυχολογική υποστήριξη που μου προσέφερε σε όλο το διάστημα σπουδών μου.

Περίληψη

Η παρούσα διπλωματική εργασία πραγματεύεται την κατασκευή, τη συναρμολόγηση, τον προγραμματισμό και τον έλεγχο ενός κινητού ρομπότ ακολουθίας γραμμής με τη βοήθεια του μικροελεγκτή Arduino. Ο επιδιωκόμενος στόχος είναι η δημιουργία ρομπότ χαμηλού κόστους που να κινείται ικανοποιητικά πάνω σε οποιαδήποτε γραμμή και σε αρκετά μεγάλες ταχύτητες. Προκειμένου το ρομπότ να κινείται με ακρίβεια πάνω στη γραμμή μειώνοντας το χρόνο ολοκλήρωσης μίας πίστας, σχεδιάστηκαν συστήματα αυτόματου ελέγχου διαμόρφωσης της ταχύτητάς του με τη χρήση κατάλληλων αισθητηρίων. Ο PID ελεγκτής στην περίπτωση αυτή φαίνεται να αποτελεί την πιο ολοκληρωμένη λύση στο προβλήματα αυτό.

Abstract

This thesis deals with the construction, assembly, programming and control of a mobile line following robot with the help of microcontroller Arduino. The intended goal is to create a low-cost robot that can move satisfactorily on any line and at sufficient high speeds. In order for the robot to move accurately on the line reducing the time to complete a track, automatic control systems were designed to modulate its speed using appropriate sensors. The PID controller in this case seems to be the most complete solution to this problem.

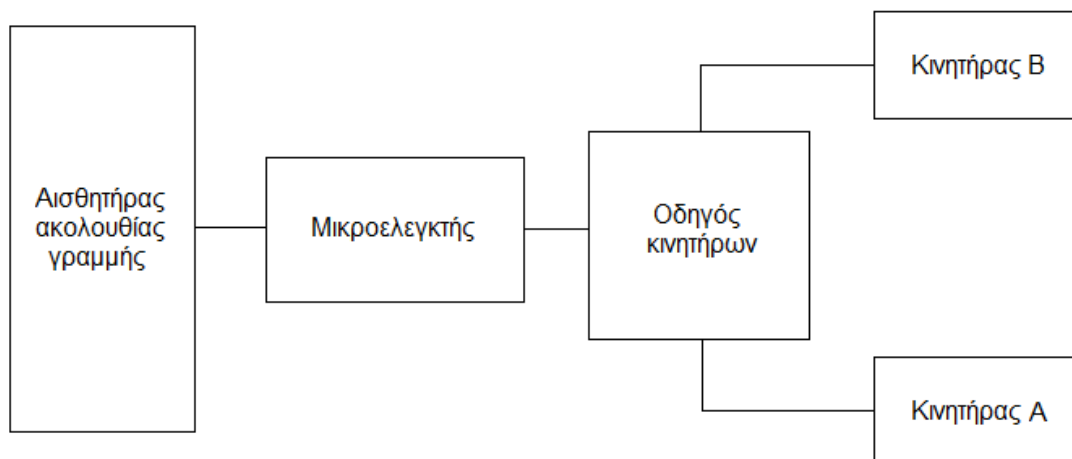
Περιεχόμενα

Κεφάλαιο 1. Κατασκευή κινητού ρομπότ ακολουθίας γραμμής	1
1.1 Περιγραφή κυκλωμάτων και εξαρτημάτων	2
1.1.1 Μικροελεγκτής Arduino Uno R3	2
1.1.2 DC Gearbox κινητήρες	3
1.1.3 Οδηγός κινητήρων L298N.....	4
1.1.4 Μονάδα αισθητήρων ανάκλασης QTR-8RC	5
1.1.5 Διακόπτες ελέγχου	9
1.1.6 Ενδεικτική λυχνία λειτουργίας LED	10
1.1.7 Σασί.....	10
1.1.8 Τροφοδοσία	11
1.1.9 Καλώδια σύνδεσης	12
1.1.10 Τροχοί	12
1.1.11 Ρόδα ελεύθερης περιστροφής	12
1.1.12 Πλακέτα δοκιμών.....	13
1.2 Συναρμολόγηση και συνδεσμολογία εξαρτημάτων και κυκλωμάτων.....	14
Κεφάλαιο 2. Προγραμματισμός κινητού ρομπότ ακολουθίας γραμμής.....	19
2.1. Bang-Bang Έλεγχος.....	19
2.2. Αναλογικός Ελεγκτής	21
2.3. Αναλογικός - Διαφορικός Ελεγκτής	24
2.4. Αναλογικός - Ολοκληρωτικός - Διαφορικός Ελεγκτής.....	26

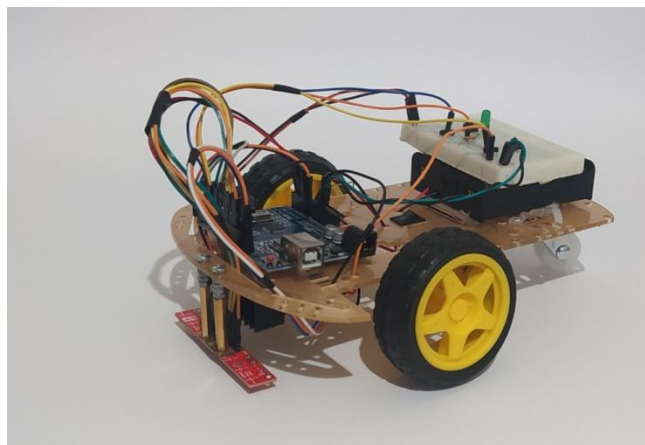
3. Συμπεράσματα - Πιθανές επεκτάσεις	29
Βιβλιογραφία - Πηγές	30
Παράρτημα Α - Κοστολόγηση υλικών	31
Παράρτημα Β: Κώδικες Arduino.....	32
Παράρτημα Γ - Πίστες δοκιμών	42

Κεφάλαιο 1. Κατασκευή κινητού ρομπότ ακολουθίας γραμμής

Το ρομπότ της παρούσας εργασίας είναι ένα απλό αυτόνομο καθοδηγούμενο ρομπότ που εντοπίζει και ακολουθεί μία μαύρη γραμμή που βρίσκεται πάνω σε άσπρη επίπεδη επιφάνεια. Στην κατασκευή θα χρησιμοποιηθεί ο πίνακας αισθητήρων ανάκλασης ή αισθητήρας γραμμής QTR-8RC λόγω της ακρίβειάς του. Η κίνηση του οχήματος βασίζεται σε δύο DC Gearbox κινητήρες οι οποίοι θα ελέγχονται ταυτόχρονα από τον οδηγό κινητήρων L298N. Ο μικροελεγκτής Arduino UNO R3 θα είναι ο εγκέφαλος του συστήματος του ρομπότ και ο υπεύθυνος επικοινωνίας μεταξύ των περιφερειακών συσκευών. Ο σκελετός του ρομπότ αποτελείται από μία βάση από οπές όπου θα συνδεθούν διάφορα εξαρτήματα και ηλεκτρονικά κυκλώματα. Παρακάτω απεικονίζονται το σύστημα λειτουργίας του ρομπότ (Εικόνα 1.1) και το μοντέλο κατασκευής που θα δημιουργηθεί στη συνέχεια (Εικόνα 1.2).



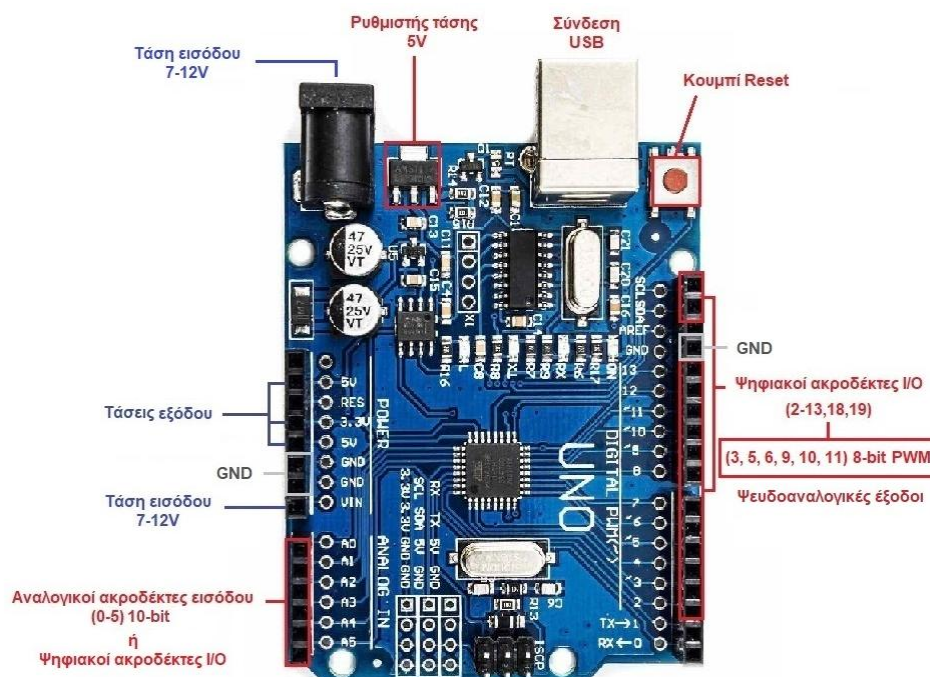
Εικόνα 1.1: Διάγραμμα συστήματος λειτουργίας κινητού ρομπότ



Εικόνα 1.2: Μοντέλο κατασκευής

1.1 Περιγραφή κυκλωμάτων και εξαρτημάτων

1.1.1 Μικροελεγκτής Arduino Uno R3



Εικόνα 1.3: Arduino Uno R3

Ο μικροελεγκτής Arduino Uno R3 είναι μία προγραμματιζόμενη ηλεκτρονική πλακέτα ανοιχτού κώδικα με ενσωματωμένο τον μικροελεγκτή ATmega328P και με τάση λειτουργίας 5V.

Στην κατασκευή για την τροφοδοσία του μικροελεγκτή αλλά και για τις συνδέσεις διαφόρων διατάξεων σε αυτόν θα χρειαστούν τα εξής σημεία πάνω στην πλακέτα:

- οι ψηφιακοί ακροδέκτες εισόδου/εξόδου (6 από αυτές με δυνατότητα PWM)
- οι αναλογικοί ακροδέκτες εισόδου οι οποίοι μπορούν να χρησιμοποιηθούν και ως ψηφιακοί ακροδέκτες εισόδου/εξόδου
- οι ακροδέκτες γείωσης
- ο ακροδέκτης παροχής τάσης 5V
- ο ακροδέκτης τάσης εισόδου 7-12V
- η θύρα USB
- το κουμπί Reset

Ο προγραμματισμός του μικροελεγκτή Arduino θα γίνεται στη γλώσσα Wiring C (μία παραλλαγή της C και C++) στο ολοκληρωμένο περιβάλλον ανάπτυξης Arduino IDE.

1.1.2 DC Gearbox κινητήρες



Εικόνα 1.4: DC κινητήρες

Οι DC κινητήρες είναι σημαντικοί ενεργοποιητές στο χώρο της ρομποτικής για την δημιουργία οχημάτων, συστημάτων κινούμενων αξόνων και βραχιόνων, τροφοδοτούνται με συνεχή τάση και η φορά περιστροφής τους αλλάζει αλλάζοντας την πολικότητα της τάσης με την οποία τροφοδοτούνται. Οι DC κινητήρες έχουν ένα συγκεκριμένο εύρος τάσης λειτουργίας και η ταχύτητα περιστροφής τους μπορεί να μεταβληθεί με τη χρήση της τεχνικής ελέγχου διαμόρφωσης πλάτους παλμού (PWM).

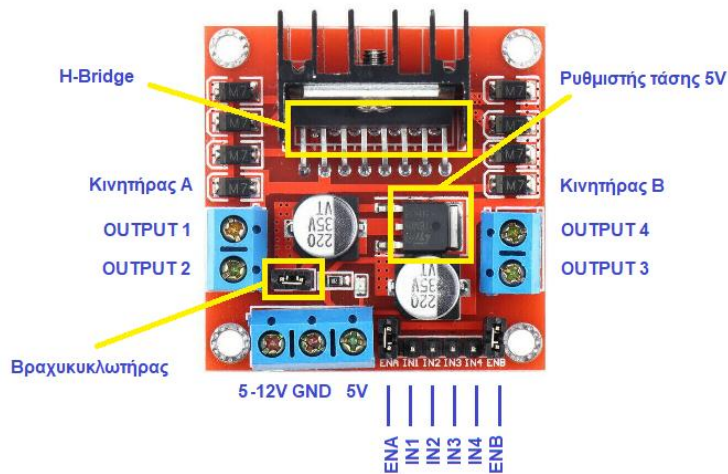


Εικόνες 1.5: DC Gearbox κινητήρες

Στην κατασκευή θα χρησιμοποιηθούν δύο DC Gearbox κινητήρες οι οποίοι στο εσωτερικό τους μέρος περιέχουν από ένα DC κινητήρα που είναι συνδεδεμένος με ένα σύστημα γραναζιών για την μείωση των στροφών τους και για την αύξηση της ροπής τους μέσα από μία σχέση μετάδοσης 1:48. Η έξοδος του συστήματός τους είναι ένας κάθετος πλαστικός άξονας στον οποίο μπορούν να εφαρμοστούν οι κινητήριοι τροχοί της ρομποτικής κατασκευής. Η τάση λειτουργίας τους είναι 3-6V και η περιστροφική τους ταχύτητα χωρίς φορτίο είναι περίπου 90-120 RPM.

Για την εύρυθμη λειτουργία τους αλλά και για την διατήρηση της αποδοτικότητάς τους θα χρειαστούν εξωτερική τροφοδοσία. Οι DC Gearbox κινητήρες θα συναποτελούν το τιμόνι του κινητού ρομπότ (διαφορική κίνηση τροχών με ανεξάρτητους κινητήρες) και ο έλεγχός τους θα γίνεται μέσω του οδηγού κινητήρων L298N.

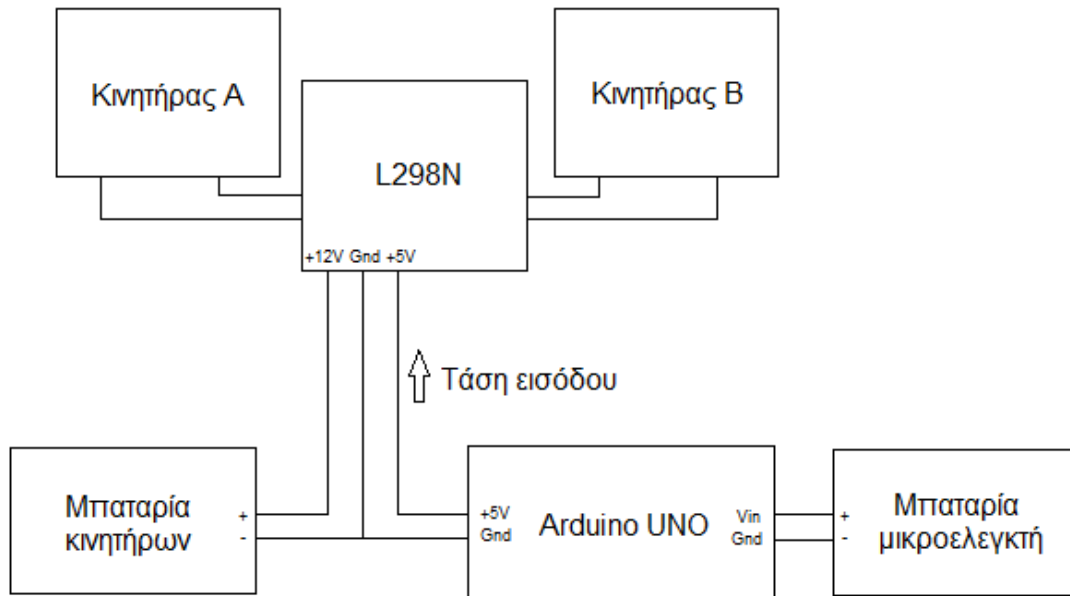
1.1.3 Οδηγός κινητήρων L298N



Εικόνα 1.6: Οδηγός κινητήρων L298N

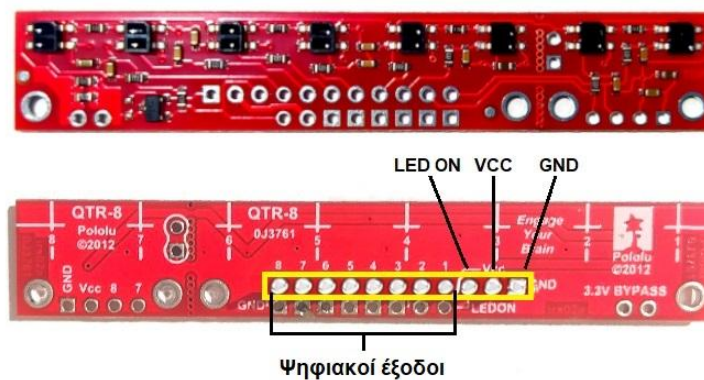
Ο οδηγός κινητήρων L298N είναι μία μονάδα οδήγησης για κινητήρες συνεχούς ρεύματος ή ενός βηματικού κινητήρα που περιέχει το ολοκληρωμένο κύκλωμα H-Bridge. Στη μονάδα του οδηγού υπάρχουν δύο μπλοκ με βιδωτούς ακροδέκτες εκατέρωθεν για τους κινητήρες A και B και ένα μπλοκ βιδωτών ακροδεκτών για την τάση τροφοδοσίας των κινητήρων 5-12V, τη γείωση (κοινό σημείο αναφοράς) και την τάση εισόδου/εξόδου 5V για την τροφοδοσία του λογικού κυκλώματος L298N ή του μικροελεγκτή Arduino αντίστοιχα. Οι ακίδες ελέγχου ταχύτητας ENA και ENB ενεργοποιούν και απενεργοποιούν τους κινητήρες A και B και ελέγχουν την ταχύτητάς τους δίνοντας μία τιμή ψευδοναλογικής εξόδου 0-255 στον καθένα. Οι ακίδες IN1 και IN2 ελέγχουν τη φορά περιστροφής του κινητήρα A και οι ακίδες IN3 και IN4 ελέγχουν τη φορά περιστροφής του κινητήρα B. Η κίνηση του ρομπότ θα ελέγχεται ορίζοντας ταυτόχρονα την φορά περιστροφής των κινητήρων A και B δίνοντας λογικές τιμές τάσης HIGH ή LOW στις ακίδες εισόδου IN1, IN2, IN3 και IN4.

Πάνω στη μονάδα υπάρχει επίσης ένας ρυθμιστής τάσης 5V ο επονομαζόμενος 78M05 ο οποίος ενεργοποιείται ή απενεργοποιείται βάζοντας ή βγάζοντας τον βραχυκυκλωτήρα αντίστοιχα. Όταν ο ρυθμιστής τάσης είναι απενεργοποιημένος, ο ακροδέκτης τροφοδοσίας 5V λειτουργεί ως είσοδος διαχωρίζοντας τη τροφοδοσία του λογικού κυκλώματος L298N από τους κινητήρες (Εικόνα 1.7). Η ύπαρξη ξεχωριστής τροφοδοσίας για του κινητήρες είναι σημαντική, γιατί πέρα του ότι καταναλώνουν πολύ ενέργεια αναμένεται από τον οδηγό L298N πτώση τάσης 2V.



Εικόνα 1.7: Διάγραμμα τροφοδοσίας κινητού ρομπότ

1.1.4 Μονάδα αισθητήρων ανάκλασης QTR-8RC

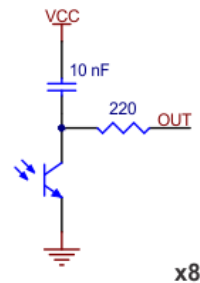


Εικόνα 1.8: Μονάδα αισθητήρων ανάκλασης QTR-8RC

Η μονάδα αισθητήρων ανάκλασης QTR-8RC είναι ένας ψηφιακός αισθητήρας γραμμής, εγγύτητας ή ανάκλασης γενικής χρήσης που μετρά σε ένα δεδομένο χρονικό όριο (timeout) την ένταση του ανακλώμενου φωτός.

Ο αισθητήρας γραμμής έχει 8 ομοιόμορφα διατεταγμένα ζεύγη πομπού (IR LED) και δέκτη υπερύθρων (φωτοτρανζίστορ) που απέχουν μεταξύ τους κατά 0,95cm και η τάση λειτουργίας του είναι 3,3-5V. Για κάθε ζεύγος πομπού/δέκτη η μονάδα φέρει οκτώ ψηφιακές εξόδους κάνοντάς την κατάλληλη για χρήση με τους διαθέσιμους ψηφιακούς ακροδέκτες εισόδου του μικροελεγκτή Arduino Uno R3. Κάθε δέκτης

χρησιμοποιεί ένα κύκλωμα εκκένωσης πυκνωτή που επιτρέπει σε μια ψηφιακή γραμμή εισόδου του μικροελεγκτή να λάβει μια αναλογική μέτρηση ανάκλασης μετρώντας πόσο χρόνο χρειάζεται η τάση εξόδου του για να αποσυντεθεί.



Εικόνα 1.9: Κύκλωμα εκκένωσης πυκνωτή δέκτη υπερύθρων

Ο μικρότερος χρόνος αποσύνθεσης είναι ένδειξη μεγαλύτερης αντανάκλασης και ο μεγαλύτερος χρόνος αποσύνθεσης είναι ένδειξη μικρότερης αντανάκλασης (ανεπεξέργαστα δεδομένα). Με αυτήν την ικανότητα, οι αισθητήρες μπορούν να διαβαστούν όλοι παράλληλα και ο καθένας απαιτεί τον χρονισμό ενός παλμού που μπορεί να διαρκέσει έως και 3ms.

Στην κατασκευή ο αισθητήρας γραμμής θα βρίσκεται σε απόσταση 0,9cm από την επιφάνεια εξασφαλίζοντας ένα αρκετά καλό ρυθμό δειγματοληψίας συχνοτήτων και στις πίστες δοκιμών θα ανιχνεύει μία μαύρη γραμμή πλάτους 1,8cm πάνω σε μία άσπρη επίπεδη επιφάνεια.

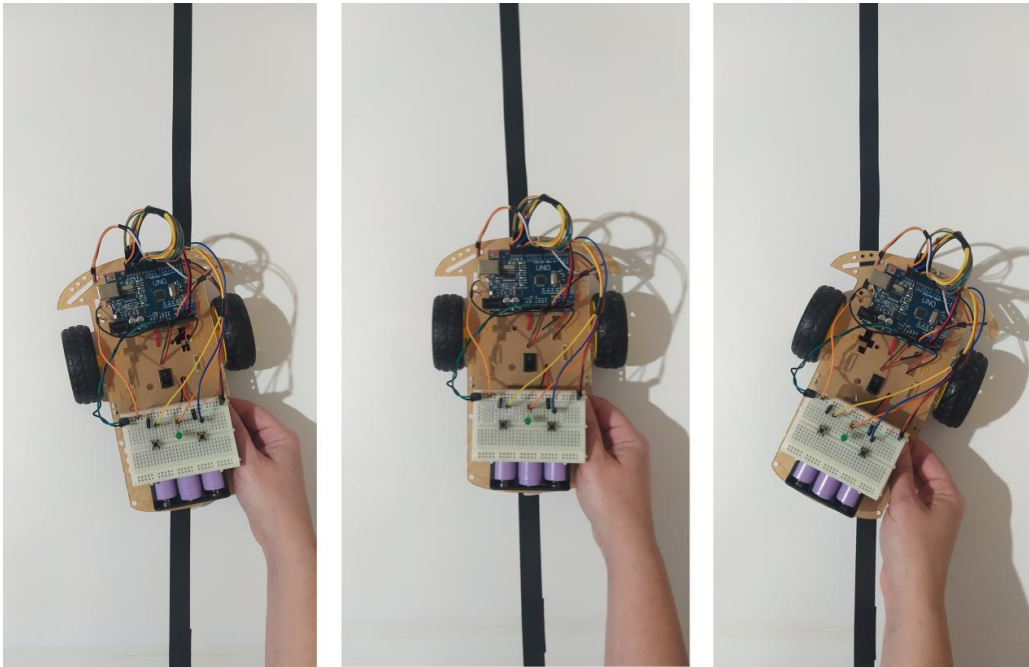
Ο αισθητήρας γραμμής QTR-8RC είναι η μονάδα εισόδου του συστήματος του ρομπότ και για τη λήψη των μετρήσεων των επιμέρους αισθητήρων του θα χρειαστεί να εγκατασταθεί στο περιβάλλον ανάπτυξης Arduino IDE η βιβλιοθήκη QTRSensors.

Η βιβλιοθήκη QTRSensors παρέχει δύο βασικές μεθόδους, την `calibrate()` και την `int readLineBlack(int sensorValues)`, η οποίες θα παίξουν καθοριστικό ρόλο στη λήψη αποφάσεων του ρομπότ.

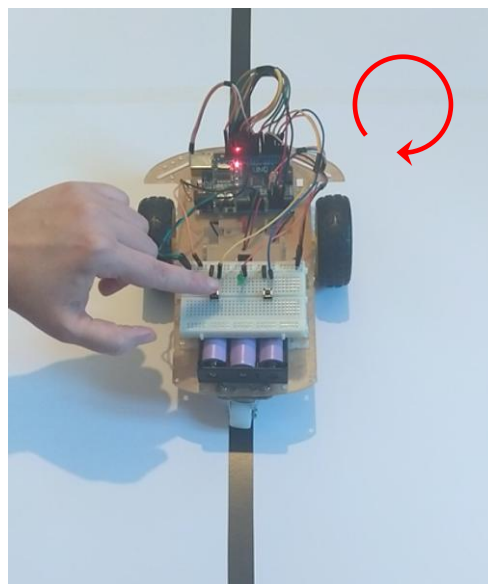
Η μέθοδος `calibrate()` διαβάζει τις τιμές των αισθητήρων πολλές φορές και καταγράφει την ελάχιστη και την μέγιστη μέτρησή τους. Αυτές οι μετρήσεις χρησιμοποιούνται στη συνέχεια για την κανονικοποίηση των αναγνώσεων των αισθητήρων σε ένα εύρος τιμών (0-1000) κάνοντας εύκολη την εύρεση της θέσης της γραμμής. Η βαθμονόμηση είναι απαραίτητη για να εξασφαλιστεί ότι οι αισθητήρες λειτουργούν σωστά και για να προσαρμοστούν στην αλλαγή της φωτεινότητας του περιβάλλοντος και της επιφάνειας που ανιχνεύουν.

Η μέθοδος `int readLineBlack(int sensorValues)` διαβάζει τις τιμές των αισθητήρων και επιστρέφει την βαθμονομημένη θέση της γραμμής (0-7000).

Κατά τη διάρκεια της βαθμονόμησης θα χρειαστεί το ρομπότ να κινηθεί πάνω στη γραμμή είτε χειροκίνητα (Εικόνα 1.9) είτε αυτόματα κινούμενο επί τόπου πάνω στη γραμμή (Εικόνα 1.10).



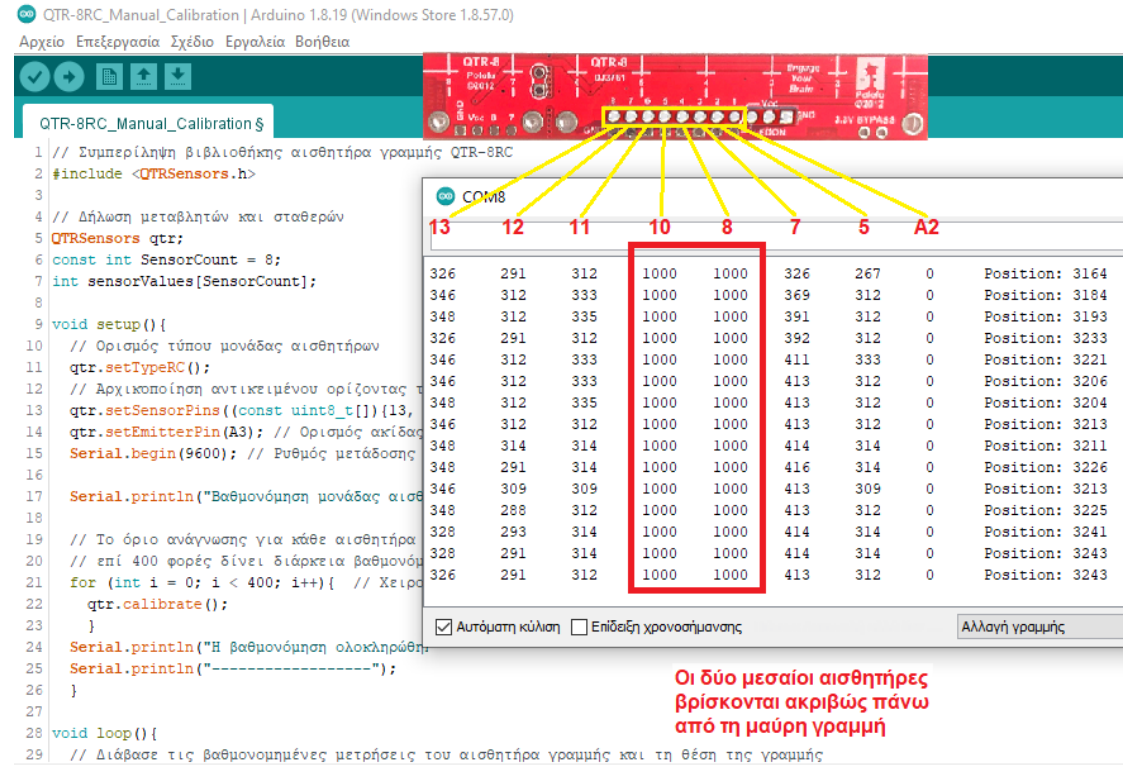
Εικόνα 1.10: Χειροκίνητη βαθμονόμηση



Εικόνα 1.11: Αυτόματη βαθμονόμηση

Παρακάτω φαίνονται τα αποτελέσματα των μετρήσεων του αισθητήρα και η θέση της γραμμής από την εκτέλεση του κώδικα στο Παράρτημα Β, Κώδικας 1 - Χειροκίνητη

βαθμονόμηση αισθητήρα γραμμής, με την χρήση του εργαλείου Serial Monitor στο Arduino IDE.



Εικόνα 1.12: Παρακολούθηση μετρήσεων αισθητήρα γραμμής

Η βαθμονομημένη θέση των αισθητήρων με τη σειρά που έχουν δηλωθεί κατά τη δημιουργία του αντικειμένου qtr είναι η εξής:

pin 13	pin 12	pin 11	pin 10	pin 8	pin 7	pin 5	pin A2
0	1000	2000	3000	4000	5000	6000	7000

Στο Serial Monitor οι μετρήσεις δείχνουν ότι η θέση της γραμμής είναι κάπου μεταξύ 3000 - 4000, που σημαίνει ότι βρίσκεται ακριβώς κάτω από το κέντρο της μονάδας QTR-8RC.

1.1.5 Διακόπτες ελέγχου

Στιγμιαίοι διακόπτες



Εικόνα 1.13: Στιγμιαίοι διακόπτες

Οι στιγμιαίοι διακόπτες ή κουμπιά (pushbuttons) είναι απλοί διακόπτες δύο θέσεων που ενεργοποιούνται με ένα κουμπί που πιέζεται και απελευθερώνεται, ανοίγοντας και κλείνοντας το κύκλωμα αντίστοιχα. Η χρήση τους είναι ιδιαίτερα σημαντική καθώς αποτελούν ένα αποτελεσματικό τρόπο αλληλεπίδρασης με διάφορες συσκευές. Στην κατασκευή θα χρησιμοποιηθούν δύο στιγμιαίοι διακόπτες επιτελώντας διαφορετικές λειτουργίες ο καθένας.

- Αριστερός διακόπτης - Αυτόματη βαθμονόμηση αισθητήρα γραμμής QTR-8RC
- Δεξιός διακόπτης - Εκτέλεση αλγορίθμου ακολουθίας γραμμής

Για την καλύτερη τακτοποίηση του χώρου που θα καταλαμβάνουν οι διακόπτες πάνω στην πλακέτα δοκιμών θα προτιμηθεί από τη μεριά της πλακέτας του μικροελεγκτή η ενεργοποίηση των ενσωματωμένων pull-up αντιστάσεων των 20KOhms των ψηφιακών ακροδεκτών εισόδου όπου θα συνδεθούν. Ο τρόπος λειτουργίας τους αναφέρεται στο Παράρτημα Β, Κώδικας 2 - Λειτουργία κουμπιών χειρισμού του ρομπότ.

Μόνιμος διακόπτης



Εικόνα 1.14: Διακόπτης On/Off

Ο μόνιμος διακόπτης είναι ένα απλό κουμπί On/Off που έχει δύο επαφές που θα ανοίγει και θα κλείνει το κύκλωμα τροφοδοσίας του ρομπότ. Στην κατασκευή θα παρεμβάλλεται ανάμεσα στην εξωτερική πηγή τάσης τροφοδοσίας του μικροελεγκτή Arduino και του ακροδέκτη του εισόδου τάσης Vin.

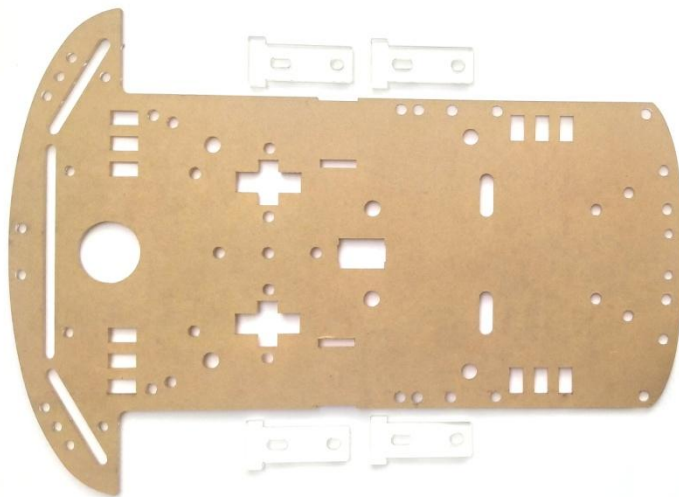
1.1.6 Ενδεικτική λυχνία λειτουργίας LED



Εικόνες 1.15: LED και αντίσταση 220Ω

Η δίοδος εκπομπής φωτεινής ακτινοβολίας ή LED (Light Emitting Diode) είναι ένα ηλεκτρονικό στοιχείο που εκπέμπει φως όταν διαρρέεται από ηλεκτρικό ρεύμα κατάλληλης τάσης και προς μία μόνο κατεύθυνση. Ανάλογα με την προβλεπόμενη τάση λειτουργίας του LED είναι σημαντικό να συμπεριληφθεί στο κύκλωμα μία αντίσταση εξισορρόπησης για την αποφυγή της καταστροφής του από υπέρταση. Τα LED χρησιμοποιούνται ευρέως σε συστήματα εποπτείας ως λυχνίες ενδείξεων και κατάστασης λειτουργίας σε διάφορες ηλεκτρονικές κατασκευές. Στην κατασκευή θα χρησιμοποιηθεί για την ένδειξη λειτουργίας βαθμονόμησης του αισθητήρα γραμμής QTR-8RC.

1.1.7 Σασί



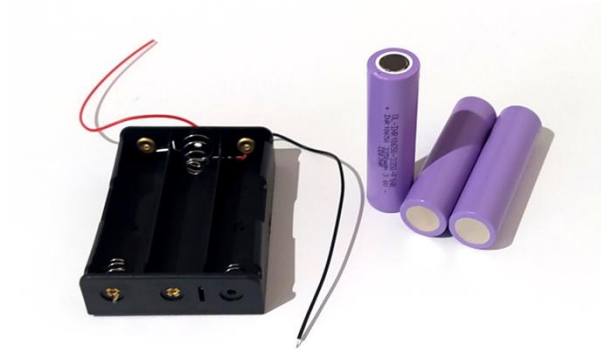
Εικόνα 1.16: Βάση με στηρίγματα για δύο κινητήρες

Το σασί η απλά η βάση του κινητού ρομπότ της κατασκευής είναι μία πλακέτα φτιαγμένη από πλεξιγκλάς που περιέχει διάφορες οπές και εσοχές για την

προσάρτηση και σύνδεση διαφόρων κυκλωμάτων και εξαρτημάτων. Το μέρος με την μεγαλύτερη καμπύλη θα είναι το μπροστινό μέρος του ρομπότ, ενώ το μέρος με την μικρότερη καμπύλη θα είναι το πίσω μέρος του ρομπότ.

1.1.8 Τροφοδοσία

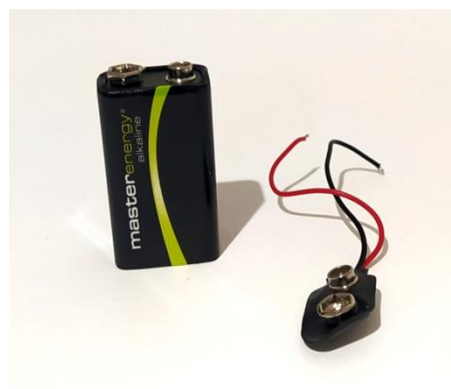
Τροφοδοσία κινητήρων



Εικόνα 1.17: Επαναφορτιζόμενες μπαταρίες τους και η θήκη τους

Η τροφοδοσία για τους κινητήρες θα είναι 3 επαναφορτιζόμενες μπαταρίες 18650 Li-Ion των 3.6 V συνδεδεμένες σε σειρά μέσα σε μία θήκη μπαταριών δίνοντας συνολική τάση 10.8V.

Τροφοδοσία της πλακέτας Arduino



Εικόνα 1.18: Πλακέ μπαταρία με το κούμπωμά της

Η τροφοδοσία της πλακέτας Arduino θα είναι μία μπαταρία πλακέ των 9V συνδεδεμένη με ένα ειδικό κούμπωμα, η οποία είναι αρκετή για να καλύψει τις ανάγκες τροφοδοσίας και των διασυνδεδεμένων περιφερειακών συσκευών σε αυτήν.

1.1.9 Καλώδια σύνδεσης



Εικόνα 1.19: Καλώδια σύνδεσης

Τα καλώδια σύνδεσης (jumper wires) που θα χρειαστούν στην κατασκευή είναι τύπου M-F και M-M και μήκους 20cm και απαραίτητα για την επικοινωνία των διαφόρων εξαρτημάτων και κυκλωμάτων. Υπάρχουν σε διάφορα χρώματα, πράγμα που διευκολύνει την αναγνώριση διαφόρων σημάτων στις συνδέσεις των ηλεκτρονικών διατάξεων του κινητού ρομπότ.

1.1.10 Τροχοί



Εικόνα 1.20: Οι τροχοί του κινητού ρομπότ

Οι τροχοί που θα κουμπωθούν στους άξονες των κινητήρων είναι φτιαγμένοι από πλαστικό με ελαστικά καουτσούκ με αντιολισθητική ιδιότητα και έχουν διάμετρο 6,5cm.

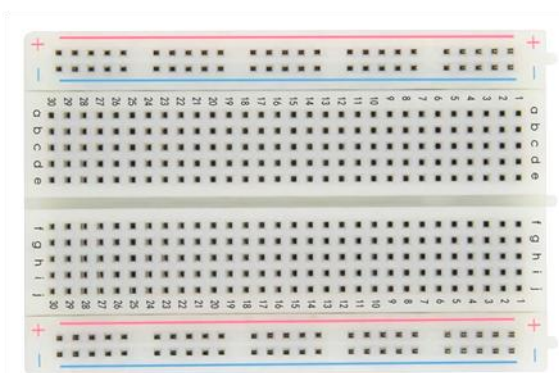
1.1.11 Ρόδα ελεύθερης περιστροφής



Εικόνα 1.21: Ρόδα ελεύθερης περιστροφής

Καθώς το ρομπότ θα έχει μόνο δύο τροχούς θα χρειαστεί για την στήριξη της βάσης του κινητού ρομπότ και μία ρόδα ελεύθερης περιστροφής.

1.1.12 Πλακέτα δοκιμών



Εικόνα 1.22: Πλακέτα δοκιμών

Η πλακέτα δοκιμών (breadboard) είναι μία βάση πάνω στην οποία προσαρτώνται και συνδέονται προσωρινά διάφορα ηλεκτρονικά κυκλώματα και εξαρτήματα για να δοκιμαστεί ένα πρότυπο ηλεκτρονικό κύκλωμα. Η τοποθέτησή τους δεν χρειάζεται κόλληση και μπορούν να επαναχρησιμοποιηθούν. Η πλακέτα δοκιμών που θα χρησιμοποιηθεί στην κατασκευή έχει διαστάσεις 8,2cm X 5,5cm.

Η κοστολόγηση των υλικών της κατασκευής αναφέρεται στο Παράρτημα Α.

1.2 Συναρμολόγηση και συνδεσμολογία εξαρτημάτων και κυκλωμάτων

Τα εργαλεία και τα υλικά που θα χρησιμοποιηθούν για τη συναρμολόγηση εξαρτημάτων και κυκλωμάτων της κατασκευής είναι τα εξής:

- Στραβοκατσάβιδο
- Βίδες M3 μήκους 10mm και 20mm
- Παξιμάδια M3
- Αποστάτες M3 μήκους 25mm
- Κοφτάκι συρμάτων
- Κολλητήρι και καλάι για τις συνδέσεις ηλεκτρονικών εξαρτημάτων
- Πιστόλι και ράβδοι σιλικόνης
- Κόλλα στιγμής για δύσκολα σημεία στερέωσης
- Δεματικά καλωδίων
- Κολλητική ταινία διπλής όψης
- Μαύρη μονωτική ταινία
- Μικρά κομμάτια από χαρτόνι για την κάλυψη κενών
- Ψαλίδι

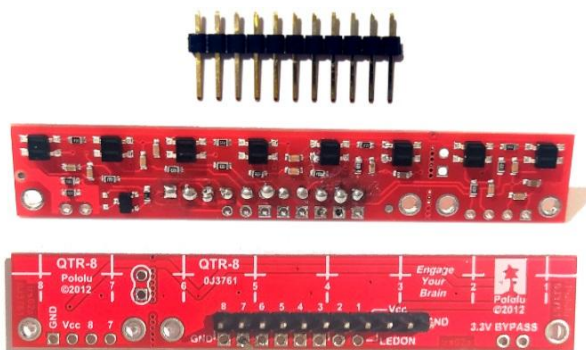
Πριν ξεκινήσει η συναρμολόγηση του σκελετού του κινητού ρομπότ, θα πρέπει να γίνουν προεργασίες σε κάποια εξαρτήματα και κυκλώματα.

Αρχικά, θα πρέπει να γίνει η κόλληση των καλωδίων στις επαφές των κινητήρων. Όπως φαίνεται παρακάτω τα καλώδια με κόκκινο και μαύρο χρώμα έχουν κολληθεί με διαφορετική σειρά σε κάθε κινητήρα. Με αυτόν τον τρόπο διατηρείται μία σειρά στις συνδέσεις αργότερα στο κύκλωμα που θα τους ελέγχει.



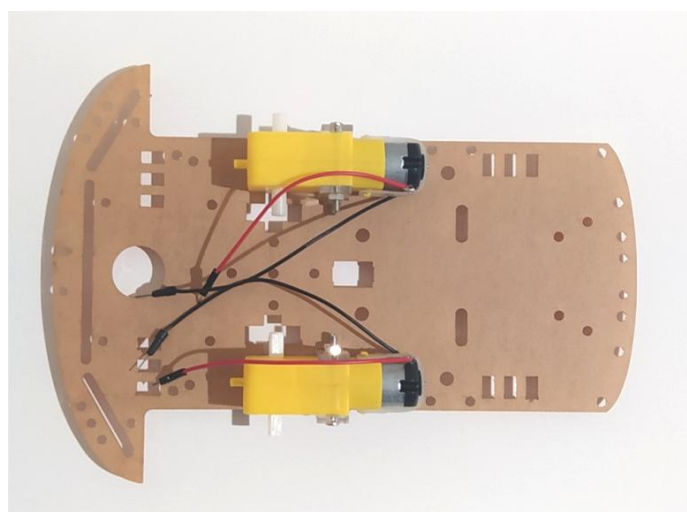
Εικόνα 1.23: Κόλληση καλωδίων στους κινητήρες

Στην μονάδα του αισθητήρα γραμμής QTR-8RC θα πρέπει να γίνει η κόλληση και των 11 ψηφιακών ακροδεκτών με μία σειρά ακίδων όπου εκεί θα μπορούν να συνδεθούν καλώδια μέτρησης, ελέγχου και τροφοδοσίας από τον μικροελεγκτή Arduino.



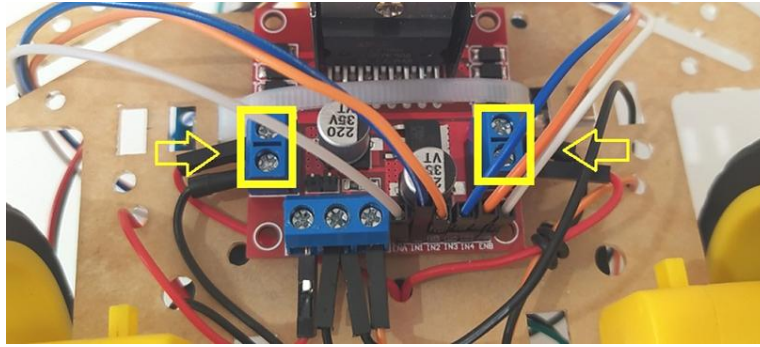
Εικόνα 1.24: Κόλληση σειράς ακίδων στην μονάδα QTR-8RC

Στη συνέχεια ακολουθεί το στήσιμο των επιμέρους εξαρτημάτων και κυκλωμάτων. Ξεκινώντας από τους κινητήρες, θα πρέπει να γίνει η στήριξή τους στην αριστερή και δεξιά πλευρά της βάσης χρησιμοποιώντας στηρίγματα, βίδες και παξιμάδια.



Εικόνα 1.25: Στήριξη κινητήρων στη βάση

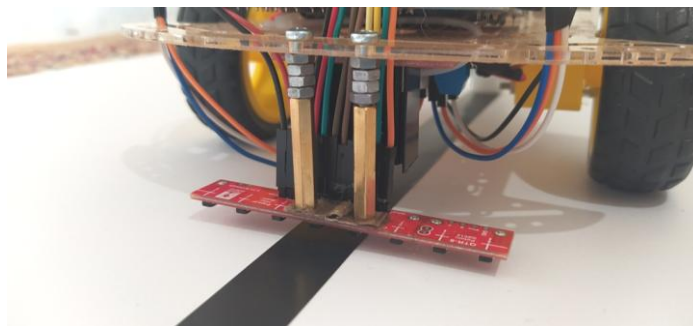
Για την οδήγησή τους θα χρειαστεί να γίνει η σύνδεσή τους με τον οδηγό κινητήρων L298N, ο οποίος θα πρέπει να έχει στερεωθεί κάπου ανάμεσα στους δύο κινητήρες και πιο κοντά στο μπροστινό μέρος της βάσης. Παρακάτω φαίνεται πως ο οδηγός είναι σταθεροποιημένος με τη βοήθεια ενός δεματικού καλωδίων. Στους πλαστικούς άξονες των κινητήρων που προεξέχουν από την εξωτερική πλευρά της κατασκευής θα κουμπωθούν οι δύο κινητήριοι τροχοί.



Εικόνα 1.26: Σύνδεση κινητήρων στη μονάδα L298N

Στην συνέχεια, η πλακέτα του μικροελεγκτή Arduino θα πρέπει να κολληθεί πάνω από την άλλη πλευρά της βάσης, έχοντας από κάτω της τον οδηγό κινητήρων. Η κόλλησή της μπορεί να γίνει με τη βοήθεια του πιστολιού σιλικόνης. Στην περίπτωση που δεν εφάπτεται καλά στην επιφάνεια της βάσης η πλακέτα, μπορούν να χρησιμοποιηθούν κάποια κομμάτια χαρτονιού για την κάλυψη των κενών ενδιάμεσα.

Στην μπροστική πλευρά της βάσης με τη μεγάλη καμπύλη υπάρχουν οπές που μπορούν να χρησιμοποιηθούν για τη συγκράτηση δύο αποστατών που θα συγκρατούν έχοντας κολλημένο πάνω τους τον αισθητήρα ακολουθίας γραμμής QTR-8RC. Η απόσταση του αισθητήρα από την επιφάνεια θα πρέπει να είναι 0.90cm.

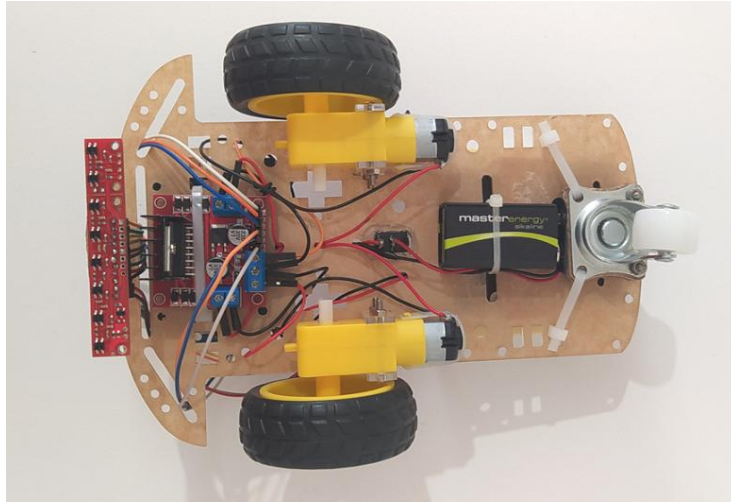


Εικόνα 1.27: Η απόσταση του αισθητήρα γραμμής από την επιφάνεια

Ακολούθως, το επόμενο βήμα είναι η προσάρτηση της ρόδας ελεύθερης περιστροφής στην κάτω πλευρά της βάσης (Εικόνα 1.27). Εδώ μπορούν να χρησιμοποιηθούν αποστάτες ή κομμάτια από χαρτόνι ανάμεσα στη βάση της ρόδας με τη βάση του ρομπότ φέρνοντας τον σκελετό παράλληλα στο επίπεδο.

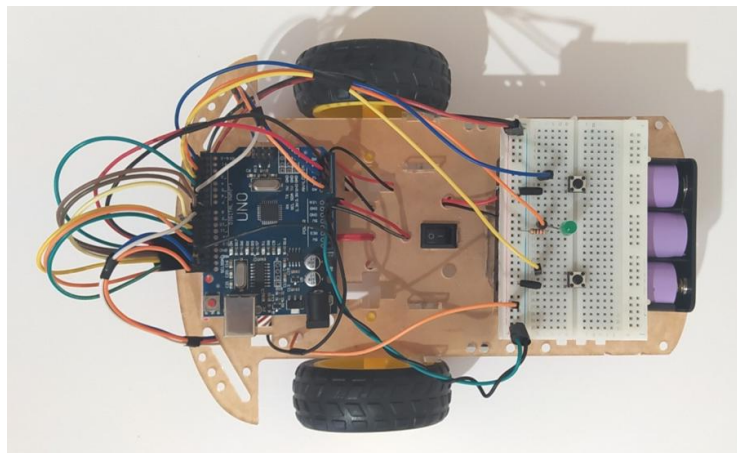
Ο χώρος που περισσεύει κοντά στην ρόδα ελεύθερης περιστροφής και εσωτερικά της κατασκευής είναι ο πλέον κατάλληλος για να μπει η πλακέ μπαταρία για την τροφοδοσία του μικροελεγκτή. Το δέσιμό της μπορεί να γίνει με δεματικό καλωδίων καθώς βολεύουν οι οπές της βάσης.

Στο κέντρο της βάσης υπάρχει μία ειδική υποδοχή για το κουμπί ελέγχου τροφοδοσίας On/Off του ρομπότ, το οποίο μπορεί να κολληθεί με το πιστόλι σιλικόνης για ένα πιο σταθερό αποτέλεσμα.



Εικόνα 1.28: Η κάτω πλευρά

Από την πάνω πλευρά της βάσης (Εικόνα 1.28), φαίνεται να είναι ξεκάθαρο το μέρος όπου θα κολληθεί η τροφοδοσία των κινητήρων. Η κόλληση της θήκης μπαταριών μπορεί να γίνει με παρόμοιο τρόπο όπως με την πλακέτα του μικροελεγκτή Arduino.

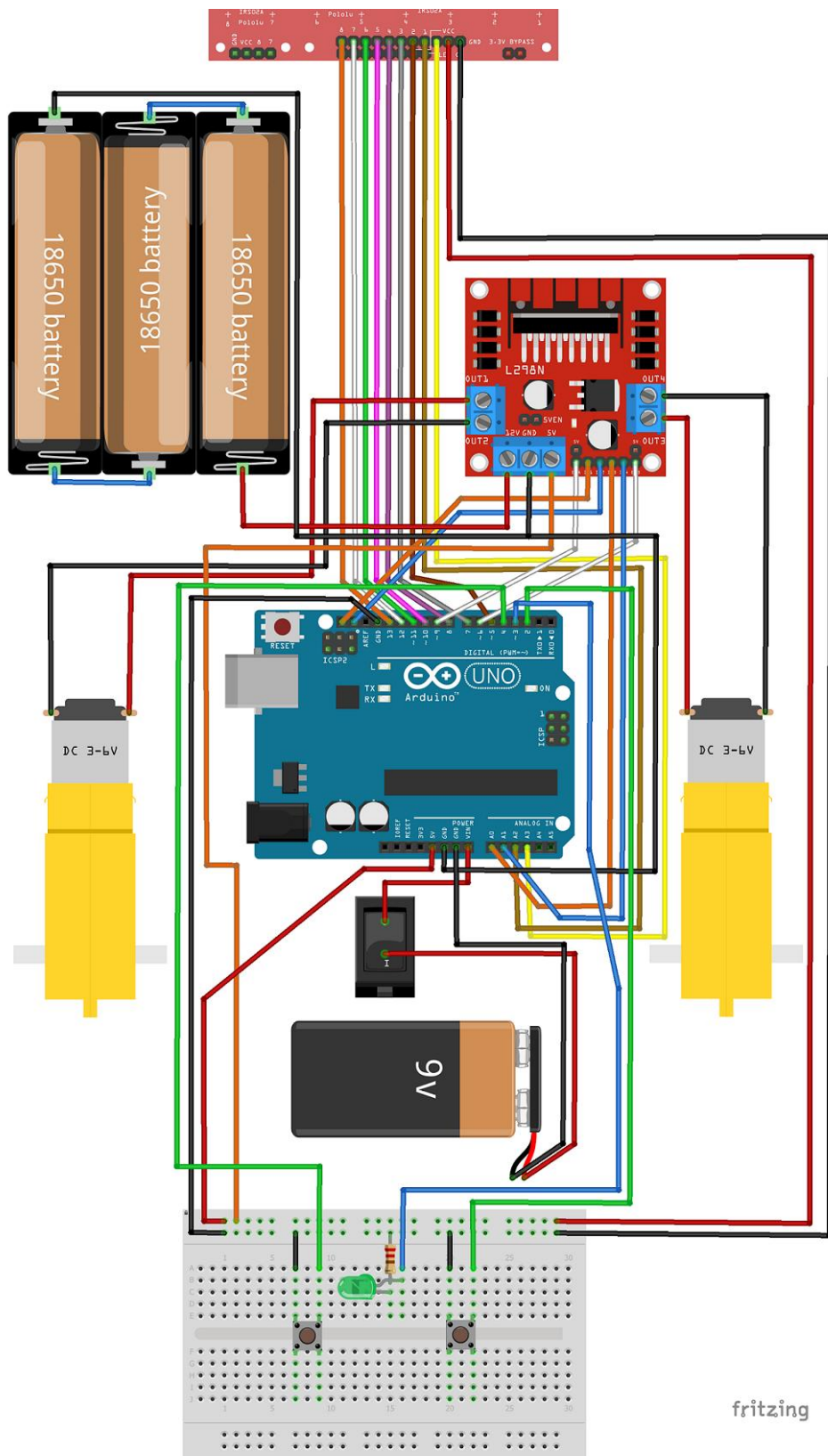


Εικόνα 1.29: Η πάνω πλευρά

Πάνω από την τροφοδοσία των κινητήρων θα τοποθετηθεί η πλακέτα δοκιμών με τα στοιχεία ελέγχου (LED, αντίσταση 220Ω, στιγμιαίοι διακόπτες και καλώδια συνδέσεων). Προκειμένου να είναι εύκολη η αφαίρεση των επαναφορτιζόμενων μπαταριών, η πλακέτα θα είναι προσωρινά στερεωμένη πάνω τους με μονωτική ταινία.

Οι διαστάσεις του ρομπότ είναι 21,3cm μήκος x 15,2cm πλάτος και ζυγίζει 538gr.

Ο τρόπος συνδεσμολογίας των εξαρτημάτων και των κυκλωμάτων του ρομπότ φαίνεται στο παρακάτω διάγραμμα.



Εικόνα 1.30: Εικονικό διάγραμμα συνδέσεων κινητού ρομπότ

Κεφάλαιο 2. Προγραμματισμός κινητού ρομπότ ακολουθίας γραμμής

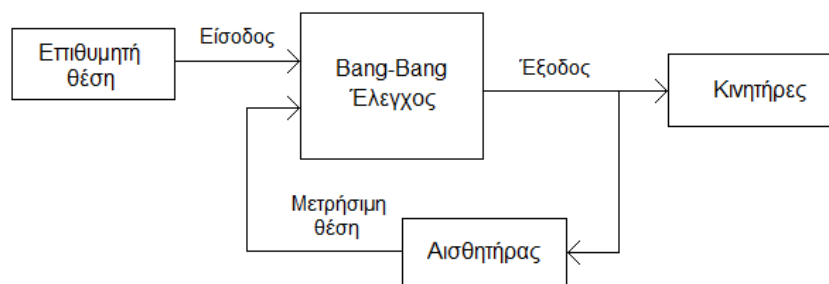
Για τον προγραμματισμό του ρομπότ θα χρησιμοποιηθούν βασικοί αλγόριθμοι αυτόματου ελέγχου ανατροφοδότησης, από τον πιο απλό στον πιο σύνθετο, δίνοντας μία βαθύτερη εικόνα της ανάγκης βελτίωσης του τρόπου κίνησής του πάνω στην γραμμή ώστε να την ακολουθεί ικανοποιητικά.

Κατά την διάρκεια αναζήτησης του αποδοτικότερου αλγορίθμου ελέγχου ακολουθίας γραμμής, το ρομπότ θα δοκιμάζεται σε διάφορες πίστες όπως φαίνονται στο Παράρτημα Γ.

2.1. Bang-Bang Έλεγχος

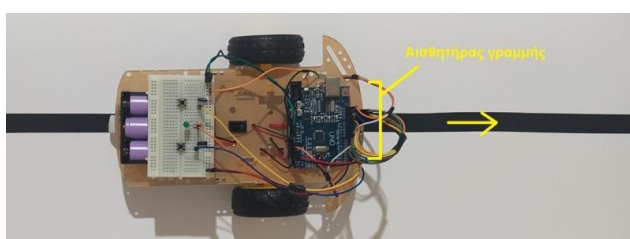
Ο Bang-Bang έλεγχος είναι ένας απλός αλγόριθμος ελέγχου ανατροφοδότησης ανοιχτού βρόχου όπου με βάση μία τιμή εισόδου (κατώφλι) δίνει μία ανάλογη έξοδο μεταξύ δύο ακραίων καταστάσεων.

Στον αλγόριθμο ακολουθίας γραμμής ο Bang-Bang έλεγχος συγκρίνει κάθε φορά την τρέχουσα θέση της γραμμής με την επιθυμητή θέση της ρυθμίζοντας ανάλογα το τιμόνι του ρομπότ.



Εικόνα 2.1: Διάγραμμα ακολουθίας γραμμής Bang-Bang

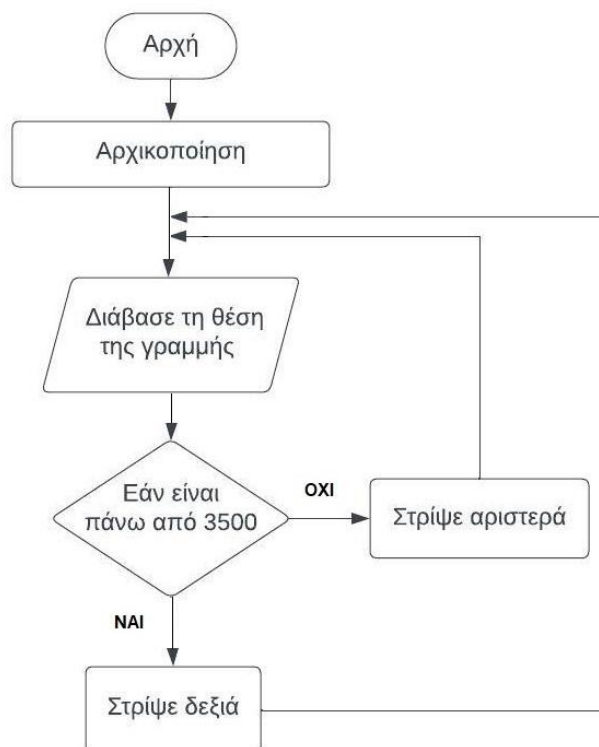
Στο πρόγραμμα το ρομπότ θα ρυθμίζει συνεχώς την κατεύθυνση στην οποία θα πρέπει να κινηθεί προκειμένου να κρατήσει τον αισθητήρα πάνω από την γραμμή.



Εικόνα 2.2: Η επιθυμητή θέση του αισθητήρα γραμμής

Καθώς το ρομπότ κινείται προς τα εμπρός θα διαβάζει την ένδειξη του αισθητήρα για να δει που βρίσκεται σε σχέση με τη γραμμή. Εάν η γραμμή βρίσκεται από τη δεξιά πλευρά του κέντρου του αισθητήρα το ρομπότ θα στρίβει προς τα δεξιά. Εάν η γραμμή βρίσκεται από την αριστερή πλευρά του κέντρου του αισθητήρα το ρομπότ θα στρίβει προς τα αριστερά. Για να παραμένει το ρομπότ πάνω στη γραμμή η τιμή κατωφλίου (επιθυμητή θέση) είναι 3500, δηλαδή η γραμμή θα πρέπει να βρίσκεται κάτω από τον 4ο και τον 5ο αισθητήρα της μονάδας QTR-8RC.

Η ταχύτητα των κινητήρων και η ρύθμιση του τιμονιού επηρεάζουν σημαντικά το πόσο καλά θα ακολουθήσει το ρομπότ τη γραμμή. Αν η ταχύτητα είναι πολύ χαμηλή το ρομπότ θα αργήσει να ακολουθήσει τη γραμμή. Αν η ταχύτητα είναι υπερβολικά υψηλή το ρομπότ αντιδρά με δυσκολία στις αλλαγές της κατεύθυνσης της γραμμής ώστε να μπορέσει να παραμείνει πάνω σε αυτή. Η ιδανική τιμή ψευδοαναλογικής εξόδου για τον ορισμό της ταχύτητας στους κινητήρες μετά από δοκιμές είναι η 65 (0-255) και το είδος στροφής του τιμονιού είναι κλειστή στροφή (κίνηση στον έναν τροχό). Αν οι κινητήρες είχαν και οι δύο ταχύτητες με κάποια διαφορά μεταξύ τους το ρομπότ θα απομακρυνόταν από τη γραμμή και θα ήταν αδύνατη η επαναφορά του.



Εικόνα 2.3: Διάγραμμα ροής αλγορίθμου ακολουθίας γραμμής Bang-Bang

Στην πρώτη δοκιμή του κώδικα στο Παράρτημα Β, Κώδικας 3 - Αλγόριθμος ακολουθίας γραμμής Bang-Bang, το ρομπότ με βάση το πώς έχει τοποθετηθεί εξαρχής πάνω στην ευθεία γραμμή, στρίβει κάθε φορά μία δεξιά μία αριστερά με σταθερή γωνία στροφής τιμονιού ανεξαρτήτου της απόστασης που απέχει από αυτή.



Εικόνα 2.4: Bang-Bang έλεγχος

Στην ευθεία και σε ομαλές στροφές το ρομπότ κινείται ικανοποιητικά αλλά αργά κάνοντας μικρές ταλαντώσεις, ενώ σε απότομες στροφές ή όταν υπάρχουν απρόβλεπτοι παράγοντες χάνει εύκολα τον έλεγχο με αποτέλεσμα να παρεκτρέπεται από τον επιθυμητό στόχο κάνοντας κύκλους προς τα δεξιά ή προς τα αριστερά.

Μία παραλλαγή του αλγορίθμου αυτού θα ήταν το ρομπότ με βάση την ανάγνωση του αισθητήρα να κάνει τρεις λειτουργίες έχοντας δύο τιμές κατωφλίου. Δηλαδή το ρομπότ να στρίβει δεξιά, να πηγαίνει ευθεία ή να στρίβει αριστερά. Παρά τη βελτίωση του συστήματος ελέγχου του ρομπότ με τη μέθοδο αυτή, το ρομπότ ενώ κινείται πιο γρήγορα πάνω σε ευθεία γραμμή, κινείται πιο αργά στις απότομες αλλαγές της κατεύθυνσης της γραμμής με σταθερή γωνία στροφής τιμονιού.

2.2. Αναλογικός Ελεγκτής

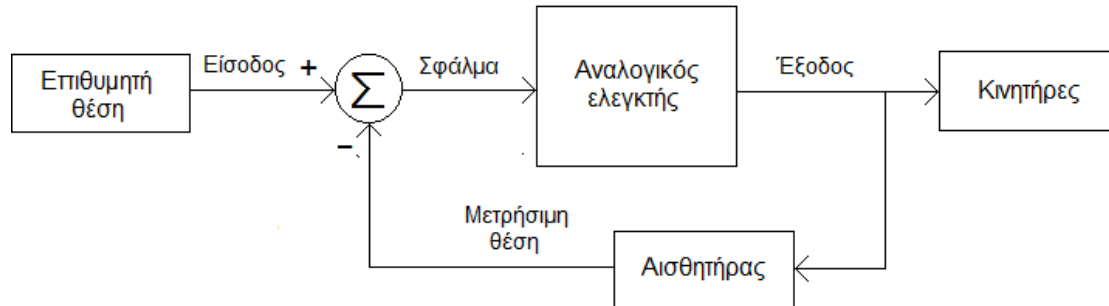
Μία πρώτη βελτίωση του αλγορίθμου ακολουθίας γραμμής Bang-Bang είναι η γωνία στροφής του τιμονιού να είναι μεγαλύτερη όταν το ρομπότ απέχει μεγαλύτερη απόσταση από τη γραμμή και όσο πλησιάζει να μικραίνει επαναφέροντάς το στη σωστή πορεία. Η μέθοδος αυτή δίνει μία καμπυλότητα στην κίνηση του ρομπότ με μαλακή οδήγηση για πιο απαλές στροφές και απότομο τιμόνι για πιο απότομες στροφές. Αυτό κάνει το ρομπότ να ανταποκρίνεται πιο γρήγορα σε αλλαγές στην κατεύθυνση της γραμμής ενώ συνεχίζει να κινείται ομαλά όταν είναι ευθεία. Αυτή η προσέγγιση ελέγχου ονομάζεται αναλογικός ελεγκτής διότι η γωνία στροφής του τιμονιού είναι ανάλογη της απόστασης του ρομπότ από τη γραμμή.

Ο αναλογικός ελεγκτής υπολογίζει αυτήν την απόκλιση κάθε φορά δίνοντας μία τιμή σφάλματος ως τη διαφορά μεταξύ της επιθυμητής θέσης της γραμμής με την

τρέχουσα θέση της γραμμής, δημιουργώντας μία δράση ανάλογη της απόκλισης αυτού του σφάλματος και με μία σταθερά αναλογικού κέρδους K_p .

$$\text{Τρέχον σφάλμα (P)} = \text{Επιθυμητή θέση} - \text{Τρέχουσα θέση}$$

$$\text{Γωνία στροφής τιμονιού} = P \times K_p$$



Εικόνα 2.5: Διάγραμμα ακολουθίας γραμμής αναλογικού ελεγκτή

Η τιμή κέρδους καθορίζει το πόσο γρήγορα αντιδρά το ρομπότ σε αλλαγές της τιμής σφάλματος. Η εύρεση της βέλτιστης τιμής κέρδους εξαρτάται από διάφορους παράγοντες όπως το βάρος του ρομπότ, την απόσταση του τιμονιού από τον αισθητήρα, το αν υπάρχει συμμετρία στην κατασκευή, την ισχύ των κινητήρων και την ευαισθησία των αισθητήρων υπερύθρων. Η επιλογή της κατάλληλης τιμής κέρδους ονομάζεται συντονισμός του ελεγκτή και η εύρεσή της θα γίνεται εμπειρικά από εδώ και στο εξής σε διάφορες πίστες δοκιμών με τη μέθοδο δοκιμής και λάθους (trial and error).

Παρακάτω στη συνάρτηση FollowLine() φαίνεται η υλοποίηση του αλγορίθμου ακολουθίας γραμμής αναλογικού ελεγκτή, καθώς προηγουμένως έχει γίνει ήδη η βαθμονόμηση του αισθητήρα γραμμής και έχει οριστεί η φορά περιστροφής των κινητήρων ώστε να κινείται το ρομπότ προς τα μπροστά.

```
// Αλγόριθμος ακολουθίας γραμμής αναλογικού ελεγκτή
void FollowLine(){
    // Διάβασε τις βαθμονομημένες τιμές του αισθητήρα και τη θέση της μαύρης γραμμής ( 0 - 7000)
    int position = qtr.readLineBlack(sensorValues);

    error = setpoint - position;
    P = error;
```

```

motorSpeed = P * Kp;

motorSpeedA = minMotorSpeedA - motorSpeed;
motorSpeedB = minMotorSpeedA + motorSpeed;

// Κανονικοποίηση τιμών μεγίστου και ελαχίστου ισχύος κινητήρων σε αποδεκτό εύρος τιμών
if (motorSpeedA > maxMotorSpeedA) {
    motorSpeedA = maxMotorSpeedA;
}
if (motorSpeedB > maxMotorSpeedB) {
    motorSpeedB = maxMotorSpeedB;
}
if (motorSpeedA < 0) {
    motorSpeedA = 0;
}
if (motorSpeedB < 0) {
    motorSpeedB = 0;
}

// Γωνία στροφής τιμονιού
analogWrite(enA,motorSpeedA);
analogWrite(enB,motorSpeedB);
}

```

Μετά από πολλά πειράματα δοκιμών του κώδικα στο ρομπότ για την εύρεση της κατάλληλης τιμής κέρδους K_p , βρέθηκαν τα εξής αποτελέσματα:

- Όταν η τιμή κέρδους είναι χαμηλή ($K_p \geq 0$), το ρομπότ κινείται αργά κάνοντας ταλαντώσεις μεγάλου πλάτους κατά μήκος της γραμμής και σε κλειστές στροφές δεν αντιδρά αρκετά γρήγορα.
- Όταν η τιμή κέρδους είναι μεσαία, το ρομπότ κινείται λίγο πιο γρήγορα αλλά οι ταλαντώσεις συνεχίζουν να υπάρχουν με λίγο μικρότερο πλάτος.
- Όταν η τιμή κέρδους είναι υψηλή ($K_p \geq 9$), το ρομπότ αντιδρά ταχύτερα στις αλλαγές, οι ταλαντώσεις του αποκτούν μικρότερο πλάτος κατά τη διάρκεια της κίνησής του μέχρι που κινείται ικανοποιητικά πάνω στη γραμμή.

Παρόλο που η υψηλή τιμή κέρδους $K_p=9$ αποτελεί την καλύτερη επιλογή, οι ταλαντώσεις στην κίνηση του ρομπότ συνεχίζουν υπάρχουν.

2.3. Αναλογικός - Διαφορικός Ελεγκτής

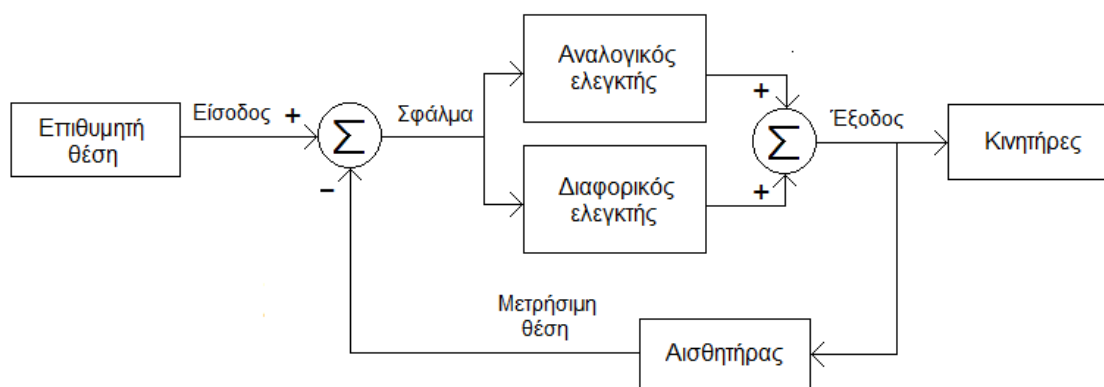
Για την εξάλειψη των ταλαντώσεων στην κίνηση του ρομπότ και για την αντιμετώπιση ξαφνικών αλλαγών στην κατεύθυνση της γραμμής (υπέρβαση), το ρομπότ θα πρέπει να κάνει μικρές αλλαγές όταν κινείται σε ευθεία γραμμή και μεγαλύτερες όταν κινείται σε καμπύλη γραμμή. Για το λόγο αυτό, στο σύστημα ελέγχου του ρομπότ προστίθεται μαζί με τον αναλογικό ελεγκτή και ο διαφορικός ελεγκτής.

Ο διαφορικός ελεγκτής είναι στην ουσία ο ρυθμός μεταβολής του σφάλματος της απόστασης του ρομπότ από τη γραμμή. Όταν η γραμμή είναι ευθεία, η διαφορική τιμή είναι μικρή επειδή η μέτρηση του αισθητήρα δεν αλλάζει πολύ. Όταν το ρομπότ φτάσει στη στροφή, η μέτρηση ξαφνικά αλλάζει σε μεγάλο βαθμό και η διαφορική τιμή γίνεται μεγάλη επαναφέροντάς το γρήγορα πάνω στην γραμμή. Επειδή ο διαφορικός ελεγκτής προσπαθεί να μειώσει την επίδραση αυτού του σφάλματος στο μέλλον είναι γνωστός για την «προληπτική δράση» του.

Στον υπολογισμό της γωνίας στροφής του τιμονιού προστίθεται επιπλέον ο διαφορικός όρος που είναι το γινόμενο του ρυθμού μεταβολής του σφάλματος με μία σταθερά διαφορικού κέρδους K_d .

$$\text{Διαφορική τιμή (D)} = \text{Σφάλμα} - \text{Προηγούμενο σφάλμα}$$

$$\text{Γωνία στροφής τιμονιού} = P \times K_p + D \times K_d$$



Εικόνα 2.6: Διάγραμμα ακολουθίας γραμμής PD Ελεγκτή

Παρακάτω φαίνεται η ενημερωμένη έκδοση της συνάρτησης FollowLine() με τον PD ελεγκτή.

```
// Αλγόριθμος ακολουθίας γραμμής αναλογικού και διαφορικού ελεγκτή
void FollowLine(){
    // Διάβασε τις βαθμονομημένες τιμές του αισθητήρα και τη θέση της μαύρης γραμμής ( 0 - 7000)
    int position = qtr.readLineBlack(sensorValues);

    error = setpoint - position;
    P = error;

    D = error - lastError;
    lastError = error;

    motorSpeed = P * Kp + D * Kd;

    motorSpeedA = minMotorSpeedA - motorSpeed;
    motorSpeedB = minMotorSpeedA + motorSpeed;

    // Κανονικοποίηση τιμών μεγίστου και ελαχίστου ισχύος κινητήρων σε αποδεκτό εύρος τιμών
    if (motorSpeedA > maxMotorSpeedA) {
        motorSpeedA = maxMotorSpeedA;
    }
    if (motorSpeedB > maxMotorSpeedB) {
        motorSpeedB = maxMotorSpeedB;
    }
    if (motorSpeedA < 0) {
        motorSpeedA = 0;
    }
    if (motorSpeedB < 0) {
        motorSpeedB = 0;
    }

    // Γωνία στροφής τιμονιού
    analogWrite(enA,motorSpeedA);
    analogWrite(enB,motorSpeedB);
}
```

Μετά από πολλά πειράματα δοκιμών του κώδικα στο ρομπότ για την εύρεση της κατάλληλης τιμής κέρδους K_d , βρέθηκαν τα εξής αποτελέσματα:

- Όταν η τιμή κέρδους είναι χαμηλή ($K_d \geq 0$), το ρομπότ δεν σημειώνει βελτίωση στην ταχύτητά του.
- Όταν η τιμή κέρδους είναι μεσαία, το ρομπότ παρουσιάζει μία μικρή βελτίωση στην κίνησή του.
- Όταν η τιμή κέρδους είναι υψηλή ($K_d \geq 9$), το ρομπότ κινείται ικανοποιητικά στη γραμμή και με τη μέγιστη δυνατή ταχύτητα στις κλειστές στροφές συνδυάζοντας κατάλληλες ποσότητες φρένου-γκαζιού κατά την είσοδό του σε αυτές.

Αν και η υψηλή τιμή διαφορικού κέρδους $K_d=9$ είναι η καλύτερη επιλογή, το ρομπότ συνεχίζει να κάνει ταλαντώσεις μικρού πλάτους, Επίσης, στην ύπαρξη απρόβλεπτων εμποδίων το ρομπότ αποκλίνει αισθητά από την επιθυμητή θέση του πάνω στη γραμμή κάνοντας ταλαντώσεις μεγαλύτερου πλάτους οι οποίες αργούν στην πάροδο του χρόνου να αποσβεστούν.

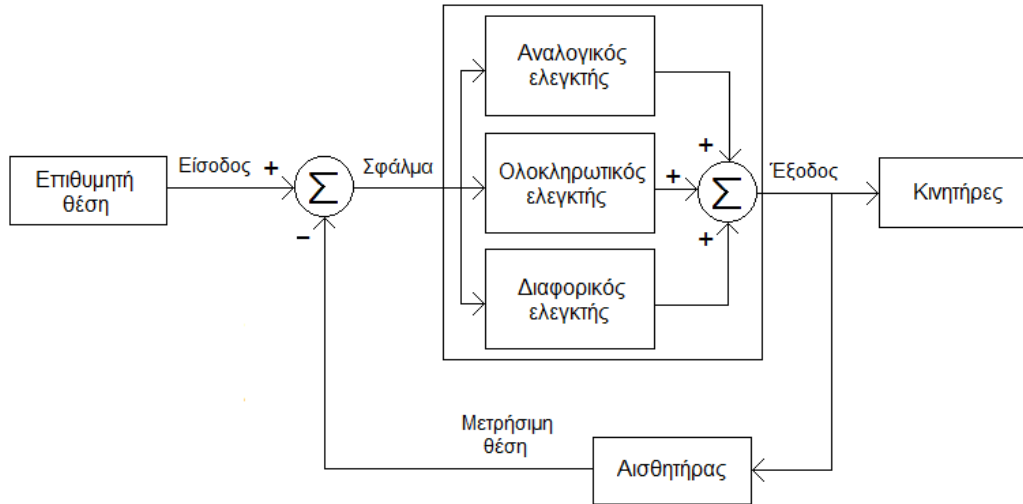
2.4. Αναλογικός - Ολοκληρωτικός - Διαφορικός Ελεγκτής

Μπορεί με τον διαφορικό όρο να μειώθηκαν αισθητά οι ταλαντώσεις του ρομπότ, παρόλα αυτά παραμένει επιρρεπείς σε απρόβλεπτους εξωτερικούς παράγοντες. Η δράση του αναλογικού και του διαφορικού όρου δεν είναι αρκετή στην περίπτωση που το ρομπότ συναντήσει εμπόδια στο δρόμο του. Μπορεί βέβαια να το επαναφέρουν στη γραμμή αλλά δημιουργείται μία σταθερή μόνιμη απόκλιση το λεγόμενο σφάλμα μόνιμης κατάστασης. Στον υπολογισμό ενός πιο ευέλικτου συστήματος ελέγχου, με το ρομπότ να ανταποκρίνεται γρηγορότερα στην ύπαρξη εμποδίων στο δρόμο του, μπαίνει και ο ολοκληρωτικός ελεγκτής ο οποίος προσπαθεί να μειώσει αυτό το σφάλμα στην πάροδο του χρόνου και είναι γνωστός για την «επίμονη δράση» του.

Στον υπολογισμό της γωνίας στροφής του τιμονιού προστίθεται επιπλέον ο ολοκληρωτικός όρος που είναι το γινόμενο του αθροίσματος όλων των προηγούμενων σφαλμάτων με μία σταθερά ολοκληρωτικού κέρδους K_i .

Νέα ολοκληρωτική τιμή (I) = Ολοκληρωτική τιμή + Σφάλμα

Γωνία στροφής τιμονιού = P x Kp + D x Kd + I x Ki



Εικόνα 2.7: Διάγραμμα ακολουθίας γραμμής PID ελεγκτή

Παρακάτω φαίνεται η τελευταία έκδοση της συνάρτησης FollowLine() με τον PID ελεγκτή.

```
// Αλγόριθμος ακολουθίας γραμμής αναλογικού, ολοκληρωτικού και διαφορικού ελεγκτή
void FollowLine(){
    // Διάβασε τις βαθμονομημένες τιμές του αισθητήρα και τη θέση της μαύρης γραμμής ( 0 - 7000)
    int position = qtr.readLineBlack(sensorValues);

    error = setpoint - position;
    P = error;

    I = I + error;

    D = error - lastError;
    lastError = error;

    motorSpeed = P * Kp + I * Ki + D * Kd;

    motorSpeedA = minMotorSpeedA - motorSpeed;
    motorSpeedB = minMotorSpeedA + motorSpeed;
}
```

```

// Κανονικοποίηση τιμών μεγίστου και ελαχίστου ισχύος κινητήρων σε αποδεκτό εύρος τιμών
if (motorSpeedA > maxMotorSpeedA) {
    motorSpeedA = maxMotorSpeedA;
}
if (motorSpeedB > maxMotorSpeedB) {
    motorSpeedB = maxMotorSpeedB;
}
if (motorSpeedA < 0) {
    motorSpeedA = 0;
}
if (motorSpeedB < 0) {
    motorSpeedB = 0;
}

// Γωνία στροφής τιμονιού
analogWrite(enA,motorSpeedA);
analogWrite(enB,motorSpeedB);
}

```

Μετά από πολλά πειράματα δοκιμών του κώδικα στο ρομπότ για την εύρεση της κατάλληλης τιμής κέρδους K_i , βρέθηκαν τα εξής αποτελέσματα:

- Όταν η τιμή κέρδους είναι χαμηλή ($K_i \geq 0$), το ρομπότ παρουσιάζει μία μικρή διόρθωση στην κίνησή του.
- Όταν η τιμή κέρδους είναι μεσαία, το ρομπότ κινείται λίγο αργά.
- Όταν η τιμή κέρδους είναι υψηλή ($K_i \geq 9$), δημιουργείται μία μόνιμη αστάθεια στην κίνησή του.

Στην περίπτωση αυτή, η καλύτερη τιμή ολοκληρωτικού κέρδους είναι η $K_i = 0.01$.

Το συμπέρασμα είναι ότι η ιδανική επιλογή για το παρόν κινητό ρομπότ ακολουθίας γραμμής ο είναι ο PID ελεγκτής παρέχοντας περισσότερη σταθερότητα στην κίνησή του με ελάχιστη μετατόπιση και μειωμένη υπέρβαση.

Στο Παράρτημα Β και στον Κώδικα 4 φαίνεται η πλήρης λειτουργικότητα του ρομπότ με την υλοποίηση του αλγορίθμου ακολουθίας γραμμής PID ελεγκτή.

3. Συμπεράσματα - Πιθανές επεκτάσεις

Κατά τη διάρκεια δοκιμών ελέγχου του ρομπότ μελετήθηκαν απλές τεχνικές για την επίτευξη ικανοποιητικής λειτουργίας ακολουθίας γραμμής με αρκετά μεγάλες ταχύτητες. Η μέθοδος δοκιμής και λάθους αποτέλεσε μία επίπονη διαδικασία παρά την απλότητά της για τον συντονισμό των επιμέρους σταθερών όρων του PID ελεγκτή. Στο πρόβλημα αυτό θα μπορούσε να βοηθήσει η δημιουργία μίας εφαρμογής ασύρματου συντονισμού παραμέτρων. Οι δυνατότητες της κατασκευής ήταν περιορισμένες από άποψη σχεδιασμού και προσάρτησης εξαρτημάτων. Η απόσταση του αισθητήρα γραμμής από το σύστημα τιμονιού θα έπρεπε να είναι μεγαλύτερη ώστε να παίρνει έγκαιρα αποφάσεις για την κίνησή του πάνω στην γραμμή. Το βάρος του θα μπορούσε να ήταν πιο λίγο διευκολύνοντας την κίνηση στους κινητήρες, καταναλώνοντας λιγότερη ισχύ και περιορίζοντας το θόρυβο που παράγουν. Πιθανή λύση στο πρόβλημα αυτό θα ήταν η χρήση κινητήρων χωρίς το σύστημα μετάδοσης κίνησης και αλλαγής κατεύθυνσης που έχουν οι DC Gearbox κινητήρων. Οι πιο φαρδιοί τροχοί θα έλυναν το πρόβλημα της ευθυγράμμισης στους τροχούς ενισχύοντας την ευστάθεια του οχήματος. Ο όγκος του λειτουργεί ανασταλτικά στην ανάπτυξη μεγαλύτερων ταχυτήτων. Ο αεροδυναμικός σχεδιασμός θα ήταν σίγουρα μία καλύτερη λύση. Η επιφάνεια δοκιμής θα πρέπει να είναι καθαρή από πιθανά εμπόδια ώστε το ρομπότ να μπορεί να κινείται απρόσκοπτα και ικανοποιητικά πάνω στη γραμμή. Τα λάστιχα των τροχών παρουσιάζουν μικρή πρόσφυση στις πίστες με αποτέλεσμα το ρομπότ να παρεκτρέπεται εύκολα στις υψηλές ταχύτητες. Η ροδέλα εμποδίζει την ομαλή κίνηση του ρομπότ κατά της εναλλαγές της κατεύθυνσής του. Μία μπίλια ελεύθερης περιστροφής θα βελτίωνε αισθητά την απόδοσή του.

Μία πιθανή βελτίωση με τα δεδομένα αυτής της κατασκευής θα ήταν η αξιοποίηση ενός δεύτερου PID ή PD ελεγκτή με τη χρήση οπτικών αισθητήρων ταχύτητας στους κινητήρες. Στην περίπτωση αυτή, το ρομπότ θα μπορούσε να κινηθεί ικανοποιητικά ακόμη περισσότερο σε μία ευθεία γραμμή εξαλείφοντας στο μέγιστο δυνατό τις ταλαντώσεις του και μειώνοντας το χρόνο ολοκλήρωσης της πίστας.

Βιβλιογραφία - Πηγές

George Gillard, An Introduction to PID Controllers (Second Edition), διαθέσιμο στο <https://georgegillard.com/resources/documents>, ημερ. ανάκ 7-12-23.

Sheikh Farhan Jibrail, Rakesh Maharana (2013). PID Control of Line Followers, διαθέσιμο στο <https://core.ac.uk/download/pdf/53189911.pdf>, ημερ. ανάκ. 4-1-24.

Proportional–integral–derivative controller, διαθέσιμο στο https://en.wikipedia.org/wiki/Proportional-integral-derivative_controller, ημερ. ανάκ. 15-12-23.

Line Follower Robot (with PID controller), διαθέσιμο στο <https://projecthub.arduino.cc/anova9347/line-follower-robot-with-pid-controller-01813f>, ημερ. ανάκ. 3-1-24.

QTR-8A and QTR-8RC Reflectance Sensor Array User's Guide, διαθέσιμο στο <https://www.pololu.com/docs/0J12/all>, ημερ. ανάκ 10-11-23.

Interface L298N DC Motor Driver Module with Arduino, διαθέσιμο στο <https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial>, ημερ. ανάκ. 1-11-2023.

Παράρτημα Α - Κοστολόγηση υλικών

Υλικά	Προμηθευτής	Ποσότητα	Αξία €
RFID Starter kit με Arduino	https://www.devobox.com	1	45 €
2WD Robot Car Kit για Arduino	https://www.devobox.com	1	9,80 €
QTR-8RC reflectance sensor array - digital - Pololu 961	https://botland.store	1	11,50 €
L298N - two channel motor controller - 12V/2A	https://botland.store	1	3,90 €
Μπαταρίες 18650 Li-Ion INR18650-F1HR 3350mAh	https://botland.store	3	4,90 €
Θήκη μπαταριών 3x 18650	https://botland.store	1	1,50 €
Φορτιστής μπαταριών LCD XTAR MC2 18650	https://botland.store	1	5,50 €
		Σύνολο	91,9 €

Παράρτημα Β: Κώδικες Arduino

Κώδικας 1. Χειροκίνητη βαθμονόμηση αισθητήρα γραμμής

```
#include <QTRSensors.h>

QTRSensors qtr;
const int SensorCount = 8;
int sensorValues[SensorCount];

void setup(){
  // Ορισμός τύπου μονάδας αισθητήρων
  qtr.setTypeRC();
  // Αρχικοποίηση αντικειμένου ορίζοντας τους ψηφιακούς ακροδέκτες εισόδου στον μικροελεγκτή
  qtr.setSensorPins((const uint8_t[]){13, 12, 11, 10, 8, 7, 5, A2 }, SensorCount);
  qtr.setEmitterPin(A3); // Ορισμός ακίδας ελέγχου των IR LED
  Serial.begin(9600);

  Serial.println("Βαθμονόμηση μονάδας αισθητήρων ανάκλασης QTR-8RC....");

  // Το όριο ανάγνωσης για κάθε αισθητήρα είναι 2.5 ms επί 10 αναγνώσεις της μεθόδου calibrate()
  // επί 400 φορές δίνει διάρκεια βαθμονόμησης 10sec.
  for (int i = 0; i < 400; i++){ // Χειροκίνητη βαθμονόμηση
    qtr.calibrate();
  }
  Serial.println("Η βαθμονόμηση ολοκληρώθηκε.");
  Serial.println("-----");
}

void loop(){
  // Διάβασε τις βαθμονομημένες μετρήσεις του αισθητήρα γραμμής και τη θέση της γραμμής( 0 - 7000)
  int position = qtr.readLineBlack(sensorValues);

  // Εκτόπωσε τις μετρήσεις των αισθητήρων ως αριθμούς από 0 έως 1000 και στη συνέχεια τη θέση
  // της γραμμής. 0 σημαίνει μέγιστη ποσότητα ανακλώμενου φωτός (ΑΣΠΡΟ)
  // και 1000 σημαίνει ελάχιστη ποσότητα ανακλώμενου φωτός (ΜΑΥΡΟ)
  for (int i = 0; i < SensorCount; i++){
    Serial.print(sensorValues[i]);
    Serial.print("\t");
  }
}
```



```
Serial.print("Position: ");
Serial.println(position);
delay(250);
}
```

Κώδικας 2. Λειτουργία κουμπιών χειρισμού του ρομπότ

```
const byte leftButton = 4;
const byte rightButton = 0;

byte leftButtonState = 1;
byte rightButtonState = 1;
bool leftButtonFlag = false;

void setup(){
  pinMode(leftButton, INPUT_PULLUP);
  pinMode(rightButton, INPUT_PULLUP);
  Serial.begin(9600);
}

void loop(){
  // Η λειτουργία του αριστερού κουμπιού εκτελείται μόνο μία φορά στην αρχή του προγράμματος
  leftButtonState = digitalRead(leftButton);
  if (leftButtonState == 0){
    Serial.println("Η βαθμονόμηση του αισθητήρα QTR-8RC ολοκληρώθηκε.");
    leftButtonFlag = true;

    // Η λειτουργία του δεξιού κουμπιού εκτελείται κάθε φορά που πατιέται εφόσον έχει πατηθεί μία
    // φορά το αριστερό κουμπί.
    while(leftButtonFlag){
      rightButtonState = digitalRead(rightButton);
      if (rightButtonState == 0){
        Serial.println("Εκτέλεση του αλγορίθμου ακολουθίας γραμμής.....");
        delay(500); // Καθυστέρηση 500 ms για την αγνόηση αναπηδήσεων δεξιού κουμπιού
      }
    }
  }
}
```

Κώδικας 3. Αλγόριθμος ακολουθίας γραμμής Bang-Bang

```
#include <QTRSensors.h>

QTRSensors qtr;
const int SensorCount = 8;
int sensorValues[SensorCount];

#define enA 9
#define in1 19
#define in2 18
#define in3 A0
#define in4 A1
#define enB 6

const byte leftButton = 4;
const byte indicationLed = 3;
const byte rightButton = 2;

byte leftButtonState = 1;
bool leftButtonFlag = false;

byte rightButtonState = 1;
bool rightButtonFlag = false;

void setup(){
  qtr.setTypeRC();
  qtr.setSensorPins((const uint8_t[]){ 13, 12, 11, 10, 8, 7, 5, A2 }, SensorCount);
  qtr.setEmitterPin(A3);
  pinMode(leftButton, INPUT_PULLUP);
  pinMode(indicationLed, OUTPUT);
  pinMode(rightButton, INPUT_PULLUP);
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  pinMode(enB, OUTPUT);
}
```

```

void loop(){
  leftButtonState = digitalRead(leftButton);
  if (leftButtonState == 0){ // Αυτόματη βαθμονόμηση
    digitalWrite(indicationLed, HIGH);
    SpinRight();
    for (int i = 0; i < 40; i++){ // Διάρκεια βαθμονόμησης 1sec
      qtr.calibrate();
      delay(20);
    }
    Stop();
    delay(500);
    SpinLeft();
    for (int i = 0; i < 40; i++){ // Διάρκεια βαθμονόμησης 1sec
      qtr.calibrate();
      delay(20);
    }
    Stop();
    delay(500);

    digitalWrite(indicationLed, LOW);
    rightButtonFlag = false;
    leftButtonFlag = true;

    // Ορισμός φοράς περιστροφής κινητήρων
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
  }

  if (leftButtonFlag == true){
    rightButtonState = digitalRead(rightButton);
    if (rightButtonState == 0){
      rightButtonFlag = true;
    }
  }
}

```

```

if(rightButtonFlag==true){
  FollowLine();
}
}

void FollowLine(){
  // Διάβασε τις βαθμονομημένες τιμές του αισθητήρα και τη θέση της μαύρης γραμμής ( 0 - 7000)
  int position = qtr.readLineBlack(sensorValues);
  for (int i = 0; i < SensorCount; i++){ //
    Serial.print(sensorValues[i]);
    Serial.print('\t');
  }
  // Γωνία στροφής τιμονιού με τιμή κατωφλίου επιθυμητής θέση 3500
  if (position > 3500){ // Η γραμμή βρίσκεται δεξιά
    PivotRight();
  }else{ // Η γραμμή βρίσκεται αριστερά
    PivotLeft();
  }
}

// Επιτόπου στροφή δεξιά
void SpinRight(){
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  analogWrite(enA,80);
  analogWrite(enB,80);
}

// Επιτόπου στροφή αριστερά
void SpinLeft(){
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  analogWrite(enA,80);
  analogWrite(enB,80);
}

```

```

// Κλειστή στροφή δεξιά
void PivotRight(){
    analogWrite(enA,65);
    analogWrite(enB,0);
}

// Κλειστή στροφή αριστερά
void PivotLeft(){
    analogWrite(enA,0);
    analogWrite(enB,65);
}

// Απενεργοποίηση κινητήρων
void Stop(){
    analogWrite(enA,0);
    analogWrite(enB,0);
}

```

Κώδικας 4. Αλγόριθμος ακολουθίας γραμμής PID ελεγκτή

```

#include <QTRSensors.h>

QTRSensors qtr;

const int SensorCount = 8;
int sensorValues[SensorCount];

#define enA 9
#define in1 19
#define in2 18
#define in3 A0
#define in4 A1
#define enB 6

const byte leftButton = 4;
const byte indicationLed = 3;
const byte rightButton = 2;
byte leftButtonState = 1;

```

```

bool leftButtonFlag = false;
byte rightButtonState = 1;
bool rightButtonFlag = false;

int motorSpeedA = 0;
int minMotorSpeedA = 65;
int maxMotorSpeedA = 80;
int motorSpeedB = 0;
int minMotorSpeedB = 65;
int maxMotorSpeedB = 80;
int motorSpeed = 0;

int setpoint = 3500;
int error = 0;
int P = 0;
int I = 0;
int D = 0;
float Kp = 9;
float Ki = 0;
float Kd = 9;
int lastError = 0;

void setup() {
  qtr.setTypeRC();
  qtr.setSensorPins((const uint8_t[]){ 13, 12, 11, 10, 8, 7, 5, A2 }, SensorCount);
  qtr.setEmitterPin(A3);
  pinMode(leftButton, INPUT_PULLUP);
  pinMode(indicationLed, OUTPUT);
  pinMode(rightButton, INPUT_PULLUP);
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  pinMode(enB, OUTPUT);
}

```

```

void loop(){
  leftButtonState = digitalRead(leftButton);
  if (leftButtonState == 0){ // Αυτόματη βαθμονόμηση
    digitalWrite(indicationLed, HIGH);
    SpinRight();
    for (int i = 0; i < 40; i++){ // Διάρκεια βαθμονόμησης 1sec
      qtr.calibrate();
      delay(20);
    }
    Stop();
    delay(500);
    SpinLeft();
    for (int i = 0; i < 40; i++){ // Διάρκεια βαθμονόμησης 1sec
      qtr.calibrate();
      delay(20);
    }
    Stop();
    delay(500);

    digitalWrite(indicationLed, LOW);
    rightButtonFlag = false;
    leftButtonFlag = true;

    // Ορισμός φοράς περιστροφής κινητήρων μπροστά
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
  }

  if (leftButtonFlag == true){
    rightButtonState = digitalRead(rightButton);
    if (rightButtonState == 0){
      rightButtonFlag = true;
    }
  }
}

```

```

if (rightButtonFlag==true){
    FollowLine();
}
}

void FollowLine(){
    // Διάβασε τις βαθμονομημένες τιμές του αισθητήρα και τη θέση της μαύρης γραμμής ( 0 - 7000)
    int position = qtr.readLineBlack(sensorValues);
    error = setpoint - position;
    P = error;
    I = I + error;
    D = error - lastError;
    lastError = error;

    motorSpeed = P * Kp + I * Ki + D * Kd;

    motorSpeedA = minMotorSpeedA - motorSpeed;
    motorSpeedB = minMotorSpeedA + motorSpeed;

    // Κανονικοποίηση τιμών μεγίστου και ελαχίστου ισχύος κινητήρων σε αποδεκτό εύρος τιμών
    if (motorSpeedA > maxMotorSpeedA) {
        motorSpeedA = maxMotorSpeedA;
    }
    if (motorSpeedB > maxMotorSpeedB) {
        motorSpeedB = maxMotorSpeedB;
    }
    if (motorSpeedA < 0) {
        motorSpeedA = 0;
    }
    if (motorSpeedB < 0){
        motorSpeedB = 0;
    }

    // Γωνία στροφής τιμονιού
    analogWrite(enA,motorSpeedA);
    analogWrite(enB,motorSpeedB);
}

```



```
// Επιτόπου στροφή δεξιά
```

```
void SpinRight(){  
    digitalWrite(in1, LOW);  
    digitalWrite(in2, HIGH);  
    digitalWrite(in3, HIGH);  
    digitalWrite(in4, LOW);  
    analogWrite(enA,80);  
    analogWrite(enB,80);  
}
```

```
// Επιτόπου στροφή αριστερά
```

```
void SpinLeft(){  
    digitalWrite(in1, HIGH);  
    digitalWrite(in2, LOW);  
    digitalWrite(in3, LOW);  
    digitalWrite(in4, HIGH);  
    analogWrite(enA,80);  
    analogWrite(enB,80);  
}
```

```
// Απενεργοποίηση κινητήρων
```

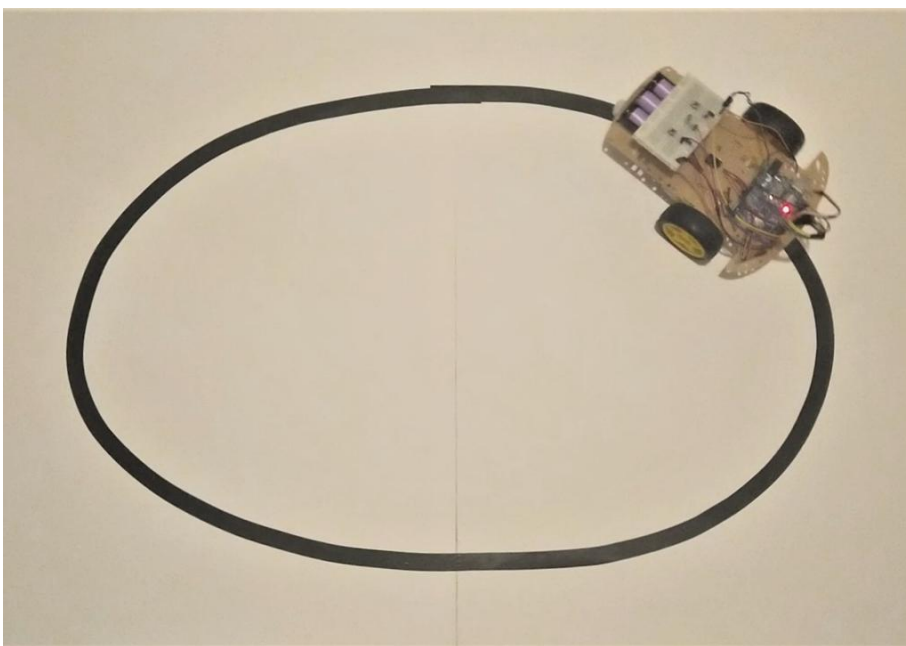
```
void Stop(){  
    analogWrite(enA,0);  
    analogWrite(enB,0);  
}
```

Παράρτημα Γ - Πίστες δοκιμών

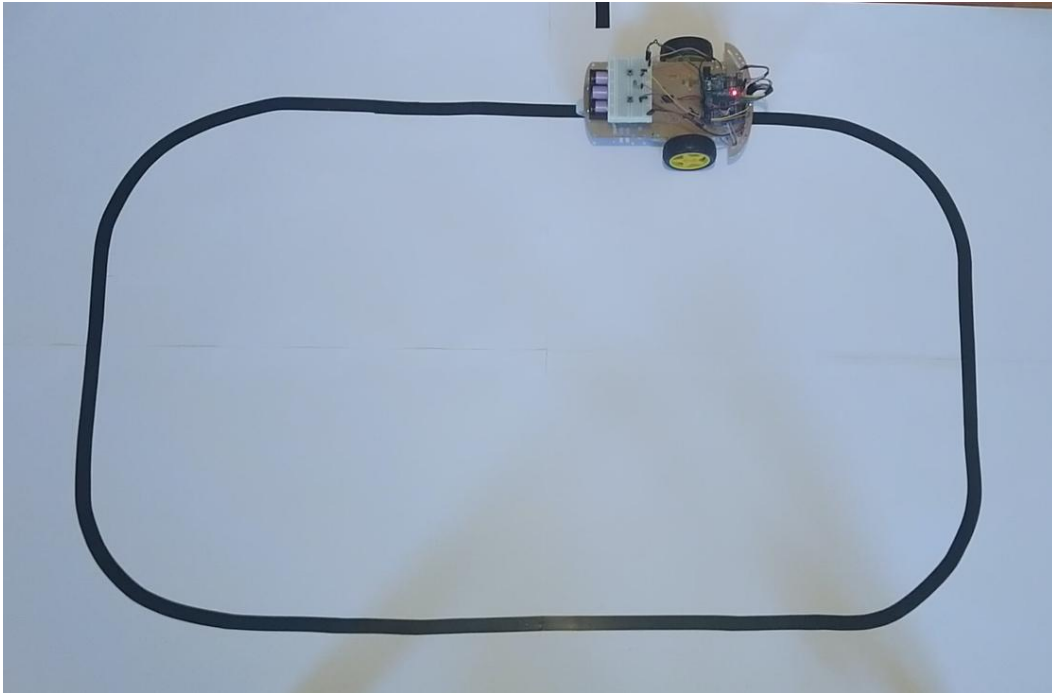
Πίστα 1 - Ευθεία γραμμή



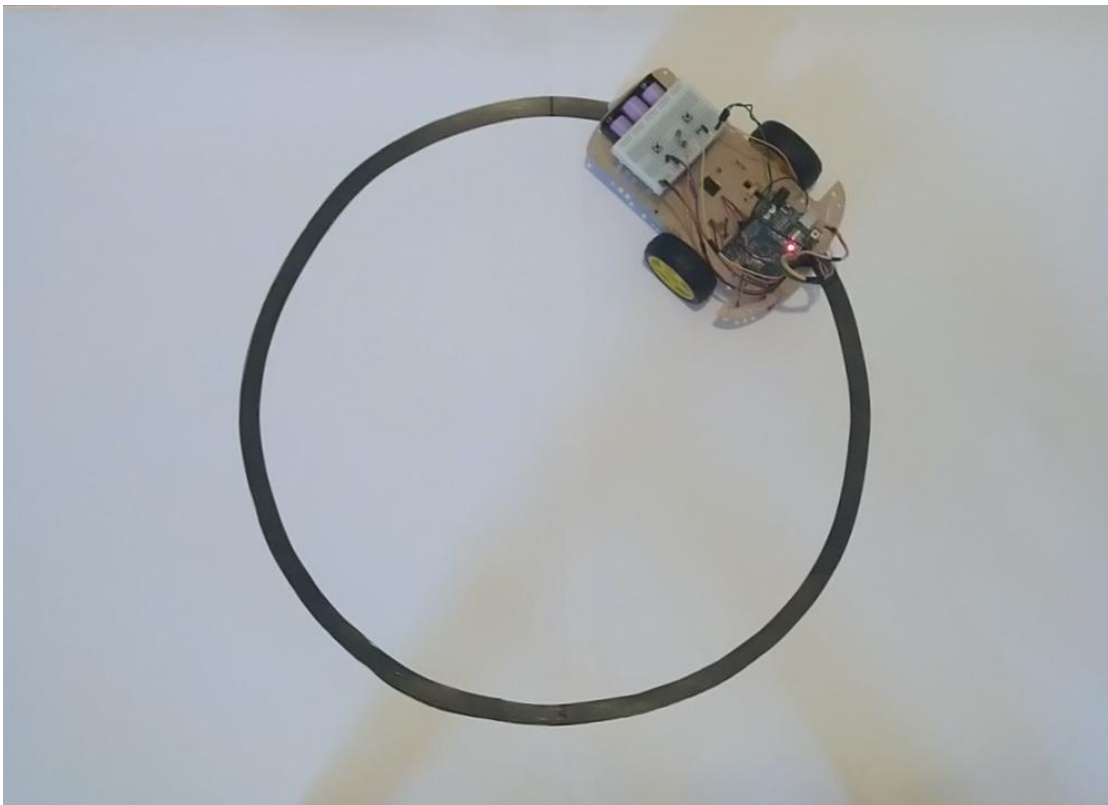
Πίστα 2 - Γραμμή σε σχήμα οβάλ



Πίστα 3 - Γραμμή σε σχήμα ορθογωνίου με στρογγυλεμένες γωνίες



Πίστα 4 - Γραμμή σε σχήμα κύκλου



Πίστα 5 - Γραμμή με μεταβαλλόμενη καμπυλότητα

