



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ
ΜΑΚΕΔΟΝΙΑΣ ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ
ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ



ΕΛΕΓΧΟΣ ΟΧΗΜΑΤΟΣ ΜΕ ARDUINO ΚΑΙ ΧΡΗΣΗ PID ΕΛΕΓΚΤΗ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

του

ΚΛΕΑΝΤ ΛΕΖΙ

Επιβλέπων: Παρίσης Κωνσταντίνος
Καθηγητής

ΚΟΖΑΝΗ/ΟΚΤΩΒΡΙΟΣ/2023



HELLENIC DEMOCRACY
UNIVERSITY OF WESTERN
MACEDONIA FACULTY OF
ENGINEERING
DEPARTMENT OF ELECTRICAL &
COMPUTER ENGINEERING



VEHICLE CONTROL WITH ARDUINO USING PID CONTROLLER

THESIS

KLEANT LEZI

SUPERVISOR: Parisses Constantinos

Professor

KOZANI/OCTOBER /2023



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Πτυχιακή Εργασία με τίτλο “ΕΛΕΓΧΟΣ ΟΧΗΜΑΤΟΣ ΜΕ ARDUINO ΚΑΙ ΧΡΗΣΗ PID ΕΛΕΓΚΤΗ” καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους

του

Τμήματος

κ.

αποτελεί

αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Copyright (C) Ονοματεπώνυμο Φοιτητή & Επιβλέποντα, Έτος, Πόλη

Copyright (C) ΚΛΕΑΝΤ ΛΕΖΙ, ΠΑΡΙΣΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ , 2023, ΚΟΖΑΝΗ

Υπογραφή Φοιτητή: _____

Πρόλογος

Στα πλαίσια σπουδών του Πανεπιστήμιο Δυτικής Μακεδονίας (Πρόγραμμα σπουδών Τ.Ε Ηλεκτρολόγων Μηχανικών) σαν φοιτητής, ο τρόπος σκέψης και λειτουργίας μου άλλαξε κατά πολύ. Η βοήθεια και η γνώση από τα μαθήματα μας έκανε να είμαστε πιο φανατικοί σε αυτό που θέλουμε να επιτύχουμε. Έτσι για πτυχιακή εργασία αποφάσισα να μελετήσω, σχεδιάσω και κατασκευάσω ένα τρίτροχο αυτοκίνητο βασισμένο στο Arduino του οποίου ο έλεγχος επιτεύχθηκε με την χρήση αισθητήρων άλλα και με την χρήση ενός PID ελεγκτή.

Σαν σπουδαστές θα έπρεπε να υποστηρίξουμε όλα όσα έχουμε διδαχτεί από την θεωρία προς την εφαρμογή τους στην πραγματικότητα. Πιστεύω πως με την παρακάτω πτυχιακή εργασία υποστήριξα αν όχι πλήρως αλλά κατά πολύ μεγάλο ποσοστό, ένα μέρος των εργαστηριακών και θεωρητικών μαθημάτων από την εφαρμογή τους στην πράξη.

Περίληψη

Το αντικείμενο της παρούσας πτυχιακής εργασίας είναι η κατασκευή ενός τρίτροχου αυτοκινήτου το οποίο θα κινείται μέσα σε μία οριοθετημένη γραμμή οπού έχουμε ορίσει. Η εργασία βασίστηκε στον μικροελεγκτή Arduino, στο Arduino IDE, και τον PID ελεγκτή.

Στόχος είναι το όχημα να ακολουθεί την πορεία που έχουμε ορίσει. Το όχημα θα πρέπει να ξεκινάει από την αφετηρία και να προχωρήσει σύμφωνα με την κλίση της γραμμής και σε περίπτωση αδιέξοδου το όχημα θα πρέπει να κάνει μια αναστροφή 180° και να συνεχίζει την πορεία μέχρι τον τερματισμό, οπού εκεί θα σταματήσει να κινείται.

Η κατασκευή του συστήματος περιλάμβανε τα ακόλουθα εξαρτήματα. Ένα Arduino (στην συγκεκριμένη πτυχιακή χρησιμοποιήθηκε ένα Arduino UNO), δύο κινητήρες DC 12V, έναν driver shield για την τροφοδοσία και τον έλεγχο των κινητήρων (τις στροφές αλλά και την φορά τους). Επίσης, μία μπαταρία των 9V για την τροφοδοσία του Arduino και δύο μπαταρίες ιόντων λιθίου 18650 οι οποίες είναι συνδεδεμένες σε σειρά για την τροφοδοσία του Driver Shield. Για τον σκελετό του οχήματος χρησιμοποιήθηκε ένα φύλλο αλουμινίου διαστάσεων 14cmx14cmx0.2cm και τέσσερις αισθητήρες υπέρυθρων (IR sensors).

Πριν την τελική υλοποίηση του συστήματος πραγματοποιείται μια λεπτομερής ανάλυση του PID ελεγκτή, των κυκλωμάτων και της αρχιτεκτονικής του Arduino που χρησιμοποιήθηκε, καθώς και όλα τα εξαρτήματα που χρησιμοποιήθηκαν για την κατασκευή.

Λέξεις-κλειδιά: arduino, PID ελεγκτής, ακολουθητής γραμμής, κινητήρας συνεχούς ρεύματος, αισθητήρας υπέρυθρων

Abstract

The subject of this thesis is the construction of a three-wheeled vehicle that will move along a predefined path. The current work is based on the Arduino microcontroller, the Arduino IDE, and the PID Controller.

The objective is for the vehicle to follow the path we have defined. The vehicle should start from the starting point and move in accordance with the slope of the path. In case of a dead end, the vehicle should perform a 180 degree turn around and continue along the path until it reaches the endpoint, where it will stop moving.

The construction of the system included the following components: an Arduino (in this specific thesis, an Arduino UNO was used), two 12V DC motors, a Driver Shield for supplying power and controlling the motors (controlling their speed and direction). Additionally, one 9V battery was used to power the Arduino, and two lithium-ion 18650 batteries, which were connected in series to power the Driver Shield. For the vehicle's chassis, a piece of aluminum measuring 14cmx14cmx0.2cm was used, along with four infrared sensors (IR sensors).

Before the final implementation of the system, a detailed analysis of the PID controller, the circuits, and the architecture of the Arduino used, as well as all the components used in the construction, is carried out.

Keywords: Arduino, PID controller, Line follower, DC motor, IR sensor

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον υποψήφιο διδάκτορα Ιωάννη Μόσχο και τον υπεύθυνο καθηγητή Κωνσταντίνο Παρίση για την εκπόνηση της πτυχιακής μου εργασίας, αλλά και για την καθοδήγηση και βοήθεια τους σε αυτήν. Επίσης, θα ήθελα να ευχαριστώ την οικογένεια μου για την στήριξη τους όλα αυτά τα χρόνια αλλά και τους φίλους συμφοιτητές για τα υπέροχα χρόνια που περάσαμε μαζί.

Περιεχόμενα

Πρόλογος	- 1 -
Περίληψη	- 2 -
Abstract	3
Ευχαριστίες	4
Κατάλογος Εικόνων	7
Κατάλογος Πινάκων	8
Κεφάλαιο 1: Εισαγωγή	9
1. Αντικείμενο της πτυχιακής	9
1.1 Τα ρομπότ στη ζωή μας	9
1.2 Οργάνωση του τόμου	9
Κεφάλαιο 2: Μικροελεγκτής Arduino	10
2.1 Οι μικροελεγκτές και εφαρμογές τους	10
2.2 Arduino	10
2.2.1 Γιατί να επιλέξουμε το Arduino;	11
2.2.2 Επίσημες πλακέτες Arduino	11
2.2.3 Χαρακτηριστικά του Arduino Uno	13
Κεφάλαιο 3: Arduino IDE	16
3.1 Τι είναι το Arduino IDE;	16
Κεφάλαιο 4: Ο ελεγκτής PID	18
Κεφάλαιο 5: Το υλικό μέρος του συστήματος	19
5.1 Motor shield L298N	19
5.2 DC Motor	20
5.3 Αισθητήρες υπέρυθρων	21
5.4 Jumper wires	22
5.5 Οι τροχοί στην ρομποτική	22
5.6 Μπαταρίες ιόντων λιθίου	24
5.7 Υλοποίηση συστήματος	25
Κεφάλαιο 6: Εύρεση παραμέτρων ελεγκτή	30
Κεφάλαιο 7: Συνολικό κόστος υλοποίησης	32
Κεφάλαιο 8: Μελλοντικές επεκτάσεις	33
Βιβλιογραφία	34

Κατάλογος Εικόνων

Εικόνα 1. Μικροελεγκτής ATmega328[16].....	10
Εικόνα 2. Λογότυπο Arduino[17]	10
Εικόνα 3. Περιγραφή Arduino[18]	11
Εικόνα 4. Μοντέλα Arduino[4], [19]	12
Εικόνα 5. Τροφοδοσίες Arduino[13], [20]	13
Εικόνα 6. Ψηφιακά Pins.....	14
Εικόνα 7. Αναλογικά Pins	15
Εικόνα 8. Arduino IDE[12], [21]	16
Εικόνα 9. Χρήση του PID στον κώδικα	18
Εικόνα 10. Υλοποίηση του κυκλώματος	19
Εικόνα 11. Motor driver και τεχνικά χαρακτηριστικά του Motor Driver [22].....	20
Εικόνα 12. DC κινητήρας[23].....	20
Εικόνα 13. Αισθητήρες υπέρυθρων[14], [24]	21
Εικόνα 14. Επεξήγηση των στοιχείων του IR sensor [15], [25].....	22
Εικόνα 15. Jumper wires[26].....	22
Εικόνα 16. Omni wheels[27]	23
Εικόνα 17. Non-Skid Wheels[28].....	23
Εικόνα 18. High-Traction wheels[29].....	24
Εικόνα 19. Μπαταρίες ιόντων λιθίου[30].....	25
Εικόνα 20. Τοποθέτηση των κινητήρων στον σκελετό.....	25
Εικόνα 21. Εγκατάσταση κεντρικού τροχού.....	26
Εικόνα 22. Εξαρτήματα και καλώδια.....	26
Εικόνα 23. Εγκατάσταση του Arduino και του Driver shield.....	27
Εικόνα 24. Ολοκλήρωση του οχήματος	27
Εικόνα 25. Ολοκλήρωση του οχήματος.....	28
Εικόνα 26. Ολοκλήρωση του οχήματος.....	29

Κατάλογος Πινάκων

Πίνακας 1: Χαρακτηριστικά Arduino.....	13
Πίνακας 2: Προδιαγραφές DC κινητήρα [23].....	21
Πίνακας 3: Τιμές του PID.....	30
Πίνακας 4: Θεωρητικές τιμές του PID με την μέθοδο Ziegler-Nichols[31].....	31
Πίνακας 5: Κόστος Υλικών	32

Κεφάλαιο 1: Εισαγωγή

1. Αντικείμενο της πτυχιακής

Το αντικείμενο της παρούσας πτυχιακής εργασίας είναι η κατασκευή ενός τρίτροχου αυτοκινήτου ρομπότ το οποίο θα κινείται μέσα σε μία οριοθετημένη γραμμή οπού έχουμε ορίσει.

1.1 Τα ρομπότ στη ζωή μας

Τα ρομπότ έχουν εισβάλει στην ζωή μας σε πολλούς τομείς και έχουν επηρεάσει την καθημερινότητα μας με πολλούς τρόπους. Μερικοί τομείς οπού τα ρομπότ έχουν έναν σημαντικό ρόλο είναι η βιομηχανία (τα ρομπότ αυτά μπορούν να εκτελούν επαναλαμβανόμενες εργασίες με ακρίβεια και ταχύτητα), στον τομέα της υγειονομικής περίθαλψης οπού τα ρομπότ χρησιμοποιούνται για χειρουργικές επεμβάσεις αλλά και ρομποτικές προσαρμοστικές συσκευές για την βοήθεια ανθρώπων με αναπηρίες. Στην διαστημική έρευνα τα ρομπότ χρησιμοποιούνται για την εξερεύνηση άλλων πλανητών αλλά και σε δορυφόρους, ένας άλλος τομέας είναι η εκπαίδευση οπού εκεί η ρομποτική χρησιμοποιείται για την εκμάθηση του προγραμματισμού αλλά και την προώθηση της επιστημονικής και μηχανικής γνώσης σε παιδιά αλλά και ενήλικες. Τέλος, τα ρομπότ χρησιμοποιούνται στην ψυχαγωγία όπωρ ρομποτικά παιχνίδια και ρομποτικές εκθέσεις.

1.2 Οργάνωση του τόμου

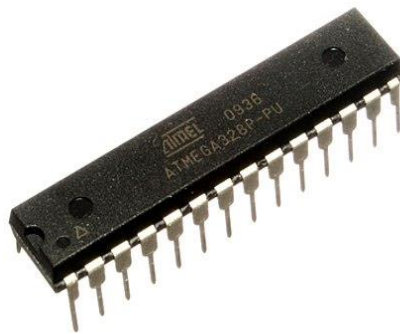
Στο κεφάλαιο 1 παρουσιάζεται το αντικείμενο της πτυχιακής. Στο κεφάλαιο 2 γίνεται μία σύντομη ανάλυση των μικροελεγκτών, του Arduino, καθώς και ο λόγος για τον οπού θα επιλέχθηκε ο μικροελεγκτής Arduino. Στο κεφάλαιο 3 περιγράφεται το Arduino IDE. Στο επόμενο κεφάλαιο αναλύεται ο PID ελεγκτής. Στο κεφάλαιο 5 παρουσιάζονται και αναλύονται τα υλικά που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος.

Κεφάλαιο 2: Μικροελεγκτής Arduino

Σε αυτό το κεφάλαιο θα παρουσιαστούν οι μικροελεγκτές και το Arduino. Θα αναλυθεί ο τρόπος λειτουργίας αλλά και οι εφαρμογές τους.

2.1 Οι μικροελεγκτές και εφαρμογές τους

Σε αυτήν την ενότητα θα γίνει μια εισαγωγή στους μικροελεγκτές και συγκεκριμένα στον ATmega328. Ο ATmega328 είναι ένας προηγμένος μικροελεγκτής εικονικού RISC (υπολογιστής περιορισμένου συνόλου εντολών), ο οποίος υποστηρίζει επεξεργασία δεδομένων 8-bit. Ο ATmega328 είναι ο μικροελεγκτής που χρησιμοποιείται σε βασικές πλακέτες Arduino.



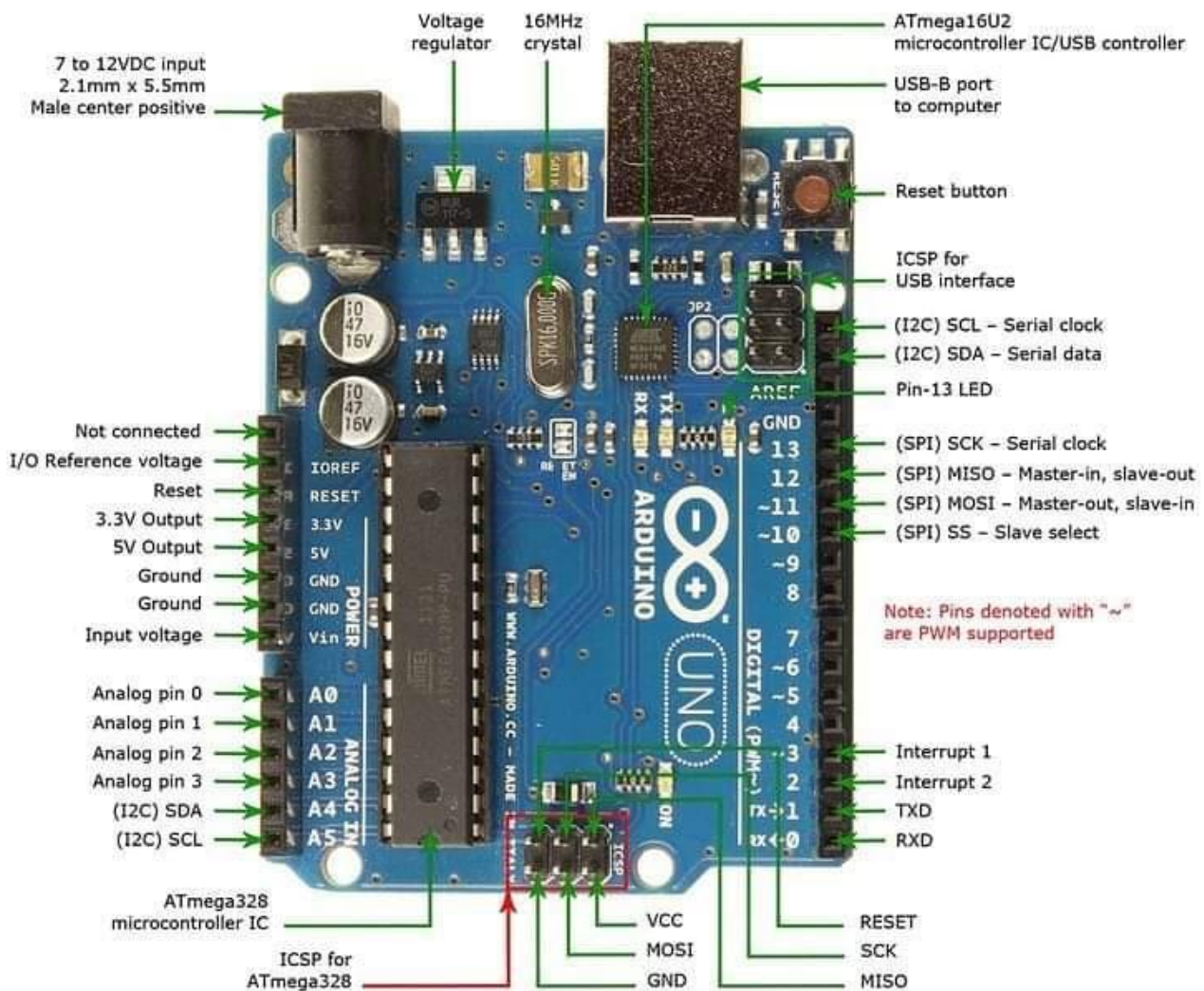
Εικόνα 1. Μικροελεγκτής ATmega328[16].

2.2 Arduino



Εικόνα 2. Λογότυπο Arduino[17].

Το Arduino είναι ένας single-board μικροελεγκτής, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή με εισόδους και εξόδους. Η πλακέτα μπορεί να προγραμματιστεί με την γλώσσα Wiring. Ουσιαστικά πρόκειται για την γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες υλοποιημένες επίσης στην C++. Οι δυνατότητες που μπορεί να προσφέρει ένα Arduino περιορίζονται από την δημιουργικότητα και φαντασία του προγραμματιστή. Μπορεί να χρησιμοποιηθεί σε εφαρμογές ρομποτικής και αυτοματισμούς, αλλά και για την λήψη πληροφοριών από διάφορους τύπους αισθητήρων, όπως θερμοκρασίας, υγρασίας, υπέρυθρων, την σειριακή επικοινωνία μεταξύ Arduino και υπολογιστή χρησιμοποιώντας γλώσσες προγραμματισμού, όπως Python και Java, την αναπαραγωγή και λήψη ήχων. Στην εικόνα 3 παρουσιάζονται αναλυτικά οι εισοδοί, οι εξοδοί, τα ολοκληρωμένα, καθώς και οι τροφοδοσίες του Arduino Uno.



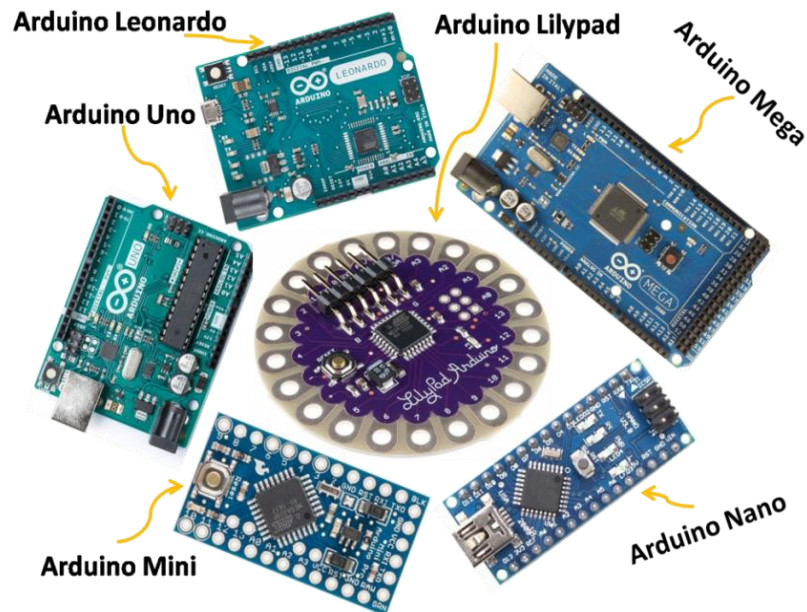
Εικόνα 3. Περιγραφή Arduino[18].

2.2.1 Γιατί να επιλέξουμε το Arduino;

Το Arduino χάρη στην απλή αρχιτεκτονική και το εύχρηστο γραφικό περιβάλλον για προγραμματισμό έχει χρησιμοποιηθεί σε χιλιάδες διαφορετικά προγράμματα και εφαρμογές. Το λογισμικό του Arduino είναι εύκολο στην χρήση για αρχάριους αλλά και αρκετά ευπροσάρμοστο για προχωρημένους χρήστες. Το Arduino μπορεί να χρησιμοποιηθεί στον προγραμματισμό και τη ρομποτική.

2.2.2 Επίσημες πλακέτες Arduino

Τα μοντέλα Arduino που κυκλοφορούν στην αγορά παρουσιάζονται συνοπτικά στην εικόνα 4. Στην παρούσα πτυχιακή εργασία, επιλέχτηκε το μοντέλο Arduino Uno, καθώς καλύπτει όλες τις ανάγκες της εργασίας. Συγκεκριμένα, οι ακροδέκτες (pins) που διαθέτει είναι επαρκείς για την υλοποίηση όλων των λειτουργιών που απαιτούνται.



Εικόνα 4. Μοντέλα Arduino[4], [19].

Κάθε μοντέλο Arduino ξεχωρίζει ανάλογα με το μέγεθος και τη χρήση του. Τα διάφορα μοντέλα Arduino που είναι διαθέσιμα στην αγορά παρουσιάζονται συνοπτικά παρακάτω με τα βασικά χαρακτηριστικά τους:

- **Arduino Leonardo**

Το Arduino Leonardo κυκλοφόρησε το 2012, χρησιμοποιεί τον μικροελεγκτή ATmega32u4, διαθέτει 20 ψηφιακούς ακροδέκτες και 12 αναλογικούς ακροδέκτες. Η ταχύτητα του ρολογιού είναι 16MHz και η διαθέσιμη μνήμη είναι 32KB

- **Arduino Mega**

Το Arduino Mega κυκλοφόρησε το 2010, χρησιμοποιεί τον μικροελεγκτή ATmega2650, διαθέτει 54 ψηφιακούς ακροδέκτες και 16 αναλογικούς ακροδέκτες. Η ταχύτητα ρολογιού είναι στα 16MHz και μνήμη 256KB.

- **Arduino Lilypad**

Το Arduino Lilypad κυκλοφόρησε το 2009 χρησιμοποιεί τον μικροελεγκτή ATmega168v, διαθέτει 14 ψηφιακούς ακροδέκτες. Η ταχύτητα του ρολογιού είναι 8MHz και η διαθέσιμη μνήμη είναι 16KB.

- **Arduino Mini**

Το Arduino Mini είναι διαθέσιμο σε εκδόσεις των 3.3V και 5V, χρησιμοποιεί τον μικροελεγκτή ATmega328P. Διαθέτει 14 ψηφιακές ακίδες εισόδου/εξόδου από τις οποίες οι 6 μπορούν να χρησιμοποιηθούν ως έξοδοι PWM και 6 αναλογικές εισόδους.

- **Arduino Nano**

Το Arduino Nano κυκλοφόρησε το 2008, χρησιμοποιεί τον μικροελεγκτή ATmega328, διαθέτει 22 ψηφιακά pins και 6 κανάλια PWM. Η ταχύτητα του ρολογιού είναι στα 16MHz και η διαθέσιμη μνήμη είναι 32KB.

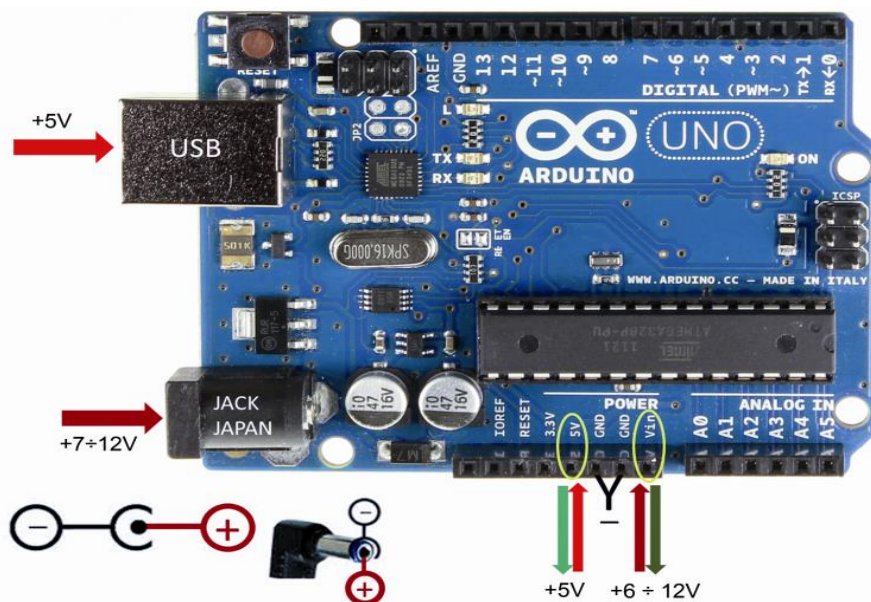
2.2.3 Χαρακτηριστικά του Arduino Uno

Στον πίνακα 1 αναλύονται τα χαρακτηριστικά του Arduino Uno.

Πίνακας 1: Χαρακτηριστικά Arduino Uno.

Μικροελεγκτής	ATmega328
Τάση λειτουργίας	5V
Τάση εισόδου	7-12V
Όρια τάσης εισόδου	6-20V
Ψηφιακοί ακροδέκτες I/O	14 (6 PWM έξοδοι)
Αναλογικοί ακροδέκτες εισόδου	6
Ισχύς συνεχόμενου ρεύματος για ακροδέκτη τάσης 3.3V	50mA
Ισχύς συνεχόμενου ρεύματος ανά ακροδέκτη	40mA
Ταχύτητα ρολογιού	16MHz
Μνήμη EEPROM	1KB (ATmega328)
Μνήμη SRAM	2KB (ATmega328)
Μνήμη flash	32KB (ATmega328)

Το Arduino μπορεί να τροφοδοτηθεί με ρεύμα από εξωτερική τροφοδοσία που παρέχεται μέσω μίας υποδοχής που βρίσκεται στην κάτω αριστερή γωνία, επίσης μπορεί να τροφοδοτηθεί με ρεύμα από τον υπολογιστή μέσω USB. Για την αποφυγή προβλημάτων, η εξωτερική πηγή τροφοδοσίας θα πρέπει να κυμαίνεται μεταξύ 7V και 12V. Η εικόνα 5 παρουσιάζει τις εισόδους και εξόδους τροφοδοσίας του Arduino Uno.



Εικόνα 5. Τροφοδοσίες Arduino[13], [20].

Οι ακροδέκτες τροφοδοσίας του Arduino Uno είναι οι εξής:

- 5V: Η τάση που παρέχεται σε αυτόν τον ακροδέκτη προέρχεται είτε από την τάση 5V που λαμβάνετε από τη θύρα USB, είτε από τη ρυθμιζόμενη τάση που παρέχεται μέσω του ακροδέκτη Vin.

- 3.3V: Η τάση αυτή παρέχεται από το ολοκληρωμένο FTDI.
- GND: Η γείωση του Arduino Uno.
- Vin: Η τάση εισόδου της πλακέτας, όταν χρησιμοποιεί εξωτερική πηγή ενέργειας.

Όπως προαναφέρθηκε, ο μικροεπεξεργαστής ATmega328 διαθέτει τρεις κύριες ομάδες μνήμης. Διαθέτει την flash memory, στην οποία αποθηκεύονται τα sketch. Η μνήμη EEPROM που χρησιμοποιείται από τους προγραμματιστές για την αποθήκευση πληροφοριών, και η μνήμη SRAM όπου δημιουργούνται και αποθηκεύονται οι μεταβλητές κατά την εκτέλεση του προγράμματος..

- **SRAM Memory 2KB:** Είναι η ωφέλιμη μνήμη που μπορεί να χρησιμοποιηθεί από τα προγράμματα για την αποθήκευση μεταβλητών. Σε περίπτωση διακοπής ρεύματος η πατηθεί το κουμπί επανεκκίνησης η μνήμη χάνει τα δεδομένα της.
- **FLASH Memory 32KB:** Το firmware του Arduino χρησιμοποιεί 2KB τα οποία έχει προεγκαταστήσει ο κατασκευαστής του. Το firmware είναι αναγκαίο για την εγκατάσταση των προγραμμάτων στον μικροελεγκτή μέσω της θύρας USB. Τα υπόλοιπα 30KB της μνήμης flash χρησιμοποιούνται για την αποθήκευση των προγραμμάτων, αφού πρώτα μεταγλωττιστούν από τον υπολογιστή. Σημαντικό είναι ότι η μνήμη flash δεν χάνει τα δεδομένα της από έλλειψη τροφοδοσίας ή επανεκκίνησης.
- **EEPROM Memory 1KB:** Η μνήμη αυτή χρησιμοποιείται για την εγγραφή ή την ανάγνωση των δεδομένων από τα προγράμματα. Σε αντίθεση με την μνήμη SRAM η μνήμη αυτή δεν χάνει τα δεδομένα της από έλλειψη τροφοδοσίας ή επανεκκίνησης.

Στην συνέχεια αναλύονται οι ψηφιακοί και αναλογικοί ακροδέκτες του Arduino UNO. Ψηφιακοί ακροδέκτες Arduino Uno:

- Ακροδέκτες 0 (RX) και 1 (TX): Λειτουργούν ως σειριακή θύρα όταν το πρόγραμμα ενεργοποιεί την σειριακή θύρα, με αποτέλεσμα όταν το πρόγραμμα στέλνει δεδομένα στην σειριακή θύρα αυτά προωθούνται και στην θύρα USB μέσω του ελεγκτή Serial-over-USB αλλά και στον ακροδέκτη 0 για να διαβάσει ενδεχομένως μία άλλη συσκευή.
- Ακροδέκτες 2 και 3: Λειτουργούν ως εξωτερικά interrupt (0 και 1) επίσης ρυθμίζονται μέσα από το πρόγραμμα ώστε να λειτουργούν αποκλειστικά ως ψηφιακές εισοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, με αποτέλεσμα η κανονική ροή του προγράμματος σταματάει με αποτέλεσμα να εκτελείται μία συγκεκριμένη συνάρτηση.
- Ακροδέκτες 4, 5, 6, 9, 10, 11: Έχουν την δυνατότητα να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι με το σύστημα PWM(Pulse Width Modulation).



Εικόνα 6. Ψηφιακά pins.

Στην απέναντί πλευρά του Arduino θα διακρίνουμε την σήμανση ANALOG IN όπως φαίνεται και στην εικόνα 7. Υπάρχουν έξι αναλογικοί ακροδέκτες (A0, A1, A2, A3, A4, A5) στους οποίους η τάση αναφοράς μπορεί να ρυθμιστεί με μία εντολή μεταξύ 2V και 5V. Αυτό επιτρέπει την προσαρμογή της τάσης αναφοράς για την ανάγνωση των αναλογικών σημάτων ανάλογα με τις ανάγκες του χρήστη.



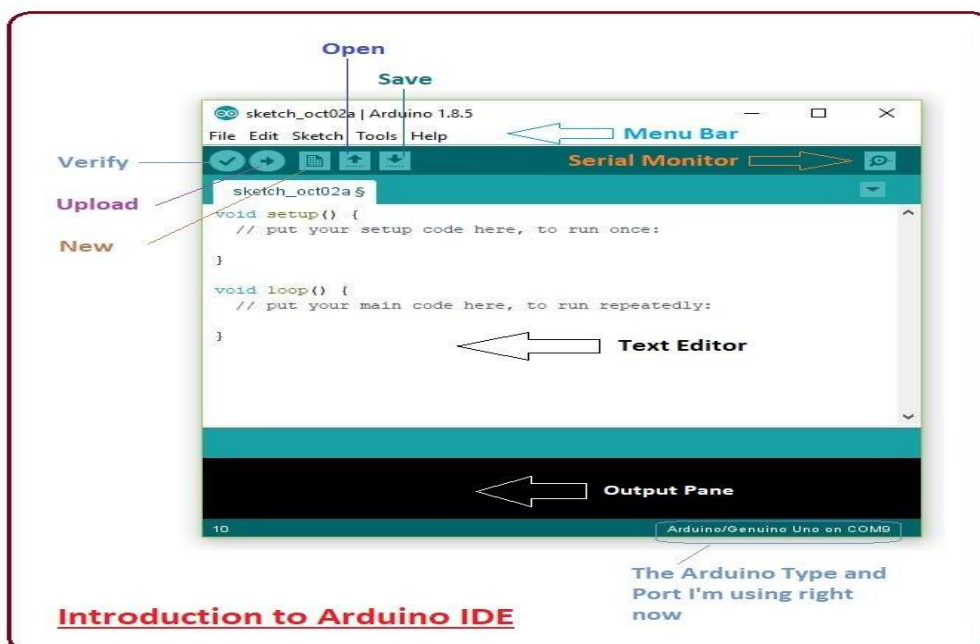
Εικόνα 7. Αναλογικά pins.

Κεφάλαιο 3: Arduino IDE

Σε αυτό το κεφάλαιο, θα αναλυθεί το Arduino IDE, το περιβάλλον ανάπτυξης προγραμμάτων όπου δημιουργείται ο κώδικας για τις πλακέτες Arduino

3.1 Τι είναι το Arduino IDE;

Το Arduino IDE (Integrated Development Environment) είναι ένα ενσωματωμένο περιβάλλον ανάπτυξης που χρησιμοποιείται για τον προγραμματισμό και την ανάπτυξη εφαρμογών που βασίζονται στις πλακέτες Arduino. Το Arduino IDE παρέχει ένα φιλικό περιβάλλον για τον προγραμματισμό των μικροελεγκτών Arduino. Περιλαμβάνει τα απαραίτητα εργαλεία για τον σχεδιασμό την ανάπτυξη και την μεταφόρτωση του κώδικα στους μικροελεγκτές.



Εικόνα 8. Arduino IDE[12], [21].

Η γλώσσα του Arduino βασίζεται στην γλώσσα Wiring που είναι μια παραλλαγή της C/C++ για τους μικροελεγκτές αρχιτεκτονικής AVR, όπως ο ATmega. Υποστηρίζει όλες τις βασικές δομές της γλώσσας C καθώς και μερικά χαρακτηριστικά της C++ και ως βασική βιβλιοθήκη χρησιμοποιείται η AVR libc.

Η γλώσσα του Arduino λόγω της καταγωγής της από την C μπορεί να χρησιμοποιήσει τις ίδιες βασικές εντολές και συναρτήσεις με την ίδια σύνταξη, και τους ίδιους τύπους δεδομένων αλλά και τελεστές όπως και στην C. Πέρα από αυτές, υπάρχουν κάποιες ειδικές εντολές σταθερές και συναρτήσεις που βοηθούν για την διαχείριση του hardware του Arduino.

Το περιβάλλον ανάπτυξης του Arduino περιέχει ένα πρόγραμμα επεξεργασίας κειμένων όπου γίνεται η σύνταξη του κώδικα. Επιπλέον, περιέχει την περιοχή στην οποία εμφανίζονται τα μηνύματα, η κονσόλα κειμένου και μια γραμμή σε μορφή κουμπιών. Στην εικόνα 8 παρουσιάζονται τα εργαλεία ανάπτυξης υπό μορφή κουμπιών αλλά και το ίδιο το περιβάλλον. Ειδικότερα:

- Verify: Ελέγχει για συντακτικά λάθη στον κώδικα.

- Upload: Γίνεται η μεταγλώττιση του κώδικα και η μεταφόρτωση στο Arduino.
- New: Δημιουργεί ένα νέο sketch.
- Serial Monitor: Ανοίγει την σειριακή οθόνη.
- Open: Μενού με όλα τα sketch και κάνοντας κλικ σε ένα από αυτά θα ανοίξει αυτόματα στο υπάρχων παράθυρο.
- Save: Γίνεται η αποθήκευση του sketch.

Τα προγράμματα του Arduino διαιρούνται σε τρία μέρη:

- Την δομή (structure).
- Τις τιμές (values).
- Τις συναρτήσεις (functions).

Κεφάλαιο 4: Ο ελεγκτής PID

Ο PID είναι ένας ελεγκτής που χρησιμοποιείται για τον έλεγχο και τη ρύθμιση συστημάτων αυτοματισμού, όπως ρομποτικά συστήματα, διαδικασίες βιομηχανικής παραγωγής, αυτοκίνητα, αεροσκάφη, κλιματισμό, και πολλά άλλα. Ο ελεγκτής PID αντιπροσωπεύει τρεις βασικές παραμέτρους ελέγχου: τον αναλογικό όρο (P), τον ολοκληρωτικό όρο (I), και τον διαφορικό όρο (D). Καθένας από αυτούς τους όρους συμβάλλει στον υπολογισμό του σήματος εξόδου του ελεγκτή. Ειδικότερα ο κάθε όρος συμβάλλει ως εξής:

- Αναλογικός όρος (P): Ο αναλογικός όρος αντιδρά στην τρέχουσα απόκλιση μεταξύ της επιθυμητής τιμής και της πραγματικής τιμής του συστήματος.
- Ολοκληρωτικός όρος (I): Ο ολοκληρωτικός όρος λαμβάνει υπόψη το ολοκλήρωμα της απόκλισης (σφάλμα μεταξύ εισόδου και εξόδου) με την πάροδο του χρόνου. Αυτός ο όρος χρησιμοποιείται για να αντιμετωπίσει τυχόν αποκλίσεις που μπορεί να παραμείνουν μετά τη δράση του αναλογικού όρου.
- Διαφορικός όρος (D): Ο διαφορικός όρος προσπαθεί να προβλέψει τη μελλοντική τάση της απόκλισης με βάση το πόσο γρήγορα αλλάζει την τρέχουσα στιγμή.

Οι τρεις αυτοί όροι συνδυάζονται με κατάλληλους συντελεστές για να δημιουργήσουν την έξοδο ελέγχου που θα επιτρέψει στο σύστημα να συγκλίνει στην επιθυμητή τιμή με τρόπο γρήγορο και σταθερό ικανοποιώντας τις προδιαγραφές σχεδίασης. Η σωστή ρύθμιση των παραμέτρων του PID ελεγκτή είναι σημαντική για την αποτελεσματική λειτουργία του ελεγκτικού συστήματος και την εξάλειψη διαταραχών που μπορεί να παρουσιαστούν. Ο μαθηματικός τύπος του PID ελεγκτή είναι ο ακόλουθος:

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (1)$$

```
// PID Constants
float Kp = 25;
float Ki = 1;
float Kd = 15;

void calculate_pid()
{
    P = error;
    I = I + previous_I;
    D = error - previous_error;

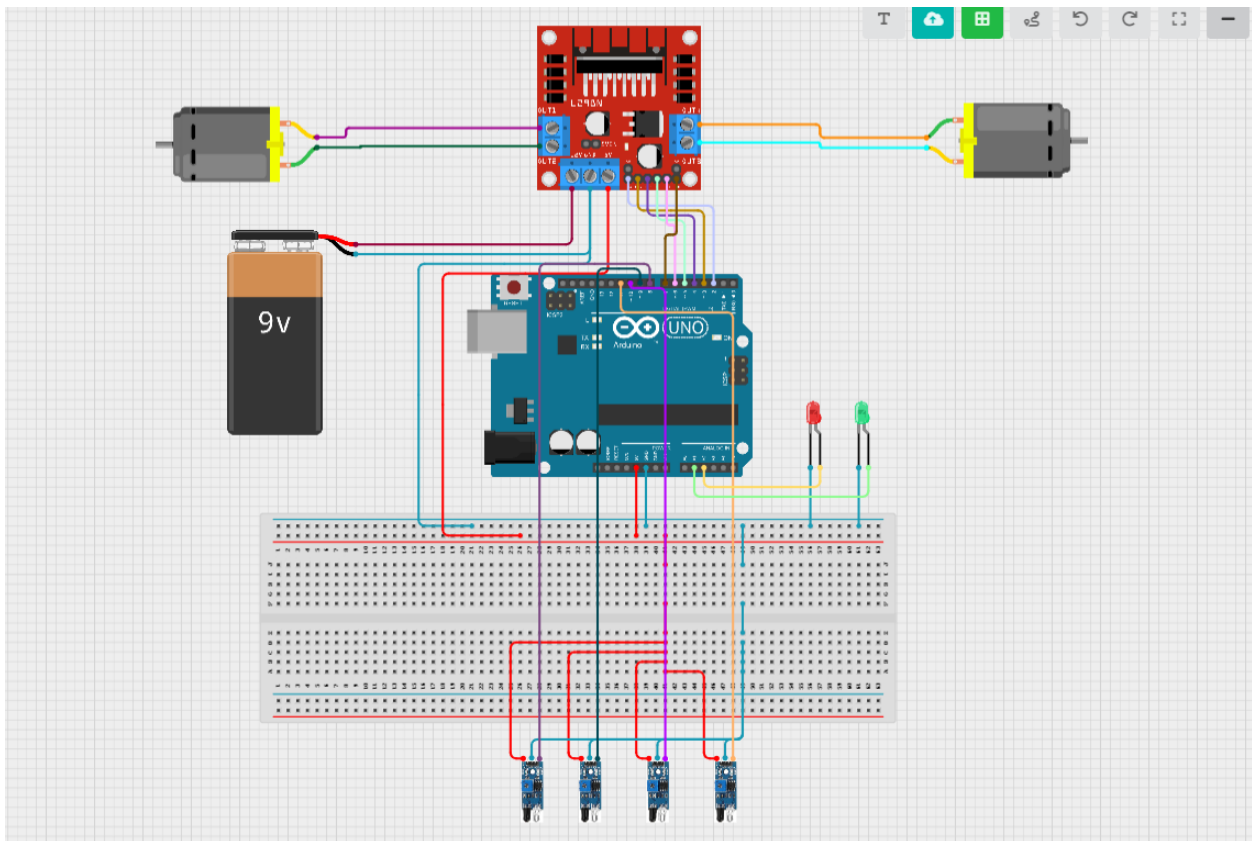
    PID_value = (Kp * P) + (Ki * I) + (Kd * D);

    previous_I = I;
    previous_error = error;
}
```

Εικόνα 9. Χρήση του PID ελεγκτή στον κώδικα.

Κεφάλαιο 5: Το υλικό μέρος του συστήματος

Το υλικό του συστήματος στήθηκε γύρω από τον μικροελεγκτή Arduino. Αυτός ήταν ο πυρήνας στον οποίο συνδέθηκαν τα υπόλοιπα εξαρτήματα του συστήματος. Αρχικά πάνω από το Arduino συνδέθηκε το motor shield. Η πλακέτα αυτή είναι υπεύθυνη για την οδήγηση των κινητήρων. Στην συνέχεια συνδέθηκαν οι πόλοι γείωσης και 5V στο Arduino, ενώ τα pins 7, 6, 5, 4, 3, 2 συνδέθηκαν στα αντίστοιχα pins του motor driver (οδηγός κινητήρα), ENB, IN4, IN3, IN2, IN1, ENA. Πρέπει να επισημανθεί ότι το motor shield το τροφοδοτούμε με δύο μπαταρίες ιόντων λιθίου συνδεδεμένες σε σειρά για να επιτύχουμε τάση ίση με 8V. Έπειτα πρέπει να συνδεθούν οι κινητήρες στις κλέμες του motor shield. Για τον δεξιό κινητήρα έχουμε την κλέμα OUT1, OUT2 και αντίστοιχα για τον αριστερό κινητήρα στην κλέμα OUT3, OUT4. Στην εικόνα 10 βλέπουμε τον motor driver με όλες τις εισόδους, εξόδους και τάσεις τροφοδοσίας. Για την σχεδίαση του κυκλώματος χρησιμοποιήθηκε το πρόγραμμα Cirkuit Designer.



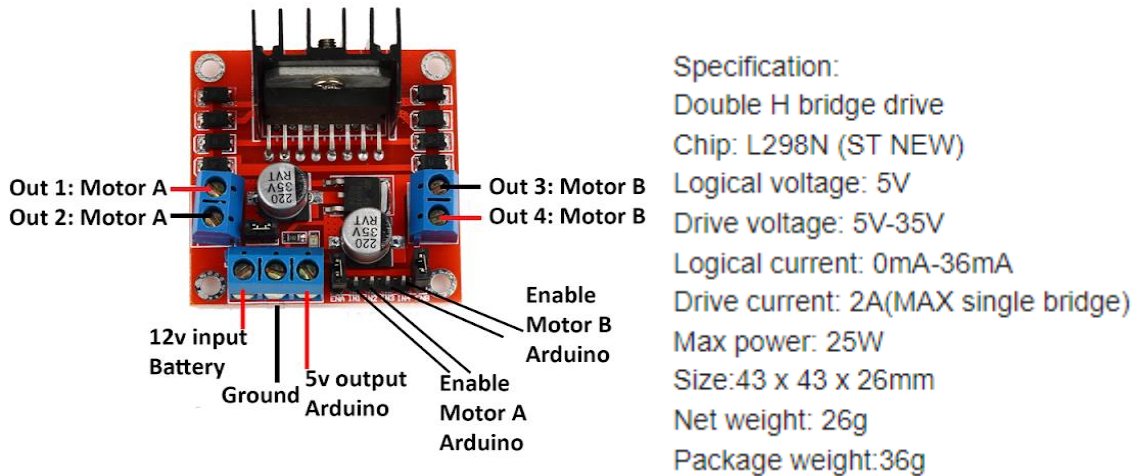
Εικόνα 10. Υλοποίηση του κυκλώματος.

5.1 Motor shield L298N

Το L298N (εικόνα 11) είναι ένας διπλός οδηγός κινητήρων που χρησιμοποιείται συχνά σε έργα ηλεκτρονικής και ρομποτικής. Το motor shield L298N είναι μία πλακέτα που περιλαμβάνει τον L298N οδηγό κινητήρων και παρέχει εύκολη σύνδεση και έλεγχο κινητήρων με ένα Arduino ή

άλλο μικροελεγκτή. Ο L298N επιτρέπει τον έλεγχο δύο DC κινητήρων ή έναν βηματικό κινητήρα με την χρήση ψηφιακών σημάτων από τον μικροελεγκτή. Μπορούμε να τον συναντήσουμε σε πολλές εφαρμογές, όπως ρομποτικά αυτοκίνητα, ρομπότ, μηχανές CNC και άλλα πολλά.

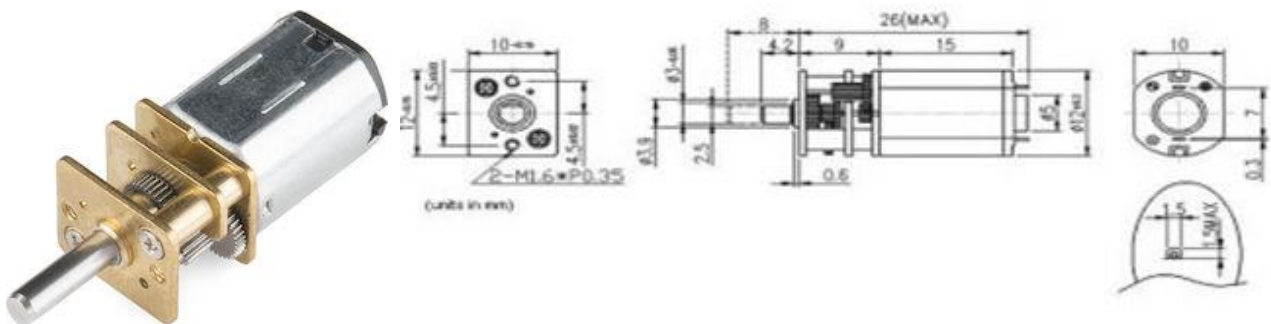
Το L298N έχει προστατευτικές λειτουργίες, όπως προστασία από υπέρταση και θερμική προστασία, προκειμένου να προστατεύσει τον εξοπλισμό στον οποίο συνδέεται.



Εικόνα 11. Motor driver και τεχνικά χαρακτηριστικά του motor driver[22].

5.2 DC Motor

Οι DC κινητήρες λειτουργούν με ηλεκτρική ενέργεια που παρέχεται σε μορφή συνεχούς ρεύματος. Μπορούν να χρησιμοποιηθούν σε πολλές εφαρμογές ανάλογα με τις απαιτήσεις. Επίσης, είναι ιδανικοί για εφαρμογές που απαιτείται ρύθμιση της ταχύτητας αλλά και της κατεύθυνσης, όπως στην βιομηχανία, την ρομποτική, τα αυτοκίνητα, σε ηλεκτρικά εργαλεία, παιχνίδια και πολλά άλλα. Στην εικόνα 12 παρουσιάζεται ο κινητήρας που χρησιμοποιήθηκε και οι διαστάσεις του, ενώ στον πίνακα 2 παρουσιάζονται τα τεχνικά χαρακτηριστικά του κινητήρα.



Εικόνα 12. DC κινητήρας[23].

Πίνακας 2: Προδιαγραφές DC κινητήρα[23].

Τάση	6-12 V
Ταχύτητα	400 rpm -12 V
Ρεύμα χωρίς φορτίο	50mA
Ρεύμα Stall	600mA
Ροπή	0.5Kg.cm
Διάμετρος άξονα	3mm
Μήκος (Εξ. άξονα)	9,3mm
Διαστάσεις	12*10mm

5.3 Αισθητήρες υπέρυθρων

Οι αισθητήρες υπέρυθρων είναι ένα module (μονάδα επέκτασης) που χρησιμοποιείται για την ανίχνευση εμποδίων με την χρήση υπέρυθρων αισθητήρων (infrared sensors). Αυτό το module συνήθως χρησιμοποιείται σε έργα ή εφαρμογές που απαιτούν την αυτόματη ανίχνευση αλλά και την αποφυγή εμποδίων, όπως τα drones, ρομποτικά αυτοκίνητα, ρομπότ καθαρισμού και αλλά παρόμοια. Οι αισθητήρες υπέρυθρων λειτουργούν εκπέμποντας υπέρυθρη ακτινοβολία και στην συνέχεια μετρούν τον χρόνο που χρειάζεται η ακτινοβολία αυτή να επιστρέψει μετά από ανακλάσεις από τυχόν εμπόδια. Σε περίπτωση που υπάρχει εμπόδιο στην διαδρομή του υπέρυθρου φωτός, ο αισθητήρας θα εντοπίσει αυτήν την ανακλαστική ακτινοβολία και θα επιστρέψει μία αντίστοιχη ένδειξη ή σήμα που υποδεικνύει την παρουσία ενός εμποδίου. Τέλος, οι αισθητήρες υπέρυθρων συνήθως περιλαμβάνουν αναγκαίους για την λειτουργία αισθητήρες, κυκλώματα επεξεργασίας σήματος και εξόδους, καθώς και τις ρυθμιζόμενες παραμέτρους για την ευαισθησία και την απόσταση ανίχνευσης.

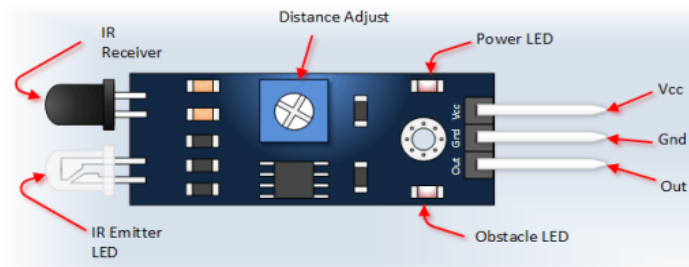


Εικόνα 13. Αισθητήρας υπέρυθρων[14], [24].

Στην παρακάτω φωτογραφία παρουσιάζονται τα τεχνικά χαρακτηριστικά του αισθητήρα υπέρυθρων, καθώς και οι ακροδέκτες του.

IR Obstacle Detection Module Pin Outs

The drawing and table below identify the function of module pin outs, controls and indicators.

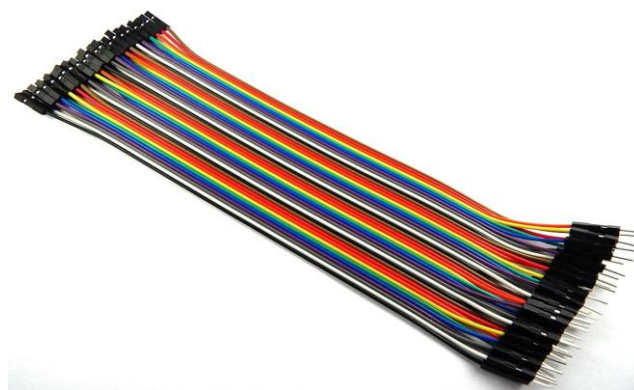


Pin, Control Indicator	Description
Vcc	3.3 to 5 Vdc Supply Input
Gnd	Ground Input
Out	Output that goes low when obstacle is in range
Power LED	Illuminates when power is applied
Obstacle LED	Illuminates when obstacle is detected
Distance Adjust	Adjust detection distance. CCW decreases distance. CW increases distance.
IR Emitter	Infrared emitter LED
IR Receiver	Infrared receiver that receives signal transmitted by Infrared emitter.

Εικόνα 14. Επεξήγηση των στοιχείων του IR Sensor[15], [25].

5.4 Jumper wires

Τα "jumper wires" (γνωστά και ως "jumpers") είναι καλώδια που τα χρησιμοποιούμε στον τομέα της ηλεκτρονικής και της ρομποτικής για να συνδέσουν διάφορα εξαρτήματα, όπως αισθητήρες ή συσκευές μεταξύ τους. Συνήθως είναι κοντά και εύκαμπτα καλώδια με αρσενικά και θηλυκά άκρα, που μπορούν να συνδεθούν σε ακροδέκτες ή αναποδογυρισμένες ακάλυπτες κεφαλές (pin headers) σε πλακέτες πρωτοτύπων (prototyping boards) ή και σε άλλα ηλεκτρονικά εξαρτήματα. Τα jumper wires είναι χρήσιμα για την προσωρινή σύνδεση και διασύνδεση διαφορετικών στοιχείων κατά την ανάπτυξη και τον προγραμματισμό πρωτοτύπων, όπως στην ρομποτική.



Εικόνα 15. Jumper wires[26].

5.5 Οι τροχοί στην ρομποτική

Στη συνέχεια, θα αναλυθούν τα διάφορα είδη τροχών και η χρήση τους σε διάφορους τομείς. Οι τροχοί συνιστούν μια κοινή μέθοδο κίνησης στην ρομποτική αλλά και στην αυτοματοποίηση. Χρησιμοποιούνται συχνά για να επιτρέπουν σε ρομπότ και άλλα αυτοματοποιημένα συστήματα να

μετακινούνται από ένα σημείο σε ένα άλλο. Οι τροχοί παρέχουν τη δυνατότητα γρήγορης, ακριβούς και εύκολης κίνησης σε ποικίλες επιφάνειες.

Στη ρομποτική, υπάρχουν διάφορα είδη τροχών, συμπεριλαμβανομένων:

1. Τροχοί με ομοκεντρικό άξονα (Omni Wheels): Αυτοί οι τροχοί έχουν πολλές μικρές ρόδες που περιστρέφονται γύρω από έναν κοινό κεντρικό άξονα. Αυτός ο σχεδιασμός επιτρέπει την κίνηση προς πολλές κατευθύνσεις χωρίς την ανάγκη για περιστροφή κάνοντας τους κατάλληλους για ρομπότ με ευελιξία στην κίνηση.



Εικόνα 16. Omni Wheels[27].

2. Τροχοί με μη ολίσθηση (Non-Skid Wheels): Αυτοί οι τροχοί συνήθως χρησιμοποιούνται σε οχήματα που κινούνται σε ομαλές επιφάνειες, όπως δρόμοι ή πλακόστρωτα.



Εικόνα 17. Non-Skid Wheels[28].

3. Τροχοί με ρόδες υψηλής πρόσφυσης (High-Traction Wheels): Αυτοί οι τροχοί έχουν ειδικό σχεδιασμό για να προσφέρουν υψηλή πρόσφυση σε επιφάνειες, όπως χωμάτινο έδαφος ή γκριπ. Χρησιμοποιούνται συχνά σε ρομπότ που χρειάζονται να περάσουν εμπόδια.



Εικόνα 18. High-Traction Wheels [29].

Οι τροχοί μπορούν να συνδυαστούν με διάφορους τρόπους, και συνήθως συνοδεύονται από κινητήρες και συστήματα ελέγχου για την κίνηση και την κατεύθυνση. Ανάλογα με τον τρόπο που χρησιμοποιούνται, μπορούν να παρέχουν διάφορες δυνατότητες κίνησης, όπως μπροστινή ή πίσω κίνηση, περιστροφική, αλλά και κίνηση στα πλάγια.

5.6 Μπαταρίες ιόντων λιθίου

Οι μπαταρίες ιόντων λιθίου (Lithium-ion batteries) είναι μια ειδική κατηγορία επαναφορτιζόμενων μπαταριών που χρησιμοποιούν λίθιο ως ένα από τα βασικά υλικά για την αποθήκευση και απελευθέρωση ηλεκτρικής ενέργειας. Αυτές οι μπαταρίες χρησιμοποιούνται σε πολλές συσκευές και εφαρμογές, συμπεριλαμβανομένων των εξής:

1. Κινητά τηλέφωνα: Οι μπαταρίες ιόντων λιθίου είναι η κύρια πηγή ενέργειας για τα κινητά τηλέφωνα, επιτρέποντάς τους να παραμένουν φορτισμένα για αρκετές ώρες χρήσης.
2. Φορητοί υπολογιστές: Οι μπαταρίες ιόντων λιθίου είναι επίσης δημοφιλείς στους φορητούς υπολογιστές, καθώς προσφέρουν υψηλή ενεργειακή πυκνότητα και μικρό μέγεθος σε σχέση με άλλους τύπους μπαταριών.
3. Ηλεκτρικά αυτοκίνητα: Οι μπαταρίες ιόντων λιθίου χρησιμοποιούνται ευρέως στα ηλεκτρικά αυτοκίνητα λόγω της υψηλής ενεργειακής απόδοσής τους καθώς έχουν την δυνατότητα να αποθηκεύουν μεγάλη ποσότητα ενέργειας.
4. Φορητές συσκευές: Από φορητές κονσόλες παιχνιδιών και φωτογραφικές μηχανές μέχρι ηλεκτρονικά τσιγάρα και φορητές συσκευές GPS, οι μπαταρίες ιόντων λιθίου είναι πηγή ενέργειας για πολλές φορητές συσκευές.

Οι μπαταρίες ιόντων λιθίου έχουν το πλεονέκτημα της μεγάλης πυκνότητας ενέργειας σε σχέση με το μέγεθός τους, καθώς και τη δυνατότητα να φορτίζονται και να αποφορτίζονται πολλές φορές πριν αποκτήσουν μείωση της απόδοσής τους. Ωστόσο, είναι απαραίτητο να τηρούνται ορισμένες προφυλάξεις όπως η αποφυγή υπερφόρτισης και υπερθέρμανσης, καθώς αυτές οι μπαταρίες μπορούν να είναι επικίνδυνες εάν δεν χρησιμοποιηθούν σωστά.



Εικόνα 19. Μπαταρίες ιόντων λιθίου[30].

5.7 Υλοποίηση συστήματος

Σε αυτό το κεφάλαιο, παρουσιάζονται φωτογραφίες από τη διαδικασία υλοποίησης του οχήματος.

Η εικόνα 21 δείχνει την εγκατάσταση των κινητήρων στον σκελετό του οχήματος.



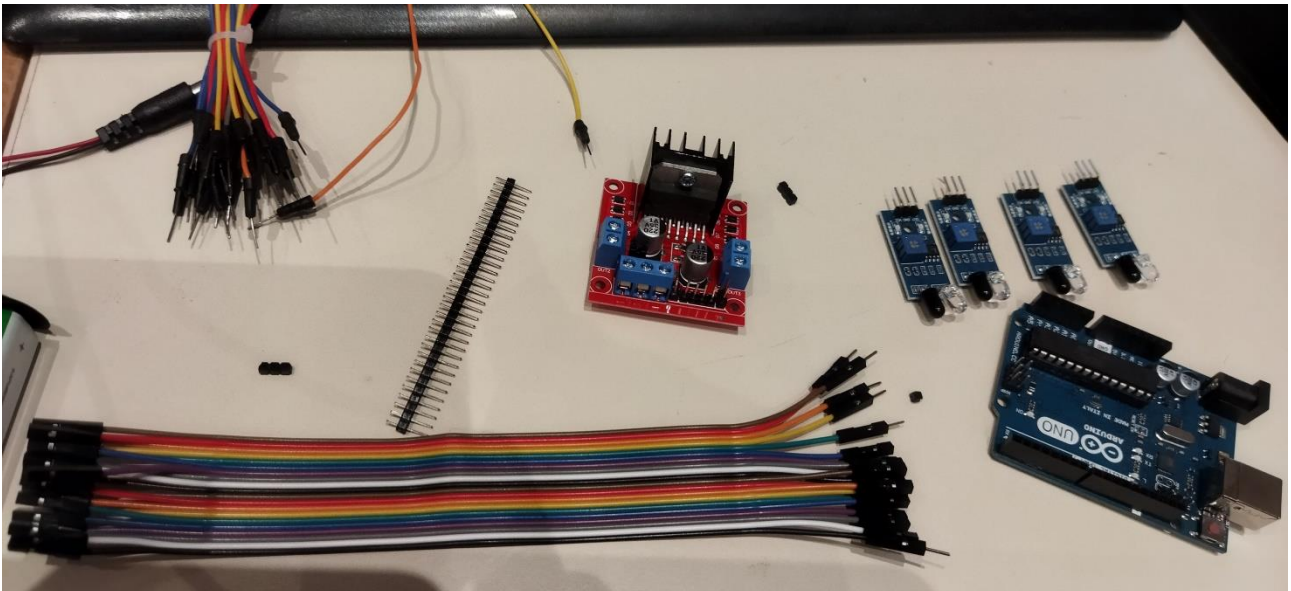
Εικόνα 20. Τοποθέτηση των κινητήρων στον σκελετό.

Στην εικόνα 22 παρουσιάζεται η εγκατάσταση του κεντρικού τροχού. Για την κατασκευή του χρησιμοποιήσαμε έναν στυπιοθλίπτη και μία γυάλινη μπίλια, ο σκοπός για τον οποίο κατασκευάστηκε αυτού του είδους ο τροχός, είναι η αποφυγή βάρους και μεγέθους αλλά και για την καλή και αποτελεσματική επαφή του στο έδαφος σε σχέση με άλλων ειδών τροχούς.



Εικόνα 21. Εγκατάσταση κεντρικού τροχού.

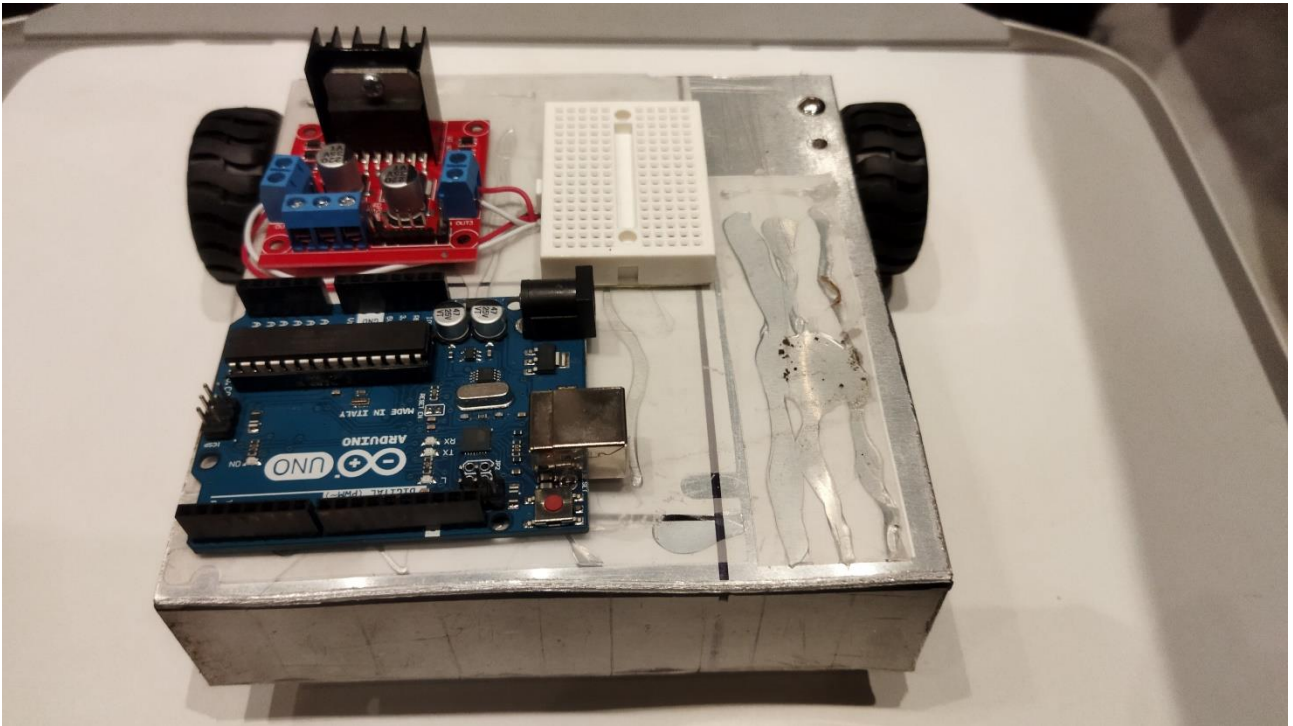
Στην εικόνα 23 βλέπουμε τα εξαρτήματα και τα καλώδια που χρειάστηκαν για την συρμάτωση του κυκλώματος.



Εικόνα 22. Εξαρτήματα και καλώδια.

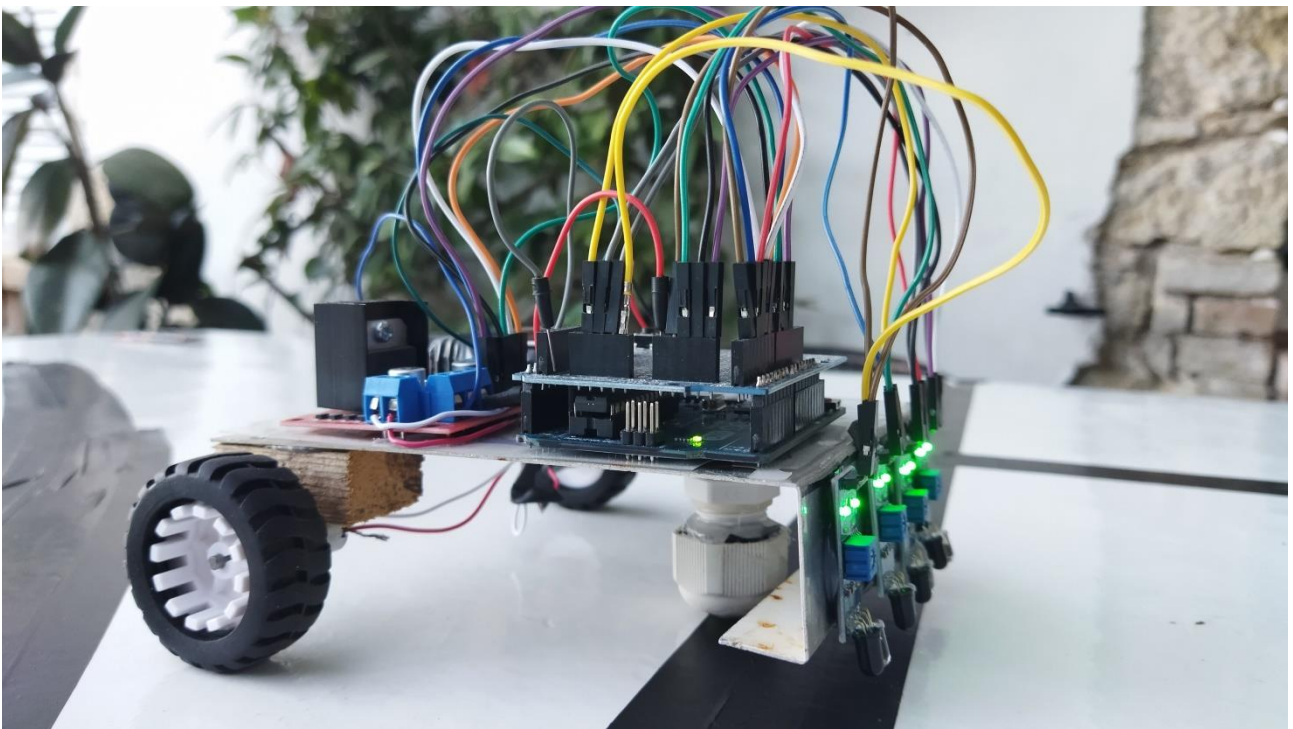
Στην εικόνα 24 βλέπουμε την τοποθέτηση των βασικών για την λειτουργία πλακετών Arduino και driver shield. Επειδή, οι πλακέτες και οι αισθητήρες δεν είναι μονωμένοι στο κάτω μέρος τους

και ο σκελετός είναι μεταλλικός εφαρμόστηκε μία επένδυση σκληρού πλαστικού της τάξεως των 2 χιλιοστών για την αποφυγή καταστροφής τους από βραχυκύκλωμα.

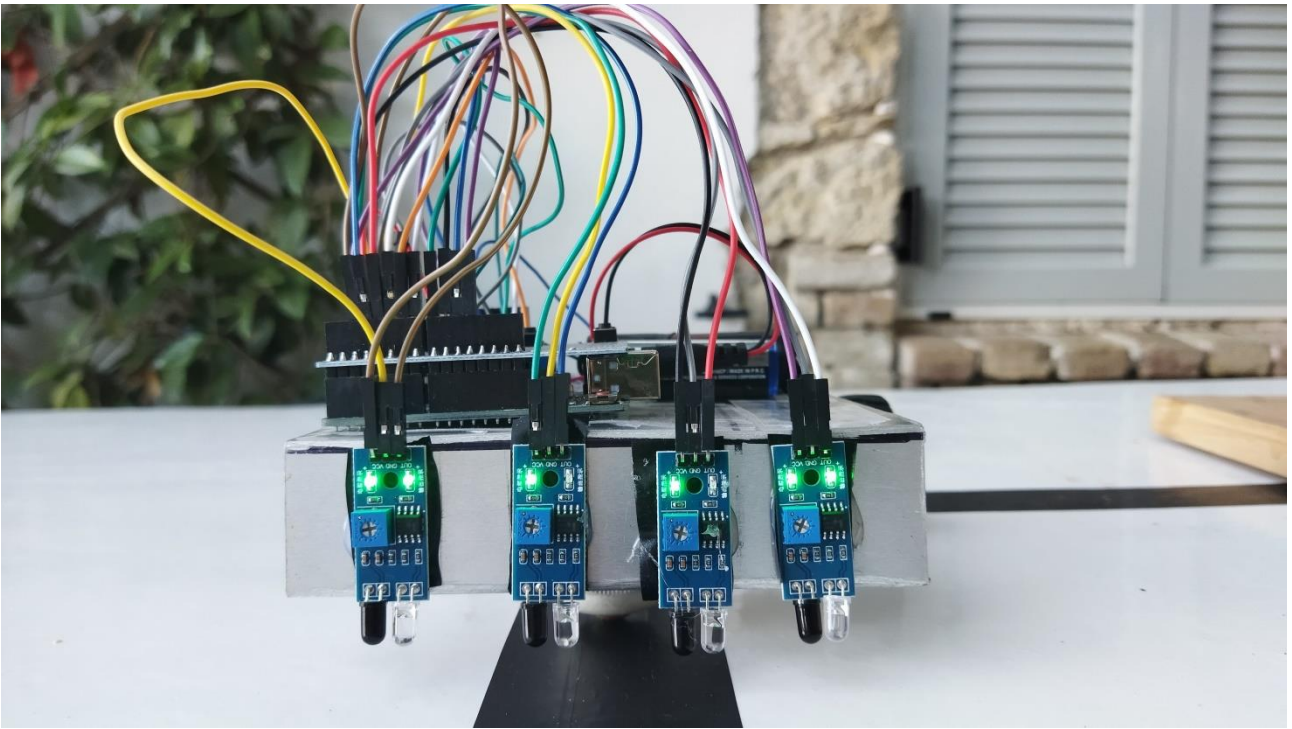


Εικόνα 23. Τοποθέτηση του Arduino και του driver shield.

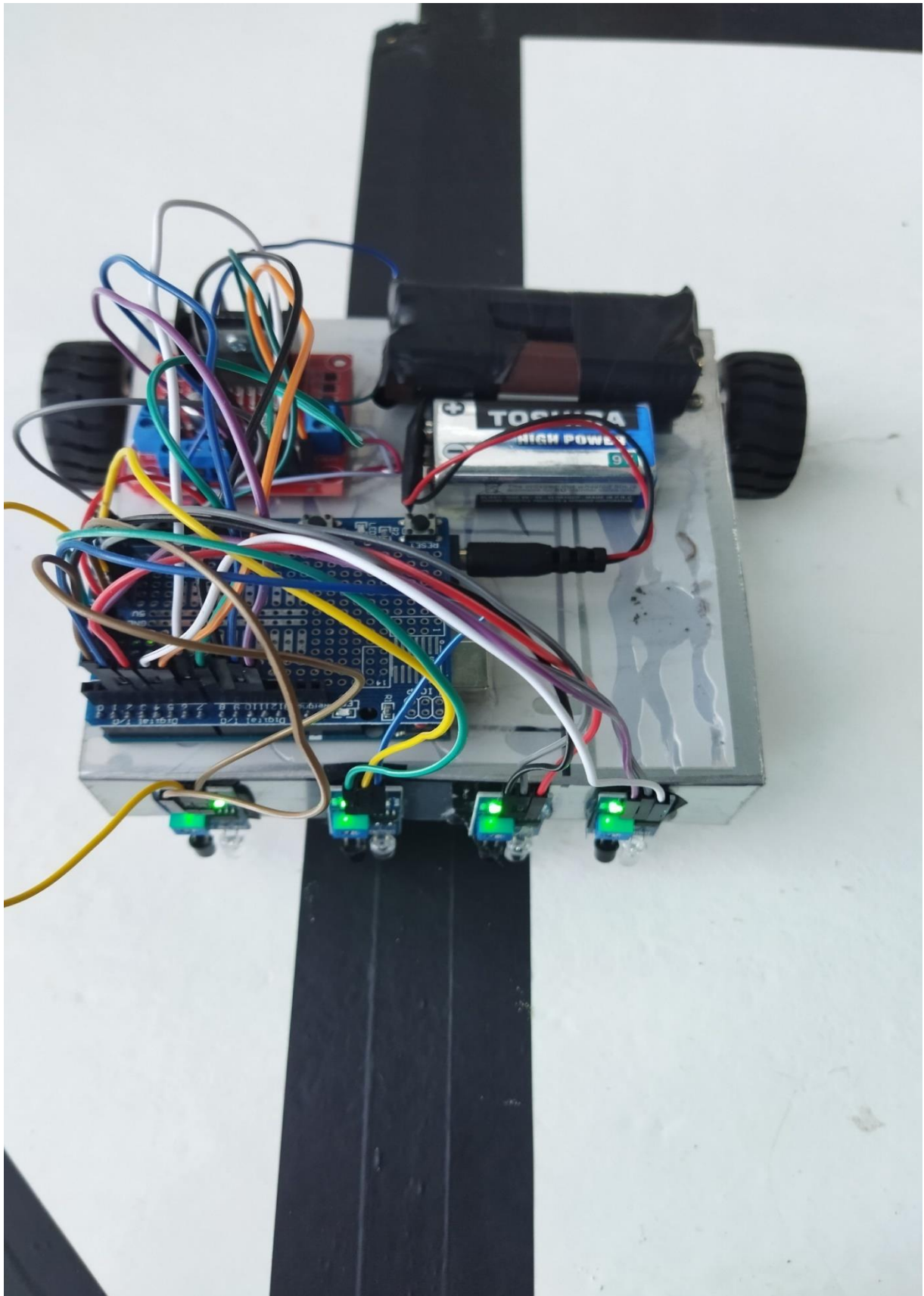
Στις εικόνες 24, 25, 26 παρουσιάζεται το όχημα ολοκληρωμένο με εγκατεστημένα όλα τα απαραίτητα εξαρτήματα.



Εικόνα 24. Ολοκλήρωση του οχήματος.



Εικόνα 25. Ολοκλήρωση του οχήματος.



Εικόνα 26. Ολοκλήρωση του οχήματος.

Κεφάλαιο 6: Εύρεση παραμέτρων ελεγκτή

Για την εύρεση των παραμέτρων του PID ελεγκτή χρησιμοποιήθηκαν 2 μέθοδοι, η “trial and error” και η δεύτερη μέθοδος των Ziegler-Nichols. Τα κριτήρια σχεδίασης για την εύρεση των κερδών ήταν τα ακόλουθα:

- Μικρό χρόνο ανόδου, δηλαδή το όχημα να έχει γρήγορη απόκριση.
- Μικρό μόνιμο σφάλμα κατάστασης, δηλαδή να αποκλίνει ελάχιστα από τα όρια της γραμμής
- Μικρό χρόνο αποκατάστασης, δηλαδή σε μικρό χρονικό διάστημα να φτάνει στην μόνιμη κατάσταση.

Πρώτα οι βέλτιστες τιμές του ελεγκτή βρέθηκαν με την μέθοδο “trial and error”. Συγκεκριμένα, πραγματοποιήθηκαν 10 διαφορετικοί συνδυασμοί κερδών για τους τρεις όρους του PID ελεγκτή. Οι τιμές των κερδών και η συμπεριφορά του συστήματος παρουσιάζονται στον πίνακα 2. Με βάση τις τιμές του πίνακα 3, η καλύτερη συμπεριφορά του συστήματος επιτεύχθηκε από την δοκιμή 10, αλλά το σύστημα παρουσίαζε μεγάλο χρόνο ανόδου και μεγάλο σφάλμα μόνιμης κατάστασης.

Πίνακας 3: Τιμές του PID με την μέθοδο trial and error.

Δοκιμή	P	I	D	Αποτέλεσμα
1	0	0	0	Καμία Ενέργεια
2	5	0	0	Δεν κινείται ευθεία
3	10	0	5	Μειωμένη ταχύτητα κινητήρων
4	5	0	10	Αργή αριστερή στροφή
5	0	5	10	Έξοδος από την πορεία
6	15	0	5	Δυσκολία στην στροφή δεξιά
7	15	0	10	Αργή πορεία
8	10	5	15	Καθυστερήση εκκίνησης αριστερού κινητήρα
9	30	0	15	Αυξημένη ταχύτητα κινητήρων
10	25	0	15	Ομαλή λειτουργία

Για την εύρεση των παραμέτρων του PID ελεγκτή με την δεύτερη μέθοδο των Ziegler-Nichols ακολουθήθηκαν τα παρακάτω βήματα:

1. Με $K_i = 0$ και $K_d = 0$ αυξήθηκε σταδιακά το K_p μέχρι να βρεθεί το οριακό κέρδος, δηλαδή η τιμή για την οποία το σύστημα κάνει αμείωτες ταλαντώσεις. Η τιμή στην οποία έκανε αμείωτες ταλαντώσεις ήταν για $K_p = 30$.
2. Υπολογίστηκε προσεγγίστεκα η οριακή συχνότητα του συστήματος. Η τιμή που υπολογίστηκε ήταν η $f_{cr} = 1.25Hz$.
3. Με βάση τον πίνακα 4 και τον τύπο (2) υπολογίστηκαν τα κέρδη του PID ελεγκτή. Ο PID ελεγκτής που προέκυψε είχε σαν $K_p = 18$, $K_i = \frac{0.6K_{cr}}{0.5P_{cr}} = 28.8$ και $K_d = 0.075K_{cr}P_{cr} = 2.8125$.
4. Βελτίωση της απόκρισης του συστήματος κάνοντας μικρές αλλαγές στις παραμέτρους του PID ελεγκτή.

$$G_{PID}(s) = K_p + \frac{K_I}{s} + K_D s = 0.6K_{cr} \left(1 + \frac{1}{0.5P_{cr}} + 0.125P_{cr} \right) \quad (2)$$

Πίνακας 4: Θεωρητικές τιμές του PID ελεγκτή με την μέθοδο Ziegler-Nichols[31].

Ελεγκτής	K_p	T_i	T_d
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Οι τελικές τιμές των παραμέτρων του PID ελεγκτή που προέκυψαν είναι $K_p = 18$, $K_i = 30.8$ και $K_d = 2.8125$. Με την εφαρμογή των συγκεκριμένων κερδών το σύστημα παρουσιάζει γρήγορη απόκριση, μικρό σφάλμα μόνιμης κατάστασης και μικρό χρόνο αποκατάστασης.

Πρέπει να σημειωθεί ότι, κατά την διάρκεια των δοκιμών με την δεύτερη μέθοδο Ziegler-Nichols παρατηρήθηκε ότι αυξάνοντας τις τιμές του K_p ή K_d το όχημα δεν είχε ομαλή πορεία με αποτέλεσμα σε ορισμένες καμπύλες να είναι αισθητό το σφάλμα.

Κεφάλαιο 7: Συνολικό κόστος υλοποίησης

Στον παρακάτω πίνακα παρουσιάζονται οι τιμές των εξαρτημάτων που χρησιμοποιήθηκαν στην πτυχιακή εργασία.

Πίνακας 5: Κόστος Υλικών.

Υλικά	Τιμή
Arduino	39€
Driver Shield	4€
IR Sensors*4	8€
Μπαταρίες *2	12€
Κινητήρες *2	16€
Ροδες *2	3€
Arduino Prototyping Shield	5€
Σύνολο	87€

Κεφάλαιο 8: Μελλοντικές επεκτάσεις

Η παρούσα πτυχιακή εργασία κατάφερε να φτάσει στο επιθυμητό σημείο που είναι το όχημα να ακολουθάει την γραμμή έχοντας γρήγορη απόκριση. Για την βελτίωση του project αυτού θα μπορούσαμε να προσθέσουμε έναν αισθητήρα απόστασης για την αποφυγή εμποδίων αλλά και άλλων ειδών αισθητήρων, όπως έναν αισθητήρα καπνού ή υγρασίας. Το όχημα αυτό με τις απαραίτητες αναβαθμίσεις θα ήταν ικανό όχι μόνο για οικιακή αλλά και για επαγγελματική χρήση όπως για την μέτρηση υγρασίας, καπνού ή κάποιου επικίνδυνου αερίου οπού ο άνθρωπος ή ένας σταθερός αισθητήρας δεν θα μπορεί να αντιληφθεί.

Βιβλιογραφία

1. <http://3gym-igoum.thesp.sch.gr/images/Arduino/book.pdf>
2. <https://www.wellpcb.com/types-of-arduino.html>
3. https://medium.com/@tharindu_peiris/arduino-d5eccf701bca
4. <https://www.create-learn.us/blog/guide-to-arduino-kits-for-kids/>
5. http://pantou.mysch.gr/kyklvmata/intro_Arduino.pdf
6. <https://www.w3schools.com/js/default.asp>
7. [https://eng.libretexts.org/Bookshelves/Industrial and Systems Engineering/Chemical Process Dynamics and Controls \(Woolf\)/09%3A Proportional-Integral-Derivative \(PID\) Control/9.02%3A P%2C I%2C D%2C PI%2C PD%2C and PID control](https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Chemical_Process_Dynamics_and_Controls_(Woolf)/09%3A_Proportional-Integral-Derivative_(PID)_Control/9.02%3A_P%2C_I%2C_D%2C_PI%2C_PD%2C_and_PID_control)
8. <https://el.jf-parede.pt/what-is-an-ir-sensor>
9. <https://www.theengineeringprojects.com/2018/10/introduction-to-arduino-ide.html>
10. <http://el.goldencellbatteries.com/news/what-is-18650-battery-cell/>
11. <https://robotics.arvtheo.com/arduino/>
12. <https://www.instructables.com/Simple-Wi-Fi-Controlled-LED-Using-Nodemcu-in-Access/>
13. <https://arduino.stackexchange.com/questions/46342/will-5v-to-input-pin-damage-the-arduino-board>
14. <http://www.rcccamp.org/day-2-supplement-ir-light-proximity-sensor.html>
15. <https://ruturajn.hashnode.dev/interfacing-ir-sensor-with-the-stm32f407-discovery-kit-using-external-interrupt>
16. <https://www.hackster.io/news/sipeed-s-tinymaix-puts-mnist-digit-recognition-on-a-modest-microchip-atmega328-microcontroller-9884bc3cc57c>
17. https://commons.wikimedia.org/wiki/File:Arduino_Logo.svg
18. https://www.researchgate.net/figure/Arduino-Uno-connections-pinouts_fig1_338345202
19. <https://pijaeducation.com/arduino/introduction-arduino/specification-of-arduino-uno/>
20. <https://www.open-electronics.org/the-power-of-arduino-this-unknown/>
21. <https://www.sacredheart.edu/academics/colleges--schools/school-of-computer-science--engineering/facilities/idea-lab/training/>
22. <https://www.projecthub.in/arduino-motor-drivers/>

23. <https://grobotronics.com/micro-metal-gearmotor-400rpm-12v.html>
24. <https://tecnocoencia.com/producto/sensor-infrarrojo-evasor-de-obstaculos-3-cables/>
25. <https://osoyoo.com/2018/01/10/arduino-tank-car-kit-lesson-9object-follow/>
26. <https://www.flipkart.com/ranvirkar-male-female-jumper-wire-bundle-20cm-40pcs-multicolor-breadboard-electronic-components-hobby-kit/p/itm64a4c8be674a2>
27. <https://www.indiamart.com/proddetail/100mm-orange-double-glass-fiber-omni-wheel-with-bush-rollers-high-quality-22322807788.html>
28. <https://grobotronics.com/wheel-42mm-for-n20-motor.html>
29. <https://www.walmart.ca/en/ip/Tires-Wheels-Set-RC-Wheels-Fluent-Driving-Strong-Buffering-Rubber-For-WLToys-124016-1-12-RC-Car/PRD7FOGOD3UXM2L>
30. <https://matshop.gr/product/%CE%BC%CF%80%CE%B1%CF%84%CE%B1%CF%81%CE%B9%CE%B1-%CE%B5%CF%80%CE%B1%CE%BD%CE%B1%CF%86%CE%BF%CF%81%CF%84%CE%B9%CE%B6%CE%BF%CE%BC%CE%B5%CE%BD%CE%B7-18650-2500mah-li-ion-3-7v-tfp1-fujicell/>
31. K. Ogata, Modern Control Engineering Fifth Edition, vol. 17, no. 3. 2009.

Παράρτημα: Κώδικας Arduino

```
// Μεταβλητές κινητήρα
int ENA = 2;          //Αριστερός κινητήρας
int motorInput1 = 3;
int motorInput2 = 4;
int motorInput3 = 5;
int motorInput4 = 6;
int ENB = 7;        //Δεξίς κινητήρας
// IR Sensors
int sensor1 = 8;    // Αριστερός αισθητήρας
int sensor2 = 9;
int sensor3 = 10;
int sensor4 = 11;   // Δεξίς αισθητήρας
// Αρχικές τιμές αισθητήρων
int sensor[4] = {0, 0, 0, 0};

//Αρχική ταχύτητα κινητήρα
int initial_motor_speed = 100;

// Output Pins for Led
int ledPin1 = A1;
int ledPin2 = A2;

// Σταθερές PID
float Kp = 18;
float Ki = 30.8;
float Kd = 2.81;

float error = 0, P = 0, I = 0, D = 0, PID_value = 0;
float previous_error = 0, previous_I = 0;

int flag = 0;

void setup()
{
  pinMode(sensor1, INPUT);
  pinMode(sensor2, INPUT);
  pinMode(sensor3, INPUT);
  pinMode(sensor4, INPUT);

  pinMode(motorInput1, OUTPUT);
  pinMode(motorInput2, OUTPUT);
  pinMode(motorInput3, OUTPUT);
  pinMode(motorInput4, OUTPUT);
  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);
}
```

```

pinMode(ledPin1, OUTPUT);
pinMode(ledPin2, OUTPUT);

digitalWrite(ledPin1, LOW);
digitalWrite(ledPin2, LOW);

Serial.begin(9600);           //ρύθμιση της σειριακής οθόνης σε προεπιλεγμένο ρυθμό μετάδοσης 9600
delay(500);
Serial.println("Started !!");
delay(1000);
}
void loop()
{
  read_sensor_values();
  Serial.print(error);
  if (error == 100) {         // Στρίψτε αριστερά μέχρι να εντοπίσει ευθεία διαδρομή
    do {                     // Στρίψτε αριστερά μέχρι να εντοπίσετε τη μέση της γραμμής
      left();
      analogWrite(ENA, 150); //Αριστερή ταχύτητα κινητήρα
      analogWrite(ENB, 90);  //Δεξιά ταχύτητα κινητήρα
      read_sensor_values();
    } while (error == 0);
  } else if (error == 101) { // δεξιά στροφή σε περίπτωση που εντοπίσει την σωστή διαδρομή
    do {                     // Στρίψε αριστερά μέχρι να εντοπίσει το κέντρο της γραμμής
      analogWrite(ENA, 140); //Αριστερή ταχύτητα κινητήρα
      analogWrite(ENB, 80);  //Δεξιά ταχύτητα κινητήρα
      sharpRightTurn();
      read_sensor_values();
    } while (error == 0);
  } else if (error == 102) {
    do {                     // Στρίψε αριστερά μέχρι να εντοπίσεις τη μέση της γραμμής
      read_sensor_values();
      analogWrite(ENA, 140); //Αριστερή ταχύτητα κινητήρα
      analogWrite(ENB, 80);  //Δεξιά ταχύτητα κινητήρα
      sharpLeftTurn();
    } while (error == 0);
  } else if (error == 103) { // Στρίψε αριστερά μέχρι να εντοπίσεις ευθεία διαδρομή ή σταμάτα εάν φτάσεις
σε αδιέξοδο.
    stop_bot();
    if (flag == 0) {
      analogWrite(ENA, 140); //Αριστερή ταχύτητα κινητήρα
      analogWrite(ENB, 80);  //Δεξιά ταχύτητα κινητήρα
      forward();
      delay(100);
      read_sensor_values();
      if (error == 103) {     /** αδιέξοδος, Τέλος! **/
        stop_bot();
        delay(200);
        digitalWrite(ledPin1, HIGH);
        digitalWrite(ledPin2, HIGH);

```

```

    flag = 1;
  } else {          /**** Κίνηση αριστερά ****/
    do {           // Στρίψε αριστερά μέχρι να εντοπίσεις το κέντρο της γραμμής
      analogWrite(ENA, 140); //Αριστερή ταχύτητα κινητήρα
      analogWrite(ENB, 80); //Δεξιά ταχύτητα κινητήρα
      sharpLeftTurn();
      read_sensor_values();
    } while (error == 0);
  }
}
} else {
  calculate_pid(); // Υπολόγισε την τιμή PID
  motor_control(); // Ρυθμίσε τον κινητήρα σύμφωνα με την πρόσφατα υπολογισμένη τιμή PID, ευθεία
κίνηση
}
}

void read_sensor_values()
{
  sensor[0] = digitalRead(sensor1);
  sensor[1] = digitalRead(sensor2);
  sensor[2] = digitalRead(sensor3);
  sensor[3] = digitalRead(sensor4);
  if ((sensor[0] == 1) && (sensor[1] == 0) && (sensor[2] == 0) && (sensor[3] == 0)) // πολύ σωστή απόκλιση
    error = -3;
  else if ((sensor[0] == 1) && (sensor[1] == 1) && (sensor[2] == 0) && (sensor[3] == 0))
    error = -2;
  else if ((sensor[0] == 0) && (sensor[1] == 1) && (sensor[2] == 0) && (sensor[3] == 0))
    error = -1;
  else if ((sensor[0] == 0) && (sensor[1] == 1) && (sensor[2] == 1) && (sensor[3] == 0)) // καμία απόκλιση
    error = 0;
  else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 1) && (sensor[3] == 0))
    error = 1;
  else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 1) && (sensor[3] == 1))
    error = 2;
  else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 0) && (sensor[3] == 1)) // πολύ αριστερά
    error = 3;
  else if ((sensor[0] == 1) && (sensor[1] == 1) && (sensor[2] == 1) && (sensor[3] == 0)) // Γύρνα το ρομπότ
στην αριστερή πλευρά
    error = 100;
  else if ((sensor[0] == 0) && (sensor[1] == 1) && (sensor[2] == 1) && (sensor[3] == 1)) // Γύρνα το ρομπότ
δεξιά
    error = 101;
  else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 0) && (sensor[3] == 0)) // Make U turn
    error = 102;
  else if ((sensor[0] == 1) && (sensor[1] == 1) && (sensor[2] == 1) && (sensor[3] == 1)) // Στρίψε αριστερά ή
σταμάτα
    error = 103;
}

void calculate_pid()

```

```

{
P = error;
I = I + previous_I;
D = error - previous_error;

PID_value = (Kp * P) + (Ki * I) + (Kd * D);

previous_I = I;
previous_error = error;
}

void motor_control()
{
// Υπολογισμός της πραγματικής ταχύτητας κινητήρα:

int left_motor_speed = initial_motor_speed - PID_value;
int right_motor_speed = initial_motor_speed + PID_value;

// Η ταχύτητα του κινητήρα δεν πρέπει να υπερβαίνει τη μέγιστη τιμή PWM
left_motor_speed = constrain(left_motor_speed, 0, 255);
right_motor_speed = constrain(right_motor_speed, 0, 255);

analogWrite(ENA, left_motor_speed-30); //Αριστερή ταχύτητα κινητήρα //0
analogWrite(ENB, right_motor_speed); //Δεξιά ταχύτητα κινητήρα //120

forward();
}

void forward()
{
digitalWrite(motorInput1, HIGH);
digitalWrite(motorInput2, LOW);
digitalWrite(motorInput3, HIGH);
digitalWrite(motorInput4, LOW);
}

void reverse()
{
digitalWrite(motorInput1, LOW);
digitalWrite(motorInput2, HIGH);
digitalWrite(motorInput3, LOW);
digitalWrite(motorInput4, HIGH);
}

void left()
{
digitalWrite(motorInput1, LOW);
digitalWrite(motorInput2, LOW);
digitalWrite(motorInput3, LOW);
digitalWrite(motorInput4, HIGH);
}

void right()
{

```

```
digitalWrite(motorInput1, HIGH);
digitalWrite(motorInput2, LOW);
digitalWrite(motorInput3, LOW);
digitalWrite(motorInput4, LOW);
}

void sharpRightTurn()
{
digitalWrite(motorInput1, HIGH);
digitalWrite(motorInput2, LOW);
digitalWrite(motorInput3, LOW);
digitalWrite(motorInput4, HIGH);
}void sharpLeftTurn()
{
digitalWrite(motorInput1, LOW);
digitalWrite(motorInput2, HIGH);
digitalWrite(motorInput3, HIGH);
digitalWrite(motorInput4, LOW);
}

void stop_bot()
{
digitalWrite(motorInput1, LOW);
digitalWrite(motorInput2, LOW);
digitalWrite(motorInput3, LOW);
digitalWrite(motorInput4, LOW);
}
```

