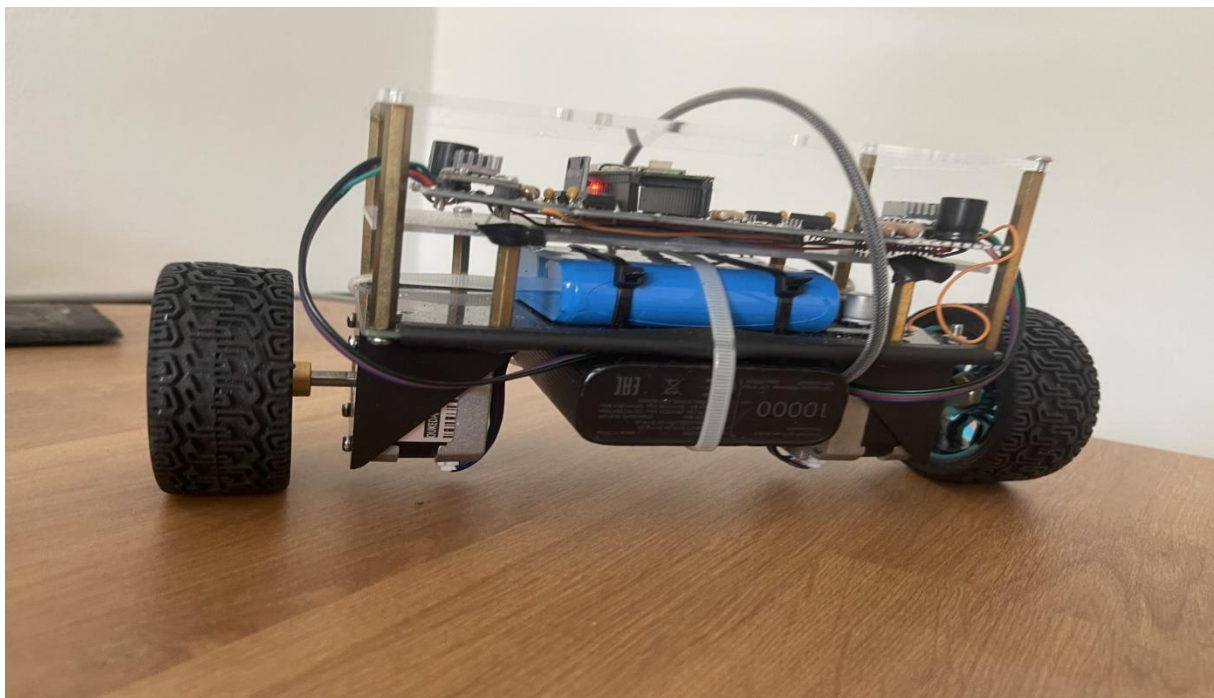


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη Πλατφόρμας Μεταφοράς Αντικειμένων με
Ασύρματο Χειρισμό»



Του φοιτητή
Αναστάσιος Μοσχόπουλος
Αρ. Μητρώου: 518088

Επιβλέπων
Ονοματεπώνυμο Γιακουμής Άγγελος
Βαθμίδα Αναπληρωτής Καθηγητής

Ιούλιος 2026

Τίτλος Δ.Ε. Ανάπτυξη Πλατφόρμας Μεταφοράς Αντικειμένων με Ασύρματο Χειρισμό

Κωδικός Δ.Ε. 25286

Ονοματεπώνυμο φοιτητή Αναστάσιος Μοσχόπουλος

Ονοματεπώνυμο εισηγητή Άγγελος Γιακουμής

Ημερομηνία ανάληψης Δ.Ε. Ιούλιος 2026

Ημερομηνία περάτωσης Δ.Ε. Σεπτέμβριος 2026

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Μοσχόπουλου Αναστάσιου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Στην οικογένεια μου.

«Αφιέρωση»

Πρόλογος

Η παρούσα διπλωματική εργασία αφορά τον σχεδιασμό και την κατασκευή ενός αυτοεξισορροπούμενου ρομπότ δύο τροχών, το οποίο ελέγχεται ασύρματα από κινητό τηλέφωνο. Η ιδέα βασίζεται στη λειτουργία του ανεστραμμένου εκκρεμούς: το σώμα του ρομπότ δεν μπορεί να παραμείνει όρθιο χωρίς συνεχή διόρθωση, γι' αυτό οι τροχοί πρέπει να κινούνται ανάλογα με την κλίση που μετρά ο αισθητήρας.

Για την υλοποίηση χρησιμοποιήθηκαν ο μικροελεγκτής Raspberry Pi Pico 2 W, ο αισθητήρας MPU6050, δύο βηματικοί κινητήρες και δύο οδηγοί DRV8825. Η ισορροπία επιτυγχάνεται με PID έλεγχο, ενώ η επικοινωνία με το κινητό γίνεται μέσω Bluetooth Low Energy. Κατά την ανάπτυξη χρειάστηκαν επαναλαμβανόμενες δοκιμές στη μηχανική διάταξη, στη φορά των κινητήρων, στον προσανατολισμό του αισθητήρα και στις παραμέτρους του ελεγκτή.

Η εργασία δεν περιορίζεται στην παρουσίαση ενός τελικού κυκλώματος. Καταγράφει τη διαδικασία με την οποία τα επιμέρους τμήματα συνδυάστηκαν και προσαρμόστηκαν μέχρι να προκύψει ένα λειτουργικό πρωτότυπο. Ιδιαίτερη βαρύτητα δίνεται στις πρακτικές δυσκολίες της κατασκευής, καθώς ακόμη και μια μικρή αλλαγή στη θέση μιας μπαταρίας, στο πρόσημο της γωνίας ή στη φορά ενός κινητήρα μπορεί να επηρεάσει άμεσα τη συμπεριφορά του ρομπότ.

Περίληψη

Η εργασία παρουσιάζει την ανάπτυξη μιας αυτοεξισορροπούμενης πλατφόρμας δύο τροχών με δυνατότητα ασύρματου χειρισμού. Κεντρική μονάδα ελέγχου είναι ο Raspberry Pi Pico 2 W, ο οποίος διαβάζει τις μετρήσεις του αισθητήρα MPU6050, υπολογίζει τη γωνία κλίσης και εκτελεί τον αλγόριθμο PID. Η κίνηση πραγματοποιείται με δύο βηματικούς κινητήρες, οι οποίοι οδηγούνται από DRV8825 και ελέγχονται μέσω σημάτων STEP και DIR.

Οι μετρήσεις του επιταχυνσιόμετρου και του γυροσκοπίου συνδυάζονται με συμπληρωματικό φίλτρο. Πριν χρησιμοποιηθούν στον έλεγχο, διορθώνονται με offsets που υπολογίστηκαν για τον συγκεκριμένο αισθητήρα και τη συγκεκριμένη θέση τοποθέτησης. Η ασύρματη επικοινωνία υλοποιήθηκε μέσω Bluetooth Low Energy, ώστε ο χρήστης να μπορεί να ενεργοποιεί ή να απενεργοποιεί το ρομπότ, να αλλάζει τις παραμέτρους του PID, να δίνει εντολές κίνησης και να λαμβάνει διαγνωστικά δεδομένα.

Κατά τις δοκιμές αντιμετωπίστηκαν προβλήματα που σχετίζονταν με τη φορά των κινητήρων, το πρόσημο της γωνίας, τις ταλαντώσεις, την αναγνώριση των εντολών BLE και τη μετακίνηση των βαριών εξαρτημάτων. Οι διορθώσεις έγιναν σταδιακά και οδήγησαν σε λειτουργικό πρωτότυπο με ισορροπία, κίνηση εμπρός και πίσω, στροφή και μηχανισμό διακοπής σε περίπτωση πτώσης.

Λέξεις-κλειδιά: αυτοεξισορροπούμενο ρομπότ, Raspberry Pi Pico 2 W, MPU6050, PID, βηματικοί κινητήρες, Bluetooth Low Energy.

«Development of a Wirelessly Controlled Object-Transport Platform»

«Anastasios Moschopoulos»

Abstract

This thesis presents the development of a two-wheeled self-balancing platform with wireless control. The system is based on a Raspberry Pi Pico 2 W, which reads data from an MPU6050 inertial sensor, estimates the tilt angle and executes a PID control algorithm. Two stepper motors provide motion through DRV8825 drivers and are controlled by STEP and DIR signals.

Accelerometer and gyroscope measurements are combined through a complementary filter. Sensor offsets were calculated for the specific MPU6050 unit and its final mounting orientation. Bluetooth Low Energy was used for communication with a mobile phone, allowing the user to arm or disarm the controller, modify PID parameters, send motion commands and monitor diagnostic data.

The development process included several practical tests. The main issues concerned motor direction, sensor-axis orientation, sign conventions, PID oscillations, BLE command termination and the mechanical stability of the batteries and electronic boards. The final prototype combined balance control, forward and backward movement, turning, wireless parameter adjustment and a fall-stop safety mechanism. The project demonstrates how mechanical construction, power electronics, sensor processing and real-time software must be tuned together in order to achieve stable operation.

Keywords: self-balancing robot, Raspberry Pi Pico 2 W, MPU6050, PID control, stepper motors, Bluetooth Low Energy.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την κοπέλα μου που ήταν πάντα δίπλα μου, τα αδέρφια μου, και τους γονείς μου που μου δίνουν δύναμη για την εκπλήρωση της, τον φίλο μου Φωκίων για την βοήθεια του, καθώς και τον καθηγητή μου κύριο Άγγελο Γιακουμή για την καθοδήγηση και την διαθεσιμότητα καθόλη την διάρκεια εκπόνησης της εργασίας.

Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract	vii
Ευχαριστίες	viii
Περιεχόμενα	ix
Κατάλογος Σχημάτων	xi
Κατάλογος Πινάκων.....	xii
Συντομογραφίες.....	xiii
Κεφάλαιο 1ο: Εισαγωγή	1
1.1 Ρομποτ.....	1
1.1.1 Τύποι κίνησης.....	1
1.2 Οδηγοί Βηματικών Κινητήρων	3
1.3 Ενσωματωμένα Συστήματα και Μικροελεγκτές.....	3
1.4 Δυναμική σταθεροποίηση και αδρανειακές μονάδες μέτρησης.....	4
1.4.1 IMUs.....	4
Κεφάλαιο 2ο: Σχεδιασμός και Κατασκευή	5
2.1 Γενική αρχιτεκτονική και λειτουργικές απαιτήσεις	5
2.2 Περιορισμοί και τεχνικές προκλήσεις	6
2.3 Μηχανικός σχεδιασμός και κατανομή βάρους.....	6
2.4 Βασικά ηλεκτρονικά εξαρτήματα	7
2.4.1 Raspberry Pi pico 2 W.....	7
2.4.2 Αισθητήρας MPU6050	11
2.4.3 Βηματικοί κινητήρες και οδηγοί DRV8825	15
2.4.4 Οπτοζεύκτες	21
2.4.5 Τροφοδοσία και step-down converter	25
2.5 Ηλεκτρονική συνδεσμολογία	30
2.6 Τροφοδοσία και ηλεκτρική προστασία	32
2.7 Υλικά και κόστος	32
2.8 Τελική διάταξη και συναρμολόγηση.....	33
Κεφάλαιο 3ο: Ανάπτυξη Λογισμικού, Έλεγχος και Δοκιμές Λειτουργίας.....	35
3.1 Περιβάλλον Ανάπτυξης και βιβλιοθήκες	35
3.2 Δομή του προγράμματος και καταστάσεις λειτουργίας	35

3.3	Ανάγνωση, βαθμονόμηση και εκτίμηση γωνίας	36
3.3.1	Πρόγραμμα βαθμονόμησης του αισθητήρα MPU6050	37
3.3.2	Ρύθμιση των καταχωρητών του MPU6050	38
3.3.3	Ρυθμός δειγματοληψίας και ψηφιακό φίλτρο	38
3.3.4	Αρχικοποίηση και διαχείριση ισχύος	39
3.3.5	Βηματικοί κινητήρες και οδηγοί DRV8825	40
3.4	Παραγωγή παλμών και έλεγχος κινητήρων	40
3.5	PID ελεγκτής και διαδικασία ρύθμισης	40
3.6	Κίνηση και στροφή	41
3.7	Ασύρματη επικοινωνία μέσω BLE	42
3.8	Μηχανισμοί ασφαλείας	46
3.9	Δοκιμές, προβλήματα και διορθώσεις	47
3.10	Τελικό αποτέλεσμα και αξιολόγηση	47
Κεφάλαιο 4ο:	Συμπεράσματα και Μελλοντικές Βελτιώσεις	48
4.1	Συμπεράσματα	48
4.2	Περιορισμοί	49
4.3	Μελλοντικές βελτιώσεις	49
ΒΙΒΛΙΟΓΡΑΦΙΑ		52
ΠΑΡΑΡΤΗΜΑ Α : Κώδικας Ρύθμισης MPU		54
ΠΑΡΑΡΤΗΜΑ Β :Κώδικας Μοκροελεγκτη		56

Κατάλογος Σχημάτων

Σχήμα 1.1 Τύποι ρομπότ.	1
Σχήμα 1.2 Ταξινόμηση βασικών ηλεκτρικών κινητήρων συνεχούς και εναλλασσόμενου ρεύματος.	2
Σχήμα 2.1 Γενική αρχιτεκτονική και βρόχος ελέγχου του αυτοεξισορροπούμενου ρομπότ.	5
Σχήμα 2.2 Block διάγραμμα του συστήματος.	6
Σχήμα 2.3 Σχέδιο βασικής κατασκευής.	7
Σχήμα 2.4 Raspberry Pi Pico 2 W.	8
Σχήμα 2.5 Διάταξη ακροδεκτών του Raspberry Pi Pico 2 W.	10
Σχήμα 2.6 Θέση του Raspberry Pi Pico 2 W και περιοχή της ενσωματωμένης κεραίας στην τελική κατασκευή.	11
Σχήμα 2.7 Διάταξη ακροδεκτών και προσανατολισμός αξόνων του MPU6050.	13
Σχήμα 2.8 Προσανατολισμός MPU-6050.	14
Σχήμα 2.9 Τυπικό κύκλωμα λειτουργίας του MPU6050.	14
Σχήμα 2.10 Διάταξη και λειτουργίες των ακροδεκτών του DRV8825.	16
Σχήμα 2.11 Εσωτερικό λειτουργικό διάγραμμα του DRV8825.	17
Σχήμα 2.12 Ρύθμιση V_{ref}	19
Σχήμα 2.13 Χρήση τοπικού πυκνωτή αποσύζευξης και πυκνωτή εξομάλυνσης στην τροφοδοσία οδηγού κινητήρα.	20
Σχήμα 2.14 Τυπικό κύκλωμα εφαρμογής του οδηγού DRV8825.	21
Σχήμα 2.15 Εσωτερική δομή και διάταξη ακροδεκτών του οπτοζεύκτη 6N137.	23
Σχήμα 2.16 Εσωτερικό λειτουργικό διάγραμμα του μετατροπέα LM2596.	27
Σχήμα 2.17 Διάταξη και λειτουργία των ακροδεκτών του LM2596.	27
Σχήμα 2.18 Τυπικό κύκλωμα της ρυθμιζόμενης έκδοσης του LM2596.	28
Σχήμα 2.19 Ρύθμιση τάσης εξόδου.	28
Σχήμα 2.20 Τοποθέτηση της πλακέτας στο ρομπότ.	29
Σχήμα 2.21 Κύκλωμα δοκιμών βηματικών κινητήρων.	30
Σχήμα 2.22 Το ηλεκτρονικό σχέδιο του κυκλώματος.	31
Σχήμα 2.23 Τελικό κύκλωμα τοποθετημένο στο ρομπότ.	34
Σχήμα 2.24 Τελική κατασκευή.	34
Σχήμα 3.1 Βιβλιοθήκες.	35
Σχήμα 3.2 Ορισμός ακροδεκτών.	35
Σχήμα 3.3 Βασικές καταστάσεις.	36
Σχήμα 3.4 Ορισμός offsets.	37
Σχήμα 3.5 Τιμές κατεύθυνσης του κάθε κινητήρα.	40
Σχήμα 3.6 Περιβάλλον σάρωσης της εφαρμογής nRF Connect.	43
Σχήμα 3.7 Σύνδεση με το ρομπότ και λήψη διαγνωστικών δεδομένων.	44
Σχήμα 3.8 Αποστολή της εντολής ARM σε μορφή UTF-8.	45

Κατάλογος Πινάκων

Πίνακας 2.1 Βασικά χαρακτηριστικά του Raspberry Pi Pico 2 W.....	9
Πίνακας 2.2 Ακροδέκτες ελέγχου.	15
Πίνακας 2.3 Επιλογές μικροβηματισμού.	18
Πίνακας 2.4 Διάταξη ακροδεκτών.	22
Πίνακας 2.5 Βασικά εξαρτήματα κατασκευής.	25
Πίνακας 2.6 Αντιστοίχιση ακροδεκτών μικροελεγκτή.	31
Πίνακας 2.7 Υλικά κατασκευής και κόστος.	32

Συντομογραφίες

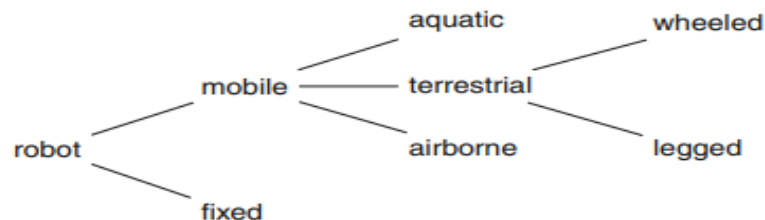
BLE — Bluetooth Low Energy
DC — Direct Current (Συνεχές Ρεύμα)
DC-DC — Direct Current to Direct Current (Μετατροπέας συνεχούς τάσης)
IMU — Inertial Measurement Unit (Αδρανειακή Μονάδα Μέτρησης)
I²C — Inter-Integrated Circuit (Δίαυλος σειριακής επικοινωνίας)
SCL — Serial Clock Line (Γραμμή ρολογιού I²C)
SDA — Serial Data Line (Γραμμή δεδομένων I²C)
GPIO — General-Purpose Input/Output (Είσοδος/Εξοδος γενικής χρήσης)
PWM — Pulse Width Modulation (Διαμόρφωση εύρους παλμού)
UART — Universal Asynchronous Receiver/Transmitter (Ασύγχρονος πομπός/δέκτης)
SPI — Serial Peripheral Interface (Σειριακή περιφερειακή διεπαφή)
USB — Universal Serial Bus (Καθολικός Σειριακός Δίαυλος)
PID — Proportional–Integral–Derivative (Αναλογικός–Ολοκληρωτικός–Παραγωγικός έλεγχος)
RP2350 — Raspberry Pi RP2350 (Μικροελεγκτής της Raspberry Pi)
MPU6050 — Motion Processing Unit 6050 (Αισθητήρας IMU έξι αξόνων)
DRV8825 — DRV8825 Stepper Motor Driver (Οδηγός βηματικού κινητήρα)
6N137 — 6N137 High-Speed Optocoupler (Οπτοζεύκτης υψηλής ταχύτητας)
NEMA — National Electrical Manufacturers Association
Li-Po — Lithium Polymer (Μπαταρία πολυμερούς λιθίου)
VREF — Reference Voltage (Τάση αναφοράς)
VDD — Supply Voltage (Τάση τροφοδοσίας ολοκληρωμένου κυκλώματος)
VLOGIC — Logic Supply Voltage (Τάση τροφοδοσίας)
GND — Ground (Γείωση)
STEP — Step (Σήμα βηματισμού κινητήρα)
DIR — Direction (Σήμα κατεύθυνσης κινητήρα)
ENBL — Enable (Σήμα ενεργοποίησης οδηγού κινητήρα)
RESET — Reset (Επαναφορά)
SLEEP — Sleep (Κατάσταση χαμηλής κατανάλωσης του οδηγού)
JOY — Joystick Command (Εντολή χειρισμού κίνησης)
ZERO — Zeroing (Αρχικοποίηση/μηδενισμός γωνίας ισορροπίας)
ARM — Arm (Ενεργοποίηση ελέγχου κίνησης)
DISARM — Disarm (Απενεργοποίηση ελέγχου κίνησης)
STATUS — Status (Αίτημα κατάστασης συστήματος)
BLE UART — Bluetooth Low Energy UART (Ασύρματη σειριακή επικοινωνία μέσω BLE)

Κεφάλαιο 1ο: Εισαγωγή

1.1 Ρομποτ

Η ρομποτική αποτελεί ένα από τα ταχύτερα αναπτυσσόμενα πεδία της επιστήμης και της τεχνολογίας, με εφαρμογές στη βιομηχανία, στην ιατρική, στις υπηρεσίες, στις μεταφορές και στην εξερεύνηση. Ένα ρομποτικό σύστημα συνδυάζει μηχανικά, ηλεκτρονικά και υπολογιστικά υποσυστήματα, ώστε να μπορεί να αντιλαμβάνεται πληροφορίες από το περιβάλλον του και να εκτελεί προκαθορισμένες ή αυτόματα ελεγχόμενες ενέργειες [1], [2].

Ιδιαίτερη κατηγορία αποτελούν τα κινητά ρομπότ (mobile robots), τα οποία έχουν τη δυνατότητα μετακίνησης στον χώρο. Ανάλογα με το περιβάλλον και τον τρόπο κίνησης μπορούν να χρησιμοποιούν τροχούς, ερπύστριες, πόδια, έλικες ή άλλα συστήματα πρόωσης. Η επιλογή του συστήματος κίνησης επηρεάζει άμεσα την ευστάθεια, την πολυπλοκότητα του ελέγχου, την κατανάλωση ενέργειας και το είδος του εδάφους στο οποίο μπορεί να λειτουργήσει το ρομπότ [1], [2].



Σχήμα 1.1 Τύποι ρομπότ.

1.1.1 Τύποι κίνησης

Τα κινητά ρομπότ μπορούν να ταξινομηθούν ανάλογα με το περιβάλλον και τον μηχανισμό κίνησής τους σε χερσαία, εναέρια, υποβρύχια και άλλες ειδικές κατηγορίες [1]. Στα χερσαία ρομπότ συναντώνται κυρίως τροχοφόρες, ερπυστριοφόρες και περπατητές διατάξεις:

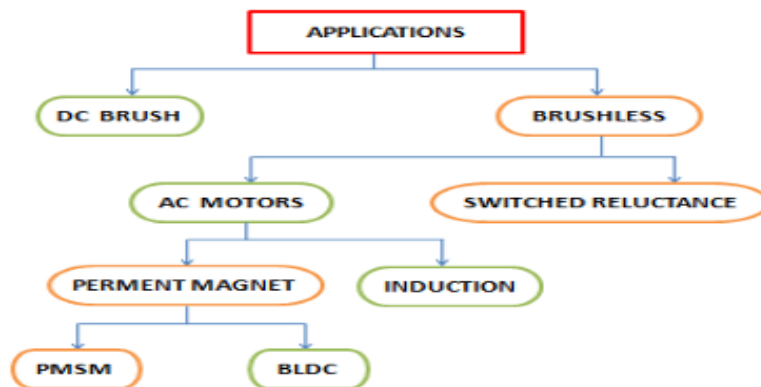
- Χερσαία ρομπότ (Land-based):
 - Τροχοφόρα ρομπότ (Wheeled Mobile Robots - WMRs): χρησιμοποιούν έναν ή περισσότερους τροχούς για την κίνησή τους και χαρακτηρίζονται συνήθως από απλή μηχανολογική κατασκευή, χαμηλότερο κόστος και καλή ενεργειακή απόδοση σε σχετικά ομαλές επιφάνειες. Οι συμβατικές τροχοφόρες πλατφόρμες με τρία ή περισσότερα σημεία στήριξης μπορούν να παρουσιάζουν στατική ευστάθεια. Αντίθετα, τα δίτροχα αυτοεξισορροπούμενα ρομπότ αποτελούν ιδιαίτερη περίπτωση, καθώς δεν μπορούν να παραμείνουν κατακόρυφα χωρίς συνεχή ενεργό έλεγχο. Η ισορροπία τους επιτυγχάνεται με συνεχή μετακίνηση των τροχών προς την κατεύθυνση της πτώσης.
 - Περπατητά (Walking or Legged) ρομπότ: Χρησιμοποιούν πόδια, προσφέροντας μεγαλύτερη ευλυγισία σε ανώμαλα και μαλακά εδάφη, αλλά με αυξημένο κόστος και πολυπλοκότητα ελέγχου, ιδιαίτερα σε θέματα στατικής και δυναμικής ισορροπίας.
 - Ρομπότ με ερπύστριες (Tracked): Προσφέρουν καλύτερη πρόσφυση και ευελιξία σε χαλαρά εδάφη, αλλά ο έλεγχος της ακριβούς πορείας τους (dead reckoning) είναι δυσκολότερος λόγω ολισθήσεων.

- Υβριδικά ρομπότ: Συνδυάζουν διαφορετικά συστήματα κίνησης (π.χ., πόδια-τροχοί) για βελτιωμένη προσαρμοστικότητα [1].
- Εναέρια ρομπότ (Air-based): Γνωστά ως μη επανδρωμένα αεροσκάφη (drones).
- Υποβρύχια ρομπότ (Water-based).
- Άλλα (π.χ., φιδοειδή, σκουληκοειδή, νανορομπότ).

Για τις απαιτήσεις της παρούσας κατασκευής επιλέχθηκε τροχοφόρα διάταξη, καθώς προσφέρει σχετικά απλή μηχανική υλοποίηση και είναι κατάλληλη για κίνηση σε ομαλές επιφάνειες εσωτερικού χώρου. Η ανεξάρτητη οδήγηση των δύο τροχών επιτρέπει τόσο τη συνεχή διόρθωση της κλίσης όσο και την πραγματοποίηση εμπρόσθιας, οπίσθιας και στροφικής κίνησης. Για τον λόγο αυτό η συγκεκριμένη διάταξη αποτέλεσε κατάλληλη βάση για την ανάπτυξη της πειραματικής πλατφόρμας της εργασίας.

Ηλεκτρικοί Κινητήρες

Οι ηλεκτρικοί κινητήρες μπορούν να ταξινομηθούν σε διάφορες κατηγορίες με βάση τον τρόπο τροφοδοσίας, τη δομή και τα λειτουργικά τους χαρακτηριστικά. Μια γενική ταξινόμηση περιλαμβάνει κινητήρες συνεχούς ή εναλλασσόμενου ρεύματος και κινητήρες ειδικού σκοπού, οι οποίοι χρησιμοποιούνται σε εφαρμογές με ιδιαίτερες απαιτήσεις ελέγχου και ακρίβειας (Σχήμα 1.2) [5].



Σχήμα 1.2 Ταξινόμηση βασικών ηλεκτρικών κινητήρων συνεχούς και εναλλασσόμενου ρεύματος.

Οι κινητήρες συνεχούς ρεύματος χαρακτηρίζονται από απλό έλεγχο ταχύτητας και ροπής. Διακρίνονται σε κινητήρες με ή χωρίς ψήκτρες (Brushless DC – BLDC). Οι BLDC κινητήρες χρησιμοποιούν μόνιμους μαγνήτες στον ρότορα και ηλεκτρονική μεταγωγή, παρουσιάζοντας υψηλή απόδοση, μειωμένες απώλειες και χαμηλές απαιτήσεις συντήρησης. Χαρακτηριστικό τους είναι η τραπεζοειδής μορφή της επαγόμενης ηλεκτρεγερτικής δύναμης (back-EMF), η οποία επιτρέπει απλούστερους αλγόριθμους ελέγχου [3].

Οι κινητήρες εναλλασσόμενου ρεύματος χωρίζονται σε ασύγχρονους και σύγχρονους. Σημαντική κατηγορία αποτελούν οι σύγχρονοι κινητήρες με μόνιμους μαγνήτες (Permanent Magnet Synchronous Motors – PMSM), στους οποίους ο ρότορας φέρει μόνιμους μαγνήτες και το μαγνητικό πεδίο του στάτη είναι ημιτονοειδές. Οι κινητήρες αυτοί χαρακτηρίζονται από υψηλή απόδοση, υψηλή πυκνότητα ισχύος και μειωμένο κυματισμό ροπής, με αυξημένες όμως απαιτήσεις σε επίπεδο ελέγχου και υπολογιστικής ισχύος [3].

Ιδιαίτερη κατηγορία είναι αυτή των ηλεκτρικών κινητήρων, οι οποίοι προορίζονται για εφαρμογές ακριβούς τοποθέτησης και ελεγχόμενης κίνησης. Σε αυτές τις εφαρμογές, η ανάγκη για προβλέψιμη και επαναλήψιμη γωνιακή μετατόπιση καθιστά απαραίτητη τη χρήση κινητήρων με διακριτή συμπεριφορά κίνησης. Σε αυτό το πλαίσιο εντάσσονται οι βηματικοί κινητήρες (stepper motors), οι οποίοι αποτελούν ξεχωριστή κατηγορία ηλεκτρικών κινητήρων και χρησιμοποιούνται ευρέως σε συστήματα αυτοματισμού, ρομποτικής και ελεγχόμενης μετακίνησης. Οι κινητήρες αυτοί χαρακτηρίζονται από διακριτά γωνιακά βήματα και δυνατότητα ακριβούς ελέγχου θέσης χωρίς απαραίτητη χρήση αισθητήρων ανάδρασης [4].

Βηματικοί Κινητήρες

Οι βηματικοί κινητήρες (stepper motors) αποτελούν ειδική κατηγορία ηλεκτρικών κινητήρων, σχεδιασμένων για εφαρμογές όπου απαιτείται ακριβής και επαναλήψιμος έλεγχος θέσης. Σε αντίθεση με τους συμβατικούς κινητήρες συνεχούς ή εναλλασσόμενου ρεύματος, στους βηματικούς κινητήρες η περιστροφή του ρότορα πραγματοποιείται σε διακριτά γωνιακά βήματα, τα οποία καθορίζονται από τη μηχανική κατασκευή και τη διάταξη των τυλιγμάτων του στάτη [4], [5].

Η αρχή λειτουργίας των βηματικών κινητήρων βασίζεται στη διαδοχική διέγερση των φάσεων του στάτη, η οποία δημιουργεί ένα περιστρεφόμενο μαγνητικό πεδίο. Ο ρότορας ευθυγραμμίζεται διαδοχικά με τις θέσεις μέγιστης μαγνητικής ροής, προκαλώντας τη μετακίνησή του από τη μία σταθερή θέση στην επόμενη. Κάθε μεταβολή στη διέγερση των φάσεων αντιστοιχεί σε ένα βήμα περιστροφής, γεγονός που επιτρέπει τον άμεσο συσχετισμό των παλμών ελέγχου με τη γωνιακή μετατόπιση του άξονα [4].

Ανάλογα με τη δομή τους, οι βηματικοί κινητήρες διακρίνονται σε τρεις βασικές κατηγορίες: κινητήρες μεταβλητής μαγνητικής αντίστασης, κινητήρες μόνιμου μαγνήτη και υβριδικούς κινητήρες. Οι κινητήρες μεταβλητής μαγνητικής αντίστασης διαθέτουν απλή κατασκευή και χαμηλό κόστος, αλλά παρουσιάζουν περιορισμένη ροπή. Οι κινητήρες μόνιμου μαγνήτη προσφέρουν μεγαλύτερη ροπή και απλούστερο έλεγχο, ενώ οι υβριδικοί κινητήρες συνδυάζουν τα πλεονεκτήματα των δύο προηγούμενων τύπων, παρέχοντας υψηλή ανάλυση βήματος και αυξημένη ακρίβεια θέσης [4], [5].

Ένα από τα σημαντικότερα πλεονεκτήματα των βηματικών κινητήρων είναι η δυνατότητα λειτουργίας σε σύστημα ανοιχτού βρόχου. Εφόσον δεν υπάρχει απώλεια βημάτων, η θεωρητική θέση του ρότορα μπορεί να εκτιμηθεί από τον αριθμό των παλμών STEP. Το χαρακτηριστικό αυτό απλοποιεί τον σχεδιασμό του συστήματος ελέγχου και μειώνει το συνολικό κόστος της εφαρμογής [4].

Παρά τα σημαντικά τους πλεονεκτήματα, οι βηματικοί κινητήρες παρουσιάζουν ορισμένους περιορισμούς, όπως μειωμένη απόδοση σε υψηλές ταχύτητες και πιθανότητα απώλειας βημάτων όταν λειτουργούν υπό αυξημένο φορτίο. Για την αντιμετώπιση αυτών των φαινομένων, σε εφαρμογές με αυξημένες απαιτήσεις ακρίβειας και ομαλής κίνησης εφαρμόζονται τεχνικές βελτιωμένου ελέγχου, όπως ο μικροβηματισμός (microstepping), καθώς και συνδυασμός με πιο εξελιγμένους αλγορίθμους ελέγχου. Με τον τρόπο αυτό επιτυγχάνεται βελτίωση της ομαλότητας της κίνησης και της συνολικής αξιοπιστίας του συστήματος. Συνολικά, οι βηματικοί κινητήρες αποτελούν μια αποδοτική και ευρέως διαδεδομένη λύση για εφαρμογές ακριβούς κίνησης και τοποθέτησης, χάρη στην απλότητα του ελέγχου τους, την καλή επαναληψιμότητα και τη συμβατότητά τους με σύγχρονα ψηφιακά συστήματα ελέγχου [4], [5].

1.2 Οδηγοί Βηματικών Κινητήρων

Οι ακροδέκτες ενός μικροελεγκτή μπορούν να δώσουν μόνο σήματα χαμηλής ισχύος και δεν είναι κατάλληλοι για την άμεση τροφοδοσία ενός κινητήρα. Για τον λόγο αυτό παρεμβάλλεται οδηγός, ο οποίος δέχεται τις εντολές του μικροελεγκτή και ρυθμίζει το ρεύμα στα τυλίγματα του κινητήρα [6].

Στους οδηγούς τύπου DRV8825, το σήμα STEP προκαλεί ένα βήμα ή μικροβήμα και το DIR ορίζει τη φορά περιστροφής. Οι ακροδέκτες μικροβηματισμού καθορίζουν την υποδιαίρεση του πλήρους βήματος, ενώ η ρύθμιση της τάσης αναφοράς περιορίζει το μέγιστο ρεύμα του κινητήρα. Πολύ χαμηλό ρεύμα μειώνει τη διαθέσιμη ροπή. Πολύ υψηλό ρεύμα αυξάνει τη θερμοκρασία των οδηγών και των κινητήρων. Η ρύθμιση επομένως γίνεται σε συνδυασμό με τις πραγματικές δοκιμές της κατασκευής.

1.3 Ενσωματωμένα Συστήματα και Μικροελεγκτές

Ένα ενσωματωμένο σύστημα σχεδιάζεται για να εκτελεί συγκεκριμένες λειτουργίες μέσα σε μια μεγαλύτερη συσκευή. Συνήθως περιλαμβάνει μικροελεγκτή, μνήμη, περιφερειακά επικοινωνίας και ακροδέκτες εισόδου και εξόδου. Σε αντίθεση με έναν υπολογιστή γενικής χρήσης, το λογισμικό του είναι άμεσα συνδεδεμένο με το υλικό που ελέγχει [7], [8].

Οι μικροελεγκτές είναι κατάλληλοι για εφαρμογές πραγματικού χρόνου, επειδή μπορούν να διαβάσουν αισθητήρες και να μεταβάλλουν ψηφιακές εξόδους χωρίς την επιβάρυνση ενός πλήρους λειτουργικού

συστήματος. Η σταθερότητα του χρόνου εκτέλεσης είναι ουσιαστική σε ένα αυτοεξισορροπούμενο ρομπότ. Ο αισθητήρας πρέπει να διαβάζεται συχνά, ο PID ελεγκτής να ενημερώνεται με γνωστό χρονικό βήμα και οι παλμοί προς τους κινητήρες να παράγονται χωρίς μεγάλες διακοπές.

Ο Raspberry Pi Pico 2 W χρησιμοποιείται στην παρούσα εργασία ως μικροελεγκτής και όχι ως υπολογιστής τύπου Raspberry Pi με Linux. Εκτελεί απευθείας το πρόγραμμα ελέγχου, επικοινωνεί με τον MPU6050 μέσω I²C, παράγει τα σήματα STEP και DIR και υποστηρίζει την ασύρματη επικοινωνία BLE. Ο προγραμματισμός πραγματοποιήθηκε στο Arduino IDE, ώστε η ανάπτυξη και η δοκιμή του κώδικα να γίνονται με ενιαίο περιβάλλον.

1.4 Δυναμική σταθεροποίηση και αδρανειακές μονάδες μέτρησης

Η ισορροπία ενός ρομπότ δύο τροχών είναι πρόβλημα δυναμικής σταθεροποίησης. Όταν το σώμα αποκλίνει από την κατακόρυφη θέση, η βαρύτητα αυξάνει την απόκλιση. Ο ελεγκτής πρέπει να μετακινήσει τους τροχούς προς την κατεύθυνση της πτώσης, ώστε ο άξονας στήριξης να επανέλθει κάτω από το κέντρο βάρους. Η διαδικασία επαναλαμβάνεται συνεχώς και σχηματίζει κλειστό βρόχο ελέγχου.

Για την εφαρμογή του βρόχου ελέγχου απαιτείται συνεχής εκτίμηση της κλίσης του ρομπότ. Στην παρούσα κατασκευή η εκτίμηση αυτή πραγματοποιείται με τη χρήση αδρανειακής μονάδας μέτρησης (IMU), ενώ η επεξεργασία των μετρήσεων και ο έλεγχος της ισορροπίας υλοποιούνται στον Raspberry Pi Pico 2 W.

Η θέση του IMU πάνω στο πλαίσιο πρέπει να παραμένει σταθερή, καθώς η μετακίνηση ή περιστροφή του αισθητήρα μπορεί να επηρεάσει την αντιστοίχιση των αξόνων μέτρησης με την πραγματική κίνηση του ρομπότ. Για τον λόγο αυτό, η βαθμονόμηση και η επιλογή των κατάλληλων προσήμων πραγματοποιήθηκαν αφού ο αισθητήρας στερεώθηκε στην τελική του θέση.

1.4.1 IMUs

Οι αδρανειακές μονάδες μέτρησης (Inertial Measurement Units – IMUs) είναι ηλεκτρονικά συστήματα που χρησιμοποιούνται για τη μέτρηση της κίνησης και του προσανατολισμού ενός σώματος. Μία τυπική IMU περιλαμβάνει τριαξονικό επιταχυνσιόμετρο και τριαξονικό γυροσκόπιο, ενώ ορισμένα συστήματα διαθέτουν επιπλέον μαγνητόμετρο [9], [11].

Το επιταχυνσιόμετρο μπορεί να χρησιμοποιηθεί για την εκτίμηση της κλίσης ως προς τη διεύθυνση της βαρύτητας. Ωστόσο, οι μετρήσεις του επηρεάζονται από κραδασμούς και από τις γραμμικές επιταχύνσεις της κατασκευής. Αντίθετα, το γυροσκόπιο μετρά τη γωνιακή ταχύτητα και παρέχει γρήγορη απόκριση στις μεταβολές του προσανατολισμού, αλλά η ολοκλήρωση της μέτρησής του με την πάροδο του χρόνου μπορεί να οδηγήσει σε συσσώρευση σφάλματος (drift) [9], [11].

Για τον περιορισμό των παραπάνω αδυναμιών χρησιμοποιείται συνδυασμός των δύο αισθητήρων. Στην παρούσα κατασκευή οι μετρήσεις του MPU6050 συνδυάστηκαν μέσω συμπληρωματικού φίλτρου (complementary filter). Το γυροσκόπιο χρησιμοποιείται κυρίως για τη γρήγορη παρακολούθηση των μεταβολών της κλίσης, ενώ το επιταχυνσιόμετρο παρέχει μακροχρόνια αναφορά ως προς τη βαρύτητα. Η εκτιμώμενη γωνία χρησιμοποιείται στη συνέχεια ως βασική είσοδος του PID ελεγκτή.

Ο MPU6050 που χρησιμοποιήθηκε στην εργασία αποτελεί IMU έξι αξόνων, καθώς περιλαμβάνει τρεις άξονες επιτάχυνσης και τρεις άξονες γωνιακής ταχύτητας. Λόγω του μικρού μεγέθους, του χαμηλού κόστους και της δυνατότητας ψηφιακής επικοινωνίας μέσω I²C, είναι κατάλληλος για πειραματικές εφαρμογές ενσωματωμένων συστημάτων και ρομποτικής. Η αναλυτική συνδεσμολογία, η βαθμονόμηση και η χρήση του στη συγκεκριμένη κατασκευή παρουσιάζονται στα επόμενα κεφάλαια.

Κεφάλαιο 2ο: Σχεδιασμός και Κατασκευή

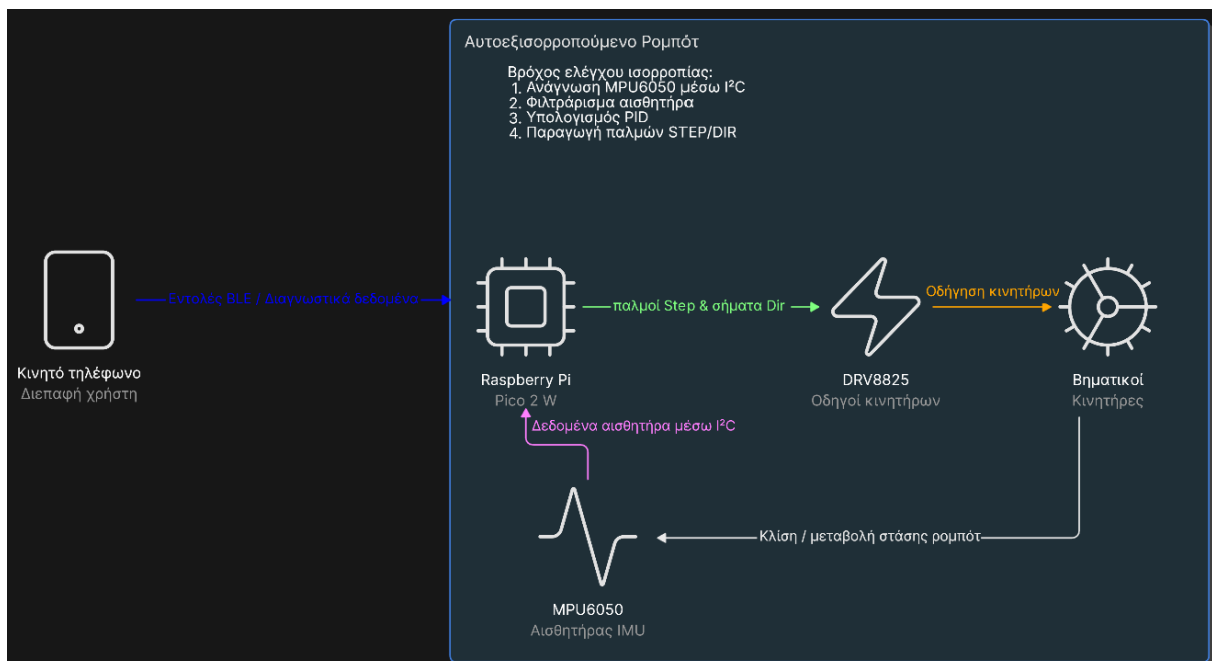
2.1 Γενική αρχιτεκτονική και λειτουργικές απαιτήσεις

Το σύστημα που κατασκευάστηκε είναι ένα αυτοεξισορροπούμενο ρομπότ δύο τροχών. Ο MPU6050 μετρά την επιτάχυνση και τη γωνιακή ταχύτητα, ο Raspberry Pi Pico 2 W υπολογίζει τη γωνία και την έξοδο του PID, και οι DRV8825 κινούν τους βηματικούς κινητήρες. Η αντίδραση επαναλαμβάνεται συνεχώς: όταν το ρομπότ γέρνει προς τα εμπρός, οι τροχοί κινούνται προς τα εμπρός· όταν γέρνει προς τα πίσω, κινούνται προς τα πίσω.

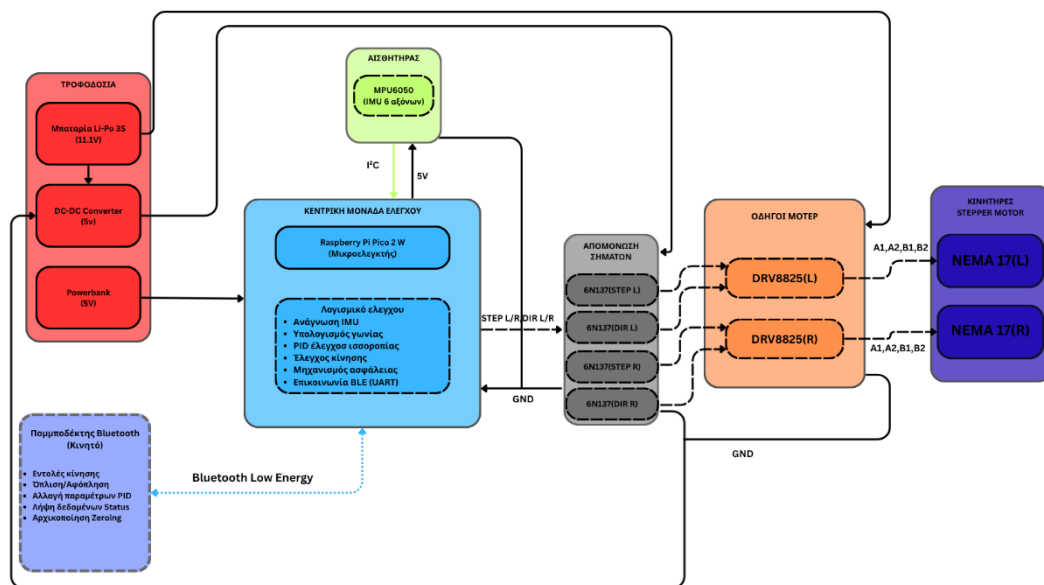
Η πρώτη απαίτηση ήταν η γρήγορη ανανέωση του βρόχου ελέγχου. Η ανάγνωση του αισθητήρα, το φιλτράρισμα, ο υπολογισμός του σφάλματος και η ενημέρωση της ταχύτητας των κινητήρων έπρεπε να εκτελούνται χωρίς μεγάλες καθυστερήσεις. Παράλληλα, οι παλμοί STEP έπρεπε να παράγονται με σταθερό ρυθμό.

Η δεύτερη απαίτηση ήταν η δυνατότητα ρύθμισης χωρίς συνεχή αλλαγή του προγράμματος. Για τον λόγο αυτό προστέθηκαν εντολές για τη γωνία μηδενισμού, τις παραμέτρους PID, την ενίσχυση των κινητήρων, τα πρόσημα και τις αντιστροφές φοράς. Η τρίτη απαίτηση ήταν η ασφαλής ενεργοποίηση. Μετά την εκκίνηση το ρομπότ παραμένει ανενεργό και ο έλεγχος ξεκινά μόνο όταν ο χρήστης στείλει την εντολή ARM.

Τέλος, προβλέφθηκε ασύρματη επικοινωνία με κινητό τηλέφωνο. Το BLE χρησιμοποιήθηκε τόσο για τον χειρισμό όσο και για τη ρύθμιση του συστήματος κατά τις δοκιμές. Η λειτουργία αυτή ήταν ιδιαίτερα χρήσιμη, επειδή επέτρεψε την αλλαγή παραμέτρων χωρίς καλώδιο USB και χωρίς νέα μεταφόρτωση του κώδικα.



Σχήμα 2.1 Γενική αρχιτεκτονική και βρόχος ελέγχου του αυτοεξισορροπούμενου ρομπότ.



Σχήμα 2.2 Block διάγραμμα του συστήματος.

2.2 Περιορισμοί και τεχνικές προκλήσεις

Η βασική δυσκολία προκύπτει από την ίδια τη μηχανική μορφή του ρομπότ. Οι δύο τροχοί διορθώνουν την πτώση μόνο στο επίπεδο της εμπρός-πίσω κίνησης. Μια πλάγια ασυμμετρία, ένας στραβός τροχός ή μεγαλύτερο βάρος στη μία πλευρά δεν μπορεί να διορθωθεί αποτελεσματικά από τον ίδιο ελεγκτή.

Το κέντρο βάρους επηρέασε επίσης τη συμπεριφορά. Τα βαριά στοιχεία δεν μπορούσαν να τοποθετηθούν τυχαία, επειδή η θέση τους μεταβάλλει τον τρόπο με τον οποίο πέφτει το σώμα. Στην τελική διάταξη το powerbank τοποθετήθηκε χαμηλά και κεντρικά, ανάμεσα στους κινητήρες, ενώ η μπαταρία των κινητήρων και τα ηλεκτρονικά στοιχεία κατανομήθηκαν στα επίπεδα του πλαισίου. Η επιλογή έγινε μέσα από δοκιμές και όχι μόνο με βάση τον διαθέσιμο χώρο.

Η ανάγνωση της γωνίας αποτέλεσε δεύτερη κρίσιμη πρόκληση. Ο MPU6050 επηρεάζεται από θόρυβο και κραδασμούς, ενώ η κατακόρυφη τοποθέτησή του απαιτούσε έλεγχο του σωστού άξονα και του πρόσημου. Αν η γωνία αυξανόταν προς την αντίθετη φορά από την πραγματική κλίση, οι κινητήρες ενίσχυαν την πτώση.

Τέλος, η χρήση βηματικών κινητήρων απαιτούσε σταθερή παραγωγή παλμών και σωστή ρύθμιση ρεύματος. Οι κινητήρες έπρεπε να έχουν αρκετή ροπή για να προλαβαίνουν την πτώση, χωρίς όμως οι οδηγοί να υπερθερμαίνονται. Οι παράμετροι του PID δεν μπορούσαν να ρυθμιστούν ανεξάρτητα από αυτή τη μηχανική και ηλεκτρική συμπεριφορά.

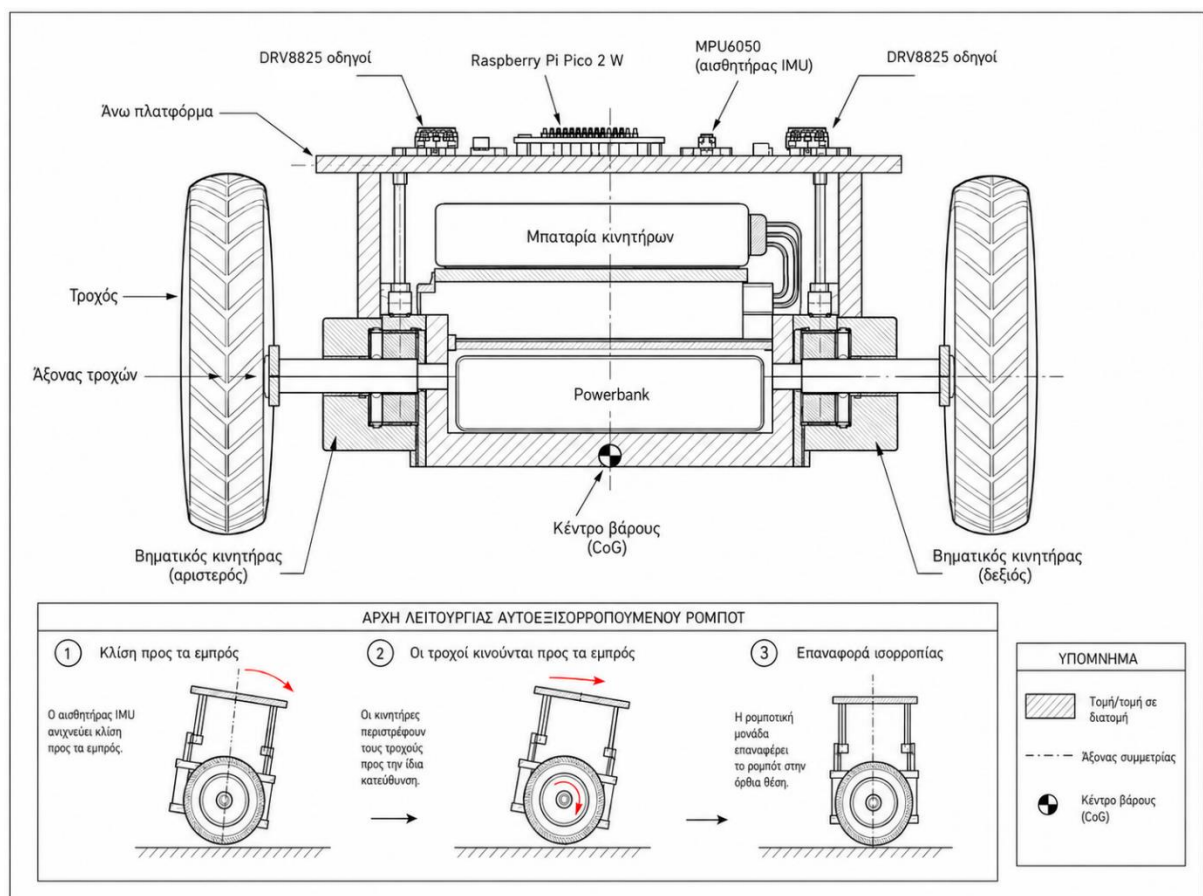
2.3 Μηχανικός σχεδιασμός και κατανομή βάρους

Το πλαίσιο αποτελείται από δύο επίπεδα που συνδέονται με αποστάτες. Οι βηματικοί κινητήρες και οι τροχοί βρίσκονται στο κάτω μέρος, συμμετρικά ως προς τον κεντρικό άξονα. Τα ηλεκτρονικά κυκλώματα τοποθετήθηκαν σε προσβάσιμες θέσεις, ώστε να μπορούν να ελεγχθούν και να τροποποιηθούν κατά τη διάρκεια των δοκιμών.

Η ακαμψία του πλαισίου και η σταθερή στερέωση των εξαρτημάτων αποδείχθηκαν πιο σημαντικές από ό,τι φαινόταν στην αρχική συναρμολόγηση. Οι συνεχείς μικροκινήσεις του ρομπότ μπορούσαν να μετακινήσουν το powerbank, τις μπαταρίες ή κάποια πλακέτα. Κάθε τέτοια μετακίνηση άλλαζε στιγμιαία το κέντρο βάρους και έκανε δυσκολότερη τη ρύθμιση του PID.

Για τη συγκράτηση των πηγών τροφοδοσίας χρησιμοποιήθηκαν δεματικά καλωδίων. Οι πλακέτες και ορισμένα βοηθητικά τμήματα σταθεροποιήθηκαν με διαφανή κολλητική ταινία ή κόλλα. Η λύση αυτή δεν έχει τη μόνιμη ακαμψία μιας ειδικά σχεδιασμένης βάσης, ήταν όμως πρακτική για το πειραματικό πρωτότυπο και επέτρεψε γρήγορες αλλαγές στη διάταξη.

Οι τροχοί πρέπει να είναι καλά στερεωμένοι στους άξονες των κινητήρων και να έχουν επαρκή πρόσφυση. Αν ένας τροχός ολισθαίνει, η εντολή του ελεγκτή δεν μετατρέπεται σε αντίστοιχη μετακίνηση της βάσης. Αντίστοιχα, διαφορετική μηχανική αντίσταση μεταξύ αριστερού και δεξιού τροχού μπορεί να προκαλέσει απόκλιση κατά την ευθύγραμμη κίνηση.



Σχήμα 2.3 Σχέδιο βασικής κατασκευής.

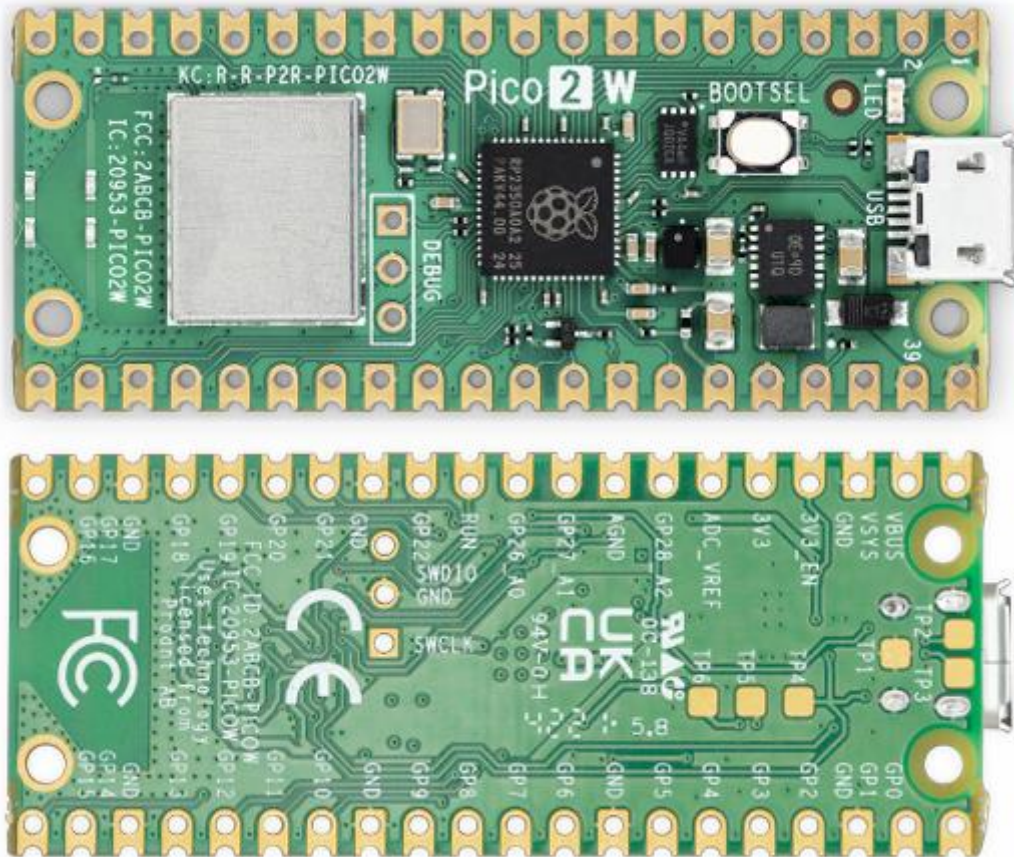
2.4 Βασικά ηλεκτρονικά εξαρτήματα

2.4.1 Raspberry Pi pico 2 W

2.4.1.1 Βασικά τεχνικά χαρακτηριστικά

Ο Raspberry Pi Pico 2 W (Σχήμα 2.4) είναι πλακέτα μικροελεγκτή που βασίζεται στο ολοκληρωμένο RP2350. Η πλακέτα διαθέτει μνήμη flash 4 MB και μπορεί να λειτουργήσει με δύο πυρήνες Arm

Cortex-M33 ή RISC-V Hazard3, με συχνότητα έως 150 MHz. Παρέχει επίσης περιφερειακά επικοινωνίας, μεταξύ των οποίων δύο μονάδες UART, δύο ελεγκτές I²C, δύο ελεγκτές SPI και δεκαέξι κανάλια PWM. Επιπλέον, το RP2350 διαθέτει τρία υποσυστήματα προγραμματιζόμενων εισόδων και εξόδων, τα οποία περιλαμβάνουν συνολικά δώδεκα state machines και μπορούν να χρησιμοποιηθούν για λειτουργίες ακριβούς χρονισμού [12].



Σχήμα 2.4 Raspberry Pi Pico 2 W.

Τα χαρακτηριστικά αυτά είναι κατάλληλα για τη συγκεκριμένη κατασκευή, επειδή η λειτουργία του ρομπότ απαιτεί την ταυτόχρονη εκτέλεση διαφορετικών εργασιών. Ο μικροελεγκτής πρέπει να διαβάζει τον MPU6050 μέσω I²C, να υπολογίζει τη γωνία κλίσης, να εκτελεί τον PID ελεγκτή, να παράγει τα σήματα STEP και DIR και να διαχειρίζεται την επικοινωνία BLE. Η διαθέσιμη υπολογιστική ισχύς και τα ανεξάρτητα περιφερειακά επιτρέπουν την εκτέλεση αυτών των λειτουργιών από μία πλακέτα, χωρίς να απαιτείται δεύτερος μικροελεγκτής.

Η πλακέτα εκθέτει 26 από τους διαθέσιμους ακροδέκτες GPIO του RP2350. Οι GPIO0 έως GPIO22 λειτουργούν ως ψηφιακές εισόδους ή εξόδους, ενώ οι GPIO26 έως GPIO28 μπορούν να χρησιμοποιηθούν είτε ως ψηφιακές γραμμές είτε ως αναλογικές εισόδους. Η λογική τάση όλων των ακροδεκτών εισόδου και εξόδου είναι σταθερά 3,3 V. Για τον λόγο αυτό, κάθε εξωτερικό κύκλωμα που συνδέεται απευθείας στον Pico πρέπει να είναι συμβατό με λογική 3,3 V ή να παρεμβάλλεται κατάλληλο κύκλωμα προσαρμογής [12].

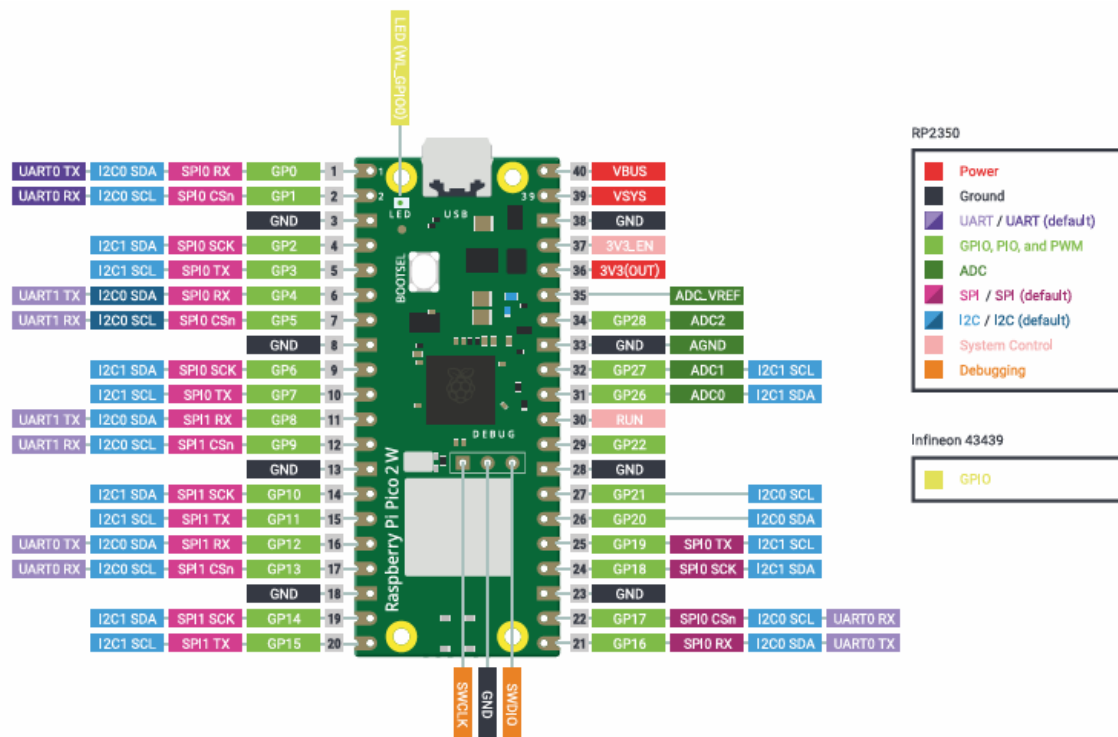
Πίνακας 2.1 Βασικά χαρακτηριστικά του Raspberry Pi Pico 2 W.

Χαρακτηριστικό	Τιμή/περιγραφή	Χρήση στην κατασκευή
Μικροελεγκτής	RP2350	Εκτέλεση του προγράμματος ελέγχου
Μνήμη flash	4 MB	Αποθήκευση του προγράμματος
Μέγιστη συχνότητα	Έως 150 MHz	Υπολογισμός γωνίας και PID
I ² C	2 ελεγκτές	Επικοινωνία με MPU6050
GPIO	26 διαθέσιμοι ακροδέκτες	STEP, DIR και βοηθητικά σήματα
Τάση GPIO	3,3 V	Καθορισμός στάθμης ψηφιακών σημάτων
Ασύρματη επικοινωνία	Wi-Fi και Bluetooth 5.2	Έλεγχος μέσω κινητού
USB	Micro-USB	Τροφοδοσία και προγραμματισμός

2.4.1.2 Διάταξη ακροδεκτών και χρησιμοποιούμενες συνδέσεις

Η πλακέτα διαθέτει σαράντα κύριους ακροδέκτες σε δύο σειρές, με απόσταση 2,54 mm μεταξύ τους. Η διάταξη αυτή επιτρέπει την τοποθέτηση σε breadboard ή σε διάτρητη πλακέτα, όπως έγινε κατά τις αρχικές δοκιμές της κατασκευής. Στο pinout εμφανίζονται για κάθε GPIO οι εναλλακτικές λειτουργίες UART, SPI, I²C και PWM. Παρέχονται επίσης αρκετοί ακροδέκτες γείωσης, ώστε να περιορίζονται οι ηλεκτρομαγνητικές παρεμβολές και η αλληλεπίδραση μεταξύ γειτονικών σημάτων [12].

Στην κατασκευή χρησιμοποιήθηκαν οι GP4 και GP5 για την επικοινωνία I²C με τον MPU6050. Οι GP16 και GP17 χρησιμοποιήθηκαν για τα σήματα STEP και DIR του αριστερού κινητήρα, ενώ οι GP18 και GP19 χρησιμοποιήθηκαν για τα αντίστοιχα σήματα του δεξιού κινητήρα. Οι τέσσερις γραμμές ελέγχου δεν οδηγήθηκαν απευθείας στους DRV8825, αλλά πρώτα πέρασαν από τους οπτοζεύκτες 6N137.



Σχήμα 2.5 Διάταξη ακροδεκτών του Raspberry Pi Pico 2 W.

Η εικόνα (Σχήμα 2.5) του datasheet δείχνει καθαρά ότι οι GP4 και GP5 μπορούν να χρησιμοποιηθούν για I²C, ενώ οι GP16 έως GP19 είναι διαθέσιμοι ως ψηφιακοί ακροδέκτες γενικού σκοπού.

2.4.1.3 Τροφοδοσία της πλακέτας

Ο Pico 2 W διαθέτει μετατροπέα τύπου buck-boost, ο οποίος δημιουργεί την απαιτούμενη τάση των 3,3 V από διαφορετικές πηγές τροφοδοσίας. Η είσοδος VSYS μπορεί να δεχθεί τάση από 1,8 έως 5,5 V, ενώ η είσοδος VBUS προορίζεται για την ονομαστική τάση των 5 V της θύρας USB. Η πλακέτα μπορεί επομένως να τροφοδοτηθεί είτε από τη θύρα micro-USB είτε από εξωτερική πηγή που συνδέεται στο VSYS [12].

Στην κατασκευή επιλέχθηκε η τροφοδοσία μέσω powerbank και της θύρας USB. Η επιλογή αυτή διαχώρισε την τροφοδοσία του μικροελεγκτή και του αισθητήρα από την τροφοδοσία των κινητήρων. Με αυτόν τον τρόπο, οι απότομες μεταβολές ρεύματος στις περιελίξεις των βηματικών κινητήρων δεν εφαρμόζονταν απευθείας στην τροφοδοσία του Pico 2 W.

Ο ακροδέκτης 3V3 αποτελεί την έξοδο του ενσωματωμένου μετατροπέα της πλακέτας. Σύμφωνα με το datasheet, μπορεί να χρησιμοποιηθεί για την τροφοδοσία εξωτερικών κυκλωμάτων, με τη σύσταση το πρόσθετο φορτίο να παραμένει κάτω από 300 mA. Στην τελική κατασκευή ο MPU6050 τροφοδοτήθηκε με 3,3 V από τον ακροδέκτη 36, 3V3(OUT), του Raspberry Pi Pico 2 W.

2.4.1.4 Ενσωματωμένη ασύρματη επικοινωνία και τοποθέτηση της κεραίας

Η ασύρματη λειτουργία του Pico 2 W υλοποιείται μέσω του ολοκληρωμένου CYW43439. Το υποσύστημα υποστηρίζει Wi-Fi 4 στη ζώνη των 2,4 GHz και Bluetooth 5.2. Για το Bluetooth

υποστηρίζονται τόσο ρόλοι Bluetooth Low Energy Central και Peripheral όσο και Bluetooth Classic [12].

Στην παρούσα εφαρμογή ο Pico 2 W χρησιμοποιήθηκε ως BLE Peripheral. Το κινητό τηλέφωνο λειτουργούσε ως Central και συνδεόταν στην υπηρεσία που παρείχε το ρομπότ. Μετά τη σύνδεση μπορούσε να στέλνει εντολές όπως ARM, ZERO, PID και JOY, καθώς και να λαμβάνει διαγνωστικές πληροφορίες.

Η κεραία βρίσκεται στο κάτω άκρο της πλακέτας (Σχήμα 2.6). Το datasheet αναφέρει ότι η παρουσία υλικών πολύ κοντά στην περιοχή της κεραίας μπορεί να μειώσει την απόδοσή της και ότι η πλακέτα δεν πρέπει να εγκλωβίζεται μέσα σε μεταλλικό περίβλημα. Η παρατήρηση αυτή είναι σχετική με τη μηχανική τοποθέτηση του Pico στο ρομπότ: η περιοχή της κεραίας πρέπει να παραμένει όσο γίνεται ελεύθερη από μπαταρίες, μεταλλικά στηρίγματα και πυκνή καλωδίωση [12].



Σχήμα 2.6 Θέση του Raspberry Pi Pico 2 W και περιοχή της ενσωματωμένης κεραίας στην τελική κατασκευή.

2.4.2 Αισθητήρας MPU6050

Ο MPU6050 τοποθετήθηκε κατακόρυφα και στερεώθηκε ώστε να ακολουθεί την κλίση του κύριου σώματος. Η θέση του δεν επιλέχθηκε μόνο για ευκολία καλωδίωσης. Έπρεπε να παραμένει ακίνητος σε σχέση με το πλαίσιο, διαφορετικά οι μετρήσεις θα περιέγραφαν και τη δική του μετακίνηση.

Οι ακατέργαστες τιμές του αισθητήρα βαθμονομήθηκαν με ξεχωριστό πρόγραμμα. Η βαθμονόμηση πραγματοποιήθηκε αφού ο MPU6050 είχε τοποθετηθεί στην τελική του θέση, επειδή τα offsets εξαρτώνται από τον συγκεκριμένο αισθητήρα και τον προσανατολισμό του.

2.4.2.1 Βασικά τεχνικά χαρακτηριστικά του MPU6050

Ο MPU6050 είναι ολοκληρωμένος αισθητήρας κίνησης έξι αξόνων, καθώς ενσωματώνει τριαξονικό επιταχυνσιόμετρο και τριαξονικό γυροσκόπιο. Για κάθε άξονα χρησιμοποιείται μετατροπέας αναλογικού σε ψηφιακό σήμα ανάλυσης 16 bit, επιτρέποντας την ανεξάρτητη καταγραφή της επιτάχυνσης και της γωνιακής ταχύτητας στους άξονες X, Y και Z. Ο αισθητήρας διαθέτει επίσης ενσωματωμένο Digital Motion Processor (DMP), αν και στη συγκεκριμένη κατασκευή η επεξεργασία της γωνίας πραγματοποιήθηκε από τον Raspberry Pi Pico 2 W και δεν χρησιμοποιήθηκε η λειτουργία DMP[16].

Το εύρος του γυροσκοπίου μπορεί να ρυθμιστεί στις τιμές ± 250 , ± 500 , ± 1000 ή ± 2000 $^{\circ}/s$, ενώ το επιταχυνσιόμετρο υποστηρίζει εύρη $\pm 2g$, $\pm 4g$, $\pm 8g$ και $\pm 16g$. Η δυνατότητα επιλογής διαφορετικού εύρους επιτρέπει την προσαρμογή της ευαισθησίας του αισθητήρα ανάλογα με τις απαιτήσεις της εφαρμογής. Στην παρούσα κατασκευή ο αισθητήρας χρησιμοποιήθηκε κυρίως για την εκτίμηση της κλίσης του ρομπότ στον άξονα εμπρός-πίσω[16].

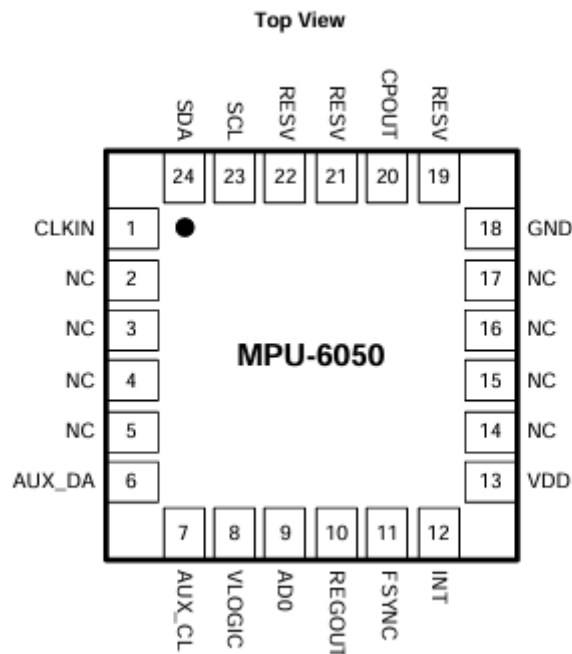
2.4.2.2 Τροφοδοσία και ψηφιακή επικοινωνία

Ο MPU6050 λειτουργεί με τάση τροφοδοσίας VDD από 2,375 V έως 3,46 V. Σε αντίθεση με τον MPU6000, ο MPU6050 διαθέτει ξεχωριστό ακροδέκτη VLOGIC, ο οποίος καθορίζει τα επίπεδα λογικής της διεπαφής I²C. Η τιμή του VLOGIC μπορεί να βρίσκεται από 1,71 V έως την τάση VDD, επιτρέποντας τη σύνδεση με μικροελεγκτές διαφορετικών λογικών επιπέδων.

Η επικοινωνία με τον Raspberry Pi Pico 2 W πραγματοποιήθηκε μέσω του διαύλου I²C. Οι γραμμές SCL και SDA του MPU6050 συνδέθηκαν αντίστοιχα στους GP5 και GP4 του Pico 2 W. Η χρήση I²C περιόρισε τον αριθμό των απαιτούμενων αγωγών και επέτρεψε την περιοδική ανάγνωση των μετρήσεων του αισθητήρα μέσα στον κύριο βρόχο ελέγχου[16].

2.4.2.3 Ακροδέκτες και pinout του MPU6050

Για τη συγκεκριμένη κατασκευή χρησιμοποιήθηκαν κυρίως οι ακροδέκτες τροφοδοσίας και οι γραμμές επικοινωνίας SCL και SDA (Σχήμα 2.7). Ο ακροδέκτης AD0 επιτρέπει την επιλογή του λιγότερο σημαντικού bit της διεύθυνσης I²C, ενώ ο ακροδέκτης INT μπορεί να χρησιμοποιηθεί για παραγωγή διακοπών από τον αισθητήρα. Οι λειτουργίες αυτές δεν ήταν απαραίτητες όλες στην τελική υλοποίηση, καθώς η ανάγνωση των δεδομένων γινόταν περιοδικά από τον κύριο βρόχο του προγράμματος[16].

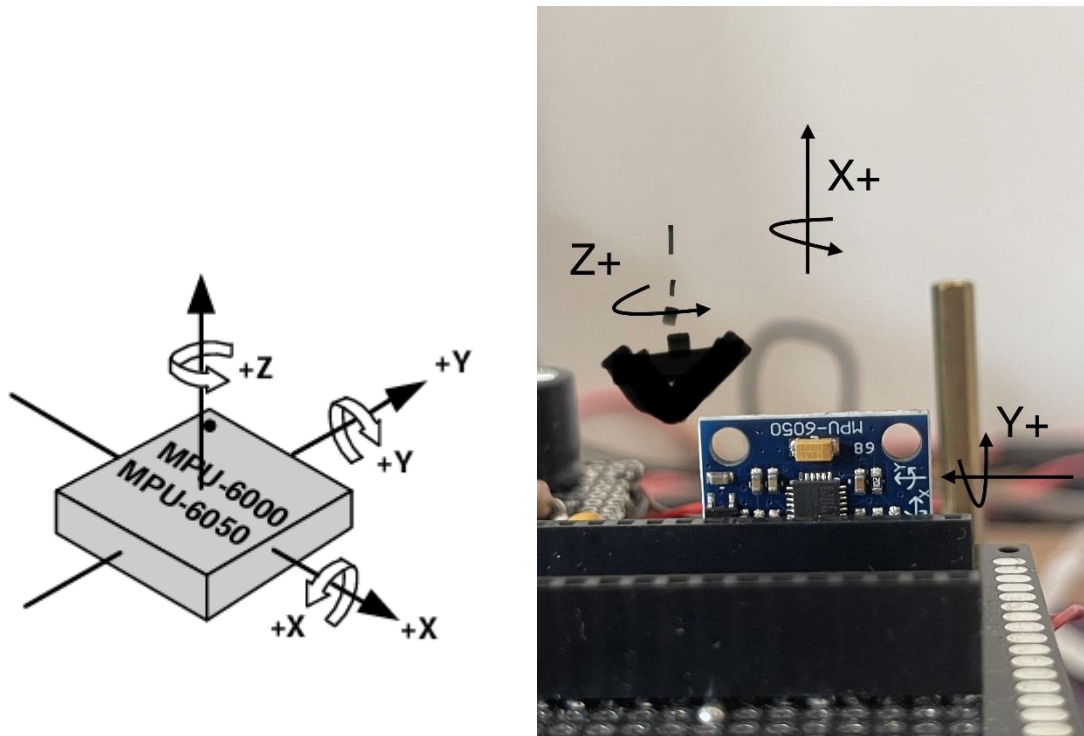


Σχήμα 2.7 Διάταξη ακροδεκτών και προσανατολισμός αξόνων του MPU6050.

2.4.2.4 Προσανατολισμός των αξόνων στην κατασκευή

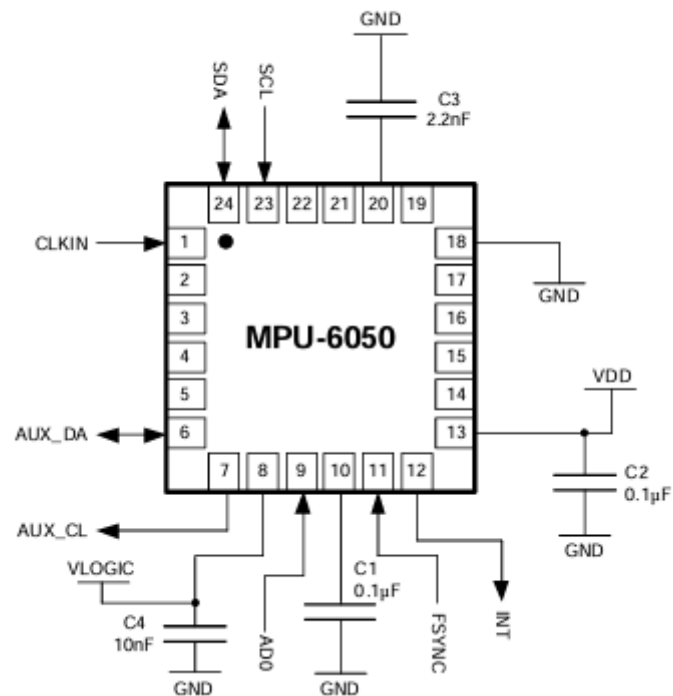
Ο προσανατολισμός των αξόνων του αισθητήρα ήταν ιδιαίτερα σημαντικός (Σχήμα 2.8), επειδή ο MPU6050 τοποθετήθηκε κατακόρυφα και όχι οριζόντια. Σύμφωνα με το datasheet, οι άξονες X, Y και Z έχουν συγκεκριμένο προσανατολισμό ως προς το περίβλημα του αισθητήρα. Μετά την τοποθέτηση της πλακέτας στο ρομπότ πραγματοποιήθηκε πρακτικός έλεγχος, ώστε να προσδιοριστεί ποιος άξονας αντιστοιχούσε στην κίνηση εμπρός-πίσω και ποιο πρόσημο έπρεπε να χρησιμοποιηθεί στον κώδικα [16].

Για τον έλεγχο, το ρομπότ μετακινήθηκε χειροκίνητα προς τις δύο κατευθύνσεις και παρακολουθήθηκε η μεταβολή της υπολογιζόμενης γωνίας στο Serial Monitor. Με αυτόν τον τρόπο επιβεβαιώθηκαν οι παράμετροι που σχετίζονταν με την επιλογή άξονα και το πρόσημο πριν ενεργοποιηθεί ο PID ελεγκτής.



Σχήμα 2.8 Προσανατολισμός MPU-6050.

2.4.2.5 Τυπικό κύκλωμα σύνδεσης



Σχήμα 2.9 Τυπικό κύκλωμα λειτουργίας του MPU6050.

Το προτεινόμενο κύκλωμα του κατασκευαστή περιλαμβάνει πυκνωτές αποσύζευξης στις γραμμές τροφοδοσίας και στις εσωτερικές γραμμές του αισθητήρα(Σχήμα 2.9)[16]. Η breakout πλακέτα που

χρησιμοποιήθηκε στην κατασκευή ενσωμάτων ήδη τα απαραίτητα βοηθητικά στοιχεία, επομένως η σύνδεση προς το κύριο κύκλωμα περιορίστηκε κυρίως στην τροφοδοσία και στις γραμμές SCL και SDA.

2.4.3 Βηματικοί κινητήρες και οδηγοί DRV8825

2.4.3.1 Εσωτερική λειτουργία του DRV8825

Ο DRV8825 είναι ολοκληρωμένος οδηγός για διπολικούς βηματικούς κινητήρες. Περιλαμβάνει δύο γέφυρες H με τρανζίστορ ισχύος, κύκλωμα μέτρησης και ρύθμισης του ρεύματος, καθώς και εσωτερικό indexer για τον μικροβηματισμό. Ο οδηγός δέχεται απλές εντολές STEP και DIR, ενώ η εσωτερική λογική αναλαμβάνει την κατάλληλη ενεργοποίηση των δύο τυλιγμάτων του κινητήρα [14].

Η προτεινόμενη περιοχή τάσης τροφοδοσίας των γεφυρών είναι από 8,2 έως 45 V. Το ολοκληρωμένο υποστηρίζει θεωρητικό ρεύμα εξόδου έως 2,5 A ανά γέφυρα, όμως η πραγματικά επιτρεπόμενη τιμή εξαρτάται από τη θερμική διαχείριση, την πλακέτα και τις συνθήκες λειτουργίας. Για τον λόγο αυτό, η τιμή των 2,5 A δεν πρέπει να παρουσιαστεί ως το ρεύμα που χρησιμοποιήθηκε στην κατασκευή.

Στην εργασία χρησιμοποιήθηκαν δύο ανεξάρτητοι οδηγοί, ένας για κάθε κινητήρα. Με αυτόν τον τρόπο ο αριστερός και ο δεξιός τροχός μπορούσαν να λαμβάνουν διαφορετικές ταχύτητες, τόσο για τη διατήρηση της ισορροπίας όσο και για την εκτέλεση στροφών.

2.4.3.2 Ομάδες ακροδεκτών του DRV8825

Οι ακροδέκτες του DRV8825 μπορούν να χωριστούν σε τέσσερις βασικές κατηγορίες.

- **Ακροδέκτες τροφοδοσίας και ισχύος**

Οι VMA και VMB τροφοδοτούν τις δύο γέφυρες H και πρέπει να συνδέονται στην ίδια πηγή τροφοδοσίας κινητήρων. Οι ακροδέκτες GND αποτελούν το σημείο αναφοράς του ολοκληρωμένου. Οι CP1, CP2 και VCP σχετίζονται με το εσωτερικό charge pump που χρησιμοποιείται για την οδήγηση των άνω MOSFET .

- **Ακροδέκτες κινητήρα**

Οι AOUT1 και AOUT2 συνδέονται στο πρώτο τύλιγμα του διπολικού κινητήρα, ενώ οι BOUT1 και BOUT2 συνδέονται στο δεύτερο τύλιγμα. Η σωστή αναγνώριση των ζευγών των πηνίων είναι απαραίτητη. Αν οι αγωγοί διαφορετικών πηνίων αναμειχθούν, ο κινητήρας μπορεί να παράγει θόρυβο και κραδασμούς χωρίς να περιστρέφεται κανονικά [14].

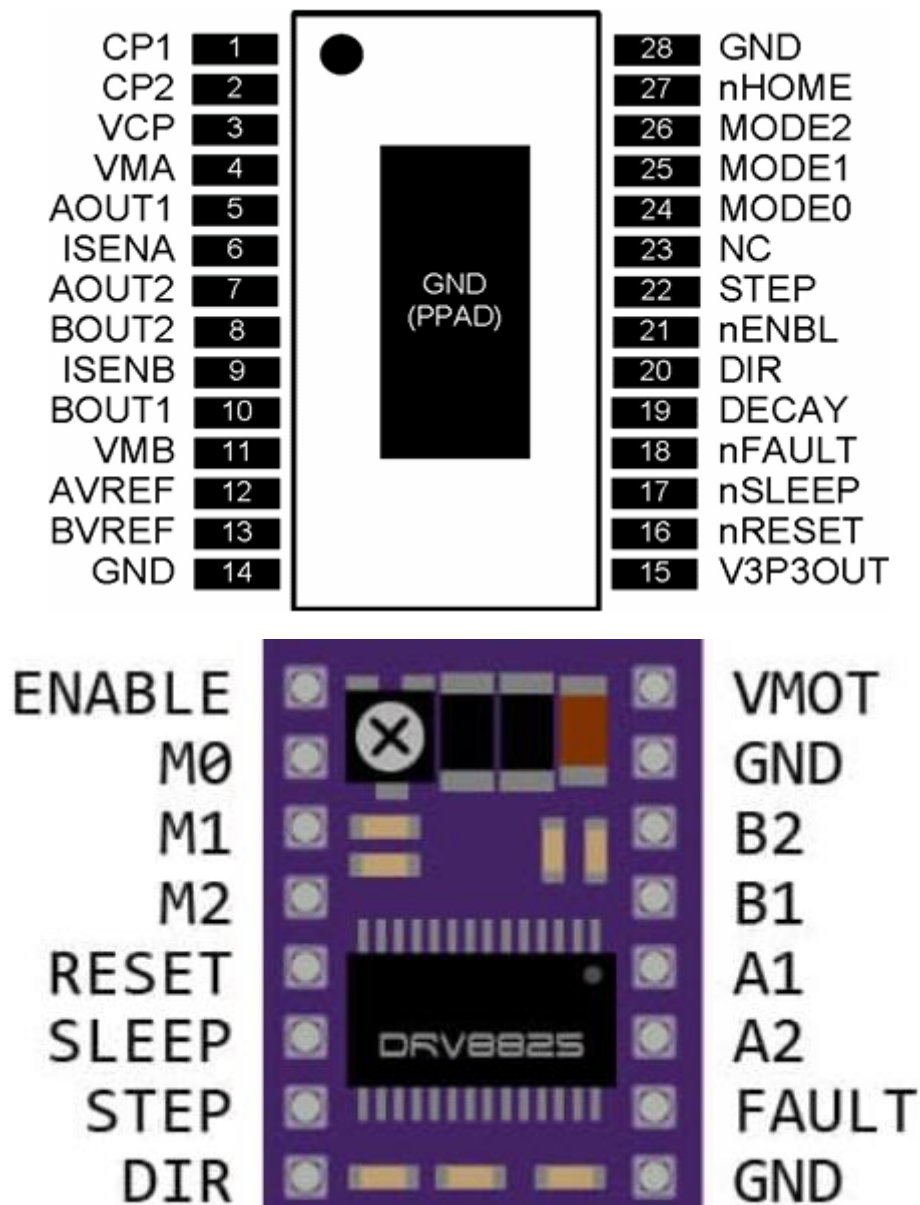
- **Ακροδέκτες ελέγχου**

Οι σημαντικότεροι ακροδέκτες παρουσιάζονται παρακάτω (Πίνακας 2.2)(Σχήμα 2.10).

Πίνακας 2.2 Ακροδέκτες ελέγχου.

Ακροδέκτης	Λειτουργία
STEP	Κάθε ανερχόμενο μέτωπο προχωρά τον indexer κατά ένα βήμα ή μικροβήμα
DIR	Καθορίζει την κατεύθυνση κίνησης του indexer
MODE0–MODE2	Επιλέγουν το επίπεδο μικροβηματισμού

nENBL	Ενεργοποιεί ή απενεργοποιεί τις εξόδους
nRESET	Επαναφέρει την εσωτερική λογική
nSLEEP	Θέτει τον οδηγό σε κατάσταση χαμηλής κατανάλωσης
DECAY	Επιλέγει τον τρόπο μείωσης του ρεύματος
nFAULT	Ενεργοποιείται σε κατάσταση σφάλματος



Σχήμα 2.10 Διάταξη και λειτουργίες των ακροδεκτών του DRV8825.

2.4.3.3 Έλεγχος μέσω STEP και DIR

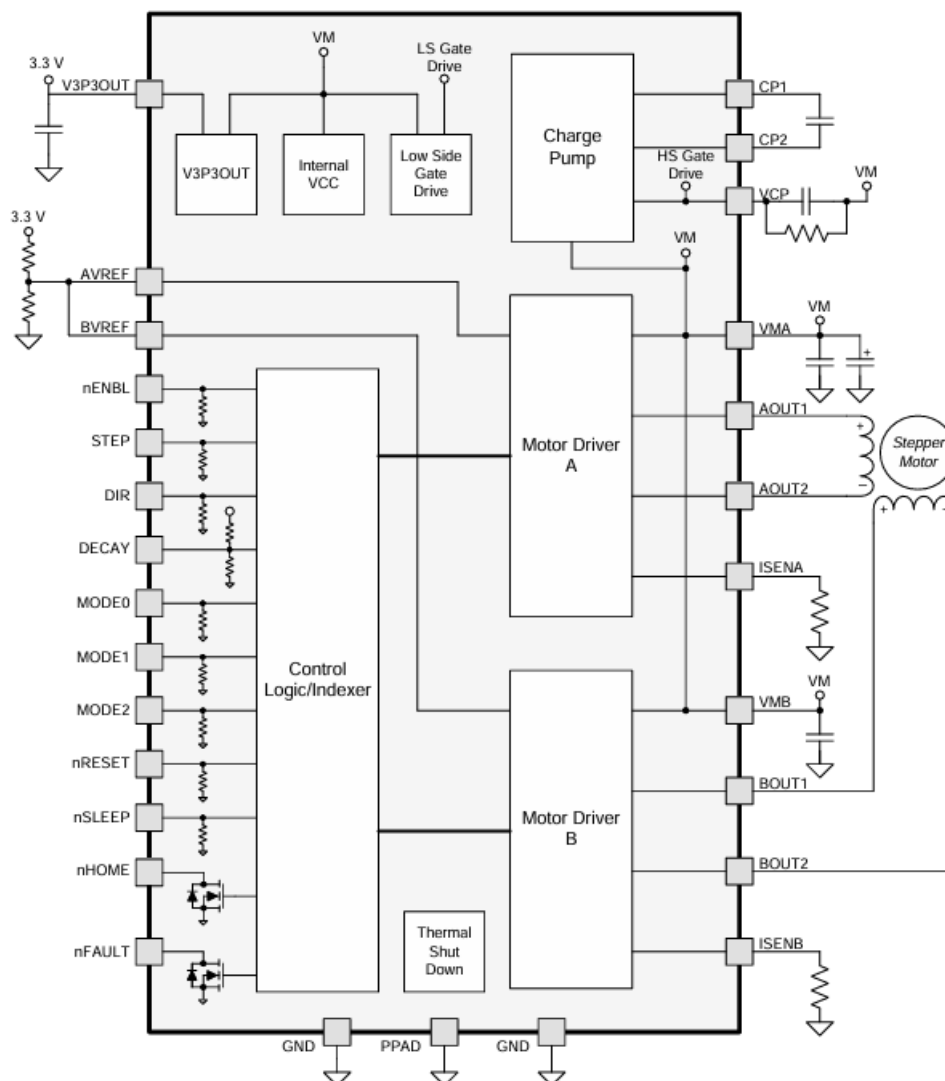
Η διεπαφή STEP/DIR επιτρέπει τον έλεγχο του κινητήρα με δύο βασικά ψηφιακά σήματα. Σε κάθε ανερχόμενο μέτωπο του STEP, ο εσωτερικός indexer μεταβαίνει στην επόμενη κατάσταση του πίνακα

μικροβηματισμού (Σχήμα 2.11). Η λογική κατάσταση του DIR καθορίζει προς ποια κατεύθυνση θα κινηθεί μέσα στον πίνακα και, κατά συνέπεια, τη φορά περιστροφής του κινητήρα [14].

Η ταχύτητα του κινητήρα δεν καθορίζεται από το πλάτος του παλμού, αλλά κυρίως από τη συχνότητα επανάληψης των παλμών STEP. Όσο μικρότερο είναι το χρονικό διάστημα μεταξύ διαδοχικών παλμών, τόσο υψηλότερη είναι η ζητούμενη ταχύτητα. Το datasheet ορίζει ελάχιστη διάρκεια υψηλού και χαμηλού επιπέδου για το STEP ίση με 1,9 μ s και μέγιστη συχνότητα εισόδου 250 kHz. Οι παλμοί της συγκεκριμένης εφαρμογής ήταν σημαντικά χαμηλότερης συχνότητας, καθώς περιορίζονταν από τις δυνατότητες των κινητήρων και της μηχανικής κατασκευής.

Η ανεξάρτητη παραγωγή παλμών για τους δύο οδηγούς επιτρέπει:

- ίση ταχύτητα των τροχών για ευθύγραμμη κίνηση,
- αντίθετη μεταβολή των ταχυτήτων για στροφή,
- συνεχή προσαρμογή της ταχύτητας και της φοράς από τον PID ελεγκτή.



Σχήμα 2.11 Εσωτερικό λειτουργικό διάγραμμα του DRV8825.

2.4.3.4 Επιλογή μικροβηματισμού

Οι ακροδέκτες MODE0, MODE1 και MODE2 καθορίζουν τον αριθμό των μικροβημάτων ανά πλήρες βήμα(Πίνακας 2.3).

Πίνακας 2.3 Επιλογές μικροβηματισμού.

MODE2	MODE1	MODE0	Λειτουργία
0	0	0	Πλήρες βήμα
0	0	1	1/2 βήματος
0	1	0	1/4 βήματος
0	1	1	1/8 βήματος
1	0	0	1/16 βήματος
1	0	1	1/32 βήματος
1	1	0	1/32 βήματος
1	1	1	1/32 βήματος

Στην τελική κατασκευή επιλέχθηκε λειτουργία μικροβηματισμού 1/16. Σύμφωνα με τη διάταξη του Πίνακα 2.3, οι ακροδέκτες ρυθμίστηκαν σε MODE2=HIGH, MODE1=LOW και MODE0=LOW. Η επιλογή αυτή παρείχε ομαλότερη περιστροφή των κινητήρων σε σχέση με μεγαλύτερο βήμα και χρησιμοποιήθηκε στις τελικές δοκιμές ισορροπίας και κίνησης.

Ο μικροβηματισμός προσφέρει πιο ομαλή μεταβολή του ρεύματος στα δύο πηνία και μειώνει τις απότομες αλλαγές ροπής. Αυτό είναι χρήσιμο σε ένα αυτοεξισορροπούμενο ρομπότ, επειδή ο PID μεταβάλλει συνεχώς την ταχύτητα και τη φορά των τροχών.

2.4.3.5 Ρύθμιση ρεύματος και τάσης αναφοράς

Το ρεύμα των πηνίων ρυθμίζεται μέσω κυκλώματος PWM current chopping. Όταν το ρεύμα σε ένα τύλιγμα φτάσει την προκαθορισμένη τιμή, η αντίστοιχη γέφυρα διακόπτει ή ανακυκλοφορεί προσωρινά το ρεύμα μέχρι την επόμενη περίοδο PWM. Η διαδικασία επιτρέπει τον έλεγχο του ρεύματος ακόμα και όταν η τάση τροφοδοσίας είναι υψηλότερη από την ονομαστική τάση του κινητήρα [14].

Η μέγιστη τιμή ρεύματος υπολογίζεται από τη σχέση:

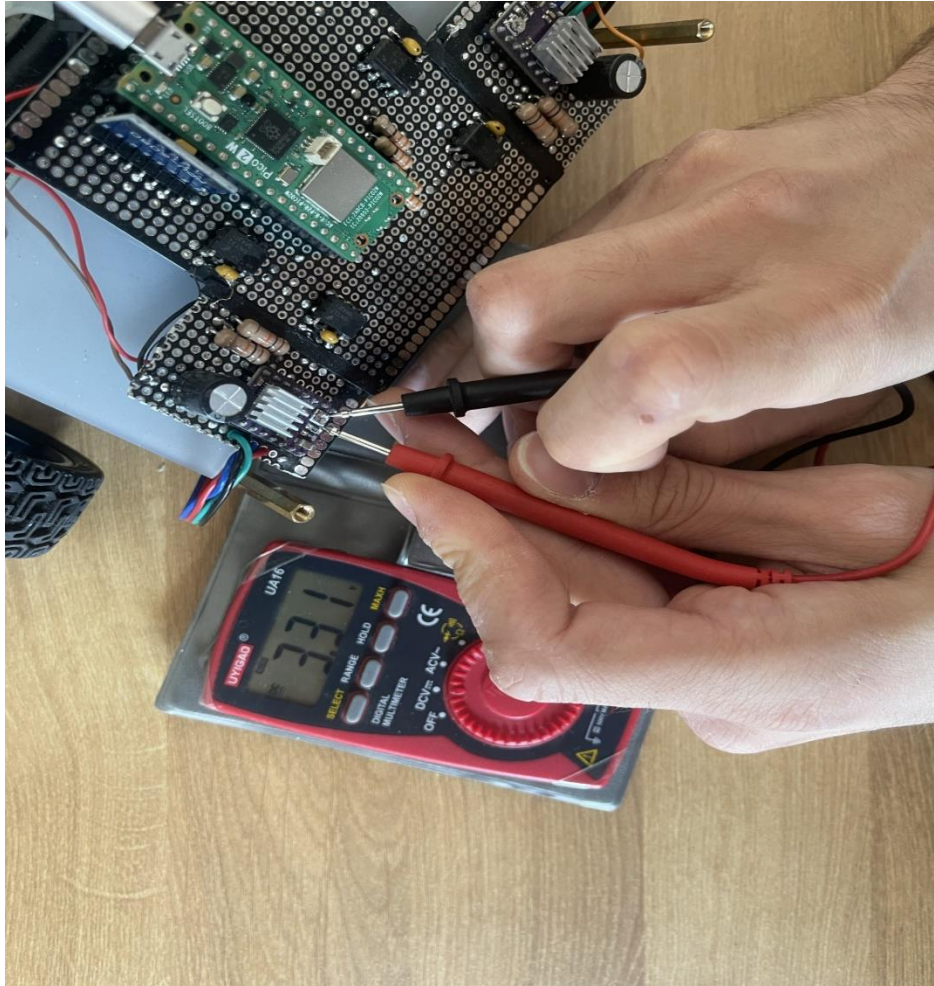
$$I_{FS} = \frac{V_{REF}}{5R_{SENSE}} \quad (2.1)$$

όπου:

- I_{FS} είναι το μέγιστο ρεύμα του πηνίου,
- V_{REF} είναι η τάση αναφοράς,
- R_{SENSE} είναι η αντίσταση μέτρησης ρεύματος της πλακέτας.

Η τιμή του R_{SENSE} πρέπει να διαβαστεί από την πραγματική carrier πλακέτα ή το δικό της datasheet, στην προκειμένη περίπτωση ο κωδικός της smd είναι R100 που αντιστοιχεί σε 0,1Ω.

Στις δοκιμές η V_{ref} μετρήθηκε με πολύμετρο και ρυθμίστηκε σταδιακά (Σχήμα 2.12). Όταν η τιμή ήταν πολύ χαμηλή, οι κινητήρες δεν παρήγαγαν επαρκή ροπή, ενώ η αύξησή της οδηγούσε σε μεγαλύτερο ρεύμα και αυξημένη θέρμανση. Η τελική ρύθμιση αποτέλεσε συμβιβασμό ανάμεσα στην απαιτούμενη ροπή και στη θερμική συμπεριφορά του κυκλώματος. Στην τελική ρύθμιση η V_{REF} μετρήθηκε περίπου στα 3,31 V και η τιμή αυτή χρησιμοποιήθηκε στις τελικές δοκιμές. Στην κεντρική πλακέτα που χρησιμοποιήθηκε, η αντίσταση μέτρησης ρεύματος φέρει τον κωδικό R100, ο οποίος αντιστοιχεί σε 0,1 Ω .



Σχήμα 2.12 Ρύθμιση V_{ref} .

2.4.3.6 RESET, SLEEP και προστασίες

Ο ακροδέκτης nRESET είναι ενεργός σε λογικό μηδέν. Όταν ενεργοποιείται, επαναφέρει την εσωτερική λογική και τον indexer στην αρχική θέση και απενεργοποιεί τις γέφυρες «H». Ο ακροδέκτης nSLEEP πρέπει να βρίσκεται σε υψηλή στάθμη για κανονική λειτουργία. Σε χαμηλή στάθμη θέτει τον οδηγό σε κατάσταση χαμηλής κατανάλωσης και σταματά τις εσωτερικές λειτουργίες. Μετά την έξοδο από την κατάσταση sleep απαιτείται περίπου 1 ms πριν δοθεί νέος παλμός STEP [14].

Στην τελική κατασκευή οι ακροδέκτες nRESET και nSLEEP κάθε DRV8825 συνδέθηκαν μεταξύ τους και διατηρήθηκαν σε υψηλή λογική στάθμη κατά την κανονική λειτουργία. Με αυτόν τον τρόπο ο

οδηγός παρέμενε εκτός κατάστασης reset και sleep, ώστε να μπορεί να δέχεται συνεχώς τα σήματα STEP και DIR. Ο ακροδέκτης nENBL συνδέθηκε στη γείωση.

Ο DRV8825 περιλαμβάνει προστασία από:

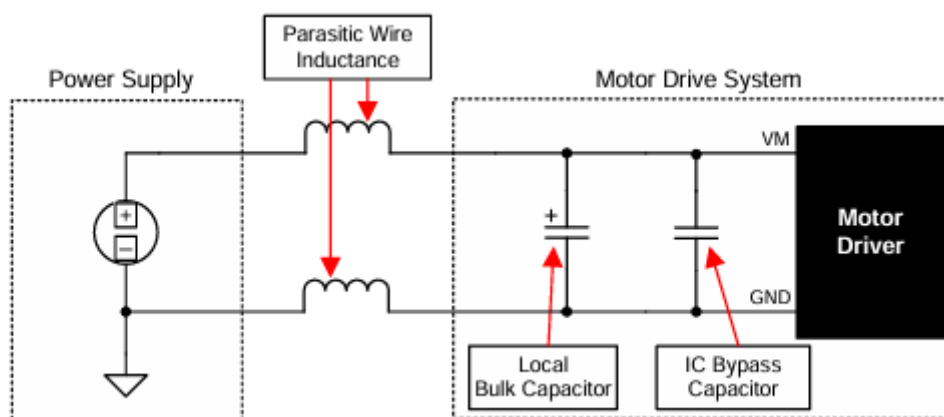
- πτώση τάσης τροφοδοσίας,
- υπερένταση και βραχυκύκλωμα,
- υπερβολική θερμοκρασία.

Σε κατάσταση σφάλματος ο ακροδέκτης nFAULT οδηγείται σε χαμηλή στάθμη. Η προστασία αυτή δεν καταργεί την ανάγκη σωστής ρύθμισης ρεύματος και ψύξης, αλλά μειώνει τον κίνδυνο μόνιμης βλάβης του ολοκληρωμένου.

2.4.3.7 Πυκνωτές αποσύζευξης και προστασία τροφοδοσίας

Το datasheet προτείνει την τοποθέτηση κεραμικών πυκνωτών 0,1 μF κοντά στους ακροδέκτες τροφοδοσίας VMA και VMB. Απαιτείται επίσης τοπικός πυκνωτής μεγαλύτερης χωρητικότητας, ώστε να απορροφά απότομες απαιτήσεις ρεύματος και ενέργεια που επιστρέφει από τον κινητήρα κατά τις μεταβολές της κίνησης (Σχήμα 2.13). Η κατάλληλη χωρητικότητα εξαρτάται από την πηγή τροφοδοσίας, την καλωδίωση, το ρεύμα εκκίνησης και την αποδεκτή κυμάτωση τάσης [14].

Στην κατασκευή χρησιμοποιήθηκε ηλεκτρολυτικός πυκνωτής 100 μF κοντά στην είσοδο τροφοδοσίας των οδηγών. Η θέση του είναι σημαντική: πρέπει να βρίσκεται κοντά στους οδηγούς και όχι μόνο κοντά στην μπαταρία, ώστε να περιορίζεται η επίδραση της επαγωγής των καλωδίων.



Σχήμα 2.13 Χρήση τοπικού πυκνωτή αποσύζευξης και πυκνωτή εξομάλυνσης στην τροφοδοσία οδηγού κινητήρα.

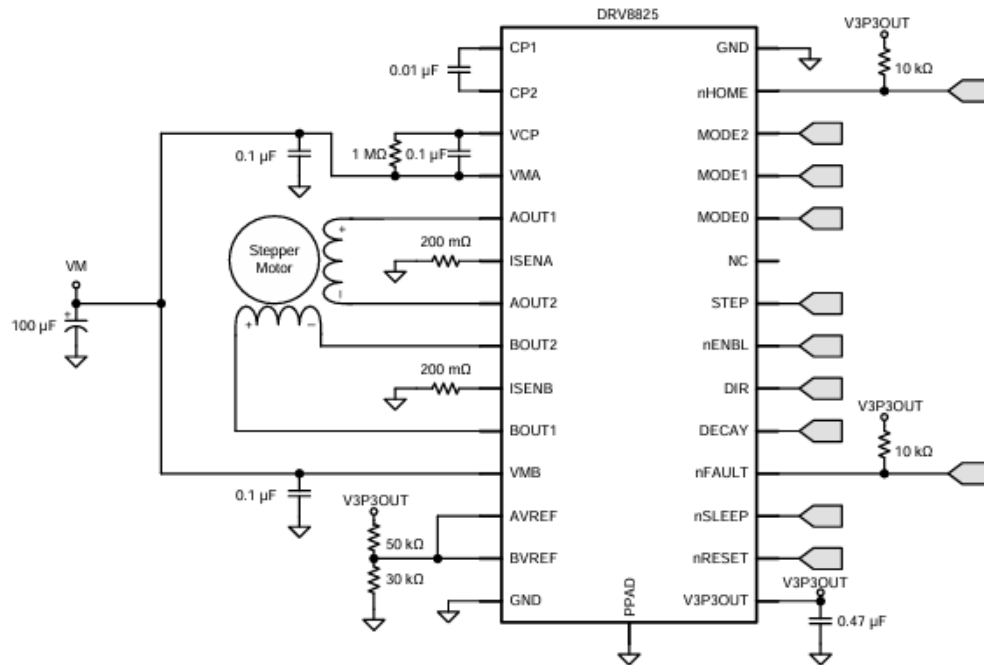
2.4.3.8 Τυπική συνδεσμολογία DRV8825

Στο datasheet του DRV8825 παρουσιάζεται ένα τυπικό κύκλωμα εφαρμογής, στο οποίο φαίνονται οι βασικές συνδέσεις που απαιτούνται για την οδήγηση ενός διπολικού βηματικού κινητήρα. Η τροφοδοσία του κινητήρα εφαρμόζεται στους ακροδέκτες VMOT, ενώ τα δύο πηνία του κινητήρα συνδέονται στις εξόδους AOUT1, AOUT2, BOUT1 και BOUT2. Τα σήματα STEP και DIR χρησιμοποιούνται αντίστοιχα για την παραγωγή των βημάτων και τον καθορισμό της φοράς περιστροφής.

Στο ίδιο κύκλωμα εμφανίζονται οι ακροδέκτες MODE0, MODE1 και MODE2 για την επιλογή του μικροβηματισμού, καθώς και οι ακροδέκτες nRESET, nSLEEP και nENBL που σχετίζονται με την

ενεργοποίηση και την κατάσταση λειτουργίας του οδηγού. Παράλληλα, χρησιμοποιούνται αντιστάσεις μέτρησης ρεύματος και η τάση αναφοράς VREF για τον καθορισμό του μέγιστου ρεύματος των πηνίων.

Κοντά στην τροφοδοσία του οδηγού προβλέπονται πυκνωτές αποσύζευξης και εξομάλυνσης, ώστε να περιορίζονται οι απότομες μεταβολές της τάσης που προκαλούνται κατά τη λειτουργία του κινητήρα.



Σχήμα 2.14 Τυπικό κύκλωμα εφαρμογής του οδηγού DRV8825.

Η συνδεσμολογία αυτή χρησιμοποιήθηκε ως τεχνική αναφορά κατά τον σχεδιασμό της κατασκευής (Σχήμα 2.14). Η τελική υλοποίηση διαφοροποιήθηκε κυρίως ως προς τη διαδρομή των σημάτων STEP και DIR, καθώς μεταξύ του Raspberry Pi Pico 2 W και των οδηγών παρεμβλήθηκαν οι οπτοζεύκτες 6N137. Η πραγματική συνδεσμολογία της κατασκευής παρουσιάζεται αναλυτικά στην Ενότητα 2.5.

2.4.4 Οπτοζεύκτες

Οι οπτοζεύκτες 6N137 παρεμβλήθηκαν στα σήματα STEP και DIR. Η πλευρά εισόδου συνδέθηκε με τον μικροελεγκτή και η πλευρά εξόδου με τους οδηγούς. Η διάταξη αυτή απομονώνει τα ψηφιακά σήματα από το κύκλωμα ισχύος και περιορίζει την επιστροφή θορύβου προς τον Pico 2 W.

Η έξοδος κάθε οπτοζεύκτη χρειάζεται σωστή τροφοδοσία και pull-up σύνδεση. Κατά τις δοκιμές, ένας κινητήρας που περιστρεφόταν αλλά δεν άλλαζε φορά αποτελούσε ένδειξη ότι το STEP έφτανε στον οδηγό, ενώ η διαδρομή του DIR χρειαζόταν έλεγχο.

2.4.4.1 Δομή και αρχή λειτουργίας

Ο 6N137 είναι οπτοζεύκτης ενός καναλιού με ονομαστικό ρυθμό μετάδοσης 10 MBs. Στην είσοδό του περιλαμβάνει φωτοεκπέμπον LED, ενώ στην πλευρά εξόδου διαθέτει φωτοανιχνευτή και ψηφιακή λογική υψηλής ταχύτητας. Η μεταφορά του σήματος γίνεται οπτικά, χωρίς άμεση αγωγίμη σύνδεση ανάμεσα στην είσοδο και την έξοδο [15].

Η διάταξη αυτή χρησιμοποιήθηκε για την απομόνωση των σημάτων STEP και DIR. Η πλευρά εισόδου συνδέθηκε στους ακροδέκτες του Raspberry Pi Pico 2 W μέσω αντιστάσεων περιορισμού ρεύματος, ενώ η πλευρά εξόδου συνδέθηκε στις ψηφιακές εισόδους των DRV8825.

Η ηλεκτρική απομόνωση περιορίζει τη δημιουργία βρόχων γείωσης και τη μεταφορά διαταραχών από το κύκλωμα ισχύος προς τον μικροελεγκτή. Ο κατασκευαστής αναφέρει ως βασικές εφαρμογές του 6N137 την απομόνωση ψηφιακών διαύλων, τη διεπαφή με συστήματα μικροεπεξεργαστών και την εξάλειψη βρόχων γείωσης.

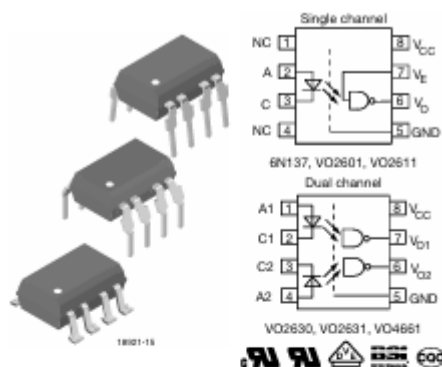
2.4.4.2 Διάταξη ακροδεκτών

Ο 6N137 διατίθεται σε περίβλημα οκτώ ακροδεκτών. Η λειτουργία τους παρουσιάζεται στον ακόλουθο πίνακα(Πίνακας 2.4).

Πίνακας 2.4 Διάταξη ακροδεκτων.

Ακροδέκτης	Ονομασία	Λειτουργία
1	NC	Δεν συνδέεται
2	Anode	Άνοδος του LED εισόδου
3	Cathode	Κάθοδος του LED εισόδου
4	NC	Δεν συνδέεται
5	GND	Γείωση πλευράς εξόδου
6	V_O	Ψηφιακή έξοδος
7	V_E	Είσοδος ενεργοποίησης
8	V_{CC}	Τροφοδοσία πλευράς εξόδου

High Speed Optocoupler, Single and Dual, 10 MBd



DESCRIPTION

The 6N137, VO2601, and VO2611 are single channel 10 MBd optocouplers utilizing a high efficient input LED coupled with an integrated optical photodiode IC detector. The detector has an open drain NMOS-transistor output, providing less leakage compared to an open collector Schottky clamped transistor output. The VO2630, VO2631, and VO4661 are dual channel 10 MBd optocouplers. For the single channel type, an enable function on pin 7 allows the detector to be strobed. The internal shield provides a guaranteed common mode transient immunity of 5 kV/μs for the VO2601 and 15 kV/μs for the VO2611 and VO4661. The use of a 0.1 μF bypass capacitor connected between pin 5 and 8 is recommended.

FEATURES

- Choice of CMR performance of 15 kV/μs, 5 kV/μs, and 100 V/μs
- High speed: 10 MBd typical
- + 5 V CMOS compatibility
- Pure tin leads
- Guaranteed AC and DC performance over temperature: - 40 °C to + 100 °C temperature range
- Meets IEC 60068-2-42 (SO₂) and IEC 60068-2-43 (H₂S) requirements
- Low input current capability: 5 mA
- Material categorization: For definitions of compliance please see www.vishay.com/doc?99912



RoHS
COMPLIANT

APPLICATIONS

- Microprocessor system interface
- PLC, ATE input/output isolation
- Computer peripheral interface
- Digital fieldbus isolation: CC-link, DeviceNet, profibus, SDS
- High speed A/D and D/A conversion
- AC plasma display panel level shifting
- Multiplexed data transmission
- Digital control power supply
- Ground loop elimination

AGENCY APPROVALS

- UL1577, file no. E52744 system code H, double protection
- cUL - file no. E52744, equivalent to CSA bulletin 5A
- DIN EN 60747-5-5 (VDE 0884-5) available with option 1
- BSI IEC 60950
- CQC GB8898-2001

ORDERING INFORMATION													
V	O	2	6	0	1	-	X	0	#	#	T		
PART NUMBER						PACKAGE OPTION				TAPE AND REEL			
											Option 7		
AGENCY CERTIFIED/PACKAGE	CHANNELS 1						CHANNELS 2						
	CMR (V/μs)						CMR (V/μs)						
	1000		5000		15 000		1000		5000		15 000		
	BSI, UL, cUL		6N137		VO2601		VO2611		VO2630		VO2631		
	DIP-8		400 mil, option 6		6N137-X006		VO2601-X006		VO2611-X006		VO2630-X006		
	SMD-8, option 7		6N137-X007T		VO2601-X007T		VO2611-X007T		VO2630-X007T		VO2631-X007T		
	VDE, BSI, UL, cUL		1000		5000		15 000		1000		5000		
DIP-8, 400 mil, option 6		-		VO2601-X016		VO2611-X016		-		-			
SMD-8, option 7		-		VO2601-X017T		VO2611-X017T		-		VO2631-X017T			

Σχήμα 2.15 Εσωτερική δομή και διάταξη ακροδεκτών του οπτοζεύκτη 6N137.

Η εικόνα δείχνει επίσης καθαρά τη διαχωριστική περιοχή μεταξύ του LED εισόδου και του φωτοανιχνευτή εξόδου (Σχήμα 2.15).

2.4.4.3 Τάση τροφοδοσίας και ρεύμα εισόδου

Η πλευρά εξόδου του 6N137 πρέπει να τροφοδοτείται με τάση από 4,5 έως 5,5 V. Επομένως, το εξάρτημα του συγκεκριμένου datasheet είναι σχεδιασμένο για ονομαστική τροφοδοσία 5 V και όχι για απευθείας λειτουργία της εξόδου στα 3,3 V. Στην τελική κατασκευή η πλευρά εξόδου των 6N137 τροφοδοτήθηκε από τη σταθεροποιημένη έξοδο 5 V του DC-DC converter, ενώ η αντίσταση pull-up συνδέθηκε στην ίδια γραμμή. Η είσοδος κάθε οπτοζεύκτη οδηγήθηκε από GPIO του Raspberry Pi Pico 2 W μέσω αντίστασης 330 Ω.

Το προτεινόμενο ρεύμα υψηλής στάθμης του LED βρίσκεται μεταξύ 5 και 15 mA. Η αντίσταση εισόδου υπολογίζεται από τη σχέση:

$$R_{in} = \frac{V_{GPIO} - V_F}{I_F} \quad (2.2)$$

όπου V_F είναι η πτώση τάσης στο LED. Το datasheet δίνει τυπική πτώση περίπου 1,38 V και μέγιστη 1,70 V για ρεύμα 10 mA. Η πραγματική επιλογή της αντίστασης πρέπει να διατηρεί το ρεύμα μέσα στην επιτρεπτή περιοχή, χωρίς να υπερφορτώνει τον ακροδέκτη του Pico 2 W.

Για τάση εξόδου του Raspberry Pi Pico 2 W ίση με 3,3 V και θεωρώντας τυπική πτώση τάσης του LED του 6N137, επιλέχθηκε ρεύμα εισόδου περίπου 6 mA. Από τη σχέση προκύπτει αντίσταση περίπου 320 Ω. Για την κατασκευή χρησιμοποιήθηκε η πλησιέστερη τυποποιημένη τιμή των 330 Ω, η οποία οδηγεί σε ρεύμα περίπου 5,8 mA.

2.4.4.4 Έξοδος ανοικτού απαγωγού και αντίσταση pull-up

Η ψηφιακή έξοδος του 6N137 είναι τύπου open collector και απαιτεί εξωτερική αντίσταση pull-up. Αυτό σημαίνει ότι το εξάρτημα μπορεί να οδηγήσει ενεργά την έξοδο σε χαμηλή στάθμη, αλλά δεν μπορεί να δημιουργήσει μόνο του υψηλή στάθμη. Για τον λόγο αυτό απαιτείται εξωτερική αντίσταση pull-up προς την τάση τροφοδοσίας της εξόδου. Το datasheet προτείνει αντίσταση pull-up από 330 Ω έως 4 kΩ [15].

Η λογική λειτουργία είναι αντιστρέφουσα:

- όταν το LED είναι ενεργό και το ENABLE βρίσκεται σε υψηλή στάθμη, η έξοδος οδηγείται σε LOW,
- όταν το LED είναι σβηστό, η pull-up αντίσταση επαναφέρει την έξοδο σε HIGH,
- όταν το ENABLE είναι LOW, η έξοδος παραμένει HIGH ανεξάρτητα από την είσοδο.

Η αντιστρέφουσα λογική του 6N137 πρέπει να λαμβάνεται υπόψη κατά την υλοποίηση των σημάτων ελέγχου. Σε συνδυασμό με τη φορά περιστροφής των δύο κινητήρων και τη συγκεκριμένη καλωδίωση του συστήματος, αυτό οδήγησε στην ανάγκη κατάλληλης προσαρμογής της λογικής των σημάτων DIR, η οποία στον κώδικα υλοποιείται μέσω των μεταβλητών dirInvertL και dirInvertR.

2.4.4.5 Ταχύτητα απόκρισης και καταλληλότητα για STEP/DIR

Ο 6N137 έχει σχεδιαστεί για ψηφιακή μετάδοση έως 10 MBs. Οι χρόνοι καθυστέρησης διάδοσης είναι της τάξης μερικών δεκάδων νανοδευτερολέπτων, με μέγιστες τιμές περίπου 100 ns στις προτεινόμενες συνθήκες λειτουργίας. Οι χρόνοι αυτοί είναι πολύ μικρότεροι από τη διάρκεια των παλμών STEP που χρησιμοποιήθηκαν στην κατασκευή, επομένως ο οπτοζεύκτης μπορεί να μεταφέρει τα συγκεκριμένα σήματα χωρίς να αποτελεί τον βασικό περιοριστικό παράγοντα της συχνότητας.

Για σωστή λειτουργία χρειάζονται:

- κατάλληλο ρεύμα στο LED εισόδου,
- σταθερή τροφοδοσία 5 V στην πλευρά εξόδου,
- αντίσταση pull-up,

- σωστή κατάσταση του ακροδέκτη ENABLE,
- κοινό σημείο αναφοράς μόνο μέσα σε κάθε πλευρά του οπτοζεύκτη.

2.4.5 Τροφοδοσία και step-down converter

Η τροφοδοσία χωρίστηκε σε κύκλωμα λογικής και κύκλωμα κινητήρων. Ο Raspberry Pi Pico 2 W και ο αισθητήρας τροφοδοτήθηκαν από powerbank. Οι DRV8825 και οι βηματικοί κινητήρες χρησιμοποίησαν ξεχωριστή πηγή, επειδή οι μεταβολές του ρεύματος των κινητήρων μπορούν να προκαλέσουν πτώσεις τάσης και ηλεκτρικό θόρυβο.

Ο step-down converter χρησιμοποιήθηκε για τη δημιουργία χαμηλότερης σταθεροποιημένης τάσης στα βοηθητικά κυκλώματα που δεν μπορούσαν να συνδεθούν απευθείας στην τάση των κινητήρων. Κοντά στην τροφοδοσία των οδηγών τοποθετήθηκε πυκνωτής προστασίας, ώστε να περιορίζονται απότομες μεταβολές τάσης.

Πίνακας 2.5 Βασικά εξαρτήματα κατασκευής.

A/A	Εξάρτημα	Ποσότητα	Ρόλος στην κατασκευή
1	Raspberry Pi Pico 2 W	1	Κεντρική μονάδα ελέγχου του ρομπότ
2	MPU6050	1	Αισθητήρας κλίσης και κίνησης του ρομπότ
3	DRV8825 οδηγοί βηματικών κινητήρων	2	Οδήγηση των βηματικών κινητήρων μέσω σημάτων STEP/DIR
4	Βηματικοί κινητήρες	2	Κίνηση των τροχών για τη διατήρηση της ισορροπίας
5	Τροχοί	2	Μηχανική επαφή με το έδαφος και μετακίνηση του ρομπότ
6	Οπτοζεύκτες 6N137	4	Ηλεκτρική απομόνωση μεταξύ μικροελεγκτή και κυκλώματος οδήγησης
7	Step-down converter	1	Μετατροπή τάσης για την τροφοδοσία επιμέρους κυκλωμάτων
8	Μπαταρία κινητήρων	1	Τροφοδοσία των οδηγών και των βηματικών κινητήρων
9	Powerbank	1	Τροφοδοσία του Raspberry Pi Pico 2 W
11	Πλαίσιο / πλατφόρμες στήριξης	1	Μηχανική στήριξη των εξαρτημάτων

2.4.5.1 Μετατροπέας DC–DC LM2596

Ο μετατροπέας DC–DC που χρησιμοποιήθηκε στην κατασκευή βασίζεται στο ολοκληρωμένο LM2596 της Texas Instruments (Σχήμα 2.16). Πρόκειται για διακοπτικό μετατροπέα υποβιβασμού τάσης (step-down ή buck converter), ο οποίος μετατρέπει μία υψηλότερη συνεχή τάση εισόδου σε χαμηλότερη και σταθεροποιημένη τάση εξόδου. Ο LM2596 λειτουργεί με σταθερή συχνότητα μεταγωγής 150 kHz και μπορεί να παρέχει ρεύμα φορτίου έως 3 A υπό τις κατάλληλες συνθήκες λειτουργίας και θερμικής διαχείρισης. Η οικογένεια διατίθεται σε εκδόσεις σταθερής εξόδου 3,3 V, 5 V και 12 V, καθώς και σε ρυθμιζόμενη έκδοση [13].

Στην κατασκευή χρησιμοποιήθηκε η ρυθμιζόμενη έκδοση του μετατροπέα, ώστε η τάση εξόδου να μπορεί να προσαρμοστεί στις ανάγκες των βοηθητικών ηλεκτρονικών κυκλωμάτων. Η χρήση διακοπτικού μετατροπέα αντί μιας απλής γραμμικής ρύθμισης περιορίζει τις απώλειες που θα εμφανίζονταν κατά τη μετατροπή της υψηλότερης τάσης της μπαταρίας σε χαμηλότερη τάση.

LM2596 SIMPLE SWITCHER® Power Converter 150-kHz 3-A Step-Down Voltage Regulator

1 Features

- New product available:
 - [LMR51430 4.5 to 36-V, 3-A, 500-kHz and 1.1-MHz synchronous converter](#)
- For faster time to market:
 - [TLVM13630 3 to 36-V, 3-A, 200-kHz to 2.2-MHz power module](#)
- 3.3-V, 5-V, 12-V, and adjustable output versions
- Adjustable version output voltage range: 1.2-V to 37-V $\pm 4\%$ maximum over line and load conditions
- Available in TO-220 and TO-263 packages
- 3-A output load current
- Input voltage range up to 40 V
- Requires only four external components
- Excellent line and load regulation specifications
- 150-kHz fixed-frequency internal oscillator
- TTL shutdown capability
- Low power standby mode, I_Q , typically 80 μ A
- High efficiency
- Uses readily available standard inductors
- Thermal shutdown and current-limit protection
- Create a custom design using the LM2596 with the [WEBENCH® Power Designer](#)

2 Applications

- [Appliances](#)
- [Grid infrastructure](#)
- [EPOS](#)
- [Home theater](#)

3 Description

The LM2596 series of regulators are monolithic integrated circuits that provide all the active functions for a step-down (buck) switching regulator, capable of driving a 3-A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3 V, 5 V, 12 V, and an adjustable output version.

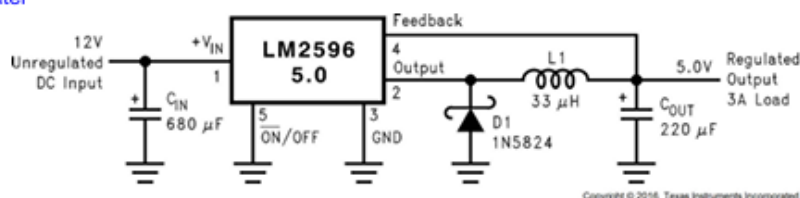
Requiring a minimum number of external components, these regulators are simple to use and include internal frequency compensation, and a fixed-frequency oscillator.

The LM2596 series operates at a switching frequency of 150 kHz, thus allowing smaller sized filter components than what can be required with lower frequency switching regulators. Available in a standard 5-pin TO-220 package with several different lead bend options, and a 5-pin TO-263 surface mount package.

Package Information

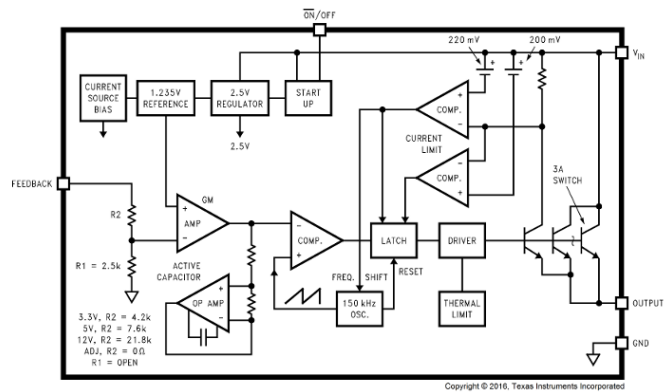
PART NUMBER	PACKAGE ⁽¹⁾	BODY SIZE (NOM)
LM2596	NDH (TO-220, 5)	14.986 mm $\times$ 10.16 mm
	KTT (TO-263, 5)	10.10 mm $\times$ 8.89 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.



(Fixed Output Voltage Versions)

Typical Application

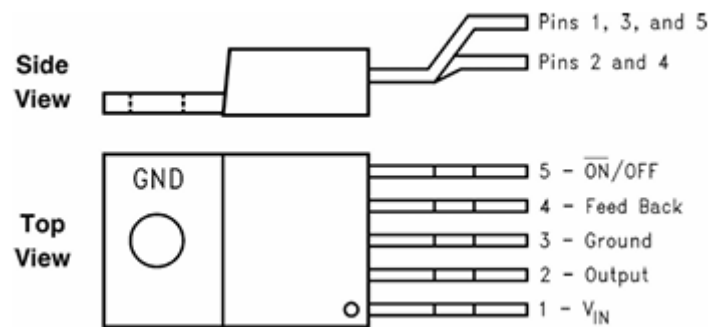


Σχήμα 2.16 Εσωτερικό λειτουργικό διάγραμμα του μετατροπέα LM2596.

2.4.5.2 Ακροδέκτες και βασική λειτουργία του LM2596

Το ολοκληρωμένο LM2596 διαθέτει πέντε βασικούς ακροδέκτες(Σχήμα 2.17). Ο ακροδέκτης VIN αποτελεί την είσοδο της τάσης τροφοδοσίας, ενώ ο ακροδέκτης Output συνδέεται στο διακοπτικό στάδιο του μετατροπέα. Ο Ground αποτελεί το κοινό σημείο αναφοράς και ο Feedback χρησιμοποιείται από τον εσωτερικό βρόχο ελέγχου για τη διατήρηση της επιθυμητής τάσης εξόδου. Ο πέμπτος ακροδέκτης, ON/OFF, επιτρέπει την ενεργοποίηση ή απενεργοποίηση του μετατροπέα μέσω λογικού σήματος.

Η λειτουργία του μετατροπέα βασίζεται στη γρήγορη διακοπτική ενεργοποίηση ενός εσωτερικού στοιχείου ισχύος. Η τάση που παράγεται στην έξοδο του διακοπτικού σταδίου εξομαλύνεται μέσω εξωτερικού πηνίου, διόδου και πυκνωτών. Ο εσωτερικός βρόχος ανάδρασης συγκρίνει συνεχώς την τάση εξόδου με την εσωτερική τάση αναφοράς και μεταβάλλει τον κύκλο λειτουργίας του διακοπτικού στοιχείου, ώστε η έξοδος να παραμένει σταθερή παρά τις μεταβολές της τάσης εισόδου ή του φορτίου[13].

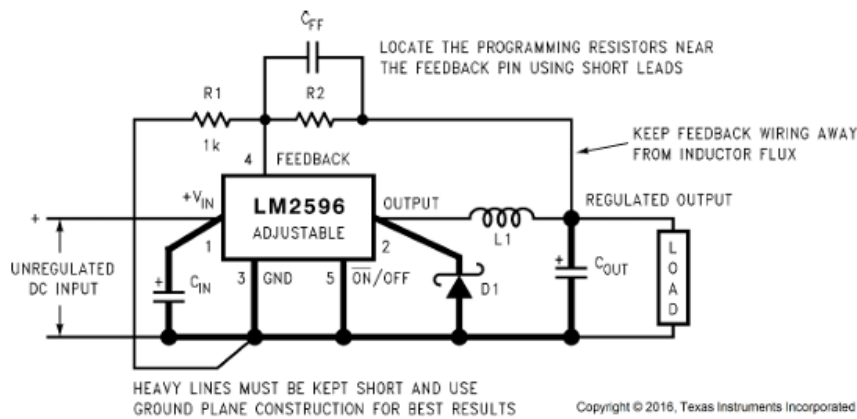


Σχήμα 2.17 Διάταξη και λειτουργία των ακροδεκτών του LM2596.

2.4.5.3 Ρύθμιση της τάσης εξόδου

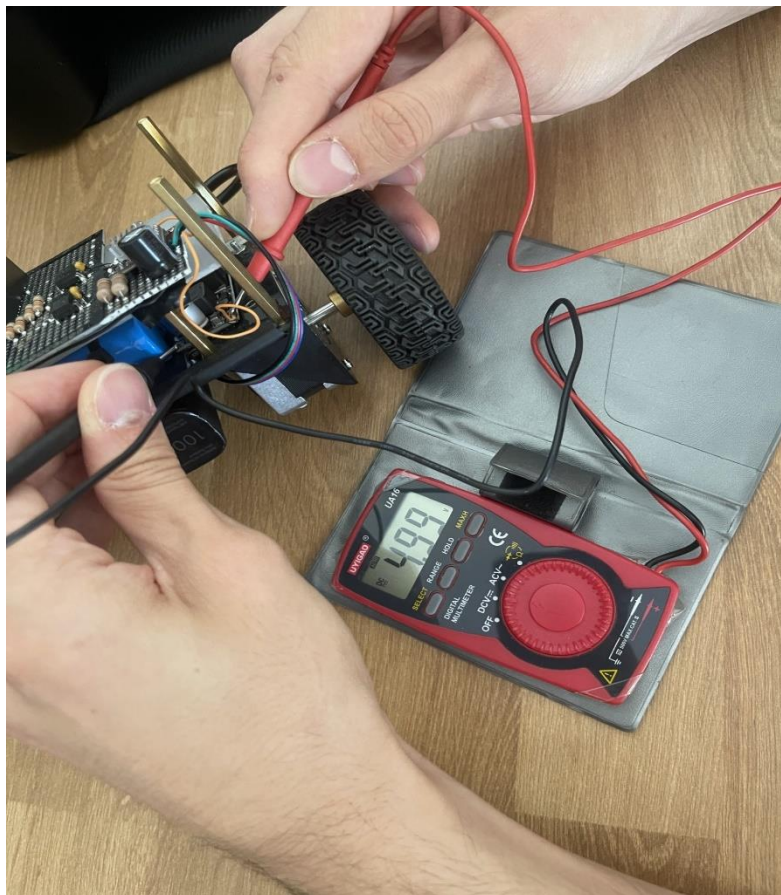
Στη ρυθμιζόμενη έκδοση του LM2596, η τάση εξόδου καθορίζεται από το δίκτυο ανάδρασης που συνδέεται στον ακροδέκτη Feedback(Σχήμα 2.18). Η εσωτερική τάση αναφοράς είναι περίπου 1,23 V και η επιθυμητή τάση εξόδου καθορίζεται από δύο αντιστάσεις σύμφωνα με τη σχέση[13]:

$$V_{OUT} = V_{REF}(1 + R1/R2) \quad (2.3)$$



Σχήμα 2.18 Τυπικό κύκλωμα της ρυθμιζόμενης έκδοσης του LM2596.

Στην έτοιμη πλακέτα DC-DC που χρησιμοποιήθηκε στην κατασκευή, η μία από τις αντιστάσεις του δικτύου ανάδρασης αντικαθίσταται πρακτικά από ρυθμιζόμενο στοιχείο, επιτρέποντας στον χρήστη να μεταβάλλει την τάση εξόδου μέσω του ποτενσιόμετρου. Πριν από τη σύνδεση των υπόλοιπων ηλεκτρονικών στοιχείων, η έξοδος του converter ρυθμίστηκε και ελέγχθηκε με πολύμετρο, ώστε να επιβεβαιωθεί η απαιτούμενη τάση.



Σχήμα 2.19 Ρύθμιση τάσης εξόδου.

Η τάση εξόδου ρυθμίστηκε στα 5 V, ώστε να χρησιμοποιηθεί για την τροφοδοσία των βοηθητικών κυκλωμάτων(Σχήμα 2.19).

2.4.5.4 Πυκνωτές εισόδου και εξόδου

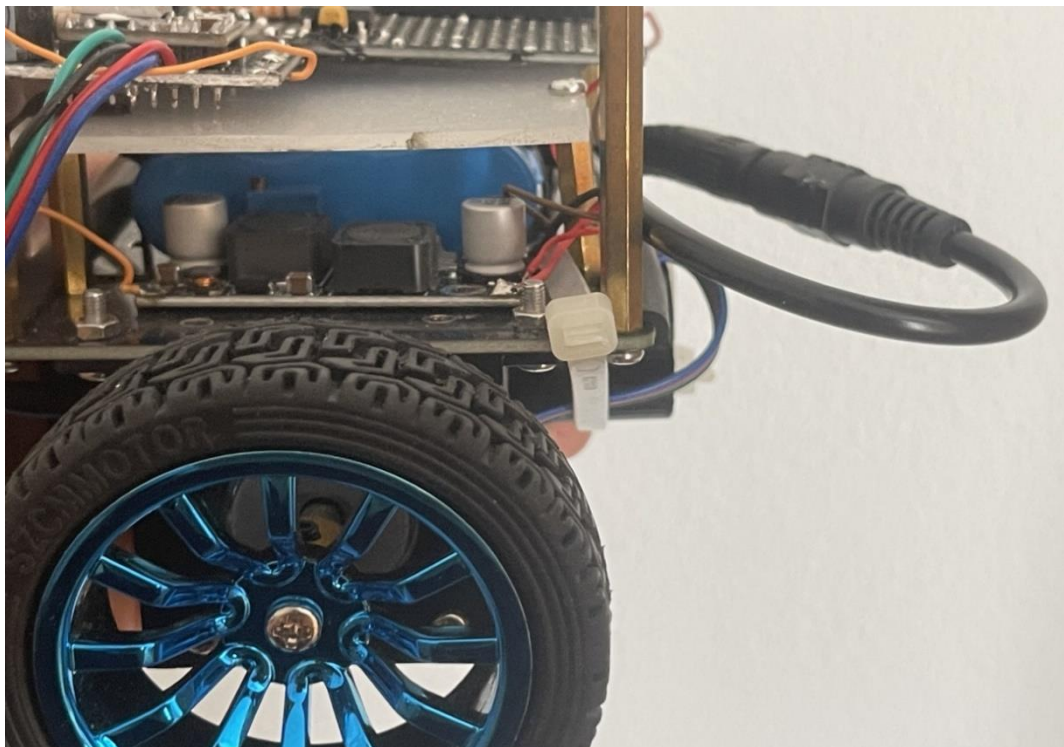
Επειδή ο LM2596 λειτουργεί ως διακοπτικός μετατροπέας, η σωστή επιλογή και τοποθέτηση των πυκνωτών εισόδου και εξόδου είναι απαραίτητη για τη σταθερή λειτουργία του. Στην είσοδο απαιτείται πυκνωτής χαμηλής ισοδύναμης σειριακής αντίστασης (low ESR), ο οποίος τοποθετείται κοντά στον ακροδέκτη VIN και στη γείωση. Ο πυκνωτής αυτός περιορίζει τις απότομες μεταβολές τάσης στην είσοδο και παρέχει το στιγμιαίο ρεύμα που απαιτείται κατά τη μεταγωγή του εσωτερικού διακόπτη[13].

Αντίστοιχα, στην έξοδο χρησιμοποιείται πυκνωτής για το φιλτράρισμα της τάσης και τη σταθερότητα του βρόχου ελέγχου. Σύμφωνα με το datasheet, σημαντικές παράμετροι του πυκνωτή εξόδου είναι η χωρητικότητα, η ονομαστική τάση, το επιτρεπόμενο ripple current και ιδιαίτερα η τιμή ESR.

2.4.5.5 Τοποθέτηση και χρήση του LM2596 στην κατασκευή

Στην τελική κατασκευή το module LM2596 τοποθετήθηκε μαζί με τα υπόλοιπα ηλεκτρονικά στοιχεία και χρησιμοποιήθηκε για τη δημιουργία της χαμηλότερης τάσης που απαιτούσαν τα βοηθητικά κυκλώματα. Η ρύθμιση της εξόδου πραγματοποιήθηκε πριν από την τελική σύνδεση, χρησιμοποιώντας πολύμετρο στους ακροδέκτες εξόδου του module. Με αυτόν τον τρόπο αποφεύχθηκε η εφαρμογή λανθασμένης τάσης στα ηλεκτρονικά στοιχεία.

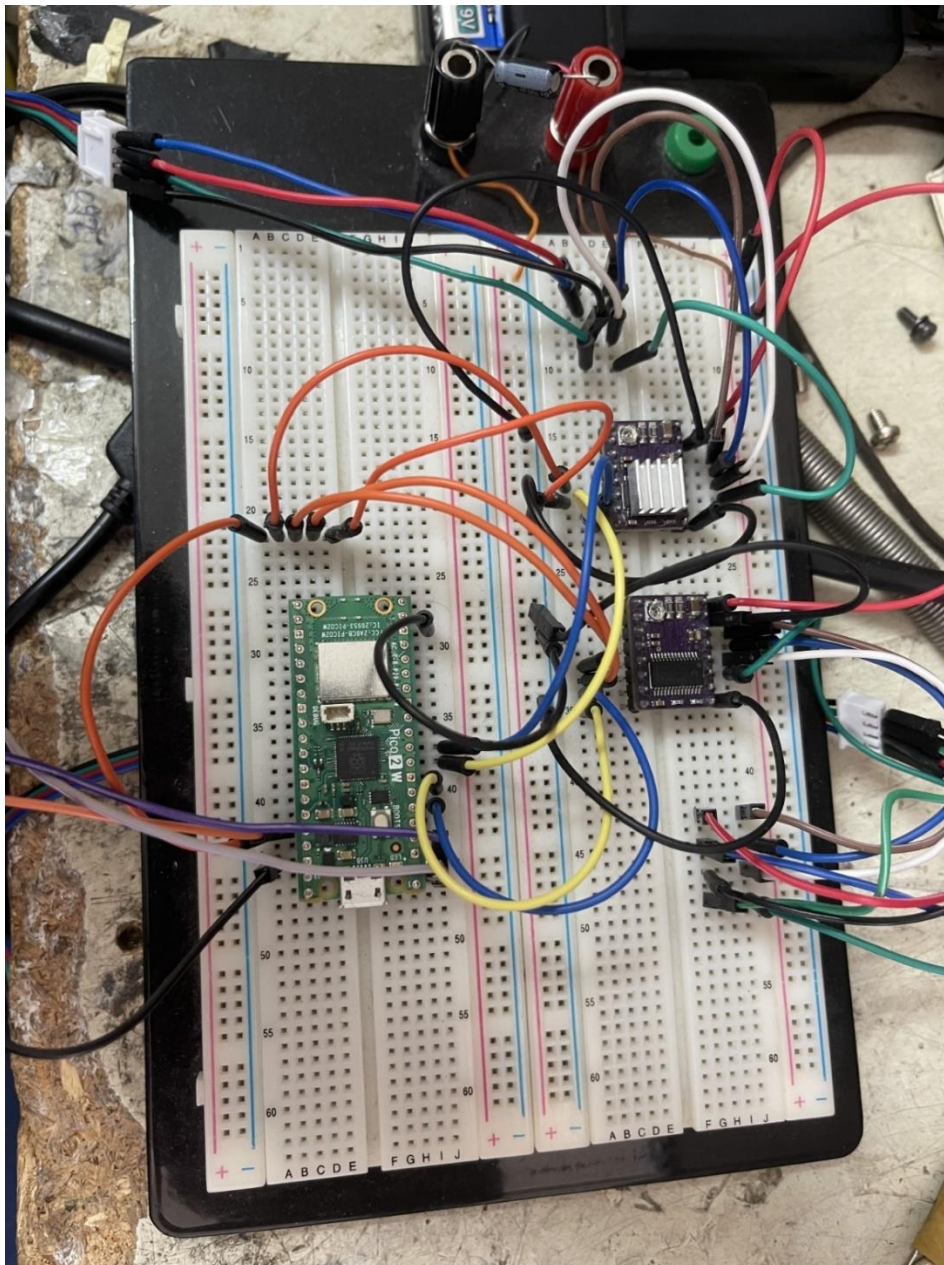
Κατά την τοποθέτηση δόθηκε προσοχή στο μήκος των αγωγών και στη σταθερή στερέωση της πλακέτας(Σχήμα 2.20). Η Texas Instruments επισημαίνει ότι στους διακοπτικούς μετατροπείς οι γρήγορες μεταβολές ρεύματος σε συνδυασμό με την επαγωγή των αγωγών μπορούν να δημιουργήσουν ανεπιθύμητες μεταβατικές τάσεις. Για τον περιορισμό τους συνιστώνται μικρές διαδρομές και περιορισμός των βρόχων ρεύματος[13].



Σχήμα 2.20 Τοποθέτηση της πλακέτας στο ρομπότ.

2.5 Ηλεκτρονική συνδεσμολογία

Αρχικά, για την κατανόηση της λειτουργίας του βηματικού κινητήρα σε συνδιασμό με τα επιμέρους εξαρτήματα πραγματοποιήθηκαν δοκιμές σε ραστερ. Τοποθετήθηκαν οι απαραίτητη βραχυκυκλωτήρες για την σύνδεση μεταξύ μικροελεγκτή, οδηγών μοτέρ και βηματικών κινητήρων, με τροφοδοσία 12V όπως φαίνεται στην παρακάτω εικόνα (Σχήμα 2.21).



Σχήμα 2.21 Κύκλωμα δοκιμών βηματικών κινητήρων.

Έπειτα, μετά την επιτυχημένη δοκιμή κίνησης των μοτέρ, η τελική συνδεσμολογία επάνω στην πλακέτα οργανώθηκε γύρω από τρεις ομάδες σημάτων: το I²C του αισθητήρα, τα STEP/DIR των κινητήρων και τις γραμμές τροφοδοσίας. Η αντιστοίχιση των βασικών ακροδεκτών φαίνεται στον Πίνακα 2.6.

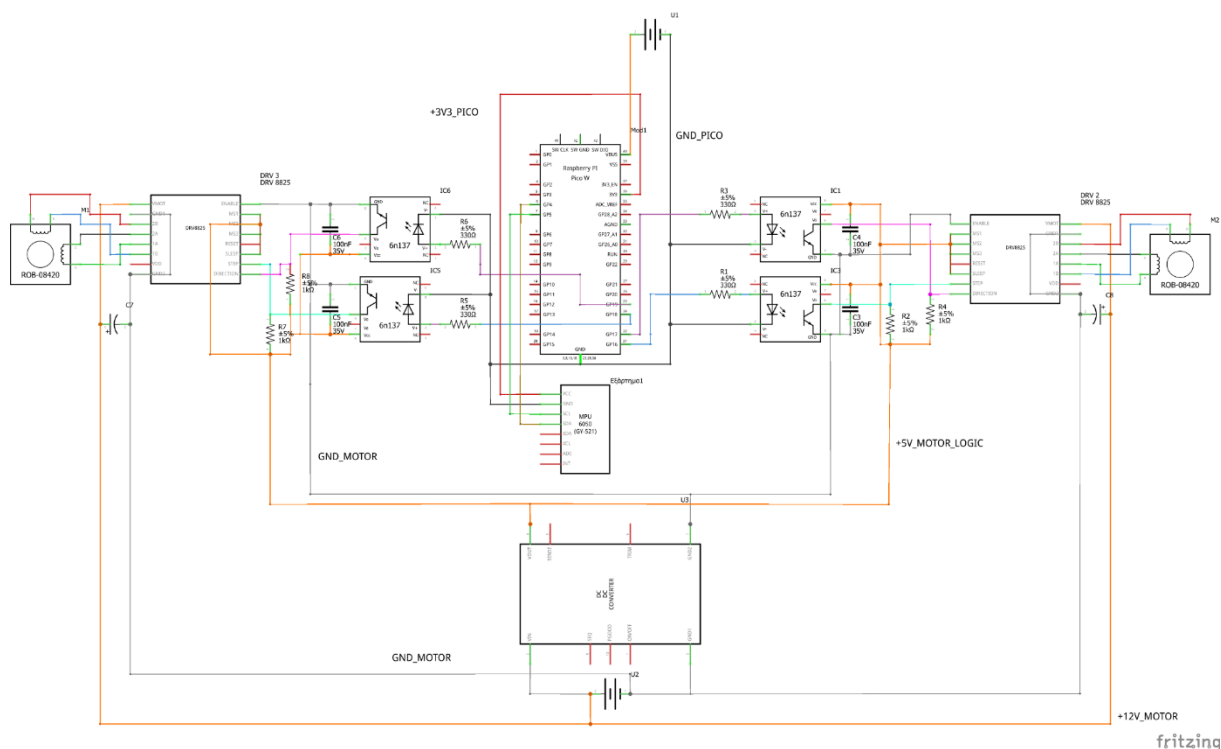
Ο MPU6050 συνδέθηκε στους ακροδέκτες GP4 και GP5. Κατά την εκκίνηση το πρόγραμμα ελέγχει την επικοινωνία πριν προχωρήσει στην κανονική λειτουργία. Η αποτυχία ανάγνωσης του αισθητήρα δεν πρέπει να επιτρέπει την ενεργοποίηση της ισορροπίας.

Οι GP16–GP19 συνδέθηκαν στις εισόδους των οπτοζευκτών και από εκεί στα STEP και DIR των δύο DRV8825. Κάθε κινητήρας συνδέθηκε στα τέσσερα άκρα εξόδου του αντίστοιχου οδηγού, αφού προηγουμένως εντοπίστηκαν τα δύο ζεύγη των πηνίων. Λανθασμένη αντιστοίχιση των πηνίων μπορεί να προκαλέσει θόρυβο, κραδασμούς ή αδυναμία περιστροφής.

Πριν από τη δοκιμή ισορροπίας ελέγχθηκαν χωριστά η περιστροφή και η αλλαγή φοράς. Η διαδικασία αυτή επέτρεψε να διαχωριστούν προβλήματα καλωδίωσης από προβλήματα του PID.

Πίνακας 2.6 Αντιστοίχιση ακροδεκτών μικροελεγκτή.

Λειτουργία	Ακροδέκτης Raspberry Pi Pico 2 W	Περιγραφή
STEP αριστερού κινητήρα	GP16	Παραγωγή παλμών για τον αριστερό οδηγό
DIR αριστερού κινητήρα	GP17	Καθορισμός φοράς περιστροφής αριστερού κινητήρα
STEP δεξιού κινητήρα	GP18	Παραγωγή παλμών για τον δεξιό οδηγό
DIR δεξιού κινητήρα	GP19	Καθορισμός φοράς περιστροφής δεξιού κινητήρα
SDA αισθητήρα MPU6050	GP4	Γραμμή δεδομένων I ² C
SCL αισθητήρα MPU6050	GP5	Γραμμή ρολογιού I ² C



Σχήμα 2.22 Το ηλεκτρονικό σχέδιο του κυκλώματος.

2.6 Τροφοδοσία και ηλεκτρική προστασία

Η ξεχωριστή τροφοδοσία του κυκλώματος λογικής περιόρισε τον κίνδυνο επανεκκίνησης του μικροελεγκτή όταν οι κινητήρες άλλαζαν απότομα ταχύτητα ή φορά. Το powerbank παρείχε ανεξάρτητη τροφοδοσία στον Pico 2 W, ενώ οι κινητήρες τροφοδοτήθηκαν από την κύρια πηγή ισχύος.

Στην πλευρά των οδηγών το ρεύμα ρυθμίστηκε μέσω της τάσης αναφοράς. Αν η ρύθμιση ήταν χαμηλή, οι κινητήρες δεν είχαν αρκετή ροπή και η αντίδραση στην πτώση ήταν αδύναμη. Αν ήταν υπερβολικά υψηλή, αυξανόταν η θερμοκρασία. Η ρύθμιση έγινε σταδιακά, με παράλληλο έλεγχο της συμπεριφοράς και της θέρμανσης.

Ο πυκνωτής κοντά στην είσοδο τροφοδοσίας των DRV8825 χρησιμοποιήθηκε για την απορρόφηση αιχμών. Ο step-down converter παρείχε την απαιτούμενη χαμηλότερη τάση στα βοηθητικά κυκλώματα. Οι οπτοζεύκτες συμπλήρωσαν τον διαχωρισμό, καθώς τα σήματα ελέγχου μεταφέρονταν χωρίς άμεση ηλεκτρική σύνδεση μεταξύ των δύο πλευρών.

Η μηχανική στερέωση των μπαταριών αποτελούσε και ηλεκτρικό μέτρο ασφάλειας. Μια μετακινούμενη πηγή τροφοδοσίας μπορεί να τραβήξει ένα καλώδιο, να δημιουργήσει χαλαρή επαφή ή να έρθει σε επαφή με άλλο στοιχείο. Για τον λόγο αυτό οι μπαταρίες και το powerbank συγκρατήθηκαν με δεματικά, ενώ τα καλώδια οργανώθηκαν μακριά από τους τροχούς.

2.7 Υλικά και κόστος

Τα βασικά υλικά της κατασκευής παρουσιάζονται στον Πίνακα 2.7.

Πίνακας 2.7 Υλικά κατασκευής και κόστος.

A/A	Υλικό / Εξάρτημα	Ποσότητα	Σύνολο (€)
1	Raspberry Pi Pico 2 W	1	8.9
2	MPU6050	1	3.6
3	Nema 17 Βηματικός κινητήρας	2	40
4	Βάση Βηματικού κινητήρα	2	4
5	DRV8825	2	6.4
6	Οπτοζεύκτης 6N137	4	7.6
7	Ρόδα και λάστιχο	2	15
8	Step-down converter	1	4.5
9	Powerbank 10000mAh	1	23
10	3S Επαναφορτιζόμενη μπαταρία λιθίου 11.1V 2200mAh	1	30
11	Κεραμικός πυκνωτής 100nF 35V	4	0.4
12	Ηλεκτρολυτικός Πυκνωτής 100μF 63V	2	0.2
13	Αντίσταση 330Ω 5%	4	0.8
14	Αντίσταση 1kΩ 5%	4	0.8

15	Πλαίσιο / ακρυλικές πλάκες / βάσεις	1	8
16	Καλώδιο stepper motor, καλώδιο χαλκού 0.4mm, connectors και pins	1 σύνολο	5
17	PCB πλακέτα διπλής όψης	2	1
18	Βίδες, αποστάτες και παξιμάδια	1 σύνολο	2
19	Δεματικά καλωδίων και υλικά στερέωσης	4	0.2
	Σύνολο		161.4

2.8 Τελική διάταξη και συναρμολόγηση

Η τελική διάταξη προέκυψε μετά από διαδοχικές αλλαγές. Οι κινητήρες και οι τροχοί παρέμειναν συμμετρικά στο κάτω μέρος. Το powerbank τοποθετήθηκε χαμηλά και κεντρικά, ανάμεσα στους δύο κινητήρες, ενώ οι πλακέτες ελέγχου στερεώθηκαν στο επάνω επίπεδο. Οι δύο DRV8825 τοποθετήθηκαν στις πλευρές της πλακέτας, ώστε οι συνδέσεις με τους κινητήρες να παραμένουν σύντομες και προσβάσιμες.

Ο MPU6050 στερεώθηκε κατακόρυφα. Η θέση του παρέμεινε σταθερή μετά τη βαθμονόμηση, επειδή οποιαδήποτε αλλαγή θα απαιτούσε νέο υπολογισμό offsets και επανέλεγχο του άξονα. Οι μπαταρίες και το powerbank συγκρατήθηκαν με δεματικά. Η πλακέτα και ορισμένα καλώδια σταθεροποιήθηκαν με διαφανή ταινία ή κόλλα(Σχήμα 2.23).

Κατά τη συναρμολόγηση (Σχήμα 2.24) ελέγχθηκε πρώτα η συνέχεια των συνδέσεων και έπειτα η λειτουργία κάθε υποσυστήματος. Το πρόγραμμα ισορροπίας δεν ενεργοποιήθηκε πριν επιβεβαιωθούν η ανάγνωση του αισθητήρα, η περιστροφή των δύο κινητήρων και η αλλαγή φοράς. Η σειρά αυτή περιόρισε τον κίνδυνο απότομης κίνησης κατά τις πρώτες δοκιμές.

Κεφάλαιο 3ο: Ανάπτυξη Λογισμικού, Έλεγχος και Δοκιμές Λειτουργίας

3.1 Περιβάλλον Ανάπτυξης και βιβλιοθήκες

Το πρόγραμμα αναπτύχθηκε στο Arduino IDE. Η επιλογή του περιβάλλοντος διευκόλυνε τη μεταγλώττιση, τη μεταφόρτωση και την παρακολούθηση των δεδομένων από το Serial Monitor. Για τη σωστή αναγνώριση του Raspberry Pi Pico 2 W εγκαταστάθηκε το αντίστοιχο πακέτο πλακετών και επιλέχθηκε η σωστή θύρα USB.

Η ανάπτυξη του BLE απαιτήσε έκδοση του περιβάλλοντος και πακέτο πλακέτας που υποστήριζαν τη βιβλιοθήκη που χρησιμοποιήθηκε. Η λειτουργία ελέγχθηκε αρχικά ανεξάρτητα από τον PID, ώστε τα προβλήματα σύνδεσης να μην συγχέονται με προβλήματα ισορροπίας.

Οι βασικές βιβλιοθήκες του προγράμματος ήταν οι ακόλουθες (Σχήμα 3.1)

```
37 #include <Arduino.h>
38 #include <Wire.h>
39 #include <BLE.h>
40 #include "pico/time.h"
41 #include "hardware/gpio.h"
```

Σχήμα 3.1 Βιβλιοθήκες.

Η Wire χρησιμοποιήθηκε για την επικοινωνία I²C με τον MPU6050. Η BLE παρείχε τις υπηρεσίες και τα χαρακτηριστικά της ασύρματης επικοινωνίας. Οι βιβλιοθήκες του Pico χρησιμοποιήθηκαν για ακριβέστερο χρονισμό και άμεσο χειρισμό των ακροδεκτών που παράγουν τους παλμούς STEP.

3.2 Δομή του προγράμματος και καταστάσεις λειτουργίας

Το πρόγραμμα χωρίστηκε σε τέσσερα λειτουργικά μέρη. Το πρώτο αρχικοποιεί τις εισόδους και τις εξόδους, το I²C και το BLE. Το δεύτερο διαβάζει τον αισθητήρα και υπολογίζει τη γωνία. Το τρίτο υπολογίζει την έξοδο του PID και ενημερώνει τις επιθυμητές ταχύτητες των κινητήρων. Το τέταρτο διαχειρίζεται τις εντολές του χρήστη και τους μηχανισμούς ασφαλείας

Η αντιστοίχιση των ακροδεκτών στον κώδικα έγινε ως εξής (Σχήμα 3.2):

```
43 #define STEP_PIN_L 16
44 #define DIR_PIN_L 17
45 #define STEP_PIN_R 18
46 #define DIR_PIN_R 19
47
48 #define I2C_SDA_PIN 4
49 #define I2C_SCL_PIN 5
```

Σχήμα 3.2 Ορισμός ακροδεκτών.

Οι βασικές καταστάσεις αποθηκεύονται σε μεταβλητές που δείχνουν αν ο έλεγχος είναι ενεργός, αν έχει ανιχνευθεί πτώση και αν εκτελείται δοκιμή κινητήρων (Σχήμα 3.3).

```

95  bool armed = false;
96  bool motorTestMode = false;
97  bool fallen = false;
98
99  float angleDeg = 0.0f;
100 float gyroRateDps = 0.0f;
101 float balanceOffsetDeg = 0.0f;
102 float forwardCmd = 0.0f;
103 float turnCmd = 0.0f;
104 float pidIntegral = 0.0f;

```

Σχήμα 3.3 Βασικές καταστάσεις.

Η μεταβλητή `armed` αποτρέπει την αυτόματη εκκίνηση των κινητήρων. Η `fallen` δηλώνει ότι ο μηχανισμός πτώσης έχει ενεργοποιηθεί. Οι `forwardCmd` και `turnCmd` αποθηκεύουν τις τελευταίες εντολές κίνησης. Όταν δεν λαμβάνεται νέα εντολή για συγκεκριμένο χρονικό διάστημα, μηδενίζονται ώστε το ρομπότ να μην συνεχίζει την πορεία του με τα παλαιά δεδομένα.

3.3 Ανάγνωση, βαθμονόμηση και εκτίμηση γωνίας

Ο MPU6050 παρέχει ακατέργαστες τιμές για την επιτάχυνση και τη γωνιακή ταχύτητα στους τρεις άξονες. Οι τιμές αυτές δεν χρησιμοποιήθηκαν απευθείας. Αρχικά εφαρμόστηκαν offsets και στη συνέχεια υπολογίστηκε η γωνία που αντιστοιχεί στην κίνηση εμπρός-πίσω.

Το επιταχυνσιόμετρο δίνει μια αναφορά ως προς τη βαρύτητα, αλλά επηρεάζεται έντονα όταν το πλαίσιο δονείται ή οι τροχοί επιταχύνουν. Το γυροσκόπιο αντιδρά γρήγορα στις μεταβολές, όμως η ολοκλήρωση της γωνιακής ταχύτητας συσσωρεύει σφάλμα. Οι δύο εκτιμήσεις συνδυάστηκαν με συμπληρωματικό φίλτρο:

$$\text{angle} = \alpha * (\text{angle} + \text{gyroRate} * \text{dt}) + (1 - \alpha) * \text{accelAngle}; \quad (3.1)$$

Η παράμετρος α καθορίζει τη σχετική βαρύτητα των δύο αισθητήρων. Μεγαλύτερη τιμή δίνει μεγαλύτερη έμφαση στο γυροσκόπιο, ενώ μικρότερη τιμή επιτρέπει στο επιταχυνσιόμετρο να διορθώνει γρηγορότερα την απόκλιση. Το χρονικό βήμα dt υπολογίζεται σε κάθε επανάληψη και δεν θεωρείται σταθερό χωρίς έλεγχο.

Η επιθυμητή γωνία δεν είναι πάντοτε ίση με μηδέν. Η μεταβλητή `balanceOffsetDeg` ορίζει τη θέση ισορροπίας που προκύπτει από τη διαδικασία ZERO. Κατά την κίνηση, η γωνία αναφοράς μετατοπίζεται ανάλογα με την εντολή του χρήστη. Επειδή κατά τις δοκιμές παρατηρήθηκε διαφορετική συμπεριφορά μεταξύ εμπρόσθιας και οπίσθιας κίνησης, χρησιμοποιήθηκαν διαφορετικές μέγιστες τιμές κλίσης για τις δύο κατευθύνσεις. Συγκεκριμένα, στην τελική έκδοση του κώδικα ορίστηκαν `forwardLeanDeg=1,5°` και `backwardLeanDeg=0,9°`. Το πρόσημο της μεταβολής επιλέχθηκε σύμφωνα με τον προσανατολισμό του αισθητήρα και τη φυσική φορά κίνησης της συγκεκριμένης κατασκευής:

```
float cmd = filteredForwardCmd;
```

```
float activeLeanDeg = (cmd >= 0.0f) ? forwardLeanDeg : backwardLeanDeg;
```

```
float targetAngle = balanceOffsetDeg - cmd * activeLeanDeg;
```

```
float error = angleDeg - targetAngle;
```

Το πρόσημο και ο άξονας ελέγχθηκαν πρακτικά. Το ρομπότ κρατήθηκε με το χέρι και γέρθηκε προς τις δύο κατευθύνσεις. Η μεταβολή της angleDeg έπρεπε να συμφωνεί με τη φυσική κίνηση. Διαφορετικά άλλαζαν οι παράμετροι angleMode, angleSign ή gyroSign.

3.3.1 Πρόγραμμα βαθμονόμησης του αισθητήρα MPU6050

Πριν από την ενσωμάτωση του MPU6050 στον τελικό κώδικα του ρομπότ, χρησιμοποιήθηκε ξεχωριστό πρόγραμμα βαθμονόμησης. Σκοπός του ήταν να εντοπιστούν οι σταθερές αποκλίσεις που εμφανίζουν το επιταχυνσιόμετρο και το γυροσκόπιο, ακόμη και όταν ο αισθητήρας παρέμενε ακίνητος. Οι αποκλίσεις αυτές διαφέρουν από αισθητήρα σε αισθητήρα και επηρεάζονται επίσης από τον τρόπο τοποθέτησής του πάνω στην κατασκευή.

Κατά τη βαθμονόμηση, ο MPU6050 παρέμεινε σταθερός στην τελική κατακόρυφη θέση στην οποία χρησιμοποιήθηκε στο ρομπότ. Το πρόγραμμα αρχικοποίησε την επικοινωνία μέσω I²C και διάβασε επαναλαμβανόμενα τις ακατέργαστες τιμές των τριών αξόνων του επιταχυνσιόμετρου και του γυροσκοπίου. Για κάθε άξονα υπολογίστηκε ο μέσος όρος ενός μεγάλου αριθμού δειγμάτων, ώστε να περιοριστεί η επίδραση του στιγμιαίου θορύβου.

Για το γυροσκόπιο, οι αναμενόμενες τιμές κατά την ακινησία ήταν κοντά στο μηδέν, αφού δεν υπήρχε περιστροφή της κατασκευής. Επομένως, οι μέσες τιμές που καταγράφηκαν θεωρήθηκαν ως το σταθερό σφάλμα κάθε άξονα. Στην περίπτωση του επιταχυνσιόμετρου, οι αναμενόμενες τιμές εξαρτήθηκαν από τον προσανατολισμό του αισθητήρα, καθώς ένας από τους άξονες μετρούσε κυρίως την επιτάχυνση της βαρύτητας. Για τον λόγο αυτό, η βαθμονόμηση πραγματοποιήθηκε με τον αισθητήρα ήδη στερεωμένο στην τελική του θέση και όχι ανεξάρτητα από την κατασκευή.

Η βασική διαδικασία που ακολούθησε το πρόγραμμα ήταν η εξής:

- Αρχικοποίηση του Raspberry Pi Pico 2 W και της επικοινωνίας I²C.
- Ενεργοποίηση και έλεγχος επικοινωνίας με τον MPU6050.
- Διατήρηση του αισθητήρα ακίνητου στην τελική θέση τοποθέτησης.
- Συλλογή πολλών ακατέργαστων μετρήσεων από τους έξι άξονες.
- Υπολογισμός του μέσου όρου για κάθε άξονα.
- Σύγκριση των μέσων τιμών με τις αναμενόμενες τιμές ακινησίας.
- Καταγραφή των διορθωτικών τιμών και εισαγωγή τους στον τελικό κώδικα.

Από τη διαδικασία αυτή προέκυψαν οι διορθωτικές τιμές που χρησιμοποιήθηκαν στην τελική υλοποίηση(Σχήμα 3.4)

```
52 #define ACCEL_X_OFFSET -17141
53 #define ACCEL_Y_OFFSET 418
54 #define ACCEL_Z_OFFSET 22301
55 #define GYRO_X_OFFSET 222
56 #define GYRO_Y_OFFSET 111
57 #define GYRO_Z_OFFSET 123
```

Σχήμα 3.4 Ορισμός offsets.

Οι τιμές εφαρμόζονται στις ακατέργαστες μετρήσεις πριν από τον υπολογισμό της γωνίας κλίσης. Με αυτόν τον τρόπο περιορίζεται το σταθερό σφάλμα του αισθητήρα και η πληροφορία που χρησιμοποιεί ο PID ελεγκτής γίνεται πιο αξιόπιστη. Οι συγκεκριμένες τιμές δεν είναι γενικές για όλους τους

αισθητήρες MPU6050. Σε περίπτωση αλλαγής του αισθητήρα, μετακίνησης της πλακέτας ή διαφορετικού προσανατολισμού, η διαδικασία βαθμονόμησης πρέπει να εκτελεστεί ξανά.

3.3.2 Ρύθμιση των καταχωρητών του MPU6050

Η επικοινωνία του MPU6050 με τον Raspberry Pi Pico 2 W πραγματοποιήθηκε μέσω I²C. Ο MPU6050 υποστηρίζει επικοινωνία I²C με συχνότητα έως 400 kHz. Μέσω του διαύλου, ο μικροελεγκτής μπορεί να μεταβάλλει τους καταχωρητές ρύθμισης και να διαβάσει τις πιο πρόσφατες μετρήσεις των έξι αξόνων [16].

Η επιλογή του εύρους του γυροσκοπίου γίνεται μέσω του καταχωρητή GYRO_CONFIG, στη διεύθυνση 27. Τα bits FS_SEL[1:0] καθορίζουν το επιλεγμένο εύρος, ενώ τα bits XG_ST, YG_ST και ZG_ST ενεργοποιούν τη λειτουργία αυτοελέγχου για κάθε άξονα. Αντίστοιχα, ο καταχωρητής ACCEL_CONFIG, στη διεύθυνση 28, περιέχει τα bits AFS_SEL, με τα οποία επιλέγεται το εύρος του επιταχυνσιόμετρου [16].

Η έξοδος του επιταχυνσιόμετρου αποθηκεύεται σε έξι διαδοχικούς καταχωρητές, από τη διεύθυνση 59 έως τη διεύθυνση 64. Για κάθε άξονα χρησιμοποιούνται δύο bytes, ένα υψηλής και ένα χαμηλής τάξης:

ACCEL_XOUT_H και ACCEL_XOUT_L,
ACCEL_YOUT_H και ACCEL_YOUT_L,
ACCEL_ZOUT_H και ACCEL_ZOUT_L.

Οι μετρήσεις του γυροσκοπίου αποθηκεύονται αντίστοιχα στους καταχωρητές 67 έως 72:

GYRO_XOUT_H και GYRO_XOUT_L,
GYRO_YOUT_H και GYRO_YOUT_L,
GYRO_ZOUT_H και GYRO_ZOUT_L.

Τα δύο bytes κάθε άξονα συνδυάζονται σε μία προσημασμένη τιμή 16 bit πριν από την εφαρμογή των offsets και των συντελεστών κλίμακας:

```
int16_t makeInt16(uint8_t hi, uint8_t lo) {
    return (int16_t)((hi << 8) | lo);
}
```

Στον τελικό κώδικα τα δύο bytes κάθε μέτρησης συνδυάζονται μέσω της συνάρτησης makeInt16(), ώστε να προκύπτει προσημασμένη τιμή 16 bit. Στη συνέχεια εφαρμόζονται τα offsets που προέκυψαν από τη διαδικασία βαθμονόμησης και, για τις μετρήσεις του γυροσκοπίου, ο αντίστοιχος συντελεστής κλίμακας. Με τον τρόπο αυτό οι ακατέργαστες τιμές του MPU6050 μετατρέπονται στη μορφή που χρησιμοποιείται για τον υπολογισμό της γωνίας και της γωνιακής ταχύτητας.

3.3.3 Ρυθμός δειγματοληψίας και ψηφιακό φίλτρο

Ο καταχωρητής SMPLRT_DIV, στη διεύθυνση 25, καθορίζει τον διαιρέτη του βασικού ρυθμού εξόδου του γυροσκοπίου. Η συχνότητα δειγματοληψίας υπολογίζεται από τη σχέση:

$$f_s = \frac{f_{gyro}}{1 + \text{SMPLRT_DIV}} \quad (3.2)$$

Στην τελική έκδοση του προγράμματος δεν πραγματοποιήθηκε ρητή εγγραφή στον καταχωρητή SMPLRT_DIV. Για τον λόγο αυτό δεν ορίστηκε μέσω του λογισμικού συγκεκριμένος διαιρέτης δειγματοληψίας και δεν χρησιμοποιείται αυθαίρετη αριθμητική τιμή για την παράμετρο αυτή στην παρούσα εργασία. Αντίθετα, ο καταχωρητής CONFIG ρυθμίστηκε με την τιμή 0x03, επομένως τα bits DLPF_CFG έλαβαν την τιμή 3. Παράλληλα, οι καταχωρητές GYRO_CONFIG και ACCEL_CONFIG γράφτηκαν με την τιμή 0x00.

Η εσωτερική συχνότητα ενημέρωσης του αισθητήρα δεν πρέπει να συγχέεται με τη συχνότητα εκτέλεσης του βρόχου ελέγχου του ρομπότ. Στην τελική υλοποίηση ο κύριος βρόχος ελέγχου ορίστηκε στα 250 Hz, δηλαδή με ονομαστική περίοδο 4 ms. Σε κάθε κύκλο πραγματοποιείται ενημέρωση της εκτιμώμενης γωνίας και εκτέλεση του αλγορίθμου ελέγχου.

Όταν το εσωτερικό Digital Low-Pass Filter είναι ενεργοποιημένο, ο βασικός ρυθμός εξόδου του γυροσκοπίου είναι 1 kHz. Όταν είναι απενεργοποιημένο, μπορεί να φτάσει τα 8 kHz. Ο ίδιος ρυθμός χρησιμοποιείται για την ενημέρωση των καταχωρητών μέτρησης και των υπόλοιπων εσωτερικών λειτουργιών του αισθητήρα [16].

Ο καταχωρητής CONFIG, στη διεύθυνση 26, περιέχει τα bits DLPF_CFG, μέσω των οποίων ρυθμίζεται το ψηφιακό χαμηλοπερατό φίλτρο του επιταχυνσιόμετρου και του γυροσκοπίου. Η ρύθμιση αυτή δεν πρέπει να συγχέεται με το συμπληρωματικό φίλτρο που εφαρμόζεται αργότερα στον κώδικα. Το DLPF λειτουργεί μέσα στον αισθητήρα και περιορίζει το εύρος ζώνης των ακατέργαστων μετρήσεων, ενώ το συμπληρωματικό φίλτρο συνδυάζει την εκτίμηση του επιταχυνσιόμετρου με την ολοκλήρωση της γωνιακής ταχύτητας.

Στην τελική υλοποίηση χρησιμοποιήθηκε συντελεστής FILTER_ALPHA=0,98. Επομένως, σε κάθε επανάληψη το μεγαλύτερο βάρος δίνεται στην προηγούμενη εκτίμηση της γωνίας και στη μεταβολή που προκύπτει από το γυροσκόπιο, ενώ μικρότερο ποσοστό της εκτίμησης προέρχεται από το επιταχυνσιόμετρο:

```
#define CONTROL_HZ 250
```

```
#define CONTROL_PERIOD_US (1000000UL / CONTROL_HZ)
```

```
#define FILTER_ALPHA 0.98f
```

3.3.4 Αρχικοποίηση και διαχείριση ισχύος

Ο καταχωρητής PWR_MGMT_1, στη διεύθυνση 107, χρησιμοποιείται για την επαναφορά του αισθητήρα, την ενεργοποίηση ή απενεργοποίηση της κατάστασης sleep και την επιλογή της πηγής χρονισμού. Μετά την τροφοδοσία, ο MPU6050 πρέπει να εξέλθει από την κατάσταση ύπνου πριν ξεκινήσει η κανονική ανάγνωση δεδομένων. Ο ίδιος καταχωρητής επιτρέπει επίσης την επιλογή του εσωτερικού ταλαντωτή ή μιας πηγής χρονισμού που βασίζεται σε έναν από τους άξονες του γυροσκοπίου [16].

Το datasheet προτείνει τη χρήση αναφοράς χρονισμού από έναν άξονα του γυροσκοπίου ή από εξωτερική πηγή για βελτιωμένη σταθερότητα. Στην τελική έκδοση του προγράμματος γράφτηκε η τιμή 0x00 στον καταχωρητή PWR_MGMT_1. Με την εγγραφή αυτή ο MPU6050 εξέρχεται από την κατάσταση sleep και το πεδίο επιλογής πηγής χρονισμού παραμένει στην τιμή 0. Η αρχικοποίηση πραγματοποιείται πριν από την έναρξη της περιοδικής ανάγνωσης των μετρήσεων.

```
mpuWrite(0x6B, 0x00);
```

3.3.5 Βηματικοί κινητήρες και οδηγοί DRV8825

Κάθε τροχός κινείται από έναν βηματικό κινητήρα και έναν ανεξάρτητο οδηγό DRV8825. Η ταχύτητα καθορίζεται από τη συχνότητα των παλμών STEP και η φορά από το DIR. Η ανεξάρτητη οδήγηση των δύο πλευρών επιτρέπει τόσο τη διόρθωση της ισορροπίας όσο και τη στροφή.

Οι οδηγοί ρυθμίστηκαν σε λειτουργία μικροβηματισμού 1/16, σύμφωνα με τη συνδεσμολογία της κατασκευής, ώστε η περιστροφή να είναι πιο ομαλή. Η τάση αναφοράς ρυθμίστηκε σταδιακά. Η αρχική προτεραιότητα ήταν να εξασφαλιστεί επαρκής ροπή χωρίς έντονη θέρμανση των οδηγών ή των κινητήρων

3.4 Παραγωγή παλμών και έλεγχος κινητήρων

Η έξοδος του PID μετατρέπεται σε ταχύτητα βηματικών κινητήρων, εκφρασμένη σε βήματα ανά δευτερόλεπτο. Η συχνότητα των παλμών STEP αυξάνεται όταν απαιτείται μεγαλύτερη ταχύτητα. Το DIR ενημερώνεται με βάση το πρόσημο της επιθυμητής ταχύτητας.

Η παραγωγή παλμών δεν βασίστηκε σε μεγάλες καθυστερήσεις τύπου delay(), επειδή αυτές θα σταματούσαν προσωρινά την ανάγνωση του αισθητήρα και την επικοινωνία. Αντίθετα, ο χρόνος του επόμενου παλμού υπολογίζεται χωριστά για κάθε κινητήρα. Με αυτόν τον τρόπο ο αριστερός και ο δεξιός τροχός μπορούν να έχουν διαφορετική ταχύτητα χωρίς να διακόπτεται ο βρόχος ελέγχου.

Οι κινητήρες είναι τοποθετημένοι σε αντίθετες πλευρές, επομένως η ίδια ηλεκτρική φορά δεν αντιστοιχεί απαραίτητα στην ίδια φυσική κατεύθυνση. Στον κώδικα χρησιμοποιήθηκαν οι εξής μεταβλητές(Σχήμα 3.5).

```
79 bool dirInvertL = false;  
80 bool dirInvertR = true;
```

Σχήμα 3.5 Τιμές κατεύθυνσης του κάθε κινητήρα.

Η βασική δοκιμή έγινε με το ρομπότ σηκωμένο από το έδαφος με τις εξής εντολές:

M 800;

M -800;

STOP;

Με την πρώτη εντολή ελέγχθηκε η περιστροφή προς μία κατεύθυνση και με τη δεύτερη η αντιστροφή. Αν ένας κινητήρας περιστρεφόταν αλλά δεν άλλαζε φορά, ο έλεγχος επικεντρωνόταν στη γραμμή DIR, στον οπτοζεύκτη, στην pull-up σύνδεση και στην είσοδο του οδηγού. Αν δεν περιστρεφόταν καθόλου, ελέγχονταν η τροφοδοσία, το STEP και η σύνδεση των πηνίων.

3.5 PID ελεγκτής και διαδικασία ρύθμισης

Ο PID ελεγκτής υπολογίζει την αντίδραση από τη διαφορά μεταξύ πραγματικής και επιθυμητής γωνίας. Η μορφή που χρησιμοποιήθηκε στο πρόγραμμα ήταν:

```
pidIntegral += error * dt;
```

```
pidIntegral = constrain(pidIntegral, -integralLimit, integralLimit);
```

$\text{float pid} = (\text{Kp} * \text{error}) + (\text{Ki} * \text{pidIntegral}) + (\text{Kd} * \text{gyroRateDps});$

$\text{float baseSps} = \text{motorSign} * \text{motorGain} * \text{pid};$

Ο αναλογικός όρος Kp αυξάνει την αντίδραση όσο μεγαλώνει η απόκλιση. Χαμηλό Kp οδηγεί σε αργή διόρθωση, ενώ υπερβολικά υψηλό Kp προκαλεί απότομη κίνηση και ταλαντώσεις. Ο ολοκληρωτικός όρος Ki μπορεί να διορθώσει μικρή μόνιμη απόκλιση, αλλά χρειάζεται περιορισμό για να μη συσσωρεύεται μεγάλη τιμή μετά από πτώση. Ο διαφορικός όρος Kd χρησιμοποιεί τη γωνιακή ταχύτητα και περιορίζει την ένταση της αντίδρασης όταν η κλίση αλλάζει γρήγορα.

Η motorGain χρησιμοποιήθηκε ως ξεχωριστή ενίσχυση μεταξύ της εξόδου του PID και της ταχύτητας των κινητήρων. Έτσι ήταν δυνατό να αλλάξει η συνολική δύναμη της αντίδρασης χωρίς ταυτόχρονη αλλαγή των Kp, Ki και Kd. Η motorSign αντιστρέφει το συνολικό πρόσημο, όταν οι τροχοί κινούνται αντίθετα από την κατεύθυνση που απαιτεί η ισορροπία.

Η ρύθμιση έγινε πειραματικά. Πρώτα επιβεβαιώθηκαν η φορά των κινητήρων και το πρόσημο της γωνίας. Έπειτα αυξήθηκε σταδιακά το Kp μέχρι το ρομπότ να αντιδρά στην πτώση. Το Kd ρυθμίστηκε για τη μείωση των ταλαντώσεων, ενώ το Ki διατηρήθηκε μηδενικό ή χαμηλό στις αρχικές δοκιμές. Η διαδικασία επαναλαμβανόταν μετά από κάθε αλλαγή στη θέση των βαριών εξαρτημάτων ή στη ρύθμιση του ρεύματος.

Μετά από διαδοχικές δοκιμές, οι τελικές βασικές τιμές που χρησιμοποιήθηκαν στον κώδικα ήταν

Kp= 35, Ki=0, Kd=0.08 και motorGain=40. Η μηδενική τιμή του ολοκληρωτικού όρου επιλέχθηκε επειδή στις συγκεκριμένες δοκιμές δεν κρίθηκε απαραίτητη η συσσώρευση διορθωτικής δράσης, ενώ μεγαλύτερες τιμές του διαφορικού όρου προκαλούσαν εντονότερο τρέμουλο και νευρική συμπεριφορά της κατασκευής. Σε ορισμένες επιμέρους δοκιμές χρησιμοποιήθηκαν ελαφρώς διαφορετικές τιμές, όπως Kd=0.1 προκειμένου να εξεταστεί η συμπεριφορά κατά την κίνηση προς τα πίσω. Οι τιμές αυτές δεν θεωρούνται γενικές για κάθε μηχανική διάταξη, αλλά αφορούν την τελική μορφή του συγκεκριμένου πρωτοτύπου.

Κατά τη διαδικασία ρύθμισης παρατηρήθηκε ότι η απόδοση του ελεγκτή δεν εξαρτιόταν αποκλειστικά από τις τιμές Kp, Ki και Kd. Η θέση των μπαταριών και του powerbank, η κατανομή της μάζας, η πρόσφυση των τροχών και η ταχύτητα με την οποία οι βηματικοί κινητήρες προσπαθούσαν να διορθώσουν μία απόκλιση επηρέαζαν άμεσα τη συμπεριφορά. Μικρή μετακίνηση του powerbank ήταν αρκετή για να αλλάξει την ισορροπία μεταξύ της εμπρόσθιας και της οπίσθιας κατεύθυνσης. Η τελική θέση των επιμέρους μαζών επιλέχθηκε επομένως πειραματικά, μετά από επαναλαμβανόμενες δοκιμές.

Επιπλέον, στον τελικό κώδικα περιορίστηκε η μέγιστη συνιστώσα ταχύτητας που μπορεί να ζητήσει ο ελεγκτής αποκλειστικά για τη διόρθωση της ισορροπίας. Η μεταβλητή balanceMaxSps ορίστηκε σε 4800 βήματα ανά δευτερόλεπτο. Η αλλαγή αυτή κρίθηκε απαραίτητη επειδή στις προηγούμενες δοκιμές ακόμη και μικρή απόκλιση της τάξης των 1–2° μπορούσε να προκαλέσει πολύ μεγάλη επιτάχυνση των κινητήρων, με αποτέλεσμα το ρομπότ να χάνει την ισορροπία αντί να την αποκαθιστά. Παράλληλα εφαρμόστηκε σταδιακή μεταβολή της ταχύτητας των κινητήρων μέσω της παραμέτρου motorSlewSpsPerSec, ώστε να αποφεύγονται απότομα άλματα στην εντολή ταχύτητας.

3.6 Κίνηση και στροφή

Στην τελική έκδοση του προγράμματος η κίνηση προς τα εμπρός και προς τα πίσω υλοποιήθηκε με συνδυασμό μεταβολής της επιθυμητής γωνίας ισορροπίας και πρόσθετης συνιστώσας ταχύτητας στους δύο κινητήρες. Η εντολή forwardCmd, η οποία προκύπτει από την πρώτη παράμετρο της εντολής JOY, χρησιμοποιείται αρχικά για τη μετατόπιση της γωνίας αναφοράς. Με τον τρόπο αυτό το ρομπότ αποκτά ελεγχόμενη κλίση προς την επιθυμητή κατεύθυνση και ο PID ελεγκτής προσπαθεί να διατηρήσει τη νέα δυναμική κατάσταση.

Επειδή η συμπεριφορά του πρωτοτύπου δεν ήταν απόλυτα συμμετρική προς τις δύο κατευθύνσεις, χρησιμοποιήθηκαν διαφορετικές τιμές επιθυμητής κλίσης. Στον τελικό κώδικα η παράμετρος

forwardLeanDeg ορίστηκε σε $1,5^\circ$, ενώ η backwardLeanDeg σε $0,9^\circ$. Η επιθυμητή γωνία υπολογίζεται σε σχέση με τη γωνία που έχει προκύψει από τη διαδικασία ZERO και μεταβάλλεται ανάλογα με το πρόσημο και το μέγεθος της εντολής κίνησης.

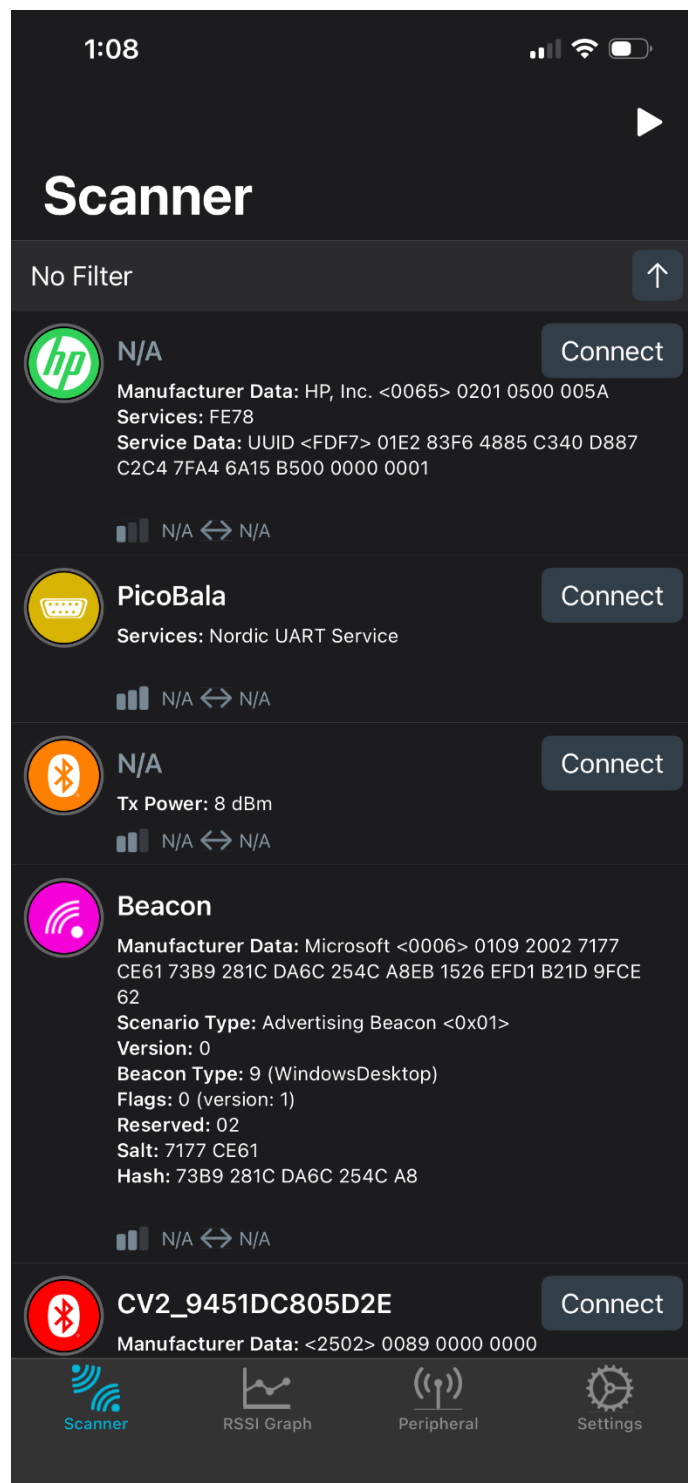
Παράλληλα, στην τελική έκδοση προστέθηκε ανεξάρτητη συνιστώσα μεταφορικής ταχύτητας. Για κίνηση προς τα εμπρός χρησιμοποιήθηκε μέγιστη τιμή forwardDriveStepsPerSec=3000, ενώ για κίνηση προς τα πίσω backwardDriveStepsPerSec=2500. Η τιμή αυτή πολλαπλασιάζεται με την αντίστοιχη εντολή του χρήστη και προστίθεται στην ταχύτητα που παράγεται από τον PID. Με αυτόν τον τρόπο, η τελική εντολή κάθε κινητήρα αποτελεί συνδυασμό της ταχύτητας που απαιτείται για την ισορροπία, της μεταφορικής κίνησης και της διαφορικής συνιστώσας που απαιτείται για τη στροφή.

Κατά τις πραγματικές δοκιμές διαπιστώθηκε ότι δεν υπήρχε μία μοναδική τιμή JOY η οποία να προκαλεί πάντοτε την ίδια μετατόπιση. Η απόκριση επηρεαζόταν από τη γωνία αναφοράς που είχε καθοριστεί κατά τη διαδικασία ZERO, από την τριβή μεταξύ των τροχών και της επιφάνειας, από μικρή πιθανή κλίση του δαπέδου και από τη στιγμιαία κατανομή του βάρους. Για τον λόγο αυτό η ίδια εντολή μπορούσε σε μία δοκιμή να προκαλέσει πραγματική μετακίνηση, ενώ σε άλλη να οδηγήσει μόνο σε μικρή αλλαγή της γωνίας του σώματος ή σε σύντομη μετατόπιση και επιστροφή προς το σημείο ισορροπίας.

Ενδεικτικά, σε ορισμένες δοκιμές τιμή JOY 0.45 0 προκαλούσε μόνο πολύ μικρή εμπρόσθια μετατόπιση, ενώ σε διαφορετική κατάσταση λειτουργίας τιμή 0,46 μπορούσε να οδηγήσει σε απώλεια της ισορροπίας. Μετά τις αλλαγές στον έλεγχο της ταχύτητας και στη μηχανική διάταξη, επιτυχής εμπρόσθια κίνηση παρατηρήθηκε σε επιφάνεια αυξημένης τριβής με τιμές περίπου 0,76 έως 0,98. Για την οπίσθια κίνηση καταγράφηκε επίσης επιτυχής δοκιμή με JOY 0.66 0. Οι τιμές αυτές παρουσιάζονται ως ενδεικτικά πειραματικά αποτελέσματα και όχι ως απόλυτες τιμές λειτουργίας του συστήματος.

3.7 Ασύρματη επικοινωνία μέσω BLE

Η αρχική σκέψη για Bluetooth Classic δεν ήταν πρακτική για την επικοινωνία με iPhone. Για τον λόγο αυτό χρησιμοποιήθηκε Bluetooth Low Energy και υπηρεσία τύπου UART. Για την πραγματοποίηση των ασύρματων δοκιμών χρησιμοποιήθηκε η εφαρμογή nRF Connect σε κινητό τηλέφωνο. Μετά την ενεργοποίηση του ρομπότ η συσκευή εμφανιζόταν στην εφαρμογή με την ονομασία PicoBala. Ο χρήστης μπορούσε να πραγματοποιήσει σύνδεση μέσω Bluetooth Low Energy και στη συνέχεια να αποκτήσει πρόσβαση στην υπηρεσία επικοινωνίας τύπου UART που είχε δημιουργηθεί από το πρόγραμμα του Raspberry Pi Pico 2 W (Σχήμα 3.6).



Σχήμα 3.6 Περιβάλλον σάρωσης της εφαρμογής nRF Connect.

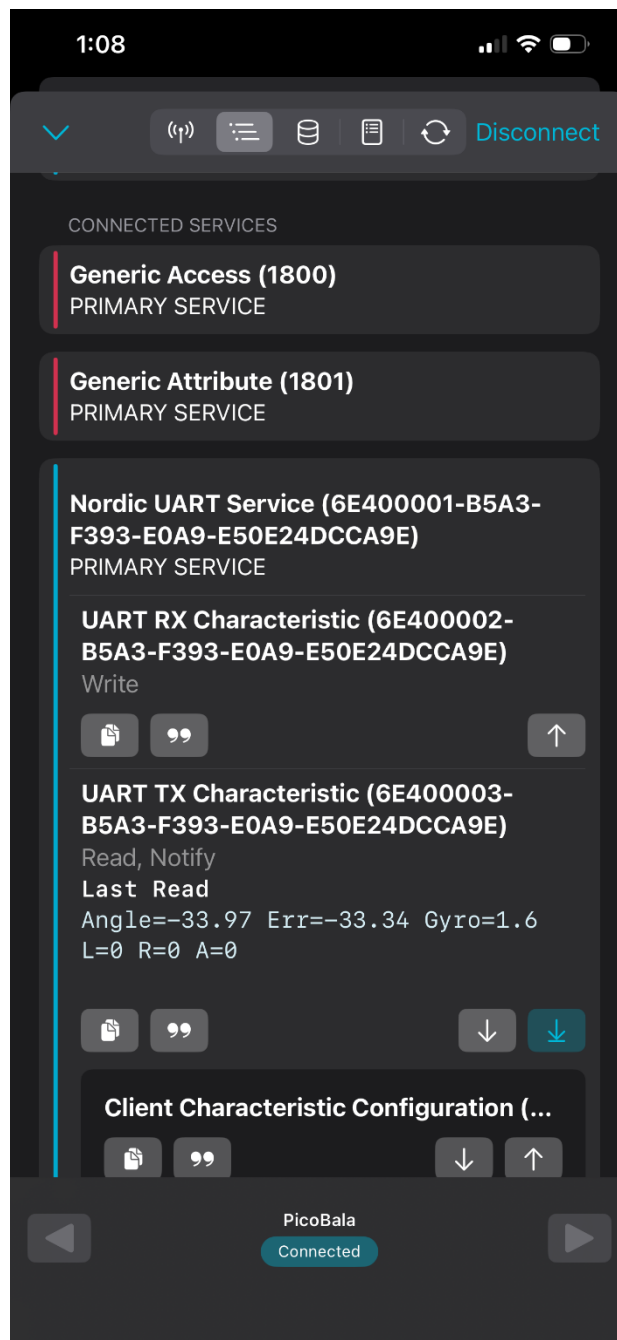
Οι ίδιες εντολές μπορούσαν να σταλούν τόσο από το Serial Monitor όσο και μέσω BLE. Η δυνατότητα αυτή διευκόλυνε σημαντικά την αποσφαλμάτωση, καθώς κάθε εντολή μπορούσε αρχικά να ελεγχθεί μέσω USB και στη συνέχεια να δοκιμαστεί ασύρματα. Ενδεικτικές εντολές της τελικής έκδοσης ήταν:

STATUS;

ZERO;

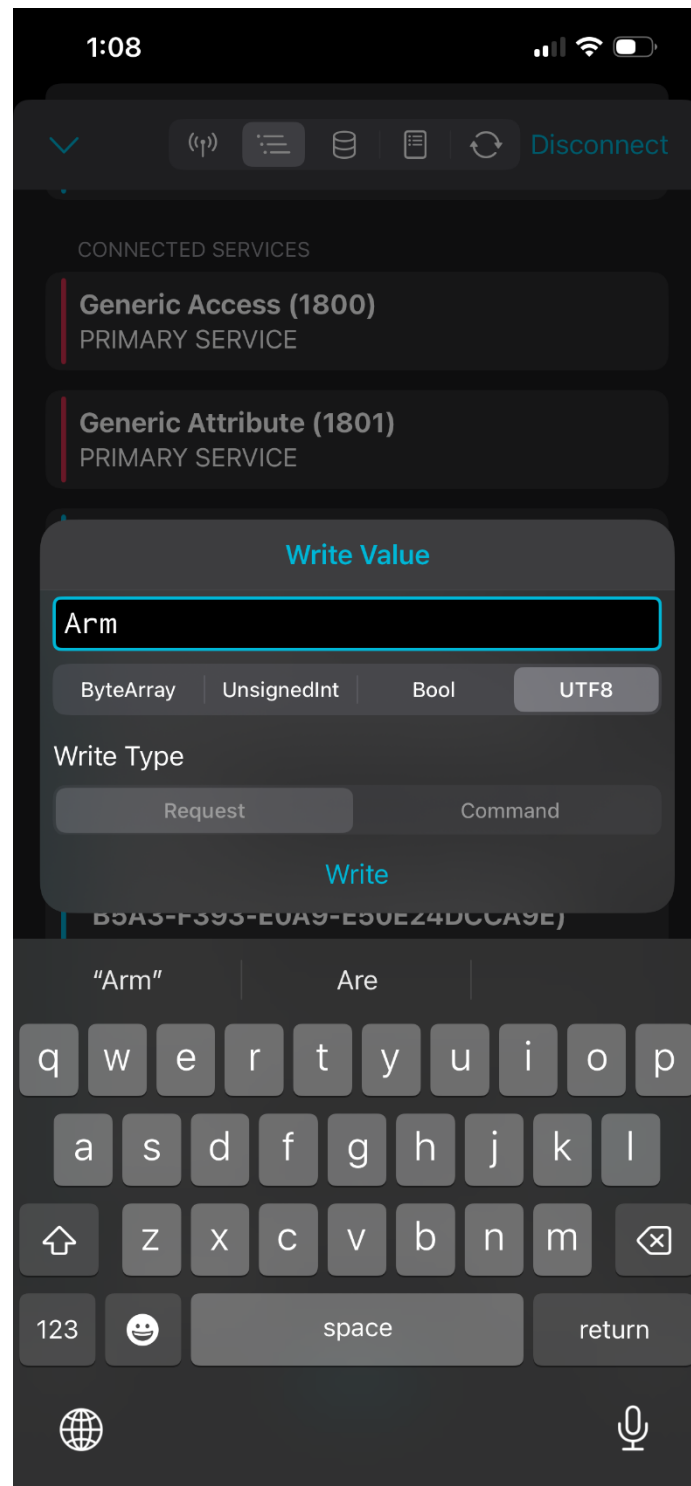
ARM;
DISARM;
PID 35 0 0.08;
GAIN 40;
JOY 0 0;

Ο χρήστης μπορούσε να πραγματοποιήσει σύνδεση μέσω Bluetooth Low Energy και στη συνέχεια να αποκτήσει πρόσβαση στην υπηρεσία επικοινωνίας τύπου UART που δημιουργήθηκε από το πρόγραμμα του Raspberry Pi Pico 2 W (Σχήμα 3.7).



Σχήμα 3.7 Σύνδεση με το ρομπότ και λήψη διαγνωστικών δεδομένων.

Η αποστολή των εντολών πραγματοποιούνται μέσω εγγραφής στο αντίστοιχο BLE characteristic. Οι εντολές καταχωρούνται ως ακολουθίες χαρακτήρων σε μορφή UTF-8, επιτρέποντας την αποστολή από το κινητό των ίδιων εντολών που ήταν διαθέσιμες μέσω του Serial Monitor(Σχήμα 3.8).



Σχήμα 3.8 Αποστολή της εντολής ARM σε μορφή UTF-8.

Κατά τις αρχικές δοκιμές χρησιμοποιήθηκε ο χαρακτήρας ; ως σαφής τερματισμός της εντολής. Στην τελική έκδοση του λογισμικού ο parser σχεδιάστηκε ώστε να αναγνωρίζει ως τέλος εντολής τόσο τον χαρακτήρα νέας γραμμής όσο και το ;. Επιπλέον, προστέθηκε χρονικός μηχανισμός περίπου 80 ms,

ώστε μία εντολή να μπορεί να εκτελεστεί ακόμη και όταν η εφαρμογή δεν αποστέλλει ρητό χαρακτήρα τερματισμού. Έτσι είναι δυνατή τόσο η αποστολή STATUS; όσο και η αποστολή STATUS χωρίς πρόσθετο χαρακτήρα στο τέλος.

Παρότι ο τελικός κώδικας μπορεί να διαχειριστεί και τις δύο περιπτώσεις, κατά τις δοκιμές χρησιμοποιήθηκε κυρίως ο χαρακτήρας ;, επειδή παρείχε σαφή οριοθέτηση μεταξύ διαδοχικών εντολών και διευκόλυνε τον έλεγχο μέσω της εφαρμογής.

Η εντολή STATUS επέστρεφε πληροφορίες όπως η τρέχουσα γωνία, το offset ισορροπίας, το σφάλμα, η γωνιακή ταχύτητα, η κατάσταση armed και οι εντολές ταχύτητας των δύο κινητήρων. Τα δεδομένα αυτά χρησιμοποιήθηκαν για τη διάκριση μεταξύ προβλημάτων αισθητήρα, κινητήρων και ρύθμισης του PID.

3.8 Μηχανισμοί ασφαλείας

Με την εκκίνηση του προγράμματος η μεταβλητή armed είναι false και οι κινητήρες δεν ελέγχονται από τον PID. Ο χρήστης τοποθετεί το ρομπότ κοντά στην κατακόρυφη θέση, στέλνει ZERO για να ορίσει τη γωνία αναφοράς και στη συνέχεια ARM. Οι DISARM και STOP διακόπτουν άμεσα την ενεργή λειτουργία.

Η πτώση ανιχνεύεται από την απόλυτη τιμή του σφάλματος:

```
bool fallCandidate = fabsf(error) > FALL_ANGLE_DEG;
```

Η υπέρβαση του ορίου δεν οδηγεί αμέσως σε διακοπή, επειδή μία μεμονωμένη μέτρηση μπορεί να οφείλεται σε θόρυβο. Η κλίση πρέπει να παραμείνει εκτός ορίου για χρονικό διάστημα FALL_TIME_MS. Τότε μηδενίζονται οι εντολές και ο ολοκληρωτικός όρος, σταματούν οι κινητήρες και αποστέλλεται το μήνυμα FALL STOP.

```
if (fallStartMs != 0 && millis() - fallStartMs > FALL_TIME_MS) {  
    armed = false;  
    fallen = true;  
    fallStartMs = 0;  
    pidIntegral = 0;  
    forwardCmd = 0;  
    turnCmd = 0;  
    stopMotors();  
    printBoth("FALL STOP");  
}
```

Για τις εντολές κίνησης εφαρμόστηκε επιπλέον χρονικό όριο τριών δευτερολέπτων. Αν δεν ληφθεί νέα εντολή για περισσότερο από 3000 ms, οι μεταβλητές forwardCmd και turnCmd μηδενίζονται. Με αυτόν τον τρόπο το ρομπότ δεν συνεχίζει επ' αόριστον να εκτελεί την τελευταία εντολή κίνησης σε περίπτωση διακοπής της επικοινωνίας.

```
if (millis() - lastCommandMs > 3000) {  
    forwardCmd = 0;
```

```
turnCmd = 0;  
}
```

Στην τελική έκδοση του προγράμματος η γωνία πτώσης ορίστηκε σε $FALL_ANGLE_DEG=20^\circ$, ενώ η κατάσταση αυτή έπρεπε να διατηρηθεί για περισσότερο από $FALL_TIME_MS=250$ ms πριν ενεργοποιηθεί η αυτόματη διακοπή. Η χρονική επιβεβαίωση χρησιμοποιήθηκε ώστε μία στιγμιαία μέτρηση ή ένας κραδασμός να μην προκαλεί άμεση απενεργοποίηση του συστήματος. Με την επιβεβαίωση της πτώσης το πρόγραμμα μηδενίζει τις εντολές κίνησης, επαναφέρει τις σχετικές μεταβλητές και διακόπτει τους κινητήρες.

3.9 Δοκιμές, προβλήματα και διορθώσεις

Οι δοκιμές έγιναν σε στάδια. Η σειρά ήταν σημαντική, επειδή η ενεργοποίηση του PID πριν ελεγχθούν οι βασικές λειτουργίες μπορούσε να οδηγήσει σε απότομη κίνηση. Αρχικά επιβεβαιώθηκε η μεταφόρτωση του προγράμματος και η σειριακή επικοινωνία. Στη συνέχεια ελέγχθηκαν ο αισθητήρας, οι κινητήρες σε ανοιχτό βρόχο, η φορά και το BLE. Ο PID ενεργοποιήθηκε τελευταίος.

Η διαδικασία αυτή έδειξε ότι τα προβλήματα δεν είχαν μία μόνο πηγή. Μια λανθασμένη αντίδραση μπορούσε να οφείλεται στο πρόσημο της γωνίας, στη φορά του ενός κινητήρα ή στο συνολικό πρόσημο της εξόδου. Για τον λόγο αυτό κάθε παράμετρος ελεγχόταν χωριστά.

Η μηχανική διάταξη επηρέαζε άμεσα το λογισμικό. Ένα επιπλέον πρόβλημα εμφανίστηκε κατά την προσπάθεια μεταφορικής κίνησης. Σε προηγούμενες εκδόσεις του προγράμματος, όταν η γωνία αποκλίνει ακόμη και κατά $1-2^\circ$, η έξοδος του ελεγκτή μπορούσε να ζητήσει πολύ υψηλή ταχύτητα από τους βηματικούς κινητήρες. Το αποτέλεσμα ήταν μία απότομη αντίδραση η οποία δεν οδηγούσε πάντα σε αποκατάσταση της ισορροπίας. Για την αντιμετώπιση του προβλήματος προστέθηκε περιορισμός στη συνιστώσα ταχύτητας που προκύπτει από τον PID, καθώς και σταδιακή μεταβολή της τελικής εντολής προς τους κινητήρες. Μετά τις αλλαγές αυτές η απόκριση έγινε πιο ελεγχόμενη και κατέστη δυνατή η πραγματική μετακίνηση του ρομπότ. Μετά από αλλαγή στη θέση του powerbank ή στη στερέωση των πλακετών, η συμπεριφορά του ίδιου PID μπορούσε να αλλάξει. Αντίστοιχα, μια χαμηλή ρύθμιση ρεύματος έμοιαζε με μικρό K_p , επειδή οι κινητήρες δεν μπορούσαν να εκτελέσουν την εντολή. Η τελική ρύθμιση επομένως ήταν αποτέλεσμα συνδυασμού ηλεκτρικών, μηχανικών και προγραμματιστικών αλλαγών.

Σημαντική επίδραση είχε επίσης η επιφάνεια δοκιμής. Σε λείο δάπεδο παρατηρήθηκε ότι η εντολή JOY μπορούσε να αλλάξει τη γωνία του ρομπότ χωρίς να προκαλέσει πάντοτε την αναμενόμενη μετατόπιση. Αντίθετα, όταν το ρομπότ τοποθετήθηκε πάνω σε πετσέτα, δηλαδή σε επιφάνεια με μεγαλύτερη τριβή, επιτεύχθηκε πιο αξιόπιστη εμπρόσθια κίνηση. Η παρατήρηση αυτή έδειξε ότι η συμπεριφορά του συστήματος δεν καθορίζεται μόνο από τις αριθμητικές παραμέτρους του ελεγκτή, αλλά και από τη δυνατότητα των τροχών να μεταφέρουν τη διαθέσιμη ροπή στην επιφάνεια χωρίς ανεπιθύμητη ολίσθηση.

Κατά τις δοκιμές εξετάστηκε επίσης η επίδραση πρόσθετου φορτίου. Με αντικείμενο μάζας περίπου 200 g το ρομπότ εξακολουθούσε να μπορεί να διατηρήσει την ισορροπία του και να λειτουργεί χωρίς να απαιτείται αλλαγή των βασικών παραμέτρων ελέγχου. Με βάση τις δοκιμές που πραγματοποιήθηκαν, το σύστημα μπορεί να υποστηρίξει φορτίο περίπου έως 200 g, γεγονός που επιβεβαιώνει τη δυνατότητα αξιοποίησής του για τη μεταφορά ελαφρών αντικειμένων.

3.10 Τελικό αποτέλεσμα και αξιολόγηση

Το τελικό πρωτότυπο ήταν σε θέση να διατηρεί δυναμικά την ισορροπία του, να δέχεται εντολές ασύρματα μέσω BLE και να μεταβάλλει ανεξάρτητα τη μεταφορική και τη στροφική συνιστώσα της κίνησης. Η τελική λειτουργία επιτεύχθηκε με συνδυασμό επεξεργασίας των μετρήσεων του MPU6050, συμπληρωματικού φίλτρου, PID ελέγχου, περιορισμού της ταχύτητας διόρθωσης και σταδιακής μεταβολής της εντολής προς τους βηματικούς κινητήρες.

Η μεταφορική κίνηση αποδείχθηκε περισσότερο ευαίσθητη στις συνθήκες λειτουργίας από την απλή διατήρηση της ισορροπίας. Η απαιτούμενη τιμή JOY μεταβαλλόταν ανάλογα με τη διαδικασία ZERO, την τριβή της επιφάνειας, πιθανές μικρές κλίσεις του δαπέδου και την κατανομή της μάζας της κατασκευής. Για τον λόγο αυτό οι τιμές JOY που καταγράφηκαν κατά τις δοκιμές αντιμετωπίζονται ως ενδεικτικές τιμές για τις αντίστοιχες συνθήκες και όχι ως σταθερές παράμετροι του συστήματος.

Επιβεβαιώθηκε η δυνατότητα ισορροπίας, ασύρματου χειρισμού, εμπρόσθιας και οπίσθιας κίνησης υπό κατάλληλες συνθήκες, καθώς και στροφής μέσω διαφορικής κίνησης των τροχών. Σε επιφάνεια αυξημένης τριβής επιτεύχθηκε πιο αξιόπιστη μετακίνηση, ενώ η δοκιμή με πρόσθετη μάζα επιβεβαίωσε τη δυνατότητα διατήρησης της ισορροπίας υπό φορτίο. Συνολικά, το τελικό σύστημα χαρακτηρίζεται ως λειτουργικό πειραματικό πρωτότυπο δίτροχης αυτοεξισορροπούμενης πλατφόρμας με δυνατότητα ασύρματου χειρισμού και μεταφοράς ελαφρών αντικειμένων..

Κεφάλαιο 4ο: Συμπεράσματα και Μελλοντικές Βελτιώσεις

4.1 Συμπεράσματα

Η εργασία οδήγησε στην κατασκευή ενός αυτοεξισορροπούμενου ρομπότ δύο τροχών με ασύρματο χειρισμό. Το τελικό σύστημα συνδύασε μηχανικό πλαίσιο, δύο βηματικούς κινητήρες, οδηγούς DRV8825, αισθητήρα MPU6050 και Raspberry Pi Pico 2 W. Η λειτουργία του δεν βασίστηκε σε ένα μεμονωμένο εξάρτημα, αλλά στη σωστή συνεργασία όλων των τμημάτων.

Η βαθμονόμηση του MPU6050 και η επιλογή σωστού άξονα αποδείχθηκαν απαραίτητες. Τα offsets δεν μπορούν να θεωρηθούν γενικά, επειδή αφορούν τον συγκεκριμένο αισθητήρα και τη συγκεκριμένη τοποθέτηση. Το συμπληρωματικό φίλτρο έδωσε πιο χρήσιμη γωνία από τη χρήση μόνο του επιταχυνσιόμετρου ή μόνο του γυροσκοπίου.

Η λειτουργία των κινητήρων εξαρτήθηκε από τρεις διαφορετικές ρυθμίσεις: τη φορά κάθε κινητήρα, το συνολικό πρόσημο της αντίδρασης και την ένταση της εξόδου. Η διάκριση αυτών των παραμέτρων διευκόλυνε την αποσφαλμάτωση. Αντί να αλλάζει κάθε φορά η καλωδίωση, οι διορθώσεις μπορούσαν να γίνουν με `dirInvertL`, `dirInvertR` και `motorSign`.

Ο PID ελεγκτής χρειάστηκε πειραματική ρύθμιση. Οι τιμές του επηρεάζονταν από τη θέση των μπαταριών, την ακαμψία του πλαισίου, τη ρύθμιση ρεύματος και την πρόσφυση των τροχών. Η εμπειρία αυτή έδειξε ότι ο έλεγχος δεν μπορεί να αντιμετωπιστεί ανεξάρτητα από την πραγματική κατασκευή.

Οι δοκιμές μεταφορικής κίνησης ανέδειξαν επίσης τη σημασία των εξωτερικών συνθηκών. Η ίδια εντολή JOY δεν οδηγούσε πάντοτε στην ίδια πραγματική μετατόπιση, επειδή η απόκριση εξαρτιόταν από τη γωνία αναφοράς μετά το ZERO, την πρόσφυση των τροχών, την κλίση της επιφάνειας και την κατανομή της μάζας. Η πιο αξιόπιστη εμπρόσθια κίνηση επιτεύχθηκε σε επιφάνεια αυξημένης τριβής. Το αποτέλεσμα αυτό επιβεβαιώνει ότι σε μία αυτοεξισορροπούμενη πλατφόρμα ο αλγόριθμος ελέγχου, η μηχανική σχεδίαση και η αλληλεπίδραση τροχού-εδάφους αποτελούν αλληλοεξαρτώμενα τμήματα του ίδιου συστήματος.

Το BLE δεν χρησιμοποιήθηκε μόνο ως μέσο χειρισμού. Αποτέλεσε εργαλείο δοκιμών, επειδή επέτρεψε αλλαγή παραμέτρων και ανάγνωση διαγνωστικών δεδομένων χωρίς νέα μεταφόρτωση. Η χρήση σαφούς τερματικού χαρακτήρα στις εντολές έλυσε το πρόβλημα διαφορετικής συμπεριφοράς μεταξύ Serial Monitor και εφαρμογής κινητού.

Η κατασκευή εκπλήρωσε τον βασικό στόχο της ανάπτυξης μιας δίτροχης αυτοεξισορροπούμενης πλατφόρμας με ασύρματο χειρισμό. Η επιτυχής διατήρηση της ισορροπίας, η δυνατότητα ασύρματης αποστολής εντολών και η εκτέλεση μεταφορικών και στροφικών κινήσεων επιβεβαιώνουν τη λειτουργικότητα του τελικού πρωτοτύπου. Παράλληλα, η δυνατότητα διατήρησης της ισορροπίας με πρόσθετη μάζα περίπου 200 g καταδεικνύει τη δυνατότητα αξιοποίησης της πλατφόρμας για τη μεταφορά ελαφρών αντικειμένων.

Τέλος, οι μηχανισμοί ασφαλείας ήταν αναγκαίοι για ένα σύστημα που μπορεί να κινηθεί απότομα. Η χειροκίνητη ενεργοποίηση, το FALL STOP και ο μηδενισμός παλιών εντολών περιόρισαν την ανεξέλεγκτη κίνηση και έκαναν τις επαναλαμβανόμενες δοκιμές ασφαλέστερες.

4.2 Περιορισμοί

Το πλαίσιο κατασκευάστηκε με διαθέσιμα υλικά και προσωρινές μεθόδους στερέωσης. Τα δεματικά και η διαφανής ταινία ήταν πρακτικά για την ανάπτυξη, αλλά δεν προσφέρουν την ακαμψία μιας ειδικά σχεδιασμένης βάσης. Μικρές μετακινήσεις εξαρτημάτων μπορούν ακόμη να αλλάξουν το κέντρο βάρους.

Επιπλέον, η τελική γεωμετρία της κατασκευής διατηρεί μεγάλο μέρος των βαρύτερων στοιχείων, όπως τις μπαταρίες και το powerbank, σχετικά κοντά στον άξονα των τροχών. Η συγκεκριμένη διάταξη προέκυψε από τις πρακτικές ανάγκες στερέωσης και από πειραματική αναζήτηση καλύτερης ισορροπίας. Ωστόσο, το ύψος και η ακριβής θέση του κέντρου μάζας αποτελούν παραμέτρους που επηρεάζουν τη δυναμική συμπεριφορά ενός αυτοεξισορροπούμενου συστήματος.

Ο MPU6050 είναι οικονομικός και κατάλληλος για πειραματική χρήση, όμως οι μετρήσεις του περιέχουν θόρυβο. Η ποιότητα της γωνίας εξαρτάται από τη στερέωση, τη βαθμονόμηση και τις παραμέτρους του φίλτρου. Οι βηματικοί κινητήρες επίσης μπορούν να χάσουν βήματα όταν η ζητούμενη ταχύτητα ή το φορτίο ξεπερνούν τις δυνατότητές τους.

Οι τιμές PID αφορούν την τελική μορφή του συγκεκριμένου πρωτοτύπου και εξαρτώνται από τα επιμέρους χαρακτηριστικά της κατασκευής. Αλλαγές στο βάρος, στους τροχούς, στην τροφοδοσία ή στη θέση του αισθητήρα μπορούν να μεταβάλουν τη δυναμική συμπεριφορά του συστήματος και, επομένως, να απαιτήσουν εκ νέου ρύθμιση των παραμέτρων του ελεγκτή. Επομένως, οι συγκεκριμένες τιμές δεν μπορούν να θεωρηθούν ανεξάρτητες από τη μηχανική και ηλεκτρονική διαμόρφωση του πρωτοτύπου.

Ένας ακόμη περιορισμός είναι ότι οι βηματικοί κινητήρες λειτουργούν χωρίς άμεση μέτρηση της πραγματικής περιστροφής των τροχών. Το πρόγραμμα γνωρίζει τη συχνότητα των παλμών STEP που έχει ζητηθεί, αλλά δεν λαμβάνει πληροφορία που να επιβεβαιώνει ότι ο αντίστοιχος αριθμός βημάτων πραγματοποιήθηκε στην πράξη. Σε συνθήκες ανεπαρκούς πρόσφυσης, αυξημένου φορτίου ή απότομης επιτάχυνσης είναι επομένως δυνατό η πραγματική μετακίνηση να διαφέρει από την επιθυμητή. Η συμπεριφορά αυτή παρατηρήθηκε και στις δοκιμές JOY, όπου σε ορισμένες περιπτώσεις δινόταν εντολή κίνησης χωρίς να επιτυγχάνεται αντίστοιχη μετατόπιση του ρομπότ.

4.3 Μελλοντικές βελτιώσεις

Η μηχανική κατασκευή μπορεί να βελτιωθεί με τη δημιουργία ενός νέου πλαισίου στο οποίο οι θέσεις των βαριών εξαρτημάτων θα είναι προκαθορισμένες και επαναλήψιμες. Στο παρόν πρωτότυπο χρησιμοποιήθηκαν ακρυλικές πλάκες, μεταλλικοί αποστάτες, δεματικά και βοηθητικές μέθοδοι στερέωσης, οι οποίες επέτρεψαν εύκολες αλλαγές κατά την ανάπτυξη. Σε μία επόμενη έκδοση θα μπορούσαν να προβλεφθούν ειδικές βάσεις για την μπαταρία, το powerbank, την πλακέτα και τον αισθητήρα, ώστε η θέση τους να μην μεταβάλλεται μεταξύ των δοκιμών. Με τον τρόπο αυτό η διαδικασία ρύθμισης του PID θα είναι περισσότερο επαναλήψιμη.

Η ηλεκτρονική συνδεσμολογία μπορεί να μεταφερθεί σε ειδικά σχεδιασμένη πλακέτα. Έτσι θα μειωθούν οι καλωδιώσεις, οι χαλαρές επαφές και ο όγκος του κυκλώματος. Μπορούν επίσης να προστεθούν ασφάλεια, κεντρικός διακόπτης, προστασία από αντίστροφη πολικότητα και ένδειξη στάθμης μπαταρίας.

Στο λογισμικό θα ήταν χρήσιμη η καταγραφή δεδομένων σε υπολογιστή ή αρχείο. Η γωνία, το σφάλμα, η έξοδος του PID και οι ταχύτητες των κινητήρων μπορούν να αποθηκεύονται για κάθε δοκιμή. Με αυτά τα δεδομένα η ρύθμιση δεν θα βασίζεται μόνο στην οπτική παρατήρηση αλλά και σε μετρήσιμα χαρακτηριστικά, όπως ο χρόνος αποκατάστασης και το εύρος της ταλάντωσης.

Μια πιο ολοκληρωμένη εφαρμογή κινητού θα μπορούσε να διαθέτει ξεχωριστά κουμπιά για ARM, DISARM και ZERO, πεδία ή sliders για τις παραμέτρους και εικονικό joystick. Η εφαρμογή θα μπορούσε επίσης να εμφανίζει γραφικά την κλίση και την κατάσταση πτώσης.

Ακόμα μία από τις σημαντικότερες μελλοντικές βελτιώσεις αφορά την προσθήκη ανατροφοδότησης από τους τροχούς. Στην παρούσα κατασκευή οι βηματικοί κινητήρες ελέγχονται μέσω των παλμών STEP και θεωρείται ότι ακολουθούν την ταχύτητα που ζητείται από το πρόγραμμα. Δεν υπάρχει όμως αισθητήρας που να επιβεβαιώνει την πραγματική περιστροφή τους. Η προσθήκη περιστροφικών κωδικοποιητών (encoders) στους τροχούς ή στους άξονες των κινητήρων θα επέτρεπε τη μέτρηση της πραγματικής γωνιακής ταχύτητας και της μετατόπισης κάθε τροχού.

Με την πληροφορία αυτή θα μπορούσε να δημιουργηθεί επιπλέον κλειστός βρόχος ελέγχου της ταχύτητας. Όταν ο χρήστης δίνει εντολή για εμπρόσθια ή οπίσθια κίνηση, το σύστημα θα μπορούσε να συγκρίνει την επιθυμητή με την πραγματική ταχύτητα και να διορθώνει αυτόματα τη διαφορά. Η προσέγγιση αυτή θα ήταν ιδιαίτερα χρήσιμη στις περιπτώσεις όπου κατά τις δοκιμές δινόταν εντολή JOY αλλά το ρομπότ δεν πραγματοποιούσε την αναμενόμενη μετατόπιση.

Η ανατροφοδότηση από encoders θα επέτρεπε επίσης καλύτερο έλεγχο της επιβράδυνσης και της ακινητοποίησης. Αντί η εντολή JOY 0 0 να μηδενίζει μόνο την επιθυμητή εντολή, ο ελεγκτής θα μπορούσε να γνωρίζει εάν οι τροχοί εξακολουθούν να περιστρέφονται και να εφαρμόσει ενεργό διόρθωση μέχρι η πραγματική ταχύτητα να πλησιάσει το μηδέν. Παράλληλα θα μπορούσαν να ανιχνεύονται απώλειες βημάτων ή αποκλίσεις μεταξύ της ζητούμενης και της πραγματικής περιστροφής των τροχών, βελτιώνοντας την επαναληψιμότητα της μετακίνησης.

Μία δεύτερη κατεύθυνση μελλοντικής έρευνας είναι η πειραματική μεταβολή του ύψους της κατασκευής και της θέσης του κέντρου μάζας. Στην τελική μορφή του πρωτοτύπου οι μπαταρίες και το powerbank βρίσκονται σχετικά χαμηλά στο πλαίσιο, ενώ η κύρια ηλεκτρονική πλακέτα βρίσκεται στο ανώτερο επίπεδο. Μία ελεγχόμενη μεταβολή της γεωμετρίας και της κατανομής της μάζας θα μπορούσε να χρησιμοποιηθεί για τη μελέτη της επίδρασής τους στη δυναμική συμπεριφορά του ρομπότ και στη ρύθμιση του ελεγκτή.

Τέλος, οι δοκιμές έδειξαν επίσης ότι η πρόσφυση αποτελεί σημαντικό παράγοντα για τη μεταφορική λειτουργία. Μελλοντικά θα μπορούσαν να εξεταστούν διαφορετικά υλικά ή γεωμετρίες τροχών με καλύτερα χαρακτηριστικά πρόσφυσης. Η χρήση κατάλληλου ελαστικού πέλματος θα μπορούσε να περιορίσει την ολίσθηση σε λείες επιφάνειες και να μειώσει την εξάρτηση της επιτυχούς κίνησης από το είδος του δαπέδου.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] C. Llopis-Albert et al., "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 2, pp. 1–22, Mar.-Apr. 2019.
- [2] M. Ben-Ari and F. Mondada, *Elements of Robotics*. Cham: Springer International Publishing, 2018.
- [3] S. Sakunthala, R. Kiranmayi, and P. Nagaraju Mandadi, "A study on industrial motor drives: Comparison and applications of PMSM and BLDC motor drives," in *Proc. Int. Conf. on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 2017, pp. 537–540.
- [4] A. Alhinqari and A. Alhengari, "Simulation of Stepper Motor Motion and Control," in *Proc. IEEE 1st Int. Maghreb Meeting of the Conf. on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, Tripoli, Libya, May 2021, pp. 573–578.
- [5] T. Wildi, *Handbook of Electric Motors*, 2nd ed. Boca Raton, FL, USA: CRC Press, Taylor & Francis Group, 2004.
- [6] P. Prodanov and D. Dankov, "Reliability of low-power stepper motor drivers," 2022 XXXI International Scientific Conference Electronics (ET), Sozopol, Bulgaria, 2022, pp. 1-5.
- [7] D. E. Bolanakis, "A Survey of Research in Microcontroller Education," in *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, vol. 14, no. 2, pp. 50-57, May 2019
- [8] K. R. Raghunathan, "History of Microcontrollers: First 50 Years," in *IEEE Micro*, vol. 41, no. 6, pp. 97-104, 1 Nov.-Dec. 2021,
- [9] B. A. Munana, A Self-Stabilizing Platform, Bachelor's thesis, Dept. of Electronics, Metropolia Univ. of Applied Sciences, Helsinki, Finland, Apr. 15, 2022.
- [10] Bin Fang, Fuchun Sun, Huaping Liu, Chunfang Liu, "3D human gesture capturing and recognition by the IMMU-based data glove", *Neurocomputing*, Volume 277, 2018, Pages 198-207, ISSN 0925-2312 ,*Procedia Engineering*, Volume 177, 2017, Pages 318-323, ISSN 1877-7058,
- [11] L. Zhou, E. Fischer, C. Tunca, C. M. Brahms, C. Ersoy, U. Granacher, and B. Arnrich, "How We Found Our IMU: Guidelines to IMU Selection and a Comparison of Seven IMUs for Pervasive Healthcare Applications," *Sensors*, vol. 20, no. 14, Art. no. 4090, 2020.
- [12] Raspberry Pi Ltd., Raspberry Pi Pico 2 W Datasheet, Release 2, 2026.
- [13] Texas Instruments, "LM2596 SIMPLE SWITCHER® Power Converter 150-kHz 3-A Step-Down Voltage Regulator," Datasheet, Rev. G, Mar. 2023.
- [14] Texas Instruments, DRV8825 Stepper Motor Controller IC, Datasheet SLVSA73F, Rev. F, Jul. 2014.
- [15] Vishay Semiconductors, 6N137, VO2601, VO2611: High Speed Optocoupler, 10 MBd, Rev. 2.3, Apr. 2024.
- [16] InvenSense Inc., "MPU-6000 and MPU-6050 Product Specification," Document No. PS-MPU-6000A-00, Rev. 3.4, Aug. 2013.

ΠΑΡΑΡΤΗΜΑ Α : Κώδικας Ρύθμισης MPU

```
#include <Wire.h>

const int CALIBRATION_SAMPLES = 2000;

long ax_offset = 0, ay_offset = 0, az_offset = 0;
long gx_offset = 0, gy_offset = 0, gz_offset = 0;

const int MPU_ADDR = 0x68;

//Prosthiki timis sto antistoixο reg
void writeRegister(uint8_t reg, uint8_t val) {
    Wire.beginTransmission(MPU_ADDR);
    Wire.write(reg);
    Wire.write(val);
    Wire.endTransmission();
}

void setup() {
    Serial.begin(115200);
    while (!Serial);

    //Pin setup
    Wire.setSDA(4);
    Wire.setSCL(5);
    Wire.begin();

    writeRegister(0x6B, 0x00); //energopoihsh mpu
    writeRegister(0x1B, 0x00); //rithmisi gyro +/- 250 deg/s
    writeRegister(0x1C, 0x00); // rithmisi accel +/- 2g

    Serial.println("=====");
    Serial.println("      MPU6050 CALIBRATION UTILITY      ");
    Serial.println("=====");
    Serial.println("1. Place the robot exactly upright and level.");
```

```

Serial.println("2. Do NOT touch, vibrate, or move the robot now.");
Serial.println("Starting calibration in 5 seconds...");
delay(5000);

```

```

Serial.println("Calculating offsets... Please wait...");

```

```

for (int i = 0; i < CALIBRATION_SAMPLES; i++) {
  Wire.beginTransaction(MPU_ADDR);
  Wire.write(0x3B);
  Wire.endTransmission(false);

  // 14 byte apo kathe timi
  Wire.requestFrom((uint8_t)MPU_ADDR, (uint8_t)14, (bool>true);

```

```

  long ax = (Wire.read() << 8) | Wire.read();
  long ay = (Wire.read() << 8) | Wire.read();
  long az = (Wire.read() << 8) | Wire.read();
  Wire.read(); Wire.read(); // thermokrasia den mas endiaferei
  long gx = (Wire.read() << 8) | Wire.read();
  long gy = (Wire.read() << 8) | Wire.read();
  long gz = (Wire.read() << 8) | Wire.read();

```

```

  ax_offset += ax;
  ay_offset += ay;
  az_offset += az;
  gx_offset += gx;
  gy_offset += gy;
  gz_offset += gz;

```

```

  if (i % 400 == 0) Serial.print(".");
  delayMicroseconds(1000);
}

```

```

ax_offset /= CALIBRATION_SAMPLES;
ay_offset /= CALIBRATION_SAMPLES;
az_offset /= CALIBRATION_SAMPLES;
gx_offset /= CALIBRATION_SAMPLES;

```

```

gy_offset /= CALIBRATION_SAMPLES;
gz_offset /= CALIBRATION_SAMPLES;

az_offset = az_offset - 16384;//logo varititas

Serial.println("\n\n=====");
Serial.println("CALIBRATION COMPLETE!");
Serial.println("=====");

Serial.print(" #define ACCEL_X_OFFSET "); Serial.println(-ax_offset);
Serial.print(" #define ACCEL_Y_OFFSET "); Serial.println(-ay_offset);
Serial.print(" #define ACCEL_Z_OFFSET "); Serial.println(-az_offset);
Serial.print(" #define GYRO_X_OFFSET "); Serial.println(-gx_offset);
Serial.print(" #define GYRO_Y_OFFSET "); Serial.println(-gy_offset);
Serial.print(" #define GYRO_Z_OFFSET "); Serial.println(-gz_offset);
Serial.println("=====");
}

void loop() {
}

```

ΠΑΡΑΡΤΗΜΑ Β :Κώδικας Μικροελεγκτη

```

#include <Arduino.h>
#include <Wire.h>
#include <BLE.h>
#include "pico/time.h"
#include "hardware/gpio.h"

//steppers
#define STEP_PIN_L 16
#define DIR_PIN_L 17

#define STEP_PIN_R 18
#define DIR_PIN_R 19

```

```

//mpu
#define I2C_SDA_PIN 4
#define I2C_SCL_PIN 5
#define MPU_ADDR 0x68

//calibrated offsets
#define ACCEL_X_OFFSET -17141
#define ACCEL_Y_OFFSET 418
#define ACCEL_Z_OFFSET 22301
#define GYRO_X_OFFSET 222
#define GYRO_Y_OFFSET 111
#define GYRO_Z_OFFSET 123
#define USE_GIVEN_OFFSETS 1

#define CONTROL_HZ 250
#define CONTROL_PERIOD_US (1000000UL / CONTROL_HZ)

#define STEP_TIMER_HZ 50000
#define STEP_TIMER_PERIOD_US (1000000 / STEP_TIMER_HZ)

#define MAX_STEPS_PER_SEC 25000
#define FALL_ANGLE_DEG 20.0f
#define FALL_TIME_MS 250

//98%gyro 2%accel
#define FILTER_ALPHA 0.98f

//tuning
int angleMode = 5;
float angleSign = 1.0f;
float gyroSign = 1.0f;
int motorSign = 1;
bool dirInvertL = false;
bool dirInvertR = true;

//PID
float motorGain = 40.0f;

```

```

float Kp = 35.0f;
float Ki = 0.0f;
float Kd = 0.08f;

float turnStepsPerSec = 900.0f;

//times gwnias k taxititas stin kinisi
float forwardLeanDeg = 1.5f;
float backwardLeanDeg = 0.9f;

float forwardDriveStepsPerSec = 3000.0f;
float backwardDriveStepsPerSec = 2500.0f;

float integralLimit = 100.0f;

float filteredForwardCmd = 0.0f;
float filteredTurnCmd = 0.0f;
float commandRampRate = 0.60f;

//taxitita diorthosis issoropias
float balanceMaxSps = 4800.0f;
float motorSlewSpsPerSec = 25000.0f;

float smoothLeftSps = 0.0f;
float smoothRightSps = 0.0f;

volatile int32_t targetSpsL = 0;
volatile int32_t targetSpsR = 0;

volatile uint32_t accumL = 0;
volatile uint32_t accumR = 0;
volatile bool stepHighL = false;
volatile bool stepHighR = false;

struct repeating_timer stepTimer;// timer kathe 20 sec
BLEServiceUART uart(128, 128);

```

```

bool armed = false;
bool motorTestMode = false;
bool fallen = false;

float angleDeg = 0.0f;
float gyroRateDps = 0.0f;
float balanceOffsetDeg = 0.0f;
float forwardCmd = 0.0f;
float turnCmd = 0.0f;
float pidIntegral = 0.0f;

uint32_t lastCommandMs = 0;
uint32_t lastStatusMs = 0;
uint32_t fallStartMs = 0;

// buffers gia USB , BLE.
String usbLine = "";
String bleLine = "";
uint32_t usbLastByteMs = 0;
uint32_t bleLastByteMs = 0;//den xriazete ; sto telos tis entolis

float lastAX = 0, lastAY = 0, lastAZ = 0;
float lastGX = 0, lastGY = 0, lastGZ = 0;

//epistrofi apolitis timis
static inline int32_t iabs32(int32_t x) {
    return (x < 0) ? -x : x;
}

bool stepperTimerCallback(struct repeating_timer *t) {
    (void)t;

    if (stepHighL) {
        gpio_put(STEP_PIN_L, 0);
        stepHighL = false;
    } else {
        uint32_t mag = (uint32_t)iabs32(targetSpsL);

```

```

if (mag > 0) {
    accumL += mag;
    //dimiourgia step
    if (accumL >= STEP_TIMER_HZ) {
        accumL -= STEP_TIMER_HZ;
        gpio_put(STEP_PIN_L, 1);
        stepHighL = true;
    }
    //step low
} else {
    accumL = 0;
    gpio_put(STEP_PIN_L, 0);
}
}

if (stepHighR) {
    gpio_put(STEP_PIN_R, 0);
    stepHighR = false;
} else {
    uint32_t mag = (uint32_t)iabs32(targetSpsR);
    if (mag > 0) {
        accumR += mag;
        if (accumR >= STEP_TIMER_HZ) {
            accumR -= STEP_TIMER_HZ;
            gpio_put(STEP_PIN_R, 1);
            stepHighR = true;
        }
    } else {
        accumR = 0;
        gpio_put(STEP_PIN_R, 0);
    }
}
return true;
}

//periorismos taxititas
int32_t clampSps(float v) {

```

```

    if (v > MAX_STEPS_PER_SEC) return MAX_STEPS_PER_SEC;
    if (v < -MAX_STEPS_PER_SEC) return -MAX_STEPS_PER_SEC;
    return (int32_t)v;
}

```

//taxitita k twm dio moter

```

void setMotorTargets(float leftSps, float rightSps) {
    int32_t l = clampSps(leftSps);
    int32_t r = clampSps(rightSps);
    bool dirL = (l >= 0);
    bool dirR = (r >= 0);

```

```

    if (dirInvertL) dirL = !dirL;
    if (dirInvertR) dirR = !dirR;

```

```

    gpio_put(DIR_PIN_L, dirL ? 1 : 0);
    gpio_put(DIR_PIN_R, dirR ? 1 : 0);

```

```

    noInterrupts();
    targetSpsL = l;
    targetSpsR = r;
    interrupts();
}

```

//stamatima moter

```

void stopMotors() {
    noInterrupts();
    targetSpsL = 0;
    targetSpsR = 0;
    accumL = 0;
    accumR = 0;
    stepHighL = false;
    stepHighR = false;
    interrupts();

```

```

    gpio_put(STEP_PIN_L, 0);
    gpio_put(STEP_PIN_R, 0);

```

```

}

//anatrofodotisi mpu
int16_t makeInt16(uint8_t hi, uint8_t lo) {
    return (int16_t)((hi << 8) | lo);
}

//kataxorise times se register
void mpuWrite(uint8_t reg, uint8_t data) {
    Wire.beginTransaction(MPU_ADDR);
    Wire.write(reg);
    Wire.write(data);
    Wire.endTransmission(true);
}

//diabase on mpu
bool readMPURaw(int16_t &ax, int16_t &ay, int16_t &az, int16_t &gx, int16_t &gy, int16_t &gz) {
    Wire.beginTransaction(MPU_ADDR);
    Wire.write(0x3B);

    if (Wire.endTransmission(false) != 0) return false;

    if (Wire.requestFrom(MPU_ADDR, (uint8_t)14, (uint8_t) true) != 14) return false;
    uint8_t b[14];
    for (int i = 0; i < 14; i++) b[i] = Wire.read();
    ax = makeInt16(b[0], b[1]);
    ay = makeInt16(b[2], b[3]);
    az = makeInt16(b[4], b[5]);
    gx = makeInt16(b[8], b[9]);
    gy = makeInt16(b[10], b[11]);
    gz = makeInt16(b[12], b[13]);
    return true;
}

//arxikopoihsh toy mpu
bool initMPU() {
    Wire.beginTransaction(MPU_ADDR);

```

```

    if (Wire.endTransmission(true) != 0) return false;
    mpuWrite(0x6B, 0x00);
    delay(100);
    mpuWrite(0x1A, 0x03);
    mpuWrite(0x1B, 0x00);
    mpuWrite(0x1C, 0x00);
    delay(100);
    return true;
}

//epistrofi timwn imu me ta offsets
bool getCorrectedIMU(float &ax, float &ay, float &az, float &gx, float &gy, float &gz) {

    int16_t rax, ray, raz, rgx, rgy, rgz;
    if (!readMPURaw(rax, ray, raz, rgx, rgy, rgz)) return false;
#ifdef USE_GIVEN_OFFSETS
    ax = (float)(rax - ACCEL_X_OFFSET);
    ay = (float)(ray - ACCEL_Y_OFFSET);
    az = (float)(raz - ACCEL_Z_OFFSET);
    gx = ((float)(rgx - GYRO_X_OFFSET)) / 131.0f;
    gy = ((float)(rgy - GYRO_Y_OFFSET)) / 131.0f;
    gz = ((float)(rgz - GYRO_Z_OFFSET)) / 131.0f;
#else
    ax = (float)rax;
    ay = (float)ray;
    az = (float)raz;
    gx = ((float)rgx) / 131.0f;
    gy = ((float)rgy) / 131.0f;
    gz = ((float)rgz) / 131.0f;
#endif
    lastAX = ax;
    lastAY = ay;
    lastAZ = az;
    lastGX = gx;
    lastGY = gy;
    lastGZ = gz;
    return true;
}

```

```
}
```

```
//epilogi angle/accel mode analoga tin topothesis toy mpu
```

```
float accelAngleDeg(float ax, float ay, float az) {  
    switch (angleMode) {  
        case 0: return angleSign * atan2f(ax, az) * 57.2957795f; // gyro Y  
        case 1: return angleSign * atan2f(ax, ay) * 57.2957795f; // gyro Z  
        case 2: return angleSign * atan2f(ay, az) * 57.2957795f; // gyro X  
        case 3: return angleSign * atan2f(az, ay) * 57.2957795f; // gyro X alt  
        case 4: return angleSign * atan2f(az, ax) * 57.2957795f; // gyro Y alt  
        default: return angleSign * atan2f(ay, ax) * 57.2957795f; // gyro Z alt  
    }  
}
```

```
float selectedGyroRate(float gx, float gy, float gz) {  
    switch (angleMode) {  
        case 0: return gyroSign * angleSign * gy;  
        case 1: return gyroSign * angleSign * gz;  
        case 2: return gyroSign * -angleSign * gx;  
        case 3: return gyroSign * angleSign * gx;  
        case 4: return gyroSign * -angleSign * gy;  
        default: return gyroSign * -angleSign * gz;  
    }  
}
```

```
//torini gwnia
```

```
void updateAngle(float dt) {  
    float ax = 0, ay = 0, az = 0, gx = 0, gy = 0, gz = 0;  
    if (!getCorrectedIMU(ax, ay, az, gx, gy, gz)) return;  
    float aAcc = accelAngleDeg(ax, ay, az);  
    float gRate = selectedGyroRate(gx, gy, gz);  
    angleDeg = FILTER_ALPHA * (angleDeg + gRate * dt) + (1.0f - FILTER_ALPHA) * aAcc;  
    gyroRateDps = gRate;  
}
```

```
//arxikopoiisi gonias pou to robot einai orthio
```

```
void zeroBalanceAngle() {  
    float sum = 0.0f;
```

```

for (int i = 0; i < 300; i++) {
    float ax = 0, ay = 0, az = 0, gx = 0, gy = 0, gz = 0;
    if (getCorrectedIMU(ax, ay, az, gx, gy, gz)) sum += accelAngleDeg(ax, ay, az);
    delay(3);
}
balanceOffsetDeg = sum / 300.0f;
angleDeg = balanceOffsetDeg;
pidIntegral = 0.0f;
}

```

//empfanisi idiou minimatos

```

void printBoth(const String &s) {
    Serial.println(s);
    if (uart) uart.println(s);
}

```

//xorismos entolis

```

int splitCommand(String line, String parts[], int maxParts) {
    line.trim();
    int count = 0, start = 0;
    while (count < maxParts) {
        while (start < (int)line.length() && line[start] == ' ') start++;
        if (start >= (int)line.length()) break;
        int end = line.indexOf(' ', start);
        if (end < 0) end = line.length();
        parts[count++] = line.substring(start, end);
        start = end + 1;
    }
    return count;
}

```

//basikes entoles

```

void handleCommand(String line) {
    line.trim();
    if (line.length() == 0) return;
    String p[6];
    int n = splitCommand(line, p, 6);
}

```

```

if (n == 0) return;
p[0].toUpperCase();
String cmd = p[0];
lastCommandMs = millis();

if (cmd == "ARM" || cmd == "A") {
    motorTestMode = false;
    fallen = false;
    fallStartMs = 0;
    pidIntegral = 0;
    armed = true;
    printBoth("ARMED");
} else if (cmd == "DISARM" || cmd == "D" || cmd == "STOP" || cmd == "S") {
    armed = false;
    motorTestMode = false;
    forwardCmd = 0;
    turnCmd = 0;
    filteredForwardCmd = 0;
    filteredTurnCmd = 0;
    smoothLeftSps = 0;
    smoothRightSps = 0;
    stopMotors();
    printBoth("STOPPED");
} else if (cmd == "M" && n >= 2) {
    armed = false;
    motorTestMode = true;
    float s = constrain(p[1].toFloat(), -6000.0f, 6000.0f);
    setMotorTargets(s, s);
    printBoth("MOTOR TEST " + String(s, 0));
} else if (cmd == "INVL" && n >= 2) {
    dirInvertL = (p[1].toInt() != 0);
    printBoth("INVL=" + String(dirInvertL ? 1 : 0));
} else if (cmd == "INVR" && n >= 2) {
    dirInvertR = (p[1].toInt() != 0);
    printBoth("INVR=" + String(dirInvertR ? 1 : 0));
} else if (cmd == "SIGN" && n >= 2) {
    motorSign = (p[1].toInt() < 0) ? -1 : 1;

```

```

    pidIntegral = 0;
    printBoth("MOTOR SIGN=" + String(motorSign));
} else if (cmd == "ASIGN" && n >= 2) {
    angleSign = (p[1].toInt() < 0) ? -1.0f : 1.0f;
    armed = false;
    motorTestMode = false;
    stopMotors();
    zeroBalanceAngle();
    printBoth("ANGLE SIGN=" + String(angleSign, 0));
} else if (cmd == "GSIGN" && n >= 2) {
    gyroSign = (p[1].toInt() < 0) ? -1.0f : 1.0f;
    pidIntegral = 0;
    printBoth("GYRO SIGN=" + String(gyroSign, 0));
} else if (cmd == "AMODE" && n >= 2) {
    angleMode = constrain(p[1].toInt(), 0, 5);
    armed = false;
    motorTestMode = false;
    stopMotors();
    zeroBalanceAngle();
    printBoth("ANGLE MODE=" + String(angleMode));
} else if (cmd == "GAIN" && n >= 2) {
    motorGain = constrain(p[1].toFloat(), 1.0f, 80.0f);
    printBoth("GAIN=" + String(motorGain, 1));
} else if (cmd == "PID" && n >= 4) {
    Kp = p[1].toFloat();
    Ki = p[2].toFloat();
    Kd = p[3].toFloat();
    pidIntegral = 0;
    printBoth("PID " + String(Kp, 2) + "," + String(Ki, 2) + "," + String(Kd, 2));
} else if (cmd == "OFF" && n >= 2) {
    balanceOffsetDeg = p[1].toFloat();
    pidIntegral = 0;
    printBoth("OFFSET=" + String(balanceOffsetDeg, 2));
} else if (cmd == "ZERO") {
    armed = false;
    motorTestMode = false;
    stopMotors();

```

```

    printBoth("ZEROING");
    zeroBalanceAngle();
    printBoth("ZERO DONE offset=" + String(balanceOffsetDeg, 2));
} else if (cmd == "JOY" && n >= 3) {
    forwardCmd = constrain(p[1].toFloat(), -1.0f, 1.0f);
    turnCmd = constrain(p[2].toFloat(), -1.0f, 1.0f);
    printBoth("JOY " + String(forwardCmd, 2) + " " + String(turnCmd, 2));
} else if (cmd == "RAW") {
    printBoth("RAW ax=" + String(lastAX, 0) + " ay=" + String(lastAY, 0) + " az=" + String(lastAZ,
0) + " gx=" + String(lastGX, 2) + " gy=" + String(lastGY, 2) + " gz=" + String(lastGZ, 2));
} else if (cmd == "STATUS" || cmd == "?") {
    String s = "STATUS angle=" + String(angleDeg, 2) + " offset=" + String(balanceOffsetDeg, 2) + "
err=" + String(angleDeg - balanceOffsetDeg, 2);
    s += " gyro=" + String(gyroRateDps, 2) + " mode=" + String(angleMode) + " assign=" +
String(angleSign, 0) + " gsign=" + String(gyroSign, 0);
    s += " sign=" + String(motorSign) + " invL=" + String(dirInvertL ? 1 : 0) + " invR=" +
String(dirInvertR ? 1 : 0) + " gain=" + String(motorGain, 1);
    s += " L=" + String(targetSpsL) + " R=" + String(targetSpsR) + " armed=" + String(armed ? 1 : 0);
    printBoth(s);
} else printBoth("UNKNOWN " + line);
}

```

```

//diavasma entolis

```

```

void readSerialCommands(Stream &port, String &lineBuffer, uint32_t &lastByteMs) {
    while (port.available()) {
        int incoming = port.read();
        if (incoming < 0) break;

        char c = (char)incoming;
        lastByteMs = millis();

        if (c == '\r') continue;

        if (c == '\n' || c == ';') {
            lineBuffer.trim();
            if (lineBuffer.length() > 0) handleCommand(lineBuffer);
            lineBuffer = "";
            lastByteMs = 0;
        } else {

```

```

    if ((uint8_t)c >= 32 && (uint8_t)c <= 126) {
        if (lineBuffer.length() < 100) lineBuffer += c;
        else {
            lineBuffer = "";
            lastByteMs = 0;
        }
    }
}

if (lineBuffer.length() > 0 && lastByteMs != 0 && (millis() - lastByteMs) > 80) {
    lineBuffer.trim();
    if (lineBuffer.length() > 0) handleCommand(lineBuffer);
    lineBuffer = "";
    lastByteMs = 0;
}

//ekkinisi toy robot
void setup() {
    pinMode(STEP_PIN_L, OUTPUT);
    pinMode(DIR_PIN_L, OUTPUT);
    pinMode(STEP_PIN_R, OUTPUT);
    pinMode(DIR_PIN_R, OUTPUT);
    stopMotors();

    Serial.begin(115200);
    delay(800);

    BLE.begin("PicoBalanceBot");
    BLE.server()->addService(&uart);
    BLE.startAdvertising();
    uart.setAutoflush(20);

    Wire.setSDA(I2C_SDA_PIN);
    Wire.setSCL(I2C_SCL_PIN);
    Wire.begin();

```

```

Wire.setClock(400000);

printBoth("\nPicoBalanceBot");

if (!initMPU()) {
    printBoth("ERROR MPU6050 not found");
    while (true) {
        stopMotors();
        delay(1000);
    }
}
printBoth("Hold upright for ZERO");
zeroBalanceAngle();

    if (!add_repeating_timer_us(-STEP_TIMER_PERIOD_US, stepperTimerCallback, NULL,
&stepTimer)) {
        printBoth("ERROR timer");
        while (true) {
            stopMotors();
            delay(1000);
        }
    }

lastCommandMs = millis();
lastStatusMs = millis();
armed = false;
motorTestMode = false;
printBoth("READY. Use M 2000, ZERO, STATUS, ASIGN, SIGN, GAIN.");
}

void loop() {
    static uint32_t lastLoopUs = micros(), nextLoopUs = micros();
    readSerialCommands(Serial, usbLine, usbLastByteMs);
    readSerialCommands(uart, bleLine, bleLastByteMs);
    uint32_t nowUs = micros();

    if ((int32_t)(nowUs - nextLoopUs) < 0) return;
    nextLoopUs += CONTROL_PERIOD_US;

```

```

float dt = (nowUs - lastLoopUs) / 1000000.0f;
lastLoopUs = nowUs;

if (dt <= 0 || dt > 0.05f) dt = 1.0f / CONTROL_HZ;
updateAngle(dt);

if (millis() - lastCommandMs > 3000) {
    forwardCmd = 0;
    turnCmd = 0;
}
float maxChange = commandRampRate * dt;

//eksomalinsi kinisewn
if (filteredForwardCmd < forwardCmd) {
    filteredForwardCmd = min(filteredForwardCmd + maxChange, forwardCmd);
} else if (filteredForwardCmd > forwardCmd) {
    filteredForwardCmd = max(filteredForwardCmd - maxChange, forwardCmd);
}

if (filteredTurnCmd < turnCmd) {
    filteredTurnCmd = min(filteredTurnCmd + maxChange, turnCmd);
} else if (filteredTurnCmd > turnCmd) {
    filteredTurnCmd = max(filteredTurnCmd - maxChange, turnCmd);
}

float cmd = filteredForwardCmd;
float activeLeanDeg = (cmd >= 0.0f) ? forwardLeanDeg : backwardLeanDeg;

float targetAngle = balanceOffsetDeg - cmd * activeLeanDeg;
float error = angleDeg - targetAngle;

if (armed) {

    bool fallCandidate = fabsf(error) > FALL_ANGLE_DEG;

    if (fallCandidate) {

```

```

    if (fallStartMs == 0) {
        fallStartMs = millis();
    }
} else {
    fallStartMs = 0;
}

if (fallStartMs != 0 && millis() - fallStartMs > FALL_TIME_MS) {
    armed = false;
    fallen = true;
    fallStartMs = 0;
    pidIntegral = 0;
    forwardCmd = 0;
    turnCmd = 0;
    filteredForwardCmd = 0;
    filteredTurnCmd = 0;
    smoothLeftSps = 0;
    smoothRightSps = 0;
    stopMotors();
    printBoth("FALL STOP");
} else {
    pidIntegral += error * dt;
    pidIntegral = constrain(pidIntegral, -integralLimit, integralLimit);

    float pid = (Kp * error) + (Ki * pidIntegral) + (Kd * gyroRateDps);

    // PID balance output
    float baseSps = motorSign * motorGain * pid;

    baseSps = constrain(baseSps, -balanceMaxSps, balanceMaxSps);

    // mpros / pisw
    float driveSps = 0.0f;

    if (filteredForwardCmd > 0.05f) {
        driveSps = forwardDriveStepsPerSec * filteredForwardCmd;
    } else if (filteredForwardCmd < -0.05f) {

```

```

    driveSps = backwardDriveStepsPerSec * filteredForwardCmd;
}

// strofi
float turnSps = filteredTurnCmd * turnStepsPerSec;

//ipologismoos taxititas gia kathe moter
float desiredLeftSps = baseSps + driveSps - turnSps;
float desiredRightSps = baseSps + driveSps + turnSps;

// periorismos taxititas ana kiklo
float maxStepChange = motorSlewSpsPerSec * dt;

if (smoothLeftSps < desiredLeftSps) {
    smoothLeftSps = min(smoothLeftSps + maxStepChange, desiredLeftSps);
} else if (smoothLeftSps > desiredLeftSps) {
    smoothLeftSps = max(smoothLeftSps - maxStepChange, desiredLeftSps);
}

if (smoothRightSps < desiredRightSps) {
    smoothRightSps = min(smoothRightSps + maxStepChange, desiredRightSps);
} else if (smoothRightSps > desiredRightSps) {
    smoothRightSps = max(smoothRightSps - maxStepChange, desiredRightSps);
}

setMotorTargets(smoothLeftSps, smoothRightSps);
}

} else if (!motorTestMode) {
    stopMotors();
}

if (millis() - lastStatusMs >= 300) {
    lastStatusMs = millis();

    String s = "Angle=" + String(angleDeg, 2) + " Err=" + String(error, 2) + " Gyro=" +
String(gyroRateDps, 1) + " L=" + String(targetSpsL) + " R=" + String(targetSpsR) + " A=" +
String(armed ? 1 : 0);

    printBoth(s);
}

```

}
}