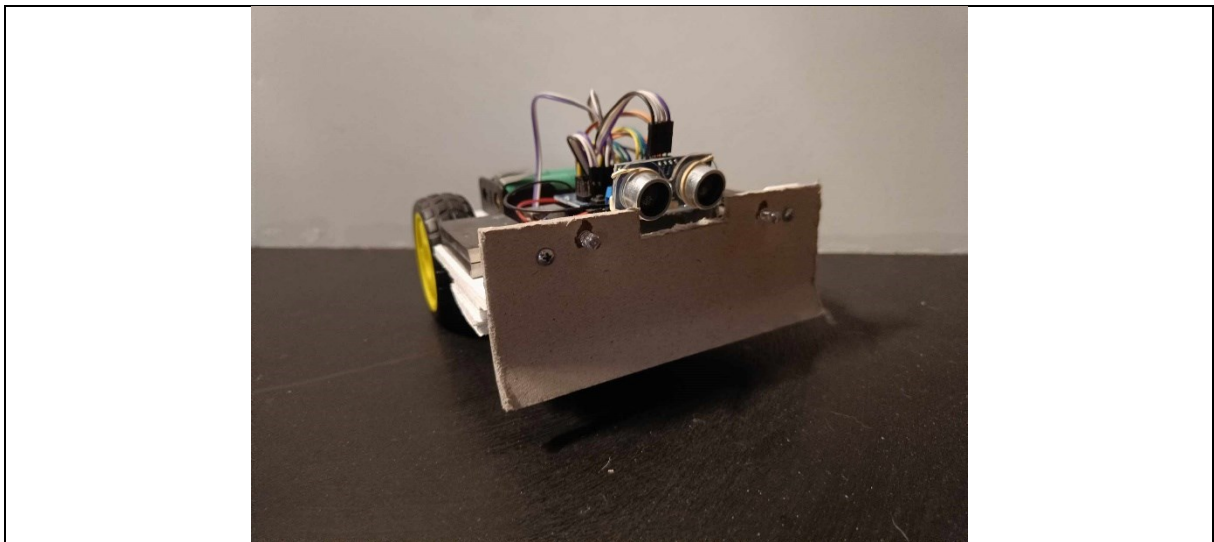


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Μελέτη και υλοποίηση Robot-Sumo για συμμετοχή
του σε αγώνες»



Του φοιτητή:
Γεώργιου Μικροβασίλη
Αρ. Μητρώου: 518086

Επιβλέπων
Άγγελος Γιακουμής
Βαθμίδα: Λέκτορας

Ημερομηνία 20-08-2024

Τίτλος Δ.Ε.: Μελέτη και υλοποίηση Robot-Sumo για συμμετοχή του σε αγώνες

Κωδικός Δ.Ε.: 24179

Όνοματεπώνυμο φοιτητή: Γεώργιος Μικροβασίλης

Όνοματεπώνυμο εισηγητή: Άγγελος Γιακουμής

Ημερομηνία ανάληψης Δ.Ε. 29-03-2024

Ημερομηνία περάτωσης Δ.Ε. 20-08-2024

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 Γεώργιου Μικροβασίλη 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»

Πρόλογος

Η ανάπτυξη και υλοποίηση ρομποτικών συστημάτων αποτελεί μία από τις πιο συναρπαστικές προκλήσεις της σύγχρονης τεχνολογίας. Η ρομποτική όχι μόνο επαναστατεί τη μηχανολογία, αλλά και ανοίγει νέους δρόμους στη μηχανική μάθηση και τον αυτοματισμό. Στο πλαίσιο της παρούσας διπλωματικής εργασίας, εξετάζεται η μελέτη και κατασκευή ενός Sumo Robot, με στόχο την συμμετοχή του σε αγώνες ρομποτικής.

Περίληψη

Σε αυτή την διπλωματική εργασία εξερευνάται η κατασκευή ενός Sumo Robot. Αρχικά γίνεται αναφορά στους αγώνες Sumo Robot και στην ιστορία τους, στους κανονισμούς και τα εξαρτήματα που χρησιμοποιήθηκαν. Στη συνέχεια επεξηγείται ο τρόπος κατασκευής και συνδεσμολογίας του ρομπότ, ενώ περιγράφεται και ο κώδικας που χρησιμοποιήθηκε στην συγκεκριμένη υλοποίηση. Τέλος αναφέρονται συμπεράσματα και βελτιώσεις.

«Design and implementation of Robot-Sumo for its participation in competitions»

«George Mikrovasilis»

Abstract

In this thesis the construction of a Sumo Robot is explored. Firstly, reference is made to Sumo Robot competitions and their history, regulations and the components used. Then the way the robot is built and connected is explained, and the code used in this implementation is described. Finally, conclusions and improvements are mentioned.

Ευχαριστίες

Πρώτα και κύρια, ευχαριστώ θερμά τον επιβλέποντα καθηγητή μου, Κύριο Άγγελο Γιακουμή, για την καθοδήγηση και τις πολύτιμες συμβουλές του καθ' όλη τη διάρκεια της έρευνας. Ιδιαίτερες ευχαριστίες απευθύνω στην οικογένειά μου, για την αδιάλειπτη στήριξη, την αγάπη και την κατανόηση που μου έδειξαν καθ' όλη τη διάρκεια αυτής της απαιτητικής περιόδου. Η υπομονή και η ενθάρρυνσή τους ήταν το μεγαλύτερο στήριγμά μου.

Περιεχόμενα

Πρόλογος.....	iv
Περίληψη.....	v
Abstract	vi
Ευχαριστίες	vii
Περιεχόμενα	viii
Κατάλογος Πινάκων.....	xi
Συντομογραφίες.....	xi
Κεφάλαιο 1ο: Εισαγωγή.....	1
1.1 Στόχοι.....	1
1.2 Δομή της εργασίας	1
Κεφάλαιο 2ο: Ιστορία και εξέλιξη.....	2
2.1 Αρχική εμφάνιση των sumo robot.....	2
2.2 Τεχνολογική εξέλιξη	3
2.3 Κορυφαίες προσωπικότητες.....	3
2.4 Επίλογος.....	3
Κεφάλαιο 3ο: Αγωνιστικές κατηγορίες και κανονισμοί.....	5
3.1 Προδιαγραφές Sumo Ring (Dohyo).....	5
3.2 Προδιαγραφές Sumo Robot.....	6
3.3 Μέθοδοι χειρισμού	6
3.4 Ταυτοποίηση	7
3.5 Περιορισμοί.....	8
3.6 Κριτήρια νίκης.....	8
3.7 Οδηγίες έναρξης, διακοπής, και τερματισμού αγώνων	8
3.8 Χρονική διάρκεια αγώνα.....	9
3.9 Απόκτηση πόντων (Yuhkoh).....	9
3.10 Παραβάσεις	10
3.11 Κυρώσεις.....	10
3.12 Τραυματισμοί και ατυχήματα κατά την διάρκεια του αγώνα	10
3.13 Ενστάσεις	11
3.14 Επίλογος.....	11
Κεφάλαιο 4ο: Arduino.....	12
4.1 Εισαγωγή στο Arduino.....	12

4.2	Ιστορική αναδρομή του Arduino.....	13
4.3	Μοντέλα πλακετών Arduino	13
4.4	Arduino Shields.....	14
4.5	Πλεονεκτήματα και μειονεκτήματα Arduino	16
4.6	Εφαρμογές Arduino	17
4.6.1	Ρομποτικές εφαρμογές	17
4.6.2	Εκπαιδευτικές εφαρμογές.....	18
4.6.3	IoT εφαρμογές	18
4.7	Περιβάλλον προγραμματισμού του Arduino.....	18
4.8	Επίλογος	20
Κεφάλαιο 5ο: Arduino Uno.....		21
5.1	Χαρακτηριστικά Arduino Uno	22
5.2	Είσοδοι / Έξοδοι.....	22
5.3	Διπλής λειτουργίας ακροδέκτες	22
5.4	Τροφοδοσία.....	23
5.5	Μνήμη.....	23
5.6	Επίλογος.....	24
Κεφάλαιο 6ο: Επιλογή και ανάλυση υλικών.....		25
6.1	Βραχυκυκλωτήρες.....	25
6.2	Ταινία διπλής όψης / κόλλα.....	25
6.3	Sumo Ring (Dohyo)	26
6.4	Πλαίσιο.....	26
6.5	Ελαστικά	27
6.6	Κινητήρες.....	28
6.7	Αισθητήρες.....	28
6.7.1	TCRT5000.....	28
6.7.2	HC-SR04	30
6.8	Motor Driver L298N.....	31
6.9	Μικροελεγκτής.....	32
6.10	Shield.....	33
6.11	Πηγή ενέργειας.....	34
6.12	Επίλογος.....	35
Κεφάλαιο 7ο: Κατασκευή		36
7.1	Περιγραφή λειτουργίας.....	36

7.2	Εξέλιξη της κατασκευής του Sumo Robot.....	36
7.3	Διάγραμμα καλωδιώσεων	42
7.4	Κατασκευή του Sumo Ring (Dohyo)	43
7.5	Τελικό αποτέλεσμα	43
7.6	Πίνακας υλικών	44
7.7	Επίλογος.....	45
Κεφάλαιο 8ο: Περιγραφή κώδικα		46
8.1	Κλήση βιβλιοθηκών	46
8.2	Δήλωση μεταβλητών και δημιουργία αντικειμένων	46
8.3	Η συνάρτηση setup()	47
8.4	Η συνάρτηση loop()	47
8.5	Υπόλοιπες συναρτήσεις.....	48
8.6	Πίνακας εντολών	49
8.7	Πίνακας λέξεων κλειδιών δήλωσης μεταβλητών	50
8.8	Διάγραμμα ροής	51
8.9	Επίλογος.....	52
Κεφάλαιο 9ο: Συμπεράσματα και προτάσεις βελτίωσης.....		53
9.1	Συμπεράσματα.....	53
9.2	Βελτιώσεις.....	53
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		54
ΠΑΡΑΡΤΗΜΑ Α : ΚΩΔΙΚΑΣ		55

Κατάλογος Πινάκων

Πίνακας 6.1: Πίνακας αληθείας κινητήρα A.....	31
Πίνακας 6.2: Πίνακας αληθείας κινητήρα B.....	32
Πίνακας 7.1: Πίνακας υλικών και κόστους.....	44
Πίνακας 8.1: Πίνακας εντολών	49
Πίνακας 8.2: Πίνακας λέξεων κλειδιών δήλωσης μεταβλητών	50

Κατάλογος Εικόνων

Εικόνα 2.1: Αφίσα πρώτου τουρνουά Sumo Robot (Ιαπωνία 1989).....	2
Εικόνα 2.2: Παλιός μικροελεγκτής (8051).....	3
Εικόνα 2.3: Σύγχρονος μικροελεγκτής (ESP32)	3
Εικόνα 3.1: Sumo Ring (Dohyo).....	5
Εικόνα 3.2: Προδιαγραφές Sumo Ring (Dohyo)	6
Εικόνα 3.3: Προδιαγραφές Sumo Robot	6
Εικόνα 3.4: Ταυτοποίηση Sumo Robot.....	7
Εικόνα 4.1: Arduino Logo.....	12
Εικόνα 4.2: Arduino Nano	13
Εικόνα 4.3: Arduino Uno	14
Εικόνα 4.4: Arduino Mega.....	14
Εικόνα 4.5: Χρήση πολλαπλών Shields	15
Εικόνα 4.6: Arduino Wifi Shield.....	15
Εικόνα 4.7: Arduino Sensor Shield.....	16
Εικόνα 4.8: Arduino Motor Shield.....	16
Εικόνα 4.9: Line Following Robot.....	17
Εικόνα 4.10: Εκπαιδευτικές εφαρμογές	18
Εικόνα 4.11: Arduino IDE	19
Εικόνα 4.12: Σειριακή οθόνη Arduino IDE	20
Εικόνα 5.1: Arduino Uno (Επεξήγηση)	21
Εικόνα 5.2: Μνήμη.....	24
Εικόνα 6.1: Βραχυκυκλωτήρες	25
Εικόνα 6.2: Ταινία διπλής όψης / κόλλα.....	25
Εικόνα 6.3: Μελαμίνη Dohyo	26
Εικόνα 6.4: Χαρτόνι μακέτας πλαισίου	26
Εικόνα 6.5: Χαρτόνι προφυλακτήρα.....	27
Εικόνα 6.6: Ρόδα κατασκευής από καουτσούκ.....	27
Εικόνα 6.7: DC Κινητήρας 3-12V	28
Εικόνα 6.8: TCRT5000	29
Εικόνα 6.9: TCRT5000 Circuit Diagram	29
Εικόνα 6.10: HC-SR04.....	30
Εικόνα 6.11: Τρόπος λειτουργίας HC-SR04.....	30
Εικόνα 6.12: L298N (PinOut).....	31
Εικόνα 6.13: Arduino Uno (PinOut)	32
Εικόνα 6.14: Sensor Shield V5 (PinOut)	33

Εικόνα 6.15: Shield Voltage Output Diagram.....	33
Εικόνα 6.16: 18650 Μπαταρία.....	34
Εικόνα 6.17: 18650 Μπαταριοθήκη.....	34
Εικόνα 6.18: Διακόπτης.....	34
Εικόνα 7.1: Κόψιμο μακετόχαρτου πλαισίου.....	36
Εικόνα 7.2: Τροποποίηση πλαισίου.....	37
Εικόνα 7.3: Σχεδιασμός εξαρτημάτων (πάνω μέρος).....	37
Εικόνα 7.4: Σχεδιασμός εξαρτημάτων (κάτω μέρος).....	38
Εικόνα 7.5: Βάση μπίλιας / IR.....	38
Εικόνα 7.6: Ολοκλήρωση κάτω μέρος πλαισίου.....	39
Εικόνα 7.7: Ολοκλήρωση άνω μέρος πλαισίου.....	39
Εικόνα 7.8: Προσθήκη βάρους / προφυλακτήρα στο άνω μέρος της κατασκευής.....	40
Εικόνα 7.9: Προσθήκη βάρους στο κάτω μέρος της κατασκευής.....	40
Εικόνα 7.10: Βάρος κατασκευής.....	41
Εικόνα 7.11: Εχθρός.....	41
Εικόνα 7.12: Διάγραμμα καλωδιώσεων.....	42
Εικόνα 7.13: Sumo Ring (Dohyo).....	43
Εικόνα 7.14: Τελικό αποτέλεσμα.....	43
Εικόνα 7.15: Αναμέτρηση.....	44
Εικόνα 8.1: Δήλωση μεταβλητών και δημιουργία αντικειμένων.....	46
Εικόνα 8.2: Setup().....	47
Εικόνα 8.3: Loop().....	47
Εικόνα 8.4: Συναρτήσεις κίνησης.....	48
Εικόνα 8.5: Συναρτήσεις ISR.....	49
Εικόνα 8.6: Συναρτηση Ultrasonic.....	49
Εικόνα 8.7: Διάγραμμα ροής.....	51

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΙΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
RUR	Rossum's Universal Robots
FCC	Federal Communications Commission
WPA2	Wi-Fi Protected Access 2
PSK	Pre-shared Key
IP	Internet Protocol
IR	Infrared
PWM	Pulse Width Modulation
IDE	Integrated Development Environment
DIY	Do It Yourself
IoT	Internet of Things
EEPROM	Electrically Erasable Programmable Read-Only Memory
SRAM	Static Random Access Memory
gf-cm	gram-force centimeter
I/O	Input / Output
I2C	Inter-Integrated Circuit
UART	Universal Asynchronous Receiver/Transmitter
ISR	Interrupt Service Routine

Κεφάλαιο 1^ο: Εισαγωγή

Η ρομποτική είναι ένας ραγδαία αναπτυσσόμενος τομέας της τεχνολογίας με πληθώρα εφαρμογών σε διάφορους κλάδους, όπως η βιομηχανία, η ιατρική και η εκπαίδευση. Μία από τις ιδιαίτερα δημοφιλείς εφαρμογές της ρομποτικής είναι οι διαγωνισμοί ρομπότ, όπως τα sumo robots, όπου τα ρομπότ αναμετρώνται σε μια αρένα με στόχο να εντοπίσουν και να εκτοπίσουν τον αντίπαλο εκτός της αρένας. Η παρούσα διπλωματική εργασία επικεντρώνεται στον σχεδιασμό, την κατασκευή και τον προγραμματισμό ενός sumo robot. Το ρομπότ αυτό θα χρησιμοποιεί διάφορους αισθητήρες και κινητήρες, προκειμένου να εντοπίζει και να αλληλεπιδρά με τον αντίπαλο ρομπότ, ενώ ταυτόχρονα θα πρέπει να αποφεύγει την έξοδο από την αρένα.

Στόχοι

Ο πρωταρχικός ρόλος αυτής της εργασίας είναι να εξερευνήσει και να επιδείξει τη διαδικασία σχεδιασμού, κατασκευής και προγραμματισμού ενός λειτουργικού sumo robot με στόχο τη συμμετοχή σε διαγωνισμούς. Μέσω αυτής της διερεύνησης, η εργασία επιδιώκει να αποτελέσει έναν ολοκληρωμένο οδηγό που συνδυάζει θεωρητική γνώση και πρακτική εφαρμογή. Καλύπτει όλο το φάσμα της ανάπτυξης ενός sumo robot, από την αρχική σύλληψη και σχεδίαση, έως την κατασκευή, τον προγραμματισμό, τις δοκιμές και την τελική αξιολόγηση, προσφέροντας πολύτιμες πληροφορίες και οδηγίες για κάθε στάδιο της διαδικασίας.

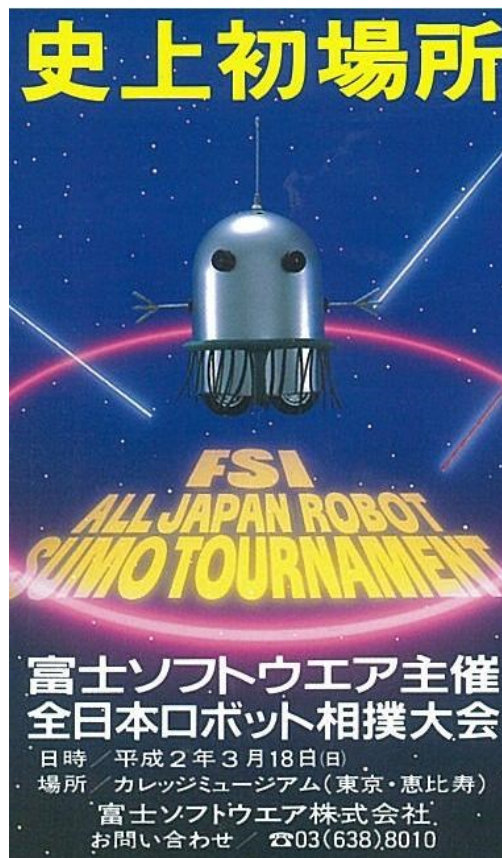
Δομή της εργασίας

Η εργασία δομείται σε εννέα κεφάλαια, ξεκινώντας με μια εισαγωγή στους στόχους και τη δομή της, και συνεχίζει με την ιστορία και την εξέλιξη των Sumo Robots, περιγράφοντας τις αγωνιστικές κατηγορίες και κανονισμούς. Ακολουθούν λεπτομερείς πληροφορίες για το Arduino συνολικά και το Arduino Uno, που αποτελούν τη βάση για την κατασκευή του ρομπότ. Αναλύεται η επιλογή και ανάλυση των υλικών, ενώ παρουσιάζεται η διαδικασία κατασκευής και το αντίστοιχο διάγραμμα καλωδιώσεων. Εν συνεχεία, περιγράφεται ο κώδικας του ρομπότ και, τέλος, η εργασία καταλήγει με τα συμπεράσματα και τις προτάσεις βελτίωσης.

Κεφάλαιο 2^ο: Ιστορία και εξέλιξη

2.1 Αρχική εμφάνιση των Sumo Robot

Οι διαγωνισμοί sumo robot έχουν καθιερωθεί στην Ιαπωνία, με τον πρώτο διαγωνισμό να διεξάγεται το 1989 και να συμμετέχουν 33 ρομπότ. Από τότε, η φήμη τους έχει αυξηθεί σημαντικά. Παράδειγμα αποτελεί το έτος 2001, όπου ο αριθμός των συμμετεχόντων ρομπότ ξεπέρασε τα 4.000, σύμφωνα με τον Pete Miles στο βιβλίο του, το οποίο αποτελεί κοινό αναφορικό σημείο για την κατασκευή ρομπότ sumo. Στον πρώτο διαγωνισμό ρομπότ sumo που παρουσιάστηκε στις Ηνωμένες Πολιτείες στις αρχές της δεκαετίας του 1990, ο Bill Harrison αναδείχθηκε ένας από τους μεγαλύτερους υποστηρικτές. Αυτός και ο συνεργάτης του δημιούργησαν μια πιο προσιτή κατηγορία sumo robot, το "μικρότερο από το αναμενόμενο μοντέλο" με βάρος έως και 500 γραμμάρια, προσελκύοντας έτσι περισσότερους σχεδιαστές και διαγωνιζόμενους λόγω του χαμηλότερου κόστους σε σχέση με τα πιο μεγάλα μοντέλα των 3 κιλών που χρησιμοποιούνταν στην Ιαπωνία [1].



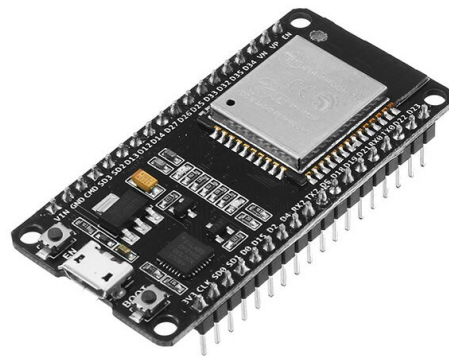
Εικόνα 2.1: Αφίσα πρώτου τουρνουά Sumo Robot (Ιαπωνία 1989)

2.2 Τεχνολογική εξέλιξη

Οι μικροελεγκτές PIC, 8051 και Basic Stamp υπήρξαν θεμελιώδεις για την ανάπτυξη των πρώτων Sumo robots, παρέχοντας ευελιξία και αξιοπιστία σε μια περίοδο που η τεχνολογία των ενσωματωμένων συστημάτων ήταν ακόμη σε πρώιμο στάδιο. Οι PIC microcontrollers της Microchip διακρίθηκαν για την ευκολία προγραμματισμού και την ποικιλία των διαθέσιμων μοντέλων τους, οι 8051 microcontrollers της Intel για την εκτεταμένη χρήση και τη σταθερότητά τους, και οι Basic Stamp της Parallax για την απλότητα της γλώσσας προγραμματισμού BASIC. Ωστόσο, με την εξέλιξη της τεχνολογίας, οι περιορισμένες υπολογιστικές δυνατότητες, η έλλειψη υποστήριξης σύγχρονων περιφερειακών και η ανάγκη για πιο φιλικές προς τον χρήστη πλατφόρμες οδήγησαν στην αντικατάστασή τους από σύγχρονους μικροελεγκτές όπως το Arduino, το Raspberry Pi και το ESP32. Οι νέοι αυτοί μικροελεγκτές προσφέρουν αυξημένη υπολογιστική ισχύ, ενσωματωμένες δικτυακές δυνατότητες και προηγμένα εργαλεία ανάπτυξης, καθιστώντας τους πιο κατάλληλους για τις απαιτήσεις των σύγχρονων ρομποτικών εφαρμογών.



Εικόνα 2.2: Παλιός μικροελεγκτής (8051)



Εικόνα 2.3: Σύγχρονος μικροελεγκτής (ESP32)

2.3 Κορυφαίες προσωπικότητες

Το άθλημα του sumo robot δημιουργήθηκε από τον Hiroshi Nozawa το 1989. Ο Nozawa, ένας Ιάπωνας μηχανικός και εφευρέτης, είχε την ιδέα να δημιουργήσει έναν διαγωνισμό ρομπότ που θα αντικατοπτρίζει τους παραδοσιακούς κανόνες του ιαπωνικού sumo. Το πρώτο πρωτάθλημα sumo robot διοργανώθηκε από την εταιρεία Fujisoft, με την υποστήριξη και την καθοδήγηση του Nozawa, με στόχο να προωθηθεί η τεχνολογία και η μηχανική μέσω μιας διασκεδαστικής και προκλητικής δραστηριότητας [2].

2.4 Επίλογος

Οι διαγωνισμοί sumo robot ξεκίνησαν στην Ιαπωνία το 1989, με πρωτοβουλία του Hiroshi Nozawa και την υποστήριξη της Fujisoft Inc. Από τον πρώτο διαγωνισμό με 33 ρομπότ, η φήμη τους αυξήθηκε εκθετικά, με πάνω από 4.000 συμμετοχές το 2001. Στις αρχές της δεκαετίας του 1990, ο Bill Harrison εισήγαγε το άθλημα στις Ηνωμένες Πολιτείες, καθιερώνοντας μια πιο προσιτή κατηγορία ρομπότ βάρους έως 500 γραμμαρίων, προσελκύοντας περισσότερους σχεδιαστές λόγω του χαμηλότερου

κόστους. Αρχικά, οι μικροελεγκτές όπως οι PIC, 8051 και Basic Stamp αποτέλεσαν τη βάση για την ανάπτυξη των sumo robots. Με την εξέλιξη της τεχνολογίας, αντικαταστάθηκαν από σύγχρονες πλατφόρμες, οι οποίες προσφέρουν αυξημένη υπολογιστική ισχύ και προηγμένα εργαλεία ανάπτυξης. Ο Hiroshi Nozawa έθεσε τα θεμέλια για την προώθηση της ρομποτικής μέσω των διαγωνισμών sumo robot, δημιουργώντας μια παγκόσμια κοινότητα που συνεχίζει να εμπνέει και να εκπαιδεύει νέες γενιές μηχανικών.

Κεφάλαιο 3^ο: Αγωνιστικές κατηγορίες και κανονισμοί

Δύο robot ανταγωνίζονται σε έναν αγώνα σώμα με σώμα, παρόμοιο με τους παραδοσιακούς αγώνες sumo. Μια αναμέτρηση διεξάγεται ανάμεσα σε δύο ομάδες, κάθε μια με έναν ή περισσότερους διαγωνιζόμενους. Μόνο ένα μέλος της κάθε ομάδας μπορεί να πλησιάσει το ρινγκ, τα υπόλοιπα μέλη της ομάδας πρέπει να παρακολουθούν από το κοινό. Τα robot με τα οποία αγωνίζεται η κάθε ομάδα θα πρέπει να είναι δικής τους κατασκευής και να ανταποκρίνονται στις προδιαγραφές της κατηγορίας τους. Απαγορεύεται η χρήση όπλων και η ανατροπή του αντιπάλου. Ο στόχος είναι να σπρώξουν το ένα το άλλο έξω από τον αγωνιστικό χώρο μέσω άμεσης ώθησης. Υπάρχουν διαφορετικές κατηγορίες βάρους και ελέγχου. Τα αυτόνομα robot διαγωνίζονται μόνο με αυτόνομα, και τα τηλεχειριζόμενα μόνο με τηλεχειριζόμενα, χωρίς να αναμειγνύονται οι κατηγορίες. Ο αγώνας ξεκινά με εντολή του διαιτητή και συνεχίζεται μέχρι ένας διαγωνιζόμενος να κερδίσει δύο Yuhkoh (γιοκό) πόντους. Ακολουθούν οι κανονισμοί και προδιαγραφές των κατασκευών [3].

3.1 Προδιαγραφές Sumo Ring (Dohyo)

Το ρινγκ πρέπει να είναι κυκλικό και να έχει τις κατάλληλες διαστάσεις για την δεδομένη κατηγορία μεγέθους. Οι γραμμές εκκίνησης (shikiri lines) αποτελούνται από δύο παράλληλες καφέ γραμμές (ή αντίστοιχο χρώμα που απορροφά το υπέρυθρο φως) στο κέντρο του ρινγκ, με το κατάλληλο πλάτος και απόσταση για την συγκεκριμένη κατηγορία. Η απόσταση μεταξύ των γραμμών μετράται από τα εξωτερικά τους άκρα. Η οριογραμμή είναι σημειωμένη ως λευκός κυκλικός δακτύλιος με πλάτος κατάλληλο για την δεδομένη κατηγορία, στην εξωτερική άκρη της επιφάνειας παιχνιδιού. Η περιοχή εντός του κυκλικού δακτυλίου πρέπει να είναι χρώματος μαύρο [3].



Εικόνα 3.1: Sumo Ring (Dohyo)

Για την κάθε κατηγορία robot υπάρχει η αντίστοιχη κατηγορία μεγέθους ring, οι οποίες παραθέτονται στον Εικόνα 3.2 .

Dohyo Construction & Painting				Shikiri Lines			White Border
Class	Height	Diameter	Material	Width	Length	Separation	Width
Mega/Humanoid	5.00 cm	154.0 cm	Polyurethane coated steel plate	2 cm	20 cm	20 cm	5 cm
Mini/Kit	2.50 cm	77.0 cm	Laminate/Melamine coated wood	1 cm	10 cm	10 cm	2.5 cm
Micro	1.25 cm	38.5 cm	Melamine composite	0.5 cm	5 cm	5 cm	1.25 cm
Nano	0.625 cm	19.25 cm	Melamine composite	0.25 cm	2.5 cm	2.5 cm	0.625 cm

Εικόνα 3.2: Προδιαγραφές Sumo Ring (Dohyo)

3.2 Προδιαγραφές Sumo Robot

Για να μπορέσει να συμμετέχει ένα robot σε αγώνες θα πρέπει να χωράει μέσα σε ένα τετράγωνο σωλήνα με τις κατάλληλες διαστάσεις για την συγκεκριμένη κατηγορία. Ένα robot μπορεί να αλλάξει μέγεθος μετά την έναρξη του αγώνα, αλλά τα κομμάτια του δεν θα πρέπει να χωριστούν από το σώμα. Το συνολικό βάρος ενός ρομπότ κατά την έναρξη ενός αγώνα πρέπει να είναι κάτω από το καθορισμένο βάρος για τη συγκεκριμένη κατηγορία. Στην Εικόνα 3.3 αναγράφονται οι διαστάσεις και η μάζα της κάθε κατηγορίας ρομπότ.

Class	Height	Width	Length	Weight
Mega Sumo - Auton	unlimited	20.0 cm	20.0 cm	3,000 g
Mega Sumo - R/C	unlimited	20.0 cm	20.0 cm	3,000 g
Mega Sumo - Network	unlimited	20.0 cm	20.0 cm	3,000 g
Humanoid - R/C	50 cm	20.0 cm	20.0 cm	4,000 g
Kit Sumo	unlimited	15.0 cm	15.0 cm	1,000 g
Mini Sumo	unlimited	10.0 cm	10.0 cm	500 g
Micro Sumo	5 cm	5.0 cm	5.0 cm	100 g
Nano Sumo	2.5 cm	2.5 cm	2.5 cm	25 g

Εικόνα 3.3: Προδιαγραφές Sumo Robot

3.3 Μέθοδοι Χειρισμού

Στην αυτόνομη κατηγορία εντάσσονται τα robot των κατηγοριών Nano (25g), Micro (100g), Mini (500g), kit (1kg) και Mega (3kg). Οποιαδήποτε μέθοδος ελέγχου μπορεί να χρησιμοποιηθεί, αρκεί να περιέχεται πλήρως μέσα στο robot και να μην λαμβάνει εξωτερικά σήματα ή οδηγίες (ανθρώπινα, μηχανικά ή άλλα). Η αυτόνομη λειτουργία του robot πρέπει να ξεκινά αυτόματα τουλάχιστον πέντε δευτερόλεπτα μετά την έναρξη λειτουργίας από τον χρήστη. Τα ρομπότ που ξεκινούν πριν από τα πέντε δευτερόλεπτα χάνουν πόντο.

Στην τηλεχειριζόμενη κατηγορία εντάσσονται τα ρομπότ των κατηγοριών R/C και Humanoid τα οποία μπορούν να ελεγχθούν εξ αποστάσεως σύμφωνα με τους κανονισμούς της FCC (Federal Communications Commission) και να ρυθμίζονται από τους υπεύθυνους του τουρνουά. Οι ελεγκτές 75Mhz απαγορεύονται ρητά. Όλα τα τηλεχειριστήρια πρέπει να είναι ζεύγη με ψηφιακή σύζευξη. Τα τηλεχειριζόμενα ρομπότ ξεκινούν με σήμα από τον υπεύθυνο του τουρνουά. Δεν επιτρέπονται πρόσθετοι αυτόνομοι έλεγχοι στο ρομπότ. Όλες οι κινήσεις του ρομπότ πρέπει να οφείλονται άμεσα στον τηλεχειρισμό του χειριστή. Στην τηλεχειριζόμενη κατηγορία εντάσσονται επίσης τα robot που ελέγχονται μέσω WIFI από τον υπολογιστή του χειριστή, μέσω ενός ενδιάμεσου δρομολογητή. Οι απευθείας συνδέσεις από και προς το ρομπότ από τον ελεγκτή/υπολογιστή του χειριστή εμποδίζονται από τη διαμόρφωση του δικτύου και δεν επιτρέπονται. Τόσο ο χειριστής όσο και το robot θα έχουν ανοικτές συνδέσεις στο Διαδίκτυο. Η εκδήλωση θα παρέχει δύο δρομολογητές με ασφάλεια WPA2 PSK, έναν για τον ελεγκτή και έναν για το robot, με διευθύνσεις IP με δημόσια διεύθυνση. Οι συμμετέχοντες θα πρέπει να εξασκηθούν στον έλεγχο μέσω πολλαπλών δρομολογητών τόσο πριν όσο και κατά την άφιξη στην εκδήλωση. Δεν επιτρέπονται επαυξημένοι αυτόνομοι έλεγχοι στο ρομπότ. Όλες οι κινήσεις του ρομπότ πρέπει να οφείλονται άμεσα στον τηλεχειρισμό του χειριστή.

3.4 Ταυτοποίηση

Το κάθε robot θα πρέπει να έχει ένα όνομα ή αριθμό για λόγους ταυτοποίησης. Το όνομα ή αριθμός πρέπει να βρίσκεται σε εμφανές σημείο για να αναγνωρίζεται από τους θεατές και τους υπεύθυνους.



Εικόνα 3.4: Ταυτοποίηση Sumo Robot

3.5 Περιορισμοί

Δεν επιτρέπονται τα παρακάτω:

- Συσκευές παρεμβολής, όπως λυχνίες IR που προορίζονται για τον κορεσμό των αισθητήρων IR των αντιπάλων.
- Εξαρτήματα που θα μπορούσαν να σπάσουν ή να προκαλέσουν ζημιά στον δακτύλιο. Μην χρησιμοποιείτε εξαρτήματα που προορίζονται να προκαλέσουν ζημιά στο αντίπαλο ρομπότ ή στον χειριστή του. Τα κανονικά σπρωξίματα και κτυπήματα δεν θεωρούνται πρόθεση για ζημιά.
- Συσκευές που μπορούν να αποθηκεύσουν υγρό, σκόνη, αέριο ή άλλες ουσίες για ρίψη στον αντίπαλο.
- Φλεγόμενες συσκευές.
- Συσκευές που ρίχνουν αντικείμενα στον αντίπαλο δεν επιτρέπονται.
- Κολλωδείς ουσίες για τη βελτίωση της πρόσφυσης. Τα ελαστικά και άλλα εξαρτήματα του ρομπότ που έρχονται σε επαφή με τον δακτύλιο δεν πρέπει να μπορούν να σηκώσουν και να κρατήσουν μια τυπική κάρτα ευρετηρίου 3" x 5" για περισσότερο από δύο δευτερόλεπτα.
- Συσκευές για την αύξηση της δύναμης προς τα κάτω, όπως αντλία κενού ή μαγνήτες, επιτρέπονται μόνο στην κατηγορία των 3 κιλών. Δεν επιτρέπονται σε όλες τις άλλες κατηγορίες.
- Όλες οι άκρες, συμπεριλαμβανομένης της μπροστινής λεπίδας, δεν πρέπει να είναι τόσο αιχμηρές ώστε να γρατζουνίζουν ή να προκαλούν ζημιά στο ρινγκ, σε άλλα ρομπότ ή στους παίκτες. Γενικά, ακμές με ακτίνα μεγαλύτερη από 0,005", όπως αυτές που θα προκύψουν από μια αμβλεία μεταλλική λωρίδα πάχους 0,010", είναι αποδεκτές. Οι κριτές ή οι υπεύθυνοι του διαγωνισμού μπορεί να απαιτήσουν να καλυφθούν οι ακμές που θεωρούνται πολύ αιχμηρές με ταινία.

3.6 Κριτήρια νίκης

Ένας αγώνας αποτελείται από 3 γύρους, με συνολική διάρκεια 3 λεπτά, εκτός αν επεκταθεί από τους κριτές. Η ομάδα που θα κερδίσει δύο γύρους ή θα λάβει δύο πόντους "Yuhkoh" πρώτη, εντός του χρονικού ορίου, κερδίζει τον αγώνα. Η ομάδα λαμβάνει έναν πόντο "Yuhkoh" όταν κερδίζει έναν γύρο. Αν λήξει το χρονικό όριο και μία ομάδα έχει έναν πόντο "Yuhkoh", αυτή η ομάδα κερδίζει.

Αν κανείς δεν κερδίσει εντός του χρονικού ορίου, μπορεί να γίνει επέκταση του αγώνα, κατά την οποία η ομάδα που θα λάβει τον πρώτο πόντο "Yuhkoh" θα κερδίσει. Εναλλακτικά, ο νικητής μπορεί να αποφασιστεί από τους κριτές, με κλήρο ή με επαναληπτικό αγώνα. Ένας πόντος "Yuhkoh" θα δοθεί στον νικητή όταν η απόφαση των κριτών ή ο κλήρος χρησιμοποιηθεί.

3.7 Οδηγίες έναρξης, διακοπής και τερματισμού αγώνων

Με τις οδηγίες του κριτή κατά την εκκίνηση, οι δύο ομάδες υποκλίνονται η μία στην άλλη στον εξωτερικό δακτύλιο, πλησιάζουν στον δακτύλιο και τοποθετούν ένα ρομπότ στο δικό τους μισό του δακτυλίου πάνω ή πίσω από τη γραμμή Shikiri. (Ένα robot ή μέρος ενός robot δεν μπορεί να τοποθετηθεί πέρα από την μπροστινή άκρη της γραμμής Shikiri προς τον αντίπαλο. Σημειώστε ότι δεν απαιτείται ένα ρομπότ να τοποθετηθεί ακριβώς πίσω από τη γραμμή Shikiri μπορεί να είναι μετατοπισμένο στο πλάι, αρκεί να βρίσκεται πίσω από μια νοητή γραμμή κολλητή με τη γραμμή Shikiri). Όταν ο κριτής ανακοινώσει την έναρξη του γύρου, οι ομάδες εκκινούν τα ρομπότ τους και μετά από μια παύση πέντε δευτερολέπτων τα ρομπότ μπορούν να αρχίσουν να λειτουργούν. Κατά τη

διάρκεια αυτών των πέντε δευτερολέπτων, οι παίκτες πρέπει να απομακρυνθούν από την περιοχή του δακτυλίου.

Ο αγώνας σταματά και συνεχίζεται όταν το ανακοινώσει ο κριτής. Ο αγώνας τελειώνει όταν το ανακοινώσει ο επικεφαλής κριτής. Οι δύο ομάδες παίρνουν τα ρομπότ τους από την περιοχή του δακτυλίου και υποκλίνονται.

3.8 Χρονική διάρκεια αγώνα

Ένας αγώνας θα διεξαχθεί για συνολικά 3 λεπτά, ξεκινώντας και τελειώνοντας με εντολή του κριτή. Το ρολόι αρχίζει να μετράει πέντε δευτερόλεπτα μετά την ανακοίνωση της έναρξης. Ένας παρατεταμένος αγώνας, αν ζητηθεί από τον κριτή, θα διαρκεί το πολύ 3 λεπτά.

Τα ακόλουθα δεν συμπεριλαμβάνονται στο χρόνο του αγώνα:

- Ο χρόνος που περνάει μετά την ανακοίνωση του Yuhkoh από τον διαιτητή και πριν η μάχη συνεχιστεί. Η τυπική καθυστέρηση πριν η μάχη συνεχιστεί είναι 30 δευτερόλεπτα.
- Ο χρόνος που παρήλθε αφού ο κριτής ανακοινώσει τη διακοπή του αγώνα και πριν από τη συνέχιση του αγώνα.

3.9 Απόκτηση πόντων (Yuhkoh)

- Μια ομάδα ωθεί νόμιμα το σώμα του αντίπαλου ρομπότ να αγγίξει το χώρο έξω από το δακτύλιο, που περιλαμβάνει και την πλευρά του δακτυλίου.
- Όταν το robot της αντίπαλης ομάδας αγγίξει μόνο του τον εξωτερικό χώρο του δακτυλίου κατά την ανακοίνωση του τέλους του αγώνα.
- Στην περίπτωση του ανθρωποειδούς sumo, οποιοδήποτε μέρος του αντίπαλου robot εκτός από τις πατούσες των ποδιών του (χέρια, γόνατα, πλάτη, στήθος κ.λπ.) αγγίζει το dohyo ή όταν σπρώχνεται ή ρίχνεται εκτός του ρινγκ.

Εξαιρέσεις:

- Όταν ένα τροχήλατο ρομπότ έχει αναποδογυρίσει στο ρινγκ ή σε παρόμοιες συνθήκες, το Yuhkoh δεν μετράται και ο αγώνας συνεχίζεται.

Όταν ζητείται η απόφαση των κριτών για την ανάδειξη του νικητή, θα λαμβάνονται υπόψη τα ακόλουθα σημεία:

- Τεχνικές αρετές στην κίνηση και τη λειτουργία του ρομπότ
- Πόντοι ποινής κατά τη διάρκεια του αγώνα
- Συμπεριφορά των παικτών κατά τη διάρκεια του αγώνα

Ο αγώνας διακόπτεται και αρχίζει επαναληπτικός αγώνας υπό τις ακόλουθες προϋποθέσεις:

- Τα ρομπότ περιπλέκονται ή περιστρέφονται το ένα γύρω από το άλλο χωρίς αντιληπτή πρόοδο για πέντε δευτερόλεπτα. Εάν δεν είναι σαφές εάν υπάρχει πρόοδος ή όχι, ο κριτής μπορεί να παρατείνει το χρονικό όριο για την παρατηρήσιμη πρόοδο έως και 30 δευτερόλεπτα.
- Και τα δύο robot κινούνται, χωρίς να σημειώνουν πρόοδο, ή σταματούν (ακριβώς την ίδια στιγμή) και παραμένουν σταματημένα για πέντε δευτερόλεπτα χωρίς να αγγίζουν το ένα το άλλο. Ωστόσο, εάν ένα robot σταματήσει πρώτο την κίνησή του, μετά από πέντε δευτερόλεπτα θα δηλωθεί ότι δεν έχει τη θέληση να αγωνιστεί. Σε αυτή την περίπτωση ο αντίπαλος θα λάβει ένα Yuhkoh, ακόμη και αν ο αντίπαλος σταματήσει επίσης. Αν και τα δύο robot κινούνται και δεν είναι σαφές αν υπάρχει πρόοδος ή όχι, ο κριτής μπορεί να παρατείνει το χρονικό όριο έως και 30 δευτερόλεπτα.
- Εάν και τα δύο robot αγγίξουν την εξωτερική πλευρά του δακτυλίου περίπου την ίδια στιγμή και δεν μπορεί να προσδιοριστεί ποιο άγγιξε πρώτο, προκηρύσσεται επαναληπτικός αγώνας.

3.10 Παραβάσεις

Οι παρακάτω ενέργειες των συμμετεχόντων βρίσκονται σε αντίθεση με τους υφιστάμενους κανονισμούς και συνεπώς αποτελούν παραβίαση αυτών.

Παραβάσεις μεγάλης σοβαρότητας:

- Προσβλητικό λεξιλόγιο στην αντίπαλη ομάδα ή στην διαιτησία οποιασδήποτε μορφής.

Παραβάσεις μικρής σοβαρότητας:

- Εισβολή του συμμετέχοντα στο ρινγκ κατά την διάρκεια του αγώνα.
- Απαίτηση διακοπής του αγώνα χωρίς κάποιον τεκμηριωμένο λόγο.
- Χρειάζεται περισσότερο από 30 δευτερόλεπτα για να συνεχίσει τον αγώνα (μετά από κάποια διακοπή ή λήξη κάποιου γύρου).
- Κατά την έναρξη του γύρου το sumo robot ξεκινάει να λειτουργεί πριν τα 5 δευτερόλεπτα
- Το sumo robot λειτουργεί κατά τρόπο που υπονομεύει την ακεραιότητα του αγώνα.

3.11 Κυρώσεις

Στην περίπτωση που κάποιος συμμετέχων χρησιμοποιήσει προσβλητικό λεξιλόγιο οποιασδήποτε μορφής ή δεν υπακούσει στους περιορισμούς, ο κριτής δίνει δύο πόντους Yuhkoh στον αντίπαλο και διατάζει τον παραβάτη να αποχωρήσει. Στην περίπτωση των παραβιάσεων μικρής σοβαρότητας οι παραβάσεις συσσωρεύονται. Οι δύο παραβάσεις δίνουν ένα πόντο Yuhkoh στον αντίπαλο.

3.12 Τραυματισμοί και ατυχήματα κατά τη διάρκεια του αγώνα

Αίτημα διακοπής του αγώνα, μπορεί να απαιτήσει κάποιος παίκτης αν τραυματιστεί ο ίδιος ή το robot του εμπλακεί σε ατύχημα. Κατά την αδυναμία συνέχισης του αγώνα λόγω τραυματισμού του παίκτη ή

ατυχήματος του ρομπότ, ο παίκτης που είναι η αιτία του τραυματισμού ή του ατυχήματος χάνει τον αγώνα. Όταν δεν είναι ξεκάθαρο ποια ομάδα είναι η αιτία, ο παίκτης που δεν μπορεί να συνεχίσει το παιχνίδι ή που ζητά να σταματήσει το παιχνίδι, κηρύσσεται ηττημένος. Το αν το παιχνίδι πρέπει να συνεχιστεί σε περίπτωση τραυματισμού ή ατυχήματος αποφασίζεται από τους κριτές και τα μέλη της επιτροπής. Η διαδικασία λήψης απόφασης δεν πρέπει να διαρκεί περισσότερο από πέντε λεπτά.

Όταν ένας παίκτης κερδίζει τον αγώνα λόγω τραυματισμού ή ατυχήματος του αντιπάλου του, λαμβάνει δύο πόντους Yuhkoh. Αν ο ηττημένος παίκτης έχει ήδη κερδίσει έναν πόντο Yuhkoh, αυτός ο πόντος παραμένει καταγεγραμμένος για τον ηττημένο. Σε περίπτωση που η κατάσταση που περιεγράφηκε στην αδυναμία συνέχισης του αγώνα συμβεί κατά τη διάρκεια ενός παρατεταμένου αγώνα, ο νικητής του αγώνα λαμβάνει έναν πόντο Yuhkoh.

3.13 Ενστάσεις

- Δεν γίνονται δεκτές αντιρρήσεις κατά των αποφάσεων των κριτών
- Ο επικεφαλής μιας ομάδας μπορεί να υποβάλει ενστάσεις στην Επιτροπή, πριν από τη λήξη του αγώνα, εάν υπάρχουν αμφιβολίες κατά την εφαρμογή των παρόντων κανόνων. Εάν δεν υπάρχουν παρόντα μέλη της Επιτροπής, η ένσταση μπορεί να υποβληθεί στον κριτή πριν από τη λήξη του αγώνα.

3.14 Επίλογος

Εν κατακλείδι, στο Κεφάλαιο 3 γίνεται εμβάθυνση στους κανονισμούς των αγώνων προκειμένου να διασφαλιστεί η ομαλή και απρόσκοπτη διεξαγωγή τους. Εξετάζονται λεπτομερώς οι κατηγορίες των αγώνων, οι οποίες διακρίνονται με βάση την κατηγορία χειρισμού, το μέγεθος και το βάρος των sumo robot. Οι κανονισμοί εφαρμόζονται ομοιόμορφα για όλες τις κατηγορίες, ενώ αναλύονται επίσης οι συνθήκες διεξαγωγής του αγώνα, οι περιορισμοί και οι απαγορεύσεις, καθώς και τα κριτήρια που θεωρούνται καθοριστικά για την επιτυχία ή την ήττα στον αγώνα.

Κεφάλαιο 4^ο: Arduino

4.1 Εισαγωγή στο Arduino

Το Arduino, το οποίο ελέγχει ένα ευρύ φάσμα λειτουργιών, μπορεί να θεωρηθεί ως ο εγκέφαλος ενός ηλεκτρικού ή ηλεκτρονικού συστήματος. Το Arduino είναι μια πλακέτα υλικού και λογισμικού ανοικτού κώδικα που περιλαμβάνει έναν μικροελεγκτή και συνδέεται με έναν υπολογιστή μέσω ενός απλού περιβάλλοντος ανάπτυξης.

Ο μικροελεγκτής ATmega της Atmel είναι το κεντρικό στοιχείο αυτού του ηλεκτρονικού κυκλώματος. Μπορεί να υλοποιηθεί από οποιονδήποτε, επειδή όλα τα ηλεκτρονικά σχέδια και το λογισμικό που απαιτούνται για τον προγραμματισμό και τη λειτουργία του διανέμονται δωρεάν. Το Arduino μπορεί να λειτουργήσει παρόμοια με έναν μικρό υπολογιστή. Με άλλα λόγια, μπορεί να συνδεθεί ένα πλήθος συσκευών εισόδου/εξόδου και ο μικροελεγκτής μπορεί να προγραμματιστεί ώστε να λαμβάνει δεδομένα από μονάδες εισόδου (όπως διακόπτες και αισθητήρες), να τα επεξεργάζεται και στη συνέχεια να στέλνει τις απαραίτητες εντολές σε μονάδες εξόδου (όπως κινητήρες, φώτα κ.λπ.). Φυσικά, το Arduino δεν είναι η μόνη επιλογή για τη δημιουργία διαδραστικών ηλεκτρονικών συσκευών υπάρχουν και άλλες πλατφόρμες και τεχνολογίες που μπορούν να χρησιμοποιηθούν για τον ίδιο σκοπό [4].



Εικόνα 4.1: Arduino Logo

4.2 Ιστορική αναδρομή του Arduino

Το έτος 2003 στο Ινστιτούτο Σχεδιαστικής Αλληλεπίδρασης Ivrea στην Ιταλία, ο Massimo Banzi και ο David Cuartielles άρχισαν να αναπτύσσουν το Arduino για να προσφέρουν μια οικονομική και εύκολη στη χρήση πλατφόρμα μικροελεγκτή για τους σπουδαστές τους. Στην υλοποίηση του λογισμικού βοήθησαν δύο φοιτητές του Ινστιτούτου. Στην ίδια περιοχή σε ένα μικρού μεγέθους εργαστήριο ξεκίνησε η παραγωγή των πρώτων πλακετών. Το 2005 το πρώτο πρωτότυπο, γνωστό ως "Arduino Serial", κυκλοφόρησε χρησιμοποιώντας τον μικροελεγκτή Atmel ATmega8 με περιορισμένη συνδεσιμότητα μέσω σειριακής θύρας, έμπνευση του ονόματος Arduino ήταν ο Βασιλιάς Arduin της Ivrea. Η πλατφόρμα ήταν ανοιχτού κώδικα, επιτρέποντας σε άλλους να τη βελτιώσουν και να τη διανέμουν ελεύθερα. Το 2006 εμφανίστηκε το "Arduino NG" (Next Generation) κυκλοφόρησε, φέρνοντας βελτιώσεις όπως η δυνατότητα επαναπρογραμματισμού μέσω USB και βελτιωμένη τροφοδοσία. Οι βελτιώσεις δεν σταμάτησαν και συνεχίστηκε η δημιουργία και κυκλοφορία πιο εξελιγμένων μοντέλων [5, 8].

4.3 Μοντέλα πλακετών Arduino

Η πλατφόρμα Arduino περιλαμβάνει δύο κύρια συστατικά: το υλικό (hardware) και το λογισμικό (software). Το υλικό του Arduino περιλαμβάνει μια σειρά από πλακέτες που μπορούν να συνδεθούν με διάφορους αισθητήρες και άλλες περιφερειακές συσκευές. Ανάλογα με τις απαιτήσεις που έχουμε καλούμαστε να διαλέξουμε το καταλληλότερο μοντέλο για την εφαρμογή που έχουμε σκοπό να υλοποιήσουμε. Οι πιο δημοφιλείς προτάσεις είναι οι εξής:

- **Arduino Nano:** Είναι μια μικρή, ευέλικτη πλακέτα που βασίζεται στον ATmega328 (ή παλαιότερα ATmega168), ιδανική για έργα που απαιτούν μικρό μέγεθος και χαμηλή κατανάλωση ενέργειας.



Εικόνα 4.2: Arduino Nano

- **Arduino Uno:** Η πιο κοινή και δημοφιλή πλακέτα, ιδανική για αρχάριους και βασικές εφαρμογές. Χρησιμοποιεί τον ATmega328P.



Εικόνα 4.3: Arduino Uno

- **Arduino Mega:** Προσφέρει περισσότερες εισόδους/εξόδους και μεγαλύτερη μνήμη, κατάλληλη για πιο σύνθετα έργα. Χρησιμοποιεί τον ATmega2560.



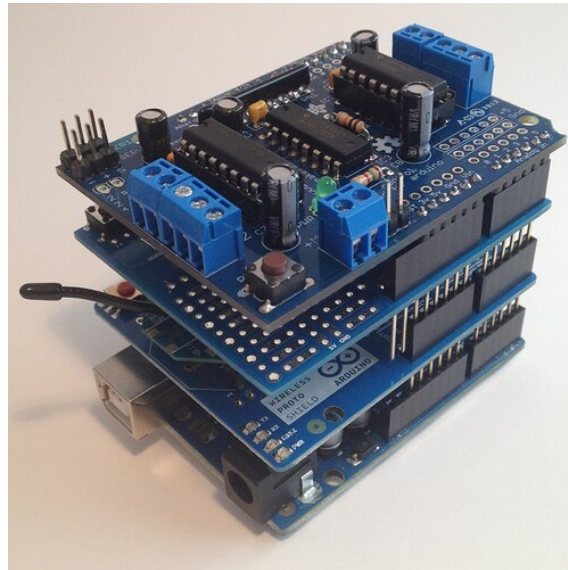
Εικόνα 4.4: Arduino Mega

4.4 Arduino Shields

Οι επεκτάσεις του Arduino, γνωστές ως Arduino Shields, στερεώνονται απευθείας σε μια πλακέτα Arduino για να προσφέρουν επιπλέον χαρακτηριστικά και δυνατότητες. Αυτές οι πλακέτες χρησιμοποιούν τους ίδιους ακροδέκτες και κουμπώνουν εύκολα πάνω στην κύρια πλακέτα Arduino, καθιστώντας την εγκατάσταση απλή και γρήγορη, χωρίς να απαιτούνται εξειδικευμένες γνώσεις ηλεκτρονικών. Με αυτόν τον τρόπο, η επέκταση των δυνατοτήτων του Arduino γίνεται εύκολη και αποφεύγονται οι περίπλοκες συνδέσεις, επιτρέποντας στους χρήστες να προσθέτουν νέες λειτουργίες χωρίς τον κόπο του σχεδιασμού και της κατασκευής πολύπλοκων κυκλωμάτων. Τα shield είναι σχεδιασμένα με τέτοιο τρόπο ώστε αφού συνδεθούν πάνω στο Arduino να προωθούν τις υποδοχές του,

Κεφάλαιο 4

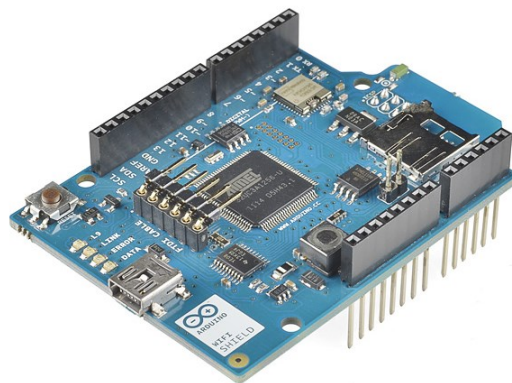
ώστε να μπορούμε να συνδέσουμε περισσότερα του ενός Arduino Shield πάνω σε μία πλακέτα Arduino και επιπλέον να συνδέσουμε και δικά μας εξαρτήματα [4, 5].



Εικόνα 4.5: Χρήση πολλαπλών Shields

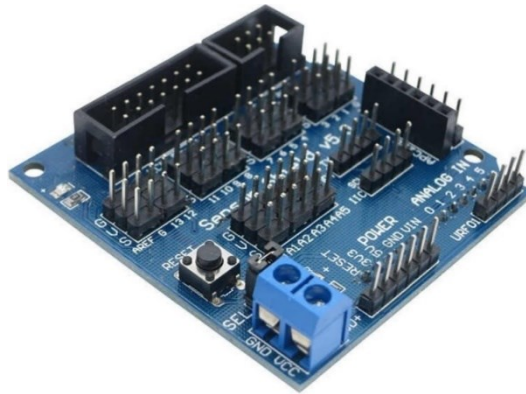
Μερικά από τα δημοφιλέστερα Arduino Shields είναι:

- **Arduino WiFi Shield:** Έχει την ίδια λειτουργία με το Ethernet Shield αλλά λειτουργεί ασύρματα.



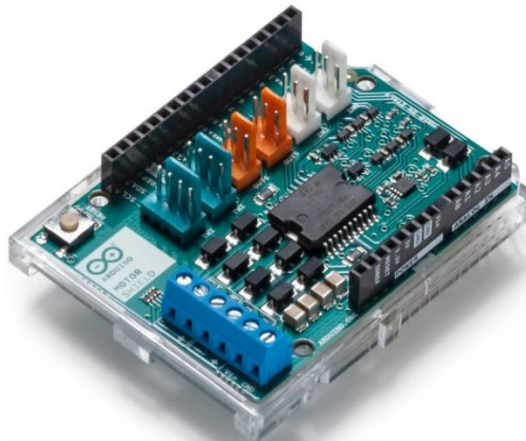
Εικόνα 4.6: Arduino Wifi Shield

- **Arduino Sensor Shield:** Επιτρέπει την εύκολη σύνδεση και διαχείριση διαφόρων αισθητήρων και συσκευών εισόδου/εξόδου.



Εικόνα 4.7: Arduino Sensor Shield

- **Arduino Motor Shield:** Επιτρέπει τον έλεγχο κινητήρων



Εικόνα 4.8: Arduino Motor Shield

4.5 Πλεονεκτήματα και μειονεκτήματα Arduino

Το Arduino είναι μια ισχυρή και προσιτή πλατφόρμα για αρχάριους και επαγγελματίες, με πληθώρα δυνατοτήτων και πλεονεκτημάτων. Ωστόσο, έχει κάποιους περιορισμούς που πρέπει να ληφθούν υπόψη ανάλογα με τις απαιτήσεις της εκάστοτε εργασίας.

Το Arduino είναι σχεδιασμένο για να είναι φιλικό προς τους αρχάριους, με απλό λογισμικό και προγραμματιστικό περιβάλλον (IDE) που καθιστά εύκολη τη δημιουργία και φόρτωση κώδικα στην πλακέτα. Το υλικό και το λογισμικό του Arduino είναι ανοιχτού κώδικα, επιτρέποντας στους χρήστες να κατανοήσουν και να τροποποιήσουν το σχέδιο και το λογισμικό. Η μεγάλη κοινότητα χρηστών του Arduino παρέχει πληθώρα από βιβλιοθήκες, κώδικες, παραδείγματα και υποστήριξη, κάνοντας την

επίλυση προβλημάτων και τη μάθηση ευκολότερη. Μπορεί να χρησιμοποιηθεί σε μια ποικιλία εφαρμογών, από απλά έργα DIY μέχρι επαγγελματικές εργασίες, όπως ρομποτική, αυτοματισμούς και διασυνδεδεμένες συσκευές (IoT). Υπάρχουν πολλά πρόσθετα (Shields) και αξεσουάρ που μπορούν να συνδεθούν με το Arduino, επεκτείνοντας τις δυνατότητές του και επιτρέποντας τη σύνδεση με διάφορους αισθητήρες. Το κόστος είναι ένας σημαντικός παράγοντας που μπορεί να ωθήσει κάποιον στην αγορά μιας πλακέτας Arduino. Το κόστος της original πλακέτας δεν ξεπερνάει τα 50 ευρώ, ωστόσο υπάρχουν και οι αντιγραφές οι οποίες έχουν ακόμα πιο προσιτή τιμή.

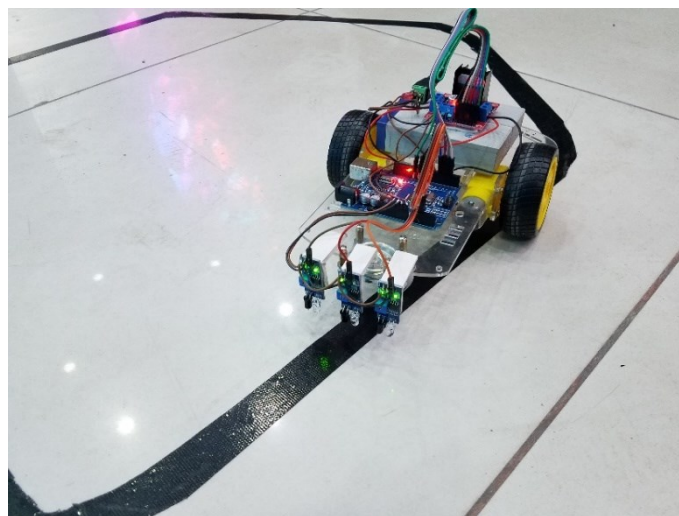
Από την άλλη μεριά ορισμένοι περιορισμοί των πλακετών Arduino μπορεί να αποθαρρύνουν κάποιον ενδιαφερόμενο στο να αγοράσει το προϊόν. Οι μικροελεγκτές του Arduino δεν έχουν την ίδια επεξεργαστική ισχύ και μνήμη με άλλες πιο προηγμένες πλατφόρμες, όπως το Raspberry Pi. Αυτό μπορεί να περιορίσει την απόδοση σε απαιτητικές εργασίες. Ενώ υπάρχουν Shields για WiFi και Ethernet, το Arduino από μόνο του δεν διαθέτει ενσωματωμένη δυνατότητα δικτύωσης, καθιστώντας τη σύνδεση στο διαδίκτυο και τις ασύρματες επικοινωνίες πιο περίπλοκες. Το IDE του Arduino, ενώ είναι φιλικό προς τους αρχάριους, μπορεί να είναι περιορισμένο για πιο προχωρημένους χρήστες που επιθυμούν περισσότερα εργαλεία και δυνατότητες ενσωματωμένες στον προγραμματισμό [7].

4.6 Εφαρμογές Arduino

Οι εφαρμογές που μπορούν να πραγματοποιηθούν με μια πλακέτα Arduino είναι αμέτρητες. Στο κεφάλαιο παρουσιάζονται κάποιες από τις δημοφιλέστερες αυτές εφαρμογές:

4.6.1 Ρομποτικές εφαρμογές

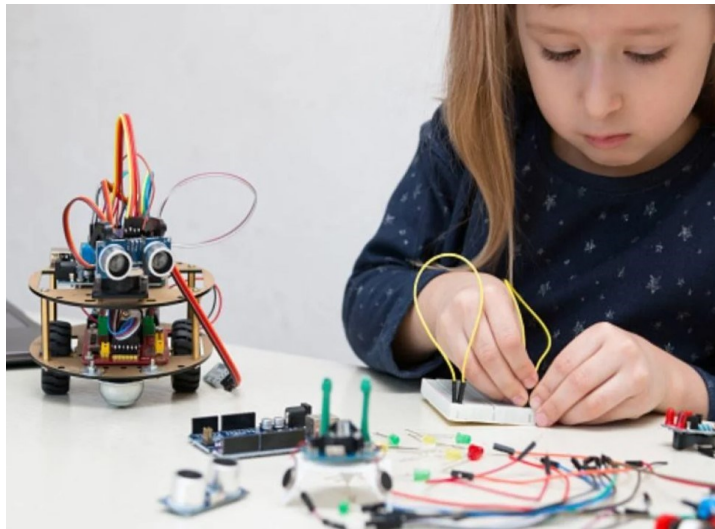
Ορισμένες από τις πιο δημοφιλείς εκπαιδευτικές εφαρμογές περιλαμβάνουν την κατασκευή αυτόνομων ρομπότ που ακολουθούν μια μαύρη γραμμή στο έδαφος και ρομπότ που ανιχνεύουν και αποφεύγουν εμπόδια, και στις δύο περιπτώσεις χρησιμοποιώντας αισθητήρες. Ο μικροελεγκτής έχει ως αποστολή τον έλεγχο των διάφορων εξαρτημάτων που συνθέτουν αυτές τις κατασκευές.



Εικόνα 4.9: Line Following Robot

4.6.2 Εκπαιδευτικές εφαρμογές

Οι εφαρμογές του Arduino στην εκπαίδευση είναι πολυάριθμες και ποικίλες. Στην εισαγωγή στην ηλεκτρονική, το Arduino χρησιμοποιείται ως βασικό εργαλείο για να διδάξει στους μαθητές θεμελιώδεις αρχές της ηλεκτρονικής και του προγραμματισμού. Οι μαθητές μαθαίνουν να δημιουργούν και να διαχειρίζονται κυκλώματα, ενώ παράλληλα εξοικειώνονται με τον προγραμματισμό μικροελεγκτών. Αυτό έχει ως αποτέλεσμα να ενισχύονται οι δεξιότητές τους στην επίλυση προβλημάτων, τη δημιουργική σκέψη και την πρακτική εφαρμογή της γνώσης.



Εικόνα 4.10: Εκπαιδευτικές εφαρμογές

4.6.3 IoT εφαρμογές

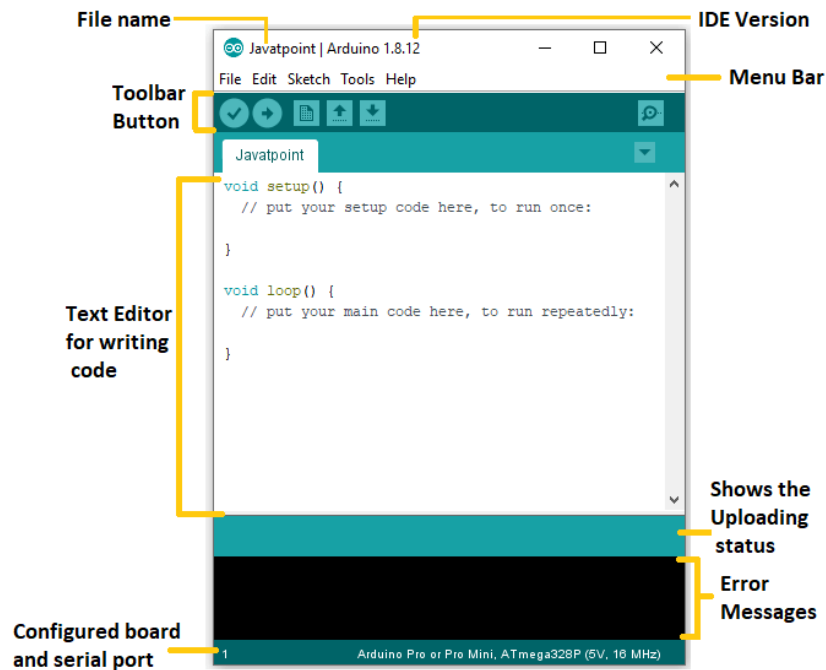
Το Διαδίκτυο των Πραγμάτων (IoT) αξιοποιεί το Arduino σε πολλές εφαρμογές. Ένα χαρακτηριστικό παράδειγμα είναι οι έξυπνοι θερμοστάτες, οι οποίοι επιτρέπουν τον απομακρυσμένο έλεγχο της θερμοκρασίας του σπιτιού μέσω αισθητήρων και την αμφίδρομη επικοινωνία μέσω διαδικτύου. Αυτός ο έλεγχος βελτιώνει την άνεση, δίνοντας στους χρήστες τη δυνατότητα να ρυθμίζουν τη θερμοκρασία του σπιτιού τους οποιαδήποτε στιγμή και από οπουδήποτε. Επιπλέον, τα συστήματα παρακολούθησης κατανάλωσης ενέργειας είναι άλλη μια εφαρμογή, όπου το Arduino μετρά και καταγράφει την ενεργειακή χρήση σε πραγματικό χρόνο.

4.7 Περιβάλλον προγραμματισμού του Arduino

Το Arduino IDE (Integrated Development Environment) είναι ένα λογισμικό που χρησιμοποιείται για τη συγγραφή, την επεξεργασία και τη μεταφόρτωση κώδικα σε πλακέτες Arduino, είναι μία εφαρμογή γραμμένη σε γλώσσα προγραμματισμού Java. Το Arduino IDE παρέχει ένα απλό και φιλικό περιβάλλον για την ανάπτυξη κώδικα, διευκολύνοντάς τους νεοεισερχομένους. Η γλώσσα προγραμματισμού του Arduino IDE βασίζεται στη C++, η οποία συνεργάζεται με την βιβλιοθήκη Wiring η οποία αναπτύχθηκε ως μέρος του Arduino project για να απλοποιήσει τη διαδικασία προγραμματισμού των μικροελεγκτών. Με την χρήση του Wiring οι χρήστες μπορούν να προγραμματίσουν ένα Arduino χωρίς να χρειάζεται

να έχουν εκτενή γνώση της C++. Το πρόγραμμα στο Arduino IDE ονομάζεται sketch και αποτελείται από δύο κύρια μέρη, το setup() και το loop() [4].

- Η συνάρτηση setup(), χρησιμοποιείται για αρχικοποιήσεις που πρέπει να εκτελούνται μία φορά, όταν ξεκινάει το πρόγραμμα. Όπως ρυθμίσεις των ακίδων (pins) ως εισόδους ή εξόδους, έναρξη σειριακής επικοινωνίας κλπ.
- Η συνάρτηση loop(), περιέχει τον κώδικα που εκτελείται συνεχώς, μέχρι να απενεργοποιηθεί το Arduino. Είναι ο κύριος "βρόχος" του προγράμματος.



Εικόνα 4.11: Arduino IDE

Τα κουμπιά της γραμμής εντολών είναι :



Νέο: Δημιουργεί νέο sketch



Άνοιγμα: Επιλογή ενός sketch που θα ανοίξει μέσα στο τρέχον παράθυρο



Σειριακή οθόνη: Αποστολή και λήψη δεδομένων μεταξύ του Arduino και του υπολογιστή



Έλεγχος/ Μεταγλώττιση: Έλεγχος για λάθη στον κώδικα

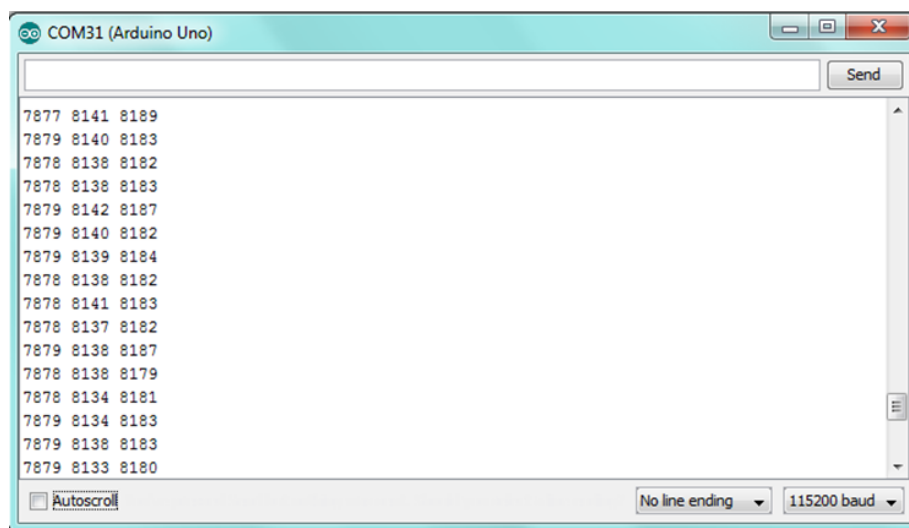


Upload: Ανέβασμα του κώδικα στο μικροελεγκτή



Αποθήκευση: Αποθηκεύει το sketch.

Η σειριακή οθόνη (Serial Monitor) στο Arduino IDE επιτρέπει την επικοινωνία μεταξύ του Arduino και του υπολογιστή, διευκολύνοντας την αποστολή και λήψη δεδομένων. Αυτό το εργαλείο είναι ιδανικό για τον έλεγχο και τον εντοπισμό σφαλμάτων στον κώδικα, καθώς επιτρέπει την προβολή μηνυμάτων και τιμών σε πραγματικό χρόνο.



Εικόνα 4.12: Σειριακή οθόνη Arduino IDE

4.8 Επίλογος

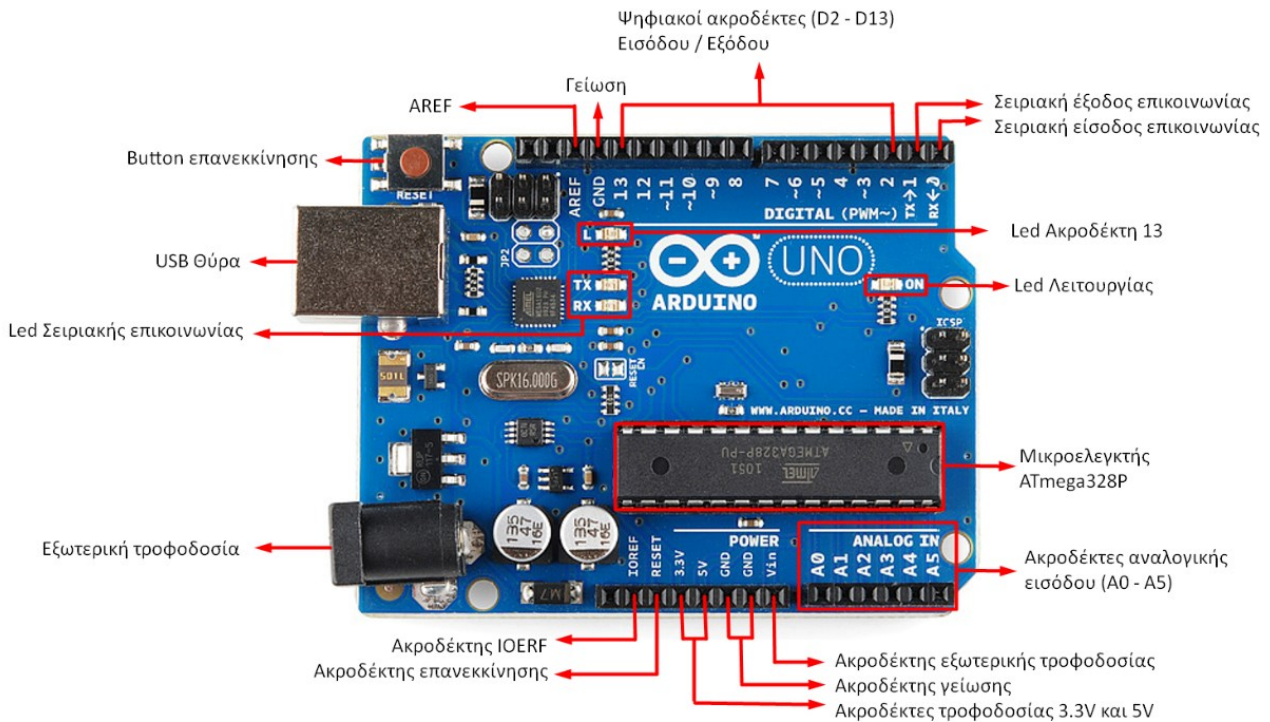
Το κεφάλαιο αυτό προσέφερε μια εκτενή εισαγωγή στο Arduino, καλύπτοντας τις βασικές λειτουργίες και τις εφαρμογές του. Αναφέραμε διαφορετικά μοντέλα πλακετών, τα πλεονεκτήματα και τα μειονεκτήματα της πλατφόρμας, καθώς και τις διάφορες επεκτάσεις μέσω των Arduino Shields.

Η ευελιξία και η προσιτή φύση του Arduino το καθιστούν ιδανικό εργαλείο για μια πληθώρα εφαρμογών, από απλά έργα DIY μέχρι σύνθετα επαγγελματικά έργα, όπως ρομποτική και IoT. Το φιλικό προς τον χρήστη IDE, σε συνδυασμό με την ανοιχτή κοινότητα και τις πολλές διαθέσιμες βιβλιοθήκες, επιτρέπει σε αρχάριους και επαγγελματίες να δημιουργούν και να αναπτύσσουν τα έργα τους με ευκολία.

Η κατανόηση του τρόπου λειτουργίας του Arduino και η δυνατότητα προγραμματισμού του ανοίγουν νέες προοπτικές για δημιουργία και καινοτομία σε διάφορους τομείς της ηλεκτρονικής και της μηχανικής. Συνολικά, το κεφάλαιο αυτό έθεσε τις βάσεις για την περαιτέρω εξερεύνηση και αξιοποίηση του Arduino σε διάφορες δημιουργικές και εκπαιδευτικές εφαρμογές.

Κεφάλαιο 5^ο: Arduino Uno

Σε αυτό το κεφάλαιο θα αναλυθεί το Arduino Uno που χρησιμοποιείται στην κατασκευή της παρούσας εργασίας.



Εικόνα 5.1: Arduino Uno (Επεξήγηση)

Η πλακέτα Arduino Uno βασίζεται στον μικροελεγκτή ATmega328P της οικογένειας AVR από την εταιρεία Atmel (νυν Microchip Technology). Αυτός ο μικροεπεξεργαστής διαθέτει 32KB μνήμης flash, 2KB SRAM και 1KB EEPROM, επιτρέποντας την αποθήκευση και την εκτέλεση προγραμμάτων με σχετική άνεση για πολλές εφαρμογές. Ενσωματώνει 14 ψηφιακές θύρες (D0-D13), από τις οποίες οι 6 μπορούν να χρησιμοποιηθούν ως PWM εξοδοί, 6 αναλογικές θύρες (A0-A5), οι οποίες μπορούν επίσης να χρησιμοποιηθούν ως ψηφιακές είσοδοι/εξοδοί αν χρειαστεί και έναν ταλαντωτή 16MHz, καθιστώντας τον ιδανικό για έργα που απαιτούν ακριβή χρονισμό και πολλαπλές συνδέσεις με αισθητήρες. Η σχεδίασή του δίνει έμφαση στην αποδοτικότητα της ενέργειας, καθιστώντας τον κατάλληλο για έργα που τροφοδοτούνται από μπαταρίες. Επιπλέον, η υποστήριξή του από την κοινότητα του Arduino παρέχει μια πλούσια βιβλιοθήκη κώδικα και παραδειγμάτων, διευκολύνοντας την ανάπτυξη πρωτοτύπων και την εκμάθηση της προγραμματιστικής λογικής. Το Arduino Uno λειτουργεί με τροφοδοσία 5V DC από την είσοδο του USB ή με 7V~12V DC από την είσοδο της τροφοδοσίας [7, 10].

5.1 Χαρακτηριστικά Arduino Uno

- Επεξεργαστής: ATmega328P
- Τάση Λειτουργίας: 5V
- Τάση Εισόδου (συνιστάμενη): 7-12V
- Τάση Εισόδου (όρια): 6-20V
- Ψηφιακές Είσοδοι/Εξοδοι: 14 (από τις οποίες οι 6 μπορούν να χρησιμοποιηθούν ως PWM έξοδοι)
- Αναλογικές Είσοδοι: 6
- Ρεύμα DC ανά I/O Pin: 20 mA
- Ρεύμα DC για 3.3V Pin: 50 mA
- Μνήμη Flash: 32 KB (0.5 KB χρησιμοποιούνται από το bootloader)
- SRAM: 2 KB
- EEPROM: 1 KB
- Ρολόι: 16 MHz
- Μέγεθος: 68.6 mm x 53.4 mm
- Βάρος: περίπου 25g

5.2 Είσοδοι / Έξοδοι

Το Arduino Uno διαθέτει 6 αναλογικές εισόδους, οι οποίες είναι αριθμημένες από A0 έως A5. Αυτές οι εισοδοι μπορούν να διαβάσουν αναλογικά σήματα, δηλαδή σήματα που έχουν συνεχή τιμή σε μια συγκεκριμένη κλίμακα. Χρησιμοποιούνται συχνά για να διαβάσουν αισθητήρες που παρέχουν τιμές τάσης που αντιπροσωπεύουν φυσικά μεγέθη όπως θερμοκρασία, φως ή πίεση. Οι αναλογικές εισοδοι μετατρέπουν το αναλογικό σήμα σε ψηφιακή τιμή μέσω του ενσωματωμένου αναλογικού-σε-ψηφιακό μετατροπέα (ADC) του Arduino, ο οποίος έχει ανάλυση 10-bit. Αυτό σημαίνει ότι η αναλογική τιμή μετατρέπεται σε μια ψηφιακή τιμή μεταξύ 0 και 1023.

Ο συγκεκριμένος μικροελεγκτής επίσης διαθέτει 14 ψηφιακές εισόδους/εξόδους, αριθμημένες από 0 έως 13. Αυτές οι ακίδες μπορούν να ρυθμιστούν είτε ως εισοδοι είτε ως έξοδοι μέσω του κώδικα. Ως εισοδοι, οι ακίδες μπορούν να διαβάσουν δύο καταστάσεις: HIGH (υψηλή τάση – 5V) ή LOW (χαμηλή τάση – 0V), γεγονός που αντιστοιχεί σε ένα δυαδικό σήμα. Το ίδιο ισχύει για και για την χρήση των ακροδεκτών ως εξόδους [6, 9].

5.3 Διπλής λειτουργίας ακροδέκτες

Οι ακροδέκτες 3, 5, 6, 9, 10 και 11 του Arduino έχουν την ικανότητα να λειτουργούν ως αναλογικές έξοδοι χρησιμοποιώντας τη μέθοδο PWM (Pulse Width Modulation). Αυτή η τεχνική, που χρησιμοποιείται επίσης στις μητρικές των υπολογιστών για τον έλεγχο της ταχύτητας των ανεμιστήρων, επιτρέπει τον λεπτομερή έλεγχο διαφόρων στοιχείων. Όταν συνδέουμε ένα στοιχείο σε έναν από αυτούς τους ακροδέκτες, μπορούμε να ρυθμίσουμε την τάση του με ακρίβεια 8 bit, προσφέροντας έναν εναλλακτικό τρόπο λειτουργίας αντί της απλής ενεργοποίησης ή απενεργοποίησης, όπως συμβαίνει με τις υπόλοιπες ψηφιακές εξόδους.

Οι ακροδέκτες 2 και 3 του Arduino έχουν διπλή λειτουργία, καθώς μπορούν να λειτουργούν τόσο ως ψηφιακές είσοδοι/έξοδοι όσο και ως ακροδέκτες διακοπής (interrupt pins). Αυτό σημαίνει ότι εκτός από τις βασικές λειτουργίες ενεργοποίησης και απενεργοποίησης ή ανάγνωσης ψηφιακών σημάτων, μπορούν να χρησιμοποιηθούν για να ανιχνεύουν αλλαγές στην κατάσταση του σήματος και να εκτελούν συγκεκριμένες λειτουργίες άμεσα μέσω των εξωτερικών διακοπών. Αυτή η δυνατότητα είναι ιδιαίτερα χρήσιμη για εφαρμογές που απαιτούν γρήγορη απόκριση σε εξωτερικά γεγονότα, όπως η ανίχνευση κίνησης.

Οι ακροδέκτες 0 και 1 του Arduino έχουν διπλή λειτουργία, καθώς μπορούν να λειτουργούν τόσο ως ψηφιακές είσοδοι/έξοδοι όσο και ως σειριακές γραμμές επικοινωνίας. Συγκεκριμένα, ο ακροδέκτης 0 χρησιμοποιείται ως σειριακή γραμμή λήψης (RX) και ο ακροδέκτης 1 ως σειριακή γραμμή μετάδοσης (TX). Αυτή η λειτουργία επιτρέπει στο Arduino να επικοινωνεί με άλλες συσκευές, όπως υπολογιστές, άλλες πλακέτες Arduino ή διάφορα περιφερειακά, μέσω σειριακής επικοινωνίας. Η χρήση των ακροδεκτών 0 και 1 για επικοινωνία UART είναι ιδιαίτερα χρήσιμη για την αποστολή και λήψη δεδομένων, τον προγραμματισμό της πλακέτας και τη διάγνωση σφαλμάτων μέσω σειριακής οθόνης [6, 9].

5.4 Τροφοδοσία

Η τροφοδοσία του Arduino μπορεί να πραγματοποιηθεί με διάφορους τρόπους, προσφέροντας ευελιξία και προσαρμοστικότητα σε διάφορες εφαρμογές. Το Arduino μπορεί να τροφοδοτηθεί μέσω του USB καλωδίου όταν είναι συνδεδεμένο σε υπολογιστή, παρέχοντας 5V απευθείας από τη θύρα USB. Επιπλέον, διαθέτει ένα βύσμα τροφοδοσίας (barrel jack) που δέχεται εξωτερική τάση από 7 έως 12V, επιτρέποντας τη χρήση εξωτερικών τροφοδοτικών ή μπαταριών. Υπάρχει επίσης η δυνατότητα τροφοδοσίας μέσω του ακροδέκτη VIN, ο οποίος επιτρέπει την παροχή τάσης απευθείας στην πλακέτα από εξωτερική πηγή. Το Arduino περιλαμβάνει ενσωματωμένο ρυθμιστή τάσης, ο οποίος διασφαλίζει ότι τα ευαίσθητα ηλεκτρονικά εξαρτήματα λαμβάνουν τη σωστή τάση λειτουργίας.

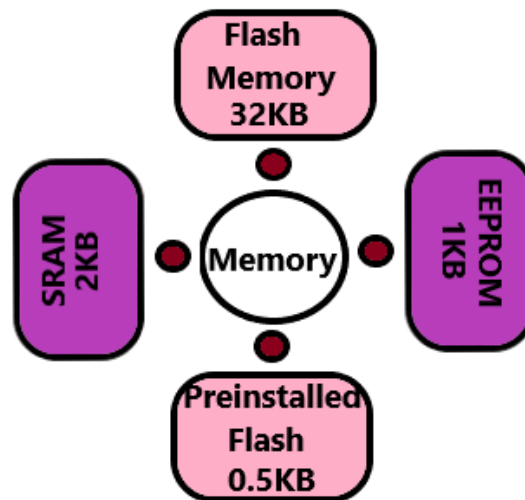
Όταν τροφοδοτείται το Arduino έχει δυνατότητα να χρησιμοποιήσει τα pins 3.3V και 5V σαν πηγή τροφοδοσίας. Στην περίπτωση που χρησιμοποιείται σαν μέθοδος τροφοδοσίας το barrel jack τότε υπάρχει η δυνατότητα μέσω του pin Vin να τροφοδοτηθούν στοιχεία με την πλήρη τάση τροφοδοσία πριν περάσει από τον ρυθμιστή τάσης οποίος είναι στα 5V [6, 9].

5.5 Μνήμη

Η μνήμη του μικροελεγκτή ATmega328P χωρίζεται σε τρεις κατηγορίες:

- Μνήμη Flash 32KB: Ένα μικρό μέρος αυτής της μνήμης, περίπου 0.5 KB, χρησιμοποιείται από το bootloader, ο οποίος επιτρέπει την εύκολη φόρτωση νέων προγραμμάτων (sketches) μέσω της σειριακής θύρας, αφήνοντας περίπου 31.5 KB διαθέσιμα για το πρόγραμμα του χρήστη. Η μνήμη Flash επιτρέπει στο Arduino να διατηρεί το πρόγραμμά του ακόμη και μετά από επανεκκίνηση ή απενεργοποίηση, κάτι που είναι απαραίτητο για την κανονική λειτουργία των συστημάτων που βασίζονται σε μικροελεγκτές.

- Μνήμη EEPROM 1 KB: Η μνήμη EEPROM (Electrically Erasable Programmable Read-Only Memory) επιτρέπει την αποθήκευση δεδομένων ακόμα και μετά την απενεργοποίηση της συσκευής.
- Μνήμη SRAM 2 KB: Η SRAM (Static Random Access Memory) είναι μια γρήγορη μνήμη η οποία χρησιμοποιείται για την αποθήκευση των μεταβλητών και των δεδομένων κατά την εκτέλεση του προγράμματος. Σε αντίθεση με τις δύο προηγούμενες μνήμες, η μνήμη SRAM χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή πατηθεί το κουμπί επανεκκίνησης.



Εικόνα 5.2: Μνήμη

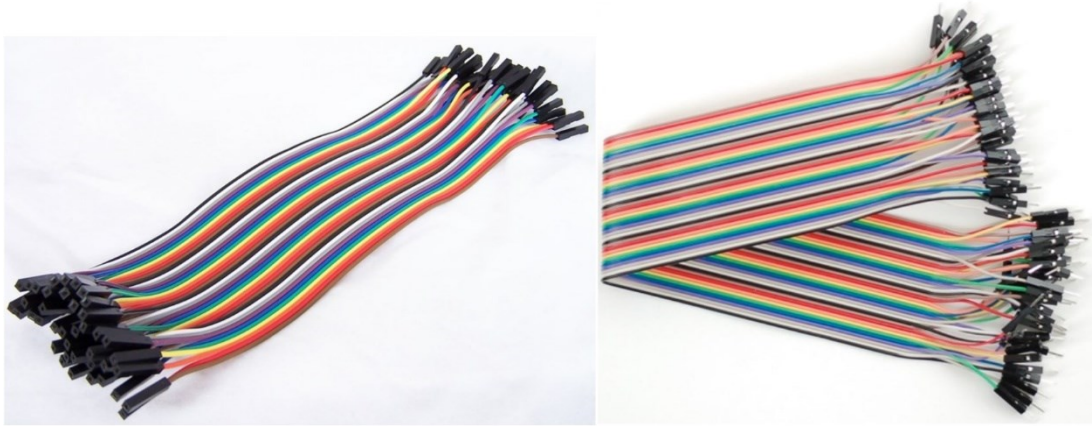
5.6 Επίλογος

Στο κεφάλαιο αυτό, εξετάστηκε λεπτομερώς η πλακέτα Arduino Uno. Η ανάλυση των χαρακτηριστικών του, όπως ο επεξεργαστής ATmega328P, οι δυνατότητες εισόδου/εξόδου, η χρήση των ακροδεκτών για PWM, διακοπές και σειριακή επικοινωνία, καθώς και οι επιλογές τροφοδοσίας, καταδεικνύουν την ικανότητα του Arduino Uno να υποστηρίζει ένα ευρύ φάσμα εφαρμογών, από απλά πειράματα μέχρι πιο περίπλοκα έργα. Επίσης, εξετάστηκε η διάρθρωση της μνήμης του μικροελεγκτή, συμπεριλαμβανομένων της μνήμης Flash, της EEPROM, και της SRAM, που εξυπηρετούν διαφορετικούς ρόλους στην αποθήκευση και εκτέλεση των προγραμμάτων. Μέσα από αυτό το κεφάλαιο, γίνεται φανερό ότι το Arduino Uno αποτελεί ένα πολύτιμο εργαλείο για κάθε δημιουργό που επιθυμεί να πειραματιστεί με την τεχνολογία και να υλοποιήσει τις ιδέες του με αποδοτικότητα και ευκολία.

Κεφάλαιο 6^ο: Επιλογή και ανάλυση υλικών

6.1 Βραχυκυκλωτήρες

Για την ευκολία σύνδεσης των εξαρτημάτων μεταξύ τους χρησιμοποιήθηκαν βραχυκυκλωτήρες του εμπορίου δύο ειδών, αρσενικό σε αρσενικό καθώς και θηλυκό σε θηλυκό.



Εικόνα 6.1: Βραχυκυκλωτήρες

6.2 Ταινία διπλής όψης / κόλλα

Για την σωστή τοποθέτηση των εξαρτημάτων επάνω στο πλαίσιο του ρομπότ χρησιμοποιήθηκε αφρώδης ταινία διπλής όψεως καθώς και UHU Super Glue στα σημεία που έχριζαν περισσότερη υποστήριξη. Ένα από αυτά τα σημεία είναι οι κινητήρες στους οποίους πέφτει το μεγαλύτερο βάρος της κατασκευής.



Εικόνα 6.2: Ταινία διπλής όψης / κόλλα

6.3 Sumo Ring (Dohyo)

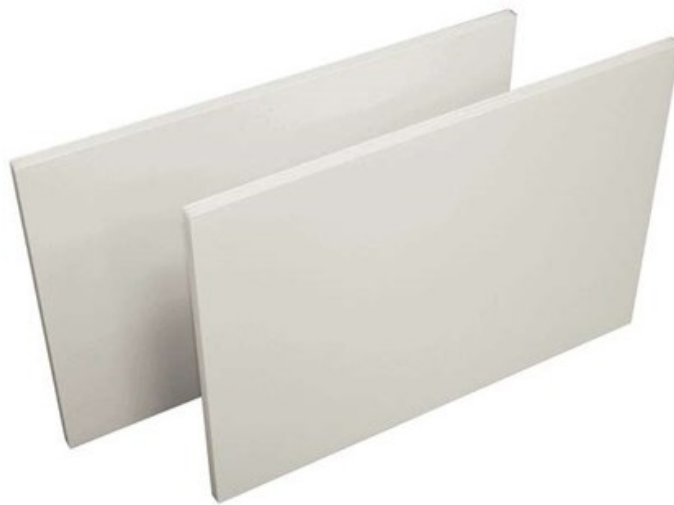
Για την κατασκευή του Dohyo χρησιμοποιήθηκε πάνελ από μελαμίνη. Επιλέχθηκε αυτό το υλικό είναι ανθεκτικό στις γρατζουνιές και στις φθορές, έχει λεία και ομαλή επιφάνεια, που μπορεί να βοηθήσει στην ομαλή κίνηση των ρομπότ. Καθώς και αντοχή στην υγρασία. Ενώ με την άσπρη επιφάνεια την οποία διαθέτει έχουμε έτοιμη την οριογραμμή. Η εσωτερική επιφάνεια του ρινγκ θα βαφτεί με πλαστικό μαύρο matt χρώμα.



Εικόνα 6.3: Μελαμίνη Dohyo

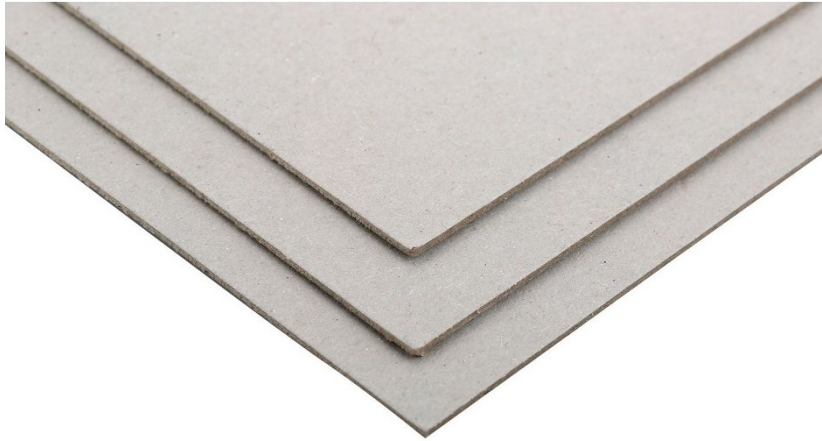
6.4 Πλαίσιο

Στην κατασκευή της βάσης χρησιμοποιήθηκε χαρτόνι μακέτας πάχους 5mm χρώματος άσπρο. Επιλέχθηκε το συγκεκριμένο πάχος γιατί παρέχει την απαραίτητη στιβαρότητα για να διατηρεί το πλαίσιο τη μορφή του, ενώ παράλληλα είναι εύκολο στην επεξεργασία του.



Εικόνα 6.4: Χαρτόνι μακέτας πλαισίου

Για την κατασκευή του προστατευτικού προφυλακτήρα, επιλέχθηκε γκρι χαρτόνι πάχους 3mm. Λόγω της θέσης του προφυλακτήρα, όπου αναμένεται να υποστεί πρόσκρουση, προτιμήθηκε ένα υλικό πιο συμπαγές και ανθεκτικό από το συνηθισμένο χαρτόνι μακέτας, διασφαλίζοντας έτσι την καλύτερη προστασία του ρομπότ κατά τη διάρκεια των αγώνων. Για την στήριξη του προφυλακτήρα επιλέχθηκαν πλαστικές γωνίες ντουλαπιών σε συνδυασμό με βίδες και παξιμάδια M3 μήκους 1.6cm.



Εικόνα 6.5: Χαρτόνι προφυλακτήρα

6.5 Ελαστικά

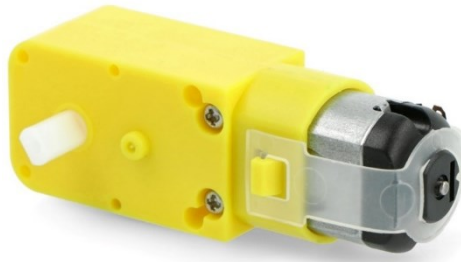
Τα ελαστικά μεταφέρουν την κίνηση από τους κινητήρες στο έδαφος. Είναι σχεδιασμένα να προσφέρουν καλή πρόσφυση και σταθερότητα. Τα υλικά κατασκευής τους περιλαμβάνουν καουτσούκ ή πολυουρεθάνη, ενώ το μέγεθός τους επιλέγεται ανάλογα με τις απαιτήσεις του αγωνιστικού πεδίου. Χρησιμοποιήθηκαν δύο τεμάχια ελαστικών κατασκευής από καουτσούκ με διάμετρο ελαστικού 65mm και πλάτος 26mm, καθώς και μια μπίλια caster για την στήριξη του ρομπότ στο εμπρός σημείο.



Εικόνα 6.6: Ρόδα κατασκευής από καουτσούκ

6.6 Κινητήρες

Οι κινητήρες είναι υπεύθυνοι για την κίνηση του ρομπότ. Συνήθως χρησιμοποιούνται κινητήρες DC, οι οποίοι προσφέρουν την απαραίτητη ροπή και ταχύτητα. Η επιλογή των κινητήρων εξαρτάται από το βάρος του ρομπότ και τις απαιτήσεις κίνησης. Για την κατασκευή χρησιμοποιήθηκαν δύο DC κινητήρες. Αυτός ο DC κινητήρας είναι μια από τις ιδανικότερες και οικονομικότερες λύσεις για ρομποτικές εφαρμογές. Διαθέτει αναλογία μετάδοσης 1:48 και παρέχει μέγιστη ροπή 800g/cm σε τάση λειτουργίας 3V, ενώ στα 3V έχει 90RPM και στα 6V έχει 200RPM. Ο κινητήρας λειτουργεί σε τάση 3V-12V DC, με ροή ρεύματος χωρίς φορτίο 150mA στα 3V ή 200mA στα 6V. Οι διαστάσεις του είναι 70mm μήκος, 20mm πλάτος, και ζυγίζει 26g. Οι άξονες έχουν διάμετρο 5mm, πλάτος 3,5mm και μήκος 9mm, καθιστώντας τον μια ευέλικτη επιλογή για πολλές εφαρμογές [11].



Εικόνα 6.7: DC Κινητήρας 3-12V

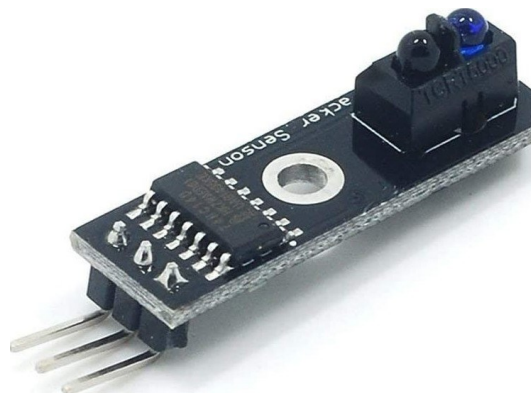
6.7 Αισθητήρες

Οι αισθητήρες είναι τα μάτια και τα αυτιά του ρομπότ. Στην κατασκευή χρησιμοποιήθηκαν αισθητήρες υπερήχων και υπέρυθρων. Αυτοί οι αισθητήρες χρησιμοποιούνται για να εντοπίζουν την παρουσία άλλων ρομπότ, να ανιχνεύουν το άκρο του αγωνιστικού πεδίου και να προσανατολίζουν το ρομπότ στο χώρο. Χρησιμοποιήθηκαν δύο αισθητήρες TCRT5000 (3 pins) καθώς ένας αισθητήρας υπερήχων HC-SR04 (4 pins).

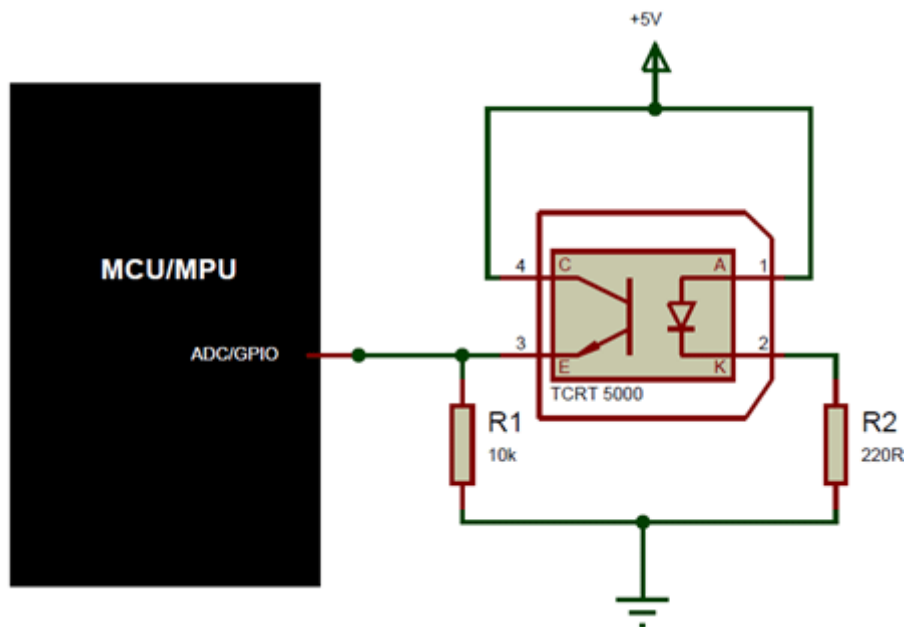
6.7.1 TCRT5000

Ο TCRT5000 είναι ένας αισθητήρας υπέρυθρων που χρησιμοποιείται κυρίως για ανίχνευση αντικειμένων και γραμμών, με βασικά χαρακτηριστικά που τον καθιστούν ευέλικτο και αξιόπιστο. Ο αισθητήρας περιλαμβάνει έναν εκπομπό υπέρυθρων και ένα φωτοτρανζίστορ σε ένα πακέτο με προστασία από το ορατό φως. Λειτουργεί σε τάση τροφοδοσίας 3.3V έως 5V και έχει ρεύμα λειτουργίας περίπου 10mA. Ο αισθητήρας διαθέτει τυπική απόσταση ανίχνευσης από 2 mm έως 15 mm.

Το φωτοτρανζίστορ δέκτης έχει χρόνο απόκρισης μικρότερο από 100 μ s, επιτρέποντας γρήγορη ανίχνευση και υψηλή ακρίβεια. Το μήκος κύματος των υπέρυθρων είναι περίπου 950nm. Επιπλέον, ο TCRT5000 έχει γωνία προβολής 15°, καθιστώντας τον κατάλληλο για εστιασμένες ανιχνεύσεις σε μικρές αποστάσεις. Το μέγεθος του αισθητηρίου είναι 10.2mm \times 5.8mm \times 7.0mm και διαθέτει τρεις ακίδες: VCC (τροφοδοσία), OUT (Output) και GND (γείωση) [12].



Εικόνα 6.8: TCRT5000



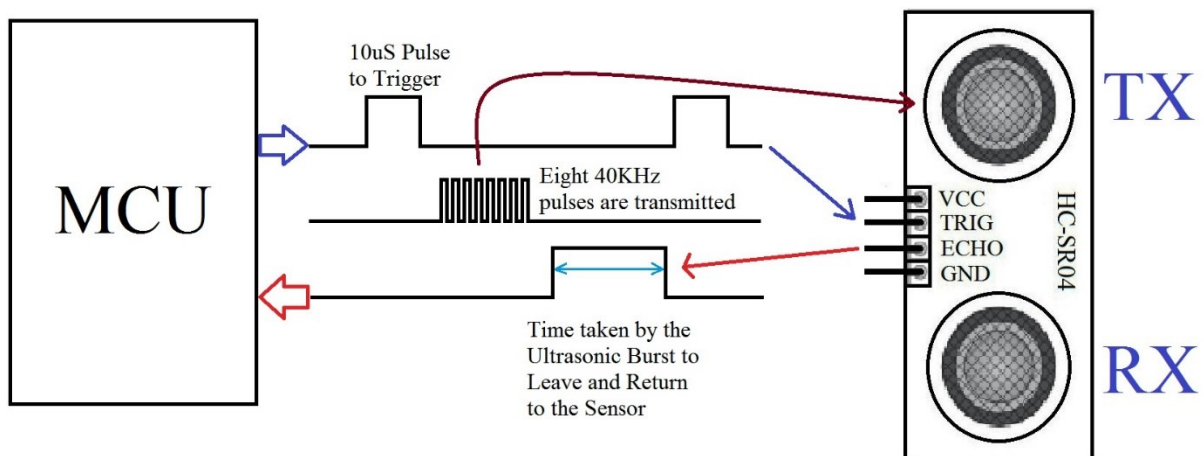
Εικόνα 6.9: TCRT5000 Circuit Diagram

6.7.2 HC-SR04

Ο HC-SR04 είναι ένας υπερηχητικός αισθητήρας απόστασης, ευρέως χρησιμοποιούμενος για τη μέτρηση αποστάσεων σε εφαρμογές ρομποτικής και αυτοματισμού. Λειτουργεί με την εκπομπή υπερηχητικών κυμάτων και τη μέτρηση του χρόνου που χρειάζεται για να επιστρέψουν τα κύματα μετά την ανάκλασή τους από ένα αντικείμενο. Η μέγιστη απόσταση ανίχνευσης είναι 4 m ενώ η ελάχιστη απόσταση ανίχνευσης είναι 2 cm. Η συχνότητα υπερήχων είναι στα 40kHz. Λειτουργεί σε τάση 5V DC και έχει κατανάλωση ρεύματος περίπου 15 mA. Η γωνία ανίχνευσης είναι 15°, κατάλληλη για ακριβείς μετρήσεις (± 3 mm). Απαιτεί παλμό Trigger πλάτους τουλάχιστον 10 μ s για την έναρξη της μέτρησης, ο χρόνος απόκρισης του Echo εξαρτάται από την απόσταση του αντικειμένου και υπολογίζεται με βάση την ταχύτητα του ήχου (340 m/s). Οι διαστάσεις του είναι 45 mm \times 20 mm \times 15 mm και διαθέτει τέσσερις ακίδες: VCC (τροφοδοσία), Trig (Trigger), Echo (Echo) και GND (γείωση) [13].

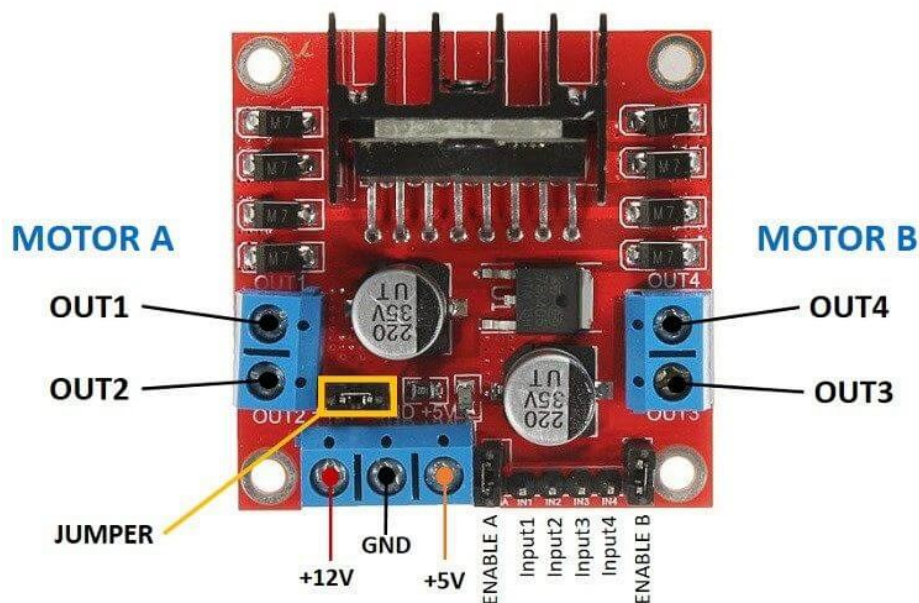


Εικόνα 6.10: HC-SR04



Εικόνα 6.11: Τρόπος λειτουργίας HC-SR04

6.8 Motor Driver L298N



Εικόνα 6.12: L298N (PinOut)

Ο L298N είναι ένας διπλός full-bridge οδηγός κινητήρα, σχεδιασμένος για να ελέγχει επαγωγικά φορτία όπως κινητήρες συνεχούς ρεύματος (DC) και βηματικούς κινητήρες. Μπορεί να λειτουργεί με τάση τροφοδοσίας από 2,5V έως 46V και να υποστηρίζει συνολικό ρεύμα έως 4A, ενώ ενσωματώνει προστασία από υπερθέρμανση, η οποία ενεργοποιείται στους 130°C για την αποφυγή βλαβών. Υποστηρίζει τη χρήση PWM (Pulse Width Modulation) για τον έλεγχο της ταχύτητας των κινητήρων στους ακροδέκτες ENA και ENB. Με τη σωστή ρύθμιση των ακροδεκτών IN1, IN2, IN3, και IN4, οι κινητήρες μπορούν να κινηθούν προς τα εμπρός, προς τα πίσω ή να σταματήσουν. Στην παρούσα εργασία χρησιμοποιήθηκε ένα τεμάχιο L298N ο οποίος οδηγεί δύο κινητήρες. Η τάση τροφοδοσίας του είναι στα 8V αλλά λόγω της πτώσης τάσης στα τρανζίστορ της γέφυρας του κάθε κινητήρα έχουμε μέγιστη τάση 6.6V στα άκρα τους. Επίσης διαθέτει έναν ενσωματωμένο ρυθμιστή τάσης στα 5V, ο οποίος χρησιμοποιήθηκε για να τροφοδοτεί το Arduino. Ο βασικότερος παράγοντας επιλογής αυτού του οδηγού είναι η τιμή του [15].

Ακολουθούν δύο πίνακες που δείχνουν τους διάφορους τρόπους λειτουργίας:

ENA	IN1	IN2	Περιγραφή
0	N/A	N/A	Απενεργοποίηση
1	0	0	Παύση
1	0	1	Όπισθεν

1	1	0	Εμπρός
1	1	1	Παύση

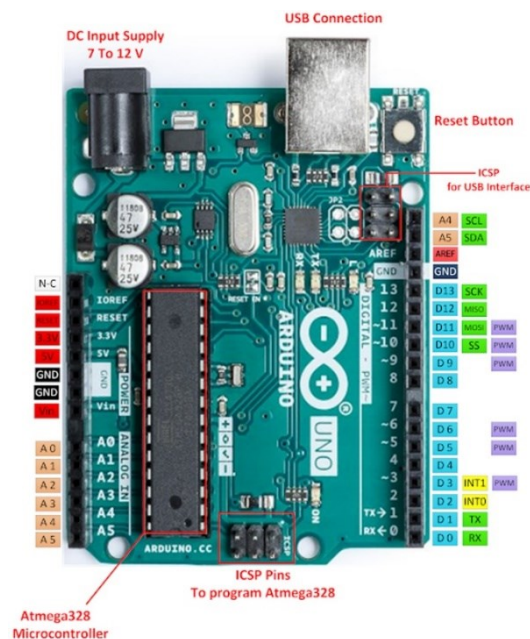
Πίνακας 6.1: Πίνακας αληθείας κινητήρα A

ENB	IN3	IN4	Περιγραφή
0	N/A	N/A	Απενεργοποίηση
1	0	0	Παύση
1	0	1	Όπισθεν
1	1	0	Εμπρός
1	1	1	Παύση

Πίνακας 6.2: Πίνακας αληθείας κινητήρα B

6.9 Μικροελεγκτής

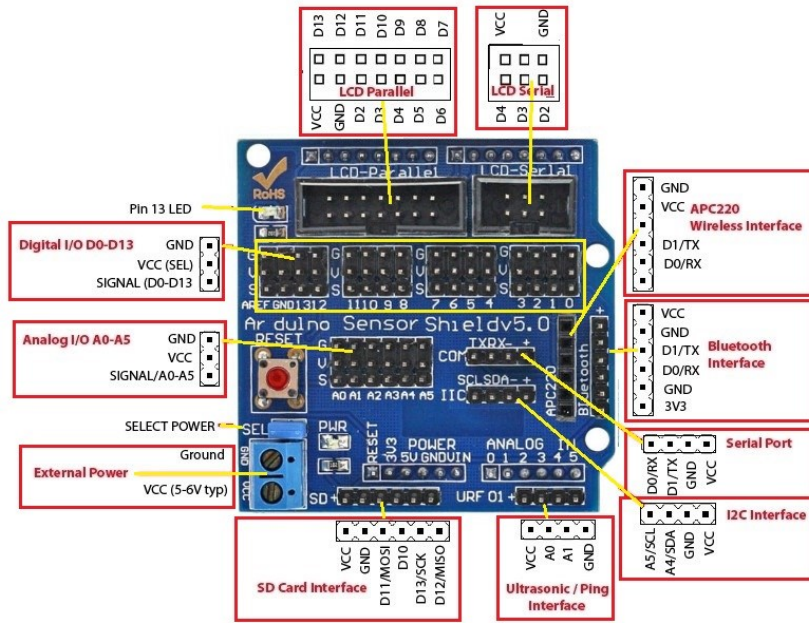
Το Arduino Uno, το οποίο αναλύθηκε λεπτομερώς στο προηγούμενο κεφάλαιο, χρησιμοποιείται ως ο κεντρικός μικροελεγκτής της παρούσας εργασίας. Η επιλογή του έγινε λόγω της ευελιξίας και των δυνατοτήτων του να διαχειρίζεται εισόδους και εξόδους με ακρίβεια, καθώς και της ευρείας υποστήριξης από την κοινότητα, διευκολύνοντας την ανάπτυξη και υλοποίηση των απαραίτητων λειτουργιών της εργασίας. Επίσης ένας από τους βασικούς παράγοντες επιλογής αυτής την πλακέτας ήταν η δυνατότητα χρήσης του Sensor Shield V5 το οποίο κάνει πιο εύκολη την συνδεσμολογία των εξαρτημάτων.



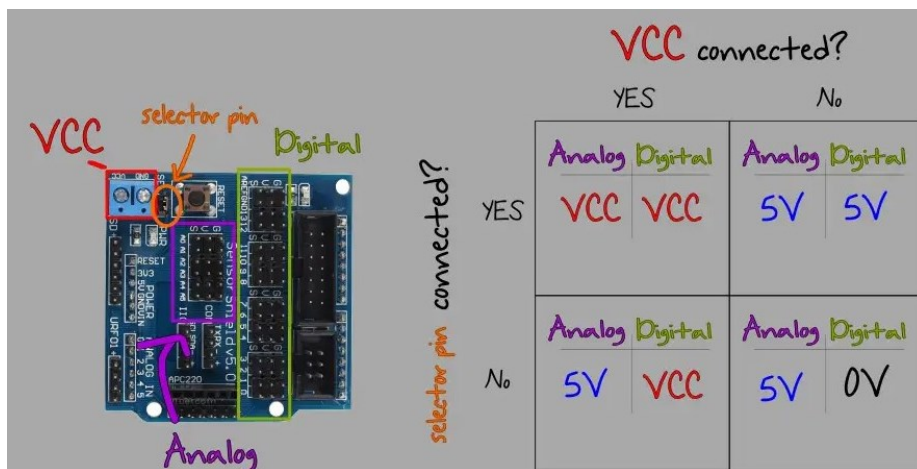
Εικόνα 6.13: Arduino Uno (PinOut)

6.10 Shield

Για την ευκολότερη σύνδεση και διαχείριση των διαφόρων αισθητήρων χρησιμοποιήθηκε το Sensor Shield V5, το οποίο είναι μια επέκταση για το Arduino Uno. Προσφέρει ευκολία στη σύνδεση των I/O pins μέσω τυποποιημένων υποδοχών που περιλαμβάνουν ξεχωριστές συνδέσεις για γείωση και τροφοδοσία, επιτρέποντας την ασφαλή και γρήγορη σύνδεση αισθητήρων. Επιπλέον, το Sensor Shield V5 περιλαμβάνει θύρες για τη σύνδεση σερβοκινητήρων και επικοινωνία μέσω πρωτοκόλλων I2C και UART, καθιστώντας το μια ιδανική επιλογή για έργα ρομποτικής και αυτοματισμού. Επίσης με την βοήθεια του select power pin μπορεί να διαμορφωθεί η τάση των pins του Shield (Εικόνα: 6.15). Στην παρούσα εργασία το Arduino τροφοδοτείται μέσω του Shield, το οποίο παίρνει σαν είσοδο 5V τα οποία παρέχει το L298N Motor Driver [14].



Εικόνα 6.14: Sensor Shield V5 (PinOut)



Εικόνα 6.15: Shield Voltage Output Diagram

6.11 Πηγή ενέργειας

Για την παροχή ενέργειας, επιλέχθηκαν μπαταρίες 18650 λόγω της υψηλής ενεργειακής τους πυκνότητας και της αξιοπιστίας τους, που είναι κρίσιμες για τις απαιτήσεις ισχύος και απόδοσης του ρομπότ. Οι μπαταρίες 18650 χρησιμοποιούνται ευρέως σε εφαρμογές όπου απαιτείται μεγάλη χωρητικότητα και σταθερή παροχή ενέργειας. Με τάση περίπου 3.7V και χωρητικότητα που κυμαίνεται συνήθως από 2000mAh έως 3500mAh, οι μπαταρίες αυτές παρέχουν αρκετή ισχύ για την κίνηση και τη λειτουργία των κινητήρων και των αισθητήρων του ρομπότ για παρατεταμένες χρονικές περιόδους. Επιπλέον, οι μπαταρίες 18650 χαρακτηρίζονται από μακροχρόνια ζωή και μεγάλη αντοχή σε πολλαπλούς κύκλους φόρτισης και εκφόρτισης, διασφαλίζοντας την αξιοπιστία του ρομπότ κατά τη διάρκεια των αγώνων. Για την συγκεκριμένη εργασία χρησιμοποιήθηκαν 2 τεμάχια Samsung 18650 στα 2500mAh. Η φόρτιση τους έγινε από ειδικό φορτιστή για τον συγκεκριμένο τύπο μπαταρίας ο οποίος στο τέλος της φόρτισης δείχνει και την χωρητικότητα της μπαταρίας η οποία ήταν στα 1800mAh. Για την σύνδεση των μπαταριών και την ενσωμάτωσή τους στο ρομπότ χρησιμοποιήθηκε μια μπαταριοθήκη με 2 υποδοχές η οποία πραγματοποιεί σύνδεση σειρά. Επίσης μεταξύ των μπαταριών και του οδηγού κινητήρων τοποθετήθηκε ένας διακόπτης για την εύκολη ενεργοποίηση / απενεργοποίηση του ρομπότ.



Εικόνα 6.16: 18650 Μπαταρία



Εικόνα 6.17: 18650 Μπαταριοθήκη



Εικόνα 6.18: Διακόπτης

6.12 Επίλογος

Συμπερασματικά, στο κεφάλαιο 6 παρουσιάστηκαν οι επιλογές των υλικών και εξαρτημάτων που χρησιμοποιήθηκαν για την κατασκευή του ρομπότ, τονίζοντας τη σημασία της προσεκτικής επιλογής για την επίτευξη των στόχων του έργου. Έγινε αναφορά στους βραχυκυκλωτήρες, ταινίες διπλής όψευς και κόλλες, το πλαίσιο, το Dohyo, τα ελαστικά, τους κινητήρες, τους αισθητήρες, τον οδηγό κινητήρων L298N, το Arduino Uno και το αντίστοιχα shield, καθώς και στις μπαταρίες 18650. Κάθε επιλογή υλικού έγινε με βάση την τιμή, αντοχή, τη λειτουργικότητα και την αποδοτικότητά τους, προκειμένου να διασφαλιστεί η αξιόπιστη και αποδοτική λειτουργία του ρομπότ κατά τη διάρκεια των αγώνων.

Κεφάλαιο 7^ο: Κατασκευή

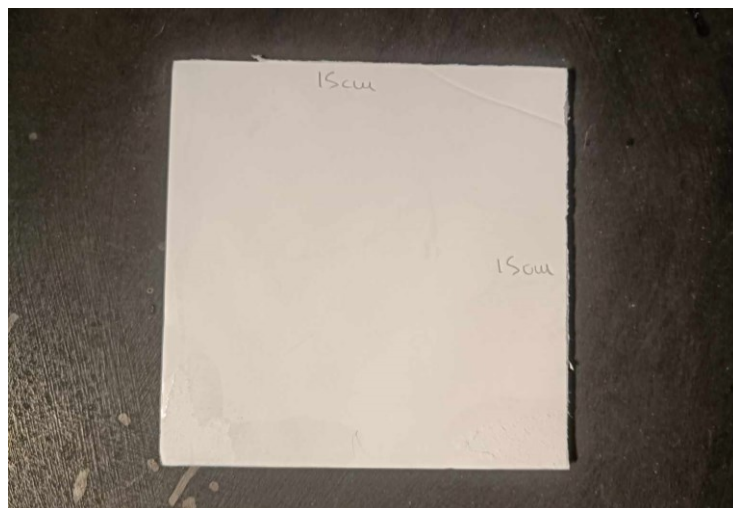
7.1 Περιγραφή λειτουργίας

Η κατασκευή που αναπτύχθηκε στο πλαίσιο αυτής της διπλωματικής εργασίας αποτελεί ένα αυτόνομο Sumo Robot, σχεδιασμένο να συμμετέχει σε αγώνες ρομποτικής με στόχο την εκτόπιση του αντιπάλου από έναν κυκλικό αγωνιστικό χώρο. Το ρομπότ είναι εξοπλισμένο με αισθητήρες για την ανίχνευση του αντιπάλου και των ορίων του Dohyo (ρινγκ), καθώς και με κινητήρες που του επιτρέπουν να εκτελεί γρήγορες κινήσεις.

7.2 Εξέλιξη της κατασκευής του Sumo Robot

Σε αυτό το σημείο θα αναφερθούν όλα τα βήματα για την υλοποίηση της κατασκευής.

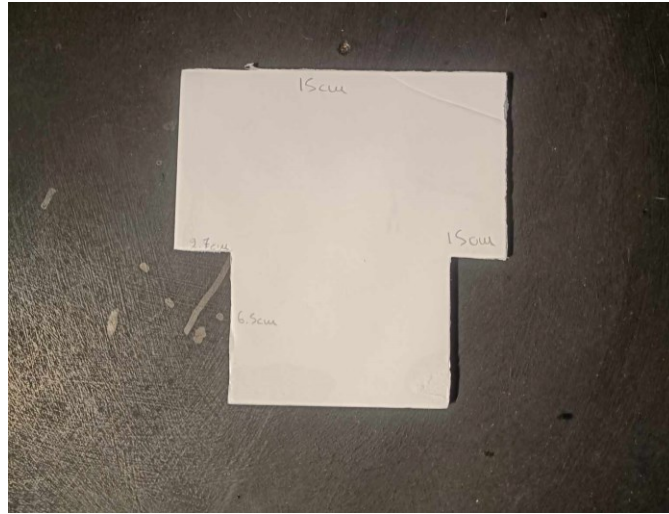
Βήμα 1: Αρχικά επιλέχθηκε η κατηγορία στην οποία θα αγωνιστεί το ρομπότ με σκοπό να τηρηθούν οι προδιαγραφές. Η κατηγορία αυτή είναι η Kit Sumo, στην οποία τα ρομπότ πρέπει να έχουν 15 εκατοστά μήκος και πλάτος καθώς και βάρος μεγαλύτερο των 500 γραμμαρίων, μικρότερο των 1000 γραμμαρίων. Στο ύψος δεν υπάρχει κάποιος περιορισμός. Σύμφωνα με τις διαστάσεις αυτής της κατηγορίας κόπηκε το αντίστοιχο κομμάτι από το μακετόχαρτο με σκοπό την δημιουργία πλαισίου.



Εικόνα 7.1: Κόψιμο μακετόχαρτου πλαισίου

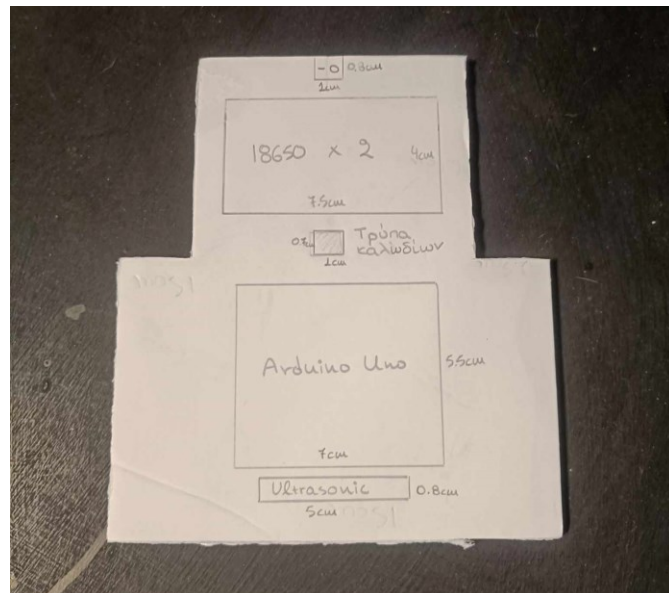
Κεφάλαιο 7

Βήμα 2: Στην συνέχεια για να τοποθετηθούν οι κινητήρες με τα ελαστικά έπρεπε να κοπεί το μακετόχαρτο έτσι ώστε οι ρόδες να μην προεξέχουν. Αφαιρέθηκε από κάθε πλευρά ένα κομμάτι 6.5cm x 2.7cm.

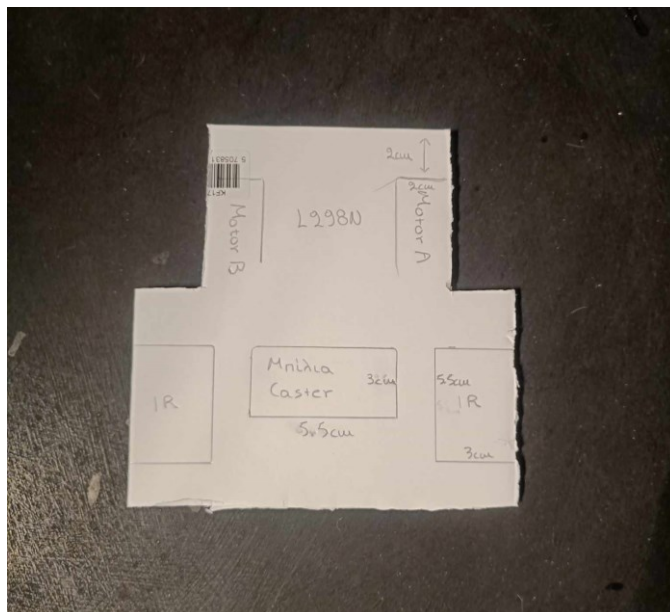


Εικόνα 7.2: Τροποποίηση πλαισίου

Βήμα 3: Έπειτα πραγματοποιήθηκε σχεδιασμός στο μακετόχαρτο για να δούμε που θα μπει το κάθε εξάρτημα.

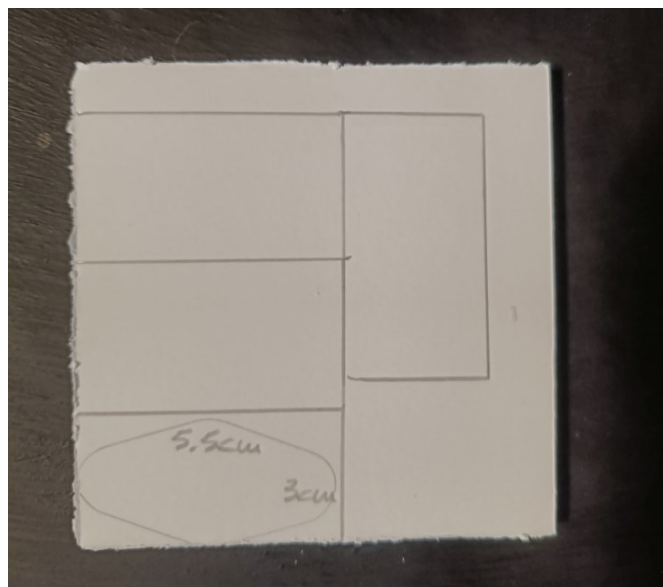


Εικόνα 7.3: Σχεδιασμός εξαρτημάτων (πάνω μέρος)



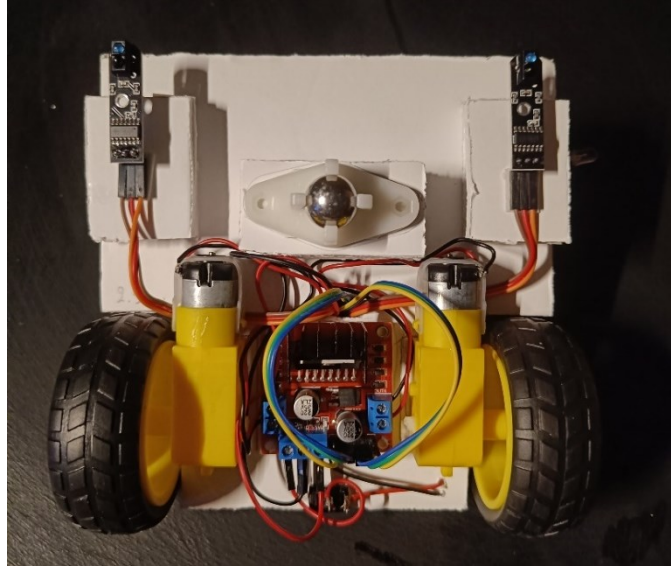
Εικόνα 7.4: Σχεδιασμός εξαρτημάτων (κάτω μέρος)

Βήμα 4: Στην συνέχεια κόπηκαν 4 κομμάτια μακετόχαρτου διαστάσεων 6cm x 2.5cm για να δημιουργηθεί μια βάση για την μπίλια caster. Επίσης κόπηκαν ακόμη 4 κομμάτια ίδιων διαστάσεων για να μειώσουν την απόσταση από το έδαφος στους IR αισθητήρες.



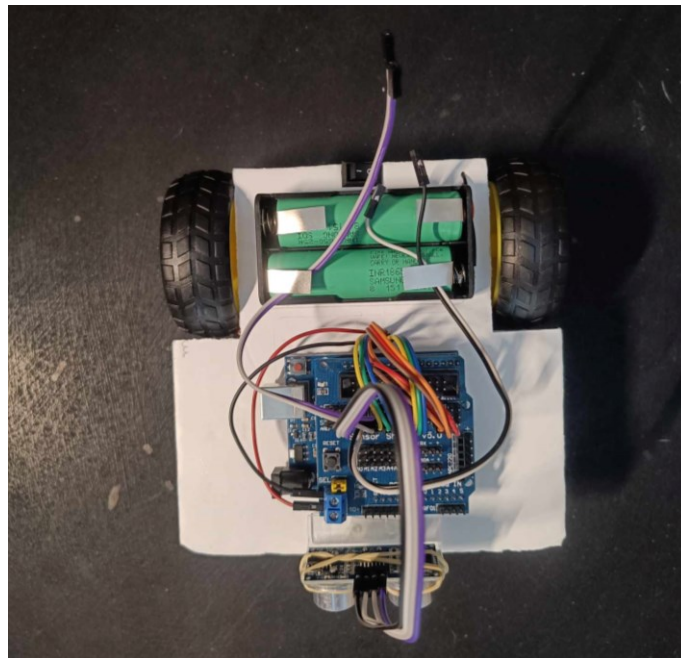
Εικόνα 7.5: Βάση μπίλιας / IR

Βήμα 5: Ακολούθως προστέθηκαν τα κομμάτια μακετόχαρτου στα σημεία που χρειαζόταν στο κάτω μέρος του πλαισίου, καθώς και τα εξαρτήματα για τα οποία οι θέσεις σχεδιάστηκαν στον βήμα 3. Επίσης έγινε η σύνδεση των κινητήρων με τον οδηγό κινητήρων και ανοίχτηκε οι τρύπα των καλωδίων. Όλα τα εξαρτήματα κολλήθηκαν με αφρώδεις ταινία διπλής όψης και σε σημεία μεγαλύτερης πίεσης όπως κινητήρες χρησιμοποιήθηκε και UHU Super Glue. Οι κινητήρες καθώς και ο διακόπτης χρειάστηκαν για την ένωση των καλωδίων καλά.



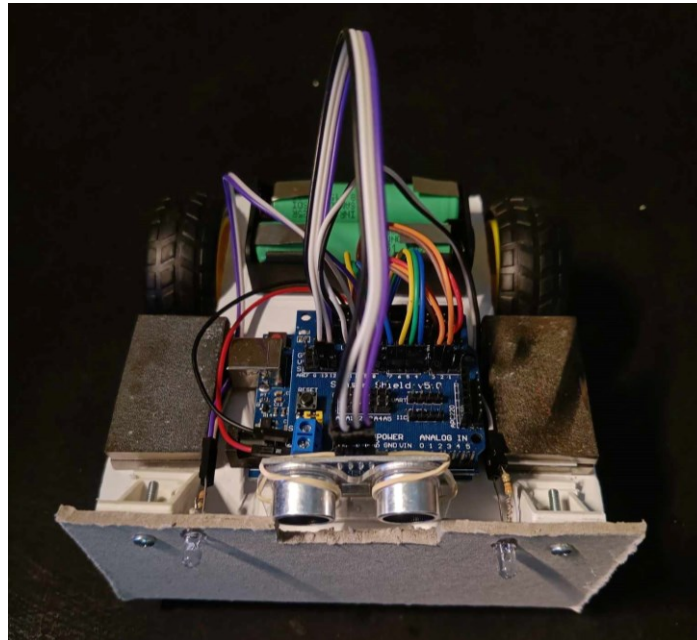
Εικόνα 7.6: Ολοκλήρωση κάτω μέρος πλαισίου

Βήμα 6: Προστέθηκαν όλα τα εξαρτήματα με χρήση αφρώδεις ταινία διπλής όψης στο πάνω μέρος και έγιναν όλες οι συνδέσεις των καλωδίων εκτός των αναμονών LED ένδειξης ανίχνευσης γραμμής.

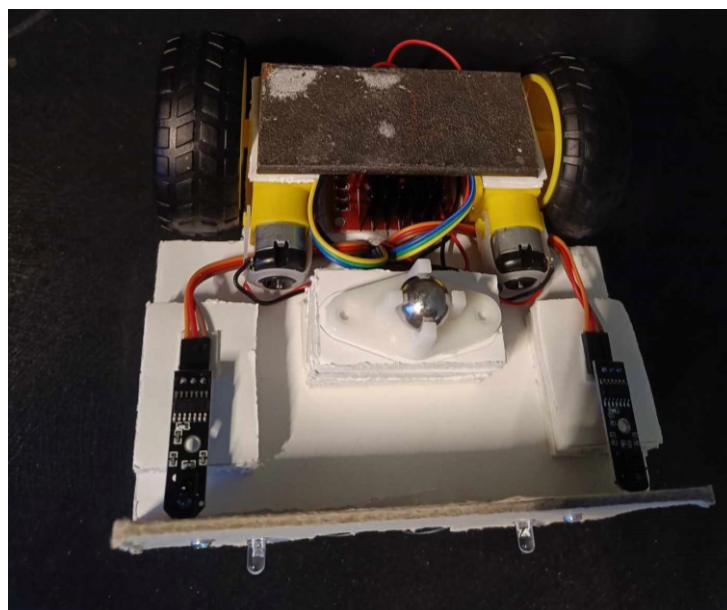


Εικόνα 7.7: Ολοκλήρωση άνω μέρος πλαισίου

Βήμα 7: Σχεδιάστηκε και τοποθετήθηκε ο μπροστινός προφυλακτήρας διαστάσεων 6cm x 15cm στον οποίο τοποθετήθηκαν LED τα οποία μας δείχνουν πότε ενεργοποιούνται οι IR αισθητήρες. Στην άνοδο των LED τοποθετήθηκαν από μια αντίσταση των 330Ω ως περιοριστής ρεύματος. Επίσης τοποθετήθηκαν μερικά κομμάτια σίδερο στο πάνω και κάτω μέρος για να αυξήσουν το βάρος της κατασκευής με σκοπό να βρίσκεται στα επιτρεπτά όρια. Στα κομμάτια σιδήρου της άνω επιφάνειας από την μεριά που το Arduino έχει την θύρα USB χρησιμοποιήθηκε Velcro ταινία για την εύκολη πρόσβαση στην θύρα.



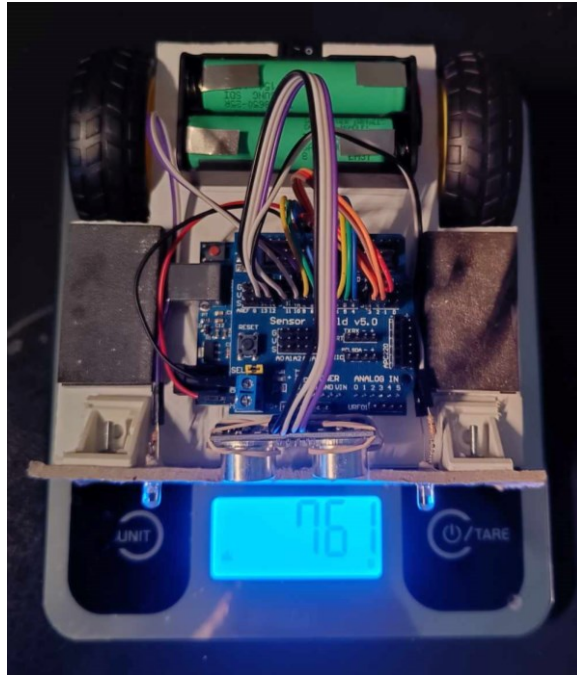
Εικόνα 7.8: Προσθήκη βάρους / προφυλακτήρα στο άνω μέρος της κατασκευής



Εικόνα 7.9: Προσθήκη βάρους στο κάτω μέρος της κατασκευής

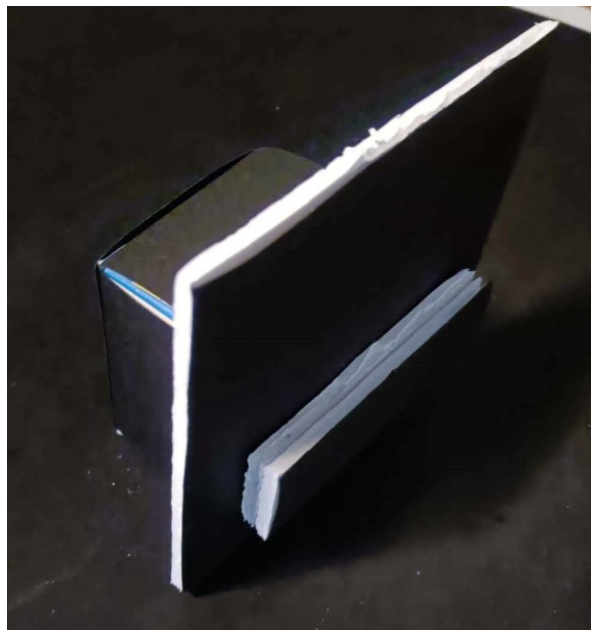
Κεφάλαιο 7

Στην συνέχεια μετρήθηκε το βάρος της κατασκευής από ζυγαριά με ακρίβεια 1 γραμμαρίου. Το βάρος του ρομπότ είναι στα 761 γραμμάρια.



Εικόνα 7.10: Βάρος κατασκευής

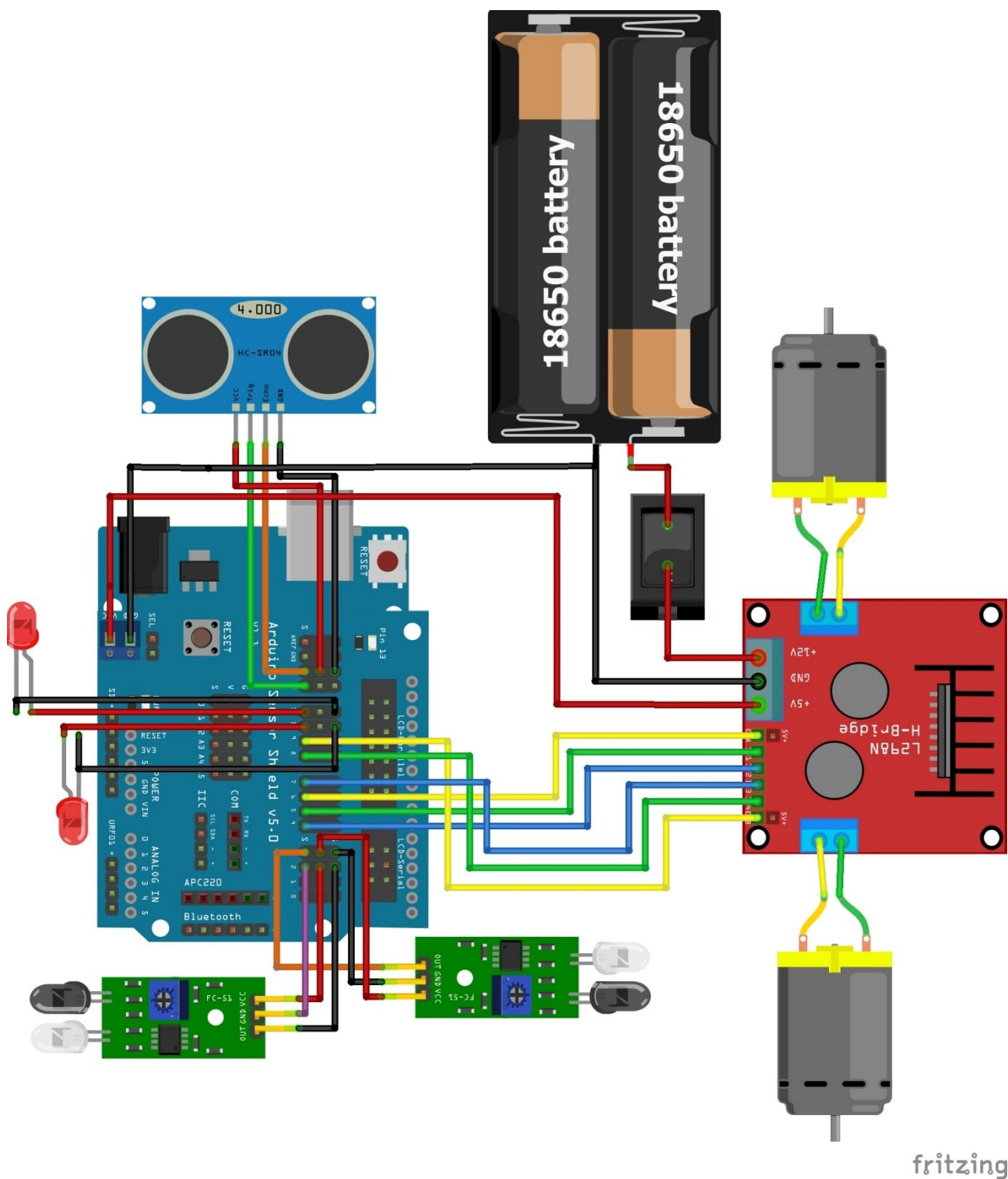
Βήμα 9: Δημιουργήθηκε ένα αντικείμενο για τον ρόλο του εχθρού από μακετόχαρτο διαστάσεων 14cm x 15cm, και έναν χάρτινο κύβο για βάση με διαστάσεις M7.5cm x Π6cm x Υ8.5cm, ενώ ο μπροστινός προφυλακτήρας από μακετόχαρτο διαστάσεων 3.5cm x 11cm . Στις επιφάνειες προστέθηκε μαύρο χαρτόνι για διατήρηση της ομοιομορφίας. Η ένωση των υλικών έγινε με αφρώδεις ταινία διπλής όψης.



Εικόνα 7.11: Εχθρός

7.3 Διάγραμμα καλωδιώσεων

Παρακάτω δίνεται το διάγραμμα των καλωδιώσεων της κατασκευής. Σε αυτό το διάγραμμα φαίνεται η συνδεσμολογία όλων των στοιχείων.



Εικόνα 7.12: Διάγραμμα καλωδιώσεων

7.4 Κατασκευή του Sumo Ring (Dohyo)

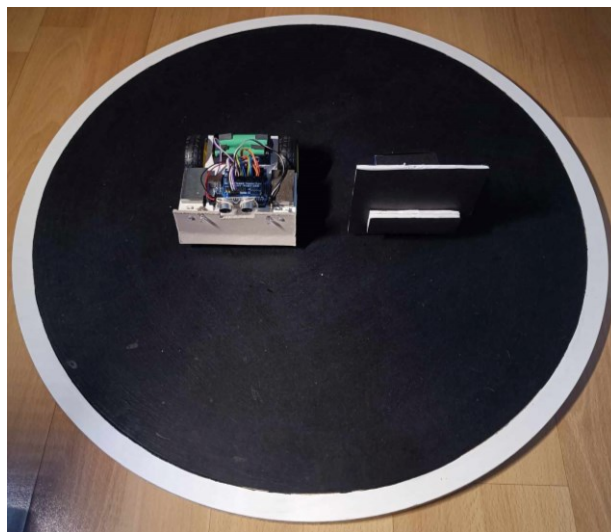
Όπως αναφέρθηκε στο κεφάλαιο 6 επιλέχθηκε σαν υλικό κατασκευής η μελαμίνη για το βασικό πλεονέκτημα του άσπρου χρώματος που είναι εξοπλισμένη και της ανθεκτικότητας της. Το πάνελ που αγοράστηκε είχε διαστάσεις 1m x 1m και κόπηκε σε έναν κύκλο διαμέτρου 77cm από τα οποία τα 2.5cm είναι η άσπρη οριογραμμή όπως ορίζουν οι προδιαγραφές για την κατασκευή του ρινγκ σε αυτή την κατηγορία. Ο εσωτερικός κυκλικός χώρος του ρινγκ βάφτηκε με μαύρο matte πλαστικό χρώμα, αυτή θα είναι η επιφάνεια που θα κινείται το ρομπότ.



Εικόνα 7.13: Sumo Ring (Dohyo)

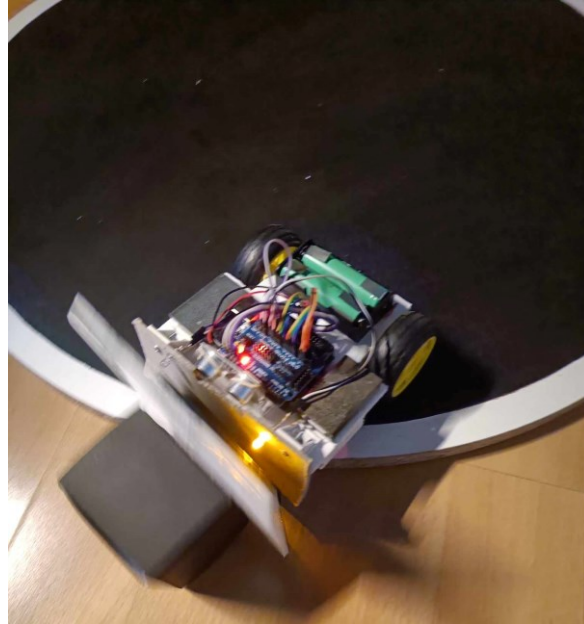
7.5 Τελικό αποτέλεσμα

Η παρακάτω φωτογραφία απεικονίζει το ολοκληρωμένο σύστημα του Sumo Robot, μαζί με τον υποτιθέμενο αντίπαλο και το ρινγκ μάχης (Dohyo). Στη φάση αυτή, όλα τα μέρη του έργου έχουν συνδυαστεί και λειτουργούν όπως σχεδιάστηκαν, έτοιμα για να διεξαχθούν οι προγραμματισμένες δοκιμές και αναμετρήσεις.



Εικόνα 7.14: Τελικό αποτέλεσμα

Κατά τη διάρκεια της αναμέτρησης, το Sumo Robot έπειτα της εφαρμογής της στρατηγικής του που περιλαμβάνει περιστροφή και ανίχνευση του εχθρού, εξωθεί επιτυχώς τον εχθρό εκτός του ριγκ (Dohyo). Αυτή η στιγμή αναδεικνύει την ακρίβεια και την αποτελεσματικότητα του σχεδιασμού και της προγραμματιστικής υλοποίησης, επιβεβαιώνοντας την επιτυχή ολοκλήρωση των στόχων του έργου.



Εικόνα 7.15: Αναμέτρηση

7.6 Πίνακας υλικών

ΥΛΙΚΟ	ΠΟΣΟΤΗΤΑ	ΚΟΣΤΟΣ
Κλώνος Arduino Uno	1	6 €
Motor Driver L298N	1	2 €
IR Sensor TCRT5000 (3 pin)	2	3 €
DC Motors	2	3.5 €
Ultrasonic Sensor HC-SR04	1	2 €
Ultrasonic Sensor Mount	1	0.60 €
Ελαστικά	2	2.4 €
Χαρτόνι μακέτας (70 X 100) 5mm	1	8 €
Χαρτόνι γκρι (50 X 70) 3mm	1	2.6 €
UHU Super Glue	2	2.9 €
Logo foam tape	1	1.5 €

Βραχυκυκλωτήρες male to male	1	2.5 €
Βραχυκυκλωτήρες female to female	1	2.5 €
Μπαταρίες 18650	2	4 €
Θήκη 2 μπαταριών	1	1.5 €
Διακόπτης	1	0.5 €
LED	2	0.4 €
Πλαστικές γωνίες ραφιών	2	0.2 €
Βίδες M3 16mm	2	0.02 €
Παξιμάδι M3	2	0.02 €
Φύλλο μελαμίνης 1m X 1m	1	10 €
Μαύρο matte πλαστικό χρώμα 200ml	1	3.5 €
Πλάκα σίδερο 1Kg	1	2.5 €
	ΣΥΝΟΛΟ	62.14 €

Πίνακας 7.1: Πίνακας υλικών και κόστους

7.7 Επίλογος

Η κατασκευή του αυτόνομου Sumo Robot που περιγράφεται σε αυτό το κεφάλαιο αποτελεί ένα ενδιαφέρον και πρακτικό παράδειγμα εφαρμογής των αρχών της ρομποτικής σε πραγματικές συνθήκες. Το ρομπότ σχεδιάστηκε και κατασκευάστηκε λαμβάνοντας υπόψη τις απαιτήσεις και τους κανονισμούς της κατηγορίας Kit Sumo, εξασφαλίζοντας έτσι τη συμμόρφωση με τις προδιαγραφές για συμμετοχή σε αγώνες. Κατά τη διάρκεια της υλοποίησης, αντιμετωπίστηκαν και επιλύθηκαν διάφορες προκλήσεις, όπως η επιλογή των κατάλληλων υλικών και εξαρτημάτων, η σωστή τοποθέτηση τους, καθώς και η εξασφάλιση της απαιτούμενης σταθερότητας και βάρους της κατασκευής. Ο τρόπος κατασκευής και η επιλογή των υλικών βασίστηκαν στην ανάγκη για λειτουργικότητα, αντοχή, αλλά και οικονομία. Το αποτέλεσμα είναι ένα ρομπότ που ανταποκρίνεται στις απαιτήσεις ενός αγώνα Sumo, με δυνατότητες ανίχνευσης του αντιπάλου και προσαρμογής στις συνθήκες του αγωνιστικού χώρου.

Κεφάλαιο 8^ο: Περιγραφή κώδικα

Ο κώδικας αποτελείται από 6 στάδια, την κλήση βιβλιοθηκών, δημιουργία αντικειμένων, δήλωση μεταβλητών, ορισμό των συναρτήσεων `setup()` και `loop()`, καθώς και τον ορισμό των υπόλοιπων συναρτήσεων.

8.1 Κλήση βιβλιοθηκών

Αρχικά έγινε η εισαγωγή της βιβλιοθήκης `NewPing`, η οποία χρησιμοποιήθηκε για την διαχείριση και τον χειρισμό του αισθητήρα απόστασης `HC-SR04`. Ο κύριος ρόλος της βιβλιοθήκης είναι να απλοποιεί την διαδικασία μέτρησης της απόστασης, προσφέροντας έναν αξιόπιστο και αποδοτικό τρόπο για τη λήψη των δεδομένων από τον αισθητήρα. Στο πλαίσιο της ανάπτυξης του `sumo robot`, η βιβλιοθήκη `NewPing` χρησιμοποιείται για να επιτρέπει στο ρομπότ να αναγνωρίζει τον εχθρό, μετρώντας την απόσταση από αυτόν σε πραγματικό χρόνο. Η βιβλιοθήκη παρέχει τη δυνατότητα να καθοριστεί το μέγιστο εύρος απόστασης που μπορεί να ανιχνεύσει ο αισθητήρας, ενώ επιτρέπει επίσης την εκτέλεση πολλαπλών μετρήσεων σε σύντομο χρονικό διάστημα με μεγαλύτερη ακρίβεια. Ένα από τα κύρια πλεονεκτήματα της βιβλιοθήκης `NewPing` είναι η δυνατότητά της να εξοικονομεί χώρο στη μνήμη του προγράμματος, καθώς και να απλοποιεί τον κώδικα με χρήση έτοιμων συναρτήσεων.

8.2 Δήλωση μεταβλητών και δημιουργία αντικειμένων

Στην συνέχεια δηλώνονται οι αισθητήρες στα `pins` που συνδέονται στο `Arduino` καθώς και διάφορες μεταβλητές του προγράμματος. Έπειτα δημιουργήθηκε ένα αντικείμενο `sonar` της κλάσης `NewPing` μέσω του οποίου αρχικοποιείται ο υπερηχητικός αισθητήρας με τα κατάλληλα `pins` καθώς και η επιθυμητή μέγιστη απόσταση. Στην συνέχεια του κώδικα χρησιμοποιείται το αντικείμενο `sonar` για να κληθεί η λειτουργία μέτρησης απόστασης `ping_cm()`.

```
// Ορισμός των pin ελέγχου των κινητήρων και των αισθητήρων
const int ENA = 6; // Έλεγχος ταχύτητας του δεξιού κινητήρα
const int IN1 = 5; // Κατεύθυνση δεξιού κινητήρα (προς τα εμπρός)
const int IN2 = 4; // Κατεύθυνση δεξιού κινητήρα (προς τα πίσω)
const int IN3 = 7; // Κατεύθυνση αριστερού κινητήρα (προς τα εμπρός)
const int IN4 = 8; // Κατεύθυνση αριστερού κινητήρα (προς τα πίσω)
const int ENB = 9; // Έλεγχος ταχύτητας του αριστερού κινητήρα

const int IRRightSensor = 3; // Αισθητήρας υπερύθρων στα δεξιά
const int IRLeftSensor = 2; // Αισθητήρας υπερύθρων στα αριστερά

const int LED_PIN_RIGHT = 10; // LED για ένδειξη ενεργοποίησης δεξιού αισθητήρα IR
const int LED_PIN_LEFT = 11; // LED για ένδειξη ενεργοποίησης αριστερού αισθητήρα IR

const int trigPin = 12; // Pin για το Trigger του υπερηχητικού αισθητήρα
const int echoPin = 13; // Pin για το Echo του υπερηχητικού αισθητήρα

volatile int rightSensorState = LOW; // Κατάσταση του δεξιού αισθητήρα IR
volatile int leftSensorState = LOW; // Κατάσταση του αριστερού αισθητήρα IR

volatile unsigned int lastInterruptTime = 0; // Χρόνος του τελευταίου interrupt
const unsigned int debounceDelay = 600; // Χρόνος καθυστέρησης σε ms για την αποφυγή εκτέλεσης συνεχόμενων interrupt

volatile bool objectDetected = false; // Σημεία ανίχνευσης αντικειμένου

const int MAX_DISTANCE = 35; // Μέγιστη απόσταση ανίχνευσης υπερηχητικού αισθητήρα
NewPing sonar(trigPin, echoPin, MAX_DISTANCE); // Δημιουργία αντικειμένου υπερηχητικού αισθητήρα
```

Εικόνα 8.1: Δήλωση μεταβλητών και δημιουργία αντικειμένων

8.3 Η συνάρτηση setup()

Η συνάρτηση setup() εκτελείται μετά από κάθε ενεργοποίηση ή επανεκκίνηση του Arduino, εκτελείται μόνο μια φορά και χρησιμοποιείται για να προετοιμάσει το σύστημα για τη λειτουργία του. Ορίζει τα pins για εισόδους / εξόδους, ενώ κάνει και ρύθμιση των διακοπών (interrupts) για άμεση απόκριση σε σήματα από τους υπέρυθρους αισθητήρες.

```
void setup() {
  // Ρυθμίσεις των pin ως έξοδοι ή εισοδοι
  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(IRRightSensor, INPUT);
  pinMode(IRLeftSensor, INPUT);

  pinMode(LED_PIN_RIGHT, OUTPUT);
  pinMode(LED_PIN_LEFT, OUTPUT);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Ρύθμιση των interrupts για τους αισθητήρες υπέρυθρων
  attachInterrupt(digitalPinToInterrupt(IRRightSensor), rightSensorISR, RISING);
  attachInterrupt(digitalPinToInterrupt(IRLeftSensor), leftSensorISR, RISING);
}
```

Εικόνα 8.2: Setup()

8.4 Η συνάρτηση loop()

Μετά την ολοκλήρωση εκτέλεσης της setup() ακολουθεί η συνάρτηση loop(), η οποία εκτελείται συνεχώς και αδιάκοπα, επαναλαμβάνοντας την εκτέλεση των εντολών που περιέχει, για όσο διάστημα η συσκευή είναι ενεργοποιημένη.

```
void loop() {
  checkultrasonic(); // Έλεγχος του υπερηχητικού αισθητήρα
  // Έλεγχος κατάστασης του δεξιού αισθητήρα
  if (rightSensorState) {
    digitalWrite(LED_PIN_RIGHT, HIGH); // Άναμμα δεξιού LED
    s(100); // Σταμάτημα
    b(800); // Κίνηση προς τα πίσω
    s(100); // Σταμάτημα
    l(350); // Στροφή αριστερά
    rightSensorState = false; // Επαναφορά κατάστασης δεξιού αισθητήρα
    digitalWrite(LED_PIN_RIGHT, LOW); // Σβήσιμο δεξιού LED
  } // Έλεγχος κατάστασης του αριστερού αισθητήρα
  else if (leftSensorState) {
    digitalWrite(LED_PIN_LEFT, HIGH); // Άναμμα αριστερού LED
    s(100); // Σταμάτημα
    b(800); // Κίνηση προς τα πίσω
    s(100); // Σταμάτημα
    r(350); // Στροφή δεξιά
    leftSensorState = false; // Επαναφορά κατάστασης αριστερού αισθητήρα
    digitalWrite(LED_PIN_LEFT, LOW); // Σβήσιμο αριστερού LED
  } // Έλεγχος αν ανιχνεύεται αντικείμενο μπροστά
  else if (objectDetected) {
    f(5); // Κίνηση προς τα μπροστά
  } else {
    r(5); // Στροφή δεξιά
  }
}
```

Εικόνα 8.3: Loop()

Κατά την εκτέλεση της συνάρτησης ελέγχεται η κατάσταση όλων των αισθητήρων. Αν ένας αισθητήρας ανιχνεύσει αλλαγή, ακολουθεί μια σειρά κινήσεων αντιμετώπισης αυτής της αλλαγής. Αν ανιχνευθεί αντικείμενο από τον υπερηχητικό αισθητήρα τότε το ρομπότ κάνει κίνηση εμπρός για όσο χρόνο το ανιχνεύει. Αν ανιχνευθεί άσπρη γραμμή τότε ανάλογα με τον αισθητήρα που την ανίχνευσε εκτελούνται μια σειρά κινήσεων προς την αντίθετη κατεύθυνση για την αποφυγή εξόδου του ρομπότ από το dohyo. Αν δεν συμβεί κάποια ανίχνευση εχθρού ή άσπρης γραμμής, το ρομπότ εκτελεί περιστροφή προς τα δεξιά, ενώ ταυτόχρονα ανιχνεύει τον χώρο για εχθρούς.

8.5 Υπόλοιπες συναρτήσεις

Με σκοπό την αποφυγή μεγάλου όγκου της συνάρτησης loop(), δημιουργήθηκαν συναρτήσεις έξω από αυτή, οι οποίες καλούνται μέσα σε αυτή. Ακολουθούν αυτές οι συναρτήσεις. Το μεγαλύτερο πλήθος συναρτήσεων έχουν οι συναρτήσεις κίνησης, οι οποίες καθορίζουν την κατεύθυνση στην οποία θα κινηθεί το ρομπότ. Τα pins IN1-IN4 καθορίζουν την φορά των ελαστικών ενώ το ENA και ENB την ταχύτητα κίνησης, της οποίας μέγιστη τιμή είναι το 255. Κατά την κλήση τους επιλέγεται σαν παράμετρος και ο χρόνος εκτέλεσης τους σε millisecond.

```
// Συνάρτηση για κίνηση προς τα μπροστά
void f(int d) {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  analogWrite(ENA, 150); // Ταχύτητα δεξιού κινητήρα
  analogWrite(ENB, 150); // Ταχύτητα αριστερού κινητήρα
  delay(d);
}

// Συνάρτηση για κίνηση προς τα πίσω
void b(int d) {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  analogWrite(ENA, 110); // Ταχύτητα δεξιού κινητήρα
  analogWrite(ENB, 110); // Ταχύτητα αριστερού κινητήρα
  delay(d);
}

// Συνάρτηση για στροφή δεξιά
void r(int d) {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  analogWrite(ENA, 100); // Ταχύτητα δεξιού κινητήρα
  analogWrite(ENB, 100); // Ταχύτητα αριστερού κινητήρα
  delay(d);
}

// Συνάρτηση για στροφή αριστερά
void l(int d) {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  analogWrite(ENA, 100); // Ταχύτητα δεξιού κινητήρα
  analogWrite(ENB, 100); // Ταχύτητα αριστερού κινητήρα
  delay(d);
}

// Συνάρτηση για σταμάτημα
void s(int d) {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
  analogWrite(ENA, 0); // Σταμάτημα δεξιού κινητήρα
  analogWrite(ENB, 0); // Σταμάτημα αριστερού κινητήρα
  delay(d);
}
```

Εικόνα 8.4: Συναρτήσεις κίνησης

Επίσης δημιουργήθηκαν δύο συναρτήσεις ISR (Interrupt Service Routine) που σχετίζονται με τους αισθητήρες υπέρυθρων. Αυτές οι συναρτήσεις ενεργοποιούνται όταν ανιχνευθεί μία διακοπή (interrupt) από τους αισθητήρες. Οι διακοπές ενεργοποιούνται όταν το σήμα αλλάζει από χαμηλή (LOW) κατάσταση σε υψηλή (HIGH) λόγω του RISING κατά την ρύθμιση των διακοπών στην setup(), επιλέχθηκε αυτού του τύπου η μετάβαση λόγω των χρωμάτων του Dohyo. Επιπρόσθετα τοποθετήθηκε μια δικλίδα ασφαλείας η οποία δεν επιτρέπει την εκτέλεση της συνάρτησης αν δεν έχουν περάσει 600ms από την ενεργοποίηση του προηγούμενου interrupt.

```

// Αυτός ο χειριστής διακοπής (ISR) εκτελείται όταν ο δεξιός αισθητήρας υπερύθρων ανιχνεύει αλλαγή.
void rightSensorISR() {
    unsigned long currentTime = millis(); // Καταγραφή του τρέχοντος χρόνου

    // Ελέγχει αν έχει περάσει ο ορισμένος χρόνος debounceDelay από την προηγούμενη διακοπή
    if (currentTime - lastInterruptTime > debounceDelay) {
        rightSensorState = digitalRead(IRRightSensor); // Ενημέρωση της κατάστασης του δεξιού αισθητήρα
        lastInterruptTime = currentTime; // Ενημέρωση του χρόνου της τελευταίας διακοπής
    }
}

// Αυτός ο χειριστής διακοπής (ISR) εκτελείται όταν ο αριστερός αισθητήρας υπερύθρων ανιχνεύει αλλαγή.
void leftSensorISR() {
    unsigned long currentTime = millis(); // Καταγραφή του τρέχοντος χρόνου

    // Ελέγχει αν έχει περάσει ο ορισμένος χρόνος debounceDelay από την προηγούμενη διακοπή
    if (currentTime - lastInterruptTime > debounceDelay) {
        leftSensorState = digitalRead(IRLeftSensor); // Ενημέρωση της κατάστασης του αριστερού αισθητήρα
        lastInterruptTime = currentTime; // Ενημέρωση του χρόνου της τελευταίας διακοπής
    }
}

```

Εικόνα 8.5: Συναρτήσεις ISR

Τέλος, δημιουργήθηκε μια συνάρτηση η οποία σχετίζεται με τον έλεγχο του υπερηχητικού αισθητήρα για την ανίχνευση εχθρών ή αντικειμένων που βρίσκονται μπροστά από το ρομπότ. Κατά την εκτέλεση της συνάρτησης, γίνεται μέτρηση της απόστασης σε εκατοστά μέσω του καλέσματος της μεθόδου `ping_cm()` του αντικειμένου `sonar`, το οποίο είναι υπεύθυνο για την αποστολή υπερηχητικών παλμών και τη λήψη των ηχώ τους. Η επιστρεφόμενη τιμή αντιπροσωπεύει την απόσταση του αντικειμένου από τον αισθητήρα. Στη συνέχεια, η συνάρτηση ελέγχει εάν η μετρούμενη απόσταση βρίσκεται εντός μιας καθορισμένης εμβέλειας. Αν πληρούνται οι προϋποθέσεις, η σημαία `objectDetected` ορίζεται σε `true`, υποδεικνύοντας ότι έχει ανιχνευθεί κάποιο αντικείμενο. Αν όχι, η `objectDetected` γίνεται `false`, σηματοδοτώντας ότι δεν υπάρχει εμπόδιο στην περιοχή ανίχνευσης. Αυτή η διαδικασία επιτρέπει στο ρομπότ να αντιδρά δυναμικά στην παρουσία αντικειμένων.

```

void checkUltrasonic() {
    // Ανάγνωση της απόστασης από τον υπερηχητικό αισθητήρα
    int distance = sonar.ping_cm();
    if (distance > 0 && distance < MAX_DISTANCE) {
        objectDetected = true; // Αν υπάρχει αντικείμενο μπροστά
    } else {
        objectDetected = false; // Αν δεν υπάρχει αντικείμενο
    }
}

```

Εικόνα 8.6: Συνάρτηση Ultrasonic

8.6 Πίνακας εντολών

ΕΝΤΟΛΕΣ	ΠΕΡΙΓΡΑΦΗ
<code>digitalWrite(pin, value)</code>	Δίνει την τιμή HIGH ή LOW στο συγκεκριμένο ψηφιακό PIN
<code>digitalRead(pin)</code>	Διαβάζει την τιμή από ένα συγκεκριμένο ψηφιακό PIN

analogWrite(pin, value)	Δίνει εύρος αναλογικών τιμών στο συγκεκριμένο αναλογικό PIN
pinMode(pin, Mode)	Διαμορφώνει το συγκεκριμένο PIN για να συμπεριφέρεται είτε σαν είσοδος είτε σαν έξοδος
attachInterrupt(digitalPinToInterrupt(pin), ISR, mode)	Ενεργοποιεί μια διακοπή (interrupt) για ένα συγκεκριμένο pin. Όταν το σήμα στο pin αλλάξει σύμφωνα με τον καθορισμένο mode (π.χ. RISING, FALLING, CHANGE), εκτελείται η συνάρτηση ISR (Interrupt Service Routine).
sonar.ping_cm()	Μετρά την απόσταση μέχρι ένα αντικείμενο χρησιμοποιώντας έναν υπερηχητικό αισθητήρα και επιστρέφει την απόσταση σε εκατοστά.
millis()	Επιστρέφει τον αριθμό των χιλιοστών του δευτερολέπτου που έχουν περάσει από τη στιγμή που ξεκίνησε να εκτελείται το πρόγραμμα.
delay(ms)	Καθυστερεί το πρόγραμμα για όσο χρόνο έχει δοθεί μέσα στην παρένθεση σε χιλιοστά του δευτερολέπτου (ms)

Πίνακας 8.1: Πίνακας εντολών

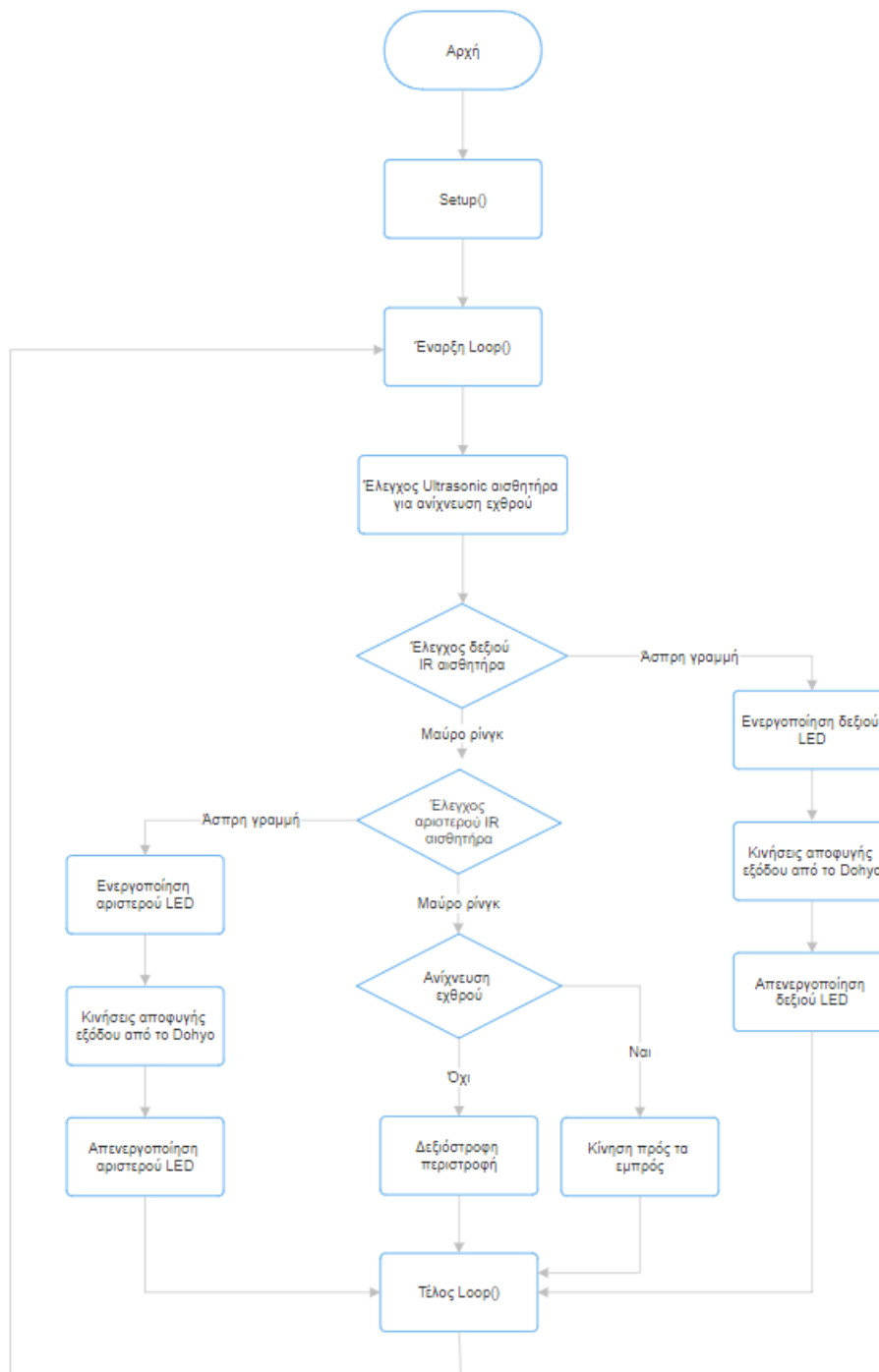
8.7 Πίνακας λέξεων κλειδιών δήλωσης μεταβλητών

ΛΕΞΕΙΣ ΚΛΕΙΔΑ	ΠΕΡΙΓΡΑΦΗ
int	Τύπος δεδομένων που χρησιμοποιείται για την αποθήκευση ακέραιων αριθμών
bool	Τύπος δεδομένων που χρησιμοποιείται για την αποθήκευση μιας τιμής που μπορεί να είναι είτε true είτε false.
const	Χαρακτηρισμός που χρησιμοποιείται για να δηλώσει ότι μια μεταβλητή είναι "σταθερή" και δεν μπορεί να αλλάξει η τιμή της μετά την αρχική της δήλωση.
volatile	Χαρακτηρισμός που χρησιμοποιείται για να δηλώσει ότι μια μεταβλητή μπορεί να αλλάξει απροσδόκητα, συνήθως από εξωτερικούς παράγοντες όπως διακοπές (interrupts) ή άλλες ταυτόχρονες διεργασίες.
unsigned	Χαρακτηρισμός που χρησιμοποιείται για να δηλώσει ότι μια μεταβλητή αποθηκεύει μόνο θετικούς αριθμούς

και διευρύνει το εύρος των τιμών που μπορεί να αποθηκεύσει.

Πίνακας 8.2: Πίνακας λέξεων κλειδιών δήλωσης μεταβλητών

8.8 Διάγραμμα ροής



Εικόνα 8.7: Διάγραμμα ροής

8.9 Επίλογος

Στο κεφάλαιο αυτό αναλύθηκε η δομή και η λειτουργία του κώδικα για τον έλεγχο του ρομπότ με Arduino. Ο κώδικας οργανώνεται σε έξι βασικά στάδια: κλήση βιβλιοθηκών, δήλωση μεταβλητών, δημιουργία αντικειμένων, ορισμός των συναρτήσεων `setup()` και `loop()`, και δημιουργία πρόσθετων συναρτήσεων. Συνολικά, το κεφάλαιο προσφέρει μια σαφή εικόνα της λειτουργικότητας του κώδικα που χρησιμοποιήθηκε.

Κεφάλαιο 9^ο: Συμπεράσματα και προτάσεις βελτίωσης

9.1 Συμπεράσματα

Μετά την περάτωση της συγκεκριμένης εργασίας κατάλαβα ότι το Arduino έχει πάρα πολλές δυνατότητες, με μερικές γνώσεις ηλεκτρονικής και προγραμματισμού μπορούν να δημιουργηθούν εξαιρετικές κατασκευές. Με την κατασκευή του Sumo Robot κατέληξα στο συμπέρασμα ότι ο συνδυασμός της μέγιστης απόδοσης με την τήρηση των προδιαγραφών της αγωνιστικής κατηγορίας χρειάζεται μία καλά σχεδιασμένη στρατηγική και προσεκτική εφαρμογή της τεχνολογίας. Η κατασκευή του Sumo Robot με Arduino ανέδειξε τις δυνατότητες του μικροελεγκτή.

9.2 Βελτιώσεις

Πάντα υπάρχει χώρος για βελτίωση, μια από αυτές θα μπορούσε να ήταν η χρήση ανώτερων εξαρτημάτων, όπως κινητήρων και αισθητήρων. Η προσθήκη δεύτερου Ultrasonic αισθητήρα καθώς και η χρήση servo κινητήρων, πιστεύω θα επέτρεπε την καλύτερη κάλυψη του περιβάλλοντος χώρου. Επιπρόσθετα καταλληλότερη θα ήταν η χρήση ενός πλαισίου του οποίου το βάρος θα ήταν ικανοποιητικό για την αγωνιστική κατηγορία χωρίς την χρήση πρόσθετων βαριδιών.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Wilson, Bryan, Tyler Germann, and Khalid Al-Olimat. "Sumo robot competition." Proceedings of the 2016 ASEE North Central Section Conference. 2016.
- [2] "About." Accessed August 10, 2024. <https://www.fsi.co.jp/sumo/robot/en/about.html>.
- [3] "Unified Sumo Robot Rules." Accessed June 30, 2024. <https://robogames.net/rules/all-sumo.php>.
- [4] Ismailov, Alisher, and Zafar Jo`rayev. "Study of Arduino Microcontroller Board," March 27, 2022.
- [5] "Arduino Hardware." Accessed August 10, 2024. <https://www.arduino.cc/en/hardware>.
- [6] Banzi, Massimo, and Michael Shiloh. Getting Started With Arduino. Maker Media, Inc., 2022.
- [7] Nayyar, Anand, and Vikram Puri. "A Review of Arduino Board's, Lilypad's & Arduino Shields." In 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 1485–92, 2016. <https://ieeexplore.ieee.org/document/7724514/?arnumber=7724514>.
- [8] "List of Arduino Boards and Compatible Systems." In Wikipedia, July 10, 2024. https://en.wikipedia.org/w/index.php?title=List_of_Arduino_boards_and_compatible_systems&oldid=1233617540.
- [9] Sudhan, R.Hari, M.Ganesh Kumar, A.Udhaya Prakash, S.Anu Roopa Devi, and Sathiya P. "ARDUINO ATMEGA-328 MICROCONTROLLER." IJIREICE 3, no. 4 (April 15, 2015): 27–29.
- [10] Fatoni, Ahmad, Dhany Dwi Nugroho, and Agus Irawan. "RANCANG BANGUN ALAT PEMBELAJARAN MICROCONTROLLER BERBASIS ATMEGA 328 DI UNIVERSITAS SERANG RAYA." PROSISKO: Jurnal Pengembangan Riset Dan Observasi Sistem Komputer 2, no. 1 (March 31, 2015). <https://e-jurnal.lppmunsera.org/index.php/PROSISKO/article/view/93>.
- [11] Einstronic Enterprise. "TT Motor Yellow Geared DC Motor." Accessed August 11, 2024. <https://einstronic.com/product/tt-motor-yellow-g geared-dc-motor/>.
- [12] Vishay Semiconductors, "Reflective Optical Sensor with Transistor Output", TCRT5000 datasheet, Aug. 2009.
- [13] ELEC Freaks, "Ultrasonic Ranging Module HC - SR04", HC - SR04 datasheet // **Δεν ξέρω αν είναι σωστό χωρίς ημερομηνία**
- [14] admin. "Arduino Sensor Shield V5 Datasheet: Features, Pinouts, and Specifications." *Components Explorer* (blog), August 24, 2020.
- [15] ST, "Dual full – bridge driver L298", L298 datasheet, Oct. 2023.

ΠΑΡΑΡΤΗΜΑ Α : ΚΩΔΙΚΑΣ

```

#include <NewPing.h>

const int ENA = 6;
const int IN1 = 5;
const int IN2 = 4;
const int IN3 = 7;
const int IN4 = 8;
const int ENB = 9;

const int IRRightSensor = 3;
const int IRLeftSensor = 2;

const int LED_PIN_RIGHT = 10;
const int LED_PIN_LEFT = 11;

const int trigPin = 12;
const int echoPin = 13;

volatile int rightSensorState = LOW;
volatile int leftSensorState = LOW;

volatile unsigned int lastInterruptTime = 0;
const unsigned int debounceDelay = 600;

volatile bool objectDetected = false;

const int MAX_DISTANCE = 35;

NewPing sonar(trigPin, echoPin, MAX_DISTANCE);

void setup() {
  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(IRRightSensor, INPUT);
  pinMode(IRLeftSensor, INPUT);

  pinMode(LED_PIN_RIGHT, OUTPUT);
  pinMode(LED_PIN_LEFT, OUTPUT);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  attachInterrupt(digitalPinToInterrupt(IRRightSensor), rightSensorISR, RISING);
  attachInterrupt(digitalPinToInterrupt(IRLeftSensor), leftSensorISR, RISING);
}

```

```

void loop() {
  checkUltrasonic();
  if (rightSensorState) {
    digitalWrite(LED_PIN_RIGHT, HIGH);
    s(100);
    b(800);
    s(100);
    l(350);
    rightSensorState = false;
    digitalWrite(LED_PIN_RIGHT, LOW);
  } else if (leftSensorState) {
    digitalWrite(LED_PIN_LEFT, HIGH);
    s(100);
    b(800);
    s(100);
    r(350);
    leftSensorState = false;
    digitalWrite(LED_PIN_LEFT, LOW);
  } else if (objectDetected) {
    f(5);
  } else {
    r(5);
  }
}

void checkUltrasonic() {
  int distance = sonar.ping_cm();
  if (distance > 0 && distance < MAX_DISTANCE) {
    objectDetected = true;
  } else {
    objectDetected = false;
  }
}

void rightSensorISR() {
  unsigned long currentTime = millis();
  if (currentTime - lastInterruptTime > debounceDelay) {
    rightSensorState = digitalRead(IRRightSensor);
    lastInterruptTime = currentTime;
  }
}

void leftSensorISR() {
  unsigned long currentTime = millis();
  if (currentTime - lastInterruptTime > debounceDelay) {
    leftSensorState = digitalRead(IRLeftSensor);
    lastInterruptTime = currentTime;
  }
}

```

```
void f(int d) {  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, LOW);  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, LOW);  
    analogWrite(ENA, 150);  
    analogWrite(ENB, 150);  
    delay(d);  
}
```

```
void b(int d) {  
    digitalWrite(IN1, LOW);  
    digitalWrite(IN2, HIGH);  
    digitalWrite(IN3, LOW);  
    digitalWrite(IN4, HIGH);  
    analogWrite(ENA, 110);  
    analogWrite(ENB, 110);  
    delay(d);  
}
```

```
void r(int d) {  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, LOW);  
    digitalWrite(IN3, LOW);  
    digitalWrite(IN4, HIGH);  
    analogWrite(ENA, 100);  
    analogWrite(ENB, 100);  
    delay(d);  
}
```

```
void l(int d) {  
    digitalWrite(IN1, LOW);  
    digitalWrite(IN2, HIGH);  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, LOW);  
    analogWrite(ENA, 100);  
    analogWrite(ENB, 100);  
    delay(d);  
}
```

```
void s(int d) {  
    digitalWrite(IN1, LOW);  
    digitalWrite(IN2, LOW);  
    digitalWrite(IN3, LOW);  
    digitalWrite(IN4, LOW);  
    analogWrite(ENA, 0);  
    analogWrite(ENB, 0);  
    delay(d);  
}
```