



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική Εργασία

**Ανάπτυξη Ρομποτικού Οχήματος Καθοδήγησης Ατόμων με
Προβλήματα Όρασης**



Φοιτητής: Γιώργο Σακελλάρη
ΑΜ:46104

Επιβλέπων: Διονύσης Κανδρής,
Καθηγητής

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, ΜΑΡΤΙΟΣ 2022



UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS
ENGINEERING


Diploma Thesis
Development of a robotic vehicle for the guidance of
people with visual impairments



Student: Giorgo Sakellari
Registration Number: 46104
Supervisor: Dionisis Kandris, Professor

ATHENS, MARCH 2022

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

Διονύσης Κανδρός Καθηγητής	Αλέξανδρος Αλεξανδρίδης, Καθηγητής	Γιώργος Πάτσας, Καθηγητής
 (Υπογραφή)	 (Υπογραφή)	 (Υπογραφή)

Copyright © Με επιφύλαξη παντός δικαιώματος. All rights reserved.

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ και (Σακελλάρη Γιώργο),
Μάρτιος, 2022**

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον/την συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Γιώργο Σακελλάρη του Πάντη, με αριθμό μητρώου 46104 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ,

δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου»

Ο Δηλών
Γιώργο Σακελλάρη



Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής μου εργασίας θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, κύριο Διονύσιο Κανδρή για την εμπιστοσύνη του αναθέτοντάς μου το συγκεκριμένο θέμα, την καθοδήγησή του, τα εποικοδομητικά σχόλια και την υπομονή του αλλά και για την συμπαράσταση που έδειξε από την αρχή έως το τέλος της διπλωματικής αυτής.

Τέλος θα ήθελα να εκφράσω την αγάπη μου στην οικογένειά μου και στην κοπέλα μου για την συνεχή στήριξη, συμπαράσταση και κατανόηση καθ' όλη την διάρκεια της πανεπιστημιακής σταδιοδρομίας.

Περίληψη

Η συγκεκριμένη διπλωματική εργασία έχει ως σκοπό την ανάπτυξη ενός ρομποτικού οχήματος για την καθοδήγηση ατόμων με προβλήματα όρασης. Αρχικά, αναλύεται το θεωρητικό κομμάτι του ρομποτικού κλάδου. Ειδικότερα, γίνεται αναφορά στην ιστορική εξέλιξη, τον προγραμματισμό και τις εφαρμογές της ρομποτικής. Στην συνέχεια, περιγράφεται ένα θεωρητικό υπόβαθρο σχετικά με το hardware αλλά και το software των Arduino, τις ρυθμίσεις τους, τις λειτουργίες και τις δυνατότητες που διαθέτουν μιας και στην συγκεκριμένη εφαρμογή ο μικροελεγκτής που χρησιμοποιήθηκε είναι το Arduino Uno. Έπειτα, αναλύονται τα υλικά της κατασκευής αλλά και τα εργαλεία που χρησιμοποιήθηκαν. Ακολουθούν τα βήματα με τα οποία στήθηκε η κατασκευή και περιγράφεται η λειτουργία του κάθε εξαρτήματος αλλά και ο ρόλος του για την συγκεκριμένη εφαρμογή. Ύστερα, αναλύεται βήμα βήμα η σκέψη με την οποία πρέπει το όχημα να ακολουθεί, ούτως ώστε να είναι ευκολότερος ο προγραμματισμός του μικροεπεξεργαστή ο οποίος και παρατίθεται σε συγκεκριμένο κεφάλαιο με αντίστοιχα σχόλια για να γίνεται κατανοητός ο τρόπος με τον οποίο λειτουργεί. Τέλος, γίνεται μία σύνοψη της διπλωματικής εργασίας υπογραμμίζοντας προβλήματα τα οποία παρουσιάστηκαν, τονίζονται μελλοντικές αναβαθμίσεις που μπορούν να προστεθούν στην παρούσα διπλωματική για την βελτίωση των δυνατοτήτων του οχήματος.

Λέξεις – κλειδιά

ρομποτικό όχημα, γλώσσες προγραμματισμού, ρομποτικές εφαρμογές, arduino, αισθητήρες, αποφυγή εμποδίων .

Περιεχόμενα

Περιεχόμενα.....	7
Κατάλογος Εικόνων.....	8
ΕΙΣΑΓΩΓΗ.....	10
ΚΕΦΑΛΑΙΟ 1. Ρομποτική.....	12
1.1 Ορισμός και Ιστορική Εξέλιξη.....	12
1.2 Είδη Ρομπότ.....	17
1.3 Αισθητήρες.....	25
1.4 Προγραμματισμός.....	27
1.5 Εφαρμογές.....	29
ΚΕΦΑΛΑΙΟ 2. Arduino.....	34
2.1 Εισαγωγή στον ελεγκτή Arduino.....	34
2.2 Πλεονεκτήματα Arduino.....	35
2.3 Χαρακτηριστικά Arduino.....	37
2.4 Προγραμματισμός ελεγκτή Arduino.....	39
ΚΕΦΑΛΑΙΟ 3. Ανάπτυξη Εφαρμογής.....	42
3.1 Υλικά Κατασκευής.....	42
3.2 Συνδεσμολογία Κατασκευής.....	49
3.3 Προγραμματισμός Κατασκευής.....	56
3.4 Πειραματική Διαδικασία.....	64
4.1 Σύνοψη.....	65
4.2 Προβλήματα και Αντιμετώπιση.....	65
4.3 Συμπεράσματα.....	65
4.4 Προτάσεις Μελλοντικής Εξέλιξης.....	66
Βιβλιογραφία – Αναφορές - Διαδικτυακές Πηγές.....	67
ΠΑΡΑΡΤΗΜΑ 1.....	71
ΠΑΡΑΡΤΗΜΑ 2.....	74

Κατάλογος Εικόνων

Εικόνα 1: “Μηχανισμός των Αντικυθήρων.....	11
Εικόνα 2: Leonardo Da Vinci Robot.....	11
Εικόνα 3: Turing Test.....	12
Εικόνα 4: Shakey robot.....	13
Εικόνα 5: Mars Pathfinder robot.....	13
Εικόνα 6: Aibo robodog.....	14
Εικόνα 7: Asimo.....	14
Εικόνα 8: Asimo evolution.....	14
Εικόνα 9: Sophia robot.....	15
Εικόνα 10: Καρτεσιανό, κυλινδρικό, σφαιρικό, αρθρωτό ρομπότ.....	16
Εικόνα 11: Scara robot.....	16
Εικόνα 12: Delta robot.....	17
Εικόνα 13: Δίποδο ρομπότ Salto.....	17
Εικόνα 14: Τετράποδο ρομπότ Aibo.....	17
Εικόνα 15: Εξάποδο ρομπότ Spider robot.....	17
Εικόνα 16: Υποβρύχιο ρομπότ NOAA.....	18
Εικόνα 17: SSI Schäfer – Weasel.....	18
Εικόνα 18: F015 Mercedes.....	19
Εικόνα 19: Tank robot.....	19
Εικόνα 20: Scorpion”.....	19
Εικόνα 21: Πολεμικό UAV με άκαμπτα φτερά.....	20
Εικόνα 22: Μικρό αερόστατο robot.....	20
Εικόνα 23: UAV με φτερά υπό-κλίση.....	20
Εικόνα 24: UAV “με φτερά”.....	20
Εικόνα 25: Με έλικες “drone”.....	20
Εικόνα 26: Mbari.....	21
Εικόνα 27: Girona 500.....	21
Εικόνα 28: Hyball.....	21
Εικόνα 29: Talos.....	21
Εικόνα 30: Romeo.....	21
Εικόνα 31: Atlas.....	22
Εικόνα 32: Ocean One.....	22
Εικόνα 33: Handle.....	22
Εικόνα 34: RoboBee.....	22
Εικόνα 35: Φωτοτρανζίστορ.....	23
Εικόνα 36: Φωτοαντίσταση.....	23
Εικόνα 37: Αισθητήρας ήχου.....	23
Εικόνα 38: Αισθητήρας πίεσης.....	24
Εικόνα 39: Χωρητικός αισθητήρας.....	25
Εικόνα 40: Επαγωγικός αισθητήρας.....	25
Εικόνα 41: Φωτοηλεκτρικός αισθητήρας.....	25
Εικόνα 42: Μαγνητικός αισθητήρας.....	25
Εικόνα 43: Γραφικό περιβάλλον Scratch.....	26
Εικόνα 44: Ρομπότ βαφής.....	27
Εικόνα 45: Pick and place βραχίονες.....	27
Εικόνα 46: Robots Assemble car parts.....	27
Εικόνα 47: Spot.....	28

Εικόνα 48: Da Vinci.....	28
Εικόνα 49: Amazon “courier.....	29
Εικόνα 50: Robonaut.....	29
Εικόνα 51: DOGO.....	30
Εικόνα 52: Guardbot.....	30
Εικόνα 53: Bomb defusing robot... ..	30
Εικόνα 54: Surveillance Robot	30
Εικόνα 55: Arduino boards	32
Εικόνα 56: Arduino Shields	33
Εικόνα 57: Arduino Uno	34
Εικόνα 58: Arduino Starter Kit	34
Εικόνα 59: Arduino Uno (pins).....	34
Εικόνα 60: Arduino Integrated Development Environment	36
Εικόνα 61: Κουμπιά IDE	37
Εικόνα 62: Arduino Uno	38
Εικόνα 63: Πλαίσιο (σασί).....	38
Εικόνα 64: Τροχοί Mecanum.....	39
Εικόνα 65: DC Gear Motor TT.....	40
Εικόνα 66: Shield L293D.....	41
Εικόνα 67: HC SR04 Sensor.....	41
Εικόνα 68: Βάση HC SR04 Sensor.....	42
Εικόνα 69: SG90 Micro Servo.....	42
Εικόνα 70: Διακόπτης	43
Εικόνα 71: M-F , M-M Καλώδια	43
Εικόνα 72: Μπαταρία.....	44
Εικόνα 73: Σύνδεσμος μπαταρίας.....	44
Εικόνα 74: Μπαστούνι.....	48
Εικόνα 75: Κόλληση καλωδίων - Κινητήρα	49
Εικόνα 76: Πλαϊνό μέρος σύνδεσης κινητήρων	50
Εικόνα 77: Ενδείξεις θέσεων κινητήρων	50
Εικόνα 78: Μπροστινό μέρος αισθητήρα	51
Εικόνα 79: Θέση σέρβο κινητήρα.....	52
Εικόνα 80: Πίσω μέρος αισθητήρα.....	52
Εικόνα 81: Υποδοχές σύνδεσης αισθητήρα	53
Εικόνα 82: Πίσω μέρος κατασκευής.....	53
Εικόνα 83 Μπροστά μέρος κατασκευής	54
Εικόνα 84 τελικές εικόνες με μπαστούνι	55

ΕΙΣΑΓΩΓΗ

Τα τελευταία χρόνια το πεδίο των αυτόνομων ρομποτικών οχημάτων αναπτύσσεται καθημερινά. Ένα αυτόνομο ρομπότ πρέπει να είναι σε θέση να λαμβάνει πληροφορίες από το περιβάλλον και να δουλεύει για μεγάλο χρονικό διάστημα χωρίς την παρέμβαση χειριστή. Στην συγκεκριμένη εργασία ασχολούμαστε με την κατασκευή ενός ρομποτικού οχήματος που θα έχει την δυνατότητα να κινείται στον χώρο και όταν θα εντοπίζει εμπόδια να μπορεί να κινηθεί προς άλλες κατευθύνσεις που δεν εμπεριέχουν εμπόδια η να επιλέγει την βέλτιστη διαδρομή που είναι ελεύθερη σε κάποια συγκεκριμένη απόσταση. Ένα τέτοιο όχημα θα μπορούσε μελλοντικά να αντικαταστήσει τους σκύλους συνοδούς που χρησιμοποιούν σε μικρό βαθμό άτομα με προβλήματα όρασης λόγω κόστους. Για την υλοποίηση αυτού του οχήματος χρησιμοποιήθηκαν απλά υλικά και αισθητήρες έχοντας έτσι μικρό κόστος κατασκευής αλλά και για την προοπτική αναβάθμισης και μελλοντικής εξέλιξης. Στο πειραματικό στάδιο χρησιμοποιήθηκαν κούτες κυρίως για εμπόδια ούτως ώστε να επαληθευτεί ο τρόπος με τον οποίο το όχημα εντοπίζει το εμπόδιο και επιλέγει την αντίστοιχη διαδρομή. Το αντικείμενο της διπλωματικής είναι : Η σκέψη για την εργασία αυτή ήταν μέσω της παρατήρησης της μετακίνησης των ατόμων με προβλήματα όρασης. Παρατηρούμε καθημερινά άτομα με προβλήματα όρασης να μετακινούνται έξω χωρίς να είναι πλήρως αυτόνομοι και να αντιμετωπίζουν δυσκολίες. Για παράδειγμα για να μεταφερθούν στον προορισμό που επιθυμούν θα χρειαστεί να ρωτήσουν πιθανόν συνανθρώπους για να τους καθοδηγήσουν στην σωστή κατεύθυνση. Σε χώρες τους εξωτερικού κάποιοι χρησιμοποιούν σκύλους οδηγούς . Οι συγκεκριμένο σκύλοι εκπαιδεύονται για να εργάζονται απεριόριστα αγνοώντας κάθε είδους οπτικά ερεθίσματα θορύβους και φυσικά να αποφεύγουν εμπόδια είτε σταθερά είτε κινητά όπως είναι οι άνθρωποι και τα οχήματα. Επειδή το κόστος εκπαίδευση τέτοιων σκυλιών μπορεί να ξεπεράσει τα 20000 ευρώ δημιουργήθηκε η σκέψη κατασκευής ενός οχήματος που με σταδιακές βελτιώσεις και αναβαθμίσεις να μπορεί να είναι σε θέση να κάνει την δουλειά των σκύλων συνοδών αλλά να μπορούν να ανταπεξέλθουν σε μεγαλύτερα προβλήματα και με μικρότερο κόστος από ότι η εκπαίδευση των σκύλων οδηγών Όσον αφορά τη δομή της ακόλουθης εργασίας έχουμε: Στο πρώτο κεφάλαιο γίνεται μια γενική αναφορά γύρω από την ρομποτική. Αναλύονται κάποια βασικά χαρακτηριστικά στους αισθητήρες τον προγραμματισμό τις εφαρμογές και τα είδη των ρομπότ ώστε να κατανοηθούν έννοιες που χρειάστηκαν στην παρούσα διπλωματική. Στο δεύτερο κεφάλαιο γίνεται ανάλυση πάνω στον επεξεργαστή Arduino υπο τόσο σε επίπεδο hardware όσο και software. Στο κεφάλαιο 3 παρουσιάζονται τα υλικά που χρειάστηκαν για την κατασκευή ο τρόπος συνδεσμολογίας καθώς και ο προγραμματισμός ο οποίος ελέγχει το όχημα Στο κεφάλαιο 4 γίνεται μια σύνοψη της διπλωματικής εργασίας παρουσιάζονται τα προβλήματα που παρουσιάστηκε καθώς και ο τρόπος αντιμετώπισης. Επίσης σχολιάζονται συμπεράσματα που σχετίζονται με την εργασία αυτή. Και τέλος παραθέτονται μερικές προτάσεις για μελλοντική εξέλιξη και αναβάθμιση της κατασκευής. Στο παράρτημα 1 παρατίθεται ο κώδικας

προγραμματισμού. Στο παράρτημα 2 παρατίθενται τα βίντεο από την πειραματική διαδικασία .

ΚΕΦΑΛΑΙΟ 1. Ρομποτική

1.1 Ορισμός και Ιστορική Εξέλιξη

Η ρομποτική είναι ένας κλάδος της τεχνολογίας που ασχολείται με το σχεδιασμό, την ανάπτυξη και τη μελέτη των ρομπότ. Είναι ένας συνδυασμός πολλών άλλων επιστημών όπως η επιστήμη των υπολογιστών και της μηχανικής. Οι επιστήμονες προσπαθούν όλο και περισσότερο να δημιουργήσουν "έξυπνα" ανδροειδή ρομπότ που θα συμπεριφέρονται σαν τους ανθρώπους [1].

Το πρώτο ρομπότ κατασκευάστηκε από τον μαθηματικό Αρχύτα τον Ταραντίνο το 428-347 π.Χ., ο οποίος δημιούργησε μία ιπτάμενη μηχανή με την ονομασία "περιστέρα" ή "πετομηχανή". Η μηχανή αυτή κινούταν με ατμό και είχε την δυνατότητα να πετάξει έως και για 200 μέτρα απόσταση. Κατά το 2^ο αιώνα π.Χ. κατασκευάστηκε αυτοματισμός που μπορούσε να εντοπίζει τις θέσεις του κάθε πλανήτη με την ονομασία "Μηχανισμός των Αντικυθήρων".



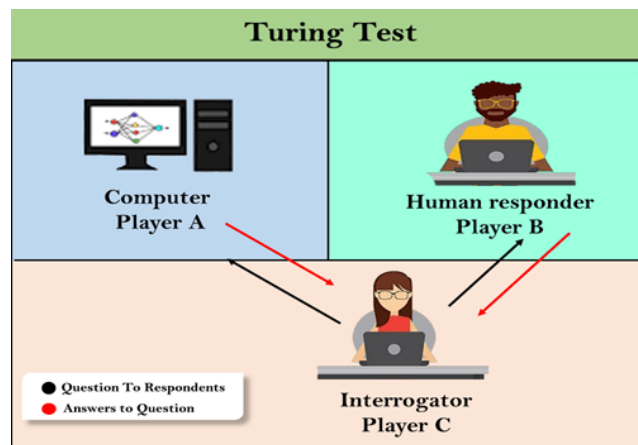
Εικόνα 1: "Μηχανισμός των Αντικυθήρων"

Ακολούθως, κατά το 1^ο αιώνα π.Χ. ο Ήρων κατασκεύασε ένα αυτοκινούμενο τρίκυκλο. Στην συνέχεια, η εμφάνιση του πρώτου ανθρωποειδούς προγραμματιζόμενου ρομπότ γίνεται το 1206 από τον Άραβα Al-Jazari. Το 1495 ο Ιταλός Leonardo Da Vinci σχεδιάζει μία μηχανική κατασκευή που φέρεται να μοιάζει με στρατιώτη φορώντας πανοπλία εκείνης της εποχής. Το συγκεκριμένο ρομπότ είχε την δυνατότητα να κινεί τα χέρια και το κεφάλι [4].



Εικόνα 2: Leonardo Da Vinci Robot

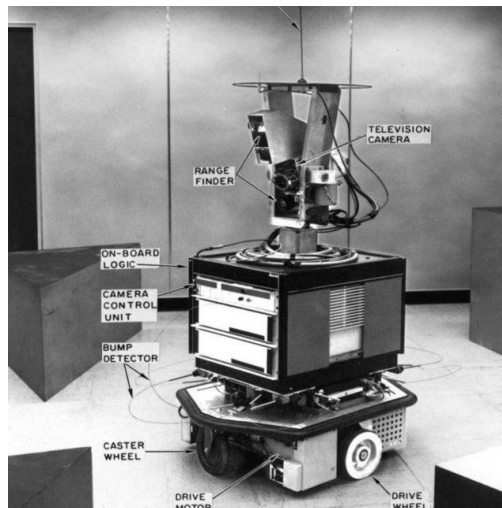
Επιπρόσθετα, η εταιρία Westinghouse Electric Corporation το 1930, δημιουργεί ένα ανθρωποειδές ρομπότ με την ονομασία Electro και είχε την δυνατότητα να περπατάει, να μιλάει και να καπνίζει. Λίγο αργότερα ο Alan Turing εισάγει την έννοια της “Υπολογιστικής Νοημοσύνης” αλλά και της “Τεχνητής Νοημοσύνης” και με την βοήθεια ενός πειράματος με την ονομασία Turing Test ορίζει τότε μία μηχανή καλείται νοήμων [3].



Εικόνα 3: Turing Test

Ο Issac Asimov το 1942 διατυπώνει σε θεωρητικό επίπεδο τρεις βασικούς νόμους της ρομποτικής. Ο πρώτος νόμος λέει πως ένα ρομπότ δεν πρέπει ποτέ να βλάψει έναν άνθρωπο. Ο δεύτερος, ότι πάντα πρέπει να υπακούει όλες τις εντολές που του έχει δώσει ο άνθρωπος εκτός και αν υπάρξει σύγκρουση με κάποιον άλλο ισχυρότερο νόμο. Τέλος, το ρομπότ καλείται να προστατέψει την ύπαρξη του αλλά όπως και στον προηγούμενο νόμο αν υπάρξει ανώτερος νόμος δεν πραγματοποιείται η διαδικασία [5]. Ακόμη, ο George Devol το 1962 υλοποιεί τον πρώτο προγραμματιζόμενο ρομποτικό βραχίονα με το όνομα Unimate το οποίο φέρει σε πέρας διάφορες εργασίες. Το Ινστιτούτο Έρευνας του πανεπιστημίου Stanford κατασκευάζει το Shakey, το πρώτο κινούμενο ρομπότ με Τεχνητή Νοημοσύνη. Είχε την δυνατότητα να

εντοπίζει αντικείμενα και να αλληλοεπιδράσει με αυτά, για παράδειγμα να τα κουνήσει [6].



Εικόνα 4: Shakey robot

Μια ακόμη θεωρητική παρέμβαση γίνεται από τον Seymour Papert το 1980, ο οποίος με την δημοσίευση του βιβλίου του με τίτλο “Mindstorms: Children, Computers, and Powerful Ideas” προτείνει για πρώτη φορά την χρήση εκπαιδευτικών ρομπότ και ταυτόχρονα εισάγει την έννοια της εκμάθησης μέσα από την κατασκευή και το πείραμα. Πάνω στο κομμάτι αυτό βασίστηκε η εταιρία Lego όπου το 1986 συνεργάζεται με το πανεπιστήμιο MIT και κυκλοφορούν το πρώτο εκπαιδευτικό προϊόν βασισμένο σε Lego, το Lego TC Logo. Χρησιμοποιήθηκε στην εκπαιδευτική διαδικασία και στην διδασκαλία εννοιών της επιστήμης των υπολογιστών. Στον χώρο αυτόν εντάσσεται και η εταιρία Honda το 1993 με την ανάπτυξη μοντέλων “Prototype, P” όπου γινόταν προσπάθεια για την κατασκευή ρομπότ που θα μπορούσαν να εκτελέσουν βασικές εργασίες. Ταυτόχρονα, το 1997 η NASA κατασκευάζει και στέλνει στον Άρη το Mars Pathfinder το οποίο είναι ένα ρομποτικό διαστημικό σκάφος με ανιχνευτή κινήσεων και είχε ως αποστολή την λήψη εικόνων και μεταφορά δεδομένων από τον Άρη στην Γη [7].



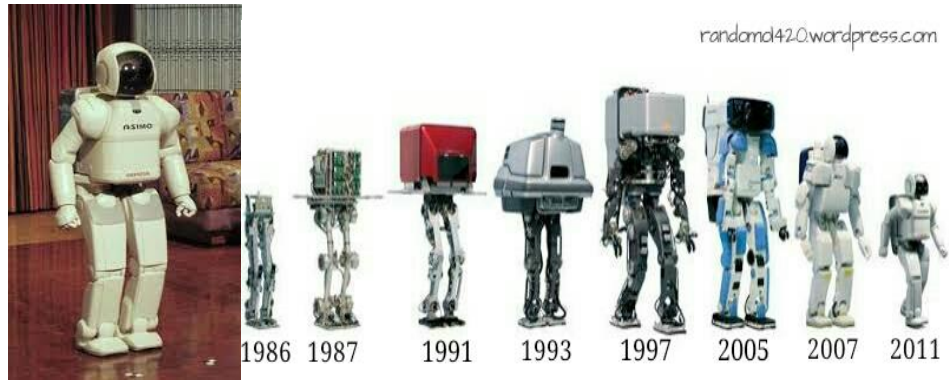
Εικόνα 5: Mars Pathfinder robot

Ύστερα από έναν χρόνο στο Τεχνολογικό Ινστιτούτο της Μασαχουσέτης δημιουργήθηκε από την Dr.Cynthia Breazeal το ρομπότ Kismet, μία κεφαλή ρομπότ που είχε ως εργασία την αναγνώριση και προσομοίωση ανθρωπίνων συναισθημάτων. Στην συνέχεια το 1999 η Sony εισέρχεται και εκείνη στην ρομποτική βιομηχανία με την παρουσίαση του Aibo. Το Aibo είχε την μορφή ενός μικρού κουταβιού σκύλου και κατά κύριο λόγο χρησιμοποιήθηκε σε διαγωνισμούς ποδοσφαίρου ρομπότ (RoboCup).



Εικόνα 6: Aibo robodog

Παράλληλα, η Honda το 2002 κυκλοφορεί ένα νέο μοντέλο ανθρωποειδούς μορφής με την ονομασία ASIMO. Το ASIMO είναι ικανό να ερμηνεύει στάσεις και χειρονομίες των ανθρώπων. Μπορεί ωστόσο να χαιρετάει τους ανθρώπους που πλησιάζουν σε εκείνο, να τους ακολουθεί στην κατεύθυνση που του υποδεικνύουν οι άνθρωποι αλλά εμπεριέχει και την τεχνική της αναγνώρισης προσώπων και προτύπων.



Εικόνα 7: Asimo

Εικόνα 8: Asimo evolution

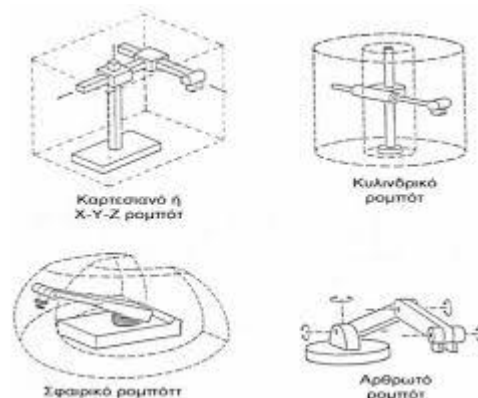
Το 2003 η Seiko Epson παρουσίασε στην Διεθνή Έκθεση Ρομπότ στο Τόκιο το μικρότερο και ελαφρύτερο ρομποτικό ελικόπτερο. Σκοπός ήταν η επίδειξη της μικρομηχαντρονικής τεχνολογίας για χρήση κάτω από καθεστώς φυσικών καταστροφών. Έπειτα από μία δεκαετία εμφανίζεται το πρώτο κοινωνικό ανθρωποειδές ρομπότ, το Pepper. Το Pepper μπορεί να αναγνωρίσει πρόσωπα και βασικά ανθρώπινα συναισθήματα. Ο βασικός στόχος του Pepper είναι να αλληλεπιδρά με τους ανθρώπους μέσω της συνομιλίας και της οθόνης αφής του. Μέχρι και σήμερα το Pepper είναι διαθέσιμο για χρήση σε επιχειρήσεις αλλά και σχολεία. Υπογραμμίζεται ότι το 2017 παρουσιάζεται η Sophia, ένα ανθρωποειδές ρομπότ που αναπτύχθηκε από την εταιρία Hanson Robotics και γίνεται το πρώτο πολιτογραφημένο ρομπότ στην Σαουδική Αραβία. Η Sophia έχει ανακηρυχθεί πρωταθλητής καινοτομίας στο πλαίσιο του Προγράμματος Ανάπτυξης των Ηνωμένων Εθνών. Μιμείται ανθρώπινες χειρονομίες και εκφράσεις του προσώπου και μπορεί να απαντάει σε ορισμένες ερωτήσεις αλλά και να κάνει απλές συνομιλίες. Ωστόσο, ο σχεδιασμός της, της δίνει την ικανότητα να γίνεται πιο έξυπνη με την πάροδο του χρόνου. Ο βασικός της ρόλος είναι για την συντροφιά των ηλικιωμένων ατόμων στα γηροκομεία ή για να βοηθά σε μεγάλες εκδηλώσεις ή πάρκα όπου εντοπίζεται συνωστισμός [8].



Εικόνα 9: Sophia robot

1.2 Είδη Ρομπότ

Στα είδη των ρομπότ διακρίνουμε δύο κατηγορίες, τα ρομπότ σταθερής βάσης και τα κινητά ρομπότ. Τα ρομπότ σταθερής βάσης είναι καρτεσιανά, κυλινδρικά, σφαιρικά, αρθρωτά, Scara και Delta. Αρχικά, τα καρτεσιανά ρομπότ αποτελούνται από έναν έως τρεις άξονες και μία βάση. Κινούνται σε ευθείες τροχιές σε έναν, δύο ή τρεις άξονες και σχηματίζουν μεταξύ του γωνία ενενήντα μοιρών. Καρτεσιανά ρομπότ συναντώνται σε πολλούς τομείς της βιομηχανίας όπως για την μεταφορά αντικειμένων, την τοποθέτηση και την ανύψωση. Ωστόσο μία εφαρμογή πλέον αρκετά διαδεδομένη των εκτυπωτών 3D. Εν συνέχεια, τα κυλινδρικά ρομπότ διαθέτουν έναν οριζόντιο άξονα που στερεώνεται σε μία κατακόρυφη κολώνα η οποία με την σειρά της στηρίζεται σε μία περιστρεφόμενη βάση. Ο βραχίονας μπορεί να κάνει μπρος πίσω κινήσεις κατά τη διεύθυνση του διαμήκου άξονά του και ανεβοκατεβαίνει στην κατακόρυφη κολώνα. Τα δύο αυτά μαζί μπορούν μαζί με τη βάση να στρέφονται σαν ένα σώμα στον κατακόρυφο άξονα. Ύστερα, τα σφαιρικά ρομπότ διαθέτουν μια στρεφόμενη βάση όπου στηρίζεται ένα ανυψωμένο στέλεχος και έναν βραχίονα. Ο βραχίονας μπορεί να κάνει κίνηση μέσα και έξω. Επιπλέον, τα αρθρωτά ρομπότ αποτελούνται από τρεις συνδέσμους, που ενώνονται με την βοήθεια στροφικών αρθρώσεων και είναι τοποθετημένα σε μια βάση που μπορεί να περιστρέφεται. Η κίνηση που υλοποιούν μοιάζει με το ανθρώπινο χέρι. Δηλαδή, η αρπάγη είναι ανάλογη της παλάμης που προσαρμόζεται στον κάτω βραχίονα μέσω του καρπού. Ο “αγκώνας” συνδέει τον κάτω βραχίονα με τον άνω και ο “ώμος” συνδέει τον άνω βραχίονα με την βάση.



Εικόνα 10: Καρτεσιανό, κυλινδρικό, σφαιρικό, αρθρωτό ρομπότ

Ο συνδυασμός των ιδιοτήτων ενός αρθρωτού και ενός κυλινδρικού ρομπότ αντιστοιχεί στα ρομπότ Scara. Τα ρομπότ Scara διαθέτουν δύο περιστροφικές αρθρώσεις και μία πρισματική. Οι άξονες όλων των αρθρώσεων είναι παράλληλοι μεταξύ τους [6] [9].



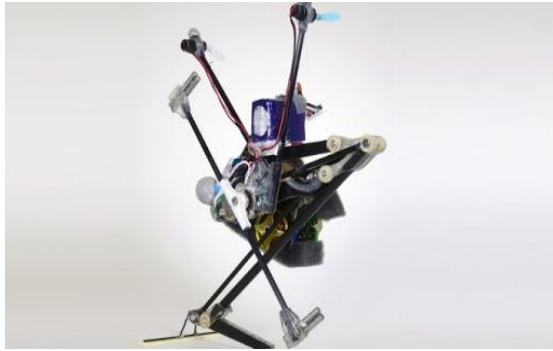
Εικόνα 11: Scara robot

Τέλος, τα ρομπότ “Delta” αποτελούνται από τρεις βραχίονες που συνδέονται στην βάση μέσω αρθρώσεων. Βασική τους ιδιότητα είναι η παραλληλόγραμμη χρήση των βραχιόνων, διατηρώντας τον προσανατολισμό του τελικού στοιχείου δράσης (end effector). Χρησιμοποιούνται σε εργοστάσια λόγω της μεγάλης ταχύτητας που αναπτύσσουν [4].



Εικόνα 12: Delta robot

Από την άλλη πλευρά στα κινητά ρομπότ χωρίζονται ανάλογα με τον βαθμό αυτονομίας τους. Για να μπορέσουν να κινηθούν χρειάζονται ειδικά συστήματα προώθησης, όπως μηχανικά πόδια, τροχούς, προπέλες και μηχανές jet. Παρακάτω διατυπώνονται κάποια από αυτά όπως, ρομπότ που διαθέτουν πόδια, έντροχα, εναέρια, υποβρύχια, ανθρωποειδή και μικρορομπότ. Τα ρομπότ που διαθέτουν πόδια κατηγοριοποιούνται σε ρομπότ με ένα πόδι όπως το Salto-1P , με δύο πόδια, με τέσσερα όπως το ρομπότ Aibo, εκείνα με 6 που μοιάζουν να έχουν μορφές όπως είναι η αράχνη κ.α.



Εικόνα 13: Δίποδο ρομπότ Salto 1P



Εικόνα 14: Τετράποδο ρομπότ Aibo



Εικόνα 15: Εξάποδο ρομπότ Spider robot

Ομοίως, στα έντροχα ρομπότ ανάλογα με την εκάστοτε εφαρμογή διακρίνονται τα τηλεκατευθυνόμενα ρομπότ δηλαδή που χρειάζονται χειριστή (Remotely Operated Vehicles), τα αυτόματα οδηγούμενα οχήματα (Automatic Guided Vehicles) και τα πλήρως αυτόνομα (Autonomous Wheeled Vehicles). Τα τηλεχειριζόμενα ρομπότ διαθέτουν πολύ μικρό βαθμό ελευθερίας μιας και είναι συνδεδεμένα ασύρματα ή ενσύρματα με τον χειριστή για να μπορούν να επικοινωνούν και να λαμβάνουν αποφάσεις.



Εικόνα 16: Υποβρύχιο ρομπότ NOAA

Με βάση το πλήθος των βαθμών ελευθερίας αμέσως μετά είναι τα αυτόματα οδηγούμενα, δηλαδή ρομπότ στα οποία έχει προκαθοριστεί η εργασία τους. Για την επίτευξη αυτού χρησιμοποιούνται διάφοροι αισθητήρες και πολλές φορές ο χώρος που εργάζονται φέρει καλώδια ή “χρωματιστούς διαδρόμους” για να μπορούν να ακολουθούν.



Εικόνα 17: SSI Schäfer – Weasel

Ωστόσο έχουμε και τα πλήρως αυτόνομα ρομπότ, τα οποία λαμβάνουν μόνα τους τις αποφάσεις και δεν χρειάζονται χειριστή. Τέτοια ρομπότ λαμβάνουν και στρατιωτικές εφαρμογές [4].



Εικόνα 18: F015 Mercedes



Εικόνα 19: Tank robot



Εικόνα 20: Scorpion"

Συνεχίζοντας, κατηγοριοποίηση υπάρχει και στο εναέριο πεδίο, τα γνωστά ως UAV ("Unmanned Aerial Vehicle"). Ο διαχωρισμός γίνεται με βάση των αεροδυναμικών χαρακτηριστικών και το είδος της ανύψωσης. Χαρακτηριστικά παραδείγματα είναι τα μικρά αερόστατα, με άκαμπτα φτερά, με φτερά υπό-κλίση, εκείνα που "φτερουγίζουν" διότι μιμούνται την κίνηση που κάνουν τα πουλιά και εκείνα με έλικες [10].



Εικόνα 22: Μικρό αερόστατο robot



Εικόνα 23: UAV με φτερά υπό-κλίση Εικόνα 24: UAV “με φτερά”



Εικόνα 25: Με έλικες “drone”

Έπειτα, σειρά έχουν τα υποβρύχια ρομπότ (“Autonomous Underwater Vehicles”). Το συγκεκριμένο πεδίο έχει ξεκινήσει να εξελίσσεται τα τελευταία χρόνια μιας και η υποβρύχια τεχνολογία είναι ένα στρυφνό κομμάτι προς υλοποίηση. Κάποια από αυτά είναι το Offshore Hyball, το Mbari και το Girona 500 [4].



Εικόνα 26: Mbari



Εικόνα 27: Girona 500



Εικόνα 28: Hyball

Επόμενη είδος ρομπότ είναι τα ανθρωποειδή. Διαθέτουν δύο χέρια και δύο πόδια μιας και έχουν ανθρώπινες προδιαγραφές και υλοποιούν διάφορες εφαρμογές όπως να έχουν οικιακά καθήκοντα, γραφείου, ψυχαγωγίας αλλά και να λειτουργήσουν σαν μηχανές πολέμου. Μερικά ανθρωποειδή ρομπότ είναι το Asimo, το Romeo, το Atlas, ο Talos κ.α.



Εικόνα 29: Talos

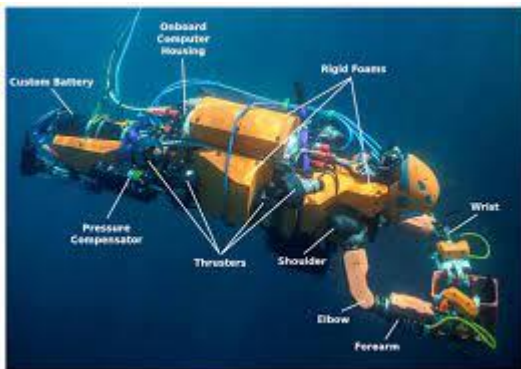


Εικόνα 30: Romeo



Εικόνα 31: Atlas

Ωστόσο, υπάρχουν και συνδυασμοί δύο ή περισσότερων ειδών ρομπότ όπως είναι τα υβριδικά ανθρωποειδή. Χαρακτηριστικά παραδείγματα είναι το Ocean One όπου είναι ένα υποβρύχιο ρομπότ με ανθρωπινή όψη, και το Handle, ρομπότ που διαθέτει χέρια κορμό και πόδια, στα οποία πόδια όμως έχουν ενσωματωθεί δύο ρόδες [11].

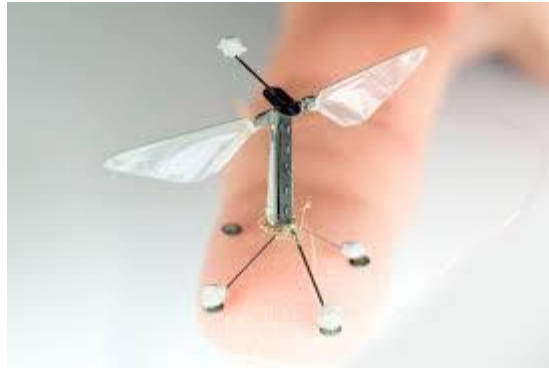


Εικόνα 32: Ocean One



Εικόνα 33: Handle

Τέλος, υπάρχουν ακόμη και μικρορομπότ (micro robots) των οποίων οι διαστάσεις είναι της τάξης μικρότερες του ενός μικρόμετρου. Τα ρομπότ αυτά αναμένεται να είναι χρήσιμα σε εφαρμογές αναζήτησης επιζώντων, ανίχνευση και μέτρηση τοξικών ή επιβλαβών αερίων για τον άνθρωπο. [12]



Εικόνα 34: RoboBee

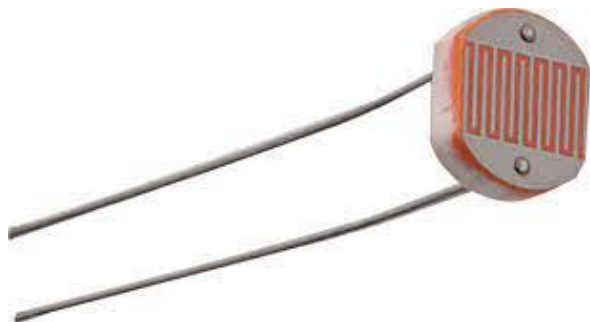
1.3 Αισθητήρες

Οι αισθητήρες που χρησιμοποιούνται στον τομέα της ρομποτικής ποικίλουν και επιλέγονται με βάση την εκάστοτε εφαρμογή που καλείται το κάθε ρομπότ να φέρει εις πέρας. Κάποια βασικά είδη αισθητήρων είναι φωτός, ήχου, θερμοκρασιακοί, επαφής, πίεσης και εγγύτητας που θα αναλυθούν σε επιμέρους υποκατηγορίες.

Ειδικότερα, ένας αισθητήρας φωτός έχει την ικανότητα να ανιχνεύει φως και να το μετατρέπει σε ηλεκτρική τάση. Δύο βασικοί αισθητήρες φωτός που χρησιμοποιούνται στα ρομπότ είναι οι φωτοαντιστάσεις, τα φωτοτρανζίστορ και τα φωτοβολταϊκά κύτταρα.



Εικόνα 35: Φωτοτρανζίστορ



Εικόνα 36: Φωτοαντίσταση

Οι αισθητήρες ήχου ανιχνεύουν τους ήχους και επιστρέφουν πίσω ηλεκτρική τάση ανάλογη με το επίπεδο του ήχου που λαμβάνουν [13].



Εικόνα 37: Αισθητήρας ήχου

Οι αισθητήρες θερμοκρασίας είναι ηλεκτρονικές συσκευές που μετρούν τη θερμοκρασία του περιβάλλοντος και μετατρέπουν τα δεδομένα εισόδου σε ηλεκτρονικά δεδομένα για καταγραφή και παρακολούθηση.

Επίσης, από τους πιο κοινούς αισθητήρες στην ρομποτική είναι οι αισθητήρες αφής ή επαφής. Χρησιμοποιούνται ευρέως για την ανίχνευση εμποδίων που βρίσκονται στο περιβάλλον τους, ή αντικειμένων που βρίσκονται εντός της αρπάγης των ρομποτικών βραχιόνων.

Επιπρόσθετα στα υδραυλικά ρομπότ υπάρχουν υγρά υπό πίεσης, και για το λόγο αυτό χρησιμοποιούνται αισθητήρες πίεσης για τη ρύθμιση της κίνησης των αρθρώσεων και την ανίχνευση διαρροών [14].



Εικόνα 38: Αισθητήρας πίεσης

Τέλος, οι αισθητήρες εγγύτητας δίνουν στο ρομπότ την ικανότητα να γνωρίζει πότε βρίσκεται κοντά σε ένα αντικείμενο ή όταν κάτι είναι κοντά του χωρίς να έρθουν σε επαφή. Συχνά οι αισθητήρες αυτοί εκπέμπουν ηλεκτρομαγνητικό πεδίο ή μία δέσμη ηλεκτρομαγνητικής ακτινοβολίας και καταγράφουν τις αλλαγές του πεδίου. Οι

επαγωγικοί αισθητήρες εγγύτητας διακρίνονται σε 4 τύπους: Τους επαγωγικούς αισθητήρες που ανιχνεύουν μεταλλικά αντικείμενα χωρίς να έρχονται σε επαφή. τους χωρητικούς αισθητήρες, που ανιχνεύουν την παρουσία ή απουσία οποιουδήποτε αντικειμένου ανεξάρτητα από το υλικό βάση της αλλαγής της χωρητικότητας του ηλεκτρικού πεδίου γύρω από την ενεργή όψη του αισθητήρα, τους μαγνητικούς αισθητήρες οι οποίοι ανιχνεύουν χωρίς επαφή τις μεταβολές του μαγνητικού πεδίου και τους φωτοηλεκτρικούς αισθητήρες που ανιχνεύουν την παρουσία αντικειμένων μέσω της αντανάκλασης σε αυτά του φωτός που εκπέμπουν [15].



Εικόνα 39: Χωρητικός αισθητήρας



Εικόνα 40: Επαγωγικός αισθητήρας



Εικόνα 41: Φωτοηλεκτρικός αισθητήρας



Εικόνα 42: Μαγνητικός αισθητήρας

1.4 Προγραμματισμός

Στην συγκεκριμένη ενότητα αναλύονται κάποιες από τις βασικότερες γλώσσες προγραμματισμού στην ρομποτική. Από τις πρώτες γλώσσες ήταν η PASAL. Αν και

στην σημερινή εποχή η χρήση της έχει μειωθεί, είναι μία γλώσσα βάση για διάφορες εφαρμογές στα ρομπότ βιομηχανιών. Έπειτα, λαμβάνοντας υπόψιν την σημαντική είσοδο της ρομποτικής στην εκπαιδευτική διαδικασία η γλώσσα Scratch είναι μία “Visual” γλώσσα προγραμματισμού και απευθύνεται σε ηλικίες 8 έως 16 ετών. Το γραφικό περιβάλλον έχει δομικά στοιχεία που συνδέοντάς τα υλοποιούνται αντίστοιχες ενέργειες. Πίσω από αυτό το περιβάλλον υπάρχει συνδυασμός γλωσσών προγραμματισμού όπως είναι το Smalltalk, ActionScript και Javascript.



Εικόνα 43: Γραφικό περιβάλλον Scratch

Επιπλέον, στο κομμάτι της τεχνητής νοημοσύνης εντάσσονται οι LISP και η Prolog. Η Lisp θεωρείται παλαιότερη γλώσσα όμως πολλά από τα στοιχεία της έχουν περαστεί σε συστήματα όπως είναι το ROS (Robotic Operation System) που συνδυάζει λογισμικά και βιβλιοθήκες για την υλοποίηση ρομποτικών εφαρμογών. Από την άλλη, η Prolog είναι μία λογική γλώσσα προγραμματισμού που δίνει την δυνατότητα στους προγραμματιστές να μεταφράσουν την “γνώση” σε μορφή που ένας αλγόριθμος τεχνητής νοημοσύνης μπορεί να κατανοήσει [16].

Επίσης, ένα διαδεδομένο λογισμικό ανοιχτού κώδικα είναι το Matlab. Βοηθάει τους ρομποτικούς μηχανικούς στην ανάλυση δεδομένων, την ανάπτυξη συστημάτων ελέγχου κατά την έρευνα και επεξεργασία των δεδομένων και την δημιουργία προηγμένων γραφημάτων.

Μία πιο ειδική γλώσσα προγραμματισμού είναι η C# που είναι η κύρια γλώσσα της Microsoft Robotics Developer Studio. Είναι μία γλώσσα δύσκολη και απαιτητική αλλά απαραίτητη σε ορισμένους τομείς της ρομποτικής.

Επίσης, τρεις από τις δημοφιλέστερες γλώσσες προγραμματισμού που χρησιμοποιούνται και στην ρομποτική είναι η Java, η Python και η C/C++. Πολλά λογισμικά και συστήματα έχουν σχεδιαστεί για να υλοποιούνται με βάση την δομή αυτών των γλωσσών. Η Java προσφέρει ένα μεγάλο εύρος διεπαφών

προγραμματισμού που καθορίζουν αλληλεπιδράσεις μεταξύ διαμεσολαβητών. Έτσι μπορεί κάποιος να καταφέρει την μεταφορά ανθρώπινων εργασιών όπως είναι η ακρόαση, η ομιλία και η κίνηση σε ρομποτικά συστήματα για να μπορέσουν να τις μιμηθούν [17]. Η Python είναι μία γλώσσα υψηλού επιπέδου προγραμματισμού, γενικού σκοπού και συνδεδεμένη με τα Raspberry Pi. Στόχος της είναι να βοηθήσουν τον προγραμματιστή στην συγγραφή σαφή και λογικού κώδικα. Χρησιμοποιείται για πειραματισμούς σε διάφορους τύπους κινητών ρομπότ αλλά και προσομοιωτών. Έτσι, όπως τα Raspberry Pi χρησιμοποιούν την Python έτσι και η C/C++ έχει ενταχθεί στους μικροελεγκτές Arduino. Πολλές εφαρμογές και λογισμικά που χρησιμοποιούνται για σχεδίαση, κατασκευή και υλοποίηση των ρομπότ είναι συμβατά με την γλώσσα C/C++, κάτι που την καθιστά ένα αναγκαίο εργαλείο για έναν μηχανικό [14].

1.5 Εφαρμογές

Σε αυτήν την ενότητα αναλύονται κάποια ρομπότ και οι εφαρμογές τους όπως είναι τα βιομηχανικά, τα ερευνητικά, τα ρομπότ στην ιατρική, τα εναέρια, τα ρομπότ στο διάστημα και τα στρατιωτικά - αστυνομικά. Αρχίζοντας από τα βιομηχανικά ρομπότ υπάρχουν διαφορετικές κατηγορίες στο εσωτερικό κάθε βιομηχανίας ανάλογα με την εφαρμογή που καλείται να υλοποιήσει. Συναντώνται ρομπότ επιλογής και τοποθέτησης ή αλλιώς “pick and place” όπου επιλέγουν αντικείμενα και τα επανατοποθετούν σε διαφορετικό μέρος. Στην αυτοκινητοβιομηχανία υπάρχουν ρομπότ βαφής αλλά και κατασκευής τα οποία περιλαμβάνουν μια πληθώρα διαδικασιών όπως κόλληση και αφαίρεση υλικών, άπλωμα κόλλας αλλά και εισαγωγή ηλεκτρονικών στοιχείων σε πλακέτες. Τα ρομπότ αυτά συνήθως διαθέτουν αρκετούς βαθμούς ελευθερίας διότι πρέπει να καθορίζουν τοποθεσία, προσανατολισμό κ.α. Επιπλέον, μπορούν να είναι σε ενεργή δράση 24 ώρες την ημέρα χωρίς να μειώνουν τον ρυθμό τους. Σαν αποτέλεσμα, έχουν την δυνατότητα να δουλεύουν συνέχεια, να μην κουράζονται, να σηκώνουν μεγαλύτερα βάρη και να αναπτύσσουν μεγαλύτερες ταχύτητες συγκρίνοντάς το με τον ανθρώπινο παράγοντα.



Εικόνα 44: Ρομπότ βαφής



Εικόνα 45: Pick and place βραχίονες



Εικόνα 46: Robots Assemble car parts

Σε ότι αφορά τα ερευνητικά ρομπότ, δημιουργούνται για την μελέτη εφαρμογών όπου για τον άνθρωπο είναι δύσκολη η πρόσβαση ή και ακατάλληλη. Τέτοια ρομπότ διαθέτουν πολλούς αισθητήρες μιας και ένας βασικός τους ρόλος είναι η συλλογή δεδομένων. Χαρακτηριστικό παράδειγμα είναι, ο ρομποτικός σκύλος της Boston Dynamics, Spot ο οποίος χρησιμοποιείται για την χαρτογράφηση του εσωτερικού της κατασκευής που περικλείει τον αντιδραστήρα 4 του Τσέρνομπιλ. Συγκεκριμένα, ο Spot είναι εξοπλισμένος με αισθητήρα ακτινοβολίας έτσι ώστε να μπορεί να μετρήσει τα επίπεδα ραδιενέργειας στον εκάστοτε χώρο [7].



Εικόνα 47: Spot

Επόμενη κατηγορία ρομποτική εφαρμογής είναι τα ρομπότ στην ιατρική. Συγκεκριμένα στην χειρουργική ρομποτική συναντάται το σύστημα Da Vinci. Αποτελείται από μία κονσόλα ελέγχου και διάφορα εργαλεία που είναι ενσωματωμένα στους βραχίονες. Ο χειρουργός δεν έρχεται σε επαφή με τον άνθρωπο αλλά κάνει τις κινήσεις που θα έκανε, μέσω του Da Vinci μπορεί και κουνάει τους αντίστοιχους βραχίονες. Έτσι καταφέρνει μεγαλύτερη σταθερότητα και ακρίβεια στις επεμβάσεις, ελάχιστα περιθώρια λάθους αλλά και οι ασθενείς από πλευράς τους αναρρώνουν πιο γρήγορα λόγω των μικρότερων τομών [12] [18].



Εικόνα 48: Da Vinci

Συνεχίζοντας, τα εναέρια ρομπότ έχουν αναπτυχθεί αρκετά τα τελευταία χρόνια. Μεγάλοι κολοσσοί όπως είναι η Amazon χρησιμοποιούν εναέρια ρομπότ για την μεταφορά δεμάτων σαν “courier”, πετυχαίνοντας ταχύτερες παραδόσεις στους καταναλωτές. Αξίζει να σημειωθεί και η συνεισφορά των ρομπότ αυτών και στον τομέα της διάσωσης όπου η ανθρώπινη ζωή βρίσκεται σε κίνδυνο. Σε αυτές τις περιπτώσεις επειδή το κάθε δευτερόλεπτο έχει μεγάλη σημασία τα εναέρια ρομπότ έρχονται να βελτιώσουν τον χρόνο της ανταπόκρισης συναγερμού, του εντοπισμού και της διάσωσης [2].



Εικόνα 49: Amazon “courier

Έχοντας μιλήσει για εναέρια ρομπότ γίνεται αναφορά λίγο παραπέρα από τη Γη, στο διάστημα. Τα ρομπότ στο διάστημα εμπεριέχουν δυσκολίες κατά την υλοποίησή τους. Αυτό συμβαίνει διότι οι συνθήκες στο διάστημα είναι αντίξοες όπως η απότομη αλλαγή θερμοκρασίας, το κενό και η έλλειψη βαρύτητας. Η NASA σχεδίασε ένα ανθρωποειδές ρομπότ το Robonaut που είχε ικανότητες ισοδύναμες με έναν αστροναύτη. Είναι εφοδιασμένο με αισθητήρες θερμοκρασίας, θέσης, δύναμης, Πανεπιστήμιο Δυτικής Αττικής, Τμήμα Ηλεκτρολόγων & Ηλεκτρονικών Μηχανικών- Γ. Σακελλάρη 31

αφής και άλλους πολλούς αισθητήρες σε κάθε βραχίονα αλλά και με στολή αστροναύτη για την προστασία από τις αλλαγές της θερμοκρασίας [13].



Εικόνα 50: Robonaut

Τέλος, τα ρομπότ έχουν διεισδύσει και στον στρατό και την αστυνομία. Μερικά παραδείγματα είναι το ρομπότ DOGO. Ένα ρομπότ μάχης που διαθέτει οπλισμό και δημιουργήθηκε ως φύλακας των στρατιωτών στο πεδίο πολέμου. Σχεδιάστηκε από την General Robotics ζυγίζει περίπου 26 κιλά και μπορεί να μεταφερθεί στο ένα χέρι.



Εικόνα 51: DOGO

Πέρα από την μάχη όμως χρησιμοποιούνται και για την φύλαξη. Το Guardbot είναι ένα αμφίβιο ρομπότ παρακολούθησης και φύλαξης, κυλάει σε οποιοδήποτε έδαφος αλλά και κολυμπάει.



Εικόνα 52: Guardbot

Στο αστυνομικό πεδίο τα ρομπότ αυτά έχουν σχεδιαστεί για να κάνουν ελέγχους, συλλήψεις, επιτήρηση, απομάκρυνση συντριμμιών από σκηνές ατυχημάτων και αφόπλιση εκρηκτικών βομβών



Εικόνα 53: Bomb defusing robot



Εικόνα 54: Surveillance Robot

ΚΕΦΑΛΑΙΟ 2. Arduino

2.1 Εισαγωγή στον ελεγκτή Arduino

Το Arduino είναι μία υπολογιστική πλατφόρμα ανοικτού κώδικα που βασίζεται σε ένα εύκολο και εύχρηστο λογισμικό. Είναι μία πλακέτα η οποία διαθέτει εισόδους/εξόδους, έναν μικροελεγκτή και ένα περιβάλλον προγραμματισμού για τη σύνταξη κώδικα. Η “καρδιά” του Arduino είναι ο μικροελεγκτής ATmega της Atmel. Για τον λόγο ότι ο μικροελεγκτής παρέχει σειριακή επικοινωνία μπορεί να γίνει σύνδεση μέσω usb μεταξύ Arduino και ηλεκτρονικού υπολογιστή πετυχαίνοντας έτσι αμφίδρομη επικοινωνία. Ο προγραμματισμός του Arduino γίνεται μέσω της γλώσσας Wiring C, μία τροποποιημένη C++. Ωστόσο, περισσότερες λειτουργίες αναπτύσσονται καθημερινά μέσα από τις βιβλιοθήκες. Στο Arduino μπορούν να εισαχθούν δεδομένα μέσα από τους αισθητήρες και τους διακόπτες και αντίστοιχα να ελέγχει ηλεκτροκινητήρες, λαμπάκια και άλλα φυσικά αντικείμενα. Μερικές εκδόσεις και τα χαρακτηριστικά από τις εκδόσεις των Arduino φαίνονται στον παρακάτω Πίνακα 1.

Πίνακας 1.

Μικροελεγκτής	ATmega328 (Arduino Uno)	ATmega2560	ATmega32u4 (Arduino Lilypad)
Τάση λειτουργίας	5V	5V	3,3V
Τάση εξωτερικής τροφοδοσίας	7-12V	7-12V	3,8-5V
Όρια εξωτερικής τροφοδοσίας	6-20V	6-20V	
Ψηφιακές εισοδοι και έξοδοι (ακροδέκτες)	14 (6 υποστηρίζουν PWM)	54 (15 υποστηρίζουν PWM)	9 (υποστηρίζει 4 κανάλια PWM)
Αναλογικές Είσοδοι (ακροδέκτες)	6	16	4
Μέγιστο συνεχές ρεύμα ανά ακροδέκτη	40mA	40mA	40mA

εισόδου/ εξόδου			
Μέγιστο ρεύμα ακροδέκτη 3,3Ω	50mA	50mA	
Memory	32KB (0.5KB bootloader)	265KB (8KB bootloader)	32KB (4KB bootloader)
SRAM	2KB	8KB	2.5KB
EEPROM	1KB	4KB	1KB
Ταχύτητα ρολογιού	16MHz	16MHz	9MHz

2.2 Πλεονεκτήματα Arduino

Στην υποενότητα αυτή αναφέρονται κάποια από τα πλεονεκτήματα των Arduino μιας και στην παρούσα διπλωματική εργασία το ρομποτικό όχημα βασίζεται στο Arduino Uno.

- Είναι ανεξάρτητη πλατφόρμα

Το Arduino μπορεί να εγκατασταθεί σε όλα τα λειτουργικά συστήματα (Windows, Mac OS, Linux) χωρίς να περιορίζονται οι χρήστες.

- Εκδόσεις

Ανάλογα με την εφαρμογή συναντάμε και μία διαφορετική έκδοση του Arduino που καλύπτει διαφορετικές ανάγκες και προσφέρει διαφορετικές δυνατότητας. Έτσι επιτυγχάνεται ευκολότερα και ταχύτερα μια κατασκευή - εφαρμογή.



Εικόνα 55: Arduino boards

- Πλακέτες επέκτασης (shields)

Οι πλακέτες επέκτασης μπαίνουν πάνω από το Arduino στα αντίστοιχα pins για να επεκτείνουν την λειτουργία του Arduino και ταυτόχρονα εισάγουν νέες δυνατότητες. Μερικές από τις επεκτάσεις είναι: ethernet shield, Wi-Fi shield, Wave shield, GPS shield, Motor shield, Proto shield, Shields οθόνης.



Εικόνα 56: Arduino Shields

- Ανοιχτός κώδικας και επέκταση λογισμικού (Software)

Ο κάθε χρήστης μπορεί μέσα από τις βιβλιοθήκες να βρει έτοιμους κώδικες, να τους τροποποιήσει για την εφαρμογή του αλλά ταυτόχρονα μπορεί να γράψει τον δικό του κώδικα και να το μοιραστεί με άλλους χρήστες.

- Επέκταση υλικού (hardware)

Ανά διαστήματα οι πλακέτες αναβαθμίζονται και επεκτείνονται δίνοντας την δυνατότητα στους χρήστες ανά πάσα στιγμή να μπορούν να εγκαταστήσουν την τελευταία έκδοση στο δικό του Arduino.

- Χαμηλό κόστος

Στην αγορά συναντάται μία μεγάλη γκάμα από την οικογένεια των Arduino με τιμές προσιτές. Πολλές φορές συνδυάζονται με άλλα εξαρτήματα και εργαλεία δημιουργώντας ένα kit που μπορεί να ξεκινήσει από την τάξη των 30 ευρώ δίνοντας την δυνατότητα στους χρήστες να ξεκινήσουν τις πρώτες τους εφαρμογές. Τα kit αυτά διαθέτουν το απαραίτητο καλώδιο usb για την σύνδεση με τον υπολογιστή, leds, καλώδια, τρανζίστορ, μπουτόν αντιστάσεις κλπ.



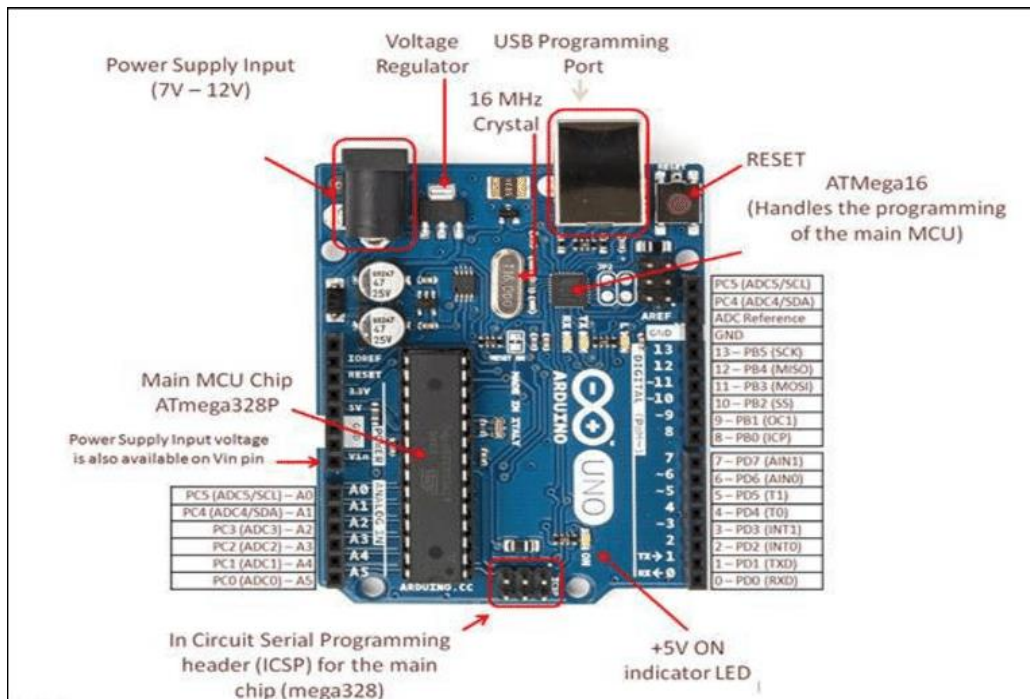
Εικόνα 57: Arduino Uno



Εικόνα 58: Arduino Starter Kit

2.3 Χαρακτηριστικά Arduino

Όπως αναφέρθηκε και παραπάνω οι εκδόσεις του Arduino ποικίλουν αλλά για τον σκοπό της διπλωματικής εργασίας στην ενότητα αυτή αναλύονται τα χαρακτηριστικά του Arduino Uno.



Εικόνα 59: Arduino Uno (pins)

Ξεκινώντας όπως φαίνεται στην παραπάνω Εικόνα 2.3-1 στην αριστερή πλευρά της πλακέτας υπάρχει μία θύρα με 8 pins. Εκεί διακρίνονται οι:

- Ακροδέκτες τροφοδοσίας, γείωσης και επανεκκίνησης.

Ο πρώτος ακροδέκτης δεν συνδέεται. Ο δεύτερος παρέχει μία λογική τάση αναφοράς για τις επεκτάσεις (shields) που χρησιμοποιούνται. Ο τρίτος έχει την ένδειξη reset, που παρέχει την δυνατότητα επανεκκίνησης του Arduino σε περίπτωση που συνδεθεί μαζί με έναν ακροδέκτη ακόμη που φέρει την ένδειξη GND (γείωση), αλλά μπορεί να γίνει και χειροκίνητα πατώντας το κουμπί στην δεξιά γωνία που φέρει και εκείνο την ένδειξη reset. Ο τέταρτος και πέμπτος ακροδέκτης δίνουν αντίστοιχα τάση της τάξεως των 3,3V και 5V με μέγιστη ένταση τα 50mA και τα 40mA. Ο έκτος και ο έβδομος είναι οι γειώσεις του Arduino και ο τελευταίος με την ένδειξη Vin παρέχει δύο δυνατότητες ανάλογα με την τροφοδοσία που δουλεύει το Arduino. Στην περίπτωση που παρέχεται τροφοδοσία στο Arduino μέσω της θύρας usb ή της υποδοχής, τότε μπορεί να τροφοδοτήσει τα εξαρτήματα με την πλήρη τάση της εξωτερικής πηγής (7-12V) αλλιώς αν συνδεθεί μαζί με έναν ακροδέκτη γείωσης μπορεί να λειτουργήσει σαν εξωτερική πηγή για την πλακέτα.

Στην συνέχεια ακριβώς από κάτω είναι η θύρα των 6 αναλογικών ακροδεκτών.

- Αναλογικοί ακροδέκτες

Οι ακροδέκτες αυτοί φέρουν την σήμανση (analog in) και είναι αριθμημένοι από το 0 έως το 5 και με αναγνώριση τάσης από το Arduino τα 0-5V. Για να λάβουν το Πανεπιστήμιο Δυτικής Αττικής, Τμήμα Ηλεκτρολόγων & Ηλεκτρονικών Μηχανικών- Γ. Σακελλάρη 38

αναλογικό σήμα οι ακροδέκτες ο μικροελεγκτής έχει ενσωματωμένο έναν αναλογικό μετατροπέα (ADC).

Από την απέναντι ακριβώς πλευρά υπάρχουν 18 ακροδέκτες εκ των οποίων οι 12 είναι ψηφιακοί ακροδέκτες.

- Ψηφιακοί ακροδέκτες

Οι ψηφιακοί ακροδέκτες ξεκινάνε από τον ακροδέκτη με την αρίθμηση 2 και φτάνουν έως τον 13. Η τάση λειτουργίας τους είναι στα 5V και παρέχουν τάση έως και 40mA. Επιπρόσθετα, τους ψηφιακούς ακροδέκτες έχουμε την δυνατότητα να τους ρυθμίσουμε είτε σαν εισόδους είτε εξόδους. Και στις δύο αυτές περιπτώσεις επειδή μιλάμε για αναλογικό σήμα οι καταστάσεις δίνουμε ή παίρνουμε αντίστοιχα είναι δύο. Την κατάσταση Low (0V) και High (5V). Ωστόσο μερικοί από αυτούς τους ακροδέκτες διαθέτουν και περισσότερες λειτουργίες, για παράδειγμα ο ακροδέκτης 2 και 3 έχουν την δυνατότητα να λειτουργήσουν και σαν εξωτερικά interrupt, δηλαδή με τον προγραμματισμό τους από τους χρήστες, αν υπάρξει αλλαγή στην προκαθορισμένη είσοδο να διακόπτουν την ροή και να εκτελείται μία άλλη ροή διαδικασίας. Οι ακροδέκτες 3, 5, 6, 9, 10, 11 έχουν την δυνατότητα λειτουργίας μέσω διαμόρφωσης πλάτους (PWM) παρέχοντας την δυνατότητα π.χ. ρύθμιση της ταχύτητα ενός κινητήρα ανάλογα της αυξομείωσης του πλάτους. Το Arduino διαθέτει στις βιβλιοθήκες την εντολή (analogWrite) η οποία έχει κλίμακα 0-255 και η τιμή αυτή αντιστοιχεί στον κύκλο εργασίας. Ωστόσο, στους ακροδέκτες αυτούς υπάρχει ο 0 και 1 και ο 13 που η χρησιμότητα τους διαφέρει. Ο ακροδέκτης 13 ελέγχει ένα led πάνω στην πλακέτα που όταν είναι αναμμένο δηλώνει ότι έχουμε HIGH τιμή (On) ενώ αντίθετα όταν είναι σβηστό ο ακροδέκτης έχει τιμή LOW (Off). Τέλος, οι ακροδέκτες 0 (RX) και 1 (TX) είναι για την επικοινωνία του Arduino. Όταν συνδέεται το Arduino για τον προγραμματισμό ο ακροδέκτης 0 δέχεται δεδομένα προς το Arduino ενώ ο 1 στέλνει δεδομένα. Στους δύο αυτούς ακροδέκτες αντιστοιχούν δύο μικρά led που είναι ακριβώς από κάτω από τον ATmega16 όπως φαίνεται στο (Figure 58) τα οποία όταν αναβοσβήνουν σημαίνει ότι γίνεται ανταλλαγή δεδομένων είτε προς το Arduino είτε από το Arduino.

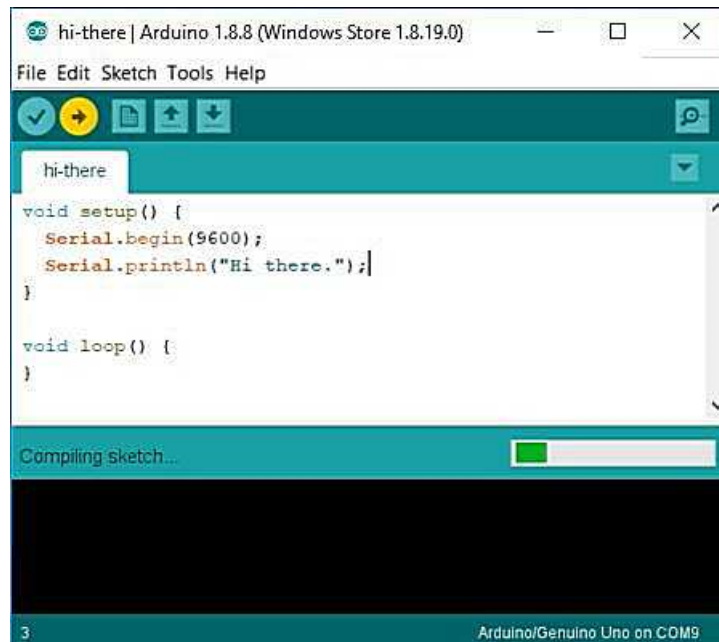
- Τροφοδοσία

Η τροφοδοσία του Arduino γίνεται με τρεις τρόπους: μέσω της θύρας usb, της 2.1mm υποδοχής και με τους ακροδέκτες στα αντίστοιχα pins της αριστερής πλευράς του Arduino (Figure 58). Τα ιδανικά Volts τροφοδοσίας είναι τα 9 Volt και προτιμάται να γίνεται μέσω της υποδοχής για βέλτιστη λειτουργία.

2.4 Προγραμματισμός ελεγκτή Arduino

Ο κώδικας σε ένα Arduino γράφεται σε γλώσσα C++ και με τον συνδυασμό κάποιων εντολών και λειτουργιών που αναφέρονται παρακάτω. Η γλώσσα C++ είναι

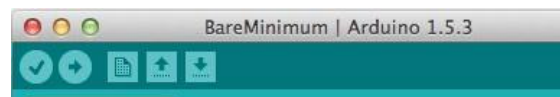
αναγνώσιμη από τον άνθρωπο. Μόλις δημιουργηθεί κάποιο <<σκίτσο>> στο Arduino ο κώδικας υποβάλλεται σε επεξεργασία και ταυτόχρονα μεταγλωττίζεται σε γλώσσα μηχανής. Σαν <<σκίτσο>> εννοείται το κύριο πρόγραμμα επεξεργασίας κειμένου που χρησιμοποιείται για τον προγραμματισμό του Arduino. Στο πεδίο αυτό πληκτρολογείτε ο κώδικας πριν ανέβει.



Εικόνα 60: Arduino Integrated Development Environment

Όπως φαίνεται στην εικόνα, το <<σκίτσο>> ή αλλιώς το IDE (Integrated Development Environment) έχει 5 επικεφαλίδες στη γραμμή μενού, καθώς και μία σειρά από κουμπιά. Στην συνέχεια, για να μπορέσει να λειτουργήσει σωστά ο προγραμματισμός, το Arduino έχει ενσωματωμένες βιβλιοθήκες που παρέχουν βασικές λειτουργίες. Αρκετές φορές κατά τον προγραμματισμό είναι απαραίτητο να εισαχθούν χειροκίνητα επιπρόσθετες βιβλιοθήκες για να μπορέσει να λειτουργήσει το πρόγραμμα. Κάθε φορά που γίνεται χρήση κάποιας βιβλιοθήκης είναι απαραίτητο να εισάγεται η εντολή `#include`. Έπειτα, πρέπει να δηλώνονται οι μεταβλητές που θα χρησιμοποιηθούν ούτως ώστε να αποθηκεύονται και να είναι έτοιμες για μελλοντική χρήση. Για την δήλωση μίας μεταβλητής χρησιμοποιείται η εντολή `(int)` και ορίζεται ο τύπος, το όνομα και η αρχική τιμή. Κάθε σκίτσο πρέπει να διαθέτει ένα `setup function`. Η συνάρτηση αυτή καθορίζει την αρχική κατάσταση του Arduino κατά την εκκίνηση και εκτελείται μόνο μία φορά και συντάσσεται με την εντολή `void setup()`. Μία ακόμη λειτουργία που είναι απαραίτητη κατά τον προγραμματισμό είναι η λειτουργία βρόχου. Εκτελείται μόλις ολοκληρωθεί η `setup()`, ορίζεται με την εντολή `void loop()` και η κύρια λειτουργία της είναι να επαναλαμβάνεται συνέχεια. Περνώντας στο βασικό μέρος του κώδικα η βασική λογική είναι μία δομή <<if-then>>. Για να υλοποιηθεί μία τέτοια δομή γίνεται η εξής διεργασία: κάτω από το `void setup()`

γράφονται οι ρυθμίσεις για πράγματα που θα εκτελεστούν μόνο μία φορά (π.χ. βαθμονόμηση αισθητήρα). Ύστερα, στην αρχή του βρόχου διαβάζονται οι είσοδοι όπου και θα χρησιμοποιηθούν σαν συνθήκες (if), μετατρέπονται τα δεδομένα και υπολογίζονται αντίστοιχες μεταβλητές και στο τέλος στην έξοδο καθορίζεται το τελικό αποτέλεσμα της λογικής (then) βάση των αποτελεσμάτων που υπολογίστηκαν. Εφόσον το πρόγραμμα έχει ολοκληρωθεί περνάμε στην διαδικασία ελέγχου. Πρέπει ο κώδικας να ελεγχθεί για συντακτικά λάθη πατώντας το κουμπί “compile”. Εάν δεν υπάρξουν λάθη ο προγραμματιστής λαμβάνει μήνυμα στην οθόνη του ότι ο έλεγχος ολοκληρώθηκε με επιτυχία χωρίς να διαπιστωθεί κάποιο λάθος. Στην περίπτωση που βρεθεί λάθος, εμφανίζεται ειδοποίηση και στο σημείο του κώδικα που υπάρχει πρόβλημα επισημαίνεται ούτως ώστε ο προγραμματιστής να προβεί στις εκάστοτε διορθώσεις. Στην συνέχεια πρέπει να προβεί στην ίδια διαδικασία ελέγχου μέχρι ο έλεγχος να μην βρίσκει λάθη. Τέλος, πατάμε το κουμπί “upload” για να μεταφέρουμε την πληροφορία στο Arduino.



Εικόνα 61: Κουμπιά IDE

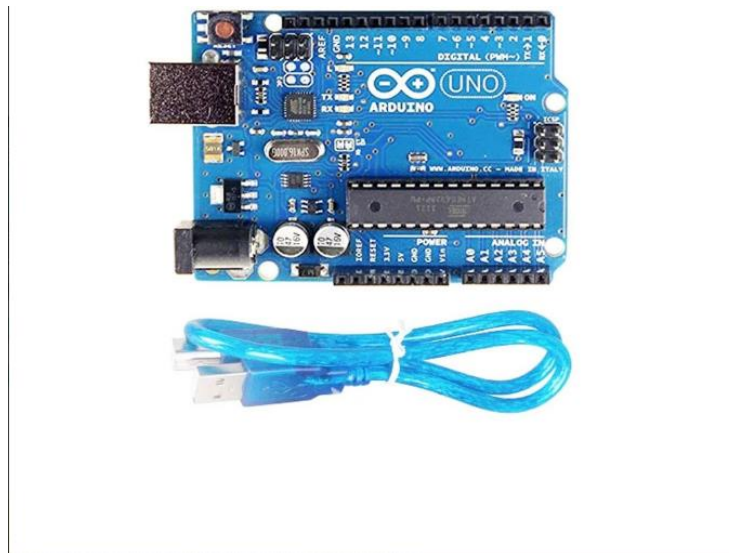
Στην Εικόνα 61 φαίνονται τα βασικά κουμπιά του IDE στα οποία το πρώτο είναι ο έλεγχος (compile) και το αμέσως επόμενο το ανέβασμα του προγράμματος στο Arduino (upload). Ενώ τα επόμενα τρία στην σειρά έχουν να κάνουν με την δημιουργία νέου «σκίτσου», το άνοιγμα κάποιου προηγούμενου και τέλος η αποθήκευση του αρχείου.

ΚΕΦΑΛΑΙΟ 3. Ανάπτυξη Εφαρμογής

3.1 Υλικά Κατασκευής

Για τον σχεδιασμό του ρομποτικού οχήματος χρησιμοποιήθηκαν υλικά βάση του κόστους και της αντίστοιχης χρήσης για την οποία προορίζεται το όχημα. Αρχικά η “καρδιά” της κατασκευής είναι:

- Arduino Uno



Εικόνα 62: Arduino Uno

- Πλαίσιο



Εικόνα 63: Πλαίσιο (σασί)

Το συγκεκριμένο πλαίσιο είναι της εταιρίας Original Equipment Manufacturer (OEM), είναι από αλουμίνιο, έχει μήκος 25cm, πλάτος 17cm, ύψος 6cm και διαθέτει όλες τις βίδες και τα παξιμάδια για να μπορέσει να προσαρμοσθεί το ένα πλαίσιο πάνω στο άλλο.

- Τροχοί



Εικόνα 64: Τροχοί Mecanum

Για την κατασκευή μας δεν μπορούσαν να τοποθετηθούν απλοί τροχούς διότι σκοπός είναι η επίτευξη της πλάγιας αριστερής και δεξιάς κίνησης. Για τον λόγο αυτό, τοποθετήθηκαν τροχοί Mecanum, οι οποίοι είναι πανκατευθυντικοί. Οι συγκεκριμένοι τροχοί έχουν διάμετρο 60mm και είναι επίσης της εταιρίας (OEM).



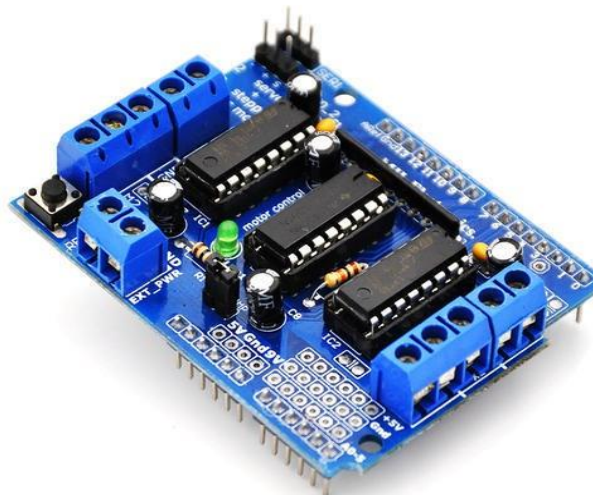
- Κινητήρες
Motor TT

Εικόνα 65: DC Gear

Γνωρίζοντας τις αντίστοιχες τιμές για το βάρος του πλαισίου, τις διαστάσεις των τροχών και των τριβών, οι απλοί DC Gear Motor TT ήταν σε θέση να κινήσουν το όχημα μας. Λαμβάνοντας υπόψιν ότι το όχημα χρειάζεται 4 τροχούς για να κινηθεί έγινε προμήθεια και 4 κινητήρων. Κάποιες ονομαστικές τιμές για τους συγκεκριμένους κινητήρες φαίνονται παρακάτω.

1. Τάση: 3 Volt
2. Ρεύμα: 100mA
3. Ταχύτητα: 125RPM
4. Ροπή εξόδου: 0.8kg.cm
5. Reduction: 48:1

- Επεκτάσεις (shield)



Εικόνα 66: Shield L293D

Για να τεθούν οι κινητήρες σε λειτουργία χρειάζεται και μία επέκταση (shield). Το Motor Shield L293D είναι ένας οδηγός κινητήρα H-Bridge διπλού καναλιού. Μπορεί έτσι εύκολα να οδηγεί μεμονωμένα έως και 4 κινητήρες DC καθιστώντας το ιδανικό για κατασκευή τετρακίνητων πλατφορμών ρομπότ.

- Αισθητήρας υπερήχων



Εικόνα 67: HC SR04 Sensor

Για την ανίχνευση των εμποδίων χρειάστηκε ένας αισθητήρα υπερήχων, ο Ultrasonic Distance Sensor – HC – SR04. Έχει κατασκευαστεί από την Elec Freaks και χρησιμοποιεί το sonar για να προσδιορίσει την απόσταση ενός αντικειμένου. Προσφέρει εξαιρετική ανίχνευση εύρους χωρίς επαφή, με υψηλή ακρίβεια και σταθερές μετρήσεις σε ένα εύρος από 2cm έως και 400cm.

- Βάση αισθητήρα υπερήχων



Εικόνα 68: Βάση HC SR04 Sensor

Για την σωστή στήριξη του αισθητήρα προστέθηκε μία βάση που είναι σχεδιασμένη για τους συγκεκριμένους αισθητήρες.

- Micro Servo – Motor



Εικόνα 69: SG90 Micro Servo

Για τον έλεγχο της αριστερής και δεξιάς πλευράς από τον αισθητήρα υπερήχων ενσωματώθηκε ένας σέρβο κινητήρα έτσι ώστε να μπορεί να περιστρέφεται ο αισθητήρας. Οι σέρβο κινητήρες είναι μικροί ελεγκτές που δίνουν την δυνατότητα μέσω του Arduino να προγραμματιστούν ώστε να περιστρέφονται σε συγκεκριμένες γωνίες, με συγκεκριμένη ταχύτητα και για τον χρόνο που εμείς θέλουμε. Για την

κατασκευή μας χρησιμοποιήθηκε ο SG90 Micro Servo της εταιρίας Tower Pro. Μαζί με τον κινητήρα είναι ενσωματωμένο το 3 pin καλώδιο (ισχύ, γείωση, έλεγχος).

- Διακόπτης On – Off



Εικόνα 70: Διακόπτης

- Καλώδια



Εικόνα 71: M-F , M-M Καλώδια

- Μπαταρίες



Εικόνα 72: Μπαταρία

Χρειάζονται δύο μπαταρίες για την κατασκευή μας. Μία για την απευθείας τροφοδοσία του Arduino μέσω της 2.1mm εισόδου και μία για την τροφοδοσία της επέκτασης (shield).

- Battery connector



Εικόνα 73: Σύνδεσμος μπαταρίας

- Μπαστούνι

Για να καθοδηγεί το όχημα τον εκάστοτε χειρίστη ενσωματώθηκε ένα μπαστούνι στο πίσω μέρος του πλαισίου το οποίο για ευκολία της κατασκευής είναι ένα μπαστούνι το οποίο τοποθετούνται τα κινητά τηλέφωνα για την λήψη φωτογραφίας. Το συγκεκριμένο μπαστούνι εξυπηρετεί στο να παρέχει την επιθυμητή κλίση μεταξύ χειριστή και οχήματος.



Εικόνα 74: Μπαστούνι

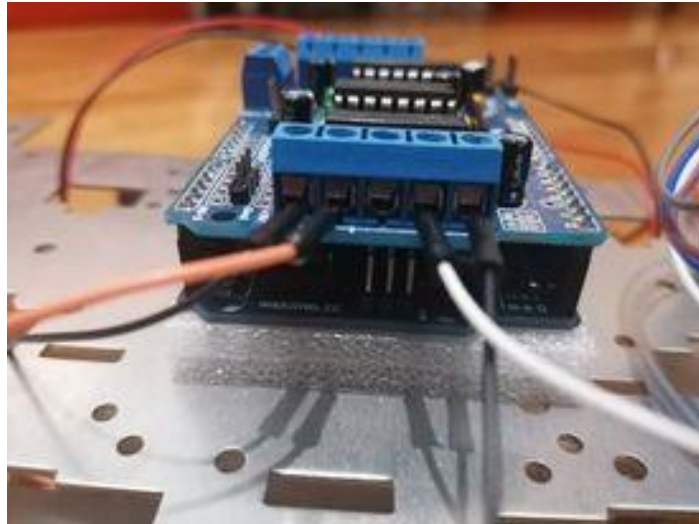
3.2 Συνδεσμολογία Κατασκευής

Στην ενότητα αυτή περιγράφεται η διαδικασία με την οποία συνδέεται το κάθε εξάρτημα για να υλοποιηθεί το τελικό αποτέλεσμα της κατασκευής. Αρχικά, το πλαίσιο που χρησιμοποιήθηκε διέθετε όλα τα απαραίτητα (βίδες, παξιμάδια) τα οποία μπήκαν στις αντίστοιχες θέσεις ούτως ώστε να πάρουμε το επιθυμητό αποτέλεσμα του κορμού. Έπειτα, επειδή οι 4 κινητήρες ήταν μεμονωμένοι χωρίς πρόσθετο καλώδιο, κολλήσαμε με το καλαί στις δύο θέσεις που είχε από ένα καλώδιο.



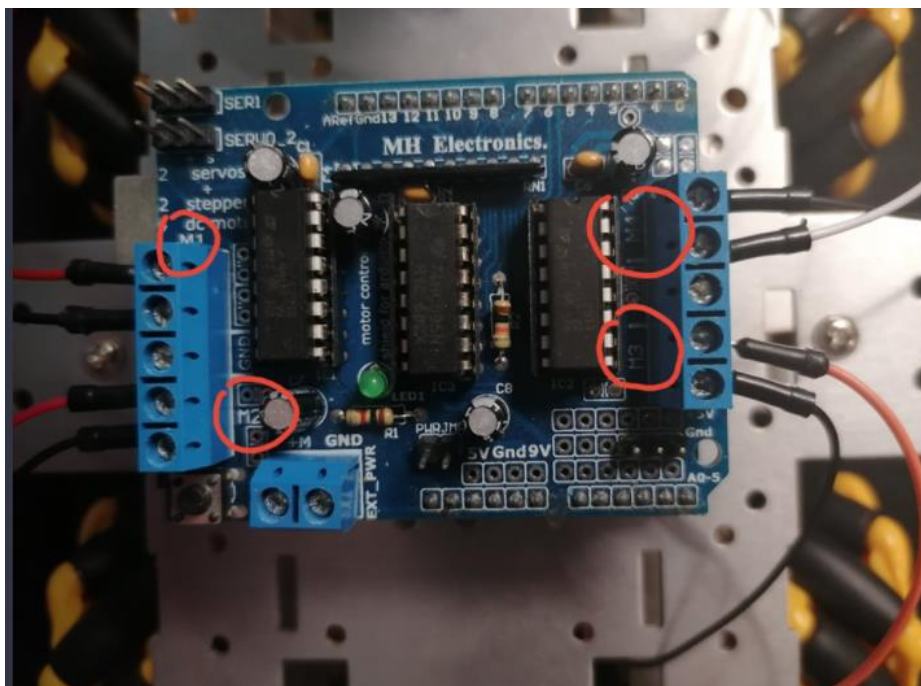
Εικόνα 75: Κόλληση καλωδίων - Κινητήρα

Στην συνέχεια, με την ενσωμάτωση των 4 κινητήρων κάτω από το πλαίσιο, συνδέθηκαν πάνω στους τροχούς. Όπως είχαμε αναφέρει και προηγουμένως χρησιμοποιήθηκαν πανκατευθυντικοί τροχοί και για την επίτευξη της πλάγιας κίνησης δεν χρειάζεται μόνο ο σωστός προγραμματισμός αλλά και η σωστή τοποθέτηση των τροχών. Δηλαδή, ο πάνω αριστερά τροχός πρέπει να έχει την ίδια φορά με τον κάτω δεξιά και ο πάνω δεξιά με τον κάτω αριστερά. Με την ολοκλήρωση του πλαισίου και των περιφερειακών στοιχείων συνέχεια έχουν τα ηλεκτρονικά εξαρτήματα. Ξεκινώντας, το Arduino Uno τοποθετήθηκε περίπου στην μέση του πλαισίου για την καλύτερη σύνδεση των καλωδίων από τους κινητήρες προς το Arduino. Για να λειτουργήσουν οι κινητήρες, ενσωματώθηκε το motor shield πάνω στο Arduino. Η “ασπίδα” έχει ακριβώς ίδια pins με το Arduino ούτως ώστε το κάθε ένα να μπαίνει στην αντίστοιχη θέση για να μπορέσει να λειτουργήσει σωστά. Η συγκεκριμένη “ασπίδα” υποστηρίζει 4 κινητήρες, έχει στις δύο άκρες 2 κλέμες στις οποίες οδηγούνται τα καλώδια των κινητήρων.



Εικόνα 76: Πλαϊνό μέρος σύνδεσης κινητήρων

Στην παραπάνω εικόνα φαίνεται η μία πλευρά της “ασπίδας” και παρατηρείται ότι έχει 5 θέσεις η κλέμα αφήνοντας την 3η κενή επειδή είναι η γείωση. Κάθε κλέμα έχει τις αντίστοιχες ένδειξης M1, M2, M3, M4. Στις θέσεις αυτές μπαίνουν τα καλώδια έτσι ώστε όταν δοθεί τροφοδοσία να μπουν σε λειτουργία οι κινητήρες. Ωστόσο, πριν την έναρξη του προγραμματισμού έγινε δοκιμή αν οι κινητήρες λειτουργούν σωστά και έπειτα αν η κίνηση που μας έδινε το όχημα ήταν σωστή και όχι ανάποδη. Στην περίπτωση ανάποδης κίνησης το μόνο που έπρεπε να γίνει ήταν να αντιστραφούν απλώς οι θέσεις των καλωδίων μεταξύ τους για να αλλάξει η πολικότητα.



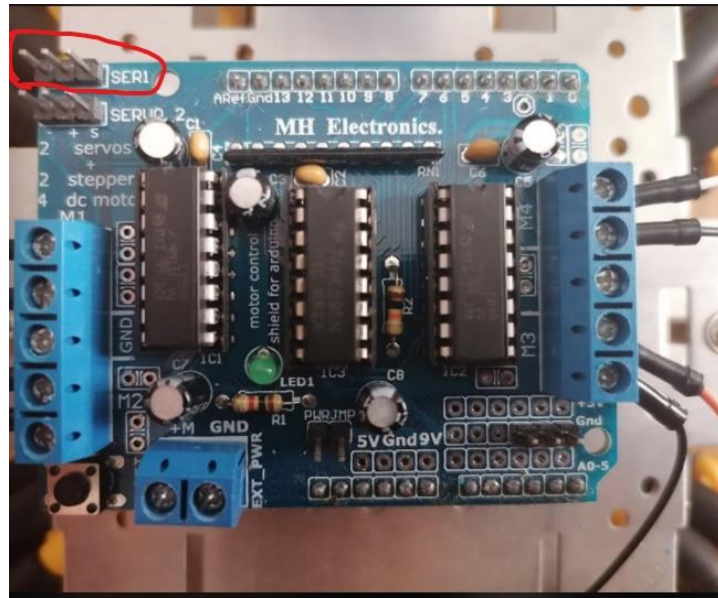
Εικόνα 77: Ενδείξεις θέσεων κινητήρων

Στις θέσεις M1 και M2 οδηγήθηκαν οι κινητήρες της αριστερής πλευράς του πλαισίου, δηλαδή του μπροστά αριστερά τροχού και του πίσω αριστερά και στις M4 και M3 αντίστοιχα του τροχού μπροστά δεξιά και πίσω δεξιά. Στην συνέχεια, σειρά είχε ο αισθητήρας υπερήχων. Στην αρχή, με την βοήθεια της θερμόκολλας ο αισθητήρας κόλλησε επάνω στην ειδική βάση για να μπορεί στην συνέχεια να στηριχτεί επάνω στον σέρβο κινητήρα. Μετά την υλοποίησή του συγκεκριμένου βήματος κολλήθηκε και ο σέρβο κινητήρας στο μπροστινό μέρος του πλαισίου.



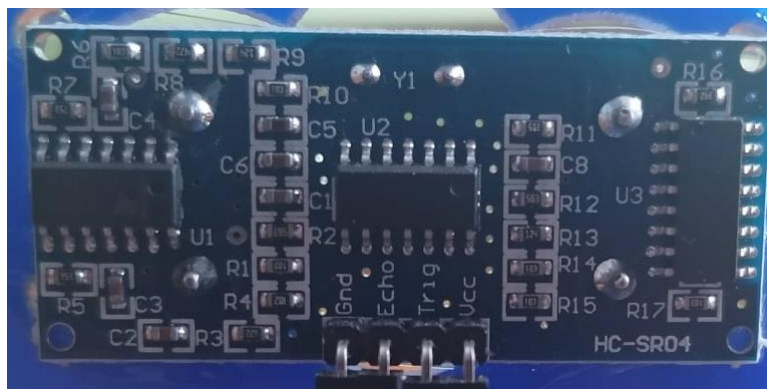
Εικόνα 78: Μπροστινό μέρος αισθητήρα

Ο σέρβο κινητήρας έχει ένα τριπλό καλώδιο με τους εξής χρωματισμούς. Καφέ για το αρνητικό, κόκκινο για το θετικό και πορτοκαλί για το σήμα. Η “ασπίδα” διαθέτει συγκεκριμένη θέση στην οποία συνδέεται ο σέρβο κινητήρας και ο χρωματισμός μπαίνει με την παραπάνω σειρά αναφοράς από τα αριστερά προς τα δεξιά και όχι με διαφορετικό τρόπο.



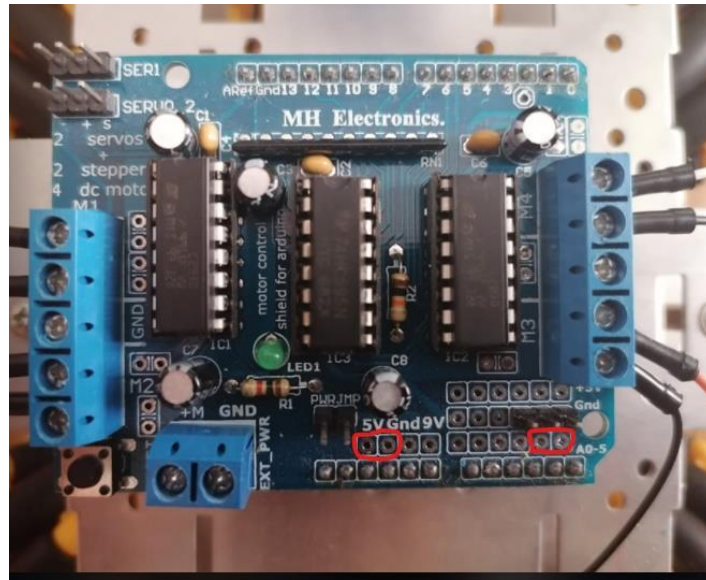
Εικόνα 79: Θέση σέρβο κινητήρα

Έπειτα, συνδέθηκε και ο αισθητήρας πάνω στην “ασπίδα”.



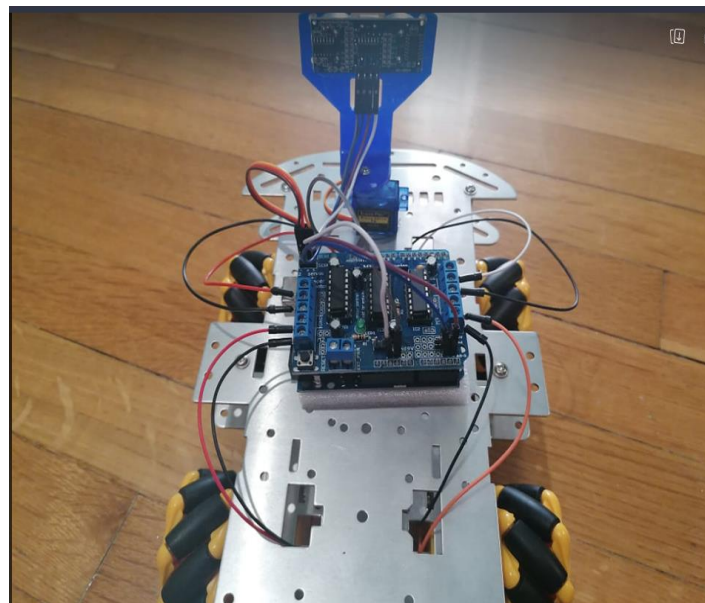
Εικόνα 80: Πίσω μέρος αισθητήρα

Όπως και τα προηγούμενα στοιχεία και ο αισθητήρας πρέπει να συνδεθούν σε συγκεκριμένες θέσεις για να αντιστοιχεί το (Gnd, Echo, Trig, Vcc).



Εικόνα 81: Υποδοχές σύνδεσης αισθητήρα

Το Vcc πηγαίνει στην θέση 5V, το Gnd στην αντίστοιχη Gnd και στην συνέχεια το Trig στο A0 και το Echo στο A1 στις δύο αυτές αναλογικές θέσεις έτσι ώστε να υπάρχει επικοινωνία με τον αισθητήρα όταν εντοπίζει τα εμπόδια. Με την ολοκλήρωση των διαδικασιών αυτών καταλήγουν οι παρακάτω εικόνες.



Εικόνα 82: Πίσω μέρος κατασκευής



Εικόνα 83 Μπροστά μέρος κατασκευής

Ύστερα, ενσωματώθηκε στο πίσω μέρος του οχήματος μαστούνι τύπου (selfie stick) για να μπορεί να καθοδηγείται ο χειριστής. Το μαστούνι αυτό τυλίχθηκε με δύο καλώδια, ούτως ώστε το όχημα να μπαίνει σε λειτουργία μέσω ενός διακόπτη On/Off.



Εικόνα 84 τελικές εικόνες με μαστούνι

Για την τροφοδοσία της κατασκευής, το Arduino τροφοδοτήθηκε με 9V μπαταρία μέσω της 2.1mm εισόδου και ξεχωριστά η ασπίδα με εξωτερική 9V μπαταρία η οποία ενσωματώθηκε στο μαστούνι.

Τέλος, για να τεθεί σε λειτουργία ανα πάσα στιγμή το ρομποτικό όχημα συνδέθηκε και ένας διακόπτης On/Off για πρακτική διευκόλυνση.

3.3 Προγραμματισμός Κατασκευής

Στην ενότητα αυτή πριν διατυπωθεί ο κώδικας, περιγράφεται και ο τρόπος λειτουργίας του οχήματος. Έστω ότι το όχημα βρίσκεται μέσα σε ένα δωμάτιο, καλείται να ξεκινήσει ευθεία πορεία. Στην περίπτωση που εντοπίσει εμπόδιο πρέπει να αποφασίσει εάν θα κάνει αριστερή κίνηση ή δεξιά ώστε να το αποφύγει και να συνεχίσει ευθεία πορεία. Για την υλοποίηση αυτής της δυνατότητας τοποθετήθηκε ένας αισθητήρας υπερήχων στο μπροστινό μέρος ο οποίος στηρίζεται σε μία βάση, η

οποία δίνει την δυνατότητα περιστροφής του, με την βοήθεια ενός σέρβο – κινητήρα. Ο σέρβο – κινητήρας είναι προγραμματισμένος να περιστρέφεται από το σημείο 0 που βρίσκεται στις εξής μοίρες: ~~XXXX~~, δίνοντας έτσι την δυνατότητα στην περίπτωση που βρεθεί εμπόδιο μπροστά να μετρήσει στις άλλες πλευρές που δεν εντοπίζει πιθανό εμπόδιο για να μπορέσει να κατευθυνθεί προς εκείνη την μεριά και έπειτα να συνεχίσει την πορεία του. Για να μπορέσει να λειτουργήσει ο κώδικας αρχικά δηλώθηκαν οι αντίστοιχες βιβλιοθήκες.

```
#include <AFMotor.h> //φορτώνουμε την βιβλιοθήκη για τους κινητήρες
#include <Servo.h> //φορτώνουμε την βιβλιοθήκη για τον σέρβο κινητήρα
#include <NewPing.h> //φορτώνουμε την βιβλιοθήκη για τον αισθητήρα Ultrasonic
```

Στην συνέχεια δηλώθηκαν τα pin σύνδεσης του αισθητήρα υπερήχων αλλά και κάποιες εντολές οι οποίες θέτουν την ταχύτητα λειτουργίας των κινητήρων και την απόσταση με την οποία ο αισθητήρας εντοπίζει το εμπόδιο.

```
#define TRIG_PIN A0 // το TRIG PIN από τον αισθητήρα υπερήχων του ορίζουμε σαν
θέση το Pin A0 της ασπίδας

#define ECHO_PIN A1 // το ECHO PIN από τον αισθητήρα υπερήχων του ορίζουμε σαν
θέση το Pin A1 της ασπίδας

#define MAX_DISTANCE 280 // ορίζουμε μέγιστη απόσταση ελέγχου του αισθητήρα
τα 280cm

#define MAX_SPEED 230 // ορίζουμε ταχύτητα κινητήρων τα 230 rpm

#define MAX_SPEED_OFFSET 20 // ορίζουμε το offset της ταχύτητας στα 20rpm

#define COLL_DIST 5 // ορίζουμε απόσταση 5cm στην οποία το όχημα πρέπει να
σταματήσει

#define TURN_DIST COLL_DIST+20 // ορίζουμε την απόσταση όπου το ρομπότ
απομακρύνεται από το εμπόδιο

NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE); // δημιουργούμε βιβλιοθήκη
ώστε ο αισθητήρας να αντιλαμβάνεται τα σωστά pins για την μέτρηση της απόστασης
```

Έπειτα, για να μπει σε λειτουργία ο εκάστοτε κινητήρας ορίζεται το αντίστοιχο pin που βρίσκεται στην ασπίδα για να δηλωθεί σαν είσοδος.

```
AF_DCMotor leftMotor1(1, MOTOR12_1KHZ); // δημιουργούμε τους κινητήρες και  
ορίζουμε θέση M1 για τον πρώτο και 1Khz PWM συχνότητα
```

```
AF_DCMotor leftMotor2(2, MOTOR12_1KHZ); // δημιουργούμε τους κινητήρες και  
ορίζουμε θέση M2 για τον δεύτερο και 1Khz PWM συχνότητα
```

```
AF_DCMotor rightMotor1(3, MOTOR34_1KHZ); // δημιουργούμε τους κινητήρες και  
ορίζουμε θέση M3 για τον τρίτο και 1Khz PWM συχνότητα
```

```
AF_DCMotor rightMotor2(4, MOTOR34_1KHZ); // δημιουργούμε τους κινητήρες και  
ορίζουμε θέση M4 για τον τέταρτο και 1Khz PWM συχνότητα
```

Τέλος πριν ξεκινήσουν οι υπορουτίνες ορίζονται η θέση, η αρχική ταχύτητα του οχήματος και δημιουργούνται οι μεταβλητές που θα παρέχουν τιμές για την αριστερή και δεξιά απόσταση που εντοπίζεται εμπόδιο.

```
int leftDistance, rightDistance; // θέτουμε μεταβλητές για αριστερή και δεξιά  
απόσταση
```

```
int curDist = 0; // θέτουμε 0 την μεταβλητή η οποία είναι η θέση που θα ξεκινάει το  
όχημα
```

```
String motorSet = "";
```

```
int speedSet = 0; // θέτουμε αρχική ταχύτητα
```

Στην υπορουτίνα Setup έγινε ο ορισμός του pin ελέγχου του σέρβο – κινητήρα, η αρχικοποίηση της θέσης με την οποία “κοιτάει”. Ωστόσο, μετά από κάποιες εντολές δημιουργείται μία καθυστέρηση μεταξύ των επόμενων εντολών για την καλύτερη διαχείριση παρατήρησης των αποτελεσμάτων και καλύτερης λειτουργίας του προγράμματος.

```
myservo.attach(10); // ορίζουμε το Pin 10 σαν είσοδος για τον σέρβο κινητήρα
```

```
myservo.write(90); // ορίζουμε αρχική συνθήκη 90 μοιρών εκκίνησης για τον σέρβο  
κινητήρα δηλαδή να κοιτάει ευθεία
```

```
delay(1000); // δημιουργούμε καθυστέρηση ενός δευτερολέπτου
```

Προχωρώντας στο βασικό σκέλος του προγράμματος main loop ξεκινάει ο αισθητήρας καθώς το όχημα κινείται να ελέγχει την απόσταση μπροστά. Ταυτόχρονα, η αποστάσεις που λαμβάνονται συγκρίνονται με την απόσταση την οποία τέθηκε αρχικά σαν COLL_DIST. Στην περίπτωση που η απόσταση που λαμβάνει το όχημα

είναι μικρότερη του COLL_DIST, το όχημα συνεχίζει ευθεία πορεία. Σε άλλη περίπτωση το όχημα σταματάει για να ελέγξει από δεξιά και αριστερά.

```
myservo.write(90); // έλεγχος μπροστά
delay(90); // δημιουργούμε καθυστέρηση
curDist = readPing (); // ανάγνωση απόστασης
if (curDist < COLL_DIST) {
    changePath(); // με την if συγκρίνουμε την θέση με την απόσταση σύγκρουσης έτσι
    ώστε να αλλάξει πορεία το όχημα ή όχι
}
moveForward(); // το όχημα συνεχίζει την ευθεία πορεία
delay(500); δημιουργία καθυστέρησης μισού δευτερολέπτου
```

Στην περίπτωση που εντοπιστεί εμπόδιο το όχημα σταματάει και ο αισθητήρας μετράει την απόσταση και από τις δύο πλευρές αριστερά και δεξιά. Οι τιμές αυτές συγκρίνονται μεταξύ τους έτσι ώστε το όχημα να κινηθεί από την πλευρά που δεν εντοπίζει εμπόδιο ή από την πλευρά που το εμπόδιο βρίσκεται σε μεγαλύτερη απόσταση.

```
void changePath() {
    moveStop(); // σταματάει η κίνηση
    myservo.write(180); // έλεγχος απόστασης δεξιά
    delay(500); // καθυστέρηση
    rightDistance = readPing(); // διαβάζουμε την απόσταση δεξιά
    delay(500); // καθυστέρηση
    myservo.write(0); // έλεγχος απόστασης αριστερά
    delay(700); // καθυστέρηση
    leftDistance = readPing(); // διαβάζουμε την απόσταση αριστερά
    delay(500); // καθυστέρηση
    myservo.write(90); // επιστροφή του αισθητήρα στην αρχική του θέση
    delay(100); // καθυστέρηση
```

```
compareDistance(); // σύγκριση αποστάσεων
}

void compareDistance() // με την συγκεκριμένη επανάληψη γίνεται έλεγχος για της
μικρότερης απόστασης που υπάρχει εμπόδιο ή όχι
{
    if (leftDistance > rightDistance) // εάν στην δεξιά πλευρά εντοπίζεται εμπόδιο σε
μικρότερη απόσταση από ότι αριστερά
    {
        turnLeft(); // τότε το όχημα πάει αριστερά
    }
    else if (rightDistance > leftDistance) // εάν αριστερά έχει εμπόδιο σε μικρότερη
απόσταση από ότι δεξιά
    {
        turnRight(); // τότε το όχημα πάει δεξιά
    }
    else // εάν η απόσταση είναι ίδια
    {
        turnAround(); // το όχημα γυρνάει από την αντίθετη κατεύθυνση
    }
}
```

```
int readPing() { // διαβάζουμε την απόσταση που βλέπει ο αισθητήρας
    delay(70); // καθυστέρηση
    unsigned int uS = sonar.ping(); // ορισμός μεταβλητής για τον αισθητήρα
    int cm = uS / US_ROUNDTRIP_CM;
    return cm;
}
```


Τέλος, για τις βασικές κινήσεις των κινητήρων δημιουργήθηκαν 4 υπορουτίνες για την κίνηση μπροστά, πίσω, αριστερά, δεξιά και περιστροφή .

Για την κίνηση μπροστά:

```
void moveForward() {  
  
    motorSet = "FORWARD";  
  
    leftMotor1.run(FORWARD); // βάζουμε τους κινητήρες να δίνουν κίνηση μπροστά  
    και οι 4  
  
    leftMotor2.run (FORWARD);  
  
    rightMotor1.run (FORWARD);  
  
    rightMotor2.run (FORWARD);  
  
    for (speedSet = 0; speedSet < MAX_SPEED; speedSet += 2) // με την συγκεκριμένη  
    εντολή γίνεται εξοικονόμηση των μπαταριών  
  
    {  
  
        leftMotor1.setSpeed(speedSet);  
  
        leftMotor2.setSpeed(speedSet);  
  
        rightMotor1.setSpeed(speedSet);  
  
        rightMotor2.setSpeed(speedSet);  
  
        delay(5);  
  
    }  
}
```

Για την πίσω κίνηση:

```
void moveBackward() {  
  
    motorSet = "BACKWARD"; // θέτουμε τους κινητήρες να μας δώσουν πίσω κίνηση  
  
    leftMotor1.run (BACKWARD);  
  
    leftMotor2.run (BACKWARD);  
  
    rightMotor1.run (BACKWARD);
```

```
rightMotor2.run (BACKWARD);  
for (speedSet = 0; speedSet < MAX_SPEED; speedSet += 2)  
{  
    leftMotor1.setSpeed(speedSet);  
    leftMotor2.setSpeed(speedSet);  
    rightMotor1.setSpeed(speedSet);  
    rightMotor2.setSpeed(speedSet);  
    delay(5);  
}  
}
```

Για την αριστερή κίνηση:

```
void turnLeft() { // κίνηση προς τα αριστερά  
    motorSet = "LEFT";  
    leftMotor1.run(BACKWARD); // ορίζουμε τον πρώτο κινητήρα σε πίσω κίνηση  
    leftMotor2.run(FORWARD); // ορίζουμε τον δεύτερο κινητήρα αριστερά σε  
    μπροστινή κίνηση  
    rightMotor1.run(FORWARD); // ορίζουμε τον τρίτο κινητήρα από δεξιά σε  
    μπροστά κίνηση  
    rightMotor2.run(BACKWARD); // ορίζουμε τέταρτο κινητήρα από δεξιά σε πίσω  
    κίνηση  
    delay(1500); // ακολουθείται αυτή η πορεία για 1.5 δευτερόλεπτα  
    motorSet = "FORWARD";  
    leftMotor1.run (FORWARD); // επαναφέρουμε την κίνηση μπροστά  
    leftMotor2.run (FORWARD);  
    rightMotor1.run (FORWARD);  
    rightMotor2.run (FORWARD);  
}
```

Για την δεξιά κίνηση:

```
void turnRight() { // το όχημα κινείται δεξιά
```

```
motorSet = "RIGHT";  
  
leftMotor1.run(FORWARD); // θέτουμε τον κινητήρα σε μπροστά κίνηση  
leftMotor2.run(BACKWARD); // θέτουμε τον δεύτερο κινητήρα αριστερά σε πίσω  
κίνηση  
rightMotor1.run(BACKWARD); // θέτουμε τον κινητήρα πάνω δεξιά σε πίσω κίνηση  
rightMotor2.run(FORWARD); // θέτουμε τον τέταρτο κινητήρα σε μπροστά κίνηση  
delay(1500); // για χρόνο 1.5 δευτερολέπτου το όχημα ακολουθεί αυτή την κίνηση  
motorSet = "FORWARD";  
  
leftMotor1.run(FORWARD); // επαναφέρουμε όλους τους κινητήρες σε μπροστά  
κίνηση  
leftMotor2.run (FORWARD);  
rightMotor1.run (FORWARD);  
rightMotor2.run(FORWARD);  
}
```

Για την περιστροφή:

```
void turnAround() { // περιστροφή του οχήματος  
motorSet = "RIGHT";  
leftMotor1.run(FORWARD); // θέτουμε αριστερούς κινητήρες σε μπροστά κίνηση  
leftMotor2.run(FORWARD);  
rightMotor1.run(BACKWARD); // θέτουμε δεξιά κινητήρες σε πίσω κίνηση  
rightMotor2.run (BACKWARD);  
delay(1700); // ακολουθείται η πορεία για 1.7 δευτερόλεπτα  
motorSet = "FORWARD";  
leftMotor1.run(FORWARD); // επαναφέρουμε την κίνηση προς τα μπροστά  
leftMotor2.run (FORWARD);  
rightMotor1.run (FORWARD);  
rightMotor2.run (FORWARD);
```

3.4 Πειραματική Διαδικασία

Για να διαπιστωθεί η σωστή λειτουργία του προγραμματισμού του οχήματος τοποθετήθηκαν κάποια εμπόδια για την εκάστοτε περίπτωση. Αρχικά, τοποθετήθηκε εμπόδιο μπροστά από το όχημα στην ευθεία αλλά και ένα από τα αριστερά σε απόσταση 10cm από το σημείο όπου το όχημα θα πρέπει να εντοπίσει το βασικό εμπόδιο και να ελέγξει αριστερά και δεξιά του. Στην περίπτωση αυτή το όχημα πρέπει να φύγει προς τα δεξιά και έπειτα να συνεχίσει την πορεία του. Έπειτα, έγινε η ίδια ενέργεια με το ίδιο βασικό εμπόδιο αλλά αυτή την φορά με το δεύτερο εμπόδιο στα δεξιά για να διαπιστωθεί αν το όχημα στην περίπτωση αυτή θα φύγει προς τα αριστερά και έπειτα να συνεχίσει ευθεία πορεία. Στην συνέχεια, τοποθετήθηκαν δύο πλαϊνά εμπόδια πέραν του βασικού στην ευθεία αλλά στην μία περίπτωση το ένα εμπόδιο είχε την διπλάσια απόσταση από το άλλο για να ελεγχθεί κατά πόσο το όχημα θα μετρήσει σωστά τις αποστάσεις των εμποδίων και θα επιλέξει εκείνη με το εμπόδιο να βρίσκεται πιο μακριά. Το πείραμα αυτό διεξάχθηκε δύο φορές. Την μία το αριστερό εμπόδιο είχε απόσταση 10 cm και το δεξιά 20 και στο δεύτερο το αντίστροφο. Επιπρόσθετα, τα εμπόδια στην επόμενη πειραματική διαδικασία τοποθετήθηκαν σε ίση απόσταση από το βασικό εμπόδιο, ούτως ώστε να παρατηρηθεί η μεταβολή στην αντίθετη κατεύθυνση από την ευθεία πορεία που διατηρεί το όχημα. Τέλος, δημιουργήθηκε και ένας μικρός διάδρομος από εμπόδια για την καλύτερη παρατήρηση της πορείας του οχήματος. Τα αντίστοιχα βίντεο από τις πειραματικές διαδικασίες εμπεριέχονται στο Παράρτημα 2.

ΚΕΦΑΛΑΙΟ 4. Επίλογος

4.1 Σύνοψη

Η συγκεκριμένη εργασία αφορά την κατασκευή ενός ρομποτικού οχήματος το οποίο ελέγχει και αποφεύγει εμπόδια ούτως ώστε να βοηθάει άτομα με προβλήματα όρασης να κινούνται στον χώρο. Αρχικά, έγινε μία εισαγωγή σε ορισμούς και στην ιστορική εξέλιξη της ρομποτικής για να κατανοηθούν οι αλλαγές του τρόπου ζωής που έχει επιφέρει η ρομποτική αλλά και εννοιών. Έπειτα, αναλύθηκε σε επίπεδο εξαρτημάτων (hardware) και λογισμικού (software) ο μικροεπεξεργαστής Arduino Uno ο οποίος είναι η “καρδιά” της συγκεκριμένης κατασκευής. Στην συνέχεια, αναλύθηκε με λεπτομέρεια το κάθε εξάρτημα της κατασκευής και τα βήματα υλοποίησης της κατασκευής καθώς και ο προγραμματισμός. Ο κώδικας που χρησιμοποιήθηκε είναι σχετικά απλός με αρκετές παραμέτρους, δίνοντας την ευχέρεια άμεσων τροποποιήσεων πάνω στο κομμάτι της κίνησης αλλά και για την προσθήκη νέων δυνατοτήτων που μπορεί να χρειαστούν για μελλοντικές αναβαθμίσεις. Η κατασκευή και ο προγραμματισμός ενός ρομποτικού οχήματος είναι ένα πολύ ενδιαφέρον αντικείμενο και κατά την διάρκεια εκπόνησης της συγκεκριμένης διπλωματικής δημιουργήθηκε το αίσθημα για συνεχή αναζήτηση.

4.2 Προβλήματα και Αντιμετώπιση

Κατά την πορεία της κατασκευής αλλά και προγραμματισμού του ρομποτικού οχήματος εμφανίστηκαν κάποια προβλήματα που είχαν να κάνουν είτε με τα εξαρτήματα και την σωστή σύνδεση είτε με τους κώδικες προγραμματισμού. Ένα αρχικό πρόβλημα που διαπιστώθηκε είναι πως οι κινητήρες της κατασκευής δεν έμπαιναν σε λειτουργία. Για την επίλυση του συγκεκριμένου προβλήματος χρειάστηκε να αναζητηθούν και να εγκατασταθούν επιπρόσθετε βιβλιοθήκες στον Arduino Uno έτσι ώστε να υπάρχει συμβατότητα. Ωστόσο η απόσταση με την οποία το όχημα σταμάταγε μπροστά από εμπόδιο δεν ήταν πάντοτε αυτή που είχαμε ορίσει στον κώδικα αλλά μπορεί να υπήρχε μία μικρή απόκλιση κάποιων εκατοστών.

4.3 Συμπεράσματα

Η διπλωματική αυτή, επικεντρώθηκε στην κατασκευή ενός ρομποτικού οχήματος που αποφεύγει εμπόδια. Με γνώμονα τις ανάγκες και τις απαιτήσεις της εργασίας αυτής αναπτύχθηκαν και παρουσιάστηκαν διάφορες λειτουργίες του μικροεπεξεργαστή, ο τρόπος λειτουργίας του καθώς και κάποιες από τις δυνατότητές του. Ο χρήστης του συγκεκριμένου συστήματος θα μπορεί να καθοδηγείται από το όχημα καθώς εκείνο μπορεί να κινείται στο χώρο αποφεύγοντας τα όποια εμπόδια υπάρχουν κοντά του.

Λαμβάνοντας υπόψιν ότι ειδικά εκπαιδευμένοι σκύλοι (σκύλοι συνοδοί) κοστίζουν χιλιάδες ευρώ μέσω της συγκεκριμένης διπλωματικής αλλά και με ενδεχόμενες μελλοντικές επεκτάσεις θα είναι ευκολότερο για τα άτομα με προβλήματα όρασης να αποκτήσουν ένα βελτιωμένο ρομποτικό όχημα καθοδήγησης.

4.4 Προτάσεις Μελλοντικής Εξέλιξης

Μέσα από την διαδικασία εκπόνησης της συγκεκριμένης διπλωματικής διαπιστώθηκε ότι ο μικροεπεξεργαστής Arduino Uno σε σχέση με τις δυνατότητες που παρέχει είναι αρκετά οικονομικός. Βάση αυτού, παρακάτω γίνεται αναφορά σε κάποιες προτάσεις μελλοντικής εξέλιξης.

- Η αναγνώριση στοιχείων του περιβάλλοντος και η χαρτογράφησή τους έτσι ώστε για συγκεκριμένα εμπόδια (π.χ. σκάλες) ο χειριστής να δέχεται ηχητικό σήμα από ένα buzzer όταν τα εντοπίζει.
- Η σύνδεση και επικοινωνία του οχήματος με το σύστημα έξυπνων φαναριών [19].
- Προσθήκη συστήματος GPS για να μπορεί ο χειριστής να βάλει μία διεύθυνση και το όχημα να ξεκινήσει την πορεία όπως ακριβώς γίνεται και στο σύστημα του κινητού τηλεφώνου.
- Η ενσωμάτωση αλγορίθμων ελέγχου μέσω εγκεφαλικών σημάτων που επιτρέπει τον έλεγχο ρομποτικών συστημάτων ακόμα και από άτομα με αδυναμία χρήσης των άνω άκρων τους [20].
- Η ενσωμάτωση στο σύστημα φωτοβολταϊκού πάνελ για την ηλεκτροδότησή του με χρήση ηλιακής ενέργειας [21].
- Η διασύνδεση του συστήματος με συστήματα διαχείρισης εκτάκτων καταστάσεων ώστε να πληροφορείται το ρομποτικό σύστημα για τις συνθήκες που επικρατούν στο ευρύτερο περιβάλλον του και να λαμβάνει κατάλληλες αποφάσεις [22].
- Η διασύνδεση του συστήματος με συστήματα αυτοματοποιημένης προστασίας από κλοπή [23].
- Η διασύνδεση του συστήματος με συστήματα ασύρματων δικτύων αισθητήρων [24] και διαδικτύου των πραγμάτων (IoT) [25].

Βιβλιογραφία – Αναφορές - Διαδικτυακές Πηγές

[1] John J. Craig. (2020) *Εισαγωγή στη ρομποτική μηχανική και αυτόματος έλεγχος*, 4η έκδοση, Αθήνα, Εκδόσεις Τζιόλα

[2] C. A. Thiels, J. M. Aho, S. P. Zietlow, and D. H. Jenkins, "Use of Unmanned Aerial Vehicles for Medical Product Transport.," vol. 34, no. 2, pp. 104–108, 2015.

[3] HJ Levesque (2017). *Common Sense, the Turing Test, and the Quest for Real AI*, διαθέσιμο στον ιστότοπο: https://books.google.gr/books?hl=en&lr=&id=ZYA1DgAAQBAJ&oi=fnd&pg=PR7&dq=HJ+Levesque+-+2017&ots=xvFmK5B8Rh&sig=iFnxBy9Yyt5A8FlxmdmowLKT2W5M&redir_esc=y#v=onepage&q&f=false (τελευταία είσοδος 13/3/2021)

[4] Τσοπανάκης, Δ. (2019). *Υλοποίηση συστήματος Delta Robot με Μηχανική Όραση για χρήση σε γραμμή παραγωγής*, διαθέσιμο στον ιστότοπο: <http://hdl.handle.net/20.500.12688/9413> (τελευταία είσοδος 13/3/2021)

[5] Μαραγκός, Χ. (2017). *Προγραμματισμός Βιομηχανικού Ρομποτικού Βραχίονα για Αποφυγή Σύγκρουσης με Συνεργαζόμενο Άνθρωπο*, διαθέσιμο στον ιστότοπο: <http://dx.doi.org/10.26240/heal.ntua.15076> (τελευταία είσοδος 13/3/2021)

[6] Καραστέργιου, Β & Τσιλομήτρου, Ο. (2010). *Ανάπτυξη εφαρμογών σε όλα τα δυνατά περιβάλλοντα προγραμματισμού του ρομπότ Katana-Neuronics*, διαθέσιμο στον ιστότοπο: <http://hdl.handle.net/10889/3838> (τελευταία είσοδος 13/3/2021)

[7] Καϊμακάμης, Α. (2002). *Εφαρμογές Ρομποτικής, Πρακτική Εφαρμογή με τον Προγραμματιζόμενο Εγκέφαλο της Lego*, διαθέσιμο στον ιστότοπο: <http://dspace.lib.uom.gr/handle/2159/975> (τελευταία είσοδος 13/3/2021)

[8] Ρωσσίδης, Δ. & Καμπουράκης, Γ. (2012). *Κατασκευή και Πλοήγηση Ρομπότ Ακουστικών Μετρήσεων*, διαθέσιμο στον ιστότοπο: <http://artemis.cslab.ece.ntua.gr:8080/jspui/handle/123456789/16409> (τελευταία είσοδος 13/3/2021)

[9] Τσούμπας, Δ. (2015). *Έλεγχος βιομηχανικού ρομπότ με βάση ανοιχτό λογισμικό ψηφιακής καθοδήγησης εργαλειομηχανών*, διαθέσιμο στον ιστότοπο: <http://dx.doi.org/10.26240/heal.ntua.10091> (τελευταία είσοδος 13/3/2021)

[10] Δουκλιάς, Α. & Ουζούνου, Ν. (2019). *Χρήση αυτόνομων UAVs για την επίτευξη επικοινωνίας οπτικής επαφής σε περιβάλλον με εμπόδια*, διαθέσιμο στον ιστότοπο: <http://artemis.cslab.ece.ntua.gr:8080/jspui/handle/123456789/17645> (τελευταία είσοδος 13/3/2021)

[11] Πιπεράκης, Σ. (2019). *Robust nonlinear state estimation for humanoid robots*, διαθέσιμο στον ιστότοπο: <http://hdl.handle.net/10442/hedi/47240> (τελευταία είσοδος 13/3/2021)

[12] Σούρβου, Φ. (2017). *Εφαρμογή της Ρομποτικής στην Επιστήμη της Ιατρικής - Χειρουργική*, διαθέσιμο στον ιστότοπο: <http://apothesis.teicm.gr/xmlui/handle/123456789/3850> (τελευταία είσοδος 13/3/2021)

[13] Καϊμακάμης, Α. (2002). *Εφαρμογές Ρομποτικής, Πρακτική Εφαρμογή με τον Προγραμματιζόμενο Εγκέφαλο της Lego*, διαθέσιμο στον ιστότοπο: <http://dspace.lib.uom.gr/handle/2159/975> (τελευταία είσοδος 13/3/2021)

[14] Βασιλείου, Μ. & Τοπαλίδης, Χ. (2019). *Κατασκευή και κίνηση ανθρωποειδούς ρομπότ*, διαθέσιμο στον ιστότοπο: <http://hdl.handle.net/11610/19500> (τελευταία είσοδος 13/3/2021)

[15] Λεπίπας Π, Γ. (2019). *Αισθητήρας για Παρακολούθηση Εντοπισμένης Υπολειμματικής Τάσης σε Επιφάνειες Χάλυβα με Χρήση της Τεχνικής των Μαγνητοσυστολικών Γραμμών Καθυστέρησης*, διαθέσιμο στον ιστότοπο:

[16] Ντιζιώνη, Δ. (2019). *Αυτόματη Παραγωγή Διεπαφής Υψηλού Επιπέδου για Συλλογή Δεδομένων Αισθητήρων Ρομπότ Αξιοποιώντας την Πλατφόρμα R4A*, διαθέσιμο στον ιστότοπο: <http://ikee.lib.auth.gr/record/302660/?ln=el> (τελευταία είσοδος 13/3/2021)

[17] Κρομμύδα, Μ. (2003). *Ανάπτυξη λογισμικού για το εκπαιδευτικό ρομπότ της LEGO σε περιβάλλον JAVA*, διαθέσιμο στον ιστότοπο: <http://dspace.lib.uom.gr/handle/2159/930> (τελευταία είσοδος 13/3/2021)

[18] Μπαγλατζή, Α. & Παπαχαρτοφύλη, Ο. (2011). *Το Ρομποτικό Σύστημα da Vinci στη Χειρουργική Διαμέσου Φυσικών Οπών*, διαθέσιμο στον ιστότοπο: <http://artemis-new.cslab.ece.ntua.gr:8080/jspui/handle/123456789/16190> (τελευταία είσοδος 13/3/2021)

[19] Montanha, A., Escalon, M. J., Domínguez-Mayo, F. J., & Polidorio, A. M. (2016, August). A technological innovation to safely aid in the spatial orientation of blind people in a complex urban environment. In *2016 International Conference on Image, Vision and Computing (ICIVC)* (pp. 102-107). IEEE.

[20] Korovesis, N., Kandris, D., Koulouras, G., & Alexandridis, A. (2019). Robot motion control via an EEG-based brain-computer interface by using neural networks and alpha brainwaves. *Electronics*, 8(12), 1387.

[21] Koukouvaos, C., Kandris, D., & Samarakou, M. (2014). Computer-aided modelling and analysis of PV systems: a comparative study. *The Scientific World Journal*, 2014.

[22] Pantazis, N. A., Nikolidakis, S. A., Kandris, D., & Vergados, D. D. (2011, September). An automated system for integrated service management in emergency situations. In *2011 15th Panhellenic Conference on Informatics* (pp. 154-157). IEEE.

[23] Papadakis, N., Koukoulas, N., Christakis, I., Stavrakas, I., & Kandris, D. (2021). An IoT-Based Participatory Antitheft System for Public Safety Enhancement in Smart Cities. *Smart Cities*, 4(2), 919-937.

[24] Kandris, D., Nakas, C., Vomvas, D., & Koulouras, G. (2020). Applications of wireless sensor networks: an up-to-date survey. *Applied System Innovation*, 3(1), 14.

[25] Zantalis, F., Koulouras, G., Karabetsos, S., & Kandris, D. (2019). A review of machine learning and IoT in smart transportation. *Future Internet*, 11(4), 94.

ΠΑΡΑΡΤΗΜΑ 1

```
.  
#include <AFMotor.h>  
#include <Servo.h>  
#include <NewPing.h>  
  
#define TRIG_PIN A0  
#define ECHO_PIN A1  
#define MAX_DISTANCE 280  
#define MAX_SPEED 230  
#define MAX_SPEED_OFFSET 20  
#define COLL_DIST 40  
#define TURN_DIST COLL_DIST+20  
NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);  
  
AF_DCMotor leftMotor1(1, MOTOR12_1KHZ);  
AF_DCMotor leftMotor2(2, MOTOR12_1KHZ);  
AF_DCMotor rightMotor1(3, MOTOR34_1KHZ);  
AF_DCMotor rightMotor2(4, MOTOR34_1KHZ);  
Servo myservo;  
  
int leftDistance, rightDistance;  
int curDist = 0;  
String motorSet = "";  
int speedSet = 200;  
  
//----- SETUP LOOP -----  
void setup() {  
  myservo.attach(10);  
  myservo.write(90);  
  delay(1000);  
}  
.  
void loop() {  
  myservo.write(90);  
  delay(90);  
  curDist = readPing();  
  if (curDist < COLL_DIST) {changePath();}  
  moveForward();  
  delay(500);  
}  
//-----  
  
void changePath() {  
  moveStop();  
  myservo.write(180);  
  delay(1000);  
  rightDistance = readPing();  
  delay(1000);  
  myservo.write(0);  
  delay(700);  
  leftDistance = readPing();  
  delay(1000);  
  myservo.write(90);  
  delay(1000);  
  compareDistance();  
}  
  
void compareDistance()  
{  
  if (leftDistance>rightDistance)  
  {  
    turnLeft();  
  }  
  else if (rightDistance>leftDistance)
```

```
int readPing() {
    delay(70);
    unsigned int uS = sonar.ping();
    int cm = uS/US_ROUNDTRIP_CM;
    return cm;
}
//-----
void moveStop() {
    leftMotor1.run(RELEASE);
    leftMotor2.run(RELEASE);
    rightMotor1.run(RELEASE);
    rightMotor2.run(RELEASE);
}
//-----
void moveForward() {
    motorSet = "FORWARD";
    leftMotor1.run(FORWARD);
    leftMotor2.run(FORWARD);
    rightMotor1.run(FORWARD);
    rightMotor2.run(FORWARD);
    for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2)
    {
        leftMotor1.setSpeed(speedSet);
        leftMotor2.setSpeed(speedSet);
        rightMotor1.setSpeed(speedSet);
        rightMotor2.setSpeed(speedSet);
        delay(5);
    }
}
//-----
void moveBackward() {
    motorSet = "BACKWARD";
    leftMotor1.run(BACKWARD);
    leftMotor2.run(BACKWARD);
    rightMotor1.run(BACKWARD);
    rightMotor2.run(BACKWARD);
    for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2)
    {
        leftMotor1.setSpeed(speedSet);
        leftMotor2.setSpeed(speedSet);
        rightMotor1.setSpeed(speedSet);
        rightMotor2.setSpeed(speedSet);
        delay(5);
    }
}
//-----
void turnLeft() {
    motorSet = "LEFT";
    leftMotor1.run(BACKWARD);
    leftMotor2.run(FORWARD);
    rightMotor2.run(FORWARD);
    rightMotor1.run(BACKWARD);
    rightMotor1.setSpeed(speedSet+MAX_SPEED_OFFSET);
    rightMotor2.setSpeed(speedSet+MAX_SPEED_OFFSET);
    delay(1500);
    motorSet = "FORWARD";
    leftMotor1.run(FORWARD);
    leftMotor2.run(FORWARD);
    rightMotor1.run(FORWARD);
    rightMotor2.run(FORWARD);
}
}
```

```
void turnLeft() {
  motorSet = "LEFT";
  leftMotor1.run(BACKWARD);
  leftMotor2.run(FORWARD);
  rightMotor2.run(FORWARD);
  rightMotor1.run(BACKWARD);
  rightMotor1.setSpeed(speedSet+MAX_SPEED_OFFSET);
  rightMotor2.setSpeed(speedSet+MAX_SPEED_OFFSET);
  delay(1500);
  motorSet = "FORWARD";
  leftMotor1.run(FORWARD);
  leftMotor2.run(FORWARD);
  rightMotor1.run(FORWARD);
  rightMotor2.run(FORWARD);
}
//-----
void turnRight() {
  motorSet = "RIGHT";
  leftMotor1.run(FORWARD);
  leftMotor2.run(BACKWARD);
  leftMotor1.setSpeed(speedSet+MAX_SPEED_OFFSET);
  leftMotor2.setSpeed(speedSet+MAX_SPEED_OFFSET);
  rightMotor1.run(BACKWARD);
  rightMotor2.run(FORWARD);
  delay(1500);
  motorSet = "FORWARD";
  leftMotor1.run(FORWARD);
  leftMotor2.run(FORWARD);
  rightMotor1.run(FORWARD);
  rightMotor2.run(FORWARD);
}
//-----
void turnAround() {
  motorSet = "RIGHT";
  leftMotor1.run(FORWARD);
  leftMotor2.run(FORWARD);
  rightMotor1.run(BACKWARD);
  rightMotor2.run(BACKWARD);
  rightMotor1.setSpeed(speedSet+MAX_SPEED_OFFSET);
  rightMotor2.setSpeed(speedSet+MAX_SPEED_OFFSET);
  delay(100);
  motorSet = "FORWARD";
  leftMotor1.run(FORWARD);
  leftMotor2.run(FORWARD);
  rightMotor1.run(FORWARD);
  rightMotor2.run(FORWARD);
}
```

ΠΑΡΑΡΤΗΜΑ 2



video-1645270663.mp4



video-1645270683.mp4



video-1645270770.mp4



video-1645270753.
mp4



video-1645270726.
mp4



video-1645270708.mp4