



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**“Εξ’ αποστάσεως έλεγχος κινούμενου
οχήματος με φωνητικές εντολές και
χρήση της πλατφόρμας Arduino”**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

του

ΝΙΚΟΛΑΟΥ ΜΟΥΣΙΟΥ

(ΑΕΜ:2431)

Επιβλέπων : **Φωτιάδης Δημήτριος**

*"Η ειδικότημη είναι ο πιο ειλικρινής φίλος του
ανδρώϊου κι ο πιο ευγενικός του βοηθός"*

Χάρβει

Πίνακας Περιεχομένων

Περιεχόμενα	
Πίνακας Περιεχομένων.....	4
Κατάλογος Εικόνων.....	5
ΠΕΡΙΛΗΨΗ.....	7
ABSTRACT.....	8
Κεφάλαιο 1 ^ο – Εισαγωγή.....	9
1.1 Ο Κλάδος της Ρομποτικής.....	9
1.1.1 Τι είναι το Ρομπότ.....	9
1.2 Εφαρμογές των ρομπότ στην καθημερινή ζωή.....	10
1.3 Πλεονεκτήματα-μειονεκτήματα στη ρομποτική ανάπτυξη.....	10
1.3.1 Πλεονεκτήματα.....	10
1.3.2 Μειονεκτήματα.....	11
Κεφάλαιο 2 ^ο – Γλώσσες Προγραμματισμού και Arduino.....	12
2.4 Γλώσσες Προγραμματισμού και Arduino.....	12
2.4.1 Project Arduino.....	13
2.4.3 Πλεονεκτήματα Arduino.....	14
2.4.4 Τι είναι τελικά το Arduino.....	15
Κεφάλαιο 3 ^ο – Υλοποίηση του Ρομπότ.....	15
3.1 Πλάνο για την κατασκευή του ρομπότ.....	16
3.2 Περιγραφή εξαρτημάτων.....	18
3.3 Κατασκευή του Ρομπότ.....	23
3.4 Προγραμματισμός του Ρομπότ.....	28
Κεφάλαιο 4 ^ο – Τρόποι λειτουργίας του συστήματος.....	29
4.1 Κώδικας.....	31
4.1.1 Σύντομη περιγραφή κώδικα.....	31
4.1.2 Κλήση βιβλιοθηκών.....	38
4.2 Παράρτημα κώδικα.....	39
Κεφάλαιο 5 ^ο - Εφαρμογή Android φωνητικών εντολών.....	45
5.1 Η εφαρμογή.....	45
5.2 Υλοποίηση Εφαρμογής.....	49
Κεφάλαιο 6 ^ο - Συμπεράσματα και μελλοντικές βελτιώσεις.....	51
6.1 Σύνοψη και συμπεράσματα.....	51
6.2 Μελλοντικές επεκτάσεις.....	52
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	53

Κατάλογος Εικόνων

Εικόνα 1. TOP 10 γλώσσες προγραμματισμού.....	13
Εικόνα 2. Arduino UNO board.....	19
Εικόνα 3. Arduino Shield.....	19
Εικόνα 4. Obstacle Avoidance Sensor Module.....	20
Εικόνα 5. Ultrasonic Distance Module.....	20
Εικόνα 6. HC05 Bluetooth Module.....	21
Εικόνα 7. SG90 Servo Micro Motor.....	21
Εικόνα 8. DC Gear Motor.....	22
Εικόνα 9. Εισαγωγή των κινητήρων μετάδοσης.....	23
Εικόνα 10. Εισαγωγή του Arduino UNO board.....	23
Εικόνα 11. Εισαγωγή Shield,θήκη μπαταριών και Servo motor.....	24
Εικόνα 12. Σύνδεση κινητήρων στο Shield.....	24
Εικόνα 13. Σύνδεση Servo στο Shield.....	24
Εικόνα 14. Σύνδεση Bluetooth και μπαταρίας.....	25
Εικόνα 15. Εισαγωγή Obstacle Avoidance Sensor.....	26
Εικόνα 16. Σύνδεση Obstacle Avoidance Sensor.....	26
Εικόνα 17. Εισαγωγή Ultrasonic Distance Module.....	27
Εικόνα 18. Σύνδεση Ultrasonic Distance Module.....	27
Εικόνα 19. Block diagram.....	30
Εικόνα 20. App main screen.....	46
Εικόνα 21. App Bluetooth screen.....	47
Εικόνα 22. App voice command screen.....	48
Εικόνα 23. App design MIT Interface.....	49
Εικόνα 24. Κώδικας APP.....	50

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή μου, κύριο Φωτιάδη Δημήτριο για την επίβλεψη της εργασίας και την βοήθεια που μου πρόσφερε καθ' όλη την διάρκεια. Επιπρόσθετα, θα ήθελα να ευχαριστήσω την οικογένεια μου για την στήριξη και την υπομονή όλα αυτά τα χρόνια της φοίτησής μου αλλά και τους στενούς μου φίλους-συμφοιτητές.

ΠΕΡΙΛΗΨΗ

Σε αυτήν την πτυχιακή εργασία, παρουσιάζεται μία ρομποτική κατασκευή, βασισμένη σε μικροελεγκτές και λογισμικό Arduino. Ο κύριος στόχος αυτού του έργου είναι η ανάπτυξη ενός οχήματος με εξ' αποστάσεως έλεγχο με φωνητικές εντολές, στο οποίο έχουν εγκατασταθεί αισθητήρες υπερήχων και bluetooth. Ο μικροελεγκτής, μέσω θύρας USB, δέχεται τον κώδικα από τον προσωπικό υπολογιστή και στη συνέχεια το ρομπότ ξεκινά την εκτέλεση των οδηγιών που φορτώθηκαν σε αυτό με ένα διακόπτη on/off. Ο διακόπτης είναι συνδεδεμένος με μία πράσινη φωτοδίοδο (LED), η οποία αν δεν ανάψει στην κατάσταση on, υποδηλώνει πρόβλημα στην τροφοδοσία του ρομπότ.

Για να γίνει η κίνηση του ρομπότ αρχικά συνδέουμε μέσω εφαρμογής, η οποία υλοποιήθηκε αποκλειστικά για το Project, το ρομπότ με Bluetooth και έπειτα ο χρήστης ξεκινά να δίνει τις φωνητικές εντολές. Κατά την κίνησή του, διαθέτει 2 πομποδέκτες υπερήχων (μπροστά και πίσω) που παρέχουν τη δυνατότητα στο όχημα αποφυγής εμποδίων.

Μελλοντικές αλλαγές και βελτιώσεις που θα μπορούσαν να γίνουν στο ρομπότ θα ήταν η εγκατάσταση ανιχνευτή μετάλλων, μετρητή θερμοκρασίας, ενσωμάτωση κάμερας, εισαγωγή λογισμικού για ενημέρωση σχετικά με την κατάσταση του ρομπότ (βλάβες, ακραίες μετρήσεις, χαμηλή μπαταρία, κτλ), χρήση WIFI Shield για επικοινωνία μεταξύ χρήστη και ρομπότ μέσω του διαδικτύου.

Λέξεις κλειδιά : Bluetooth , ρομπότ , φωνητικές εντολές , Arduino , πομποδέκτες υπερήχων

ABSTRACT

In this thesis, a robotic construction based on microcontrollers and Arduino software is presented. The main objective of this project is to develop a remote-control vehicle with voice commands, in which ultrasonic and Bluetooth sensors are installed. The microcontroller, through a USB port, receives the code from the personal computer, and then the robot starts executing the instructions loaded into it with an on/off switch. The switch is connected to a green light emitting diode (LED), which if it does not light up in the on state, indicates a problem with the robot's power supply.

To start the robot, we first connect via an application, which was implemented exclusively for the Project, to the robot with Bluetooth and then the user starts giving voice commands. When moving, it has 2 ultrasonic transmitters (front and rear) that enable the vehicle to avoid obstacles.

Future changes and improvements that could be made to the robot will be the installation of metal detector, temperature gauge, camera integration, software import to inform about the status of the robot (faults, extreme readings, low battery, etc.), use of WIFI Shield for communication between user and robot via the internet.

Keywords: Bluetooth, robot, voice commands, Arduino, ultrasonic transmitters

Κεφάλαιο 1^ο – Εισαγωγή

1.1 Ο Κλάδος της Ρομποτικής

Η ρομποτική είναι ένας επιστημονικός κλάδος που περιλαμβάνει τη μηχανολογία, ηλεκτρολογία, πληροφορική και άλλες επί μέρους επιστήμες. Ασχολείται με την κατασκευή, τον χειρισμό και τη λειτουργία των ρομπότ, καθώς επίσης τον προγραμματισμό τους και την ανατροφοδότηση πληροφοριών από αυτά. Αυτές οι τεχνολογίες χρησιμοποιούνται για την ανάπτυξη μηχανών που θα μπορούν να λειτουργούν ως υποκατάστατα του ανθρώπου. Η ιδέα της δημιουργίας μηχανών που θα βασίζονται στην αυτονομία και τη λήψη αποφάσεων ανεξάρτητα από την ανθρώπινη παρουσία ξεκίνησε πολλά χρόνια πριν, ωστόσο δε γνώρισε πραγματική άνθηση μέχρι τον 20ο αιώνα [1].

1.1.1 Τι είναι το Ρομπότ

Πρόκειται για μία μηχανή, προγραμματισμένη από ηλεκτρονικό υπολογιστή, η οποία μπορεί να εκτελεί διάφορες λειτουργίες με ταχύτητα και ακρίβεια. Υπάρχουν ρομπότ, τα οποία κατασκευάζονται με σκοπό να μοιάζουν στον άνθρωπο, στην ανατομία του και στον τρόπο κινήσεών του. Αυτά λέγονται ανδροειδή. Ωστόσο, ο αριθμός των ανδροειδών είναι περιορισμένος, καθώς ως επί των πλείστων η εμφάνιση των ρομπότ είναι αδιάφορη στον άνθρωπο. Αυτό που διαχωρίζει τα ρομπότ από τις κοινές αυτοματοποιημένες συσκευές που συναντώνται σε διάφορους τομείς της ζωής του ανθρώπου είναι η δυνατότητα επαναπρογραμματισμού της μνήμης τους, προσδίδοντάς τους έτσι νοημοσύνη [2].

Ένα ρομπότ θα πρέπει να είναι ευπροσάρμοστο. Δηλαδή, να έχει τη δυνατότητα να δέχεται πληροφορίες και ερεθίσματα από το γύρω περιβάλλον του και να προσαρμόζεται ανάλογα σε αυτό. Έτσι, αποτελείται από αισθητήρες και από το σύστημα ελέγχου, δηλαδή τον «εγκέφαλο» του ρομπότ. Αισθητήρες, όπως ένα θερμομέτρο ή μία κάμερα, παίρνουν κατάλληλες μετρήσεις από τις επικρατούσες συνθήκες του περιβάλλοντος στο οποίο βρίσκεται το ρομπότ. Στην προσαρμογή αυτή, κυρίαρχο ρόλο παίζει το σύστημα ελέγχου του [3]. Ανάλογα με το είδος του συστήματος ελέγχου που έχει εγκατεστημένο το ρομπότ, τα χωρίζουμε σε 2 κατηγορίες

- Προγραμματισμένα εκ των προτέρων. Δηλαδή πρόκειται για ρομπότ, τα οποία είναι στην πιο απλή μορφή τους και εκτελούν κινήσεις που επαναλαμβάνονται συνεχώς ή μπορούν να συλλέξουν περιορισμένο αριθμό πληροφοριών γενικότερα.
- Αυτόνομα ρομπότ. Περισσότερο σύνθετες μηχανές, όπου μέσω προγραμματισμού τους έχουμε προσδώσει «νοημοσύνη». Συμβάλλουμε στην ομαλότερη προσαρμογή του ρομπότ δίνοντάς του διεξόδους, για τυχόν εμπόδια ή ασυνέχειες κατά την κίνηση-λειτουργία του. Μπορεί να συλλέξει πολύ μεγαλύτερη γκάμα πληροφοριών.

1.2 Εφαρμογές των ρομπότ στην καθημερινή ζωή

Η θεμελιώδης αρχή στη χρήση των ρομπότ, όπως προαναφέραμε είναι να εξυπηρετούν τον άνθρωπο στην καθημερινότητά του. Τα τελευταία χρόνια, η έρευνα στη χρήση των ρομπότ έχει βελτιώσει σημαντικά τη ζωή του ανθρώπου και τον έχει οδηγήσει σε καινοτομίες. Η Ιαπωνία θεωρείται μέχρι σήμερα πρωτόπορος στον κλάδο της Ρομποτικής, καθώς έχει εισάγει τη χρήση των ρομπότ ολοκληρωτικά στις ζωές των ανθρώπων. Υπάρχουν διάφοροι τρόποι για να κατηγοριοποιηθούν τα ρομπότ. Ένας από αυτούς είναι οι εφαρμογές τους στη ζωή του ανθρώπου [4].

1.3 Πλεονεκτήματα-μειονεκτήματα στη ρομποτική ανάπτυξη

Είναι αδιαμφισβήτητη η επιρροή της Ρομποτικής και των εφαρμογών της στην καθημερινότητα του ανθρώπου. Όμως, αυτή η ολοένα αυξανόμενη εξάρτηση μας από την τεχνολογία, πέρα από τα σημαντικά πλεονεκτήματα και διευκολύνσεις που παρέχει, συνδέεται και με κάποια μειονεκτήματα, τα οποία και αυτά με τη σειρά τους έχουν επίδραση σε εμάς. Ακολουθούν κάποια από αυτά.

1.3.1 Πλεονεκτήματα

Πιθανότατα το μεγαλύτερο πλεονέκτημα των ρομποτικών κατασκευών, είναι το ευρύτατο φάσμα εφαρμογών τους. Το γεγονός ότι μπορούν να χρησιμοποιηθούν με κάθε δυνατό τρόπο για να παρέμβουν στη ζωή του ανθρώπου, κάνει ακόμα πιο αισθητή την παρουσία τους. Είτε πρόκειται για τους κλάδους των επιστημών, είτε για πιο κοινότερες εφαρμογές, είναι αδιαμφισβήτητη η αλληλεπίδραση της τεχνητής νοημοσύνης με την καθημερινότητα του ανθρώπου, με σκοπό φυσικά τη βελτίωση των συνθηκών ζωής και επιβίωσής του.

Είδαμε πόσο σημαντική είναι η παρουσία των ρομπότ στον τομέα της βιομηχανίας. Τα βιομηχανικά αυτοματοποιημένα ρομπότ έχουν αφενός την ικανότητα να βελτιώνουν δραματικά την ποιότητα του εκάστοτε προϊόντος, αφετέρου με τη χρήση τους αυξάνεται ο ρυθμός παραγωγής, αφού δρουν ταχύτατα και με μεγάλη ακρίβεια. Εκτός αυτού, είναι ασφαλέστερη επιλογή, όταν οι συνθήκες κάτω από τις οποίες εκτελούνται κάποιες εργασίες είναι ακατάλληλες για τον άνθρωπο, κυρίως λόγω επικινδυνότητας. Έτσι, η βελτιωμένη ασφάλεια των εργαζομένων οδηγεί και σε εξοικονόμηση πόρων, καθώς λειτουργούν όλο το 24ωρο, δεν πληρώνονται και φυσικά δε «βαριούνται».

Επιπρόσθετα, δημιουργούνται νέες θέσεις εργασίας με τη χρήση ρομπότ. Συγκεκριμένα, είναι απαραίτητος ο τηλεχειρισμός τους καθώς επίσης και η συντήρηση τους. Με δεδομένο ότι εργάζονται ακατάπαυστα, είναι ευάλωτα σε τυχόν βλάβες και εκεί παρεμβαίνει ο τεχνικός. Επίσης, στην περίπτωση κάποιου ατυχήματος, θα αποφευχθούν τραυματισμοί, θανατηφόροι ή μη του εργατικού δυναμικού που αντικαταστάθηκε από το ρομπότ και το ζήτημα θα επιλυθεί άμεσα με την επισκευή του ή επανεγκατάσταση ενός νέου [5].

1.3.2 Μειονεκτήματα

Είδαμε πόσο σημαντικό ρόλο παίζουν οι εφαρμογές της Ρομποτικής στη ζωή του ανθρώπου. Εκτελούν δύσκολες εργασίες, διασκεδάζουν, παρέχουν ασφάλεια. Ωστόσο, φέρουν και κάποιες αρνητικές συνέπειες, οι οποίες συνυπάρχουν με τα οφέλη τους απέναντι στην ανθρωπότητα, και αυτές αφορούν είτε το χειρισμό τους, είτε τον αντίκτυπό τους σε αυτήν.

Το κυριότερο πρόβλημα κατά την λειτουργία ενός ρομπότ είναι η τροφοδοσία του. Ως ένα σύμπλεγμα ηλεκτρονικών εξαρτημάτων, απαιτεί ενέργεια. Η εξάρτησή του από μία πηγή ενέργειας είναι σημαντικός παράγοντας που λαμβάνεται υπόψιν κατά την εγκατάστασή του, διότι το ρομπότ περιορίζεται κινητικά λόγω καλωδίων. Η ενσύρματη ανταλλαγή πληροφοριών μεταξύ ρομπότ και κέντρου ελέγχου μπορεί να λυθεί με ασύρματους πομπούς και δέκτες. Ωστόσο, το ζήτημα της τροφοδοσίας παραμένει. Είτε θα πρέπει να γίνεται με καλωδίωση, επομένως συνεχίζει να περιορίζεται η κίνηση του ρομπότ, είτε μπορεί να τροφοδοτείται μέσω μπαταρίας. Στην προκειμένη περίπτωση έχει περιορισμένη διάρκεια η λειτουργικότητα του ρομπότ, καθώς είναι απαραίτητη κάποια στιγμή η αντικατάσταση της μπαταρίας ή η επαναφόρτισή της.

Ένα ακόμα μεγάλο πρόβλημα που εμφανίστηκε με τη χρήση των ρομπότ είναι αυτό της ανεργίας. Καθώς, τα ρομπότ αποδεικνύονται πιο γρήγορα και γενικότερα αποδοτικότερα από τον άνθρωπο σε εργασίες τυποποιημένες ή δύσκολες, αυξάνεται όλο και περισσότερο ο αριθμός τους με αποτέλεσμα όλο και μεγαλύτερο πλήθος ανθρώπων να οδηγείται στην ανεργία.

Επιπρόσθετα, η κατασκευή, η συντήρηση αλλά και η επισκευή των ρομπότ κοστίζει μεγάλα ποσά χρημάτων, ανάλογα με το μέγεθός του και το πλήθος των διεργασιών που μπορεί να εκτελεί. Συχνά, χρειάζεται αναβάθμιση του λογισμικού τους ώστε να αποφεύγονται προβλήματα συμβατότητας με το κέντρο ελέγχου αλλά και για να βελτιώνεται η λειτουργικότητά τους. Στην περίπτωση καταστροφής του ρομπότ, η αντικατάστασή του θα έχει μεγάλη οικονομική επιβάρυνση στον ιδιοκτήτη του και εκτός αυτού η επαναφορά χαμένων δεδομένων από τυχόν διαγραφή τους είναι επίσης ιδιαίτερα χρονοβόρα και πολυέξοδη.

Αυτά είναι μερικά από τα μειονεκτήματα που προκύπτουν από την αισθητή παρουσία των ρομπότ στην ανθρώπινη καθημερινότητα. Έτσι, είναι κρίσιμο να λαμβάνονται υπόψιν κατά την κατασκευή και ανάπτυξή τους, καθώς ο κυριότερός τους στόχος πρέπει να παραμένει η εξυπηρέτηση των αναγκών του ανθρώπου και η βελτίωση της ποιότητας ζωής του. Πάντως, φαντάζει αδύνατον η τεχνολογία και οι εφαρμογές της να αντικαταστήσουν κάποτε την ικανότητα της κριτικής σκέψης, της εξαγωγής συμπερασμάτων και της λήψης αποφάσεων σε κρίσιμες ή μη καταστάσεις του επίσης εξελισσόμενου ανθρώπινου νου [5].

Κεφάλαιο 2^ο – Γλώσσες Προγραμματισμού και Arduino

2.4 Γλώσσες Προγραμματισμού και Arduino

Γλώσσα προγραμματισμού είναι ένα σύνολο κανόνων λέξεων και γραμματικής και με τη βοήθεια αυτού του συνόλου ο άνθρωπος έχει τη δυνατότητα να καθοδηγήσει έναν υπολογιστή ή υπολογιστική μηχανή να εκτελέσει κάποιο συγκεκριμένο έργο. Μπορούν να χρησιμοποιηθούν επίσης για τη δημιουργία προγραμμάτων για την υλοποίηση αλγορίθμων.

Οι πρώτες γλώσσες προγραμματισμού γράφτηκαν κατά τη δεκαετία του 1950 με σκοπό να εισάγονται οδηγίες σε έναν υπολογιστή. Το 1949 ο John Mauchly εισάγει τον ονομαζόμενο Σύντομο Κώδικα. Ήταν μία από τις πρώτες γλώσσες προγραμματισμού υψηλού επιπέδου, όπως οι σημερινές C, C++ κτλ. Οι προτάσεις του κώδικα αυτού, αναπαριστούν μαθηματικές εκφράσεις σε κατανοητή μορφή. Ωστόσο η γλώσσα αυτή έπρεπε να μεταφραστεί σε κώδικα μηχανής για κάθε της εκτέλεση με αποτέλεσμα η όλη διαδικασία να γίνεται πολύ αργή. Έτσι, δημιουργήθηκαν κάποια προγράμματα ονομαζόμενα μεταγλωττιστές (compilers) ή διερμηνείς (interpreters) με σκοπό την επιτάχυνση της μετάφρασης του κώδικα.

Στις αρχές του 1950, στο Πανεπιστήμιο του Μάντσεστερ, ο Alick Glennie ανέπτυξε τον Autocode. Ήταν μία γλώσσα προγραμματισμού που διέθετε μεταγλωττιστή, ώστε να μετατρέπει αυτόματα τη γλώσσα σε κώδικα μηχανής.

Ο πρώτος κώδικας και μεταγλωττιστής αναπτύχθηκαν το 1952 για τον υπολογιστή Mark 1 στο ίδιο πανεπιστήμιο και θεωρείται η πρώτη μεταγλωττισμένη γλώσσα προγραμματισμού υψηλού επιπέδου. Ακολούθησε τα επόμενα χρόνια η ανάπτυξη περισσότερων γλωσσών προγραμματισμού και η περαιτέρω βελτίωσή τους με την επινόηση των προγραμματιστικών παραδειγμάτων (τέλη 1960 με τέλη 1970). Αυτά στην ουσία είναι μία μέθοδος κατηγοριοποίησης των γλωσσών προγραμματισμού, με γνώμονα τις λειτουργίες τους.

Η δεκαετία του 1980 δε χαρακτηρίζεται από ιδιαίτερες καινοτομίες στο χώρο του προγραμματισμού. Αντί να επινοηθούν νέα προγραμματιστικά παραδείγματα, προτιμήθηκε η χρήση των ήδη υπαρχόντων. Ωστόσο, εμφανίστηκε ένα έντονο ενδιαφέρον στο χώρο της διαμόρφωσης γλώσσας στον προγραμματισμό για συστήματα μεγάλης κλίμακας με τη χρήση των modules. Εκτός αυτού, ήταν και η δεκαετία που κατασκευάστηκε η C++, συγκεκριμένα το 1980 με ονομασία C with Classes που μετονομάστηκε σε C++ το 1983 [6].



Εικόνα 1 TOP 10 Γλώσσες προγραμματισμού [18]

Κατά τη δεκαετία του 1990 ο προγραμματισμός γνώρισε ραγδαία άνθηση χάρη στην ταχύτατη διάδοση του Ίντερνετ. Η εμφάνιση του Ίντερνετ έκανε δυνατή την ανταλλαγή και υιοθέτηση ιδεών και απόψεων μεταξύ των προγραμματιστών μέσω του παγκόσμιου ιστού. Συγκεκριμένα, η JavaScript έγινε ιδιαίτερα γνωστή επειδή ενσωματώθηκε γρήγορα με τον περιηγητή Netscape Navigator , ο πιο περίφημος περιηγητής διαδικτύου εκείνης της δεκαετίας. Ποικίλες άλλες γλώσσες απέκτησαν ευρεία χρήση κυρίως στην δημιουργία εφαρμογών αποκλειστικά για σέρβερ διαδικτύου. Εν κατακλείδι, ήταν ένα διάστημα ορόσημο για την επιστήμη του προγραμματισμού. Άλλωστε, δεν είναι τυχαίο που κάποιες από τις σημαντικότερες γλώσσες αναπτύχθηκαν μέσα σε αυτή τη δεκαετία, όπως η Python το 1991 και η java το 1995 [6] .

2.4.1 Project Arduino

Το πρότζεκτ του Arduino ξεκίνησε στο Ινστιτούτο Σχεδιασμού Αλληλεπίδρασης (IDII) στην Ίβρεα της Ιταλίας, το 2003. Εκείνη την περίοδο, οι μαθητευόμενοι χρησιμοποιούσαν έναν μικροελεγκτή κλασσικό για την εποχή με τίτλο Basic Stamp το μεγάλο μειονέκτημα του οποίου ήταν η τιμή του που έφθανε τα 100\$ καθιστώντας δύσκολη την αγορά του. Το 2003, ένας φοιτητής του Ινστιτούτου, ο Hernando Barragán ως έργο διδακτορικής διατριβής ανέπτυξε μία πλατφόρμα ανάπτυξης ανοιχτής πηγής που ονόμασε Wiring. Επιβλέποντες της εργασίας ήταν οι καθηγητές Massimo Banzi και Casey Reas. Σκοπός της εργασίας αυτής ήταν η δημιουργία απλών και χαμηλού κόστους εργαλείων, με τη χρήση των οποίων θα αναπτυχθούν ψηφιακά προγράμματα χωρίς προαπαιτούμενες γνώσεις ηλεκτρονικής ή προγραμματισμού [8].

Την ίδια χρονιά, ο καθηγητής Massimo Banzi, ο David Cuartielles και ο τότε φοιτητής του IDII David Mellis προσέθεσαν σύστημα υποστήριξης στον φθινό μικροελεγκτή Atmega8 που χρησιμοποιούσε η πλατφόρμα Wiring. Ωστόσο, ο Banzi ενδιαφερόταν να αναπτύξει τον μικροελεγκτή και να τον μετασχηματίσει σε ένα εργαλείο τέχνης και σχεδίου. Ως ομάδα, επιθυμούσαν μία συσκευή που να καθιστά δυνατή και εύκολη τη σύνδεση διαφόρων κυκλωμάτων μεταξύ τους (όπως κινητήρες, αισθητήρες κτλ) και εξίσου εύκολο τον προγραμματισμό της από σχεδιαστές, χομπίστες ή αρχάριους γενικότερα. Έτσι, πραγματοποίησαν κάποιες αλλαγές στον πηγαίο κώδικα του Barragán και το τελικό αποτέλεσμα μετονομάστηκε σε Arduino, ονομασία εμπνευσμένη από τον βασιλιά της Ιταλίας Arduin την περίοδο 1002 με 1014 [9].

Χρησιμοποίησαν έναν 8-bit μικροελεγκτή της εταιρίας Atmel AVR και σχεδίασαν μία αυτοτελή πλακέτα με εύκολες συνδέσεις, όπου φόρτωσαν ένα υλικολογισμικό (boot loader) για τον μικροελεγκτή, ο οποίος περιέχει και ενσωματωμένο περιβάλλον προγραμματισμού (Integrated development environment IDE). Ο νέος πια μικροελεγκτής Arduino δημιουργούσε ένα πιο άνετο και φιλικό περιβάλλον στους

χρήστες με περιορισμένο φάσμα γνώσεων και ήταν πολύ πιο προσιτός οικονομικά από την αντίστοιχη πλακέτα Wiring που κυκλοφορούσε.

Οι περισσότερες πλακέτες Arduino βασίζονται σε οχτάμπιτους (8-bit) μικροελεγκτές της Atmel AVR με ποικιλία μνήμης flash , pins και άλλων ιδιοτήτων. Φορτώνονται με κώδικα μέσω μίας σειριακής σύνδεσης με κάποιον υπολογιστή. Οι σύγχρονες πλακέτες διαθέτουν προσαρμογέα (adaptor) από USB σε σειριακό σήμα, με αποτέλεσμα να καθίσταται δυνατός ο προγραμματισμός τους με χρήση USB. Διάφορες εκδόσεις Arduino, επίσημες ή ανεπίσημες διευρύνουν το φάσμα ιδιοτήτων, καθώς συναντώνται για παράδειγμα μερικές πλακέτες με αφαιρούμενο προσαρμογέα USB ή chip που καθιστά δυνατή την επικοινωνία με Bluetooth.

Με το πέρασμα των χρόνων οι προγραμματιστές στην Arduino.cc ανέπτυξαν έναν αριθμό από πλακέτες με πιο γρήγορους επεξεργαστές, μεγαλύτερες μνήμες κτλ. Η πρώτη που έγινε γρήγορα δημοφιλής ονομαζόταν Diecimila και κυκλοφόρησε το 2007. Έκτοτε η οικογένεια του Arduino εξελίχθηκε ακόμα περισσότερο αξιοποιώντας μια ποικιλία μικροελεγκτών που παρείχε η Atmel AVR. Το 2012 κυκλοφορεί ο πρώτος μικροελεγκτής (MCU) που υλοποιεί επεξεργαστή 32-bit, ονομάζεται Due και διαφοροποιείται από τις υπόλοιπες πλακέτες Arduino από άποψη διαχείρισης ενέργειας και διαμόρφωσης επαφών εισόδου-εξόδου (pins). Παρόμοια με το Due, υπάρχουν και άλλες πλακέτες όπως η Lilypad και η Nano που χρησιμοποιούνται σε διαφορετικά φάσματα εφαρμογών [10].

2.4.3 Πλεονεκτήματα Arduino

Τα βασικότερα πλεονεκτήματα του Arduino είναι τα εξής:

- **Χαμηλό κόστος.** Μπορεί κανείς να αγοράσει τον μικροεπεξεργαστή Arduino UNO R3 με κόστος μικρότερο των 12 ευρώ. Επίσης, υπάρχουν στα διάφορα ηλεκτρονικά καταστήματα starter kit που περιλαμβάνουν μαζί με την πλακέτα, πλήθος καλωδίων, monitors, αισθητήρων με τα οποία ο καθένας μπορεί να

πειραματιστεί. Όσο για το περιβάλλον ανάπτυξης κώδικα, είναι δωρεάν και μπορεί να το κατεβάσει κανείς από την επίσημη ιστοσελίδα arduino.cc.

- **Συμβατότητα**. Το λογισμικό του Arduino εκτελείται σε περιβάλλον των Windows, Macintosh αλλά και Linux.
- **Απλότητα**. Το περιβάλλον προγραμματισμού (IDE) είναι αρκετά απλό ώστε να χρησιμοποιείται άνετα και από αρχάριους χρήστες.
- **Επεκτάσιμο λογισμικό ανοιχτού κώδικα**. Τα προγράμματα του Arduino είναι ανοιχτού κώδικα που σημαίνει ότι οποιοσδήποτε μπορεί να βρει τμήματα πηγαίου κώδικα, να τα μελετήσει και να τα τροποποιήσει σύμφωνα με τις ανάγκες του. Επίσης, μέσα από την προσθήκη βιβλιοθηκών C++ μπορούν να προστεθούν δυνατότητες στο περιβάλλον ανάπτυξης [7].

Κάποιος πεπειραμένος χρήστης μπορεί να γράψει δικό του κώδικα χαμηλού επιπέδου και να τον καταστήσει διαθέσιμο μέσω βιβλιοθηκών, ενώ οι αρχάριοι χρήστες μπορούν να προσθέσουν υποστήριξη για νέες διατάξεις χωρίς να χρειάζεται να γνωρίζουν τις χαμηλού επιπέδου λεπτομέρειες του προγραμματισμού των διατάξεων αυτών, όπως αισθητήρες.

2.4.4 Τι είναι τελικά το Arduino

Είναι ουσιαστικά μία πλατφόρμα ανοιχτού κώδικα που απευθύνεται σε ένα ευρύ κοινό από επαγγελματίες προγραμματιστές, σχεδιαστές μέχρι άτομα περιορισμένων γνώσεων στον προγραμματισμό. Είναι χαμηλού κόστους και χρησιμοποιείται για τη δημιουργία ηλεκτρονικών Projects. Αποτελείται από την πλακέτα (hardware) όπου βρίσκονται οι συνδέσεις, ο MCU, φωτεινές ενδείξεις LED, θύρα USB κτλ και από το λογισμικό (software) με τη βοήθεια του οποίου ο χρήστης αλληλεπιδρά με τον μικροελεγκτή μέσω του υπολογιστή του. Μπορεί δηλαδή να γράφει κώδικα στον υπολογιστή και να τον φορτώνει στην πλακέτα μέσω του USB.

Το περιβάλλον ανάπτυξης (IDE) που βρίσκεται ενσωματωμένο στο Arduino χρησιμοποιεί ως γλώσσα προγραμματισμού μία απλοποιημένη έκδοση C/C++. Είναι δωρεάν και μπορεί να το εγκαταστήσει ο καθένας από την επίσημη ιστοσελίδα μέσω Windows, Mac ή Linux. Αυτό καθιστά το συνολικό έργο του Arduino εύκολα προσβάσιμο από οποιονδήποτε χρήστη του υπολογιστή και του διαδικτύου. Το πρόγραμμα που αναπτύσσει ο χρήστης, αποκαλούμενο «σκετς», μόλις το εκτελέσει μέσω του IDE, υφίσταται μικρές αλλαγές και περνά απευθείας σε έναν C/C++ μεταγλωττιστή (compiler) [11].

Κεφάλαιο 3^ο – Υλοποίηση του Ρομπότ

Σε αυτό το κεφάλαιο γίνεται αναλυτική περιγραφή των υλικών που χρησιμοποιήθηκαν για την υλοποίηση της ρομποτικής κατασκευής και η βήμα προς βήμα επεξήγηση των συνδέσεων που πραγματοποιήθηκαν.

3.1 Πλάνο για την κατασκευή του ρομπότ

Σε πρώτο στάδιο έπρεπε να ληφθούν αποφάσεις σχετικά με τις επιθυμητές λειτουργίες του ρομπότ. Επρόκειτο να είναι ένα ρομπότ που εκτελεί φωνητικές εντολές, βασισμένο σε πλατφόρμα ανάπτυξης Arduino. Εκτός αυτού, θα πρέπει να διαθέτει την ικανότητα να λειτουργεί και να κινείται αυτόνομα. Επομένως, απαραίτητη είναι μία αυτόνομη πηγή τροφοδοσίας, δηλαδή μπαταρία καθώς και η «νοημοσύνη» που χρειάζεται ώστε να αποφύγει εμπόδια κατά την κίνησή του.

Αρχικά έγινε μία έρευνα αγοράς διότι υπάρχει πλήθος ηλεκτρονικών καταστημάτων εντός και εκτός Ελλάδος που παρέχουν ολοκληρωμένα κυκλώματα και πλακέτες ανάπτυξης Arduino. Έτσι, τα εξαρτήματα που κρίθηκαν χρήσιμα για την επίτευξη του σκοπού μας είναι τα εξής.

- Μακετόχαρτο 5mm Μαύρο 70x50εκ. (το οποίο κόπηκε σε διαστάσεις 17x11εκ.)



- 4 ρόδες σχεδιασμένες για smart cars



- DC Gear Motor TT 130 RPM (With Wire) για να δίνει κίνηση στις ρόδες



- UNO R3 Atmega328P Board για Arduino



- Motor L293D Shield για Arduino



- SG90 Servo Micro Motor 9G



- Battery Holder 2x18650 (with Cables)



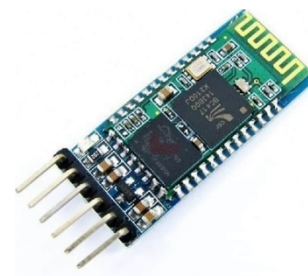
- ElecFreaks HC-SR04 Ultrasonic Distance Module για Arduino (αισθητήρας κίνησης)



- Goobay Διακόπτης Συρόμενος



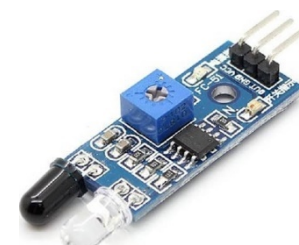
- HC05 Bluetooth Module για Arduino



- LG HG2 Επαναφορτιζόμενη Μπαταρία 18650 Li-ion 3000mAh 3.7V 2τμχ



- Infrared Obstacle Avoidance Sensor Module



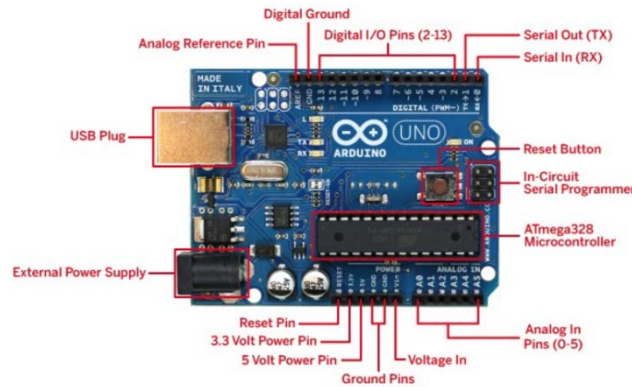
- Haitronic Female Dupont Wires/Cables for Arduino



3.2 Περιγραφή εξαρτημάτων

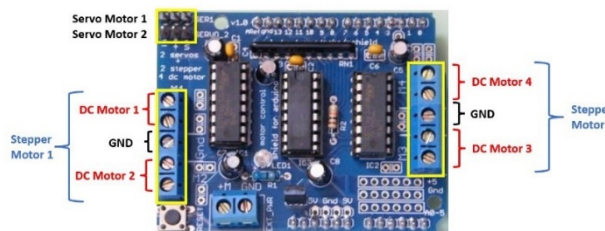
❖ Ο μικροελεγκτής **Arduino Uno R3**. Μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (Atmega328 και Atmega168 στις νεότερες εκδόσεις, Atmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωσή του σε άλλα κυκλώματα. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz (ή κεραμικό αντηχητή σε

κάποιες παραλλαγές).Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα boot loader, έτσι ώστε να μη χρειάζεται εξωτερικός προγραμματιστής. Διαθέτει μεταξύ άλλων 14 ψηφιακά pins εισόδου/εξόδου (6 από τα οποία μπορούν να χρησιμοποιηθούν ως έξοδοι PWM, θα αναλυθεί στην ενότητα της συνδεσμολογίας του ρομπότ), 6 αναλογικές εισόδους, ένα κουμπί reset, θύρα USB, μια υποδοχή τροφοδοσίας και flash μνήμη 32KB από τα οποία τα 0.5KB χρησιμοποιούνται για το boot loader. Η τάση λειτουργίας του είναι τα 5V. Η τάση εξωτερικής τροφοδοσίας είναι από 7 έως 12V ενώ τα όρια της τάσης είναι 6 με 20 Volts. Συνοδεύεται από το καλώδιο USB μαζί με δική μου προέκταση για το ανέβασμα ή επεξεργασία του κώδικα [13].



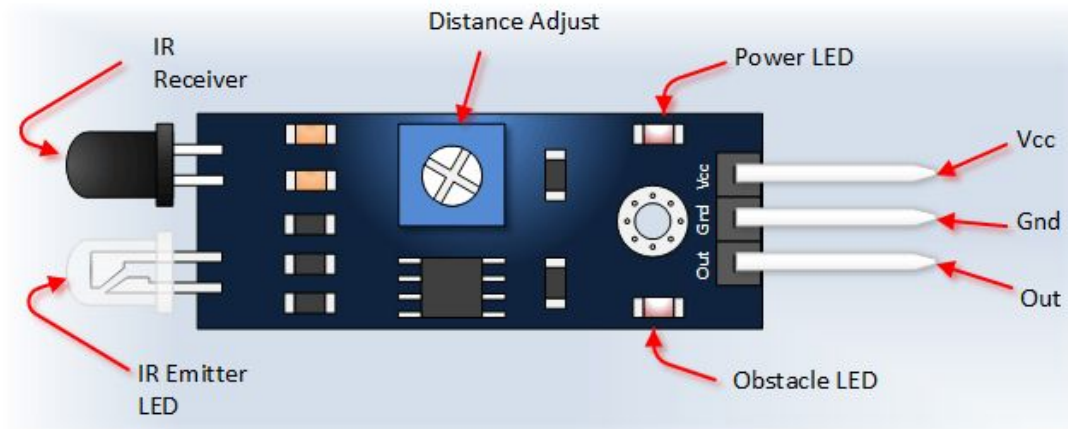
Εικόνα 2 : Διάγραμμα Arduino UNO board [19]

❖ Ο **L293D motor driver shield** είναι ένα έργο ρομποτικής που περιλαμβάνει την οδήγηση διαφόρων τύπων κινητήρων. Οι πιο συνηθισμένοι τύποι που χρησιμοποιούνται για ρομποτικές εφαρμογές περιλαμβάνουν κινητήρες συνεχούς ρεύματος, σερβομηχανισμού και βηματικούς κινητήρες. Ωστόσο, αυτοί οι κινητήρες συνήθως δεν μπορούν να οδηγηθούν απευθείας από το Arduino ή άλλο μικροελεγκτή. Αυτό οφείλεται στις υψηλότερες ονομασίες ρεύματος και ισχύος, επομένως χρησιμοποιούνται ασπίδες κινητήρα ή IC οδηγών. Αυτές οι ασπίδες ή τα IC απομονώνουν το τροφοδοτικό ενός κινητήρα και χρησιμοποιούν λογική έλεγχου από το κύκλωμα του μικροελεγκτή. Μία από τις πιο δημοφιλείς ασπίδες οδήγησης κινητήρα που χρησιμοποιούνται με το Arduino είναι το L293D. Η πλήρης ασπίδα οδήγησης κινητήρα L293D μπορεί να ελέγξει έως και τέσσερις αμφίδρομους κινητήρες DC με επιλογή ταχύτητας 8-bit, δύο βηματικούς κινητήρες και δύο σερβοκινητήρες [12].



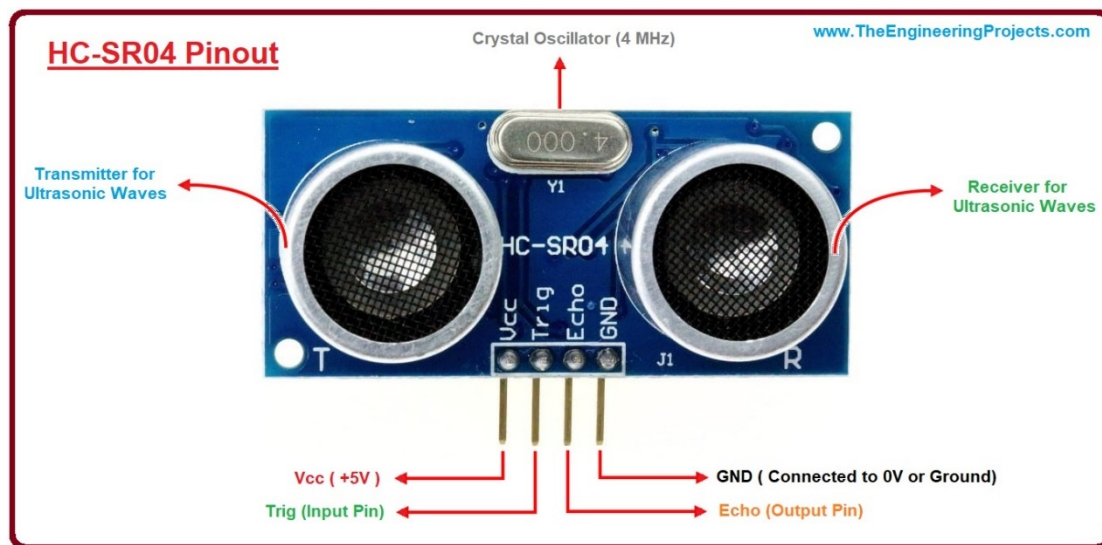
Εικόνα 3 : Διάγραμμα Arduino shield [20]

❖ Ο **Infrared Obstacle Avoidance Sensor** διαθέτει ένα ζεύγος αισθητήρων εκπομπής και λήψης υπέρυθρων. Το υπέρυθρο LED εκπέμπει σήματα υπέρυθρων σε συγκεκριμένη συχνότητα και όταν εμφανίζεται ένα εμπόδιο στη γραμμή υπέρυθρου φωτός, ανακλάται πίσω από το εμπόδιο που γίνεται αντιληπτό από τον δέκτη [15].



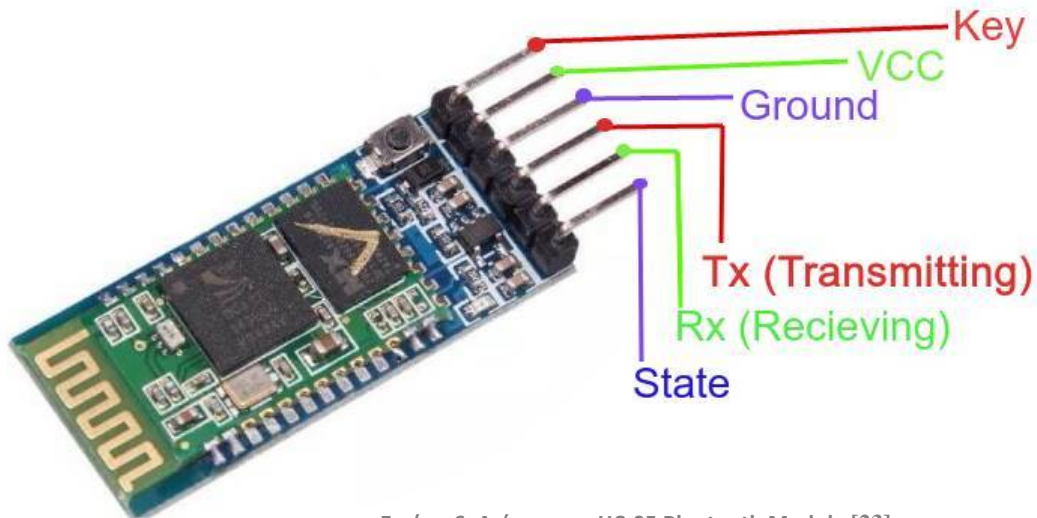
Εικόνα 4 : Διάγραμμα Infrared Obstacle Avoidance Sensor [21]

❖ Ο **HC-SR04 Ultrasonic Distance Sensor** είναι ένας αισθητήρας που χρησιμοποιείται για την ανίχνευση της απόστασης από ένα αντικείμενο χρησιμοποιώντας σόναρ. Είναι ιδανικό για οποιαδήποτε έργα ρομποτικής έχετε, που απαιτούν να αποφύγετε αντικείμενα, ανιχνεύοντας πόσο κοντά βρίσκονται, μπορείτε να απομακρυνθείτε από αυτά [14].



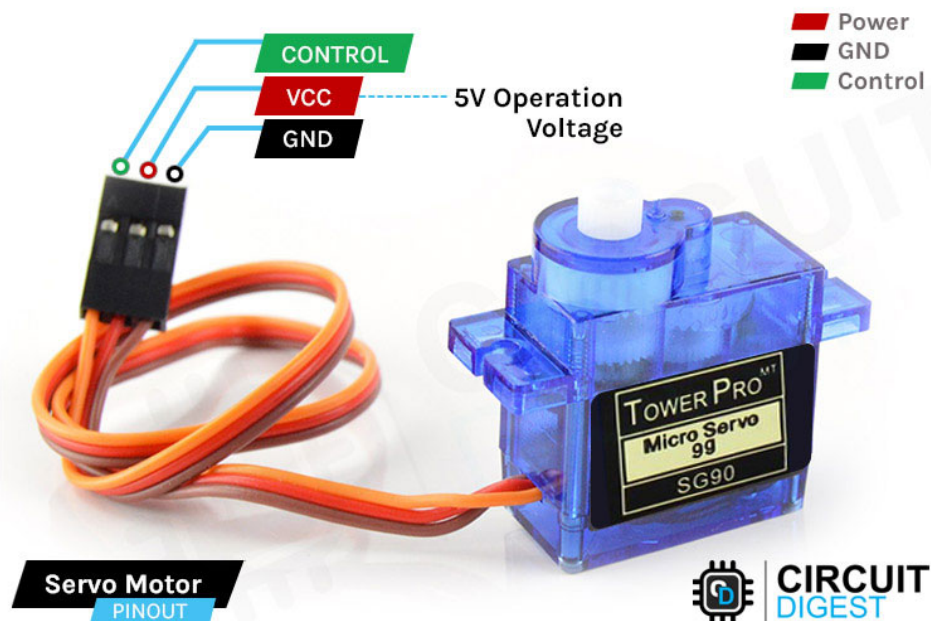
Εικόνα 5: Διάγραμμα HC-SR04 Ultrasonic Distance Sensor [22]

❖ Το **HC-05 Bluetooth Module** είναι μια εύχρηστη μονάδα Bluetooth SPP (Serial Port Protocol), σχεδιασμένη για διαφανή ρύθμιση ασύρματης σειριακής σύνδεσης. Η επικοινωνία του γίνεται μέσω σειριακής επικοινωνίας που καθιστά έναν εύκολο τρόπο διασύνδεσης με ελεγκτή ή υπολογιστή [16].



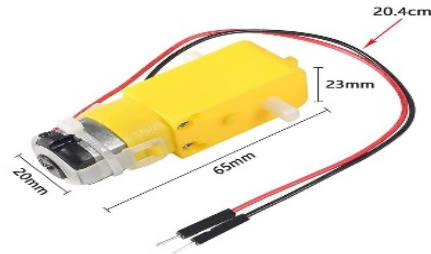
Εικόνα 6: Διάγραμμα HC-05 Bluetooth Module [23]

❖ Το **SG90** είναι ένας δημοφιλής σερβοκινητήρας μικρού μεγέθους που χρησιμοποιείται συνήθως σε έργα χομπίστας και DIY. Είναι ένας μικρός σερβοκινητήρας χαμηλού κόστους που μπορεί να περιστρέφεται 180 μοίρες με μέγιστη ροπή 1,8 kg-cm. Λειτουργεί στα 4,8-6V και έχει βάρος περίπου 9 γραμμάρια, καθιστώντας το ιδανικό για μικρής κλίμακας εφαρμογές ρομποτικής και ελέγχου μοντέλων [17].



Εικόνα 7: Διάγραμμα Servo motor SG90 [24]

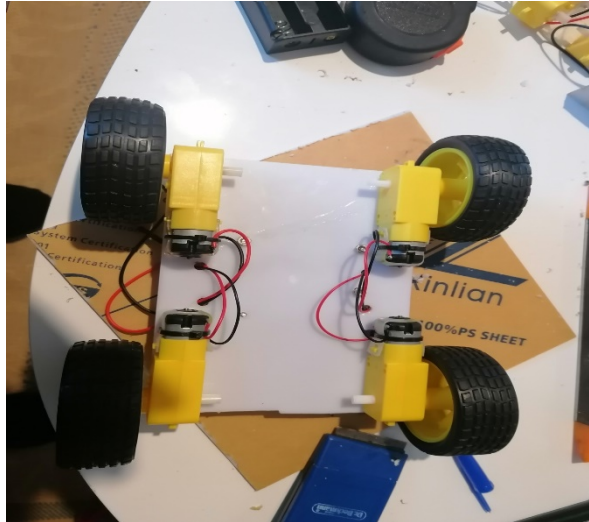
- Ο **κινητήρας μετάδοσης (DC Gear Motor TT 130 RPM)** είναι μια περιστρεφόμενη ηλεκτρομηχανική συσκευή που μετατρέπει την ηλεκτρική ενέργεια σε μηχανική ενέργεια. Όταν εφαρμόζεται συνεχής τάση στους ακροδέκτες του κινητήρα, ένας επαγωγέας (πηνίο) δημιουργεί ένα μαγνητικό πεδίο που δημιουργεί περιστροφική κίνηση



Εικόνα 8: κινητήρας μετάδοσης (DC Gear Motor TT 130 RPM) [25]

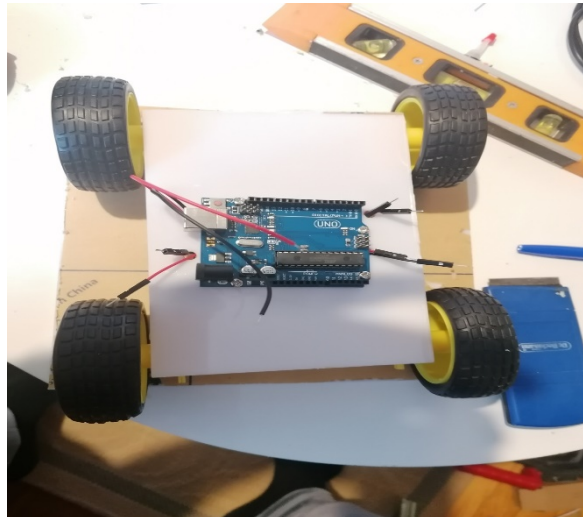
3.3 Κατασκευή του Ρομπότ

➤ Το σασί που θα αποτελεί το σκελετό του οχήματος μαζί με 4 ρόδες οι οποίες θα συνεισφέρουν στην κίνηση μαζί με τους κινητήρες που θα συνδεθούν σε αυτές.



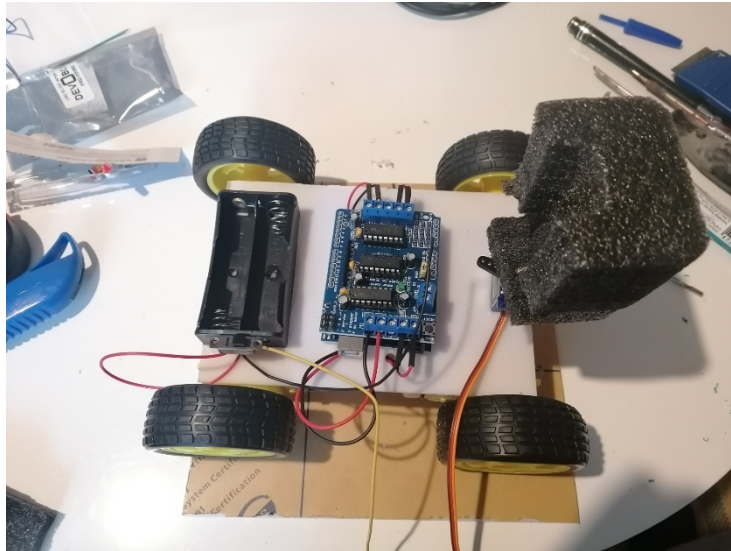
Εικόνα 9 : Εισαγωγή των κινητήρων μετάδοσης (DC Gear Motor TT 130 RPM) στο σασί του Ρομποτικού οχήματος

- Η πλακέτα (UNO R3 Atmega328P) τοποθετήθηκε πάνω στην βάση η οποία τρυπήθηκε για να περάσουν τα καλώδια των κινητήρων



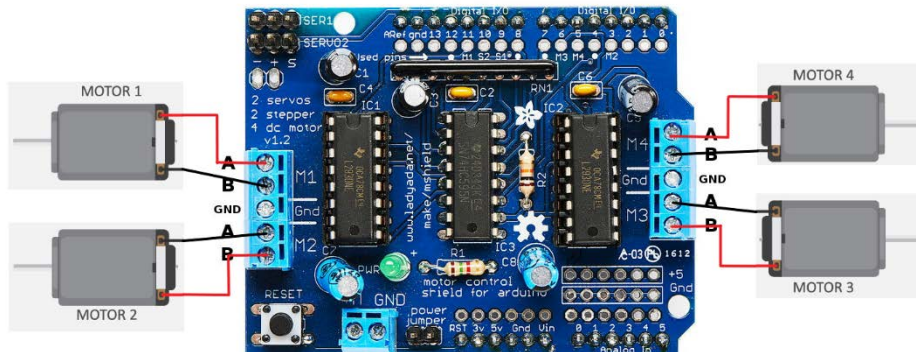
Εικόνα 10 Εισαγωγή της πλακέτας Arduino UNO στο σασί του Ρομποτικού οχήματος

- Τοποθέτηση του **Motor L293D Shield** το οποίο κουμπώνει πάνω στην πλακέτα και σύνδεση των κινητήρων στο shield. Επίσης στην παρακάτω εικόνα φαίνεται και η βάση για τις μπαταρίες όπως επίσης και το **SG90 Servo Micro Motor 9G**

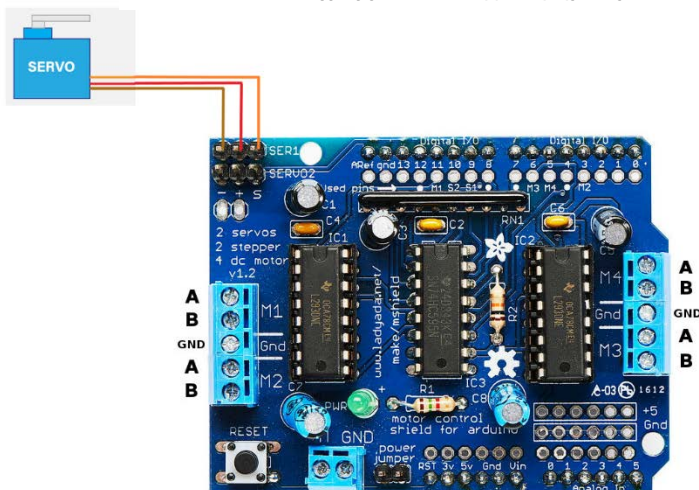


Εικόνα 11 : Εισαγωγή Arduino Shield,θήκη μπαταριών και Servo motor στο σασί του Ρομπωτικού οχήματος

- Στην παρακάτω εικόνα φαίνεται η σύνδεση των κινητήρων στο L293D Shield

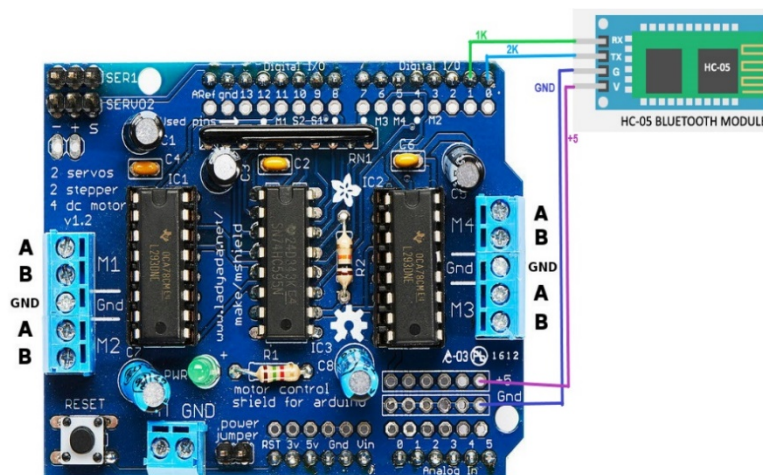
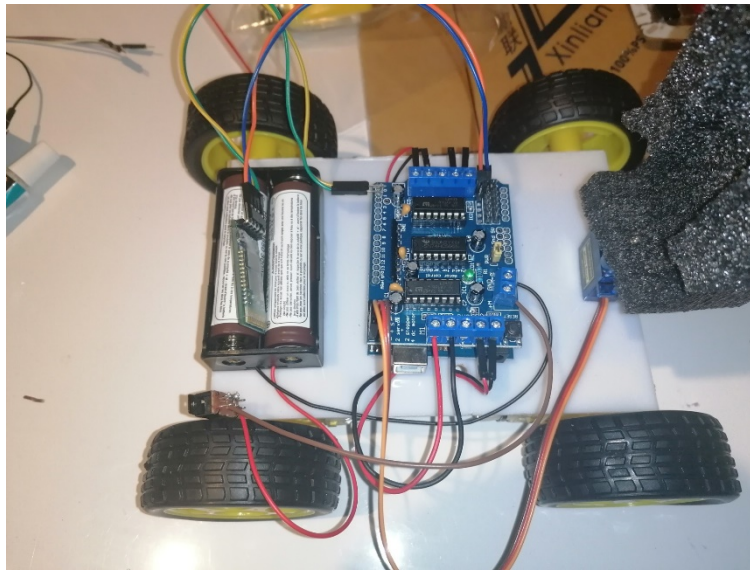


Εικόνα 13 : Διάγραμμα σύνδεσης κινητήρων μετάδοσης



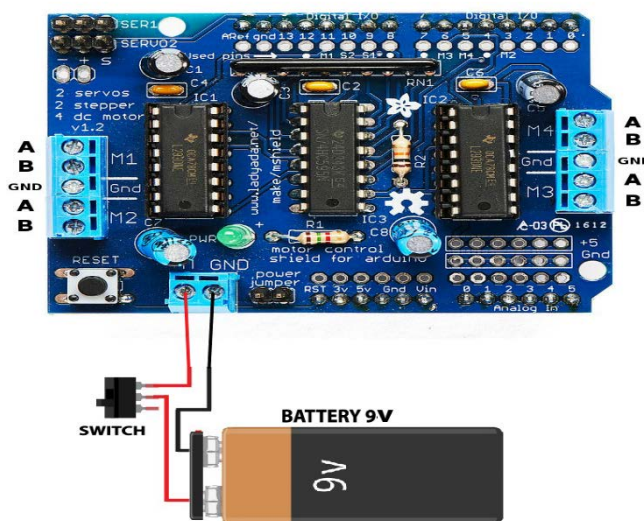
Εικόνα 12 : Διάγραμμα σύνδεσης Servo motor στο Shield

- Σύνδεση του **HC05 Bluetooth Module**, του διακόπτη (ο οποίος συνδέεται με το **Shield**) και των μπαταριών.



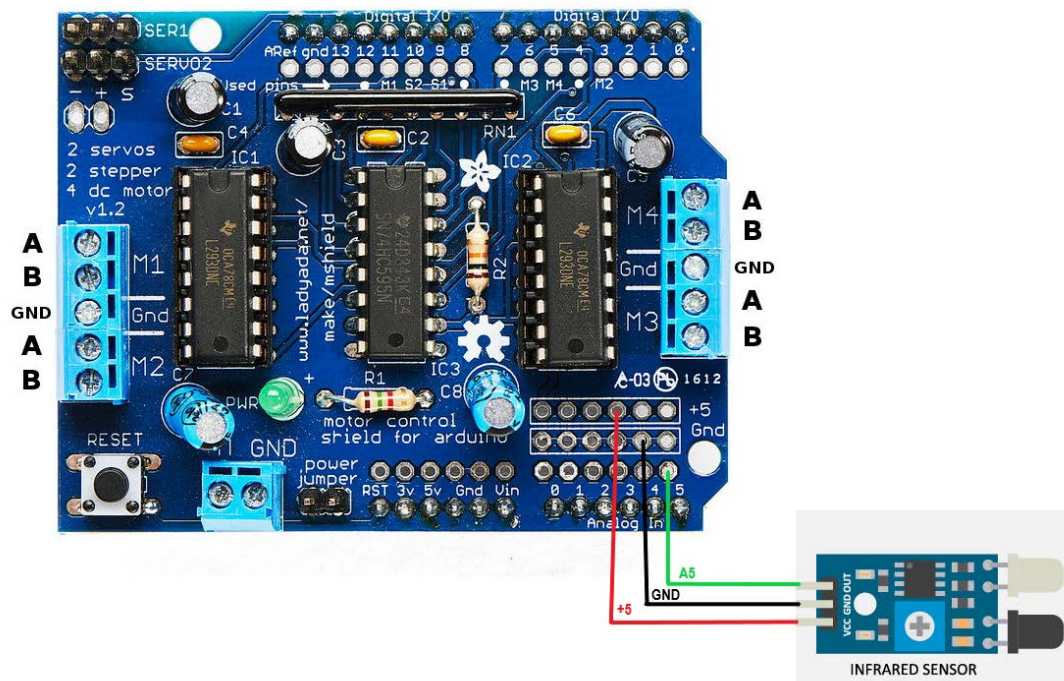
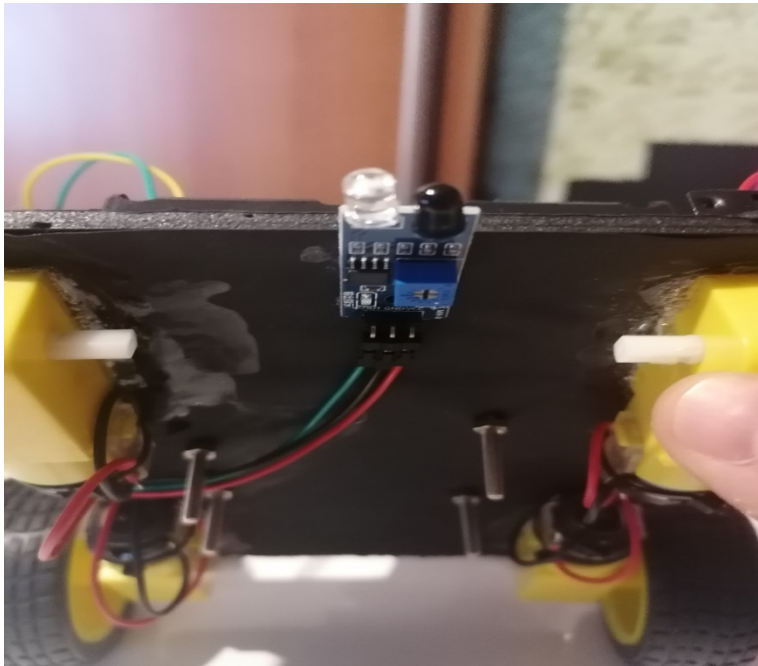
Η σύνδεση των καλωδίων έχει ως εξής :

- TX->TX
- RX->RX
- Vcc->5V
- GND->GND



Εικόνα 14 : Διάγραμμα σύνδεσης HC05 Bluetooth Module και μπαταρίας στο Shield

- Σύνδεση του **Infrared Obstacle Avoidance Sensor Module** στο πίσω μέρος του οχήματος.

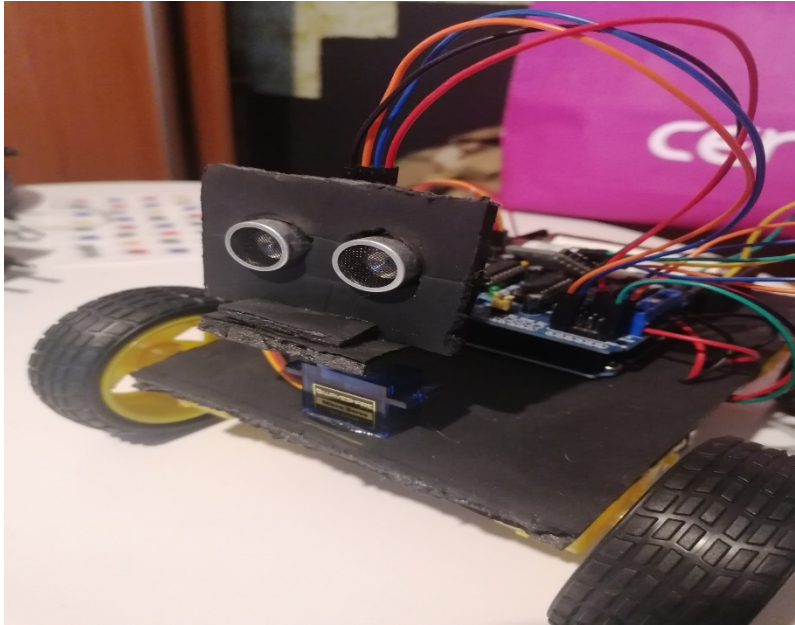


Εικόνα 16 : Διάγραμμα σύνδεσης του Infrared Obstacle Avoidance Sensor Module

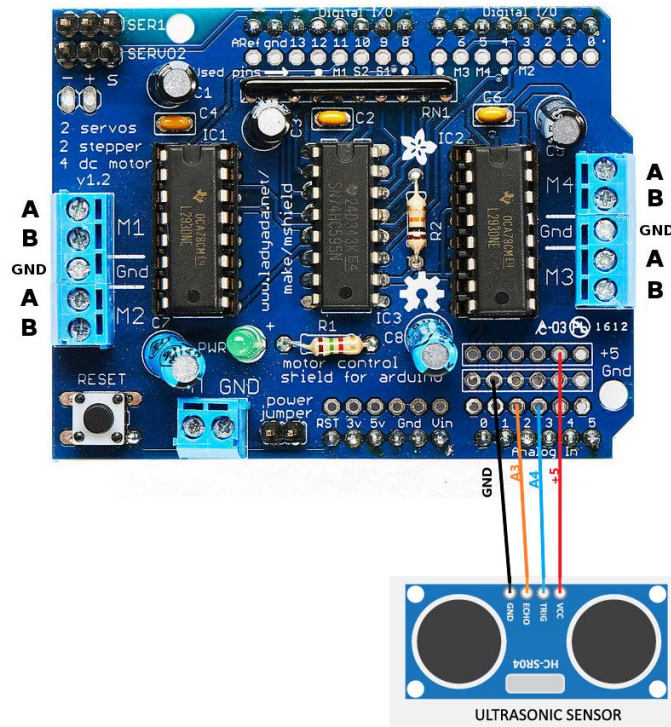
Η σύνδεση των καλωδίων έχει ως εξής :

- OUT->A5
- GND->GND
- VCC->5V

- Σύνδεση του **ElecFreaks HC-SR04 Ultrasonic Distance Module** στο μπροστά μέρος του οχήματος και τοποθέτηση σε βάση που δημιουργήθηκε με μακετόχαρτο



Εικόνα 17 : Εισαγωγή HC-SR04 Ultrasonic Distance Module



Η σύνδεση των καλωδίων έχει ως εξής :

- GND->GND
- ECHO->A3
- TRIG->A4
- VCC->+5

Εικόνα 18 : Διάγραμμα σύνδεσης HC-SR04 Ultrasonic Distance Module

3.4 Προγραμματισμός του Ρομπότ

Κάθε μικροελεγκτής χρειάζεται λογισμικό για να μπορεί να προγραμματιστεί. Η κατασκευή του κώδικα γίνεται με τη βοήθεια Software που όπως έχουμε αναφέρει ονομάζεται Integrated Development Environment (IDE) και μάλιστα η πλακέτα Arduino έχει δικό της IDE. Η εισαγωγή του κώδικα που κατασκευάζει ο χρήστης γίνεται με τη βοήθεια καλωδίου USB στο Arduino Uno R3 και από εκεί μέσω των αναλογικών και ψηφιακών pins μεταφέρεται στους αισθητήρες, shields και γενικότερα οποιαδήποτε συσκευή έχει συνδεθεί με τον μικροελεγκτή. Το περιβάλλον ανάπτυξης διαθέτει 3 στάδια, τον μεταγλωττιστή (compiler), τον επεξεργαστή κειμένου (text editor) και τον εντοπιστή σφαλμάτων (debugger). Ο text editor είναι η περιοχή όπου γράφεται ο κώδικας. Αφού ολοκληρωθεί ο κώδικας, πάνω αριστερά στο λογισμικό υπάρχει το εργαλείο επικύρωσης, το οποίο πατά ο χρήστης ώστε να σκανάρει τον κώδικα για ενδεχόμενα λάθη. Δίπλα στην επικύρωση βρίσκεται η επιλογή της μεταγλώττισης. Μέσω δηλαδή του compiler και του καλωδίου USB, μεταφέρεται ο κώδικας από τον υπολογιστή στην πλακέτα Arduino, αφού πρώτα έχει γίνει η επικύρωσή του. Το τελευταίο στάδιο είναι ο debugger, δηλαδή μία επιλογή που διαθέτει το Arduino να παρουσιάζουμε τα δεδομένα και τις μετρήσεις σε κάποιο monitor που θα έχουμε συνδέσει στην πλακέτα.

Όταν, λοιπόν, τροφοδοτείται η πλακέτα του μικροελεγκτή, το boot loader αρχικά περιμένει ένα πολύ μικρό χρονικό διάστημα για την εισαγωγή ενός νέου «sketch», όπως λέγεται ένα σύνολο γραμμών κώδικα στα συστήματα Arduino, από τη θύρα Arduino, μέσω του IDE. Εάν ανιχνεύσει sketch, αυτό περνά στη μνήμη flash και εκτελείται, διαφορετικά το boot loader μεταφέρει στη μνήμη flash το sketch που «ανέβασε» ο χρήστης την προηγούμενη φορά. Στα προϊόντα Arduino, υπάρχουν δύο περιβάλλοντα ανάπτυξης κώδικα (IDE), αναλόγως με τις προτιμήσεις και ανέσεις του χρήστη:

1. Το πρώτο ονομάζεται Arduino Web Editor και πρόκειται για πλατφόρμα online όπου μπορεί κανείς να προγραμματίσει τις συσκευές του και να αποθηκεύσει τα “sketches” του επίσης online, μέσω cloud. Το μεγαλύτερο πλεονέκτημα αυτού του IDE είναι πως η διαδικτυακή πλατφόρμα είναι πάντα αναβαθμισμένη στην τελευταία διαθέσιμη έκδοση. Έτσι, είναι μονίμως εκσυγχρονισμένη, όσον αφορά το λογισμικό αλλά και τις βιβλιοθήκες που διαθέτει, χωρίς να υποχρεώνεται ο χρήστης να κάνει ελέγχους για updates. Το μειονέκτημα είναι πως απαραίτητη προϋπόθεση είναι η ύπαρξη Internet.

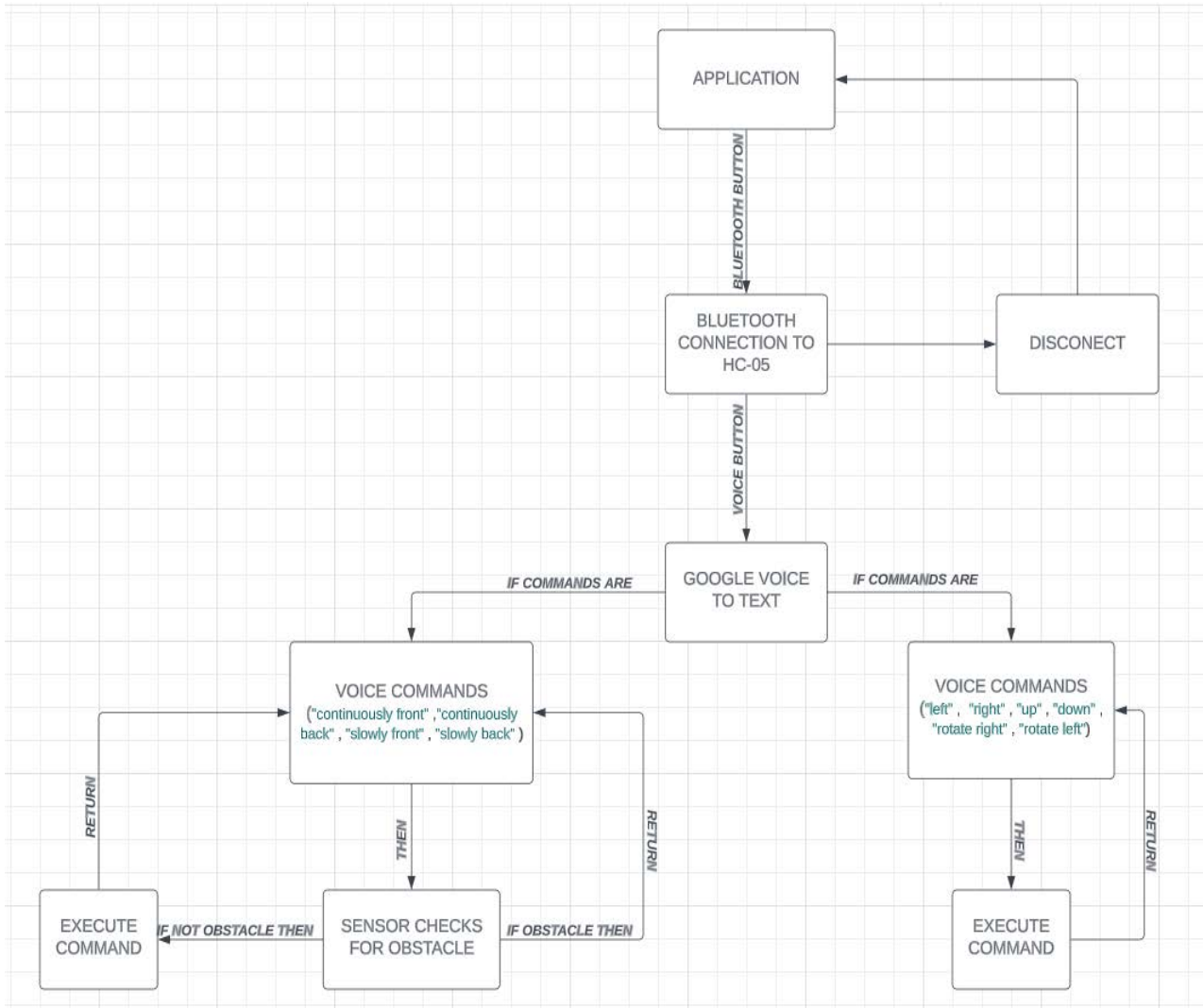
2. Το δεύτερο είναι λογισμικό που χρειάζεται «κατέβασμα» και εγκατάσταση από το χρήστη. Πρόκειται για IDE ανοιχτού κώδικα, δηλαδή προσβάσιμο από τον καθένα και διατίθεται ελεύθερο για αλλαγές και επιπλέον σχεδιασμό. Είναι εύκολο στη χρήση με πληθώρα επιλογών και ρυθμίσεων. Είναι συμβατό με Windows, Mac OS X και Linux. Συνολικά, το περιβάλλον είναι γραμμένο σε Java και μπορεί να χρησιμοποιηθεί με οποιαδήποτε πλακέτα Arduino. Ωστόσο, προγραμματιστής είναι υπεύθυνος να βρίσκει τις βιβλιοθήκες που χρειάζεται και να τις ενσωματώνει στο πρόγραμμα, καθώς και να αναβαθμίζει την πλατφόρμα.

Κεφάλαιο 4^ο – Τρόποι λειτουργίας του συστήματος

Για το χειρισμό του οχήματος ο χρήστης χρειάζεται το συγκεκριμένο ANDROID APP που δημιουργήθηκε. Ανοίγοντας το APP αντικρίζονται 2 επιλογές (Bluetooth και Speak). Επιλέγοντας την επιλογή Bluetooth εμφανίζονται οι διαθέσιμες συσκευές για σύνδεση, η συγκεκριμένη συσκευή θα εμφανιστεί ως HC-05. Μετά την επιλογή της συσκευής αν εμφανιστεί στην οθόνη με πράσινα γράμματα το CONNECTED τότε το όχημα έχει συνδεθεί με την συσκευή του χρήστη. Στην συνέχεια επιλέγοντας την επιλογή SPEAK ο χρήστης δίνει μια από της παρακάτω φωνητικές εντολές και το όχημα εκτελεί.

ΦΩΝΗΤΙΚΕΣ ΕΝΤΟΛΕΣ :

- 1) "continuously front"
- 2) "continuously back"
- 3) "slowly front"
- 4) "slowly back"
- 5) "left"
- 6) "right"
- 7) "up"
- 8) "down"
- 9) "rotate right"
- 10) "rotate left"



Εικόνα 19: Block diagram λειτουργίας του Ρομποτικού οχήματος

4.1 Κώδικας

Σε αυτό το παράρτημα ακολουθεί ο κώδικας που θα δώσει «ζωή» στο ρομπότ. Όπως αναφέρθηκε παραπάνω χρησιμοποιήθηκε μικροελεγκτής και λογισμικό Arduino με γλώσσα που αποτελεί μίξη των συναρτήσεων (functions) της C και C++.

4.1.1 Σύντομη περιγραφή κώδικα

✓ Ο κώδικας ξεκινάει με το κάλεσμα των απαραίτητων βιβλιοθηκών που χρησιμοποιούνται.

```
#include <AFMotor.h>
#include <NewPing.h>
#include <Servo.h>
#define TRIGGER_PIN A1
#define ECHO_PIN A0
#define MAX_DISTANCE 300
#define IR A5
```

✓ Ο κώδικας εδώ δηλώνει τα 4 motors, το sonar και το servo

```
AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
Servo myservo;
String voice;
```

✓ Αρχική θέση του Servo

```
void setup() {
  Serial.begin(9600);
  myservo.attach(10);
  myservo.write(90);
  pinMode(IR, INPUT);
}
void loop() {
  int distance = sonar.ping_cm();

  if(Serial.available()>0) {
    voice="";
    delay(2);
    voice = Serial.readString();
    delay(2);
    Serial.println(voice);
```

```
✓ Οι φωνητικές εντολές του κώδικα που δέχεται το όχημα για να εκτελέσει
if (voice == "left") {
left();
○ Left = στροφή 90 μοιρών αριστερά
}
else if(voice == "right") {
right();
○ Right = στροφή 90 μοιρών δεξιά
}
else if(voice == "up") {
up();
○ Up = Εντολή που εκτελείται ανεξαρτήτως εμποδίου και το όχημα κινείται
μπροστά για λίγα εκατοστά
}
else if(voice == "down") {
down();
○ Down = Εντολή που εκτελείται ανεξαρτήτως εμποδίου και το όχημα κινείται
πίσω για λίγα εκατοστά
}
else if(voice == "rotate right") {
rotateright();
○ rotate right = περιστροφή 180 μοιρών του οχήματος προς τα δεξιά
}
else if(voice == "rotate left") {
rotateleft();
○ rotate left = περιστροφή 180 μοιρών του οχήματος προς τα αριστερά
}
}
while(voice == "continuously front") {
forward();
○ continuously front = Συνεχής κίνηση εμπρός έως ότου βρει κάποιο εμπόδιο
}
while(voice == "continuously back") {
backward();
○ continuously back = Συνεχής κίνηση πίσω έως ότου βρει κάποιο εμπόδιο
}
while(voice == "slowly front") {
slowlyfront();
○ slowly front = Συνεχής κίνηση σε χαμηλή ταχύτητα εμπρός έως ότου βρει
κάποιο εμπόδιο
}
while(voice == "slowly back") {
slowlyback();
}
}
```


- `slowly back` = Συνεχής κίνηση σε χαμηλή ταχύτητα πίσω έως ότου βρει κάποιο εμπόδιο
-

✓ Η εντολή αυτή δίνει την δυνατότητα στο όχημα να σταματήσει την κίνηση του εάν υπάρξει εμπόδιο μπροστά του (είναι η επιστροφή της εντολής (`continuously front`))

```
}  
void forward() {  
  int distance = sonar.ping_cm();  
  if(distance < 20){  
    Stop();  
    voice="";  
  }  
}
```

✓ Εάν δεν υπάρξει εμπόδιο τότε θα εκτελέσει αυτήν την εντολή οπου θα δώσει κίνηση στους κινητήρες προς τα εμπρός έως ότου συναντήσει εμπόδιο

```
else {  
  motor1.setSpeed(255);  
  motor1.run(FORWARD);  
  motor2.setSpeed(255);  
  motor2.run(FORWARD);  
  motor3.setSpeed(255);  
  motor3.run(FORWARD);  
  motor4.setSpeed(255);  
  motor4.run(FORWARD);  
}
```

✓ επιστροφή της εντολής (`up`) οπου δίνει κίνηση στους κινητήρες προς τα εμπρός για λίγο χρόνο

```
void up(){  
  motor1.setSpeed(255);  
  motor1.run(FORWARD);  
  motor2.setSpeed(255);  
  motor2.run(FORWARD);  
  motor3.setSpeed(255);  
  motor3.run(FORWARD);  
  motor4.setSpeed(255);  
  motor4.run(FORWARD);  
  delay(500);  
  motor1.run(RELEASE);  
  motor2.run(RELEASE);  
  motor3.run(RELEASE);  
  motor4.run(RELEASE);  
}
```

✓ Η εντολή αυτή δίνει την δυνατότητα στο όχημα να σταματήσει την κίνηση του εάν υπάρξει εμπόδιο μπροστά του (είναι η επιστροφή της εντολής (**slowly front**))

```
void slowlyfront() {  
  int distance = sonar.ping_cm();  
  if(distance < 20){  
    Stop();  
    voice="";  
  }  
  else
```

✓ Εάν δεν υπάρξει εμπόδιο τότε θα εκτελέσει αυτήν την εντολή όπου θα δώσει κίνηση στους κινητήρες σε χαμηλή ταχύτητα προς τα εμπρός έως ότου συναντήσει εμπόδιο

```
{  
  motor1.setSpeed(100);  
  motor1.run(FORWARD);  
  motor2.setSpeed(100);  
  motor2.run(FORWARD);  
  motor3.setSpeed(100);  
  motor3.run(FORWARD);  
  motor4.setSpeed(100);  
  motor4.run(FORWARD);  
}
```

✓ Η εντολή αυτή δίνει την δυνατότητα στο όχημα να σταματήσει την κίνηση του εάν υπάρξει εμπόδιο πίσω του (είναι η επιστροφή της εντολής (**continuously back**))

```
}  
void backward() {  
  int IR_Sensor = digitalRead(IR);  
  if(IR_Sensor == 0) {  
    Stop();  
    voice="";  
  }else
```

✓ Εάν δεν υπάρξει εμπόδιο τότε θα εκτελέσει αυτήν την εντολή όπου θα δώσει κίνηση στους κινητήρες προς τα πίσω έως ότου συναντήσει εμπόδιο

```
{  
  motor1.setSpeed(255);  
  motor1.run(BACKWARD);  
  motor2.setSpeed(255);  
  motor2.run(BACKWARD);  
  motor3.setSpeed(255);  
  motor3.run(BACKWARD);  
  motor4.setSpeed(255);  
  motor4.run(BACKWARD);  
}  
}
```

✓ επιστροφή της εντολής (**down**) οπού δίνει κίνηση στους κινητήρες προς τα πίσω για λίγο χρόνο

```
void down(){
  motor1.setSpeed(255);
  motor1.run(BACKWARD);
  motor2.setSpeed(255);
  motor2.run(BACKWARD);
  motor3.setSpeed(255);
  motor3.run(BACKWARD);
  motor4.setSpeed(255);
  motor4.run(BACKWARD);
  delay(500);
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
}
```

✓ Η εντολή αυτή δίνει την δυνατότητα στο όχημα να σταματήσει την κίνηση του εάν υπάρξει εμπόδιο μπροστά του (είναι η επιστροφή της εντολής (**slowly front**))

```
void slowlyback() {
  int IR_Sensor = digitalRead(IR);
  if(IR_Sensor == 0) {
    Stop();
    voice="";
  }
  else
```

✓ Εάν δεν υπάρξει εμπόδιο τότε θα εκτελέσει αυτήν την εντολή οπού θα δώσει κίνηση στους κινητήρες σε χαμηλή ταχύτητα προς τα εμπρός έως ότου συναντήσει εμπόδιο

```
{
  motor1.setSpeed(100);
  motor1.run(BACKWARD);
  motor2.setSpeed(100);
  motor2.run(BACKWARD);
  motor3.setSpeed(100);
  motor3.run(BACKWARD);
  motor4.setSpeed(100);
  motor4.run(BACKWARD);
}
}
```

✓ επιστροφή της εντολής (**right**) οπού δίνει κίνηση μόνο στους αριστερούς κινητήρες προς τα εμπρός έτσι ώστε το όχημα να στρίψει προς τα δεξιά

```
void right() {  
myservo.write(0);  
delay(500);  
myservo.write(90);  
delay(500);  
motor1.run(BACKWARD);  
motor1.setSpeed(255);  
motor2.run(BACKWARD);  
motor2.setSpeed(255);  
motor3.run(FORWARD);  
motor3.setSpeed(255);  
motor4.run(FORWARD);  
motor4.setSpeed(255);  
delay(500);  
motor1.run(RELEASE);  
motor2.run(RELEASE);  
motor3.run(RELEASE);  
motor4.run(RELEASE);  
}
```

✓ επιστροφή της εντολής (**left**) οπού δίνει κίνηση μόνο στους δεξιούς κινητήρες προς τα εμπρός έτσι ώστε το όχημα να στρίψει προς τα αριστερά

```
void left() {  
myservo.write(180);  
delay(500);  
myservo.write(90);  
delay(500);  
motor1.run(FORWARD);  
motor1.setSpeed(255);  
motor2.run(FORWARD);  
motor2.setSpeed(255);  
motor3.run(BACKWARD);  
motor3.setSpeed(255);  
motor4.run(BACKWARD);  
motor4.setSpeed(255);  
delay(500);  
motor1.run(RELEASE);  
motor2.run(RELEASE);  
motor3.run(RELEASE);  
motor4.run(RELEASE);  
}
```

✓ επιστροφή της εντολής (**rotateright**) οπού δίνει κίνηση εμπρός στους αριστερούς κινητήρες και κίνηση πίσω στους δεξιούς κινητήρες έτσι ώστε το όχημα να κάνει στροφή 180 μοιρών προς τα δεξιά

```
void rotateright() {  
  myservo.write(0);  
  delay(500);  
  myservo.write(90);  
  delay(500);  
  motor1.run(BACKWARD);  
  motor1.setSpeed(255);  
  motor2.run(BACKWARD);  
  motor2.setSpeed(255);  
  motor3.run(FORWARD);  
  motor3.setSpeed(255);  
  motor4.run(FORWARD);  
  motor4.setSpeed(255);  
  delay(1000);  
  motor1.run(RELEASE);  
  motor2.run(RELEASE);  
  motor3.run(RELEASE);  
  motor4.run(RELEASE);  
}
```

✓ επιστροφή της εντολής (`rotateleft`) οπού δίνει κίνηση εμπρός στους δεξιούς κινητήρες και κίνηση πίσω στους αριστερούς κινητήρες έτσι ώστε το όχημα να κάνει στροφή 180 μοιρών προς τα αριστερά

```
void rotateleft() {  
  myservo.write(180);  
  delay(500);  
  myservo.write(90);  
  delay(500);  
  motor1.run(FORWARD);  
  motor1.setSpeed(255);  
  motor2.run(FORWARD);  
  motor2.setSpeed(255);  
  motor3.run(BACKWARD);  
  motor3.setSpeed(255);  
  motor4.run(BACKWARD);  
  motor4.setSpeed(255);  
  delay(1000);  
  motor1.run(RELEASE);  
  motor2.run(RELEASE);  
  motor3.run(RELEASE);  
  motor4.run(RELEASE);  
}
```

✓ Η εντολή αυτή σταματά το όχημα αμέσως από οποιαδήποτε κίνηση.

```
void Stop() {  
  motor1.run(RELEASE);
```

```
motor2.run(RELEASE);  
motor3.run(RELEASE);  
motor4.run(RELEASE);  
}
```

4.1.2 Κλήση βιβλιοθηκών

`#include <AFMotor.h>`

Αυτή η βιβλιοθήκη παρέχει έλεγχο ταχύτητας και κατεύθυνσης για έως και τέσσερις κινητήρες DC όταν χρησιμοποιείται με το Adafruit Motor Shield

`#include <NewPing.h>`

Αυτή η βιβλιοθήκη απλοποιεί τη διαδικασία χρήσης αισθητήρων υπερήχων και διευκολύνει την εφαρμογή τους.

`#include <Servo.h>`

Επιτρέπει στις πλακέτες Arduino να ελέγχουν μια ποικιλία σερβοκινητήρων. Αυτή η βιβλιοθήκη μπορεί να ελέγξει μεγάλο αριθμό σερβομηχανισμών.

4.2 Παράρτημα κώδικα

```
#include <AFMotor.h>
#include <NewPing.h>
#include <Servo.h>
//Libraries
#define TRIGGER_PIN A1
#define ECHO_PIN A0
#define MAX_DISTANCE 300
#define IR A5

AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);
//DC MOTORS

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
Servo myservo;
String voice;

void setup() {
  Serial.begin(9600);
  myservo.attach(10);
  myservo.write(90);
  pinMode(IR, INPUT);
  //Servo starting position
}
void loop() {
  int distance = sonar.ping_cm();

  if(Serial.available()>0) {
    voice="";
    delay(2);
    voice = Serial.readString();
    delay(2);
    Serial.println(voice);

    if (voice == "left") {
      left();
    }
    else if(voice == "right") {
      right();
    }
    else if(voice == "up") {
      up();
    }
    else if(voice == "down") {
      down();
    }
  }
}
```

```
}  
else if(voice == "rotate right") {  
  rotateright();  
}  
else if(voice == "rotate left") {  
  rotateleft();  
}  
  
}  
while(voice == "continuously front") {  
  forward();  
}  
while(voice == "continuously back") {  
  backward();  
}  
while(voice == "slowly front") {  
  slowlyfront();  
}  
while(voice == "slowly back") {  
  slowlyback();  
}  
//Functions commands  
}  
void forward() {  
  int distance = sonar.ping_cm();  
  if(distance < 20){  
    Stop();  
    voice="";  
  }  
  //Distance for sensor to stop  
  }  
  else {  
    motor1.setSpeed(255);  
    motor1.run(FORWARD);  
    motor2.setSpeed(255);  
    motor2.run(FORWARD);  
    motor3.setSpeed(255);  
    motor3.run(FORWARD);  
    motor4.setSpeed(255);  
    motor4.run(FORWARD);  
  }  
  //motors speed  
  }  
  }  
void up(){  
  motor1.setSpeed(255);  
  motor1.run(FORWARD);  
  motor2.setSpeed(255);  
  motor2.run(FORWARD);
```



```
motor3.setSpeed(255);
motor3.run(FORWARD);
motor4.setSpeed(255);
motor4.run(FORWARD);
  delay(500);
motor1.run(RELEASE);
motor2.run(RELEASE);
motor3.run(RELEASE);
motor4.run(RELEASE);
}

void slowlyfront() {
  int distance = sonar.ping_cm();
  if(distance < 20){
    Stop();
    voice="";
  }else {
    motor1.setSpeed(100);
    motor1.run(FORWARD);
    motor2.setSpeed(100);
    motor2.run(FORWARD);
    motor3.setSpeed(100);
    motor3.run(FORWARD);
    motor4.setSpeed(100);
    motor4.run(FORWARD);
  }
}

void backward() {
  int IR_Sensor = digitalRead(IR);
  if(IR_Sensor == 0) {
    Stop();
    voice="";
  }else {
    motor1.setSpeed(255);
    motor1.run(BACKWARD);
    motor2.setSpeed(255);
    motor2.run(BACKWARD);
    motor3.setSpeed(255);
    motor3.run(BACKWARD);
    motor4.setSpeed(255);
    motor4.run(BACKWARD);
  }
}

void down(){
  motor1.setSpeed(255);
  motor1.run(BACKWARD);
  motor2.setSpeed(255);
  motor2.run(BACKWARD);
}
```

```
motor3.setSpeed(255);
motor3.run(BACKWARD);
motor4.setSpeed(255);
motor4.run(BACKWARD);
  delay(500);
motor1.run(RELEASE);
motor2.run(RELEASE);
motor3.run(RELEASE);
motor4.run(RELEASE);
}
void slowlyback() {
  int IR_Sensor = digitalRead(IR);
  if(IR_Sensor == 0) {
    Stop();
    voice="";
  }else {
    motor1.setSpeed(100);
    motor1.run(BACKWARD);
    motor2.setSpeed(100);
    motor2.run(BACKWARD);
    motor3.setSpeed(100);
    motor3.run(BACKWARD);
    motor4.setSpeed(100);
    motor4.run(BACKWARD);
  }
}

void right() {
  myservo.write(0);
  delay(500);
  myservo.write(90);
  delay(500);
  motor1.run(BACKWARD);
  motor1.setSpeed(255);
  motor2.run(BACKWARD);
  motor2.setSpeed(255);
  motor3.run(FORWARD);
  motor3.setSpeed(255);
  motor4.run(FORWARD);
  motor4.setSpeed(255);
  delay(500);
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
}

void left() {
```

```
myservo.write(180);
delay(500);
myservo.write(90);
delay(500);
motor1.run(FORWARD);
motor1.setSpeed(255);
motor2.run(FORWARD);
motor2.setSpeed(255);
motor3.run(BACKWARD);
motor3.setSpeed(255);
motor4.run(BACKWARD);
motor4.setSpeed(255);
delay(500);
motor1.run(RELEASE);
motor2.run(RELEASE);
motor3.run(RELEASE);
motor4.run(RELEASE);
}
void rotateright() {
myservo.write(0);
delay(500);
myservo.write(90);
delay(500);
motor1.run(BACKWARD);
motor1.setSpeed(255);
motor2.run(BACKWARD);
motor2.setSpeed(255);
motor3.run(FORWARD);
motor3.setSpeed(255);
motor4.run(FORWARD);
motor4.setSpeed(255);
delay(1000);
motor1.run(RELEASE);
motor2.run(RELEASE);
motor3.run(RELEASE);
motor4.run(RELEASE);
}
void rotateleft() {
myservo.write(180);
delay(500);
myservo.write(90);
delay(500);
motor1.run(FORWARD);
motor1.setSpeed(255);
motor2.run(FORWARD);
motor2.setSpeed(255);
motor3.run(BACKWARD);
motor3.setSpeed(255);
```

```
motor4.run(BACKWARD);  
motor4.setSpeed(255);  
delay(1000);  
motor1.run(RELEASE);  
motor2.run(RELEASE);  
motor3.run(RELEASE);  
motor4.run(RELEASE);  
}  
void Stop() {  
motor1.run(RELEASE);  
motor2.run(RELEASE);  
motor3.run(RELEASE);  
motor4.run(RELEASE);
```

Κεφάλαιο 5^ο - Εφαρμογή Android φωνητικών εντολών

Το Application υλοποιήθηκε με την πλατφόρμα **MIT App Inventor** το οποίο είναι ένα διαισθητικό, οπτικό περιβάλλον προγραμματισμού που επιτρέπει σε όλους να δημιουργούν πλήρως λειτουργικές εφαρμογές για smartphone και tablet. Όσοι είναι νέοι στο MIT App Inventor μπορούν να έχουν μια απλή πρώτη εφαρμογή σε λειτουργία σε λιγότερο από 30 λεπτά. Χρησιμοποιήθηκε η συγκεκριμένη πλατφόρμα για την πιο εύκολη υλοποίηση του app καθώς είναι μια απλή εφαρμογή μετατροπής voice σε text και δεν χρειάζεται κάτι πιο πολύπλοκο.



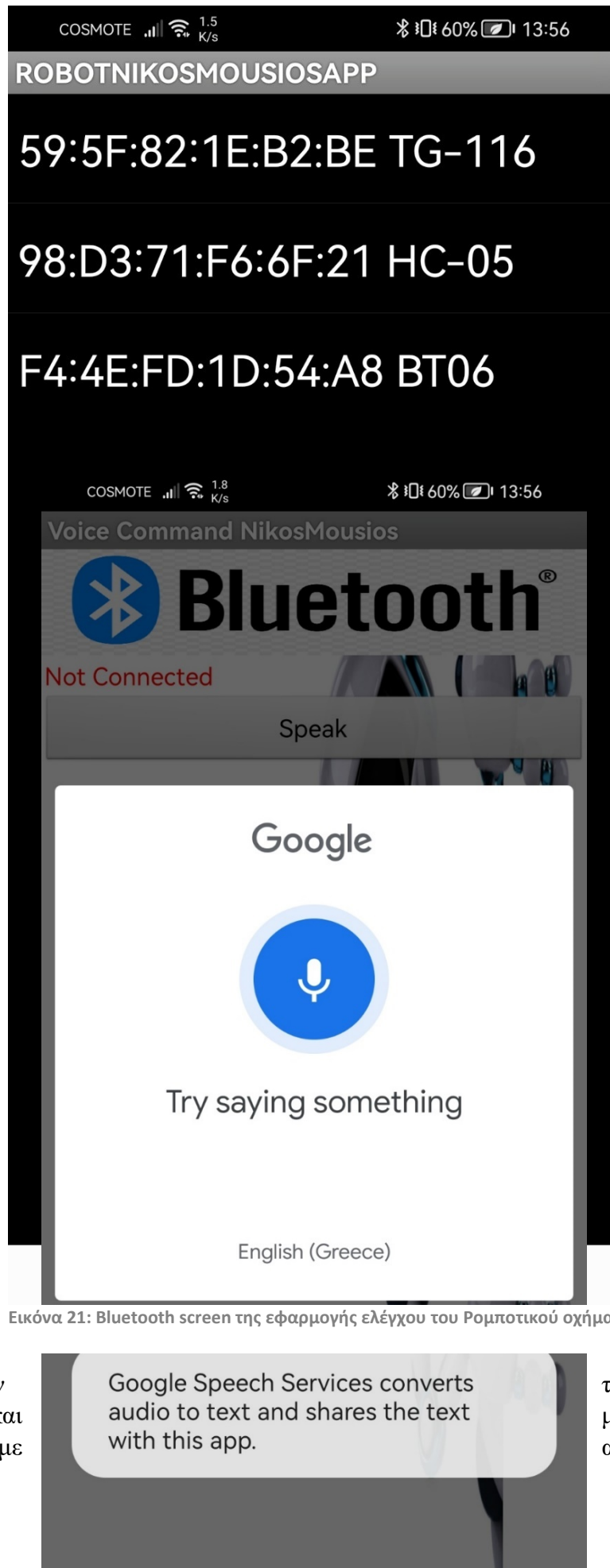
Εικόνα 20: Main screen της εφαρμογής ελέγχου του Ρομποτικού οχήματος

Όπως βλέπουμε στην παραπάνω εικόνα το App περιέχει 2 κουμπιά.

Κουμπιά (Buttons) :

Bluetooth -> το κουμπί Bluetooth μας μεταφέρει στην καρτέλα με τις διαθέσιμες συσκευές Bluetooth που είναι για σύνδεση, εάν το App συνδεθεί με κάποια συσκευή τότε θα εμφανιστεί η σήμανση **Connected** .

Speak -> το κουμπί Speak μας μεταφέρει στην υπηρεσία της google και μας δίνει την δυνατότητα να δώσουμε την φωνητική εντολή και αυτή να μετατραπεί σε text και να σταλεί στην συσκευή μας για εκτέλεση



❖ Οι ομιλίας Google

μετατρέπουν κείμενο και κείμενο με

Εικόνα 21: Bluetooth screen της εφαρμογής ελέγχου του Ρομποτικού οχήματος

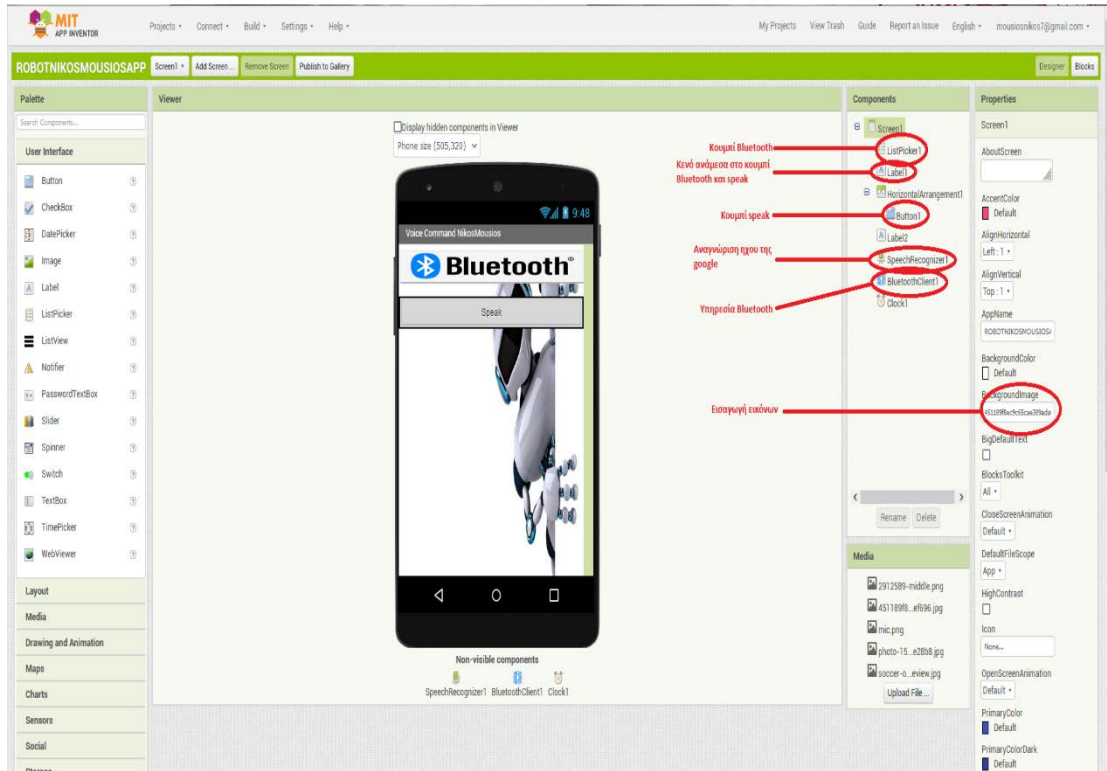
υπηρεσίες της

τον ήχο σε μοιράζεται το αυτήν την

εφαρμογή

5.2 Υλοποίηση Εφαρμογής

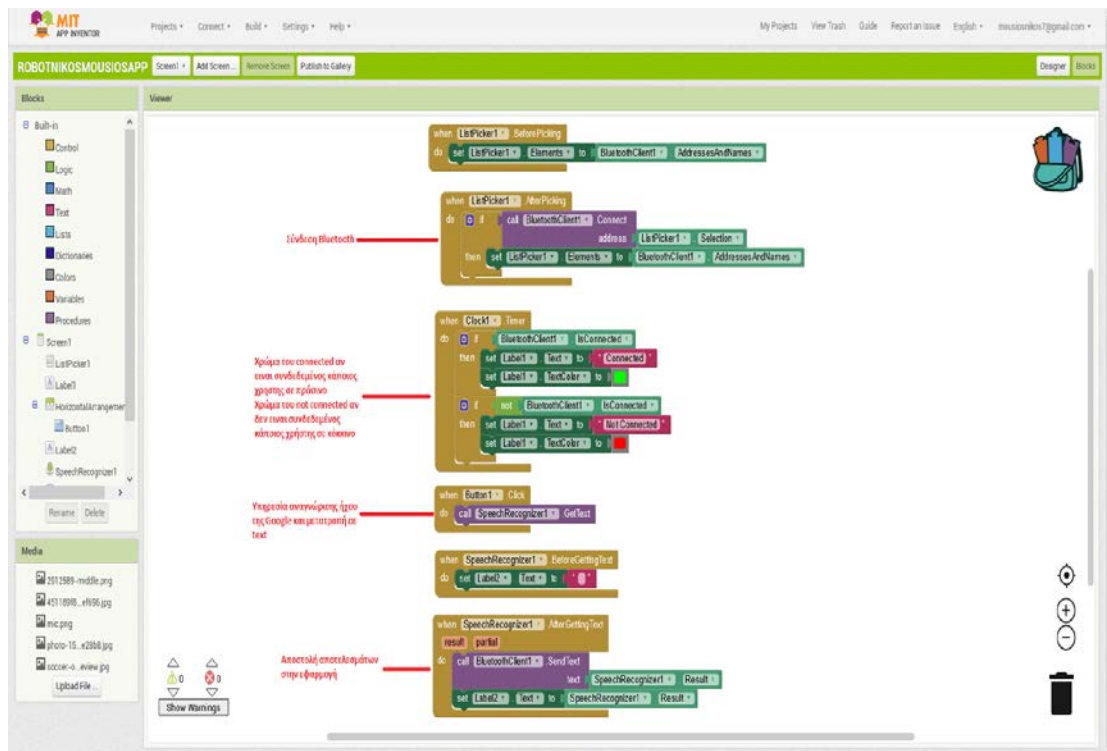
➤ App Design:



Εικόνα 23: Πλατφόρμα MIT INVENTOR για την σχεδίαση της εφαρμογής ελέγχου του Ρομπωτικού

❖ Στην παραπάνω εικόνα φαίνεται η σχεδίαση του interface της εφαρμογής στην πλατφόρμα MIT APP INVENTOR.

➤ Κώδικας Εφαρμογής:



Εικόνα 24: Ο κώδικας της εφαρμογής ελέγχου του Ρομπωτικού οχήματος στην πλατφόρμα MIT INVENTOR

➤ Περιγραφή κώδικα εφαρμογής:

```
when ListPicker1 .BeforePicking
do set ListPicker1 .Elements to BluetoothClient1 .AddressesAndNames
```

- Σε αυτό το κομμάτι κώδικα υλοποιείται το κουμπί που όταν πατηθεί θα εμφανίζει τις διαθέσιμες συσκευές Bluetooth για συνδέσει.
- Σε αυτό το κομμάτι κώδικα, εφόσον διαλέξουμε την συσκευή Bluetooth, γίνεται η σύνδεση με την συσκευή μας

```
when ListPicker1 .AfterPicking
do if call BluetoothClient1 .Connect
    address ListPicker1 . Selection
then set ListPicker1 .Elements to BluetoothClient1 .AddressesAndNames
```

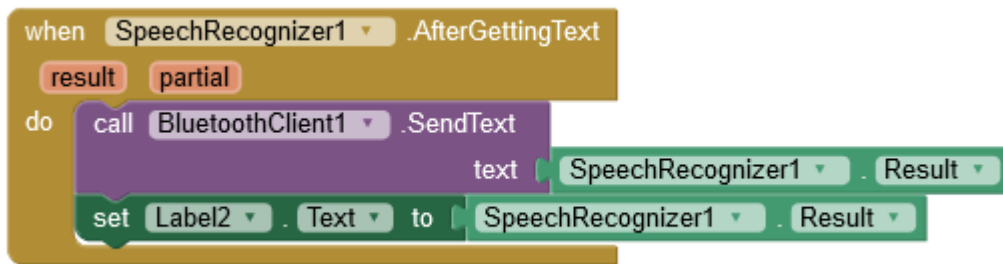
```
when Clock1 .Timer
do if BluetoothClient1 .IsConnected
then set Label1 .Text to "Connected"
    set Label1 .TextColor to green
if not BluetoothClient1 .IsConnected
then set Label1 .Text to "Not Connected"
    set Label1 .TextColor to red
```

- Σε αυτό το κομμάτι κώδικα εμφανίζεται το αν είναι υπάρχει κάποια σύνδεση με κάποια συσκευή Bluetooth με την σήμανση (Connected) αν είναι συνδεδεμένο και αν όχι με την σήμανση (Not Connected) , καθώς και την επιλογή του χρώματος σήμανσης που όπως φαίνεται και στην εικόνα επιλέγηκε το πράσινο και το κόκκινο.

```
when Button1 .Click
do call SpeechRecognizer1 .GetText
```

```
when SpeechRecognizer1 .BeforeGettingText
do set Label2 .Text to " "
```

- Σε αυτό το κομμάτι κώδικα προγραμματίζεται το Speech to Text δηλαδή η μετατροπή των φωνητικών εντολών σε κείμενο



```
when SpeechRecognizer1 .AfterGettingText
  result partial
  do
    call BluetoothClient1 .SendText
      text SpeechRecognizer1 . Result
    set Label2 . Text to SpeechRecognizer1 . Result
```

- Σε αυτό το κομμάτι κώδικα στέλνονται τα αποτελέσματα του Speech to Text μέσω Bluetooth στην συσκευή για την εκτέλεση της εντολής.

Κεφάλαιο 6^ο - Συμπεράσματα και μελλοντικές βελτιώσεις

6.1 Σύνοψη και συμπεράσματα

Η συγκεκριμένη πτυχιακή εργασία παρουσιάζει την σχεδίαση και την υλοποίηση ενός ρομποτικού οχήματος με την χρήση της πλακέτας Arduino. Η εργασία διαχωρίστηκε σε δύο μέρη. Το πρώτο μέρος ήταν ο σχεδιασμός, η μορφοποίηση του οχήματος και η σύνδεση του κυκλώματος με την πλακέτα. Το δεύτερο μέρος ήταν ο προγραμματισμός της πλακέτας, έτσι ώστε ο χρήστης δώσει εντολή και να έχει τα επιθυμητά αποτελέσματα.

Οι δυνατότητες του οχήματος είναι η εκτέλεση των εντολών που δίνει ο χρήστης οι οποίες είναι:

- "continuously front"
- "continuously back"
- "slowly front"
- "slowly back"
- "left"
- "right"
- "up"
- "down"
- "rotate right"
- "rotate left"

Το όχημα θα σταματήσει αν βρεθεί κάποιο εμπόδιο κατά την κίνηση του εκτός και αν ο χρήστης χρησιμοποιήσει τις εντολές (up ή down), οι οποίες χρησιμοποιούνται κυρίως όταν το έδαφος είναι ανώμαλο η ανηφορικό.

Αναλυτική περιγραφή εντολών:

- **"continuously front"** -> Η εντολή αυτή παραθέτει στο όχημα να κινείται συνεχώς μπροστά μέχρι να βρεθεί εμπόδιο μπροστά του.
- **"continuously back"** -> Η εντολή αυτή παραθέτει στο όχημα να κινείται συνεχώς πίσω μέχρι να βρεθεί εμπόδιο πίσω του.
- **"slowly front"** -> Η εντολή αυτή παραθέτει στο όχημα να κινείται συνεχώς μπροστά σε αργή κίνηση μέχρι να βρεθεί εμπόδιο μπροστά του.
- **"slowly back"** -> Η εντολή αυτή παραθέτει στο όχημα να κινείται συνεχώς πίσω σε αργή κίνηση μέχρι να βρεθεί εμπόδιο πίσω του.
- **"left"** -> Η εντολή αυτή παραθέτει στο όχημα να στρίψει 90' μοίρες προς τα αριστερά.
- **"right"** -> Η εντολή αυτή παραθέτει στο όχημα να στρίψει 90' μοίρες προς τα δεξιά.
- **"up"** -> Η εντολή αυτή παραθέτει στο όχημα να κινηθεί μπροστά για ελάχιστο χρόνο ανεξαρτήτως εμποδίου
- **"down"** -> Η εντολή αυτή παραθέτει στο όχημα να κινηθεί πίσω για ελάχιστο χρόνο ανεξαρτήτως εμποδίου
- **"rotate right"** -> Η εντολή αυτή παραθέτει το όχημα να κάνει στροφή 180' μοιρών δεξιόστροφα.
- **"rotate left"** -> Η εντολή αυτή παραθέτει το όχημα να κάνει στροφή 180' μοιρών αριστερόστροφα.

6.2 Μελλοντικές επεκτάσεις

Το συγκεκριμένο Project θα μπορούσε να επεκταθεί περισσότερο στο μέλλον και να έχει περισσότερες λειτουργίες για καλύτερη χρήση.

Μία πιθανή μελλοντική επέκταση θα ήταν η εγκατάσταση ανιχνευτή μετάλλων έτσι το όχημα θα μπορούσε να χρησιμοποιηθεί για ανίχνευση διάφορων μετάλλων ή ακόμα και για ανίχνευση ναρκών σε εμπόλεμη ζώνη.

Άλλη μια πιθανή μελλοντική επέκταση θα ήταν η εγκατάσταση μετρητή θερμοκρασίας για μέρη στα οποία δεν μπορεί ο άνθρωπος να πάει λόγω υψηλής η χαμηλής θερμοκρασίας.

Η ενσωμάτωση κάμερας θα ήταν επίσης μια πολύ καλή αναβάθμιση, έτσι ώστε να υπάρχει εικόνα όταν το όχημα χρησιμοποιείται σε απόκρυφα μέρη.

Μια πολύ καλή αναβάθμιση θα ήταν η εισαγωγή λογισμικού για ενημέρωση σχετικά με την κατάσταση του ρομπότ (βλάβες, ακραίες μετρήσεις, χαμηλή μπαταρία, κτλ.)

Τέλος η χρήση WIFI Shield για επικοινωνία μεταξύ χρήστη και ρομπότ μέσω του διαδικτύου για ακόμα πιο απομακρυσμένη χρήση του οχήματος.

Εν κατακλείδι, θα μπορούσε κανείς να σκεφτεί άπειρες επεκτάσεις, οι οποίες θα μπορούσαν να πραγματοποιηθούν πάνω στο συγκεκριμένο Project, αρκεί να υπάρχει όρεξη και θέληση για βελτίωση.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] <https://en.wikipedia.org/wiki/Robotics> [20/4/2023]
- [2] <https://en.wikipedia.org/wiki/Robot> [20/4/2023]
- [3] <http://whatis.techtarget.com/definition/robot-insect-robot-autonomous-robot> [22/4/2023]
- [4] <http://peopleof.oureverydaylife.com/robots-used-everyday-life-6028.html> [24/4/2023]
- [5] <https://www.online-sciences.com/robotics/advantages-and-disadvantages-of-using-robots-in-our-life/> [25/4/2023]
- [6] http://www.webopedia.com/TERM/P/programming_language.html [1/5/2023]
https://en.wikipedia.org/wiki/History_of_programming_languages [1/5/2023]
https://en.wikipedia.org/wiki/Programming_paradigm [1/5/2023]
- [7] <https://www.arduino.cc/en/Guide/Introduction> [5/5/2023]
- [8] http://creativityprojects.blogspot.gr/2013/03/history-of-arduino_4195.html [5/5/2023]

- [9] https://en.wikipedia.org/wiki/Arduin_of_Ivrea [6/5/2023]
- [10] <https://en.wikipedia.org/wiki/Arduino> [7/5/2023]
- [11] <https://learn.sparkfun.com/tutorials/what-is-an-arduino> [10/5/2023]
- [12] <https://docs.arduino.cc/learn/> [10/5/2023]
- <https://docs.arduino.cc/hardware/motor-shield-rev3> [11/5/2023]
- [13] <https://www.farnell.com/datasheets/1682209.pdf> [11/5/2023]
- [14] <https://einhugur.com/blog/index.php/xojo-gpio/hc-sr04-ultrasonic-sensor/> [11/5/2023]
- [15] https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjawoOjm6eBAxWGjqQKHQiABFoQFnoECBgQAQ&url=https%3A%2F%2F5.imimg.com%2Fdata5%2FYT%2FKV%2FMY-1833510%2Farduino-ir-infrared-obstacle-avoidance-sensor-module.pdf&usg=AOvVaw03lxw3_Hvfr8v83Iy3RxX8&opi=89978449 [12/5/2023]
- [16] https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjpvrqcm6eBAxUK3AIHHZpNA4kQFnoECA0QAQ&url=https%3A%2F%2Fcomponents101.com%2Fsites%2Fdefault%2Ffiles%2Fcomponent_datasheet%2FHFC-05%2520Datasheet.pdf&usg=AOvVaw1eLj8QVLw8nVWcIjONSp1A&opi=89978449 [12/5/2023]
- [17] https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwi_5pKRnKeBAxUhiP0HHVcpCDUQFnoECBoQAQ&url=http%3A%2F%2Fwww.ee.ic.ac.uk%2Fpcheung%2Fteaching%2FDE1_EE%2Fstores%2Fsg90_datasheet.pdf&usg=AOvVaw0mGSoDobEfBgrMGA5MSSwn&opi=89978449 [13/5/2023]

EIKONEΣ :

- [18] <https://codingschool.eu/> [25/9/2023]
- [19] <https://www.electroschematics.com/arduino-uno-pinout/> [25/9/2023]
- [20] <https://discountsales.2023factory.ru/category?name=projekt%20eses%20motor%20shield%20l293d%20pro%20arduino%20uno> [25/9/2023]
- [21] <https://osoyoo.com/2018/12/21/ir-obstacle-avoidance-module/> [25/9/2023]
- [22] <https://www.theengineeringprojects.com/2018/10/introduction-to-hc-sr04-ultrasonic-sensor.html> [25/9/2023]
- [23] <https://www.geeksforgeeks.org/all-about-hc-05-bluetooth-module-connection-with-android/> [25/9/2023]
- [24] <https://circuitdigest.com/microcontroller-projects/interfacing-sg90-servo-motor-with-esp32> [25/9/2023]
- [25] <https://www.amazon.com/AEDIKO-Motor-Gearbox-200RPM-Ratio/dp/B09N6NXP4H> [25/9/2023]

Copyright © 2023 – ΜΟΥΣΙΟΣ ΝΙΚΟΛΑΟΣ

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.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Δυτικής Μακεδονίας.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.