ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Τμήμα Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής

Θέμα : Ρομποτικό όχημα αποφυγής εμποδίων και τηλεχειρισμός μέσω Bluetooth.

Σπουδαστής : Ματσούκας Μιλτιάδης , 38002

Επιβλέπων καθηγητής : Δρόσος Χρήστος

Έτος : 2018

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο / Η κάτωθι υπογεγραμμένος / η …Ματσούκας Μιλτιάδης….., του…Ευσταθίου…, με αριθμό μητρώου …38002.. φοιτητής / τρια του μήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής του πανεπιστημίου Δυτικής Αττικής πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασης της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

Επίσης δηλώνω υπεύθυνα ότι έχω παρακολουθήσει το σεμινάριο συγγραφής και εκπόνησης πτυχιακής εργασίας που διοργανώνεται από το Τμήμα Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής κατά το Χειμερινό/Εαρινό Εξάμηνο του Ακ. Έτους ....2016.....

Ο Δηλών Ημερομηνία

ΠΕΡΙΛΗΨΗ

Στόχος αυτής της πτυχιακής εργασίας είναι η σχεδίαση και η κατασκευή εξ ΄ ολοκλήρου ενός ρομποτικού οχήματος το οποίο κινείται στο χώρο αυτόνομα καθώς επίσης και μέσω έτοιμης εφαρμογής android . Η εφαρμογή χρησιμοποιεί το Bluetooth του κινητού και επιτυγχάνει ζεύξη με το ρομποτικό όχημα χρησιμοποιώντας το Bluetooth αισθητήριο HC-05. Το σασί του οχήματος κατασκευάστηκε από plexiglass και χρησιμοποιήθηκε μικροεπεξεργαστής Arduino Uno .Για την αποφυγή των εμποδίων έγινε χρήση αισθητηρίου υπερήχων HC-SR04 τοποθετημένο σε κινητήρα servo με ειδική βάση ώστε να επεκτείνει την γωνία μέτρησης. Η τοποθέτηση του αισθητηρίου πάνω στη βάση δυσχεραίνει την αναγνώριση των εμποδίων που βρίσκονται χαμηλά για το λόγο αυτό μπήκαν διακόπτες επαφής. Η λειτουργία γίνεται με δύο επιλογές, με την πρώτη επιλογή το όχημα κινείται εντελώς αυτόνομα στο χώρο μέσω των αισθητηρίων και με την δεύτερη μέσω της εφαρμογής .

Λέξεις κλειδιά : Android ,Bluetooth, μικροεπεξεργαστής , Arduino , αισθητήρας υπερήχων .

ABSTRACT

The aim of this dissertation is the design and construction of a robot vehicle that moves in the space autonomously as well as through an android application. The application uses mobile's Bluetooth and connects to the robotic vehicle using the Bluetooth transducer HC-05. The chassis of the vehicle was made of plexiglass and Arduino Uno microprocessor. To avoid the obstacles, a HC-SR04 ultrasonic sensor mounted on a servo motor with a special base to extend the measuring angle. Positioning the sensor on the base makes it difficult to identify obstacles that are low for this reason contact switches were used. The operation is made with two options , the first option is that the vehicle moves completely autonomously through the sensors and the second through the application.

Key words : Android, Bluetooth , microprocessor, Arduino , ultrasonic sensor.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον καθηγητή του Α.Ε.Ι. Πειραιά Τ.Τ κ. Δρόσο Χρήστο για την υποστήριξη και την καθοδήγηση που μου παρείχε κατά την διάρκεια της πτυχιακής μου εργασίας. Επίσης θα ήθελα να ευχαριστήσω τους φίλους που με στήριξαν σε όλες τις δυσκολίες που αντιμετώπισα και που ήταν συνοδοιπόροι στην πορεία όλων αυτών των χρόνων .Τέλος δεν θα μπορούσα να παραλείψω τους γονείς μου που με όλη τους την αγάπη και την συμπαράσταση με βοηθάνε να κάνω τα όνειρά μου πραγματικότητα και να πετύχω τους στόχους μου .

Ματσούκας Μιλτιάδης

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ

Στην παρούσα πτυχιακή εργασία θα ασχοληθούμε με την κατασκευή ενός ρομποτικού οχήματος το οποίο χρησιμοποιεί μικροελεγκτή Arduino και θα λειτουργεί υπό δύο συνθήκες , στη πρώτη θα κινείται αυτόνομα στο χώρο χρησιμοποιώντας μόνο τα αισθητήριά του για την αποφυγή εμποδίων και στη δεύτερη θα κινείται με τηλεχειρισμό ο οποίος επιτυγχάνεται μέσω εφαρμογής η οποία επικοινωνεί με το ρομποτικό όχημα μέσω Bluetooth.

Στο πρώτο κεφάλαιο θα ασχοληθούμε με τους μικροελεγκτές δίνοντας πληροφορίες γενικά για τις πλακέτες ανοιχτού κώδικα και στη συνέχεια θα αναλυθεί ο μικροελεγκτής ο οποίος χρησιμοποιήθηκε συγκεκριμένα στην εφαρμογή που δεν είναι άλλος παρά ο Arduino Uno. Θα αναφερθούμε στα τεχνικά χαρακτηριστικά του όπως η μνήμη του , οι είσοδοι και οι έξοδοί του καθώς και τα pins τροφοδοσίας του.

Στο δεύτερο κεφάλαιο θα αναφερθούμε στο προγραμματιστικό περιβάλλον του Arduino και την σύνδεση του με τον υπολογιστή . Θα ασχοληθούμε κυρίως με τη γλώσσα προγραμματισμού, τις κύριες εντολές που μπορούμε να χρησιμοποιήσουμε για να προγραμματίσουμε έναν Arduino . Στο τέλος του κεφαλαίου αυτού θα γίνει μία μικρή αναφορά στις βιβλιοθήκες που μπορούμε να εντάξουμε στα προγράμματα.

Στο τρίτο κεφάλαιο θα γίνει η παρουσίαση του hardware που χρησιμοποιήθηκε για την κατασκευή του ρομποτικού οχήματος. Θα γίνει αναλυτική αναφορά στα εξαρτήματα και στα τεχνικά χαρακτηριστικά αυτών καθώς και τα αισθητήρια της εφαρμογής και μία μικρή αναφορά στον προγραμματισμό των αισθητηρίων και των κινητήρων.

Στο τέταρτο κεφάλαιο θα ασχοληθούμε με την εφαρμογή η οποία χρησιμοποιήθηκε για τον τηλεχειρισμό του οχήματος. Θα υπάρχουν αναλυτικά βήματα για την σύνδεση της εφαρμογής με το όχημα και η αντιστοίχηση των εντολών που στέλνονται από την εφαρμογή για τον προγραμματισμό του Arduino.

Στο πέμπτο κεφάλαιο θα παρουσιάσουμε την κατασκευή του ρομποτικού οχήματος όλα τα βήματα. Αρχικά θα αφοσιωθούμε στην κατασκευή του σασί σχολαστικά διότι είναι κατασκευασμένο και μελετημένο εξ’ ολοκλήρου από εμάς και στη συνέχεια θα προχωρήσουμε στην ένταξη των αισθητηρίων καθώς και όλων των εξαρτημάτων που χρησιμοποιεί το όχημα.

Τέλος στο έκτο κεφάλαιο θα γίνει ανάλυση του κώδικα που είναι προγραμματισμένος ο μικροελεγκτής και θα υπάρχει και ο κώδικας ολόκληρος με σχόλια για τις περισσότερες λειτουργίες του.

ΚΕΦΑΛΑΙΟ 1

Υπολογιστική πλατφόρμα Arduino

1.1Γενικά

Η υπολογιστική πλατφόρμα Arduino πρόκειται για μια πλατφόρμα «ανοιχτού κώδικα» η οποία βασίζεται σε ένα εύκολο και ευπροσάρμοστο hardware αλλά και software το οποίο απευθύνεται σε οποιονδήποτε από άτομα με ελάχιστη εμπειρία στον προγραμματισμό αλλά και σε άτομα που έχουν μεγάλη εμπειρία καθώς μπορεί να χρησιμοποιηθεί και σε αρκετά περίπλοκες εφαρμογές.

Εικόνα 1.1 Διάφοροι τύποι Arduino

Ανάλογα με τις απαιτήσεις της εφαρμογής πρέπει να επιλεγεί ο κατάλληλος τύπος πλατφόρμας Arduino . Οι διαφορές είναι κυρίως στον μικροελεγκτή που χρησιμοποιεί η κάθε πλατφόρμα καθώς και στις εισόδους/εξόδους .

Οι επίσημες πλακέτες Arduino που υπάρχουν είναι οι παρακάτω :

1. Serial Arduino

2. Arduino Extreme

3. Arduino Mini

4. Lily Pad Arduino

5. Arduino Nano

6. Arduino NG

7. Arduino NG Plus

8. Arduino Bluetooth

9. Arduino Diecimila

10. Arduino Duemilanove

11. Arduino Mega

12. Arduino Uno

13. Arduino Mega 2560

14. Arduino Leonardo

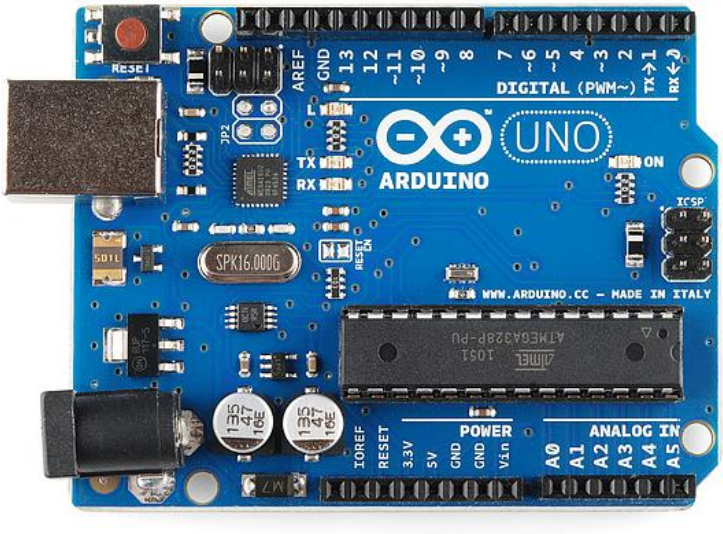
15. Arduino Esplora

16. Arduino Due

Στην εφαρμογή του ρομποτικού οχήματος χρησιμοποιήθηκε η πλατφόρμα Arduino Uno. Ο μικροελεγκτής που χρησιμοποιεί είναι ο ATmega328 ο οποίος είναι κατάλληλος για τέτοιου είδους εφαρμογές.

1.2 Arduino Uno

Η υπολογιστική πλατφόρμα Arduino Uno βασίζεται στον ελεγκτή ATmega328 ο οποίος μέσω του προγραμματισμού διαχειρίζεται ψηφιακές και αναλογικές εισόδους/εξόδους .



Εικόνα 1.2 Υπολογιστική πλατφόρμα Arduino Uno

1.2.1 Μνήμη του Arduino Uno

Η μνήμη του ελεγκτή είναι εσωτερική και είναι τριών τύπων :

1. 2kb SRAM μνήμη στην οποία αποθηκεύονται μεταβλητές οι οποίες χρησιμοποιούνται κατά τη διάρκεια της εκτέλεσης του προγράμματος και χάνονται μόλις γίνει reset ή διακοπεί η παροχή ρεύματος τροφοδοσίας του Arduino.

2. 1Kb EEPROM μνήμη που χρησιμοποιείται για ανάγνωση και εγγραφή δεδομένων από το πρόγραμμα και σε αντίθεση με την SRAM δεν χάνει τα

δεδομένα μόλις γίνει reset ή διακοπεί η παροχή ρεύματος τροφοδοσίας του Arduino.

3. 32Kb FLASH .Τα 2 Kb τα καταλαμβάνει το firmware το οποίο είναι εγκατεστημένο από τον κατασκευαστή και είναι απαραίτητο για το upload του προγράμματος που γράφει ο χρήστης μέσω της θύρας USB.Τα υπόλοιπα 30Kb χρησιμοποιούνται για την αποθήκευση του προγράμματος αυτού καθώς δεν χάνονται τα δεδομένα ούτε σε αυτή την περίπτωση.

1.3 Τροφοδοσία Arduino

Η τροφοδοσία του Arduino γίνεται εξωτερικά από το φις που διαθέτει είτε από μπαταρίες είτε έναν τυπικό μετασχηματιστή που υπάρχει στο εμπόριο διαφορετικά η παροχή του μπορεί να γίνει μέσω USB καλωδίου συνδεδεμένο με τον υπολογιστή. Η τάση που θα δοθεί εξωτερικά θα πρέπει να είναι 7-12V dc .Υπάρχουν έτοιμες θήκες μπαταριών με ακροδέκτη ο οποίος ταιριάζει στο φις του Arduino , αυτό ωφελεί στο να γίνεται η εκάστοτε εφαρμογή αυτόνομη.

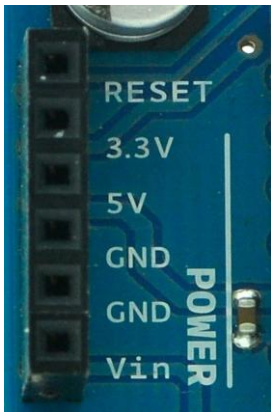
Εικόνα 1.3 Εξωτερική τροφοδοσία με θήκη 4x1,5V

Εικόνα 1.4 Εξωτερική τροφοδοσία με 9V

Ο Arduino διαθέτει 6 pins τα οποία λειτουργούν και αυτά ως τροφοδοσία .

Αναλυτικότερα :

1. Ένα Pin RESET . Γειώνοντας αυτό το Pin επιτυγχάνουμε την επανεκκίνηση του Arduino.
2. Ένα Pin 3.3V το οποίο χρησιμοποιείται συνήθως για εξαρτήματα που λειτουργούν με αυτή την τάση .Η έξοδος αυτή παράγει τάση από τον ελεγκτή και όχι από την εξωτερική τροφοδοσία και η μέγιστη ένταση που μπορεί να παραχθεί είναι 50mA .
3. Ένα Pin 5V που μπορεί να τροφοδοτήσει τα κυκλώματα και τα εξαρτήματα της εφαρμογής . Τα 5V προέρχονται από την εξωτερική τροφοδοσία ή από το καλώδιο USB το οποίο λειτουργεί και αυτό στα 5V.
4. Δύο Pins GND όπου είναι οι γειώσεις .
5. Ένα Pin Vin που έχει δύο δυνατότητες .Η μία μπορεί να λειτουργήσει σαν τροφοδοσία του Arduino σε περίπτωση που δεν μπορεί να χρησιμοποιηθεί το φις . Έτσι τοποθετώντας ένα καλώδιο στο Vin και άλλο ένα στο GND έχουμε πετύχει ότι και με το φις. Η άλλη δυνατότητα είναι ότι εάν έχουμε εξωτερική τροφοδοσία τότε μπορούμε πολύ απλά να τροφοδοτήσουμε τα κυκλώματα παρέχοντάς τους την τάση αυτή που τροφοδοτείται και ο Arduino.



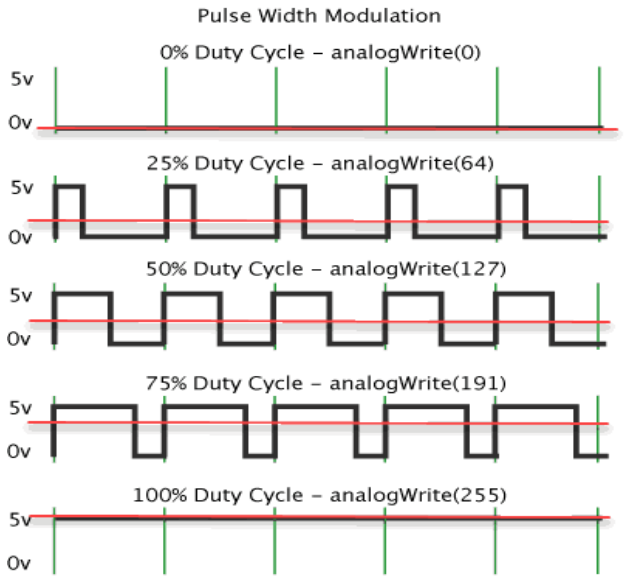
Εικόνα 1.5 Pins τροφοδοσίας Arduino Uno

1.4 Είσοδοι / Έξοδοι

Ο Arduino διαθέτει 14 Pins τα οποία μπορούν να χρησιμοποιηθούν σαν ψηφιακές έξοδοι και ψηφιακές είσοδοι. Η τάση λειτουργίας τους είναι στα 5V και η μέγιστη ένταση που μπορεί να δώσει και να δεχθεί είναι 40mA.Η επιλογή αν τα Pins αυτά θα λειτουργήσουν σαν είσοδοι ή έξοδοι ορίζεται από το πώς θα δηλωθεί στο πρόγραμμα.

1.4.1Έξοδοι

Οι έξοδοι δηλώνονται στο πρόγραμμα και παίρνουν δύο τιμές 0 και 5V. Από τα 14 Pins τα 6 έχουν τη δυνατότητα να λειτουργήσουν σαν PWM έξοδοι (Pulse With Modulation). Το PWM είναι μια αναλογική έξοδος χρησιμοποιώντας ψηφιακά μέσα. Ο ψηφιακός έλεγχος χρησιμοποιείται για να δημιουργήσει έναν τετραγωνικό παλμό , ένα σήμα που εναλλάσσεται από 0 σε 1 αυτό το μοτίβο προσομοιώνει τιμές από 0-5V. Η διάρκεια στην οποία ο παλμός βρίσκεται στην θέση 1 (5V) ονομάζεται πλάτος παλμού, άρα διαμορφώνοντας αυτή τη διάρκεια αλλάζουμε την τιμή στην έξοδο. Οπότε για παράδειγμα εάν συνδέσουμε ένα led στην PWM έξοδο μπορούμε να μεταβάλουμε την φωτεινότητά του από 0 εώς 5V και όχι 0 ή 5V που μας βγάζει μια απλή ψηφιακή έξοδος. Δίνοντάς 127bit θα μεταφραστεί στην έξοδο ως 2,5V. Αυτό χρησιμεύει πολύ και σε εφαρμογές που χρησιμοποιούμε κινητήρες για να ρυθμίσουμε την ταχύτητα την οποία θα κινείται ένα ρομποτικό όχημα ή ένας βραχίονας.



Εικόνα 1.6 Λειτουργία εξόδου PWM

1.4.2 Είσοδοι

Πέρα από τις ψηφιακές εισόδους οι οποίες παίρνουν τιμές μόνο 0 και 1 ο Arduino διαθέτει 6 αναλογικές εισόδους. Αυτές οι είσοδοι χρησιμοποιούνται κυρίως για την λήψη και επεξεργασία δεδομένων από αισθητήρια τα οποία μετατρέπουν τα δεδομένα που λαμβάνουν σε τάση 0-5V. Ο Arduino έχει ενσωματωμένο ADC (Analog to Digital Converter) και από το πρόγραμμα μπορούμε να διαβάσουμε από 0 – 1023 bits.Τα δεδομένα αυτά μπορεί να είναι φώς, υγρασία, θερμοκρασία , απόσταση και άλλα.

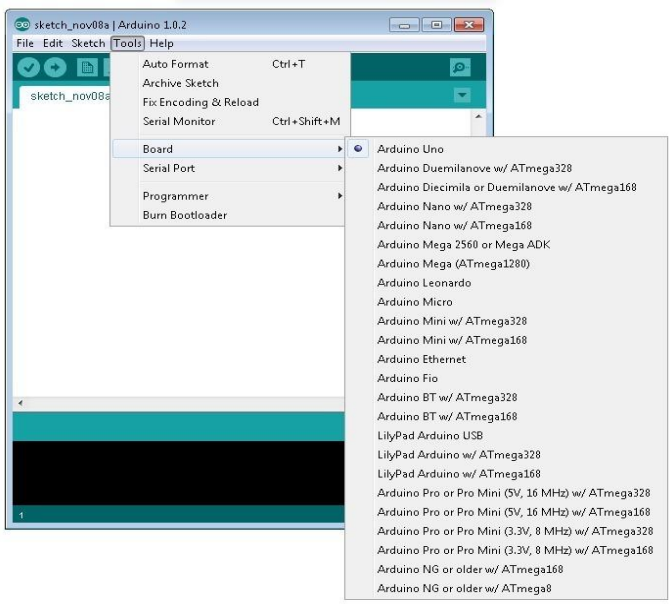
Εικόνα 1.7 Διάφορα αισθητήρια για Arduino

ΚΕΦΑΛΑΙΟ 2

Προγραμματισμός του Arduino

2.1 Σύνδεση Arduino με υπολογιστή

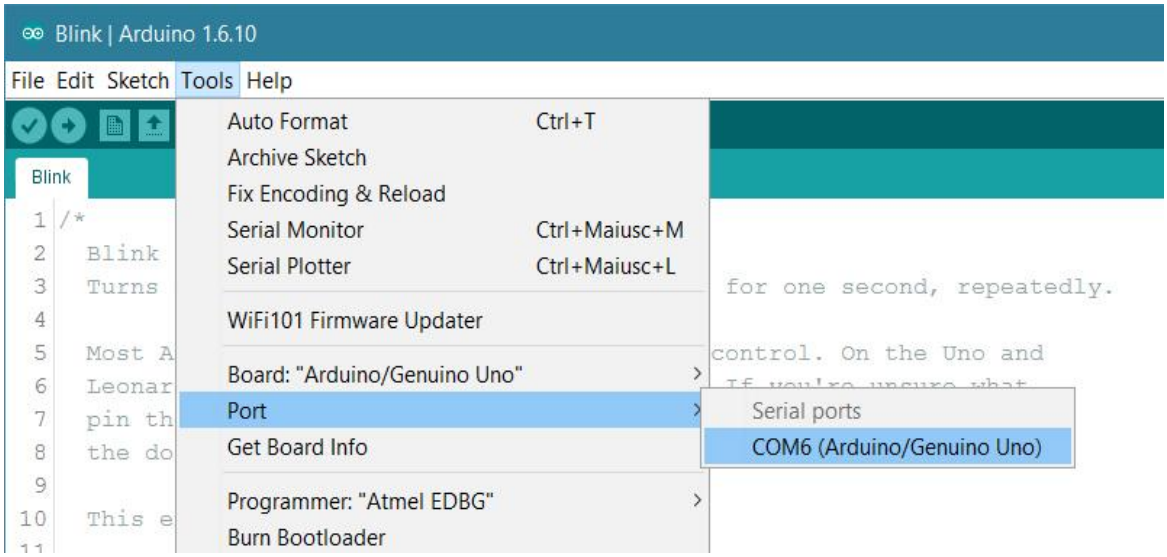
Για τη σύνδεση του Arduino με τον υπολογιστή απαιτείται ένα καλώδιο USB από type A σε type B ίδιο με αυτό των εκτυπωτών .Ο ελεγκτής είναι serial-over USB και αναγνωρίζεται από το λειτουργικό σύστημα ως σειριακή θύρα. Για την διαχείριση και τον προγραμματισμό του Arduino χρειάζεται το προγραμματιστικό περιβάλλον το οποίο κατεβάζουμε δωρεάν από την επίσημη σελίδα του Arduino (https://www.arduino.cc/en/Main/Software). Εκεί βρίσκεται πάντα η τελευταία ενημερωμένη έκδοση στην συγκεκριμένη περίπτωση είναι η 1.8.7 . Ανοίγοντας το πρόγραμμα θα πρέπει να επιτύχουμε την σύνδεση του υπολογιστή με τον Arduino αυτό γίνεται μέσω της εικονικής θύρας COM. Για να το κάνουμε αυτό πρώτα επιλέγουμε τον τύπο του Arduino που χρησιμοποιούμε πηγαίνοντας Tools -> Board -> ArduinoUno



.

Εικόνα 2.1 Επιλογή Arduino Board

Στη συνέχεια επιλέγουμε την σειριακή θύρα επικοινωνίας ακολουθώντας τα εξής βήματα : tools -> port -> COM6 (Aduino/GenuinoUno).

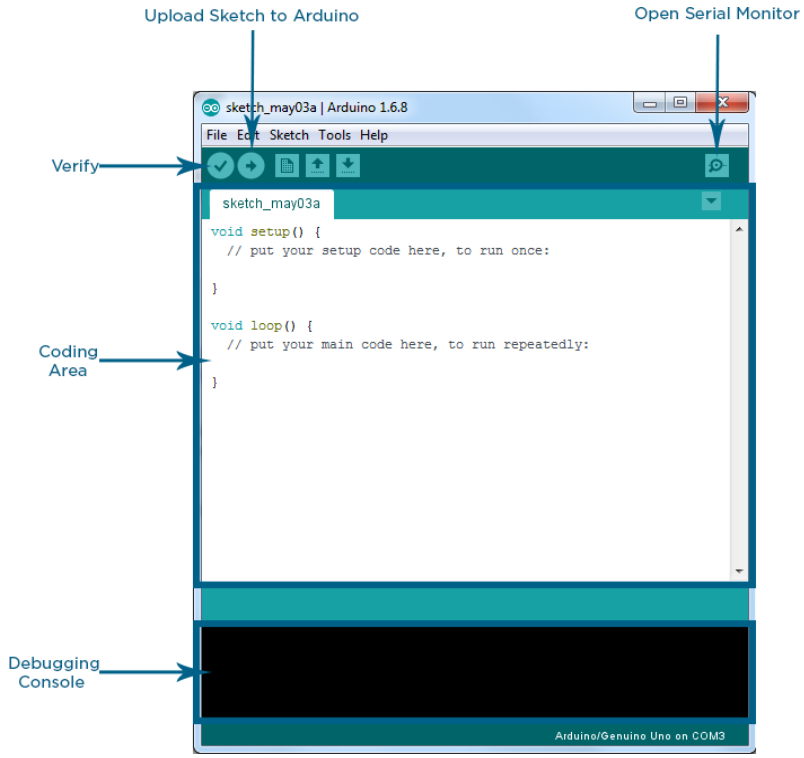


Εικόνα 2.2 Επιλογή σειριακής θύρας

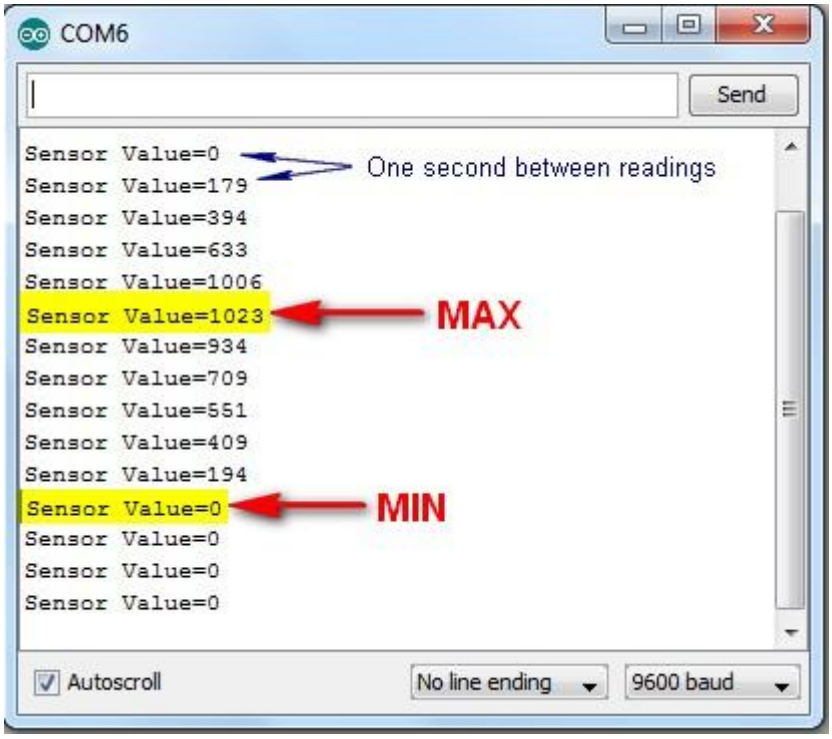
2.2 Προγραμματισμός Arduino

2.2.1 Προγραμματιστικό Περιβάλλον

Το Arduino IDE μας παρέχει ότι χρειαζόμαστε για να διαχειριστούμε τον μικροελεγκτή. Είναι ένα εύκολο και πολύ πρακτικό περιβάλλον και υπάρχουν πάρα πολλές πληροφορίες και tutorials στο site (www.arduino.cc) για να ξεκινήσουμε να μάθουμε να προγραμματίζουμε τον Arduino. Ανοίγοντας το πρόγραμμα μας βάζει στο χώρο όπου μπορούμε να ξεκινήσουμε να γράφουμε το κώδικα . Τα προγράμματα τα ονομάζει αυτόματα sketch .Επίσης δίνεται η δυνατότητα να χρησιμοποιήσουμε τα έτοιμα προγράμματα που είναι αποθηκευμένα . Ανάλογα με την εφαρμογή που έχουμε κληθεί να προγραμματίσουμε καθώς και τα εξαρτήματα που θα συνδέσουμε κάποιες φορές είναι αναγκαία η χρήση βιβλιοθηκών , οι περισσότερες υπάρχουν εγκατεστημένες στο πρόγραμμα το μόνο που θα πρέπει να κάνουμε είναι να τις καλέσουμε στην αρχή του προγράμματος. Τέλος διαθέτει serial monitor το οποίο παρακολουθεί την επικοινωνία της σειριακής θύρας με τον Arduino αυτό μας χρησιμεύει για παράδειγμα να διαβάζουμε τα δεδομένα από ένα αισθητήριο. Τελειώνοντας το κώδικα το μόνο που μένει είναι να κάνουμε το Upload αφού πριν έχουμε κάνει verify το πρόγραμμά μας.



Εικόνα 2.3 Προγραμματιστικό περιβάλλον Arduino



Εικόνα 2.4 Serial Monitor (τιμές από αισθητήριο)

2.2.2Γλώσσα προγραμματισμού Arduino

Η γλώσσα που χρησιμοποιείται για τον προγραμματισμό του Arduino είναι γλώσσα wiring μία παραλλαγή της γλώσσας C/C++ για να προγραμματίζονται ελεγκτές τεχνολογίας AVR όπως και ο Arduino που χρησιμοποιεί ATmega. Ουσιαστικά βασίζεται σε βασικές εντολές και τις συναρτήσεις της C αλλά υπάρχουν και οι εντολές που χρειάζονται για την εκτέλεση του προγράμματος που σχετίζονται με το hardware του Arduino.

Μερικές από τις πιο βασικές εντολές οι οποίες χρησιμοποιήθηκαν και στο ρομποτικό όχημα είναι οι ακόλουθες :

Όρισμα	**Είδος**	**Περιγραφή**
HIGH	Σταθερά	Έχει την τιμή 1 και αντιστοιχεί με λογικό TRUE
LOW	Σταθερά	Έχει την τιμή 0 και αντιστοιχεί με λογικό FALSE
INPUT	Σταθερά	Έχει την τιμή 1 και αντιστοιχεί με λογικό TRUE
OUTPUT	Σταθερά	Έχει την τιμή 0 και αντιστοιχεί με λογικό FALSE
pinMode	Εντολή	Σε συνδυασμό με τις παραπάνω σταθερές ορίζουμε κάποιο pin σαν είσοδο(π.χ pinMode (3,INPUT);), και αντίστοιχα ορίζουμε καποιο pin σαν έξοδο (π.χ pinMode(7,OUTPUT);)
digitalWrite	Εντολή	Θέτει την κατάσταση του pin που θα ορίσουμε σε HIGH ή LOW.
digitalRead	Συνάρτηση	Επιστρέφει την τιμή (0 ή 1, HIGH ή LOW) που δέχεται κάποιο από τα pins εισόδου.
analogWrite	Εντολή	Θέτει το ψηφιακό pin που θα ορίσουμε σε έξοδο με τη μέθοδο PWM.Για παράδειγμα analogWrite(5,127); Που σημαίνει ότι στην έξοδο θα έχουμε περίπου 2,5V.
analogRead	Συνάρτηση	Επιστρέφει έναν ακέραιο αριθμό από 0 εώς 1023 ανάλογα με την τάση που διαβάζει το

		pin
delay	Εντολή	Σταματάει τη ροή του προγράμματος για κάποιο χρόνο που θα ορίσουμε εμείς .Ο χρόνος αυτός δηλώνεται σε ms. Παράδειγμα για 1s θα πρέπει να γράψουμε delay(1000);

Η γλώσσα προγραμματισμού του μικροελεγκτή μας υπόκειται σε δύο βασικές ρουτίνες .Η πρώτη πρόκειται για την setup και η δεύτερη για την loop.

Αναλυτικότερα:

- Η ρουτίνα setup εκτελείται μόνο μια φορά στην αρχή της εκτέλεσης του προγράμματος .Σε αυτή τη ρουτίνα δηλώνουμε και αρχικοποιούμε τις σταθερές όπως και να δηλώσουμε τις βιβλιοθήκες που τυχόν χρειαστούν στην εφαρμογή μας.
- Η ρουτίνα loop κάνει ακριβώς αυτό που λέει και η ονομασία της. Η ρουτίνα αυτή περιέχει τη βασική δομή του προγράμματος και η εκτέλεσή της επαναλαμβάνεται συνεχώς σαν βρόγχος επιτρέποντας στο πρόγραμμα να αλλάζει και να ανταποκρίνεται στις εντολές και στα ερεθίσματα που λαμβάνει από τα αισθητήρια.

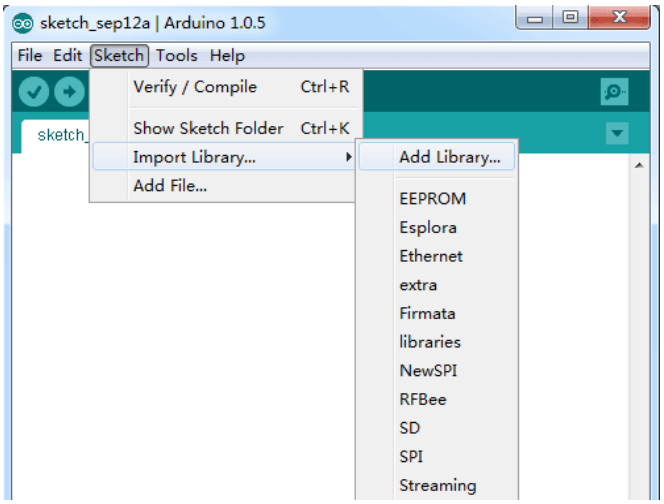
Ένα απλό παράδειγμα το οποίο υπάρχει και έτοιμο μέσα στο προγραμματιστικό περιβάλλον είναι αυτό με το led.

```
int ledPin = 13;

 void setup()

 { pinMode(ledPin, OUTPUT); }  // δηλώνουμε το ledPin ως έξοδο

void loop()

{ digitalWrite(ledPin, HIGH);  // δίνουμε την τιμή 1 ή HIGH

delay(1000);   //ένα δευτερόλεπτο καθυστέρηση

digitalWrite(ledPin, LOW);  // δίνουμε την τιμή 0 ή LOW

delay(1000);   //ένα δευτερόλεπτο καθυστέρηση
```

Παρατηρώντας το πρόγραμμα αυτό μπορούμε να δούμε το led να αναβοσβήνει ανά ένα δευτερόλεπτο .Ο κώδικας ο οποίος αναβοσβήνει το led βρίσκεται στη ρουτίνα loop , οπότε επαναλαμβάνεται συνεχώς .

2.3 Βιβλιοθήκες Arduino

Χρησιμοποιώντας τις βιβλιοθήκες μας βοηθάει πολύ στην επέκταση αλλά και απλοποίηση του προγραμματισμού. Παρέχουν επιπλέον λειτουργικότητα για χρήση διαφόρων εξαρτημάτων αλλά και χρήση δεδομένων . Οι βιβλιοθήκες υπάρχουν ήδη εγκατεστημένες στο προγραμματιστικό περιβάλλον του Arduino καθώς και μεγάλη γκάμα βιβλιοθηκών διατίθεται δωρεάν στο διαδίκτυο έτοιμες προς χρήση.Σε αυτές τις βιβλιοθήκες μπορούμε επίσης να επέμβουμε με το δικό μας τρόπο ώστε να κάνουμε πιο απλό το κυρίως πρόγραμμά μας. Τις βιβλιοθήκες μπορούμε να τις βρούμε ακολουθώντας sketch->import Library .Εκεί προσθέτουμε και τις βιβλιοθήκες που υπάρχουν έτοιμες είτε αυτές που θέλουμε να προσθέσουμε τις οποίες κατεβάζουμε ή δημιουργούμε από την αρχή.



Εικόνα 2.5 Εισαγωγή βιβλιοθήκης

ΚΕΦΑΛΑΙΟ 3

HARDWARE ΚΑΤΑΣΚΕΥΗΣ

3.1 Αισθητήρια και εξαρτήματα ρομποτικού οχήματος

Στην εφαρμογή αυτή χρησιμοποιήθηκαν τα παρακάτω αισθητήρια και εξαρτήματα :

- Αισθητήρας υπερήχων HC-SR04
- Bluetooth module HC-06
- L298N motor driver module (h bridge)
- Κινητήρας servo 180°
- Βάση servo
- 2 διακόπτες επαφής
- 2 κινητήρες DC
- Βύσμα DC power jack
- Θήκη – βάση για μπαταρίες AAx4
- Breadboard και Jumpers

3.1.1 Αισθητήρας Υπερήχων

Στην εφαρμογή αυτή χρησιμοποιήθηκε ο αισθητήρας υπερήχων HC-SR04 ο οποίος συναντάται συχνά σε εφαρμογές τέτοιου τύπου οι οποίες είναι βασισμένες σε μικροεπεξεργαστή Arduino .

Εικόνα 3.1 Αισθητήρας υπερήχων HC-SR04

Ανάλυση του αισθητήρα HC-SR04

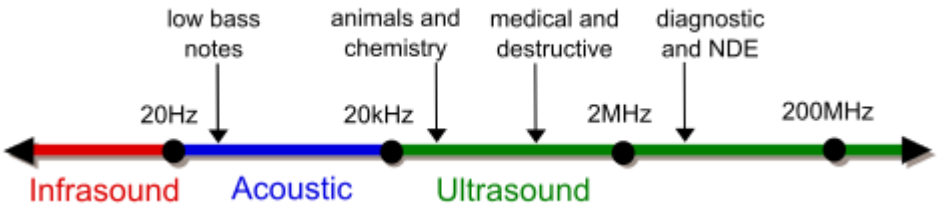
Βλέποντας την παραπάνω εικόνα 13 αριθμώντας τα pins από αριστερά προς τα δεξιά ακολουθεί η επεξήγηση τους :

Αριθμός Pin	Ονομασία	Περιγραφή
1	Vcc	Τροφοδοσία αισθητηρίου +5V
2	Trigger	Το Trigger pin είναι είσοδος. Το pin αυτό πρέπει να χρησιμοποιηθεί για 10μs σε κατάσταση 1 για να προετοιμάσει την μέτρηση στέλνοντας σήμα
3	Echo	Το Echo pin είναι έξοδος .Το pin αυτό γίνεται 1 για διάρκεια ίση με αυτή που χρειάζεται ώστε να γυρίσει το σήμα στον αισθητήρα

4	Gnd	Το pin αυτό συνδέεται στη γείωση

Ορισμός :

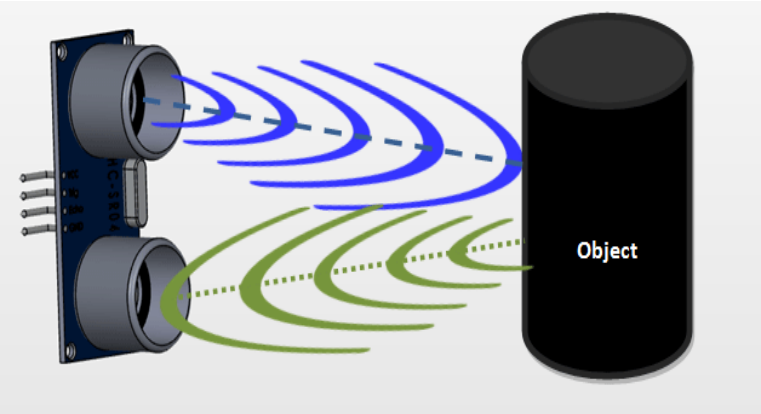
Υπέρηχος θεωρείται το κύμα το οποίο βρίσκεται πάνω από το ακουστό φάσμα που μπορεί να γίνει αντιληπτό από το ανθρώπινο αυτί. Το ανθρώπινο αυτί μπορεί να ακούσει από 20 Hz έως 20 kHz. Ωστόσο σε αντίθεση με τον άνθρωπο υπάρχουν ζώα που είναι ικανά να ακούσουν υπερήχους για παράδειγμα ο σκύλος καθώς επίσης και η νυχτερίδα που χρησιμοποιεί τους υπερήχους για να ‘’βλέπει’’ , όπως και το δελφίνι για να υπολογίζει την απόσταση έτσι και το ρομποτικό όχημά μας χρησιμοποιεί το αισθητήριο των υπερήχων για να υπολογίζει τα αντικείμενα που βρίσκονται στο χώρο έτσι ώστε να αποφύγει την επαφή μαζί τους.



Εικόνα 3.2 Διάγραμμα φάσματος υπερήχων

Τρόπος λειτουργίας :

Ο αισθητήρας υπερήχων είναι ένα όργανο μέτρησης της απόστασης χρησιμοποιώντας υψηλής συχνότητας ραδιοκύματα. Αποτελείται από έναν πομπό και έναν δέκτη , ο πομπός δημιουργεί και στέλνει τα ραδιοκύματα και ο δέκτης λαμβάνει την αντανάκλαση τους .Η απόσταση καθορίζεται από τον χρόνο που έκανε το σήμα να φτάσει στο αντικείμενο από τον αισθητήρα και πίσω.



Εικόνα 3.3 Λειτουργία αισθητήρα υπερήχων

Χαρακτηριστικά αισθητηρίου HC-SR04

- Τάση λειτουργίας : +5V
- Θεωρητικό εύρος μέτρησης απόστασης : 2cm - 80cm
- Συχνότητα λειτουργίας 40Hz
- Ρεύμα λειτουργίας : <15mA
- Ακρίβεια : 3mm
- Καλυπτόμενη γωνία μέτρησης : 15°

Σύνδεση αισθητήρα με Arduino

Στο παρακάτω πίνακα φαίνεται η συνδεσμολογία του αισθητήρα με τον Arduino .Η τροφοδοσία του αισθητήρα γίνεται από τον Arduino και το pin 12 ορίζεται σαν είσοδο του επεξεργαστή και το pin 11 σαν έξοδο του επεξεργαστή.

Arduino pin	**HC SR-04**
+5V	Vcc
Gnd	Gnd
12	Trig
11	Echo

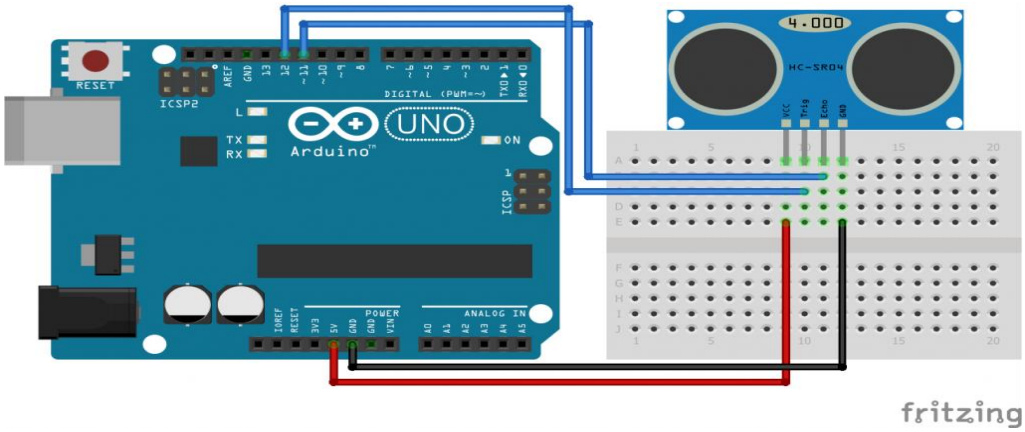
Στον προγραμματισμό τα Trig και Echo του αισθητήρα ονομάζονται trigPin και echoPin .Επίσης χρειάζεται μία μεταβλητή για την διάρκεια του χρόνου που θα λάβει ο αισθητήρας η οποία περιέχει και μία μεταβλητή για την απόσταση.

Στην ρουτίνα setup όπως προαναφέραμε δηλώνονται τα pins ως είσοδος και έξοδος και γίνεται η έναρξη της σειριακής επικοινωνίας.

Στην ρουτίνα loop μηδενίζει το trigPin για 2 μs και στη συνέχεια για να δημιουργηθεί το ηχητικό σήμα θέτουμε το echoPin σε κατάσταση HIGH για 10μs . Με τη λειτουργία pulseIn η μεταβλητή της απόστασης παίρνει την τιμή του χρόνου του ταξιδιού του κύματος.

Η μεταβλητή αυτή έχει δύο παραμέτρους η μία είναι το όνομα το echoPin και η άλλη είναι η τιμή που παίρνει. Όταν έχει την τιμή HIGH περιμένει να πάρει την τιμή HIGH που προκύπτει από το ηχητικό κύμα και τότε ξεκινάει η χρονομέτρηση . Στη συνέχεια περιμένει να πάρει την τιμή LOW . Η χρονομέτρηση σταματάει όταν το ηχητικό κύμα παύει και θα επιστρέψει το μήκος του σήματος σε microseconds. Τέλος η απόσταση υπολογίζεται πολλαπλασιάζοντας το χρόνο επί 0.034 και διαιρώντας δια δύο . Το αποτέλεσμα θα εμφανίζεται ανοίγοντας την σειριακή οθόνη.

```
1.  // ορισμός των pin
2.  const int trigPin = 12;
3.  const int echoPin= 11;
4.
5.  // καθορισμός μεταβλητών
6.
7.  long duration;
8.  int distance;
9.
10. void setup() {
11. pinMode(trigPin, OUTPUT); // Ορισμός του trigPin ως έξοδος
12. pinMode(echoPin, INPUT); // Ορισμός  echoPin ως είσοδος
13. Serial.begin(9600); //Εναρξη σειριακής επικοινωνίας
14.
15. }
16.
17. void loop()
18.
19. {
20. // Μηδενισμός του trigPin
21.
22. digitalWrite(trigPin, LOW);
23. delayMicroseconds(2);
24.
25. // Θέτουμε το trigPin σε κατάσταση HIGH για 10 micro seconds
26.
27. digitalWrite(trigPin, HIGH);
28. delayMicroseconds(10);
29. digitalWrite(trigPin, LOW);
30.
31. // Διαβάζουμε το echoPin, και επιστρέφουμε το χρόνο που κάνει το μικρόκυμα σε
    microseconds
32.
33. duration = pulseIn(echoPin, HIGH);
34.
35. //Υπολογισμός της απόστασης
36.
37. distance= duration*0.034/2;
38.
39. // Εμφάνιση της απόστασης στην σειριακή οθόνη
40.
41. Serial.print("Distance: ");
42. Serial.println(distance);
43.
44.
45.  }
46.
```



Εικόνα 3.4 Συνδεσμολογία αισθητήρα sr 04 σε Arduino

Στην εφαρμογή αυτή η καλυπτόμενη γωνία μέτρησης δεν επαρκεί ώστε να καλύψει τις ανάγκες του ρομποτικού οχήματος χρησιμοποιήθηκε κινητήρας servo στον οποίο τοποθετήθηκε βάση για τον αισθητήρα υπερήχων ο συνδυασμός αυτός επεκτείνει τη καλυπτόμενη γωνία μέτρησης από 15° στις 180° περίπου και θα αναλυθεί παρακάτω.

3.1.2 Bluetooth Module

Η τεχνολογία του Bluetooth έχει μπει για καλά στη ζωή μας εδώ και πολλά χρόνια. Είναι μία μορφή ασύρματης τηλεπικοινωνίας μεταξύ συσκευών που βρίσκονται σε μικρή απόσταση, η οποία μπορεί να μεταδώσει πληροφορίες μέσω μικροκυμάτων . Το Bluetooth στην ουσία στην εφαρμογή μας είναι ένα πρωτόκολλο επικοινωνίας μεταξύ του κινητού και του ρομποτικού οχήματος. Στην εφαρμογή χρησιμοποιήθηκε το Bluetooth module HC-06.

Εικόνα 3.5 Bluetooth module HC-06

Η αρίθμηση των pins ξεκινώντας από αριστερά προς τα δεξιά και η λειτουργία τους αναλύεται παρακάτω :

Pin	**Όνομα**	**Λειτουργία**
1	STATE	Κατάσταση σύνδεσης
2	RXD	Receive Pin
3	TXD	Transmit Pin
4	GND	Γείωση
5	+5V	Τροφοδοσία
6	KEY	HIGH=προγραμματισμός LOW= λειτουργία

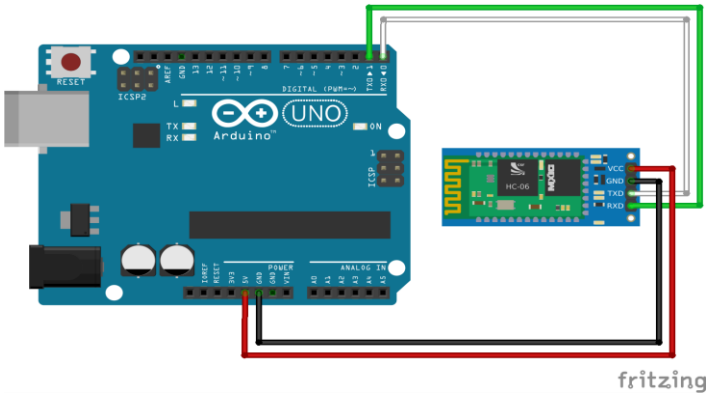
Χαρακτηριστικά Bluetooth Module HC-06

- Πρωτόκολλο Buetooth : Bluetooth 2.0+ EDR standard
- Συχνότητα λειτουργίας : 2.4GHz
- Ισχύς μετάδοσης : ≤ 4dBm
- Ευαισθησία : ≤-84dBm στο 0.1% Bit Error Rate

- Ταχύτητα μετάδοσης : 2.1Mbps(Max)/160 kbps(Ασύγχρονη) ; 1Mbps/1Mbps(Σύγχρονη)
- Τάση λειτουργίας : +3.3 VDC 50mA
- Θερμοκρασία λειτουργίας : -20 to 55℃

Συνδεσμολογία Bluetooth Module με Arduino

Για να επιτύχουμε την επικοινωνία μεταξύ Bluetooth με τον Arduino πρέπει να συνδέσουμε το pin Rxd με το Txd του Arduino και το Pin Txd με το Rxd του Arduino .Σαφώς θα πρέπει να τροφοδοτήσουμε το Bluetooth με 3,3V ή 5V , αυτή την παροχή μπορούμε να την πάρουμε κατευθείαν από τον Arduino καθώς επίσης και την γείωση στο Gnd του Arduino. Εδώ πρέπει να τονίσουμε ότι κάθε φορά που θα χρειάζεται να κάνουμε upload το πρόγραμμα μας θα πρέπει να αποσυνδέουμε τα pins Rxd και Txd , μόλις το πρόγραμμα περαστεί τότε μπορούμε να επανασυνδέσουμε τα pins.



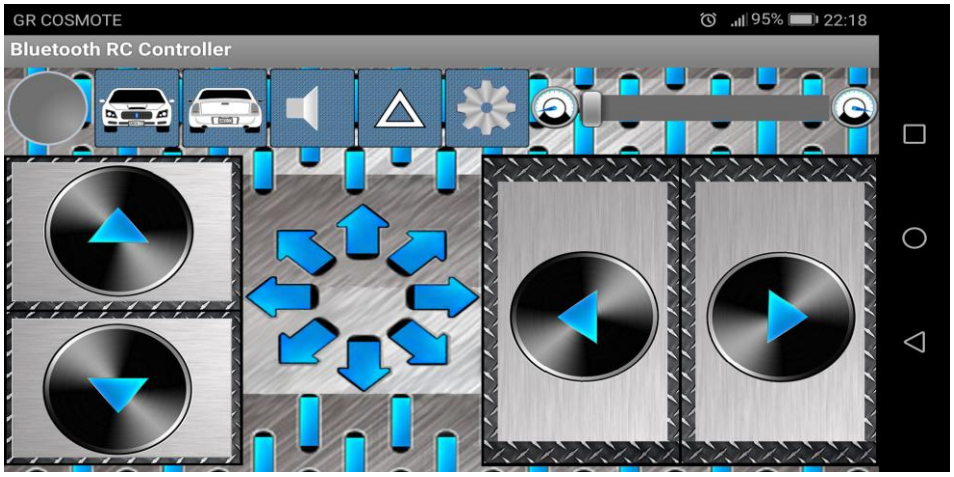
Εικόνα 3.6 Σύνδεση Bluetooth με Aduino

Επικοινωνία Bluetooth με Arduino

Ξεκινάμε συνδέοντας τον Arduino με τον υπολογιστή μέσω usb καλωδίου, ανοίγοντας το ide στον υπολογιστή επιλέγουμε να ανοίξουμε την σειριακή οθόνη .Στη συνέχεια ρυθμίζουμε τον σωστό ρυθμό μετάδοσης στα 9600 τον οποίο χρησιμοποιεί το Bluetooth module ,αυτή είναι και η ποιο συνηθισμένη τιμή .Μετά πληκτρολογούμε AT και αν μας δοθεί απάντηση OK είμαστε έτοιμοι να προγραμματίσουμε το Bluetooth. Δεν είναι απαραίτητο αλλά εάν θελήσουμε να αλλάξουμε όνομα στο

module πρέπει να πληκτρολογήσουμε AT+NAME βάζοντας το νέο όνομα τότε το Bluetooth μας απαντάει OKsetname . Μπορούμε να αλλάξουμε και το ρυθμό μετάδοσης ακόμα και το pin. Δεν έκρινα απαραίτητο να αλλάξω τίποτα από όλα αυτά εάν θα άλλαζα κάτι ίσως να ήταν το όνομα του Bluetooth έτσι ώστε να βρίσκεται ποιο εύκολα η συσκευή στην αναζήτηση από το κινητό η υπολογιστή.

Κατεβάζοντας την εφαρμογή Bluetooth RC Controller από το App Store μπορούμε να δοκιμάσουμε την ζεύξη μεταξύ του Bluetooth και του κινητού. Η εφαρμογή έχει το περιβάλλον που φαίνεται στην εικόνα παρακάτω :

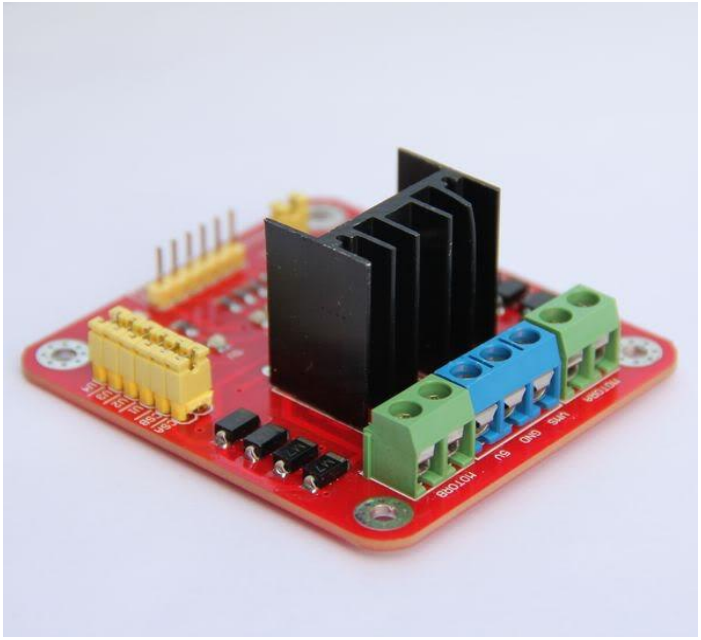


Εικόνα 3.7 Περιβάλλον εφαρμογής

Κάθε ‘’μπουτόν ‘’ που πατάμε στην εφαρμογή αυτή το μεταφράζει και το στέλνει στο Bluetooth με μορφή χαρακτήρων. Παίρνοντας τους χαρακτήρες αυτούς μπορούμε να δοκιμάσουμε την επικοινωνία των συσκευών , αυτό επιτυγχάνεται ανοίγοντας το serial monitor και βλέποντας αν οι εντολές που δίνουμε από το κινητό έρχονται σωστά στον Arduino.

3.1.3 Πλακέτα «γέφυρας Η» L298N

Η «γέφυρα Η» είναι μία πλακέτα η οποία χρησιμοποιείται για την οδήγηση των κινητήρων .Για την οδήγηση των κινητήρων συνεχούς ρεύματος αυτό που έχουμε να κάνουμε είναι να ελέγξουμε την πολικότητα τους, αυτό είναι στην ουσία που κάνει η «γέφυρα Η» η οποία είναι κατασκευασμένη χρησιμοποιώντας τρανζίστορ.

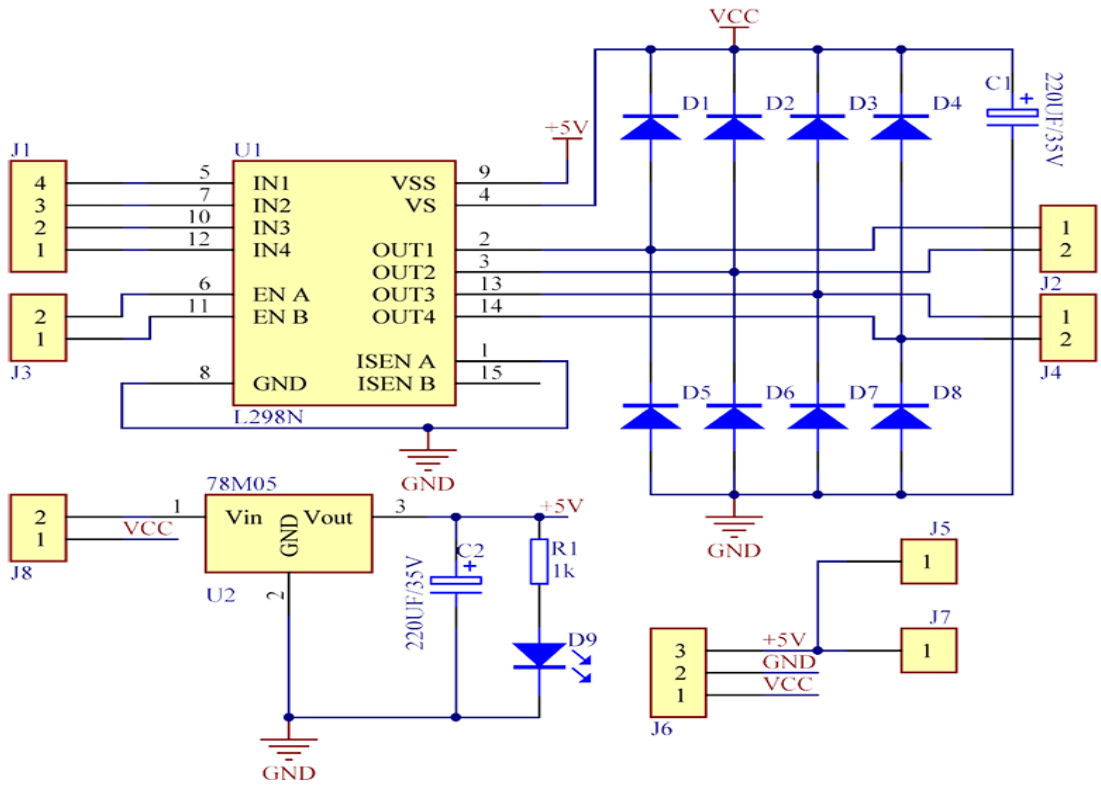


Εικόνα 3.8 L298N «H Bridge»

Η πλακέτα αυτή χρησιμοποιεί το ολοκληρωμένο κύκλωμα L298N το οποίο ανήκει στην οικογένεια των IC (integrated circuit) που έχουν το ίδιο όνομα L298 με τη διαφορά στα ρεύματα τα οποία μπορούν να διαχειριστούν. Το ολοκληρωμένο που χρησιμοποιεί η πλακέτα αυτή μπορεί να διαχειριστεί ρεύματα ύψους 2A και τάσεις εώς 46V dc πράγμα που κάνει την πλακέτα αυτή κατάλληλη για τους περισσότερους κινητήρες dc.

Το L298N εμπεριέχει δύο «γέφυρες Η» αυτό σημαίνει ότι μπορεί να χειριστεί δύο κινητήρες , αυτό σημαίνει ότι μπορεί να χρησιμοποιηθεί για ρομποτικά οχήματα που χρησιμοποιούν δύο κινητήρες για την κίνησή τους .

Σχηματικό διάγραμμα L298N



Εικόνα 3.9 Σχηματικό διάγραμμα πλακέτας

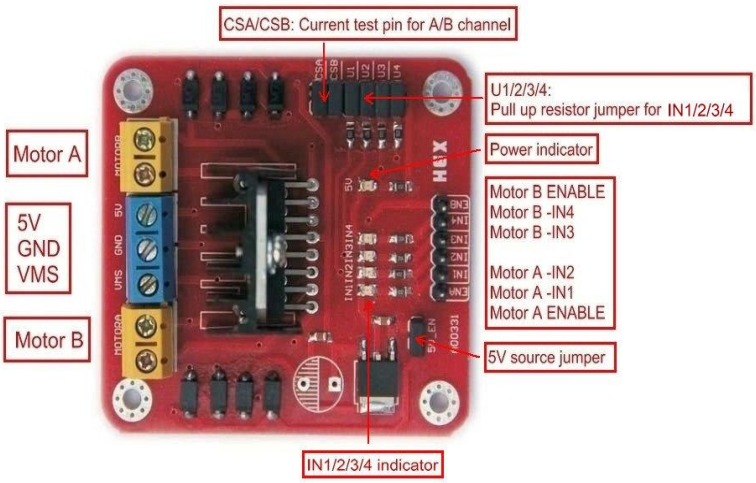
Τεχνικά χαρακτηριστικά L298N

- **Driver** : L298N
- Τροφοδοσία : +5V~+46V
- Ρεύμα εξόδου : 2A
- Λογική τάση εξόδου : +5~+7V (εσωτερική τάση +5V)
- Λογικό ρεύμα : 0~36mA
- Τάση ελέγχου : Low -0.3V~1.5V, high: 2.3V~Vss

- Τάση σήματος Enable : Low -0.3V~1.5V, high: 2.3V~Vss
- Μέγιστη ισχύς : 25W
- Θερμοκρασία λειτουργίας : -25 °C~+130 °C
- Διαστάσεις : 60mm*54mm

Ανάλυση πλακέτας L298N

Στην παρακάτω εικόνα βλέπουμε αναλυτικά την πλακέτα και τα εξαρτήματα που διαθέτει.



Εικόνα 3.10 Ανάλυση Pins της πλακέτας L298N

Η πλακέτα όπως προαναφέραμε χρησιμοποιεί το L298N μαζί με μία ψύκτρα για την απαγωγή της θερμοκρασίας.

Αναλυτικότερα :

- **5V** : Αυτή η τάση είναι προαιρετική και όπως είναι προφανές είναι στα 5V dc
- **GND** : Γείωση
- **VMS** : Σε αυτό το pin παρέχουμε εξωτερική τροφοδοσία αυτή που θα παρέχουμε στους κινητήρες
- **Motor A , Motor B** : Εκεί συνδέουμε τα άκρα των κινητήρων

- **MotorA Enable** : Είσοδος τροφοδοσίας κινητήρα Α
- **MotorA In1**: Είσοδος 1 κινητήρα Α.
- **MotorA In2** : Είσοδος 2 κινητήρα Α
- **MotorB Enable** : Είσοδος τροφοδοσίας κινητήρα Β
- **MotorB In3**: Είσοδος 3 κινητήρα Β
- **MotorB In4**: Είσοδος 4 κινητήρα Β
- **Power Indicator Led** : Led ένδειξης τροφοδοσίας
- **IN1,2,3,4 indicator leds** : Led ένδειξης εισόδου σε κινητήρα
- **CSA ,CSB** : Λειτουργία ανίχνευσης ρεύματος κινητήρα Α και Β. Μεταξύ αυτού του Pin και μιας γείωσης συνδέεται μία αντίσταση η οποία είναι υπεύθυνη για την έλεγχο του ρεύματος φορτίου. Σε αυτό το σημείο είναι τοποθετημένος ένας βραχυκυκλωτήρας ο οποίος στις περισσότερες εφαρμογές δεν χρειάζεται να αφαιρεθεί.
- **U1,U2,U3,U4** : Αντιστάσεις ανύψωσης εισόδου 1,2,3 και 4 .Αυτή ενεργοποιεί μία αντίσταση 10k για την είσοδο.
- **5V source jumper** : Ο βραχυκυκλωτήρας αυτός είναι τοποθετημένος by default .Αυτό επιτρέπει στα λογικά κυκλώματα 7805 να πάρουν λογική ισχύ από την τάση που θα τροφοδοτήσουμε τους κινητήρες για αυτό το λόγο το pin 5V είναι προαιρετικό . Πράγμα που σημαίνει ότι αν θέλουμε να αφαιρέσουμε το βραχυκυκλωτήρα θα χρειαστεί να τροφοδοτήσουμε το pin με 5V.

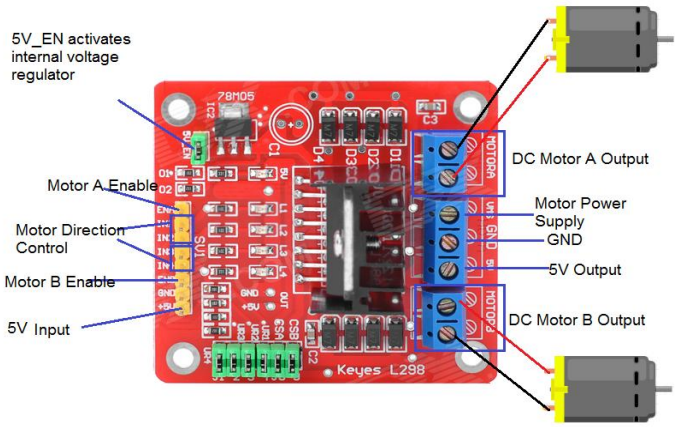
  Στην πρώτη περίπτωση θα λόγω της πτώσης τάσης περίπου 1,4V που δημιουργείται στα τρανζίστορ που είναι εφοδιασμένη η «γέφυρα Η».

Παράδειγμα γέφυρας με κινητήρες

Για μία εφαρμογή με έναν κινητήρα τον οποίο θέλουμε να ελέγξουμε θα πρέπει να ακολουθήσουμε τα εξής βήματα :

Συνδέουμε τα καλώδια του στα τερματικά για τον κινητήρα Α . Για να ενεργοποιήσουμε τις εισόδους θα πρέπει να δώσουμε 5V στο ΕΝΑ στη συνέχεια εάν θέλουμε να κάνει κίνηση δεξιά πρέπει να τροφοδοτήσουμε το IN1 με 5V και το IN2 στη γείωση και αντίστοιχα για κίνηση αριστερά θα πρέπει να κάνουμε το ανάποδο .Για να σταματήσει ο κινητήρας οποιαδήποτε κίνηση θα πρέπει να γειώσουμε και τις δύο εισόδους IN1 και IN2.

Για εφαρμογή με δύο κινητήρες οι συνδέσεις δεν διαφέρουν ιδιαίτερα , απλά προσθέτουμε έναν ακόμα κινητήρα στα τερματικά για τον κινητήρα Β και συνδέουμε το ENA στα 5V.Οι κινήσεις δεν διαφέρουν ότι ισχύει για τον ένα κινητήρα ισχύει και για τους δύο. Το μόνο που πρέπει να προσέξουμε είναι ότι εάν οι κινητήρες τοποθετηθούν σε ρομποτικό όχημα θα πρέπει να παρατηρήσουμε τους κινητήρες διότι εφόσον έχουν τοποθετηθεί ο ένας απέναντι από τον άλλο στην ουσία ο ένας είναι ανάποδα. Άρα αυτό σημαίνει ότι για κίνηση εμπρός ο ένας κινητήρας θα δουλεύει δεξιόστροφα και ο άλλος αριστερόστροφα.



Εικόνα 3.11 Σύνδεση δύο κινητήρων σε L298N

Σύνδεση πλακέτας L298N με Arduino

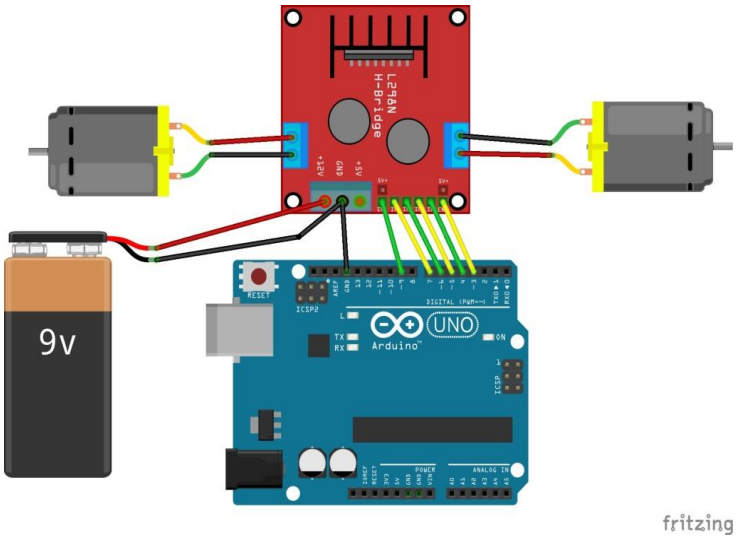
Συνδέοντας την πλακέτα οδήγησης κινητήρων με τον Arduino μας δίνεται η δυνατότητα πλήρης ελέγχου κινήσεων αλλά και της ταχύτητας τους .Αυτό γίνεται συνδέοντας όλα τα inputs των κινητήρων της πλακέτας στις εξόδους του Arduino, άρα αντί να χρειάζεται να συνδέουμε κάθε φορά ανάλογα με την κίνηση που θελήσουμε να κάνει ο κινητήρας σε 5V και γείωση προγραμματίζουμε τις εξόδους του Arduino να κάνει αυτή την δουλειά. Συνεπώς έχουμε τον πλήρη έλεγχο των κατευθύνσεων των κινητήρων.

Όπως προαναφέραμε σε προηγούμενο κεφάλαιο ο Arduino διαθέτει έξι εξόδους με την δυνατότητα λειτουργίας PWM. Αξιοποιώντας αυτή την δυνατότητα έχουμε και τον έλεγχο της ταχύτητας των κινητήρων άρα και του ρομποτικού οχήματός μας.

Παρακάτω ακολουθεί ένα παράδειγμα σύνδεσης της πλακέτας με τον Arduino :

Πλακέτα L298N	**Arduino Uno** **Digital Output**
IN1	7
IN2	6
IN3	5
IN4	4
ENA	9
ENB	3

Να σημειωθεί ότι τα σήματα ENA και ENB έχουν οδηγηθεί σε εξόδους PWM για τον έλεγχο της ταχύτητάς τους ,αυτό γίνεται όταν στη πλακέτα έχει δοθεί εξωτερική τροφοδοσία για την οδήγηση των κινητήρων.



Εικόνα 3.12 Παράδειγμα δύο κινητήρων με Arduino

Παραδείγματα κινήσεων με προγραμματισμό

Εάν υποθέσουμε ότι το ρομποτικό όχημα μας διαθέτει τους παραπάνω κινητήρες όπως απεικονίζονται στην παραπάνω εικόνα21 ακολουθεί πρόγραμμα για διάφορες κινήσεις .

Κίνηση εμπρός και πίσω

```
// Αριστερός κινητήρας

int enA=9;

int in1 = 7;

int in2 = 6;


// Δεξιός κινητήρας

Int enB = 3;

Int in3 = 5 ;

Int in4 = 4 ;
```

```
void  setup () {

//Δήλωση των  Pins των κινητήρων ως έξοδοι
pinMode (enA , OUTPUT);
pinMode (in1 , OUTPUT);
pinMode (in2 , OUTPUT);
pinMode (enB , OUTPUT);
pinMode (in3 , OUTPUT);
pinMode (in4 , OUTPUT);
}
void loop ()
{
// Κίνηση Εμπρός
digitalWrite(in1,HIGH);
digitalWrite(in2,LOW);
digitalWrite(in3,LOW);
digitalWrite(in4,HIGH);

//Θέτουμε ταχύτητα 180
analogWrite(enA,180);
analogWrite(enB,180);

//Καθυστέρηση ενός δευτερολέπτου
delay(3000);
//Κίνηση πίσω
digitalWrite(in1,LOW);
```

```
digitalWrite(in2,HIGH);
digitalWrite(in3,HIGH);
digitalWrite(in4,LOW);

//Θέτουμε ταχύτητα 180
analogWrite(enA,180);
analogWrite(enB,180);
delay(3000);

//Ακινησία οχήματος
digitalWrite(in1,LOW);
digitalWrite(in2,LOW);
digitalWrite(in3,LOW);
digitalWrite(in4,LOW);
}
```

Σε αυτό το πρόγραμμα μπορούμε να διακρίνουμε ξεκινώντας ότι δηλώνουμε τις εξόδους που αντιστοιχούν στους κινητήρες. Επίσης χρησιμοποιούμε τις εξόδους PWM για να ελέγξουμε την ταχύτητα τους. Στη συνέχεια το όχημα θα κινηθεί εμπρός για τρία δευτερόλεπτα μετά θα αλλάξει κατεύθυνση και θα κινηθεί προς τα πίσω για άλλα τρία δευτερόλεπτα και θα σταματήσει. Η ταχύτητα που θα κινείται θα είναι 180.

Κίνηση δεξιά και αριστερά

Σε αυτές τις κινήσεις η ρουτίνα του setup δεν θα αλλάξει καθόλου το μόνο που θα αλλάξει είναι η ρουτίνα του loop και θα γίνει ως εξής :

```
void loop()
```

```
{
// Κίνηση δεξία
digitalWrite(in1,HIGH);
digitalWrite(in2,LOW);
digitalWrite(in3,LOW);
digitalWrite(in4,LOW);

//Θέτουμε ταχύτητα 180
analogWrite(enA,180);
analogWrite(enB,180);

//Καθυστέρηση τριών δευτερολέπτων
delay(3000);
//Κίνηση αριστερά
digitalWrite(in1,LOW);
digitalWrite(in2,LOW);
digitalWrite(in3,LOW);
digitalWrite(in4,HIGH);

//Θέτουμε ταχύτητα 180
analogWrite(enA,180);
analogWrite(enB,180);
//Καθυστέρηση τριών δευτερολέπτων
delay(3000);

//Ακινησία οχήματος
digitalWrite(in1,LOW);
```

```
digitalWrite(in2,LOW);

digitalWrite(in3,LOW);

digitalWrite(in4,LOW);

}
```

Το όχημα τώρα θα κινηθεί για τρία δευτερόλεπτα με κίνηση δεξιά θα περιμένει τρία δευτερόλεπτα και μετά θα κινηθεί αριστερά και στη συνέχεια θα σταματήσει. Στα οχήματα αυτά που χρησιμοποιούν δύο κινητήρες υπάρχουν δύο τρόποι για κινήσεις δεξιά και αριστερά. Ο πρώτος είναι ακινητοποιώντας τελείως τον ένα κινητήρα από τους δύο αναλόγως την κίνηση που επιθυμούμε . Με αυτό τον τρόπο το όχημα κάνει αργή στροφή. Ο δεύτερος τρόπος είναι να αναλόγως την κίνηση που θέλουμε αντί να σταματήσουμε τον ένα κινητήρα να τον κάνουμε να κινείται ανάποδα είτε με λιγότερη ταχύτητα είτε με την ίδια , αυτό κρίνεται από τις απαιτήσεις της εφαρμογής και όπως κρίνει ο χρήστης. Το παράδειγμα ακολουθεί παρακάτω:

```
void loop()

{

// Κίνηση δεξία

digitalWrite(in1,HIGH);

digitalWrite(in2,LOW);  // Αντίστροφη κίνηση των δύο κινητήρων

digitalWrite(in3,LOW);

digitalWrite(in4,HIGH);

analogWrite(enA,180); // Ταχύτητα αριστερού κινητήρα 180

analogWrite(enB,100); // Ταχύτητα δεξιού κινητήρα 100

}
```

3.1.4 Κινητήρες DC

Γενικά

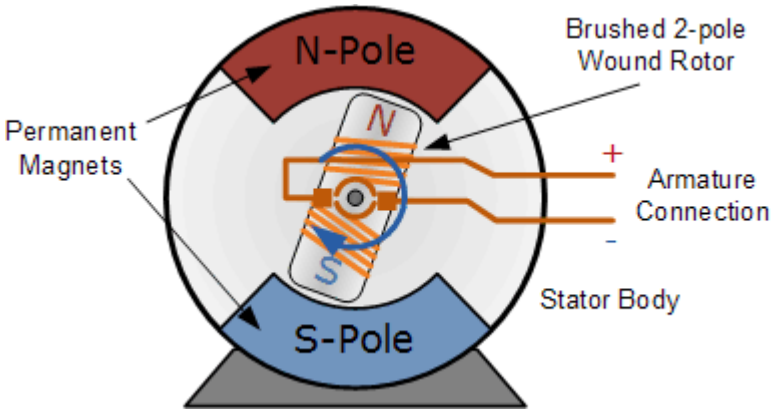
Οι κινητήρες DC (Direct Current) είναι ηλεκτρομηχανικές συσκευές οι οποίες για να μετατρέψουν την ηλεκτρική ενέργεια σε μηχανική περιστροφική ενέργεια χρησιμοποιούν την αλληλεπίδραση των μαγνητικών πεδίων και αγωγών.

Οι κινητήρες συνεχούς ρεύματος είναι ευρέως διαδεδομένοι για την σχετικά εύκολη χρήση και τον απλό έλεγχο τους .Χρησιμοποιείται σε εφαρμογές όπως ο έλεγχος της ταχύτητας και έλεγχος θέσης .

Λειτουργία κινητήρων συνεχούς ρεύματος

Στους κινητήρες συνεχούς ρεύματος ο στάτης είναι ένας σταθερός μαγνήτης ο οποίος δημιουργεί μόνιμα ένα μαγνητικό πεδίο. Ο οπλισμός είναι ένα απλό πηνίο και αποτελεί το κινούμενο μέρος του κινητήρα. Γύρω από τον άξονα του κινητήρα υπάρχει ένας δακτύλιος ο οποίος χρησιμοποιείται ώστε να δίνει παροχή συνεχούς ρεύματος στον οπλισμό. Ο δακτύλιος αποτελείται από δύο τεμάχια και ονομάζονται ‘’Δακτύλιοι μεταγωγού’’ και συνδέονται στα άκρα του πηνίου οπλισμού. Το ρεύμα για να φτάσει στους δακτυλίους του συλλέκτη χρησιμοποιεί δύο βούρτσες οι οποίες τρίβονται σε αντίθετη φορά από τους δακτυλίους. Παρέχοντας συνεχές ρεύμα στους δακτυλίους του συλλέκτη αυτό διαπερνά το πηνίο οπλισμού και παράγει μαγνητικό πεδίο το οποίο έλκεται από το μαγνήτη του στάτη προκαλώντας έτσι την περιστροφή τον άξονα του κινητήρα. Ο κινητήρας περιστρέφεται μέχρι να φτάσει στη διακλάδωση των δύο μισών του μεταγωγού εκεί όπου οι βούρτσες του κινητήρα συναντάνε το άλλο μισό των δακτυλίων και προκαλούν την αναστροφή της πολικότητας του πηνίου. Έχοντας κάνει αυτή τη λειτουργία ο άξονας έχει περιστραφεί 180° και για να συνεχίσει την κίνηση ο άξονας θα πρέπει να αναστραφεί πάλι η πολικότητα στο μαγνητικό πεδίο.

Οι ποιο συνηθισμένοι κινητήρες συνεχούς ρεύματος για χρήση μικρών εφαρμογών χρησιμοποιούν τάση από 6 εώς και 12 V . Για τον έλεγχο της ταχύτητας ενός κινητήρα χρειάζεται μόνο η αυξομείωση της τάσης τροφοδοσίας του. Αυτό βέβαια έχει το μειονέκτημα της μείωσης της ροπής του κινητήρα όσο μειώνεται η τάση άρα και η ταχύτητά του , επίσης αν η τάση τροφοδοσίας πέσει κάτω από ένα όριο ο κινητήρας θα σταματήσει να περιστρέφεται .Για την αλλαγή φοράς περιστροφής ενός κινητήρα συνεχούς ρεύματος αρκεί μόνο να αλλάξει η πολικότητα του ρεύματος που ο κινητήρας τροφοδοτείται.



Εικόνα 3.13 Εσωτερικό κινητήρα συνεχούς ρεύματος

Στην εφαρμογή του ρομποτικού οχήματος χρησιμοποιήθηκαν κινητήρες συνεχούς ρεύματος τοποθετημένος σε κιβώτιο με γρανάζια. Τα γρανάζια που χρησιμοποιεί το kit αυτό ωφελούν στην χρήση του κινητήρα με χαμηλή τάση τροφοδοσίας , με λίγα λόγια αυξάνει την ροπή του κινητήρα .

Εικόνα 3.14 Κινητήρας με εσωτερικό κιβώτιο γραναζιών

Τεχνικά χαρακτηριστικά κινητήρα

- Τάση λειτουργίας : 3 - 6 Vdc
- Ταχύτητα περιστροφής χωρίς φορτίο : 130 rpm
- Ρεύμα χωρίς φορτίο : 200mA
- Ροπή : 0.8 kg.cm
- Σχέση μετάδοσης : 48:1

3.1.5 Κινητήρας Servo

Ο κινητήρας Servo χρησιμοποιήθηκε στην εφαρμογή αυτή ως σκοπό την επέκταση της γωνίας μέτρησης του αισθητηρίου υπερήχων. Επειδή επιλέχτηκε να χρησιμοποιηθεί ένας και μόνο αισθητήρας, ο μόνος τρόπος να επιτευχθεί επαρκώς μία σωστή ανίχνευση εμποδίων εντός ενός χώρου που θα κινείται το όχημα είναι η χρήση ενός κινητήρα ο οποίος θα κινεί τον αισθητήρα περίπου 120°. Ο αισθητήρας τοποθετήθηκε στην ειδική βάση και έπειτα όλο μαζί βιδώθηκε στον άξονα του κινητήρα.

Οι κινητήρες Servo είναι ιδανικοί για τέτοιου τύπου εφαρμογές γιατί μέσω προγράμματος μπορούμε να κατευθύνουμε τον κινητήρα όπως και για όσο εμείς θέλουμε γνωρίζοντας πάντα την ακριβής θέση του .

Γενικά

Οι servo κινητήρες αποτελούνται από έναν κινητήρα συνεχούς ρεύματος, ένα κύκλωμα το οποίο είναι υπεύθυνο για τον έλεγχο της θέσης του άξονα του κινητήρα καθώς επίσης και ένα κιβώτιο γραναζιών το οποίο υποβιβάζει την σχέση μετάδοσης του .

Παρότι κινητήρας δεν εκτελεί πλήρης περιστροφή παρά μόνο μερικές μοίρες, αυτές εξαρτάται με τον τύπο του κινητήρα και τα τεχνικά χαρακτηριστικά του. Για τον έλεγχο της περιστροφής απαιτείται σαφώς τάση στον κινητήρα αλλά επίσης ένα σήμα ελέγχου. Το σήμα ελέγχου στη περίπτωση μας το τοποθετούμε σε μία ψηφιακή έξοδο στην οποία θα βάζουμε τις ανάλογες τιμές που χρειάζονται για την περιστροφή.

Τα πλεονεκτήματα των κινητήρων Servo είναι ότι είναι χαμηλοί σε κόστος , είναι μικροί σε μέγεθος πράγμα που τους κάνει ιδανικούς για ρομποτικές εφαρμογές όπως βραχίονες κτλ , έχουν υψηλή ροπή λόγω του κιβωτίου γραναζιών και τέλος δεν απαιτείται κάποιο αισθητήριο για τον προσδιορισμό της θέσης του άξονα του.

Εικόνα 3.15 Κινητήρας Servo

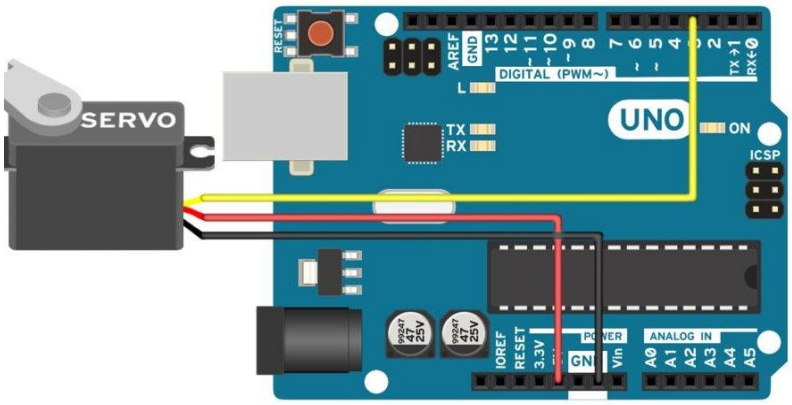
Στην παραπάνω εικόνα 25 φαίνεται ο κινητήρας που χρησιμοποιήθηκε στην εφαρμογή. Δεν υπάρχουν συγκεκριμένες απαιτήσεις στη συγκεκριμένη εφαρμογή όπως για παράδειγμα υψηλή ροπή ή ταχύτητα οπότε ο κινητήρας αυτός κρίθηκε ιδανικός.

Τεχνικά χαρακτηριστικά κινητήρα Servo

- Τάση λειτουργίας : 4,8 – 6 Vdc
- Περιστροφή : 120°
- Ταχύτητα (6V) : 0.10sec/60°
- Ροπή (6V) : 1.5Kg.cm
- Τύπος γραναζιών : πλαστικό

Σύνδεση Servo κινητήρα με Arduino

Arduino Pin	**Servo κινητήρας**
+5 V	Κόκκινο καλώδιο
Gnd	Μαύρο καλώδιο
Digital Output 3	Κίτρινο καλώδιο



Εικόνα 3.16 Συνδεσμολογία Servo κινητήρα με Arduino

Παράδειγμα προγραμματισμού με Servo

Για την χρήση του Servo κινητήρα σε μία εφαρμογή απαιτείται να συμπεριληφθεί η κατάλληλη βιβλιοθήκη και δεν είναι άλλη από την <servo.h> . Ακολουθεί ένα απλό πρόγραμμα παρακάτω :

```
//Συμπεριλαμβάνουμε την βιβλιοθήκη

#include <Servo.h>

// Δήλωση του Pin

int servoPin = 3;

// Δημιουργία ενός αντικειμένου Servo

Servo Servo1;

void setup() {

  // Συμπεριλαμβάνουμε το αντικείμενο στο  Pin που ορίσαμε

  Servo1.attach(servoPin);

}

void loop(){

  // Μετακίνηση του servo στις 0 μοίρες

  Servo1.write(0);
```

```
  delay(1000);
  // Μετακίνηση του servo στις 90 μοίρες
  Servo1.write(90);
  delay(1000);
  // Μετακίνηση του servo στις 180 μοίρες
  Servo1.write(180);
  delay(1000);
}
```

Στο πρόγραμμα αυτό ο κινητήρας θα κινείται και θα πηγαίνει στις 0 μοίρες θα μένει εκεί για ένα δευτερόλεπτο στη συνέχεια θα μετακινείται στις 90 μοίρες θα μένει για άλλο ένα δευτερόλεπτο και τέλος θα μετακινείται στις 180 μοίρες θα μένει για ένα δευτερόλεπτο και θα αρχίζει πάλι από την αρχή έως ότου αφαιρέσουμε την τροφοδοσία του arduino.

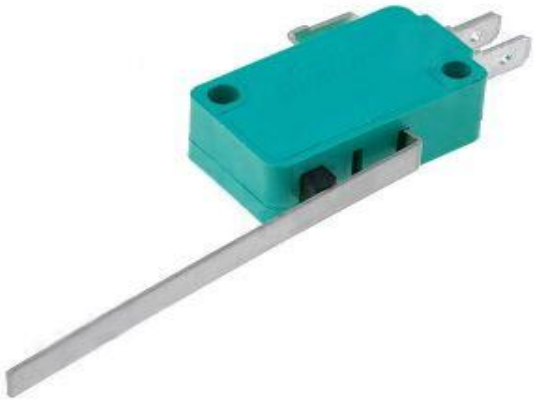
Επέκταση γωνίας μέτρησης

Όπως προαναφέραμε για την επέκταση γωνίας μέτρησης χρησιμοποιήθηκε η βάση που φαίνεται στην παρακάτω εικόνα 27. Με την χρήση αυτής της βάσης και την επέκταση της γωνίας αποφεύγουμε την χρήση παραπάνω αισθητηρίων .Σε περίπτωση που είχαμε χρησιμοποιήσει τρείς αισθητήρες για παράδειγμα θα έπρεπε να προγραμματίσουμε τρείς αισθητήρες αντί για έναν που σημαίνει περισσότερος κώδικας περισσότερα δεδομένα προς επεξεργασία και επίσης περισσότερο κόστος.

Εικόνα 3.17 Βάση στήριξης αισθητήρα υπερήχων

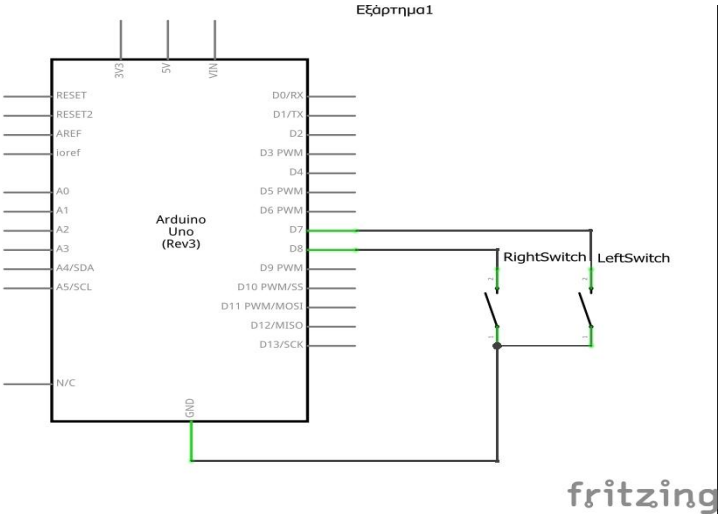
3.1.5 Διακόπτες επαφής NO (Normally Open)

Χρησιμοποιώντας τον κινητήρα Servo καθώς και την βάση για τον αισθητήρα υπερήχων αυτός τοποθετήθηκε σχετικά ψηλά με αποτέλεσμα να καταστεί αδύνατον να ανιχνεύσει τα αντικείμενα που βρίσκονται χαμηλότερα από το εύρος μέτρησης του, για αυτό το λόγο τοποθετήθηκαν διακόπτες επαφής NO(Normally Open) .



Εικόνα 3.18 Διακόπτης επαφής

Οι διακόπτες που χρησιμοποιήθηκαν έχουν μεταγωγική επαφή , δηλαδή έχει ένα κοινό άκρο και τα άλλα δύο σε θέση ηρεμίας το ένα άκρο είναι ανοιχτό ενώ το άλλο κλειστό. Στη περίπτωση του οχήματος χρησιμοποιήθηκε η ανοιχτή δηλαδή σε θέση ηρεμίας που σημαίνει ότι το όχημα δεν ακουμπάει κάποιο αντικείμενο η επαφή είναι ανοιχτή και όταν αυτό έρθει σε επαφή με αντικείμενο τότε αυτή κλείνει. Στο κύκλωμα συνδέεται σε μία είσοδο του Arduino το κοινό άκρο της επαφής και η άλλη στη γείωση. Η είσοδος του Arduino είναι προγραμματισμένη με μία εντολή να είναι μόνιμα σε κατάσταση HIGH οπότε κατά συνέπεια όταν η επαφή κλείσει να οδηγηθούν τα 5V στη γείωση δηλαδή θα γίνουν 0V. Στο πρόγραμμα αυτό θα μεταφραστεί σε συνθήκη ότι όταν διαβάσει 0V θα κάνει μία κίνηση τέτοια ώστε να αποφύγει το αντικείμενο από την αντίστοιχη επαφή από την οποία ήρθε σε επαφή.



Εικόνα 3.19 Σύνδεση επαφών στον Arduino

Σύνδεση επαφής με τον Arduino και προγραμματισμός

Για να επιτύχουμε τη επιθυμητή λειτουργία της επαφής θα πρέπει να προγραμματίσουμε την είσοδο να λειτουργεί σαν αντίσταση pullup. Αυτό γίνεται πολύ απλά με την εντολή pinMode(2,INPUT_PULLUP); Που σημαίνει ότι την ψηφιακή είσοδο 2 του Arduino θα την χρησιμοποιεί με την εσωτερική αντίσταση Pullup.

Στην αρχή του προγράμματος της εφαρμογής θα δηλωθούν σαν σταθερές τα αντίστοιχα pins που θα έχουν συνδεθεί οι επαφές και θα δηλωθεί η κατάστασή τους ως μεταβλητή. Μετά θα οριστούν σαν Pullup , και στη συνέχεια θα διαβάζεται η κατάστασή τους και θα εκτελείται η αντίστοιχη κίνηση από το όχημα.

Απόσπασμα από κώδικα του ρομποτικού οχήματος :

```
//Σταθερές

const int rightSW = 8;

const int leftSW = 7;

//μεταβλητές

int rightSWState;

intleftSWState;

void setup()
```

```
{
//Ορισμός
pinMode(rightSW,INPUT_PULLUP);
pinMode(leftSW,INPUT_PULLUP);
}
void loop ()
{
//Ανάγνωση δεδομένων από κατάσταση επαφών
rightSWstate = digitalRead (rightSW);
leftSWstate = digitalRead(leftSW);
//Επαφή με αντικείμενο στη δεξιά επαφή
if  (rightstate = = LOW ) {
backward () ;                    // κίνηση πίσω
delay(500);
left ();                         //κίνηση αριστερά
delay(300);
}
```

```
//επαφή με αντικείμενο στην αριστερή επαφή
if  (leftstate = = LOW ) {
backward () ;                    //κίνηση πίσω
delay(500);
```

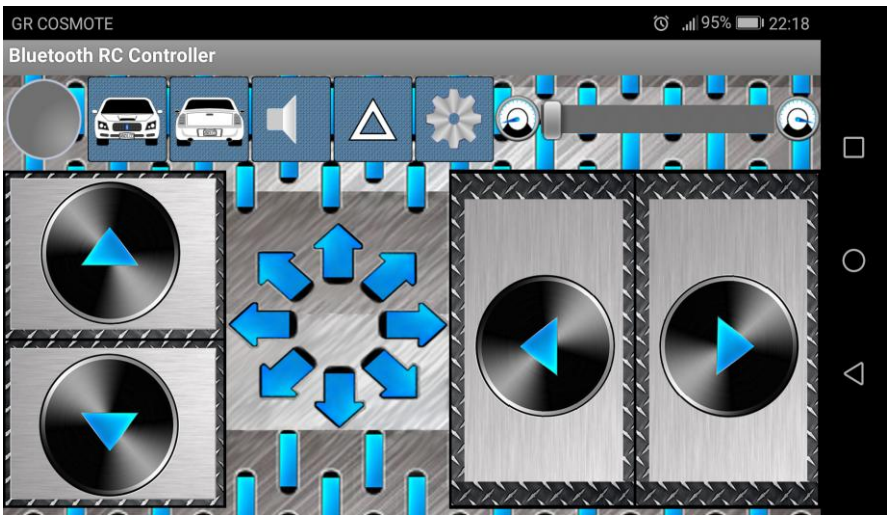
```
right ();                          //κίνηση δεξιά
delay(300); }
else {
forward ()}                        // κίνηση εμπρός
```

ΚΕΦΑΛΑΙΟ 4

Εφαρμογή BLUETOOTH

4.1 Περιβάλλον εφαρμογής

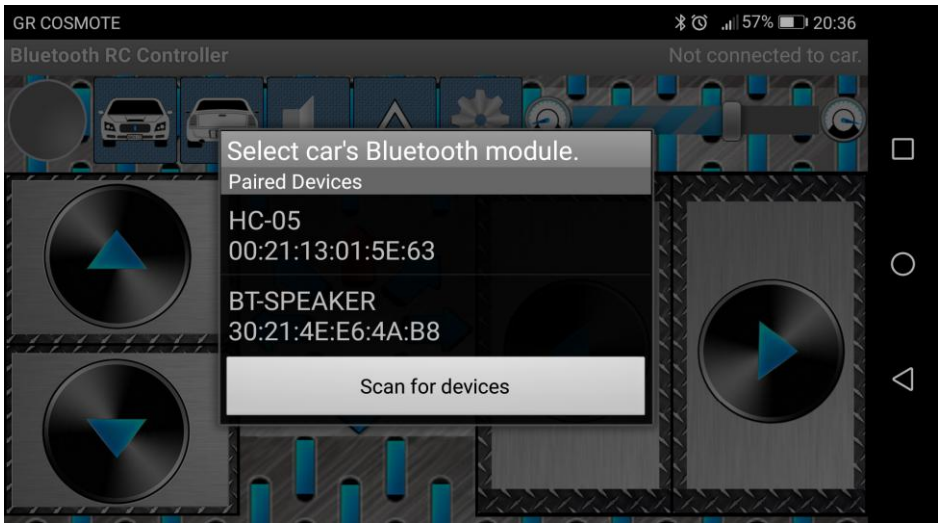
Η εφαρμογή διατίθεται δωρεάν στο playstore ονομάζεται ‘’ Bluetooth RC controller’’ και προορίζεται για τις συσκευές που χρησιμοποιούν android λογισμικό. Πρόκειται για μία εφαρμογή η οποία έχει δημιουργηθεί για τηλεχειρισμό οχημάτων και έχει κουμπιά για κατευθύνσεις και για διάφορες άλλες λειτουργίες τις οποίες μπορούμε να χρησιμοποιήσουμε. Έχει όλες τις κινήσεις που μπορεί να κάνει ένα όχημα καθώς επίσης και έναν ρυθμιστή ταχύτητας. Οι υπόλοιπες λειτουργίες είναι για φώτα εμπρός και πίσω ,κόρνα και ‘’alarm’’. Οι λειτουργίες αυτές δεν είναι υποχρεωτικές για αυτό το σκοπό μπορούμε να τις αξιοποιήσουμε όπως εμείς επιθυμούμε. Στην εφαρμογή του ρομποτικού οχήματος χρησιμοποιήθηκαν οι βασικές κινήσεις ,ο ρυθμιστής ταχύτητας και ο διακόπτης των ‘’alarm’’ ο οποίος είχε σκοπό την εναλλαγή του mode από χειροκίνητο σε αυτόματο.



Εικόνα 4.1 Περιβάλλον εφαρμογής

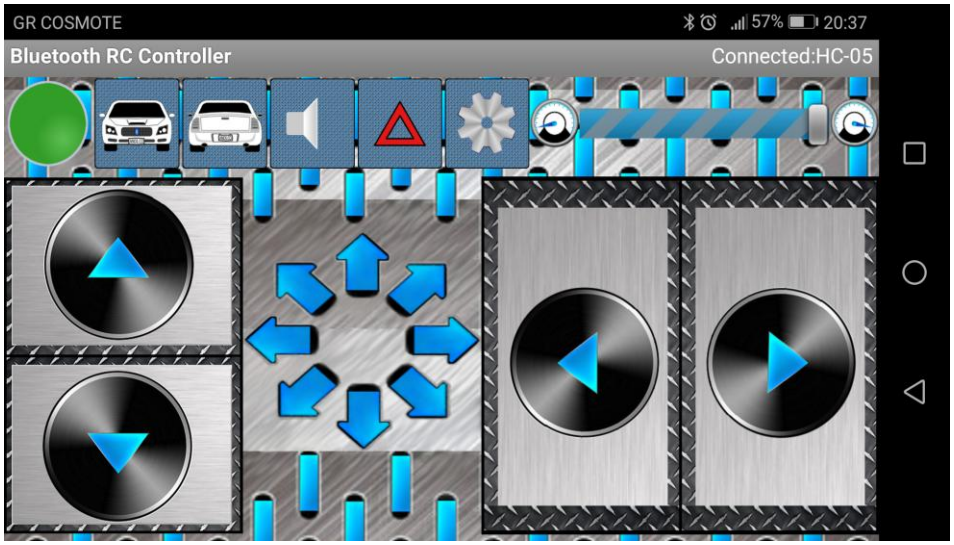
Σύνδεση με το όχημα

Είναι αρκετά εύχρηστο και προσιτό για τον οποιοδήποτε χρήστη .Η σύνδεση με το όχημα γίνεται καθώς ανοίγει η εφαρμογή διότι μας ζητά να ενεργοποιήσουμε το Bluetooth του κινητού , πατώντας αποδοχή ενεργοποιείται και κάνουμε αναζήτηση το όχημα μας .



Εικόνα 4.2 Αναζήτηση του οχήματος

Βρίσκοντας το όνομα της συσκευής του Bluetooth του οχήματος στη συγκεκριμένη περίπτωση είναι ο τύπος του αισθητήρα δηλαδή '' HC-05 '' πατάμε σύνδεση και εάν είναι επιτυχημένη η σύνδεση θα πρασινίσει το ''λαμπάκι'' που αναβοσβήνει πριν γίνει η σύνδεση με το όχημα.



Εικόνα 4.3 Επιτυχημένη σύνδεση οχήματος με την εφαρμογή

Στην εικόνα 32 φαίνεται ότι το όχημα έχει επιτυχημένη σύνδεση με την εφαρμογή και επίσης ότι είναι ‘’πατημένο’’ το πλήκτρο με τα ‘’alarm’’ που σημαίνει ότι το όχημα κινείται χειροκίνητα δηλαδή μέσω των κινήσεων που αποστέλλονται από την εφαρμογή . Επίσης μπορούμε να διακρίνουμε το ρυθμιστή ταχύτητας ότι είναι στο maximum οπότε το όχημα μας κινείται με την μέγιστη ταχύτητα που μπορεί όταν εμείς αποστέλλουμε εντολή κίνησης , εάν όμως μετακινήσουμε τον ρυθμιστή ταχύτητας στο minimum τότε το όχημα θα πάψει να κινείται είτε αποστέλλουμε εντολές είτε όχι. Με το πάτημα του ‘’πλήκτρου’’ με τα ‘’alarm’’ τότε το όχημα θα ξεκινήσει την αυτόνομη λειτουργία του και θα κινείται μόνο βάση των αισθητηρίων που εμείς έχουμε προγραμματίσει .

4.2 Αποστολή εντολών κίνησης οχήματος στον Arduino

Το κάθε πλήκτρο που εμείς πατάμε αντιστοιχεί και σε κάποια εντολή η οποία αποστέλλεται στον Arduino και βάσει αυτών προγραμματίζουμε αντίστοιχα .Οι εντολές είναι χαρακτήρες τύπου char , και για να διαπιστώσουμε τι αποστέλλεται την στιγμή που πατάμε το πλήκτρο στη συσκευή ανοίγουμε την σειριακή οθόνη .Εκεί απεικονίζεται συνεχώς ο χαρακτήρας που αποστέλλεται.

Αντιστοιχία λειτουργιών με χαρακτήρες

Πλήκτρο (κίνηση)	**Χαρακτήρας**
Εμπρός	F
Πίσω	B
Αριστερά	L
Δεξιά	R
Μπροστά αριστερά	G
Μπροστά δεξιά	I
Πίσω αριστερά	H

Πίσω δεξιά	J
Στοπ	S
Μπροστά φώτα On	W
Μπροστά φώτα Off	w
Πίσω φώτα On	U
Πίσω φώτα Off	u
Κόρνα On	V
Κόρνα Off	v
Alarm On	X
Alarm Off	x
Ταχύτητα 0	0
Ταχύτητα 10	1
Ταχύτητα 20	2
Ταχύτητα 30	3
Ταχύτητα 40	4
Ταχύτητα 50	5
Ταχύτητα 60	6
Ταχύτητα 70	7
Ταχύτητα 80	8
Ταχύτητα 90	9
Ταχύτητα 100	10

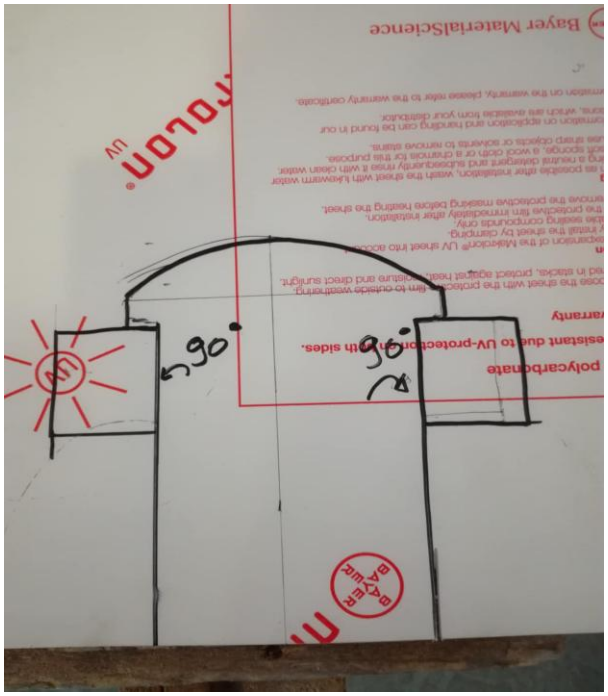
Στην εφαρμογή του ρομποτικού οχήματος δεν χρησιμοποιήθηκαν όλες αυτές οι εντολές .Αυτές που αξιοποιήθηκαν θα αναλυθούν σε επόμενο κεφάλαιο που αφορά τον προγραμματισμό του Arduino .

ΚΕΦΑΛΑΙΟ 5

ΚΑΤΑΣΚΕΥΗ ΡΟΜΠΟΤΙΚΟΥ ΟΧΗΜΑΤΟΣ

5.1 Κατασκευή σασί οχήματος

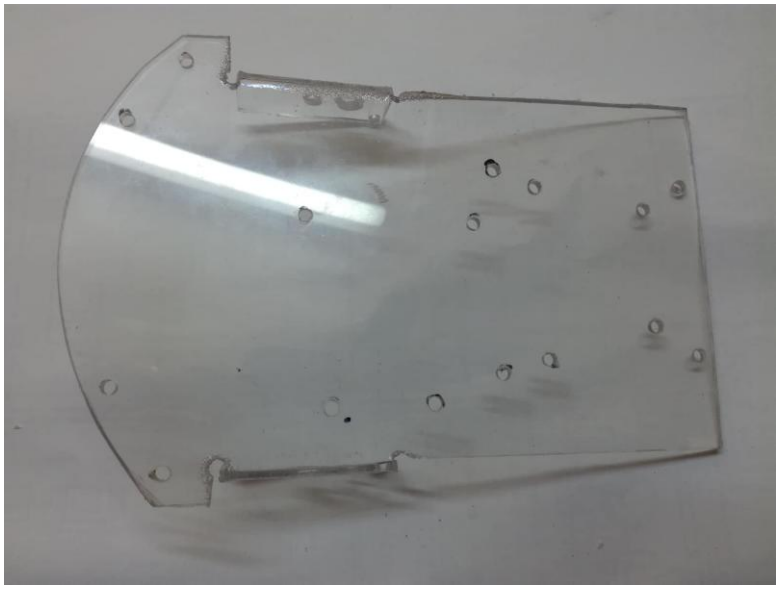
Το σασί του οχήματος είναι εξ' ολοκλήρου σχεδιασμένο και κατασκευασμένο από την αρχή . Το υλικό που επιλέχθηκε είναι το plexi glass ένα υλικό ελαφρύ και εύχρηστο πράγμα που το κάνει κατάλληλο για αυτή την κατασκευή. Οι διαστάσεις αποφασίστηκαν όταν συγκεντρώθηκαν όλα τα υλικά που θα τοποθετούνταν πάνω σε αυτό. Υπάρχουν έτοιμα σασί στο εμπόριο αλλά η κατασκευή εξ 'ολοκλήρου ήταν μια πρόκληση για αυτό το λόγο προτιμήθηκε αυτή η λύση.



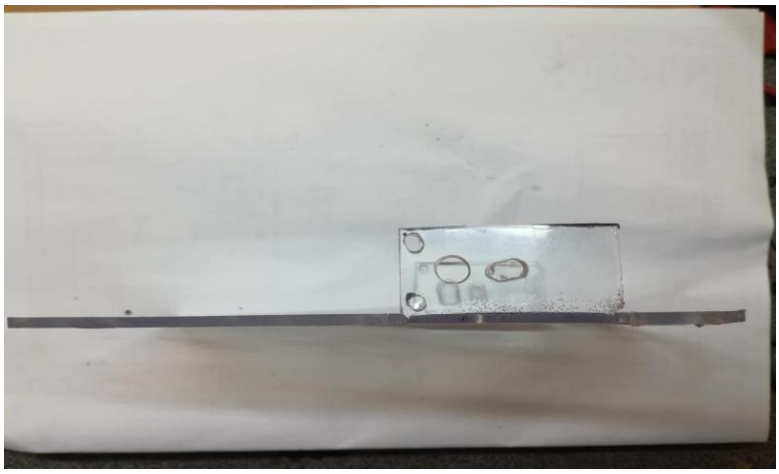
Εικόνα 5.1 Σχεδιασμός σασί σε plexiglass

Όπως φαίνεται στην παραπάνω εικόνα 33 φαίνεται το σχέδιο του σασί που έχει γίνει στο plexiglass πριν κοπεί .Τα τετράγωνα πλαίσια που υπάρχουν δεξιά και αριστερά είναι οι βάσεις των κινητήρων, για να γίνει σωστά η βάση για τον κάθε κινητήρα πρέπει αφού κοπεί το σασί να γυρίσουμε τα τετράγωνα πλαίσια 90° δεξιά και αριστερά αντίστοιχα.

Αφού κόπηκε το σασί επόμενο βήμα ήταν να τοποθετηθούν τα υλικά πάνω με τρόπο τέτοιο ώστε να είναι λειτουργικό το όχημα και να μην ενοχλούν την λειτουργία του. Οι τρύπες για τα υλικά και τα αισθητήρια έγιναν όπως επίσης και οι βάσεις των κινητήρων γύρισαν 90° με τη βοήθεια πιστολιού θερμού αέρα και έγιναν και οι κατάλληλες τρύπες για να υποδεχθούν τους κινητήρες .



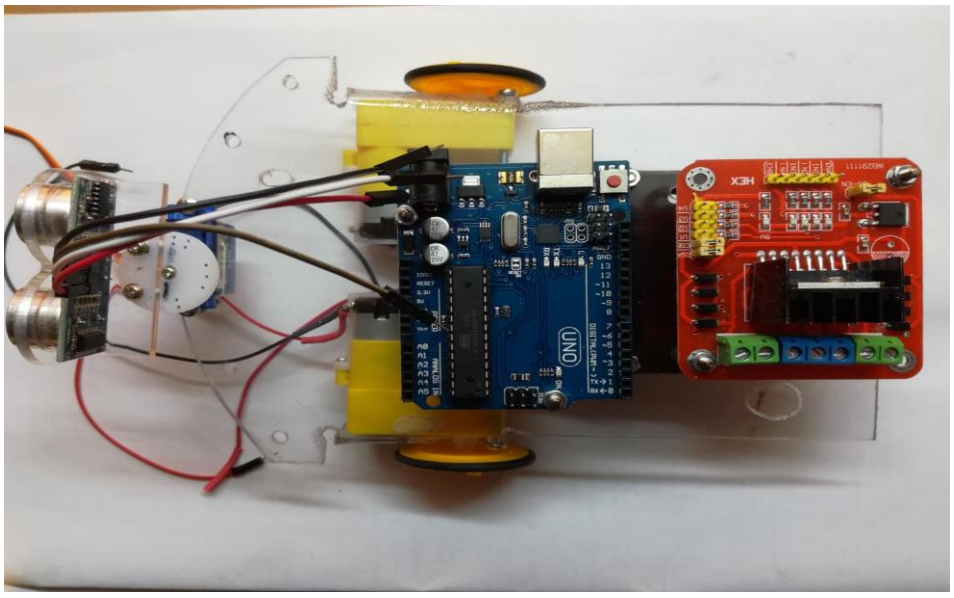
Εικόνα 5.2 Σασί οχήματος



Εικόνα 5.3 Βάσεις κινητήρων

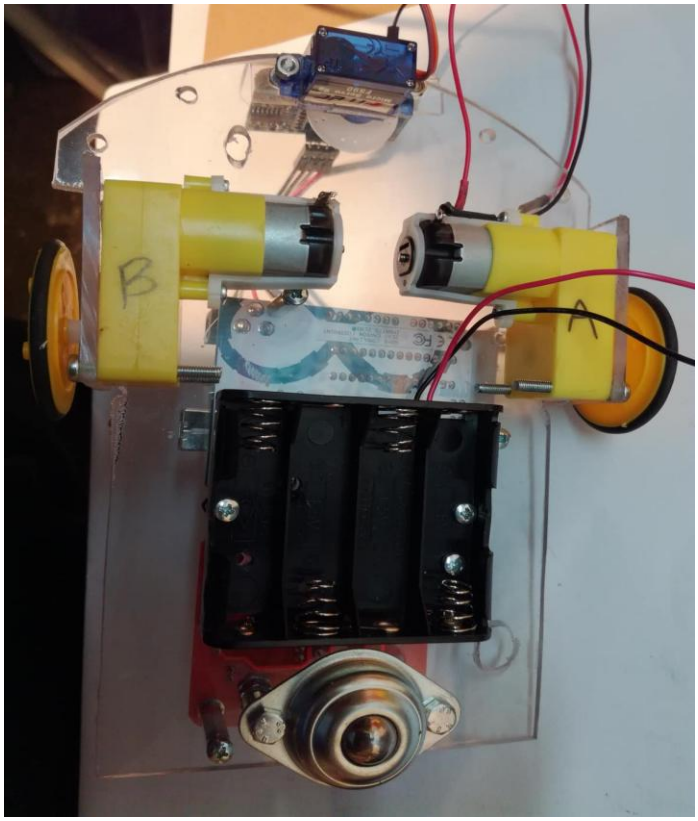
5.2Τοποθέτηση υλικών στο όχημα

Τοποθετήθηκαν υλικά πάνω καθώς και κάτω από το σασί , αυτό έγινε για την εξοικονόμηση χώρου και όγκου του οχήματος . Για την στήριξη των υλικών χρησιμοποιήθηκαν αποστάτες οι οποίοι βοηθούν στο να τοποθετηθούν τα υλικά λίγο ψηλότερα από το σασί .Αυτό εξυπηρετεί και στο να περνούν καλώδια κάτω από αυτά αλλά και να στηρίζονται σωστά τα υλικά.



Εικόνα 5.4 Τοποθέτηση των πρώτων υλικών

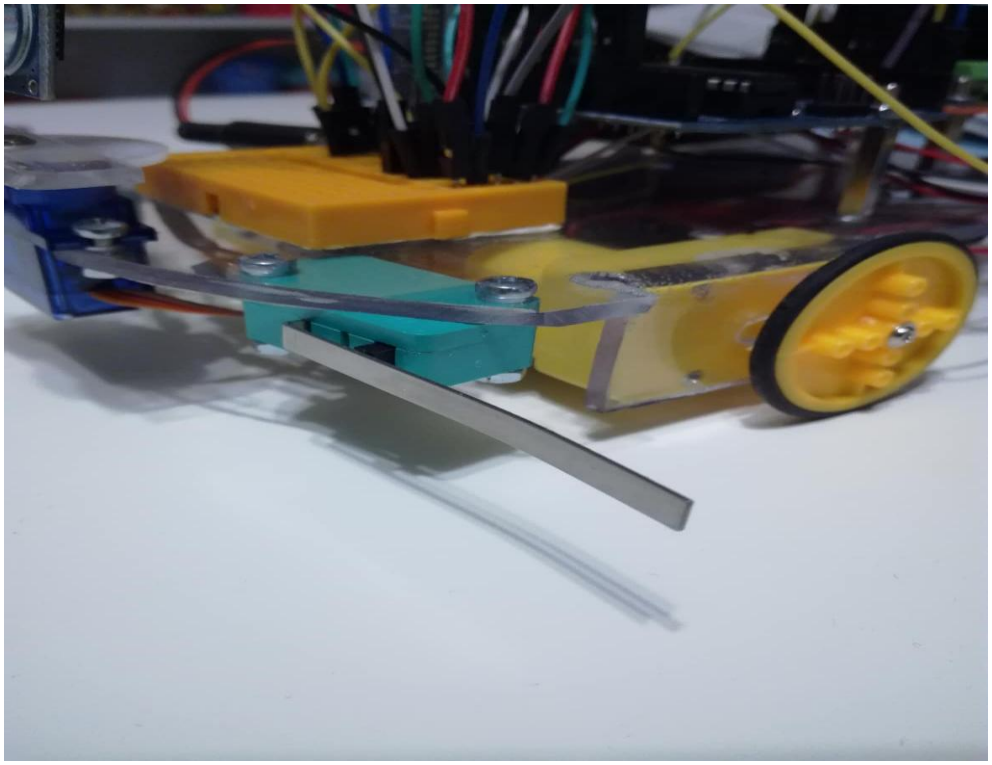
Οι κινητήρες τοποθετήθηκαν μπήκαν οι ρόδες καθώς και τα πρώτα υλικά όπως ο μικροεπεξεργαστής η ‘’γέφυρα Η’’ και ο σερβοκινητήρας με τον αισθητήρα υπερήχων. Όπως φαίνεται περισσεύουν τρύπες αυτές είναι για το πέρασμα των καλωδίων που θα έρχονται από τα κάτω υλικά τα οποία θα φανούν παρακάτω.



Εικόνα 5.5 Όχημα όπως φαίνεται από κάτω

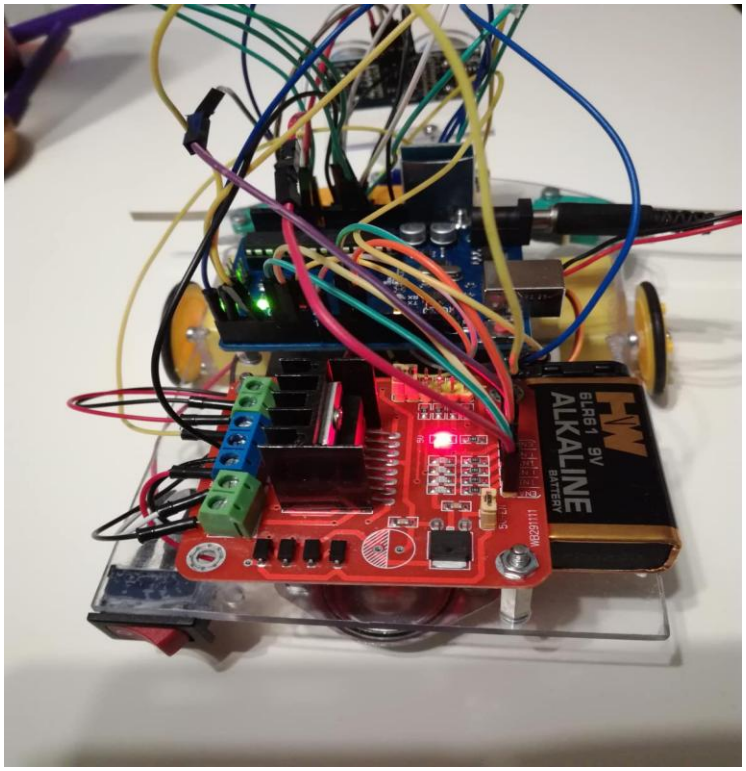
Στο ρομποτικό όχημα από την κάτω μεριά τοποθετήθηκαν σαφώς οι κινητήρες όπως επίσης και η θήκη των μπαταριών . Η θήκη αυτή χρησιμοποιείται για την τροφοδοσία των κινητήρων. Το όχημα χρησιμοποιεί δύο κινητήρες οπότε για την σωστή στήριξη αλλά και για το ευκολότερο στρίψιμο του απαιτείται η μπίλια που φαίνεται δίπλα στην θήκη των μπαταριών. Τοποθετείται στο πίσω μέρος διότι μπροστά το βάρος μοιράζεται στους κινητήρες οπότε η υπόλοιπη κατανομή του βάρους γίνεται πίσω στην μπίλια αυτή .Επίσης φαίνεται ο σερβοκινητήρας από την κάτω μεριά .Για την σωστή λειτουργία του οχήματος δημιουργήθηκε μια εσοχή στο σασί. Επειδή θα τοποθετηθούν διακόπτες επαφής και επειδή ο σερβοκινητήρας βρίσκεται πιο ψηλά από αυτές θέλαμε να τοποθετηθεί όσο το δυνατόν πιο πίσω από αυτές ,αυτό διότι στην αυτόνομη λειτουργία του οχήματος όταν ένα εμπόδιο βρίσκεται χαμηλά να

προλαβαίνει το όχημα να αλλάζει κατεύθυνση από την επαφή του οχήματος με τους διακόπτες επαφής. Έτσι αποφεύγουμε την επαφή του αισθητήρα με κάποιο εμπόδιο που δεν μπορεί να εντοπίσει επειδή βρίσκεται χαμηλότερα από αυτόν.



Εικόνα 5.6 Διακόπτες επαφής του οχήματος

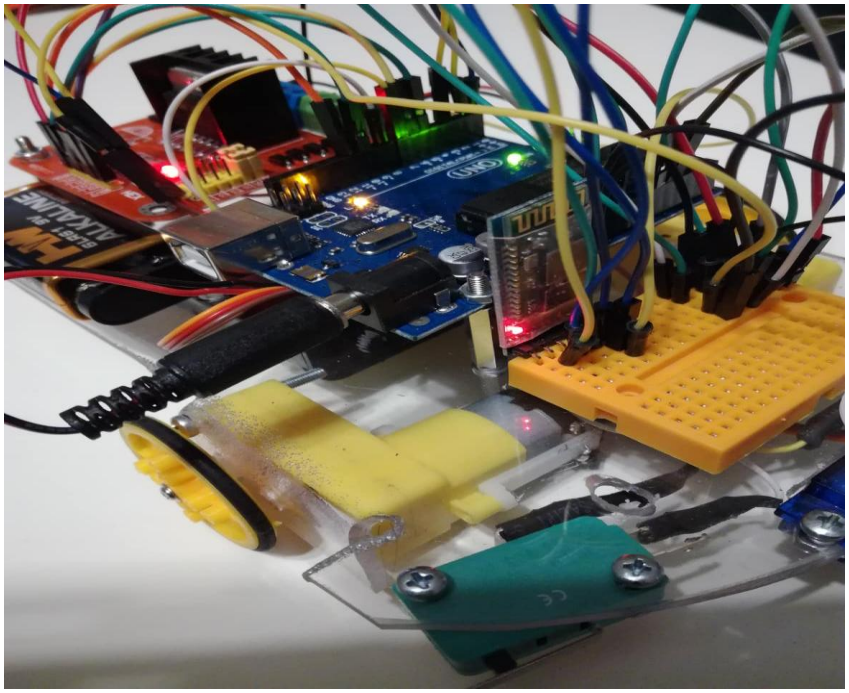
Οι διακόπτες επαφής τοποθετήθηκαν στα μπροστινά άκρα του οχήματος. Επιλέχθηκαν διακόπτες με μακριά άκρα για καλύτερο εντοπισμό εμποδίων. Επίσης δεν θα μπορούσε να λείπει το γνωστό σε όλους breadboard για τις συνδεσμολογίες και την στήριξη της συσκευής του Bluetooth. Οι συνδεσμολογίες όλες έγιναν με καλώδια jumper . Στις επαφές χρειάστηκε να κολλήσουμε καλώδια καθώς όταν αγοράζονται δεν υπάρχουν καλώδια οπότε κολλήθηκαν και συνδέθηκαν .Το ίδιο έγινε και με τα καλώδια των κινητήρων .



Εικόνα 5.7 Πίσω όψη του οχήματος

Στην εικόνα 39 φαίνεται ολοκληρωμένο το όχημα από την πίσω μεριά του. Μπορούμε να διακρίνουμε έναν διακόπτη και μία ακόμη μπαταρία 9V. Ο διακόπτης ανήκει στην τροφοδοσία της πλακέτας οδήγησης των κινητήρων με αυτό τον τρόπο μπορούμε να διακόψουμε την λειτουργία των κινητήρων ανά πάσα στιγμή. Όπως έχει αναφερθεί ξανά το όχημα θα λειτουργεί εντελώς αυτόνομα οπότε δεν θα μπορούσε να λείπει η τροφοδοσία του μικροελεγκτή . Αυτή γίνεται με μία μπαταρία 9V και συνδέεται με τον Arduino με έτοιμο καλώδιο από μπαταρία 9V το ένα άκρο και το άλλο για το βύσμα εξωτερικής τροφοδοσίας του Arduino.

Εικόνα 5.8 Αντάπτορας τροφοδοσίας Arduino



Εικόνα 5.9 Δεξιά μεριά οχήματος

Σε αυτή τη μεριά μπορούμε να διακρίνουμε και την συσκευή του Bluetooth και όλες τις καλωδιώσεις που έχουν γίνει για το όχημα. Με όλα αυτά το όχημα είναι έτοιμο να δεχθεί τον κατάλληλο κώδικα και να λειτουργήσει . Έχοντας κάνει αρκετές δοκιμές με το Bluetooth και την σειριακή επικοινωνία καθώς και με τα αισθητήρια καταλήξαμε στο κώδικα που να αναλυθεί στο επόμενο κεφάλαιο .

ΚΕΦΑΛΑΙΟ 6

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ARDUINO

6.1 Ανάλυση λειτουργίας του κώδικα

Στην αρχή του κώδικα δηλώνονται όλες οι σταθερές καθώς και οι μεταβλητές . Πριν από την setup ορίζουμε τον σερβοκινητήρα στις 90 μοίρες πράγμα που σημαίνει ότι το αισθητήριο των υπερήχων κοιτάζει μπροστά. Επίσης προκαθορίζεται η ταχύτητα που θα έχει το όχημα στην αυτόνομη χρήση και αυτό γίνεται διότι χρησιμοποιούμε pwm εξόδους για τους κινητήρες.

Στη setup γίνεται η δήλωση των εισόδων – εξόδων. Εδώ να τονίσουμε ότι οι διακόπτες επαφής δηλώνονται με την μέθοδο των pullup , και στη συνέχεια γίνεται η έναρξη της σειριακής επικοινωνίας.

Περνώντας στη ρουτίνα επανάληψης ξεκινάει διαβάζοντας το byte το οποίο είναι υπεύθυνο για το mode που θα επιλέξουμε εμείς ωστόσο μέχρι να δώσουμε εμείς το ανάλογο byte το όχημα παραμένει ακινητοποιημένο και με τον αισθητήρα να κοιτάζει εμπρός.

Χειροκίνητο mode

Επιλέγοντας το manual mode τότε το όχημα παραμένει ακινητοποιημένο μέχρι να λάβει εντολή από την εφαρμογή . Λαμβάνοντας εντολή τότε κάνει την αντιστοίχηση της εντολής με την κίνηση. Στο τέλος του κώδικα υπάρχουν όλες οι κινήσεις του οχήματος έτσι ανάλογα με την εντολή το πρόγραμμα καλεί την κίνηση .Στο όχημα αυτό έχουμε δώσει την δυνατότητα οχτώ κινήσεων στο manual mode. Να αναφέρουμε εδώ ότι σε αυτό το mode το όχημα δεν λαμβάνει υπ όψιν του κανένα από τα αισθητήρια.

Αυτόματο mode

Επιλέγοντας το αυτόματο mode ξεκινάει να χρησιμοποιεί τα αισθητήρια. Έτσι ο αισθητήρας υπερήχων διαβάζει και μέσω της βιβλιοθήκης η οποία μεταφράζει τα μικροκύματα σε απόσταση σε cm . Εάν εντοπίσει εμπόδιο σε απόσταση 5 εκατοστών τότε ακινητοποιείται και κάνει έλεγχο δεξιά στρέφοντας τον αισθητήρα στις 180 μοίρες εκεί κάνει πάλι έλεγχο και αποθηκεύει μια τιμή και στη συνέχεια κάνει έλεγχο και από αριστερά δηλαδή ο αισθητήρας μετακινείται στις 0 μοίρες , αποθηκεύοντας και εκεί την τιμή. Μετά τιε μετρήσεις δεξιά και αριστερά επιστρέφει τον αισθητήρα στην αρχική του θέση δηλαδή 90 μοίρες. Έχοντας λάβει μετρήσεις και από τις δύο μεριές τότε γίνεται σύγκριση των δύο τιμών που έλαβε και εάν η απόσταση από δεξιά είναι μικρότερη από αυτήν στα αριστερά τότε το όχημα κινείται πίσω, αριστερά και μετά συνεχίζει να κινείται εμπρός ,εάν η απόσταση από αριστερά είναι μικρότερη από αυτήν στα δεξιά τότε το όχημα κινείται πίσω , αριστερά και μετά συνεχίζει εμπρός πάλι μέχρι ο αισθητήρας να εντοπίσει αντικείμενο σε απόσταση περίπου πέντε εκατοστών. Σε περίπτωση όμως που στην σύγκριση των τιμών δεξιά και αριστερά οι τιμές που λάβει είναι ίσες τότε το όχημα κινείται πίσω για περισσότερο χρόνο , μετά αριστερά και στη συνέχεια πάλι εμπρός .

Σε περίπτωση που κάποιο εμπόδιο δεν μπορέσει να εντοπιστεί από τον αισθητήρα υπερήχων τότε θα γίνει αντιληπτό από τους διακόπτες επαφής δεξιά η αριστερά .Εάν κλείσει η επαφή στα δεξιά τότε το όχημα κινείται πίσω ,αριστερά και συνεχίζει εμπρός. Εάν τώρα κλείσει η επαφή στα αριστερά τότε κινείται πίσω, δεξιά και στη συνέχεια εμπρός .

Ρύθμιση ταχύτητας

Η ρύθμιση της ταχύτητας του οχήματος είναι εφικτή και στα δύο mode. Η ασφαλέστερη ταχύτητα για το αυτόματο mode είναι μία ταχύτητα στο μισό περίπου για εγκυρότερη και σιγουρότερη μέτρηση από τα αισθητήρια ή τους διακόπτες επαφής.

Κινήσεις οχήματος

Οι κινήσεις του οχήματος βρίσκονται στο τέλος του προγράμματος και είναι όπως προαναφέραμε οχτώ. Η κάθε κίνηση δίνει στα άκρα των κινητήρων τις αντίστοιχες τιμές . Η τιμή που παίρνει η έξοδος είναι η τιμή αυτή που επηρεάζεται από την εφαρμογή μετακινώντας τον ρυθμιστή ταχύτητας.

6.2 Κώδικας

```
#include <Ultrasonic.h>



#include <Servo.h>

//Σταθερές
const int motorR1 = 6;
const int motorR2 = 5;
const int motorL1 = 11;
const int motorL2 = 10;

const int servo = 3;

const int rightSW = 8;
const int leftSW  = 7;

Ultrasonic ultrasonic(A0,A1); // Trig, Echo
Servo myservo;



//Μεταβλητές
int distance=100;
int rightDistance;
int leftDistance;
int rightSWincomingData;
int leftSWincomingData;
```

```
int pos = 90;                 // Για τον  servo, 90 μοίρες κοιτά μπροστά
int incomingData;
char mode = 'm';              // χαρακτήρας m για χειροκίνητο, a για  αυτόματο
int vSpeed = 200;             //προκαθορισμός ταχύτητας κινητήρων (PWM)

void setup(){

  //Ορισμός εισόδων και εξόδων
  pinMode(motorR1, OUTPUT);
  pinMode(motorR2, OUTPUT);
  pinMode(motorL1, OUTPUT);
  pinMode(motorL2, OUTPUT);
  pinMode(leftSW,  INPUT_PULLUP);
  pinMode(rightSW, INPUT_PULLUP);

  // Αρχικοποίηση servo και θέση εκκίνησης
  myservo.attach(servo);
  myservo.write(pos);

  //Έναρξη σειριακής επικοινωνίας
  Serial.begin(9600);
}



void loop(){
```

```
  if(Serial.available() > 0){
      // read the incoming byte:
      incomingData = Serial.read();
      if (incomingData == 'X'){mode = 'X'; }
      if (incomingData == 'x'){mode = 'x';
         stop();
         pos=90;
         myservo.write(pos);
      }
   }


  //Αλλαγή ταχύτητας κίνησης του οχήματος εάν το  incomingData είναι ίσο από 0
εώς 10 τιμές από  0 εώς 255 (PWM)


    if (incomingData == '0'){
      vSpeed=0;}
    else if (incomingData == '1'){
      vSpeed=60;}
    else if (incomingData == '2'){
      vSpeed=80;}
    else if (incomingData == '3'){
      vSpeed=100;}
    else if (incomingData == '4'){
      vSpeed=130;}
    else if (incomingData == '5'){
      vSpeed=150;}
    else if (incomingData == '6'){
```

```
  vSpeed=170;}
 else if (incomingData == '7'){
  vSpeed=190;}
 else if (incomingData == '8'){
  vSpeed=210;}
 else if (incomingData == '9'){
  vSpeed=230;}
 else if (incomingData == '10'){
  vSpeed=255;}




//Χειροκίνητο  mode

 if (mode == 'X'){
  if (incomingData=='F'){
    forward();
 }
  else if (incomingData=='B'){
    backward();
  }
       else if (incomingData=='J'){
    rbackward();
  }
   else if (incomingData=='H'){
```

```
    lbackward();
  }


  else if (incomingData=='R'){
    right();
  }
  else if (incomingData=='L'){
    left();
  }
  else if (incomingData=='I'){
    rforward();
  }

    else if (incomingData=='G'){
    lforward();
  }

  else if (incomingData=='S'){
    stop();
  }
}

//Αυτόματο mode

else if (mode == 'x'){
  distance = ultrasonic.Ranging(CM);
```

```
  // Εάν εντοπιστεί αντικείμενο στα 5cm , ακινητοποίηση οχήματος και εύρεση δρόμου διαφυγής
       if (distance <= 10){
      stop();
      //Έλεγχος δεξιά
      for (pos = 0; pos <= 180; pos += 1) { // κίνηση αισθητήρα από 0 έως 180 μοίρες
        myservo.write(pos);
          delay(5);                       // αναμονή να φτάσει ο αισθητήρας στη θέση του
}


//Εντοπισμός νέας απόστασης από  την δεξιά μεριά


        leftDistance = ultrasonic.Ranging(CM);
        delay(100);
        // ¨Έλεγχος αριστερά
        for (pos = 180; pos >= 0; pos -= 1) { // κίνηση αισθητήρα από 0 έως 180 μοίρες


          myservo.write(pos);
          delay(5);                       // αναμονή να φτάσει ο αισθητήρας στη θέση του
        }


//Εντοπισμός νέας απόστασης από  την αριστερή μεριά
        rightDistance = ultrasonic.Ranging(CM);
        delay(100);
        pos = 90;    // Αρχική θέση αισθητήρα
        myservo.write(pos);
```

```
    //Σύγκριση αποστάσεων δεξιά και αριστερά και επιλογή καλύτερης οδού διαφυγής


      if (leftDistance > rightDistance){
        left();
        delay(500);   // κίνηση αριστερά για 0,5 sec
      }
      else if (leftDistance < rightDistance){
        right();          // κίνηση δεξιά για 0,5 sec
        delay(500);
      }
      else{     //¨Αν οι αποστάσεις δεξιά και αριστερά είναι ίσες τότε πίσω για 1 sec
        backward();
        delay(1000);
        left();   // και αριστερά για 0,5 sec
        delay(500);
      }
   }
//Ελεύθερο πεδίο , κίνηση εμπρός και έλεγχος με τις επαφές


   else{


     leftSWincomingData = digitalRead(leftSW);
     rightSWincomingData = digitalRead(rightSW);


     //¨Έλεγχος για επαφή από αριστερά
     if (leftSWincomingData == LOW){
```

```
        backward();
        delay(500);       //κίνηση πίσω και δεξιά
        right();
        delay(400);
      }
// ¨Ελεγχος για επαφή από δεξιά
      if (rightSWincomingData == LOW){
        backward();
        delay(500);          // κίνηση πίσω και αριστερά
        left();
        delay(400) }

     forward();
    }
    }}
// Συναρτήσεις κινήσεων

//κίνηση εμπρός
void forward(){
  analogWrite(motorR1, vSpeed);
  analogWrite(motorR2, 0);
  analogWrite(motorL1, vSpeed);
  analogWrite(motorL2, 0);
}
//κίνηση εμπρός αριστερά
void lforward(){
  analogWrite(motorR1, vSpeed/2);
```

```
  analogWrite(motorR2, 0);
  analogWrite(motorL1, vSpeed);
  analogWrite(motorL2, 0);
}
//κίνηση εμπρός δεξιά
void rforward(){
  analogWrite(motorR1, vSpeed);
  analogWrite(motorR2, 0);
  analogWrite(motorL1, vSpeed/2);
  analogWrite(motorL2, 0);
}
//κίνηση πίσω
void backward(){
  analogWrite(motorR1, 0);
  analogWrite(motorR2, vSpeed);
  analogWrite(motorL1, 0);
  analogWrite(motorL2, vSpeed);
}
//κίνηση πίσω δεξιά
void rbackward(){
  analogWrite(motorR1, 0);
  analogWrite(motorR2, vSpeed);
  analogWrite(motorL1, 0);
  analogWrite(motorL2, vSpeed/2);
}
//κίνηση πίσω αριστερά
void lbackward(){
```

```
  analogWrite(motorR1, 0);
  analogWrite(motorR2, vSpeed/2);
  analogWrite(motorL1, 0);
  analogWrite(motorL2, vSpeed);
}
//κίνηση πίσω δεξιά
void left(){
  analogWrite(motorR1, 0);
  analogWrite(motorR2, vSpeed);
  analogWrite(motorL1, vSpeed);
  analogWrite(motorL2, 0);
}
//κίνηση δεξιά
void right(){
  analogWrite(motorR1, vSpeed);
  analogWrite(motorR2, 0);
  analogWrite(motorL1, 0);
  analogWrite(motorL2, vSpeed);
}
//ακινητοποίηση
void stop(){
  analogWrite(motorR1, 0);
  analogWrite(motorR2, 0);
  analogWrite(motorL1, 0);
  analogWrite(motorL2, 0);
}
```

ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ

Οι προτάσεις για το μέλλον είναι οι εξής :

1. Ενσωμάτωση module wi-fi στον Arduino έτσι ώστε να χειρίζεται από οποιαδήποτε συσκευή είναι συνδεδεμένη στο διαδύκτιο.
2. Προσθήκη κάμερας ώστε να μπορούμε να έχουμε εικόνα από το χώρο που κινείται το ρομποτικό όχημα.
3. Κατασκευή μεγαλύτερου οχήματος ώστε να μπορεί να υποδεχθεί τις παραπάνω προσθήκες.
4. Προσθήκη περισσότερων και πιο αξιόπιστων αισθητηρίων για καλύτερη ανίχνευση εμποδίων .
5. Αλλαγή του μικροελεγκτή με κάποιον με περισσότερες δυνατότητες καθώς και εισόδους/εξόδους για τις ανάγκες της εξέλιξης του οχήματος.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. https://en.wikipedia.org/wiki/Arduino_Uno
2. https://learn.sparkfun.com/tutorials/what-is-an-arduino/all
3. https://web.stanford.edu/class/archive/engr/engr40m.1178/slides_sp17/arduino-io.pdf
4. http://wiki.sunfounder.cc/index.php?title=Bluetooth_Transceiver_Module_HC-06
5. https://components101.com/wireless/hc-06-bluetooth-module-pinout-datasheet
6. https://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html
7. https://www.electronics-tutorials.ws/io/io_7.html
8. https://iknowvations.in/arduino/ultrasonic-sensor-hc-sr04-arduino/
9. https://arduinobots.wordpress.com/2015/03/09
10. http://enigma.freebsdworld.gr/sites/default/files/enigma-part2.pdf
11. http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/
12. http://www.aishack.in/tutorials/l293d-hbridge/
13. http://www.ardumotive.com/arduino-ide-gr.html
14. https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/
15. Suman, R. P. (2014). *CElebrating Arduino Day .*
16. Πουλάκης,Εμμ. Προγραμματίζοντας με τον μικροελεγκτή Arduino
17. Γιάννης, Φ. *Μηχατρονικά συστήματα 1 .*
18. Άγγελος, Ρ. *Η Χρήση του Μικροελεγκτή Arduino στην Πρωτοβάθμια Εκπαίδευση με τη Γλώσσα Προγραμματισμού Scratch for Arduino (S4A).*

ΠΑΡΑΡΤΗΜΑ

Βιβλιοθήκη

Η βιβλιοθήκη βρέθηκε στο διαδίκτυο και προστέθηκε για τον αισθητήρα υπερήχων για την μετατροπή του χρόνου που δημιουργεί και στέλνει ο αισθητήρας μέχρι να τον λάβει σε εκατοστά. Παρακάτω φαίνεται και η πηγή της βιβλιοθήκης .

```
/*
  Ultrasonic.cpp - Library for HC-SR04 Ultrasonic Ranging Module.library

  Created by ITead studio. Apr 20, 2010.
  iteadstudio.com
*/

#include "Arduino.h"
#include "Ultrasonic.h"

Ultrasonic::Ultrasonic(int TP, int EP)
{
   pinMode(TP,OUTPUT);
   pinMode(EP,INPUT);
   Trig_pin=TP;
   Echo_pin=EP;
}

long Ultrasonic::Timing()
{
  digitalWrite(Trig_pin, LOW);
```

```
  delayMicroseconds(2);
  digitalWrite(Trig_pin, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trig_pin, LOW);
  duration = pulseIn(Echo_pin,HIGH);
  return duration;
}

long Ultrasonic::Ranging(int sys)
{
  Timing();
  distacne_cm = duration /29 / 2 ;
  distance_inc = duration / 74 / 2;
  if (sys)
  return distacne_cm;
  else
  return distance_inc;
}
```