ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
Τμήμα Μηχανικών Βιοϊατρικής

Πτυχιακή Εργασία

«ΑΝΑΠΗΡΙΚΟ ΑΜΑΞΙΔΙΟ ΤΗΛΕΧΕΙΡΙΖΟΜΕΝΟ ΑΠΟ ARDUINO»

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΗ: ΡΑΜΑΝΤΖΑΣ ΜΙΧΑΗΛ
ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ: 14084
ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:ΑΣΒΕΣΤΑΣ ΠΑΝΤΕΛΗΣ

ΑΘΗΝΑ 2021

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
Τμήμα Μηχανικών Βιοϊατρικής

Η ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Εξεταστής 1	Εξεταστής 2	Εξεταστής 3
Π. Ασβεστάς, Αναπλ. Καθηγητής, Τμήμα Μηχανικών Βιοϊατρικής, Πανεπιστήμιο Δυτικής Αττικής	**Δ. Γκλώτσος**, Αναπλ. Καθηγητής, Τμήμα Μηχανικών Βιοϊατρικής, Πανεπιστήμιο Δυτικής Αττικής	**Σ. Κωστόπουλος**, Επικ. Καθηγητής, Τμήμα Μηχανικών Βιοϊατρικής, Πανεπιστήμιο Δυτικής Αττικής
Υπογραφή Digitally signed by Panteleimon Asvestas Date: 2021.07.15 15:06:34 +03'00'	Υπογραφή Dimitrios Gklotsos Digitally signed by Dimitrios Gklotsos Date: 2021.07.15 16:29:41 +03'00'	Υπογραφή Digitally signed by Spyridon Kostopoulos

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

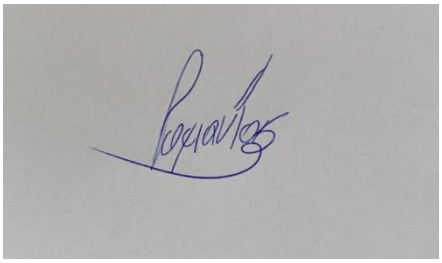
Ο υπογράφων **Ραμαντζάς Μιχαήλ** του **Στυλιανού** με αριθμό μητρώου **48014084** φοιτητής του Τμήματος **Μηχανικών Βιοϊατρικής** της Σχολής **Σχολή Μηχανικών** του Πανεπιστημίου Δυτικής Αττικής, δηλώνω υπεύθυνα ότι:

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

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

Ημερομηνία Ο/Η Δηλών/ούσα

10/7/2021



Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της πτυχιακής μου, Κύριο Παντελή Ασβεστά, ο οποίος με την βοήθεια του με ώθησε να μάθω και να φέρω εις πέρας την παρούσα πτυχιακή εργασία. Ακόμη θα ήθελα να ευχαριστήσω του κάλους μου φίλους και συμφοιτητές Κυριακό Αναστάσιο και Αργύρη Νικόλαο για την στήριξη και την βοήθεια τους. Τέλος, ένα μεγάλο ευχαριστώ στην οικογένεια μου για την ψυχολογική και οικονομική στήριξη που μου παρείχαν.

ΠΕΡΙΛΗΨΗ

Σκοπός της συγκεκριμένης πτυχιακής εργασίας είναι η δημιουργία ενός αναπηρικού αμαξιδίου οπού μπορεί, με την χρήση ενός μοχλού (Joystick) να κινείται ευθεία, δεξιά, πίσω και αριστερά με την βοήθεια 2 ηλεκτροκινητήρων συνεχούς τάσης. Συμπληρωματικά, μπορεί να σταματάει την κίνηση των κινητήρων σε περίπτωση ενός εμποδίου μπροστά η πίσω από το αμάξι με την χρήση αισθητήρων που εκπέμπουν υπέρηχους. Επιπλέον έχει ηχητικό συναγερμό και φωτισμό. Για να επιτευχθεί η συγκεκριμένη κατασκευή, εξηγείται αναλυτικά το Arduino και ο τρόπος λειτουργίας του. Στην συνέχεια γίνεται αναλυτική αναφορά στα εξαρτήματα που χρησιμοποιήθηκαν μαζί με το Arduino και στην μεταξύ τους συνδεσμολογία. Πρόσθετα, γίνεται αναλυτική επεξήγηση στον προγραμματισμού του Arduino αναλύοντας το πρόγραμμα Arduino IDE και αναλυτικά τις εντολές που χρησιμοποιήσαμε. Επιπροσθέτως, αναλύουμε πως συντάσσουμε τον κώδικα του Arduino για το αναπηρικό αμαξίδιο αλλά και τους ελέγχους που πραγματοποιούμε για να βεβαιωθούμε ότι λειτουργεί σωστά. Κλείνοντας, γίνεται αναφορά στα συμπεράσματα και στις μελλοντικές επεκτάσεις όπου μπορούν να γίνουν πάνω στην συγκεκριμένη εφαρμογή.

Λέξεις κλειδιά: Αναπηρικό καροτσάκι, Arduino, ArduinoATMEGA2560, Μικροελεγκτής, Μικροεπεξεργαστής, Κινητήρες συνεχούς τάσης, αισθητήρας υπερήχου(HC-SR04), μοχλός, φωτοδίοδος, Γέφυρα ελέγχου κινητήρα, Arduino IDE,

ABSTRACT

The purpose of this graduate work is to create a wheelchair where, using a joystick, it can move straight, right, back and left with the help of 2 continuous voltage electric motors. In addition, it can be able to stop the engines from moving in the event of an obstacle in front of or behind the car using ultrasound-emitting sensors. In addition, it is having a sound alarm and lighting. In order to achieve this construction, Arduino and the way it works are explained in detail. Detailed reference is then made to the components to be used together with Arduino and to the connection between them. In addition, a detailed explanation is made in Arduino's programming by analyzing the Arduino IDE program and in detail the commands that we used. In addition, we analyze how we draw up the Arduino wheelchair code and the checks we carry out to make sure it works properly. In conclusion, reference is made to the conclusions and future extensions that can be made on this application.

Key Words:

Wheelchair, Arduino, ArduinoATMEGA2560, Microcontroller, Microprocessor, DC motor, Supersonic sensor(HC-SR04), Joystick, Lighting emitting diode (LED), Bridge motor controller, Arduino IDE,

Περιεχόμενα

1 ΕΙΣΑΓΩΓΗ

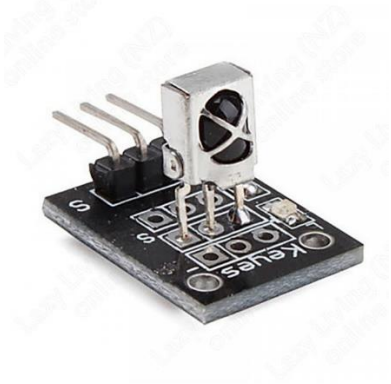
Στις μέρες μας, υπάρχουν άνθρωποι οι οποίοι πάσχουν από αναπηρίες οπού στην καθημερινότητα τους δυσκολεύουν στην κίνηση τους και στις δραστηριότητες τους. Αυτοί οι άνθρωποι, είναι κυρίως εξαρτημένοι από άλλους ανθρώπους για να επιβιώσουν. Ένα από τα βασικότερα εργαλεία που υπάρχουν έτσι να ανεξαρτητοποιηθούν οι άνθρωποι με ειδικές ανάγκες είναι το αναπηρικό καροτσάκι. Το αναπηρικό καροτσάκι είναι, μια καρεκλά με ροδές δεξιά και αριστερά, οπού χορηγείται σε κάποιον λόγο κάποιας ασθένειας, αναπηρίας ή τραυματισμού με σκοπό την εύκολη μετακίνηση με την βοήθεια κάποιου αλλού ανθρώπου στον χώρο. Ακόμη υπάρχουν καροτσάκια οπού ο ασθενής μπορεί να μετακινηθεί μονός του σπρώχνοντας τους τροχούς με τα χεριά του. Υπάρχουν όμως και περιπτώσεις που υπάρχουν άνθρωποι οπού δεν μπορούν να σπρώξουν τους τροχούς λόγο αδυναμίας στα χεριά η λόγο ασθένειας. Για αυτόν τον λόγο υπάρχουν τα ΄Έξυπνα¨ αναπηρικά καροτσάκια. Το έξυπνο αναπηρικό καροτσάκι είναι ένα αναπηρικό καροτσάκι το οποίο κινείται με ρεύμα. Έχει μια μπαταριά και ένα ισχυρό κινητήρα με σκοπό την κίνηση του ΑΜΕΑ με την χρήση ενός μοχλού και με πολύ εύκολο τρόπο και χωρίς να κουράζεται. Ανάλογα το αναπηρικό καροτσάκι, υπάρχουν πολλά επιπλέον ηλεκτρονικά εξαρτήματα όπως μικροϋπολογιστές και αισθητήρες οπού βοηθούν την καθημερινότητα και του ΑΜΕΑ αλλά και των γύρω του που τον προσέχουν.

ΕΙΚΟΝΑ 1 ΑΠΛΟ ΑΝΑΠΗΡΙΚΟ ΑΜΑΞΙΔΙΟ / ΕΞΥΠΝΟ ΑΝΑΠΗΡΙΚΟ ΑΜΑΞΙΔΙΟ

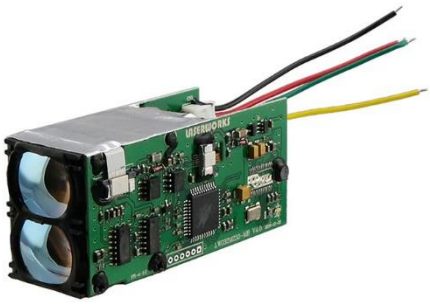
Τα πρώτα έξυπνα αναπηρικά καροτσάκια ξεκίνησαν το 1992 στην Ιαπωνία (NEC CORP) οπού με αισθητήρες IR μπορούσαν να σταματήσουν άμα έβλεπαν κάποιο εμπόδιο μπροστά τους. Το 1996 στην Αλγερία (CHARHM CDTA) φτιάχτηκε αναπηρικό καροτσάκι οπού μπορούσε να κινηθεί αυτόνομα μέσα στο κτίριο με την βοήθεια ψηφιακού χάρτη. Μεγάλη σημαντική εξέλιξη υπήρξε όταν το 1998 έως και 2003 στην Ιαπωνία(OSAKA UNIVERSITY) κατάφεραν με δυο κάμερες που κοιτούσαν τον ασθενή μπορούσαν ανάλογα την κίνηση του κεφαλιού του ΑΜΕΑ να μπορεί να κινηθεί ανάλογα το καροτσάκι.

Με την πάροδο του χρόνου υπήρξαν πάρα πολλοί τρόποι με τους οποίους έπαιρνε εντολή το αναπηρικό αμαξιού. Τηλεχειριστήριο με μοχλό, με πνευματικούς διακόπτες, με φωνητικές εντολές (με την βοήθεια αναγνώρισης φωνής) και σε πιο ακριβούς μεθόδους όπως με την βοήθεια Ηλεκτροκοκκογραφίας.

Για την πιο άνετη και ασφαλέστερη καθημερινότητα των ΑΜΕΑ τοποθετούνται διάφοροι αισθητήρες στον αναπηρικό καροτσάκι. Με μεγάλη διαφορά οι καλύτεροι αισθητήρες για την αποφυγή τον εμποδίου είναι αυτοί του υπέρηχου(ΕΙΚΟΝΑ 4). Είναι οικονομικοί και έχουν μεγάλη ακρίβεια. Άλλο ένας αισθητήρας που χρησιμοποιήθηκε για την αποφυγή των εμποδίων είναι ένας αισθητήρας IR infrared(ΕΙΚΟΝΑ 2). Εκπέμπει φως (υπέρυθρο) αντί για υπέρηχους αλλά αντιμετωπίζει πρόβλημα με της επιφάνειες που απορροφάνε φως και παίρνει λάθος αποτελέσματα ο αισθητήρας. Ακόμα ένας τρόπος είναι με LRF (ΕΙΚΟΝΑ 3) η αλλιώς Laser ακτίνες οπού υπολογίζουν αποστάσεις. Παρέχει απολυτή ακρίβεια αλλά έχουν μεγάλη κατανάλωση σε ισχύ με αποτέλεσμα να μην αντέχουν οι μπαταρίες.



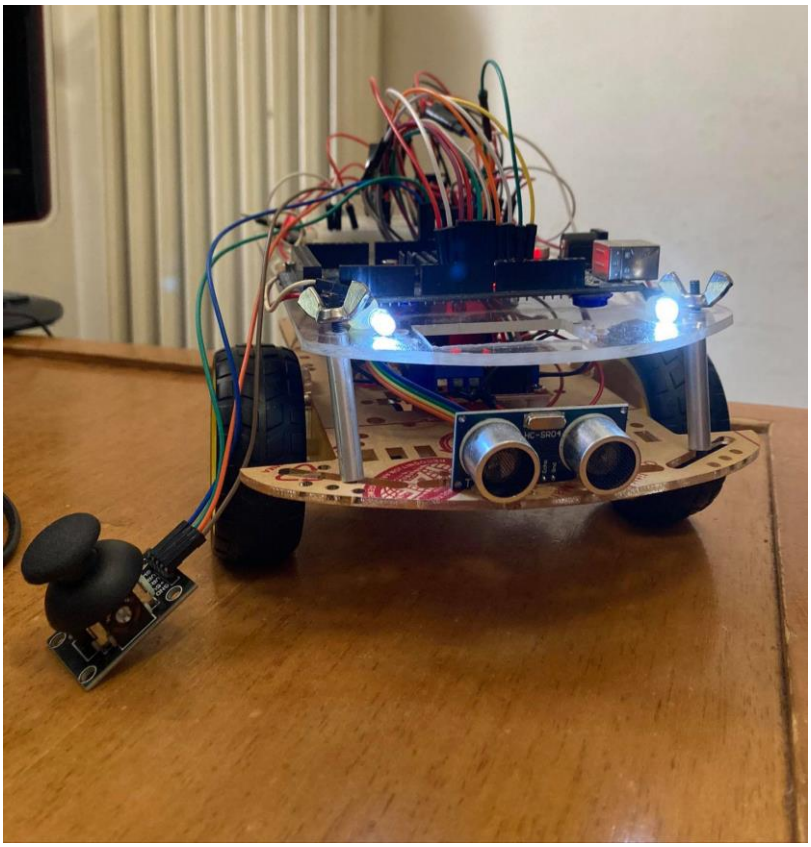
ΕΙΚΟΝΑ 2 ΑΙΣΘΗΤΗΡΕΣ INFRARED (IR)



ΕΙΚΟΝΑ 3 ΑΙΣΘΗΤΗΡΑΣ LRF

ΕΙΚΟΝΑ 4 ΑΙΣΘΗΤΗΡΑΣ HC-SR04

Στην δική μας εφαρμογή θα χρησιμοποιήσουμε για την αποφυγή εμποδίων αισθητήρες υπέρηχου και για να δίνουμε εντολές στο αναπηρικό αμάξι θα χρησιμοποιήσουμε ένα μοχλό η αλλιώς Joystick ώστε να ελέγχουμε την κίνηση. Θα προσθέσουμε επιπλέον ένα ηχείο για συναγερμό και LED για να υπάρχει φωτισμός μπρος πίσω.



ΕΙΚΟΝΑ 5 ΟΛΟΚΛΗΡΩΜΕΝΟ ΤΟ ΑΝΑΠΗΡΙΚΟ ΑΜΑΞΙΟ

2 HARDWARE – ΥΛΙΚΑ

Σε αυτό το κεφάλαιο θα αναφερθούμε στο Arduino και στα πλεονεκτήματα του .Ακόμη, θα μιλήσουμε αναλυτικά για τα εξαρτήματα και την λειτουργία του καθενός ξεχωριστά. Επιπλέον θα γίνει αναφορά στην συνδεσμολογία και στον τρόπο λειτουργίας του αναπηρικού αμαξιού.

2.1 Μικροελεγκτής Arduino

Σύμφωνα με την επίσημη ιστοσελίδα του Arduino, είναι μια πλατφόρμα ηλεκτρονικών, ανοιχτού κώδικα με εύκολο στην χρήση εξοπλισμό τόσο υλικό όσο και λογισμικό. Με το Arduino μπορούμε να κατασκευάσουμε σχεδόν τα πάντα. Από ένα μικρό θερμόμετρο έως και ένα αγωνιστικού η κινηματογραφικού τύπου Drone. Το Arduino κατασκευαστικέ με σκοπό να μπορεί να είναι εύκολο και να μπορούν όλοι να το χρησιμοποιήσουν. Από φοιτητές, ερασιτέχνες που ψάχνουν ένα νέο χόμπι έως και επαγγελματίες προγραμματιστές που θέλουν να κατασκευάσουν πιο δύσκολα project. Υπάρχουν πάρα πολλές παραλλαγές του Arduino. Κάποια είναι πιο μικρά σε μέγεθος με λιγότερες εισόδους/εξόδους και κάποια είναι πολύ μεγαλύτερα με πολλές περισσότερες εισόδους/εξόδους που μπορούν ακόμα να πάρουν και προεκτάσεις τα λεγόμενα Shields. Τα μεγάλα πλεονεκτήματα του Arduino είναι ότι

1) Είναι οικονομικό και μπορούν να το αγοράσουν όλοι

2) Μπορεί να λειτουργήσει σε όλα λειτουργικά συστήματα υπολογιστών (windows,Mac, Linux κλπ.)

3) Είναι ανοιχτού κώδικα λογισμικού και μπορεί να εμπλουτισθεί με τμήματα από άλλες γλώσσες προγραμματισμού

4) Μπορεί να χρησιμοποιήσει κάποιος οτιδήποτε εξαρτήματα θέλει.

5) Δωρεάν το πρόγραμμα Arduino IDE που είναι υπεύθυνο για τον προγραμματισμό.

Επειδή το Arduino είναι ο πιο διαδεδομένος μικροελεγκτής και σε συνδυασμό ότι είναι ανοικτού κώδικα. Μπορεί κάποιος να βρει έτοιμο τον κώδικα για την εφαρμογή που θέλει η ακόμα και να πάρει ιδέες να φτιάξει την δικιά του εφαρμογή. Ακόμη υπάρχει η δυνατότητα να επεξεργαστείς ένα ήδη έτοιμο πρόγραμμα έτσι όπως το θέλεις. Για παράδειγμα, στο διαδίκτυο υπάρχει δωρεάν ο κώδικας για την κατασκευή ενός 3D printer. Το μόνο που πρέπει να κάνει κάποιος είναι να αγοράσει τα εξαρτήματα, να το συναρμολογήσει και απλά να περάσει τον κώδικα.

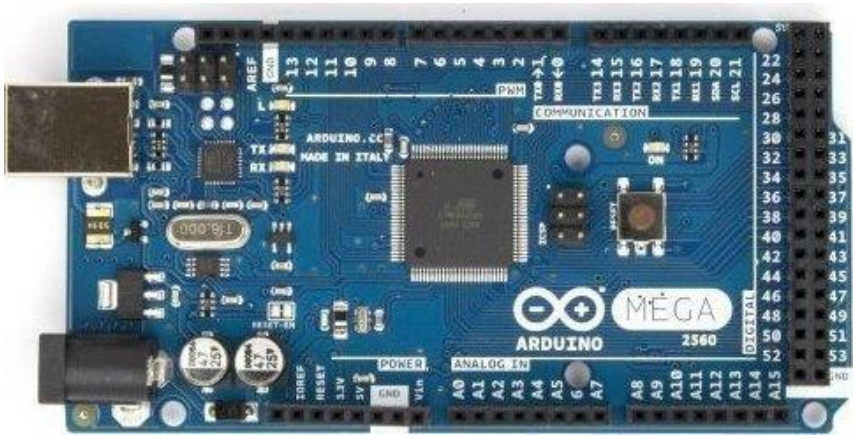
Υπάρχουν απεριόριστες δυνατότητες με το Arduino και άπειρες εφαρμογές που μπορεί κάνεις να βρει στο διαδίκτυο, είτε είναι ερασιτέχνης είτε επαγγελματίας.

2.2 Υλικά – Εξαρτήματα Αμαξιού

2.2.1 Arduino Mega 2560

Το Arduino Mega 2560 είναι ένας μικροελεγκτής βασισμένος στον ATMega2560 (ΕΙΚΟΝΑ 6). Περιλαμβάνει 54 ψηφιακές εισόδους και εξόδους όπου οι 14 μπορούν να χρησιμοποιηθούν και ως έξοδοι PWM. Έχει 16 αναλογικές εισόδους, 4 UARTs (σειριακές πύλες), σύνδεση με USB, ένα ακροδέκτη τροφοδοσίας και ένα κουμπί

επανεκκίνησης RESET. Το Arduino Mega 2560 είναι ικανό να πραγματοποιήσει άπειρες εφαρμογές καθώς υποστηρίζει το μεγαλύτερο ποσοστό από επεκτάσεις που βρίσκονται στο εμπόριο. [1]



ΕΙΚΟΝΑ 6 ARDUINO MEGA AT2560

Πίνακας 1. ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ARDUINO MEGA 2560

Μικροελεγκτής	Atmega2560
Τάση λειτουργίας	5V
Τάση λειτουργίας τροφοδοσίας	7-12V
Όρια τάσης εξωτερικής τροφοδοσίας	6-20V
Μέγιστο συνεχές ρεύμα ανά ακροδέκτη εισόδου/εξόδου Μμέγιστό ρεύμα για τον ακροδέκτη 3,3V	40Ma 50mA
SRAM EEPROM	8KB 4KB
Ταχύτητα ρολογιού	16MHZ
Flash Memory	256KB (8KB χρησιμοποιούνται από τον BootLoader)

2.2.2 HC-SR04 Αισθητήρα Υπερήχων

Ο αισθητήρα HC-SR04 χρησιμοποιείται για την μέτρηση απόστασης (ΕΙΚΟΝΑ 7). Διαθέτει ένα πομπό και ένα δέκτη. Έχει 4 ακροδέκτες Vcc(Τροφοδοσίας),

GND(Γείωσης), Trig(Ακροδέκτης ενεργοποίησης), Echo(Ακροδέκτης εξόδου). Μπορεί να μετρήσει αποστάσεις από 2cm έως και 400cm με ακρίβεια 0,3cm.

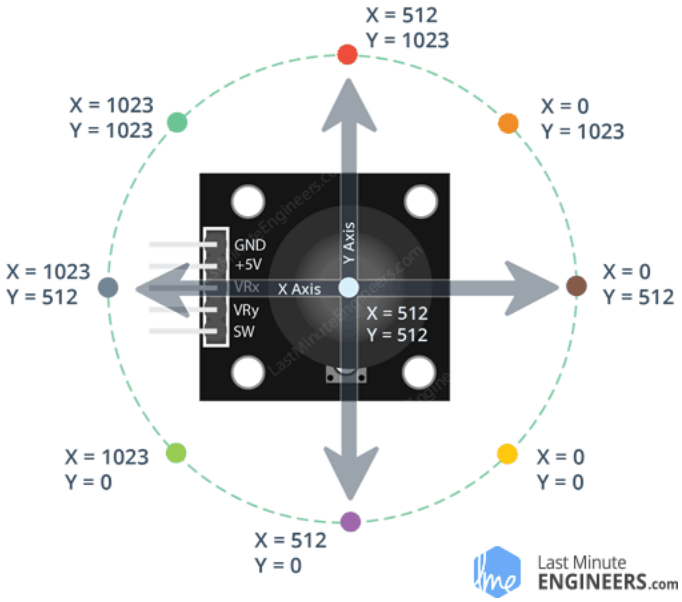
Ο HC-SR04 (ΕΙΚΟΝΑ 7) λειτουργεί με υπερήχους. Για να ξεκινήσει την μέτρηση ο αισθητήρας πρέπει να λάβει ένα παλμό στον ακροδέκτη Trig. Με το που λάβει το παλμό στέλνει μια ακολουθία υπερήχων οι οποίοι ταξιδεύουν στον χώρο , ανακλώνται στο πρώτο αντικείμενο που συναντάνε και επιστρέφουν στο αισθητήρα. Όταν ο δέκτης λάβει την αντανάκλαση τότε στον ακροδέκτη Echo αποστέλλεται ένας παλμός HIGH. Η διάρκεια του παλμού αυτού είναι ίδια με τον χρόνο που χρειάστηκε ο υπέρηχος να αποσταλεί και να ληφθεί από τον αισθητήρα. Συνεπώς, έχουμε αρκετά δεδομένα για την απόσταση του αντικειμένου από τον αισθητήρα. [1]

ΕΙΚΟΝΑ 7 ΑΙΣΘΗΤΗΡΑΣ HC-SR04

2.2.3 Joystick – Μοχλός

Σκοπός του μοχλού (ΕΙΚΟΝΑ 33) είναι να μεταφέρει την κίνηση του στίκ σε 2 άξονες x και y στο Arduino. Για να επιτευχθεί αυτό χρησιμοποιούνται δύο ξεχωριστά ποτενσιόμετρα 10KΩ όπου χρησιμοποιούνται ως διπλός διαιρέτης τάσης.

Για την μέτρηση της θέσης του μοχλού πρέπει, να μετρήσουμε την αντίσταση στην φυσική θέση του μοχλού. Γνωρίζοντας αυτό, ανάλογα την αλλαγή στην αντίσταση το Arduino μπορεί να καταλάβει την αλλαγή του μοχλού. [2]



ΔΙΑΓΡΑΜΜΑ ΜΟΧΛΟΥ/JOYSTICK

2.2.4 LED

Lighting Emitting Diode ή Αλλιώς δίοδος LED (ΕΙΚΟΝΑ 8) είναι ένας ημιαγωγός όπου εάν πολωθεί ορθά φωτοβολεί. Αποτελείται από δύο ακροδέκτες + - που όταν συνδεθούν σε μια πηγή (ορθά) εκπέμπει φως. [1]

ΕΙΚΟΝΑ 8 LED

2.2.5 Buzzer

Το Buzzer (ΕΙΚΟΝΑ 9) η αλλιώς ο βομβητής είναι μια συσκευή παράγει ήχο. Υπάρχουν δυο τύποι από Buzzer. Ο ενεργητικός τύπος (Active) και ο παθητικός τύπος (Passive). Το ενεργητικό Buzzer έχει μέσα ενσωματωμένο ταλαντωτή. Αυτό σημαίνει ότι με το που δεχθεί τροφοδοσία στα άκρα του ξεκινάει και παράγει ήχο. Επειδή έχει ενσωματωμένο ταλαντωτή είναι πιο ακριβό το ενεργητικό από το παθητικό. Από την άλλη μεριά, το παθητικό είναι πιο οικονομικό επειδή δεν έχει ενσωματωμένο ταλαντωτή. Αυτό έχει ως αποτέλεσμα να είναι ποιο δύσκολο στο προγραμματισμό. Εάν εφαρμόσουμε τροφοδοσία στα άκρα του παθητικού Buzzer τότε δεν θα ακούσουμε κάποιον ήχο. Για να δουλέψει το Buzzer και να το ακούσουμε θα πρέπει να του δώσουμε ένα τετραγωνικό παλμό με έναν ξεχωριστό ταλαντωτή ή

δίνοντας του παλμό από το Arduino. Η συχνότητα που χρειάζεται για να λειτουργήσει ένα Buzzer παθητικού τύπου είναι από 2KHz – KHz

ΕΙΚΟΝΑ 9 ACTIVE BUZZER/PASSIVE BUZZER

Στην δική μας εφαρμογή, χρησιμοποιήσαμε παθητικού τύπου Buzzer. Του παρείχαμε τετραγωνικό παλμό από το Arduino και χρησιμοποιώντας την εντολή tone/notone το ενεργοποιούσαμε η το απενεργοποιούσαμε αντίστοιχα. [3]

2.2.6 Μοτέρ Κίνησης Αμαξιού

Το Μοτέρ η αλλιώς ο Κινητήρας συνεχομένης τάσης είναι ένα εξάρτημα μετατροπής ηλεκτρικής ενέργειας σε μηχανική ενέργεια. Για να ενεργοποιήσουμε το μοτέρ, εφαρμόζουμε τροφοδοσία στα δυο άκρα του μοτέρ και ξεκινάει να περιστρέφεται. Εάν αλλάξουμε την πολικότητα της τάσης που εφαρμόσαμε τότε ο κινητήρας αρχίζει να περιστρέφεται αντίθετα. [4] [5]

Υπάρχουν 3 κατηγορίες βασικών κινητήρων:

1)Dc Motor: Κινητήρας ο οποίος μόλις εφαρμόσουμε τροφοδοσία τότε ενεργοποιείται, για να αλλάξουμε περιστροφή, αλλάζουμε πολικότητα.

ΕΙΚΟΝΑ 10 ΚΙΝΗΤΗΡΑΣ ΣΥΝΕΧΟΥΣ ΤΑΣΗΣ DC

2)Servo motor: Πολύ πιο ακριβά από τα απλά DC μοτέρ. Έχουν απολυτή ακρίβεια στην κίνηση τους γνωρίζοντας γραμμικά την θέση τους .

ΕΙΚΟΝΑ 11 ΚΙΝΗΤΗΡΑΣ SERVO

3)Stepper motor: Βημάτικα μοτέρ, Συγκεκριμένες οι κινήσεις τους λειτουργούν με τετραγωνικούς παλμούς. Πολύ ισχυρά σε δύναμη μοτέρ. Ιδανικά για τους 3d Εκτυπωτές.

ΕΙΚΟΝΑ 12 ΚΙΝΗΤΗΡΑΣ ΒΗΜΑΤΙΚΟΣ (STEP)

2.2.7 Bridge Module L298N – Γέφυρα ελέγχου κινητήρων

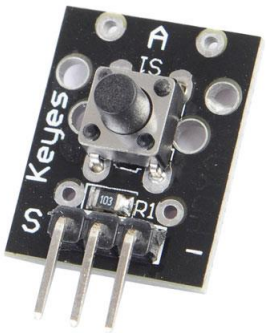
Στο προηγούμενο κεφάλαιο μιλήσαμε για τους κινητήρες. Επειδή όλα τα είδη κινητήρων έχουν μεγάλη κατανάλωση ρεύματος είναι επικίνδυνο να κάψει το Arduino. Για αυτόν τον λόγο, οι κινητήρες συνδέονται σε μια γέφυρα με εξωτερική τροφοδοσία. Συνεπώς το Arduino έχει την ικανότητα να ελέγχει τους κινητήρες χωρίς να κινδυνεύει ούτε να καταπονείτε. Μέσω της γέφυρας μπορούμε να ελέγχουμε την ταχύτητα και την φορά περιστροφής. [6]

ΕΙΚΟΝΑ 13 ΓΕΦΥΡΑ L298N

Το L298N (ΕΙΚΟΝΑ 13) είναι ένα από τα πιο διάσημα εξαρτήματα στον χώρο της ρομποτική. Μπορεί να ελέγξει δυο μοτέρ και να τα υποστηρίξει με εξωτερική τροφοδοσία. Μπορεί να δεχθεί τροφοδοσία DC από 5V έως και 35V και μέγιστο ρεύμα έως και 3A Έχει την ικανότητα να παράλληλα με τα μοτέρ να τροφοδοτεί και το Arduino. Με τους 4 ακροδέκτες εισόδου μπορεί να γίνει ο έλεγχος των μοτέρ. Συνδέουμε τους ακροδέκτες της γέφυρας με 4 από τους ακροδέκτες I/O του Arduino(Digital I/O).

2.2.8 Διακόπτες – Push Button

Ο διακόπτης τύπου Push-Button είναι ένα εξάρτημα 2 σημείων που ενώνουν ένα κύκλωμα κατά την ενεργοποίηση του. [7]



ΕΙΚΟΝΑ 14 PUSH BUTTON

2.2.9 Μπαταριοθήκη – Πηγή τροφοδοσίας

Για να λειτουργήσει το κύκλωμα μας, χρειάζεται τροφοδοσία. Επειδή η συγκεκριμένη εφαρμογή χρειάζεται φορητότητα. Για να το πέτυχουμε χρησιμοποιήσαμε μια Μπαταριοθήκη. Η Μπαταριοθήκη, είναι ένα εξάρτημα στο οποίο μπορούμε να

βάλουμε μια και περισσότερες μπαταρίες ιδίου τύπου έτσι ώστε να αυξήσουμε την συνολική τάση.

ΕΙΚΟΝΑ 15 ΜΠΑΤΑΡΙΟΘΗΚΗ 4 ΘΕΣΕΩΝ ΑΑ

Υπάρχουν πάρα πολλές παραλλαγές για το συγκεκριμένο εξάρτημα. Υπάρχουν στην αγορά μπαταριοθήκες 2,4,6 θέσεων για όλες τις κατηγορίες μπαταριών ανάλογα την εφαρμογή που θέλουμε να υλοποιήσουμε. Στην δική μας εφαρμογή χρησιμοποιήσαμε Μπαταριοθήκη η οποίος παίρνει 6 μπαταρίες τύπου ΑΑ και συνολικής τάσης 9V.

2.3 Διάγραμμα και συνδεσμολογία

Σε αυτό το κεφάλαιο θα μιλήσουμε για την συνδεσμολογία των εξαρτημάτων του αναπηρικού αμαξιού.

Για την κατασκευή του αναπηρικού αμαξιού θα χρειαστούμε τα παρακάτω:

1) 1 Arduino Mega

2) 2 DC κινητήρες

3) 1 περιστρεφόμενο ροδάκι με βάση

4) 1 γέφυρα L298N

5) 1 Μπαταριοθήκη 6xAA

6) 2 LED λευκά

7) 2 LED κόκκινα

8) 4 αντιστάσεις

9) 2 αισθητήρες HC-SR04

10) 2 διακόπτες ON/OFF

11) 1 Joystick μοχλό X/Y

12) 1 Push/Button Διακόπτη

13) 1 Buzzer ηχείο

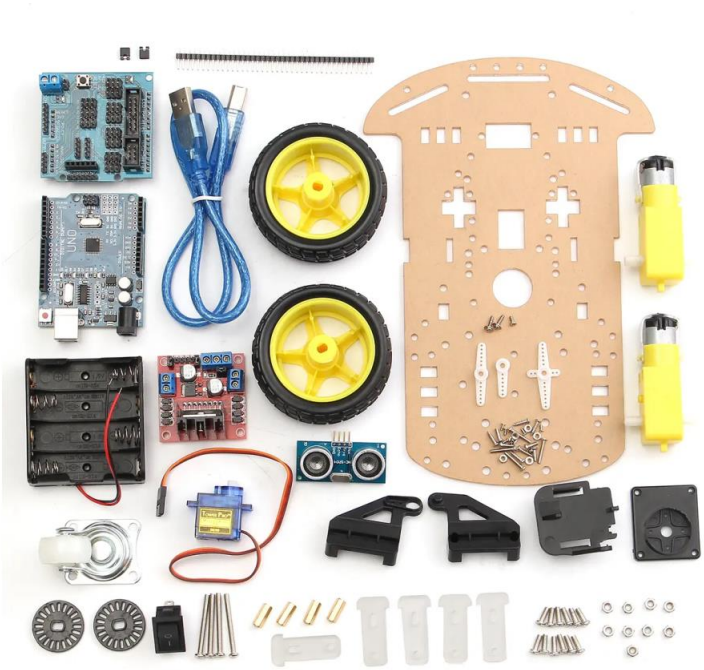
14) 1 Breadboard mini 4.5cm x 8cm

15) 2 Βάσεις από πλεξιγκλάς

16) Καλώδια Jumpers

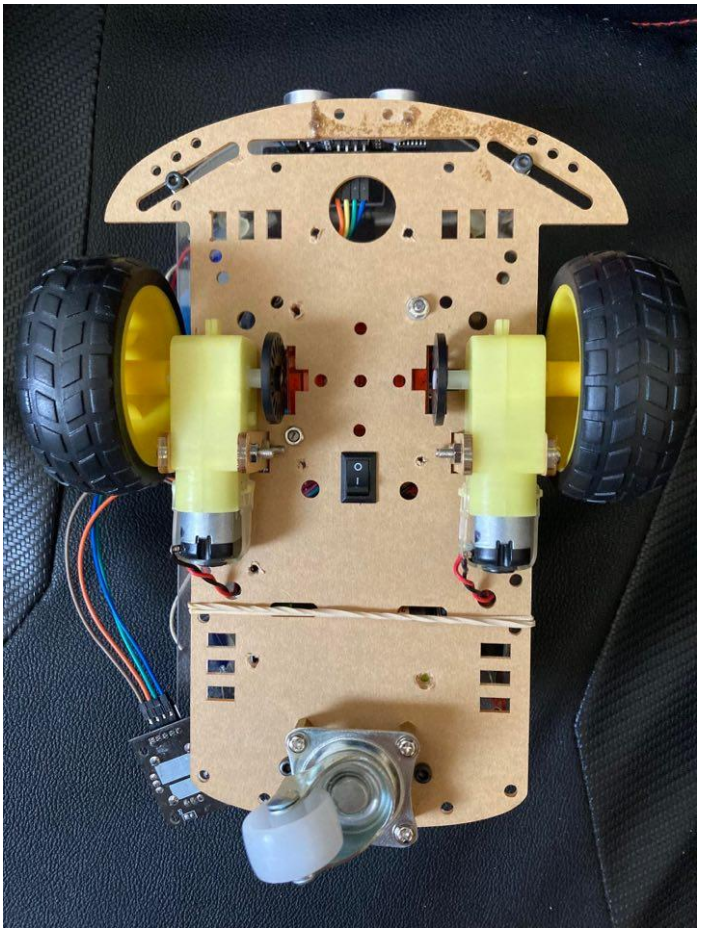
17) Αποστάτες, Βίδες και πεταλούδες

Τα πρώτα εξαρτήματα που θα χρησιμοποιήσουμε φαίνονται στην παρακάτω φωτογραφία.



ΕΙΚΟΝΑ 16 ΚΙΤ ΣΥΝΑΡΜΟΛΟΓΗΣΗΣ ARDUINO CAR RC

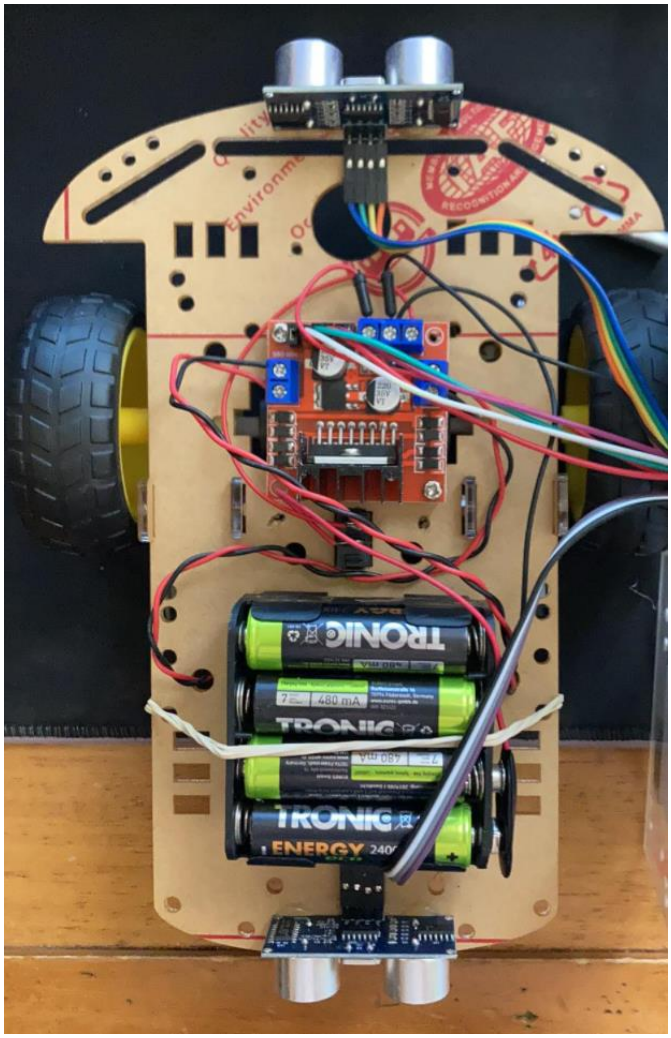
Στην παραπάνω εικόνα φαίνεται το πρώτο κιτ που αγοράσαμε για την κατασκευή μας. Το πρώτο πράγμα που πρέπει να κάνουμε, περνώντας ένα κομμάτι πλεξιγκλάς σχεδιάζουμε και κόβουμε ένα αντίγραφο της καφέ βάσης, θα το χρειαστούμε στην συνέχεια. Το επόμενο βήμα είναι να βιδώσουμε τους κινητήρες πάνω στην καφέ βάση μην την βοήθεια των πλαστικών βάσεων που περιλαμβάνονται στην συσκευασία. Εφόσον βιδώσουμε τα μοτέρ βιδώνουμε στο κάτω μέρος το περιστρεφόμενο ροδάκι με την ασημένια βάση. Έχοντας βιδώσει λοιπόν τα παραπάνω εξαρτήματα τοποθετούμε στο κέντρο και τον διακόπτη On/off. Το τελικό μας αποτέλεσμα θα φαίνεται κάπως έτσι: (ΕΙΚΟΝΑ 17)



ΕΙΚΟΝΑ 17 ΚΑΤΩ ΟΨΗ ΑΜΑΞΙΟΥ

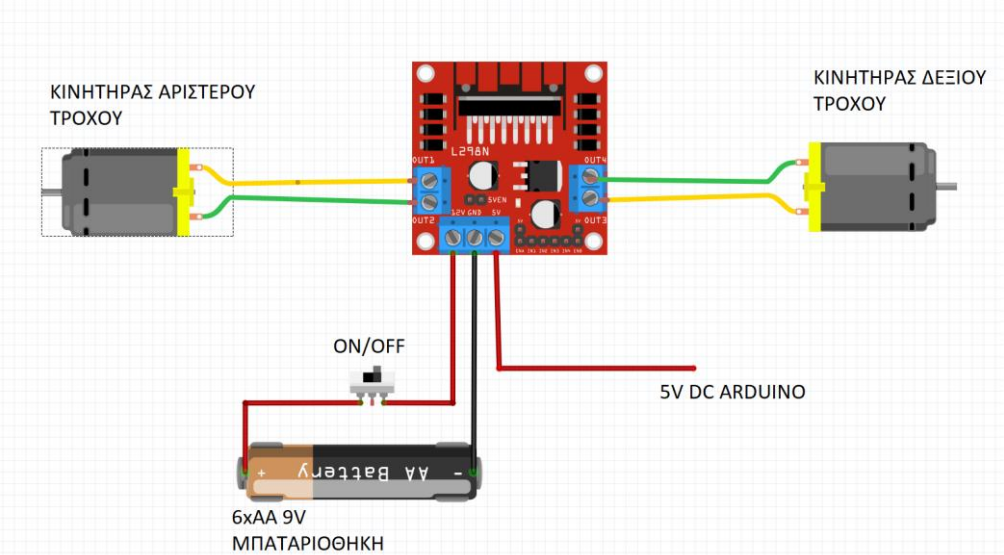
Για να μπορέσει το περιστρεφόμενο ροδάκι να είναι στο ίδιο ύψος μαζί με τους μεγαλύτερους τροχούς, χρησιμοποιήσαμε τους 4 επιχρυσωμένους αποστάτες. Στην παραπάνω εικόνα βλέπουμε την κάτω μεριά της κατασκευής μας.

Στην άλλη μεριά της βάσης θα πρέπει να τοποθετήσουμε την Μπαταριοθήκη, την γέφυρα (L298N), τους δυο αισθητήρες HC-SR04 και δυο μικρούς αποστάτες. Για αρχή με την βοήθεια των 2 αποστατών τοποθετούμε την γέφυρα με δυο βίδες διαγώνια με το μέρος που βρίσκεται η ψήκτρα προς τα κάτω. Λίγο πιο κάτω υπάρχουν οι ακροδέκτες του διακόπτη. Κάτω από τους ακροδέκτες του διακόπτη, τοποθετούμε την Μπαταριοθήκη με τις 6 μπαταρίες. Έχοντας τοποθετήσει όλα τα προηγούμενα το τελευταίο που πρέπει να τοποθετήσουμε είναι οι δυο αισθητήρες Hc-Sr04. Είναι σημαντικό πριν τοποθετήσουμε τους αισθητήρες, να ενώσουμε τους 4 ακροδέκτες με καλώδια Jumber έτσι να είναι πιο εύκολο μετρά να τα συνδέσουμε με το Arduino. Χρησιμοποιώντας το πιστόλι θέρμης κόλας(Glue Gun) Τοποθετούμε τον ένα αισθητήρα πάνω κοιτώντας προς τα έξω και τον άλλο αισθητήρα κάτω κοιτώντας πάλι προς τα έξω. Έχοντας κάνει όλα τα παραπάνω θα πρέπει να έχουμε αυτό το αποτέλεσμα:



ΕΙΚΟΝΑ 18 ΠΑΝΩ ΟΨΗ

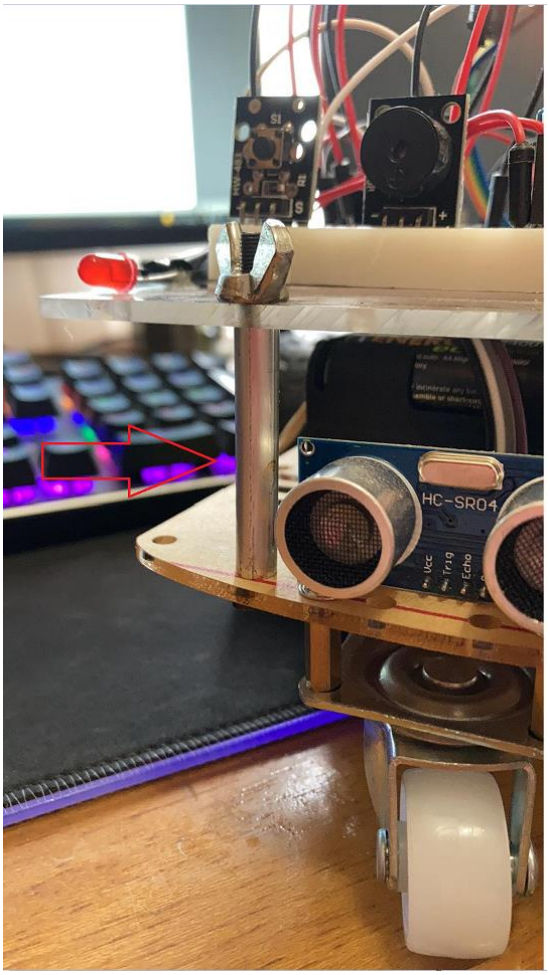
Εφόσον έχουν τοποθετηθεί τα εξαρτήματα όπως στην παραπάνω εικόνα, με την βοήθεια του Fritzing θα δούμε αναλυτικά την συνδεσμολογία των καλωδίων.



ΣΧΕΔΙΑΓΡΑΜΜΑ 1 ΚΙΝΗΤΗΡΕΣ ΜΕ ΓΕΦΥΡΑ L298N

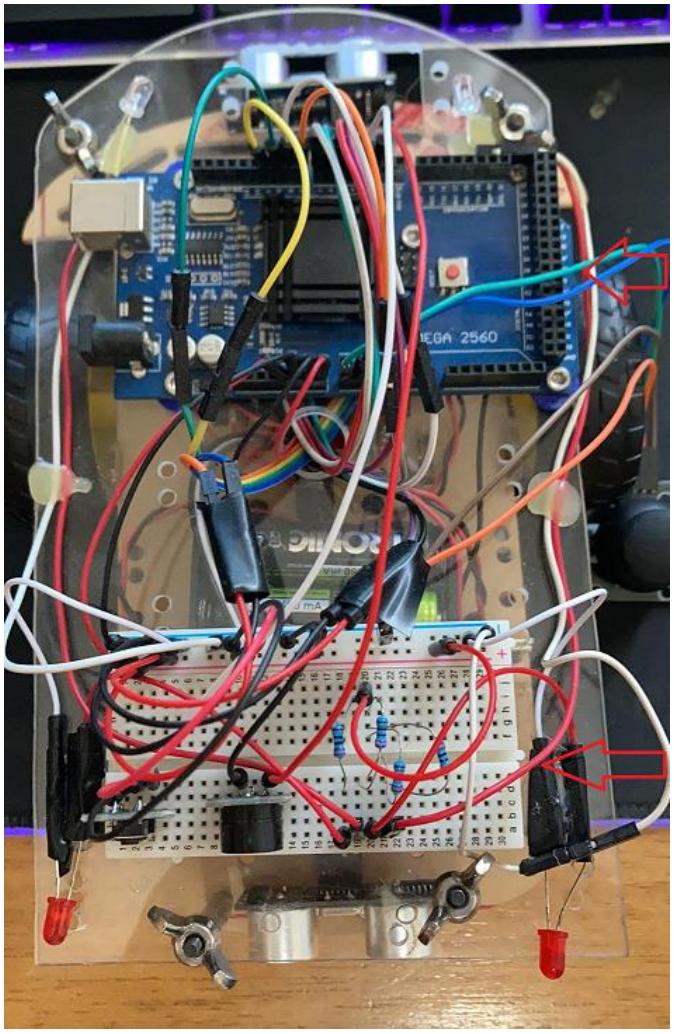
Στην παραπάνω εικόνα, παρατηρούμε ότι έχουμε συνδέσει τους 2 κινητήρες και την πηγή(Μπαταριοθήκη). Το L298N μπορεί να πάρει τάση έως 12V DC και έχει την ικανότητα να μας δώσει έξοδο 5VDC έτσι ώστε να τροφοδοτήσουμε το Arduino. Συνεπώς, βιδώσαμε ένα καλώδιο jumper στην έξοδο των 5V έτσι ώστε να δώσουμε τροφοδοσία στο Arduino αργότερα. Επίσης βιδώσαμε ακόμα ένα Jumper στον Gnd έτσι ώστε να συνδέσουμε την γείωση του Arduino. Ακόμη βιδώσαμε 4 jumper στους ακροδέκτες in1,in2,in3,in4. Με τους 4 αυτούς ακροδέκτες γίνεται ο έλεγχος των κινητήρων μέσο του Arduino. Πχ εάν in1=1 και in2=0 τότε ο αριστερά κινητήρας κινητέ μπροστά.

Έχοντας ολοκληρώσει και αυτό το κομμάτι αναπηρικού αμαξιού, είμαστε έτοιμοι να περάσουμε στο πιο πάνω τμήμα της κατασκευής μας που αφορά το Arduino και το Breadboard. Έχοντας λοιπόν κατασκευάσει 4 αποστάτες από σωληνάκια των 3,5cm και παίρνοντας τις ανάλογες βίδες και πεταλούδες μπορούμε να βιδώσουμε και την πάνω βάση.



ΕΙΚΟΝΑ 19 ΠΡΟΣΟΨΗ ΑΝΑΠΗΡΙΚΟΥ ΑΜΑΞΙΟΥ

Για να φτιάξουμε την πάνω βάση πήραμε το περίγραμμα από την καφέ βάση και κόψαμε ένα πλεξιγκλάς στις ιδίες διαστάσεις και δημιουργήσαμε οπές στο κέντρο, μπροστά και πίσω έτσι να περάσουν όλα τα καλώδια που χρειαζόμαστε από το κάτω τμήμα. Στην συνέχεια ανοίγουμε 4 οπές ώστε να βιδώσουμε την βάση πάνω από τους αποστάτες. Εφόσον βιδώσαμε την πάνω βάση μπορούμε με μία διπλής όψεως ταινία να κολλήσουμε το Breadboard(4,5cmX8cm) στο κάτω μέρος της βάσης. Με την χρήση 2 ακόμα αποστατών βιδώνουμε το Arduino Mega στο πάνω μέρος της βάσης. Έχοντας κάνει τα παραπάνω το αποτέλεσμα θα πρέπει να είναι ως εξής:



ΕΙΚΟΝΑ 20 ΠΑΝΩ ΟΨΗ ΑΜΑΞΙΟΥ (ΟΛΟΚΛΗΡΩΜΕΝΗ)

Έχοντας λοιπόν τελειώσει με την τοποθέτηση του Arduino Mega και του Breadboard, θα μιλήσουμε αναλυτικά για την συνδεσμολογία των καλωδίων και εξαρτημάτων.

Αρχικά έχοντας περάσει όλα τα καλώδια που έρχονται από το κάτω τμήμα, από την κεντρική οπή θα πρέπει να έχουμε τα εξής καλώδια στον αέρα:

4 ακροδέκτες HC-SR04 ΜΠΡΟΣΤΑ

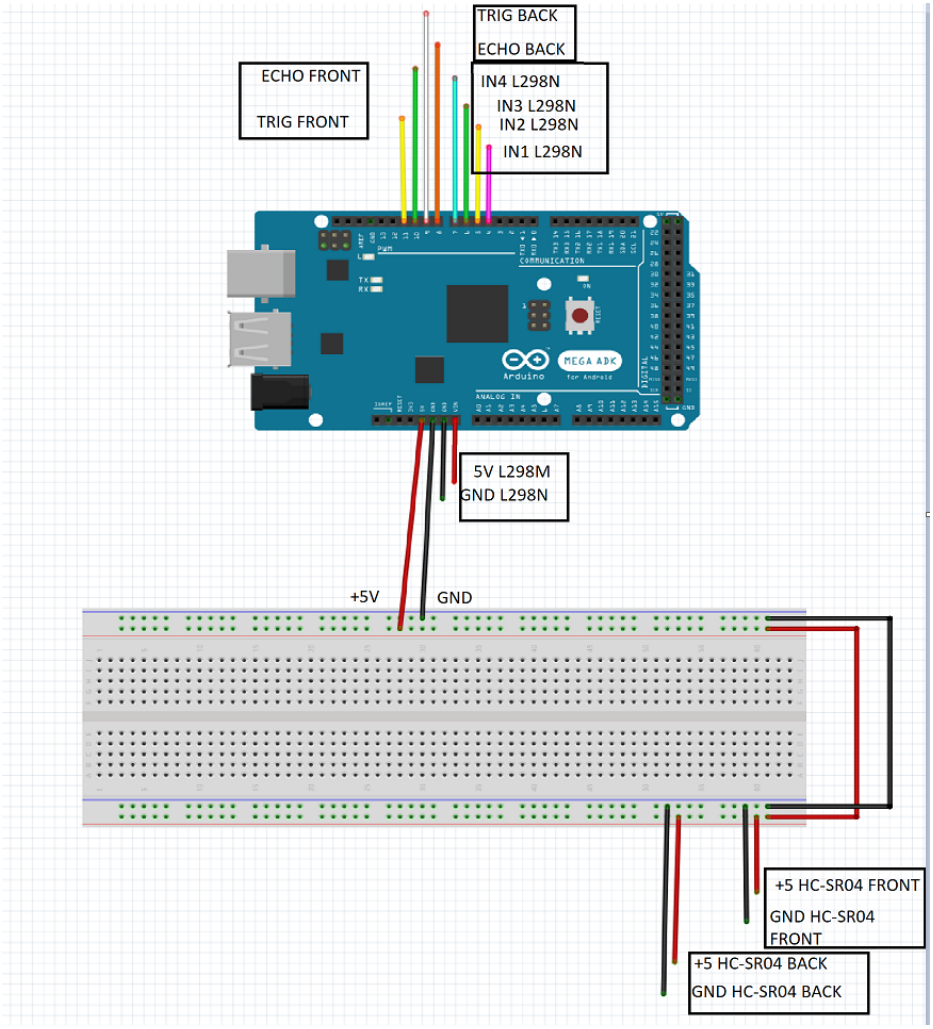
4 ακροδέκτες HC-SR04 ΠΙΣΩ

1 ακροδέκτης 5V DC

1 ακροδέκτης GND (L298N)

4 ακροδέκτες in1, in2, in3, in4

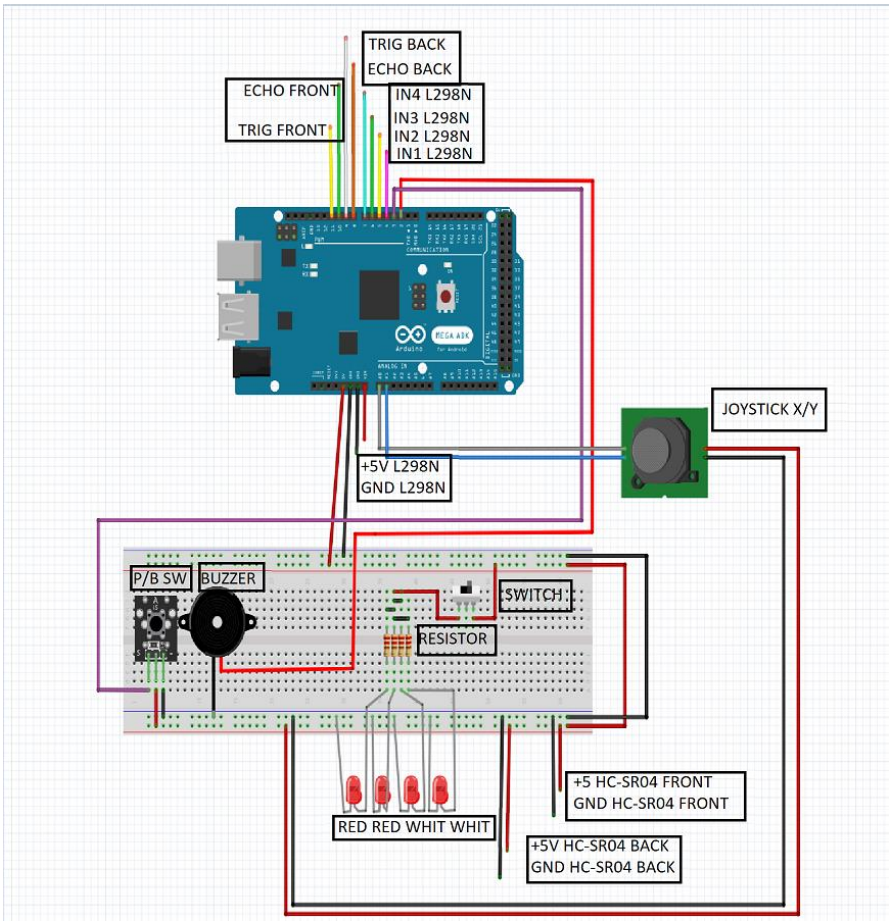
Έχοντας λοιπόν και τους 14 παραπάνω ακροδέκτες προκύπτει το παρακάτω διάγραμμα:



ΣΧΕΔΙΑΓΡΑΜΜΑ 2 ARDUINO ΚΑΙ BREADBOARD

Αρχικά συνδέσαμε την τροφοδοσία των 5V και την γείωση GND στο Arduino Mega. Στην συνέχεια πήραμε την έξοδο των 5V και την γείωση και την συνδέσαμε στο Breadboard. Έχοντας δώσει τροφοδοσία στο Breadboard συνδέσαμε τους ακροδέκτες τάσης των 2 αισθητήρων HC-SR04 στο +5V και τις γειώσεις τους αντίστοιχα. Στην συνέχεια, εντοπίζουμε τους 4 ακροδέκτες των κινητήρων οι οποίοι συνδέονται στην γέφυρα L298N και τους συνδέσαμε στις ψηφιακές I/O του Arduino Mega(I/O4 με IN1,I/O5 με IN2, I/O6 με IN3 και I/O7 με IN4). Με αυτόν τον τρόπο ελέγχονται οι κινητήρες. Στο κομμάτι των αισθητήρων HC-SR04, εφόσον έχουμε ήδη συνδέσει τις τροφοδοσίες τους, πρέπει να συνδέσουμε και τους δυο ακροδέκτες(Trig/Echo) του καθενός στο Arduino. Όπως έχουμε προαναφέρει ο ακροδέκτης Trig είναι υπεύθυνος για να στείλει τον παλμό και ο ακροδέκτης Echo για να μετρήσει την διάρκεια του παλμού. Συνεπώς, για τον πίσω αισθητήρα συνδέουμε τον Echo ακροδέκτη στην ψηφιακή I/O 8 και τον Trig ακροδέκτη στην I/O 9. Αντίστοιχα για τον μπροστά αισθητήρα συνδέουμε τον Echo ακροδέκτη στην I/O 10 και τον Trig ακροδέκτη στην I/O 1.

Έχοντας ολοκληρώσει το κομμάτι που αφορά τους κινητήρες και του αισθητήρες που είναι υπεύθυνοι για τα εμπόδια, πρέπει να προσθέσουμε τον ηχητικό συναγερμό το Joystick και τον Φωτισμό. Συνεπώς το διάγραμμα μας διαμορφώνεται ως εξής:

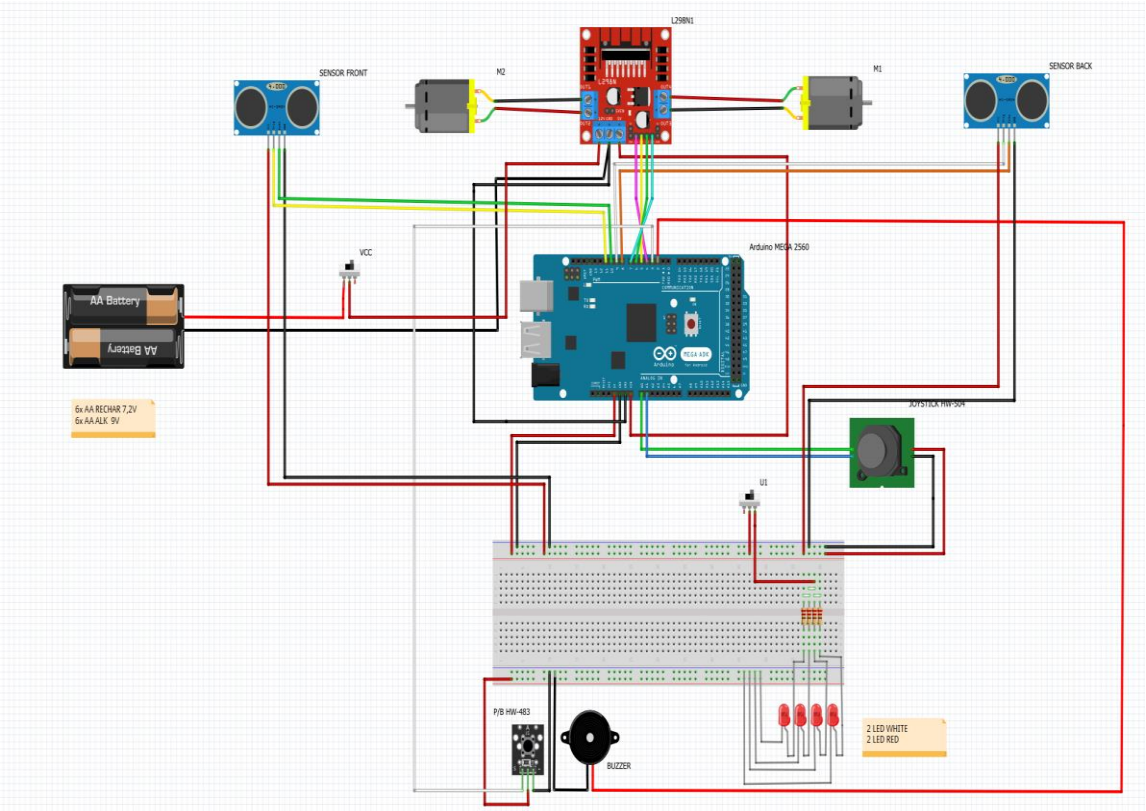


ΣΧΕΔΙΑΓΡΑΜΜΑ 3 ΟΛΟΚΛΗΡΩΜΕΝΟ ΚΥΚΛΩΜΑ

Για την σύνδεση του Joystick συνδέουμε να +5V και Gnd στο Breadboard και το X,Y αντίστοιχα στην A0 και A1. Επειδή ο μοχλός είναι αναλογικός, παίρνουμε 2 από τις Αναλογικές εισόδους του Arduino για να ΄διαβάσουμε΄ τις τιμές του. Όσον αφορά τον ηχητικό συναγερμό, τοποθετούμε τον Push Button διακόπτη στο Breadboard, συνδέουμε τους δυο δεξιούς ακροδέκτες στην τροφοδοσία και τον ακροδέκτη SWITCH στην ψηφιακή I/O 3. Κάθε φορά που πατάμε τον διακόπτη αλλάζει τιμή η είσοδο, από 1 γίνεται 0. Στην συνέχεια παίρνουμε το Buzzer και το τοποθετούμε στο Breadboard. Τον μαύρο ακροδέκτη τον συνδέουμε στο GND και το κόκκινο στην I/O 2. Οπότε θα πατάμε τον P/B διακόπτη τότε με βάση τον κώδικα που θα μιλήσουμε παρακάτω, το Buzzer θα ενεργοποιείτε. Τα τελευταία κομμάτια που θα τοποθετήσουμε είναι τα LED. Έχουμε τοποθετήσει 2 LED λευκά στο μπροστινό μέρος του αμαξιού και 2 κόκκινα στο πίσω μέρος. Το κάθε LED το έχουμε συνδέσει με μεγάλα Jumber έτσι να φωτίζουν προς την κατεύθυνση που θέλουμε και για να μπορέσουμε να τα συνδέσουμε στο Breadboard με τις αντιστάσεις. Για κάθε LED πήραμε μια αντίσταση 220Ω. Όλες οι αντιστάσεις και τα LED καταλήγουν σε ένα διακόπτη στον οποίο θα μπορεί ο χειρίστης να ελέγχει εάν θέλει να ενεργοποιήσει τον φωτισμό η όχι.

Έχοντας τοποθετήσει όλα τα εξαρτήματα μας το επόμενο κομμάτι είναι ο προγραμματισμός του Arduino ώστε να εκτελέσει τις εντολές που θα του δώσουμε.

Στο παρακάτω διάγραμμα φαίνεται πως θα πρέπει να είναι η κατασκευή μας ολοκληρωμένη (όσον αφορά το Hardware κομμάτι).



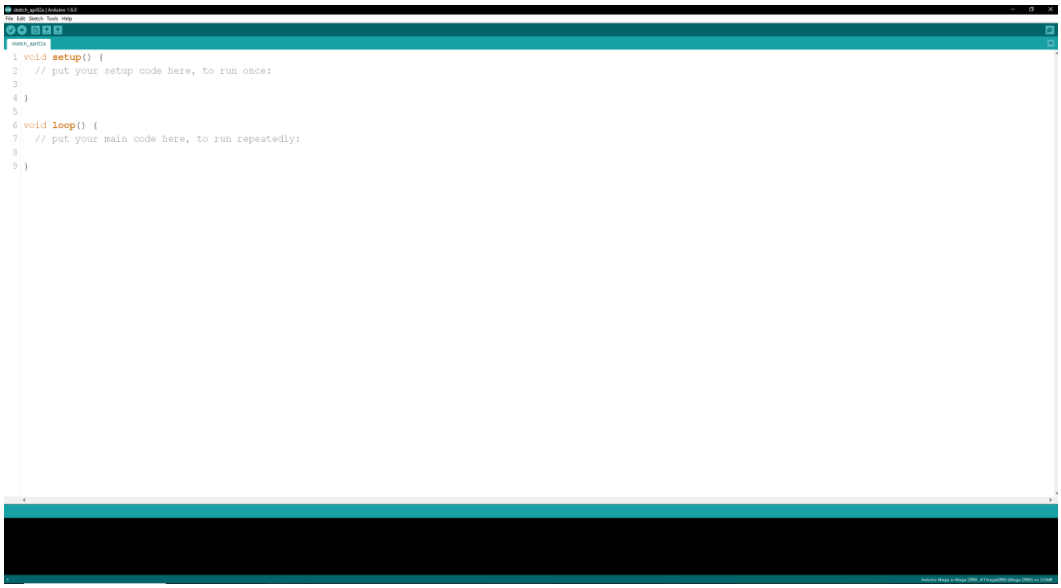
ΣΧΕΔΙΑΓΡΑΜΜΑ 4 ΟΛΟΚΛΗΡΗ ΚΑΤΑΣΚΕΥΗ/ΣΥΝΔΕΣΜΟΛΟΓΙΑ

3 Software – Λογισμικό

Όπως αναφέρεται και στην Wikipedia, ‘Προγραμματισμός καλείται το σύνολο των διαδικασιών σύνταξης ενός υπολογιστικού προγράμματος, συνήθως ως υλοποίηση κάποιων αλγορίθμων υστέρα από προσεκτική σχεδίαση, για την αυτοματοποιημένη εκτέλεση εργασιών η την επίλυση κάποιου υπολογιστικού προβλήματος από έναν υπολογιστή’. Ο προγραμματισμός υπάρχει παντού πλέον γύρω μας σε όλες τις συσκευές. Το μεγαλύτερο εύρος των οικιακών συσκευών μπορεί να εκλεχθεί από ένα κινητό μέσω Internet. Τα αυτοκίνητα πλέον με ένα κουμπί και τους καταλλήλους αισθητήρες μπορεί να παρκάρει μόνο του σε ένα δύσκολο δρόμο η ακόμα για να κάνει μια διαδρομή χωρίς τον οδηγό, σαν αυτόματος πιλότος. Με την βοήθεια πλέον των μικροϋπολογιστών μπορούν πλέον να γίνουν όλα πιο ευκολά στην ζωή μας . Με ένα Arduino μπορεί κάποιος να κάνει σχεδόν τα πάντα για να κάνει την ζωή του όσο πιο εύκολη μπορεί. Για παράδειγμα με ένα αισθητήρα φωτισμού και ένα κινητήρα μπορεί να κάποιος ευκολά να φτιάξει ένα πρόγραμμα έτσι ώστε όταν βγαίνει ο ήλιος να κατεβαίνει η τέντα και να ανεβαίνει αντίστοιχα όταν δύει ο ήλιος. Ακόμα και σε ιατρικό τομέα μπορούμε να φτιάξουμε συναγερμό οπού εάν αυξηθεί επικινδυνά η θερμοκρασία σε μια θερμοκοιτίδα να χτυπήσει ένα συναγερμός στο γραφείο της νοσοκόμας έτσι ώστε να πάει να το ελέγξει. Υπάρχουν πάρα πολλές παράλεγες του Arduino. Οποία και να διαλέξουμε το πρόγραμμα που θα χρησιμοποιήσουμε είναι το Arduino IDE. Είναι ένα πρόγραμμα ανοιχτού κώδικα (Δωρεάν) και που είναι προσβάσιμο από όλους. Έχει πολλά παραδείγματα και επεξηγήσεις μέσω σχολίων. Οπότε να μπορεί να το χρησιμοποιήσει και ένας επαγγελματίας προγραμματιστής αλλά και ένας ερασιτέχνης.

3.1 Περιβάλλον Arduino IDE

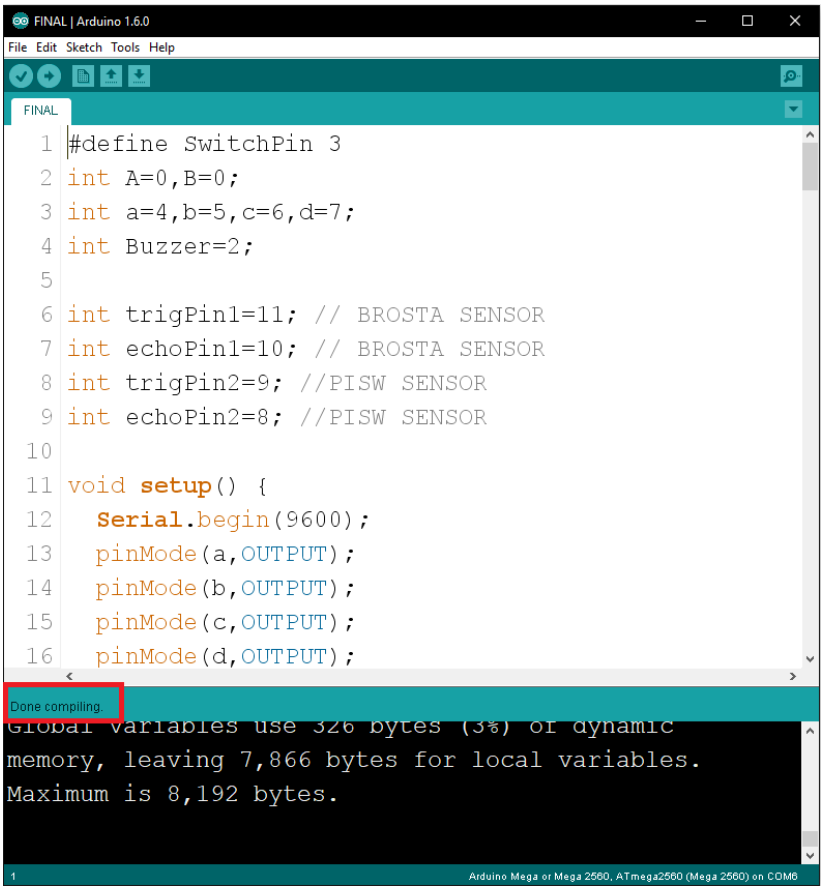
Το Arduino IDE είναι το πρόγραμμα με το οποίο γίνεται η μεταφορά του κώδικα από τον υπολογιστή στον μικροελεγκτής.



ΕΙΚΟΝΑ 21 ΠΕΡΙΒΑΛΛΟΝ ARDUINO IDE

Για να γίνει η μεταφορά του κώδικα από τον υπολογιστή στον μικροελεγκτής θα πρέπει να αρχικά να τον συνδέσουμε μέσω ένα καλώδιο USB. Ακόμη, θα πρέπει να έχουμε δηλώσει την έκδοση του Arduino που θα χρησιμοποιήσουμε (στην δικιά μας περίπτωση είναι το Arduino Mega 2560) και να ρυθμίσουμε και την θύρα USB να είναι ιδιά με το περιβάλλον του Arduino IDE. Εφόσον γίνουν όλες οι διαδικασίες που προαναφέραμε τότε μπορούμε να μεταφέρουμε το κώδικα μας πατώντας το File και μετρά Upload.

Ολοκληρώνοντας τον κώδικα μας, για να μεταφέρουμε τον κώδικα στον μικροελεγκτή θα πρέπει να γίνει μια προσωρινή αποθήκευση του κώδικα στον υπολογιστή και επαλήθευση του στην συνέχεια πατώντας την επιλογή Verify(Επαλήθευση).



ΕΙΚΟΝΑ 22 ΟΛΟΚΛΗΡΩΣΗ ΜΕΤΑΦΟΡΑΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

Με το που γίνει η επαλήθευση, και ο κώδικας μας είναι σωστός τότε θα ξεκινήσει η μεταφορά του στον μικροελεγκτή. Εάν ο κώδικας μας δεν είναι σωστός η έχει γίνει σε κάποια γραμμή λάθος το Arduino IDE θα μας ενημερώσει σε ποιο σημείο βρίσκεται το σφάλμα έτσι ώστε να το εντοπίσουμε ευκολά και να το διορθώσουμε η αλλάξουμε. [1]

3.2 Εντολές κώδικα

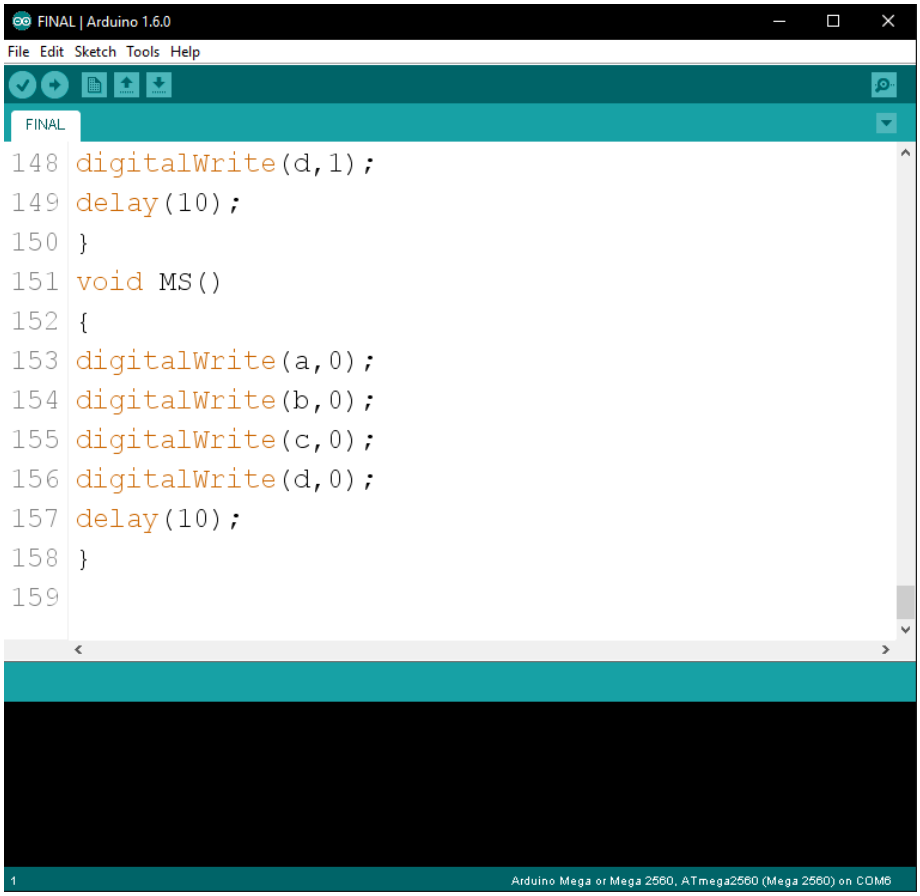
Οι εντολές στον κώδικα είναι ο τρόπος με τον οποίο λεμέ στον μικροελεγκτή πώς να εκτελέσει μια διεργασία ή εφαρμογή. Για παράδειγμα μπορούμε να με ένα σύνολο εντολών να βάλουμε τον μικροελεγκτή να αναβοσβήνει ένα LED με μια συγκεκριμένη συχνότητα. Ένα πιο προχωρημένο παράδειγμα είναι ο μικροελεγκτής να θερμομετρεί με την βοήθεια ενός καταλλήλου αισθητήρα την θερμοκρασία ενός χώρου και να αποστέλλει τις πληροφορίες(Αποτελέσματα) στον κινητό μας μέσω Bluetooth η ακόμα και μέσω Internet.

Το σύνολο των εντολών που είναι καταλληλά διαταγμένο μπορεί να οδηγήσει τον μικροελεγκτή στην εκτέλεση μιας εργασίας.

3.2.1 Εντολή Void()

Με την εντολή Void μπορούμε να δημιουργήσουμε μια συνάρτηση την οποία θα μπορούμε να καλούμε μέσα στο πρόγραμμα μας. Μια συνάρτηση αποτελεί ένα μικρό τμήμα ενός εκτεταμένου προγράμματος στην C. Πρέπει να είναι αυτόνομη και σωστά δομημένη. Όταν καλεστεί δημιουργεί ένα αντίγραφο του κώδικα της στην μνήμη στοίβας. Εκεί εκτελείται και αποθηκεύονται τα δεδομένα της. Όταν εκτελεστεί η συνάρτηση αυτό αντίγραφο σβήνεται και ενημερώνεται με την επιστρεφόμενη τιμή η καλούσα συνάρτηση. [7]

Παράδειγμα σύνταξης Void() :



ΕΙΚΟΝΑ 23 ΠΑΡΑΔΕΙΓΜΑ VOID

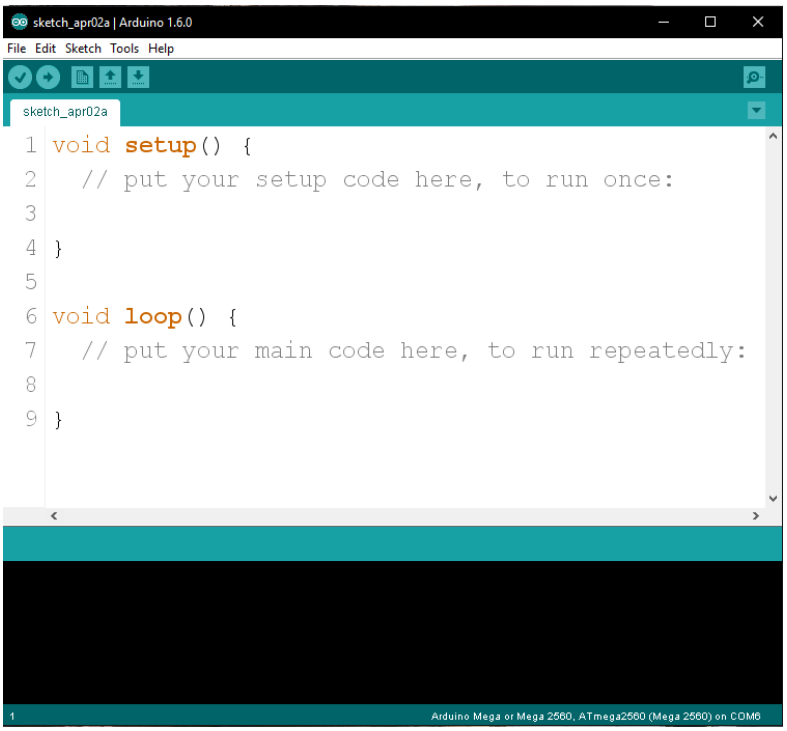
Πρέπει να σημειωθεί πως δεν μπορούμε να δώσουμε ότι όνομα θέλουμε στις συναρτήσεις μας. Υπάρχουν ήδη κάποιες προ εγκατεστημένες στην βιβλιοθήκη του Arduino IDE και δεν μπορούμε να αλλάξουμε το όνομα τους όπως η setup και η void.

3.2.2 Εντολή Void Setup – Void Loop

Το πρώτο πράγμα που συναντάμε ανοίγοντας το Arduino IDE, είναι οι δυο συναρτήσεις void setup() και void loop().

Η πρώτη συνάρτηση ονομάζεται **setup**. Εκεί περιλαμβάνεται ο κώδικας που εκτελείται μόνο μια φορά με το τροφοδοτήσουμε το Arduino. Εκεί γίνεται η αρχικοποίηση.

Η δεύτερη συνάρτηση ονομάζεται **loop** και εκτελείται συνέχεια όσο υπάρχει τροφοδοσία στο Arduino. [1]



ΕΙΚΟΝΑ 24 VOID SETUP/VOID LOOP

3.2.3 Εντολή #define

Με την εντολή **#define** γίνεται ο καθορισμός του ακροδέκτη για την ανάγνωση των δεδομένων(DATA) από τον αισθητήρα. [1]

3.2.4 Εντολή int

Με την εντολή **int** μπορούμε να δηλώσουμε μια μεταβλητή στην Wiring C. ο τύπος της μεταβλητής καθορίζει το χώρο οπού καταναλώνει το δεδομένο μέσα στην μνήμη καθώς και την μορφή του. [1]

Μερικά παραδείγματα:

Char	Ακέραιοι αριθμοί ενός Byte που αντιπροσωπεύουν χαρακτήρες ASCII
Byte	Ακέραιοι αριθμοί του ενός Byte.

Int	Ακέραιοι αριθμοί των 2 Byte για όλα τα Arduino.
Long	Μεγάλοι ακέραιοι αριθμοί μήκους 4 Byte

3.2.5 Εντολή Serial.begin

Η εντολή **serial.begin()** χρησιμοποιείται για την αρχικοποίηση της σειριακής επικοινωνίας με τον υπολογιστή. Η τιμή μέσα στην παρένθεση είναι η ταχύτητα επικοινωνίας . Η πιο συνηθισμένη τιμή είναι serial.begin(9600). [1]

3.2.6 Εντολή pinMode

Με την εντολή **pinMode** μπορούμε να ορίσουμε τους ψηφιακούς ακροδέκτες ως είσοδο η ως έξοδο. Για παράδειγμα pinmode(5,Output); . Στο συγκεκριμένο παράδειγμα ορίσαμε τον ψηφιακό ακροδέκτη 5 του Arduino ως έξοδο. Οι τιμές των παραμέτρων μέσα στην παρένθεση μπορούνε να είναι INPUT,OUTPUT και πιο εξειδικευμένα INPUT_PULLUP.

Το σύνολο των ψηφιακών ακροδεκτών είναι διαφορετικό ανάλογα τον τύπο του Arduino. Για παράδειγμα στο Arduino mega 2560 έχουμε από 0 έως και 13 ψηφιακούς ακροδέκτες. [1]

3.2.7 Εντολή digitalWrite

Η εντολή digitalWrite χρησιμοποιείται για τον καθορισμό της στάθμης του σήματος σε κάποιον από του ακροδέκτες. Η σύνταξη την εντολής γίνεται ως εξής : digitalWrite(ακροδέκτης, Στάθμη). [1]

3.2.8 Εντολή Delay – DelayMicroseconds

Σκοπός της εντολής delay είναι να δημιουργεί μια χρονική καθυστέρηση μέσα στο πρόγραμμα. Για παράδειγμα εάν θα ήθελα να δημιουργήσω μια χρονική καθυστέρηση 1 sec, η σύνταξη της εντολής θα ήταν ως εξής : delay(1000). [1]

3.2.9 Εντολή If/else

Σε ένα πρόγραμμα, μπορεί να χρειαστεί να εκτελέσουμε μια σειρά από εντολές, μόνο άμα ισχύει κάποια συνθήκη. Επίσης μπορεί να χρειαστεί να εκτελέσουμε κάποιες εντολές αν ισχύει μια συνθήκη και κάποιες άλλες εάν δεν ισχύει αυτήν η συνθήκη. Σε αυτή την περίπτωση φτιάχνουμε μια δομή ελέγχου.

Ένας τρόπος για να υλοποιήσουμε μια δομή ελέγχου είναι η εντολή If .

ΠΑΡΑΔΕΙΓΜΑ 1

If (Κάποια συνθήκη) {

Εντολή 1;

Εντολή 2;

}

Στο παραπάνω παράδειγμα εάν η συνθήκη είναι αληθές τότε θα εκτελεστεί η Εντολή 1 και 2. Εάν δεν είναι αληθές τότε δεν θα εκτελεστούν η εντολή 1 και 2.

ΠΑΡΑΔΕΙΓΜΑ 2

If(Κάποια συνθήκη) {

Εντολή 1;

Εντολή 2;

}

else if (Κάποια συνθήκη) {

Εντολή 3;

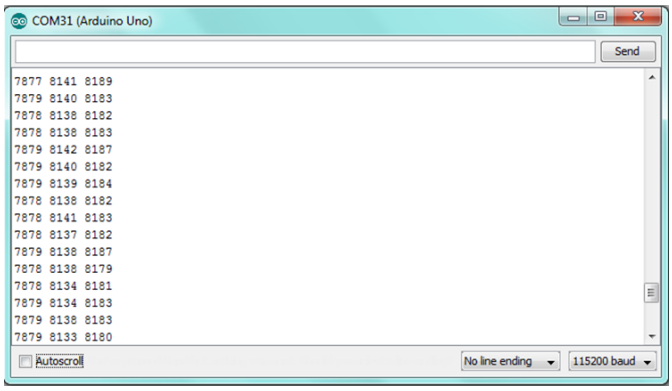
Εντολή 4;

}

Στο παραπάνω παράδειγμα εάν η πρώτη συνθήκη είναι αληθές τότε εκτελούνται οι εντολές 1 και 2. Εάν δεν είναι αληθές η πρώτη συνθήκη και είναι αληθές η δεύτερη συνθήκη τότε δεν εκτελούνται οι εντολές 1 και 2 αλλά εκτελούνται οι εντολές 3 και 4. [1]

3.2.10 Εντολή Serial.print – serial.println

Οι εντολές serial.print και serial.println, χρησιμοποιούνται στο Arduino για να εμφανίσουν δεδομένα στην σειριακή οθόνη (Serial Monitor). Ακόμη μπορούν να χρησιμοποιηθούν ώστε να μεταφέρουν πληροφορίες προς την θύρα USB με αποτέλεσμα να είναι ορατά σε ένα πρόγραμμα επεξεργασίας στον υπολογιστή. [1]



ΕΙΚΟΝΑ 25 ΣΕΙΡΙΑΚΗ ΟΘΟΝΗ ARDUINO IDE

3.2.11 Εντολή analogRead

Σκοπός της εντολής analogRead είναι, να διαβάσουμε την τιμή που έχει ένας από τους αναλογικούς ακροδέκτες του Arduino. Παράδειγμα:

A=analogRead(A0);

Στο παραπάνω παράδειγμα διαβάζουμε την τιμή του αναλογικού ακροδέκτη A0 και την αποθηκεύουμε στο A. [1]

3.2.12 Εντολή tone/Notone

Με την εντολή tone, δημιουργούμε ένα τετραγωνικό παλμό σε ένα ψηφιακό ακροδέκτη με την συχνότητα που εμείς θέλουμε. [7]Παράδειγμα:

tone(Ακροδέκτης, Συχνότητα);

Για να σταματήσει να παράγεται ο παλμός πρέπει να μπει η εντολή notone. Παράδειγμα:

Notone(Ακροδέκτης);

3.2.13 Εντολή Digital.Read

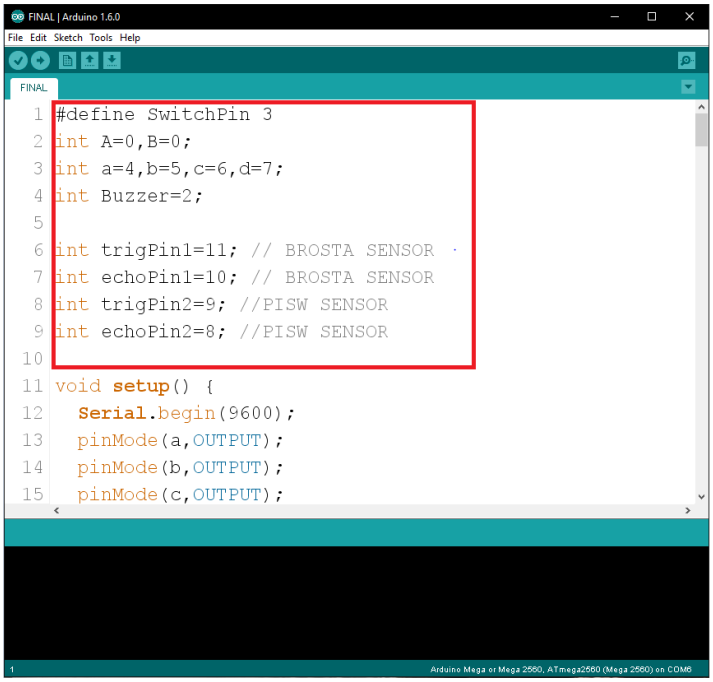
Με την εντολή digital.read μπορούμε να μετρήσουμε την τιμή σε ένα ψηφιακό ακροδέκτη [7]

3.3 Προγραμματισμός Arduino

Όπως προαναφέραμε, πρόγραμμα είναι ένα σύνολο εντολών με τις οποίες πρέπει ο μικροϋπολογιστής να ακολουθήσει έτσι ώστε να παράγει το επιθυμητό αποτέλεσμα για το χρήστη. Εφόσον έχουμε εξηγήσει όλες τις παραπάνω εντολές στην συνέχεια θα δείξουμε πως μπορούμε βάζοντας στην σωστή σειρά και ορίζοντας σωστά τις μεταβλητές μπορούμε να φτιάξουμε το πρόγραμμα μας. [5]

3.3.1 Αρχικοποίηση

Πριν ξεκινήσουμε να γράφουμε στην πρώτη συνάρτηση (Void setup) πρέπει να ορίσουμε τις μεταβλητές μας [1]



ΕΙΚΟΝΑ 26 ΑΡΧΙΚΟΠΟΙΗΣΗ ΜΕΤΑΒΛΗΤΩΝ

Με την εντολή **int** δηλώνουμε τις μεταβλητές:

1) **A** και **B** οπού αφορούν το Joystick στις αναλογικές εισόδους.

2) Τις μεταβλητές **a**, **b**, **c**, **d** τις οποίες έχουμε συνδέσει αντίστοιχα στο Hardware στους ψηφιακούς ακροδέκτες 4, 5, 6, 7 (αφορά τα μοτέρ κίνησης)

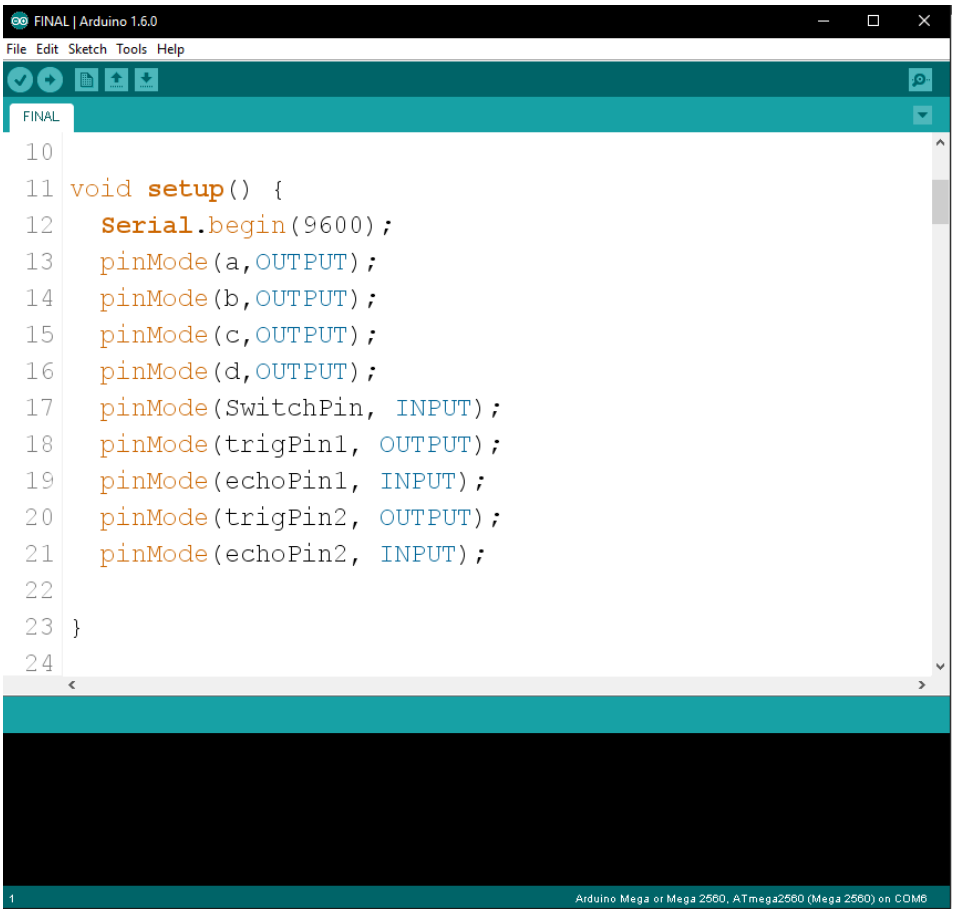
3) Buzzer με τον ψηφιακό ακροδέκτη 2 (αφορά το ηχείο)

4) Το **trigPin1** με τον ψηφιακό ακροδέκτη 11 και **echoPin1** με τον ψηφιακό ακροδέκτη 10. (αφορά τον μπροστά αισθητήρα HC-SR04)

5) Το **trigPin2** με τον ψηφιακό ακροδέκτη 9 και **echoPin2** με τον ψηφιακό ακροδέκτη 8. (αφορά τον πίσω αισθητήρα HC-SR04)

3.3.2 Void Setup και Void Loop

Εφόσον έχουμε τελειώσει με τις αρχικοποιήσεις των μεταβλητών είμαστε έτοιμοι να προχωρήσουμε στην συνάρτηση **Void Setup**.



ΕΙΚΟΝΑ 27 ΠΡΟΓΡΑΜΜΑ ΜΕΣΑ ΣΤΟ VOID SETUP

Όπως έχουμε ήδη πει η Void Setup είναι μια συνάρτηση η οποία εκτελείτε μια φορά κατά την ενεργοποίηση του μικροϋπολογιστή. Στην Void Setup συνάρτηση γίνεται ο

ορισμός των ακροδεκτών του Arduino ως είσοδος και ως έξοδος ανάλογα τις ανάγκες μας.

Η πρώτη εντολή η οποία ξεκινάμε είναι **Serial.begin(9600)** για την αρχικοποίηση της σειριακής επικοινωνίας με τον υπολογιστή. 9600 είναι η ταχύτητα της επικοινωνίας με τον υπολογιστή.

Η επόμενες εντολές αφορούν τον ορισμό των εισόδων και εξόδων χρησιμοποιώντας την εντολή **pinmode** όπως βλέπουμε και στην παραπάνω εικόνα.

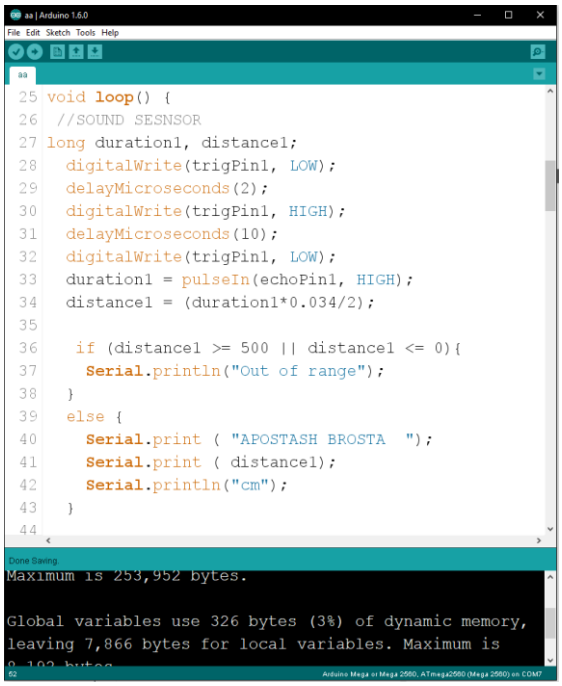
Ορίζουμε της μεταβλητές **a, b, c, d** ως έξοδοι. Οι συγκεκριμένες μεταβλητές αφορούν τα μοτέρ. Όταν η έξοδος θα γίνεται **HIGH** τότε το ανάλογο μοτέρ θα ενεργοποιείται.

Οι μεταβλητές **trigPin1** ορίζεται ως έξοδο γιατί θέλω να στείλω στον αισθητήρα το παλμό ενεργοποίησης και η **echoPin1** ως είσοδο γιατί θέλω να πάρω την μέτρηση που έκανε ο αισθητήρας. Το ίδιο ακριβώς ισχύει και για τον δεύτερο αισθητήρα **HC-SR04** για **trigPin2** και **echoPin2** αντίστοιχα.

Όταν ολοκληρωθεί η συνάρτηση Void Setup τότε μπαίνουμε στο κύριος πρόγραμμα οπού γράφεται εντός της συνάρτησης Void loop. Στις επόμενες τρεις παραγράφους έχει χωριστεί το πρόγραμμα σε 3 μέρη. Το πρώτο οπού αφορά τους αισθητήρες HC-SR04, το δεύτερο οπού αφορά την ενεργοποίηση του ηχητικού συναγερμού και τέλος τον έλεγχο των μοτέρ με την χρήση Joystick. [1]

3.3.3 Κώδικας Αισθητήρων HC-SR04

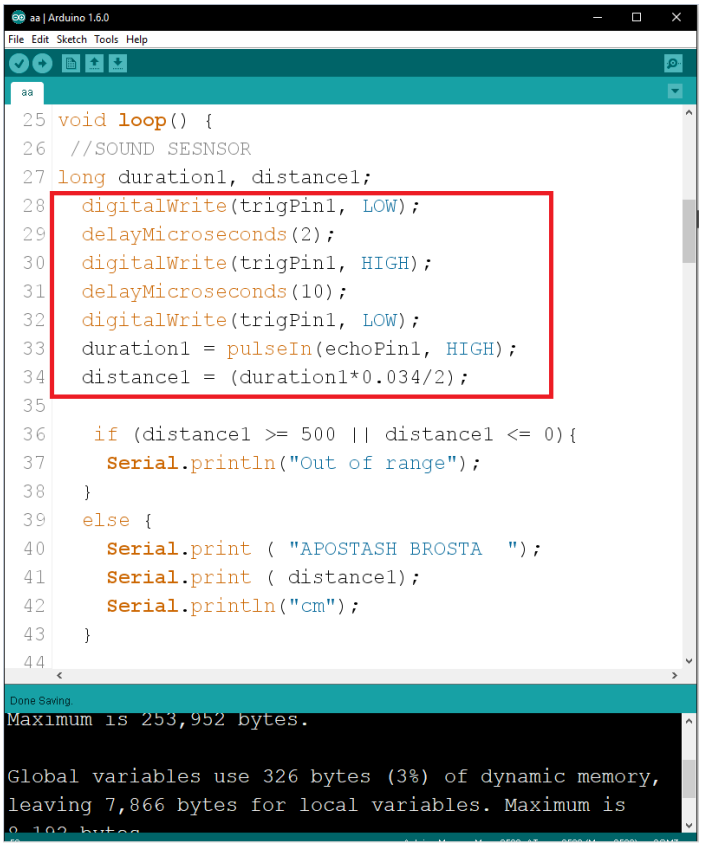
Το πρώτο μέρος του κυρίου προγράμματος μας αφορά τον μπροστά και πίσω αισθητήρα HC-SR04. Την ενεργοποίηση των αισθητήρων, την εξίσωση οπού θα υπολογιστεί η απόσταση από το εμπόδιο και η έξοδος του αποτελέσματος στην σειριακή οθόνη.



ΕΙΚΟΝΑ 28 ΚΩΔΙΚΑΣ HC-SR04

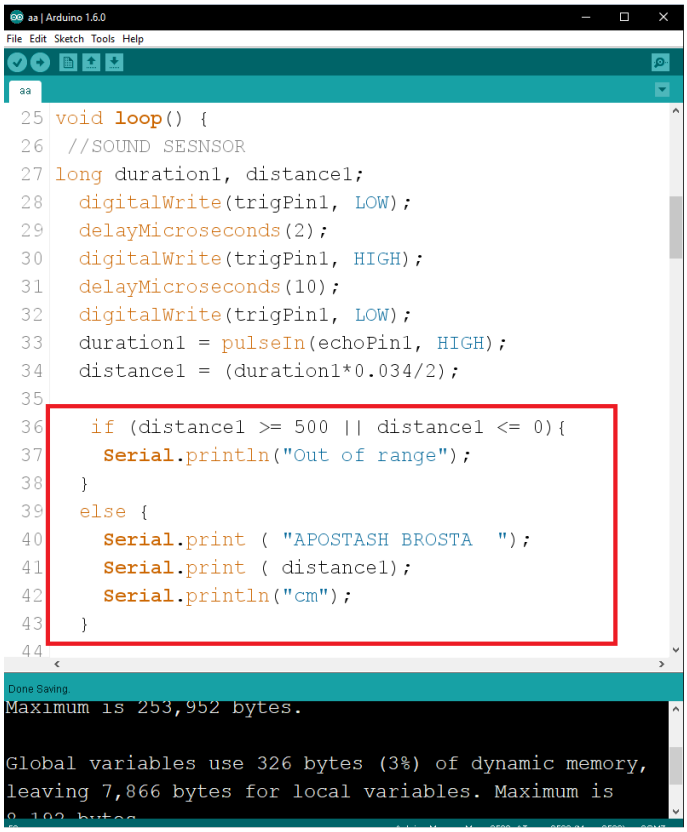
Για αρχή χρησιμοποιώντας την εντολή ορίζουμε τις μεταβλητές duration1 και distance1 ώστε να μπορούν να πάρουν ακέραιες τιμές.

Στην συνέχεια η επόμενες 5 γραμμές κώδικα αφορούν την ενεργοποίηση του μπροστά αισθητήρα HC-SR04.



ΕΙΚΟΝΑ 29 ΕΝΕΡΓΟΠΟΙΗΣΗ ΑΙΣΘΗΤΗΡΑ HC-SR04

Με την χρήση της εντολής **digitalWrite** δίνουμε τον παλμό οπού χρειάζεται ο HC-SR04 για να ενεργοποιηθεί και να στείλει το πρώτο κύμα υπέρηχων. Εφόσον ενεργοποιηθεί ο παλμός και σταλθούν οι υπέρηχοι, με την εντολή pulseIn καταχωρώ τον χρόνο οπού χρειάστηκε να πάει και να γυρίσει ο παλμός στον αισθητήρα στην μεταβλητή **duration1**. Εφόσον γνωρίζω πλέον τον χρόνο μπορώ να υπολογίσω την απόσταση(**distance1**). Επειδή η διάρκεια του υπέρηχου περιλαμβάνει το χρόνο που χρειάστηκε ο υπέρηχος να πάει και να γυρίσει από εμπόδιο, τον διαιρούμε με το 2 και τον πολλαπλασιάζουμε με 0,034 ώστε να βρούμε την απόσταση σε Cm. Ο λόγος που πολλαπλασιάσαμε με 0,034 είναι επειδή η ταχύτητα του ήχου είναι 343 m/s.



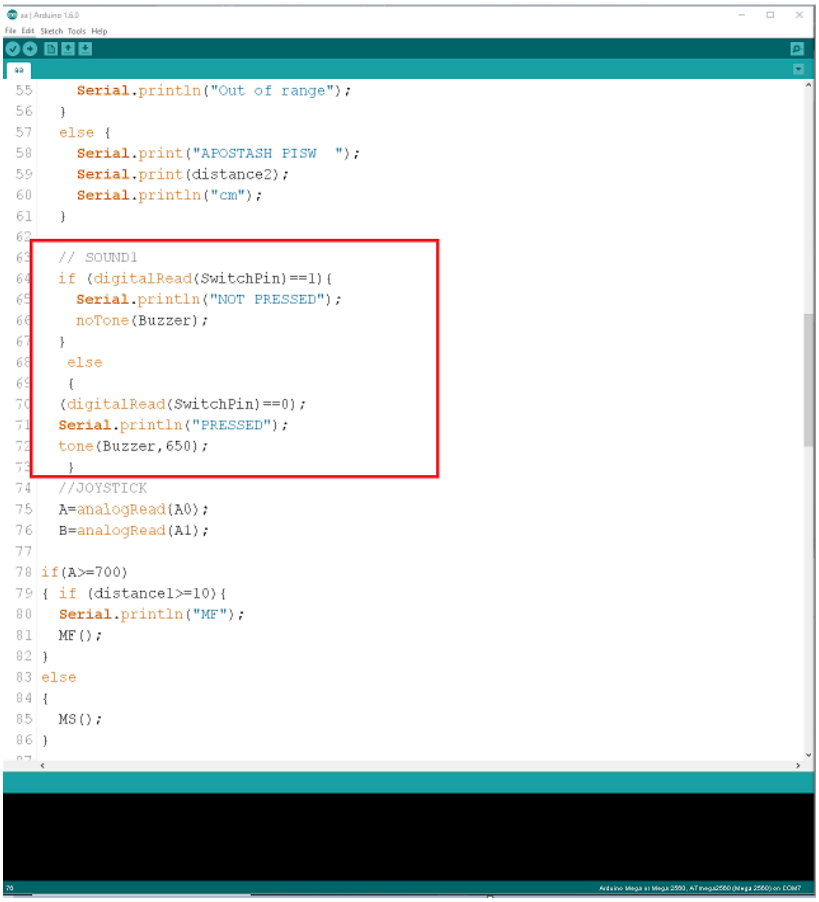
ΕΙΚΟΝΑ 30 ΕΞΟΔΟ ΑΠΟΤΕΛΕΣΜΑΤΟΣ ΣΤΗΝ ΣΕΙΡΙΑΚΗ ΟΘΟΝΗ

Εφόσον έχουμε ήδη υπολογίσει και πλέον γνωρίζουμε την απόσταση του αντικειμένου από τον αισθητήρα, ήρθε η ώρα να προβάλουμε το αποτέλεσμα. Όπως έχουμε προαναφέρει ο αισθητήρας HC-SR04 έχει εμβέλεια από 2cm έως και 400cm με ακρίβεια 3mm. Με την εντολή if, μπορούμε να περιορίσουμε αυτό το εύρος και να λαμβάνουμε μόνο το εύρος του αισθητήρα. Με κάποιες δοκιμές παρατηρήθηκε ότι ο αισθητήρας μπορεί να μετρήσει έως και 500cm χωρίς σφάλματα. Συνεπώς με την εντολή if εάν η απόσταση ήταν πάνω από 500cm αυτό που θα προβληθεί στην σειριακή οθόνη είναι “Out of range”. Εάν όμως η μετρούμενη απόσταση είναι από 0cm έως και 500cm τότε για τον μπροστά αισθητήρα, στην σειριακή οθόνη θα εμφανιστεί “Apostash Brosta Xcm” όπου X είναι η απόσταση που μετρήθηκε από τον αισθητήρα.

Χρησιμοποιώντας την εντολή Serial.print() μπορώ να προβάλω τα αποτελέσματα που θέλω στην σειριακή οθόνη (Στον υπολογιστή). Μέσα στο πρόγραμμα χρησιμοποιώ την εντολή Serial.print τρείς φορές έτσι ώστε να παρατηρώ συνεχώς στις δοκιμές μου την μέτρηση απόστασης από τους αισθητήρες. [8]

3.3.4 Κώδικας ενεργοποίησης ηχητικού συναγερμού

Άλλη μια από τις λειτουργίες του Αναπηρικού αμαξιού είναι ο ηχητικός συναγερμός. Δεν είναι λίγες οι φορές που οι άνθρωποι που χρησιμοποιούν αναπηρικό αμαξιού να μην μπορούν είτε να φωνάξουν είτε να ενημερώσουν τους γύρω τους ότι υπάρχει ανάγκη η κίνδυνος. Για να καλύψουμε αυτήν την ανάγκη, προσθέσαμε ένα διακόπτη το οποίο ενεργοποιεί ένα ηχητικό συναγερμό σε περίπτωση ανάγκης.



```
    Serial.println("Out of range");
  }
  else {
    Serial.print("APOSTASH PISW  ");
    Serial.print(distance2);
    Serial.println("cm");
  }

  // SOUND1
  if (digitalRead(SwitchPin)==1){
    Serial.println("NOT PRESSED");
    noTone(Buzzer);
  }
   else
   {
  (digitalRead(SwitchPin)==0);
  Serial.println("PRESSED");
  tone(Buzzer,650);
   }
  //JOYSTICK
  A=analogRead(A0);
  B=analogRead(A1);

if(A>=700)
{ if (distance1>=10){
  Serial.println("MF");
  MF();
}
else
{
  MS();
}
```

ΕΙΚΟΝΑ 31 ΠΡΟΓΡΑΜΜΑ ΕΝΕΡΓΟΠΟΙΗΣΗΣ BUZZER

Για να προσθέσουμε τον ηχητικό συναγερμό χρειαστήκαμε ένα Push-Button διακόπτη και ένα Buzzer. Όταν το κουμπί του διακόπτη είναι πατημένο, ο διακόπτης μας δίνει 0. Έστω ότι **SwitchPin1** είναι ο διακόπτης, όταν πατηθεί το κουμπί και για όσο είναι πατημένο τότε **SwitchPin1=0**. Εάν δεν πατάμε το κουμπί ο διακόπτης μας δίνει 1. Τότε θα περνάμε **SwitchPin1=1**. Γνωρίζοντας αυτά συντάσσουμε τον παρακάτω κώδικα.

Εάν διαβαστεί(digitalRead) ο διακόπτης, και είναι 1 τότε να μην ενεργοποιηθεί το Buzzer.

If (digitalRead(SwitchPin)==1);

notone Buzzer;

Αλλιώς, Εάν διαβαστεί ο διακόπτης και είναι 0 τότε να ενεργοποιηθεί το Buzzer στην συχνότητα Χ (εμείς χρησιμοποιήσαμε μια τυχαία τιμή 650).

else

{

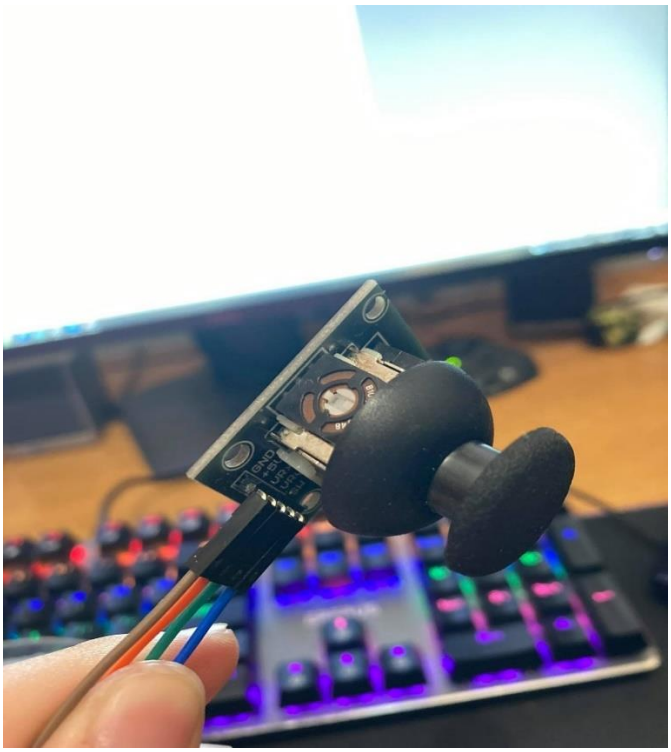
(digitalRead(SwitchPin)==0);

tone(Buzzer,650);

Επιπροσθέτως, στον κώδικα έχει προσδεθεί δυο φορές η εντολή Serial.println που προβάλει στην σειριακή οθόνη εάν ο διακόπτης είναι πατημένος η όχι (PRESSED NOT PRESSED) έτσι ώστε να ελέγχουμε ότι τα υλικά δουλεύουν σωστά. [9]

3.3.5 Κώδικας Χειριστηρίου Joystick

Το βασικότερο χαρακτηριστικό του αναπηρικού αμαξιδίου είναι η κίνηση με τηλεχειριστήριο. Για να επιτευχθεί αυτό γίνετε με την χρήση ενός μοχλού η αλλιώς Joystick. [10]



ΕΙΚΟΝΑ 33 JOYSTICK ΜΟΧΛΟΣ

Ο μοχλός μπορεί να κινηθεί προς όλες τις κατευθύνσεις και να λάβει τιμές από όλα τα σημεία, ανάλογα την θέση που βρίσκεται. Έχει δυο άξονες, τον άξονα Χ (οριζόντιος) και τον άξονα Υ(Κάθετος). Τον Άξονα Υ στον κώδικα μας τον έχουμε ορίσει ως Α και Β τον άξονα Χ. Έχουμε ήδη προαναφέρει ότι ο μοχλός πέρα από την τροφοδοσία έχει δυο εισόδους Α0 και Α1. Η Α0 αναφέρεται στον άξονα Υ και η Α1 αναφέρεται στον άξονα Χ. Συνεπώς οι πρώτες δυο γραμμές που αφορούν τον κώδικα του μοχλού είναι η αρχικοποίηση των δυο αξόνων

A=analogRead(A0);

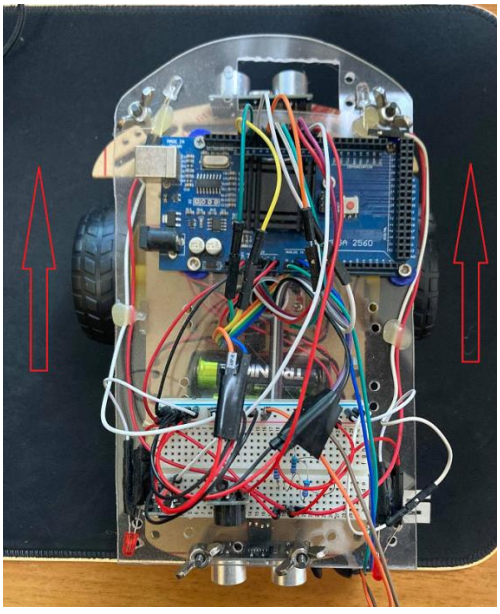
B=analogRead(A1);

Έχοντας ορίσει τους δυο αξίωνες το επόμενο βήμα είναι να φτιάξουμε τις 5 συναρτήσεις που θα καλούμε στο πρόγραμμα έτσι ώστε να γίνονται οι 4 βασικές κινήσεις Ευθεία, Πίσω, Αριστερά, δεξιά και την Stop

Βγαίνοντας λοιπόν από το Void Loop πηγαίνουμε στο τέλος του κώδικα και χρησιμοποιούμαι την εντολή Void X(), οπού X θα είναι το όνομα της συνάρτησης. Η πρώτη κίνηση που θα φτιάξουμε μέσω της συνάρτησης είναι η μπροστά κίνηση (MF). Για να γίνει αυτό θα πρέπει να ορίσουμε καταλληλά τις ψηφιακές εξόδους a,b,c, και d. Ανάλογα την επιλογή που θα κάνουμε στις εξόδους θα είναι ανάλογη και η κίνηση του μοτέρ(τροχός). Για παράδειγμα ο αριστερός τροχός ελέγχεται από τις εξόδους a και b. Εάν a=1 και b=0 τότε ο αριστερός τροχός θα κινηθεί ευθεία. Εάν a=0 και b=1 τότε ο τροχός θα γύριζε προς τα πίσω. Εφόσον η πρώτη μας συνάρτηση αφορά την μπροστά κίνηση θέλουμε και οι δυο τροχοί να γυρίζουν με την ιδιά φορά προς την ιδιά κατεύθυνση. Οπότε χρησιμοποιώντας την εντολή **digitalWrite** ορίζουμε τις ψηφιακές εξόδους a,b,c,d ως εξής.

```
void MF()
{
digitalWrite(a,1);
digitalWrite(b,0);
digitalWrite(c,1);
digitalWrite(d,0);
delay(10);
}
```

Χρησιμοποιώντας λοιπόν, τον παραπάνω κώδικα ρυθμίζουμε την πολικότητα στα μοτέρ έτσι ώστε να μπορεί να γυρίσουν και οι δυο τροχοί προς την ιδιά κατεύθυνση. Οπότε, όταν καλούμε την συνάρτηση MF οι τροχοί θα κατευθύνονται προς τα μπροστά.



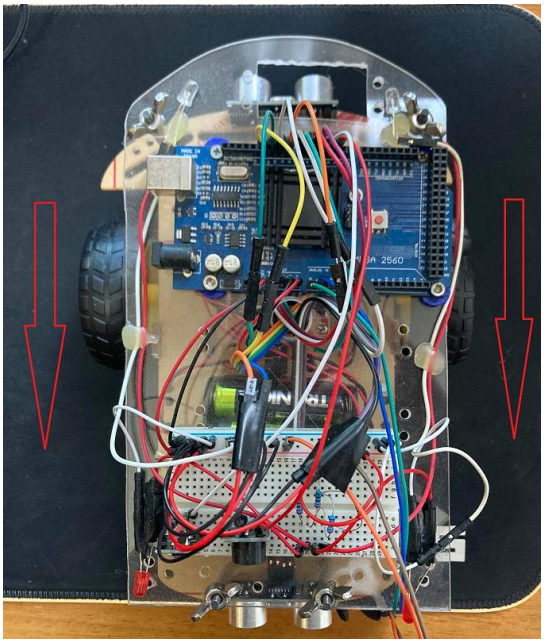
ΕΙΚΟΝΑ 33 ΚΙΝΗΣΗ ΜΠΡΟΣΤΑ

Με τον ίδιο ακριβώς τρόπο, φτιάχνουμε και την δεύτερη συνάρτηση η οποία αφορά την κίνηση προς τα πίσω. Η συνάρτηση που αφορά την συγκεκριμένη κίνηση θα την ονομάσουμε MB. Για να φτιάξουμε την συνάρτηση MB θα πρέπει να ορίσουμε τις

εξόδους ακριβώς το αντίθετο από την συνάρτηση MF διότι θέλουμε να κάνει ακριβώς το αντίθετο.

```
void MB()
{
digitalWrite(a,0);
digitalWrite(b,1);
digitalWrite(c,0);
digitalWrite(d,1);
delay(10);
}
```

Έχοντας ορίσει τις ψηφιακές εξόδους a,b,c,d με αυτόν τον τρόπο η συλλογική κίνηση των μοτέρ θα κινεί τους τροχούς προς τα πίσω.

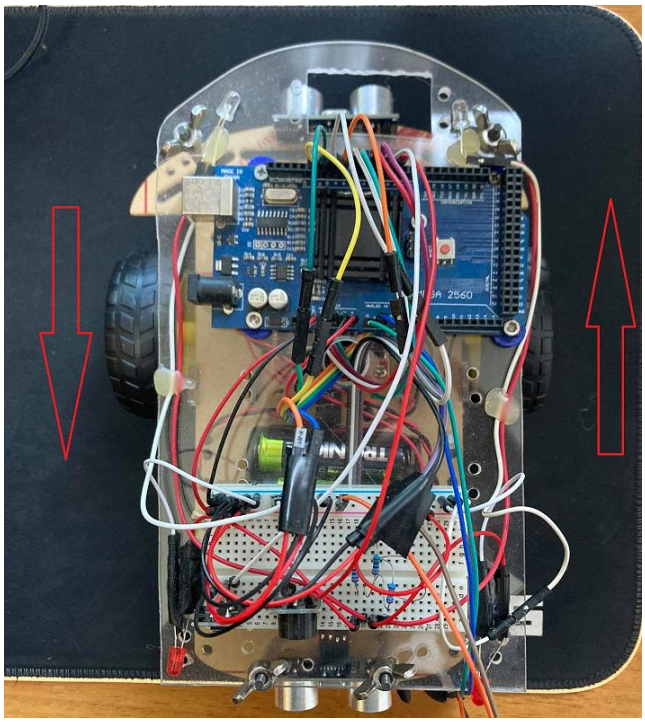


ΕΙΚΟΝΑ 34 ΚΙΝΗΣΗ ΠΙΣΩ

Με την ολοκλήρωση των συναρτήσεων MF και MB, πρέπει να δημιουργηθούν άλλες δυο συναρτήσεις που θα αφορούν την κίνηση του αμαξιού Αριστερά και δεξιά. Για να γίνει αυτό θα πρέπει οι δυο τροχοί να γυρίζουν αντίθετα ο ένας από τον άλλο. Έτσι ώστε να κινηθεί το αμαξιού προς τα αριστερά, θα πρέπει ο αριστερά τροχός να γυρίζει προς τα πίσω ενώ ταυτόχρονα ο δεξιά θα γυρίζει προς τα μπροστά. Φτιάχνοντας την συνάρτηση ML ως εξής:

```
void ML()
{
digitalWrite(a,1);
```

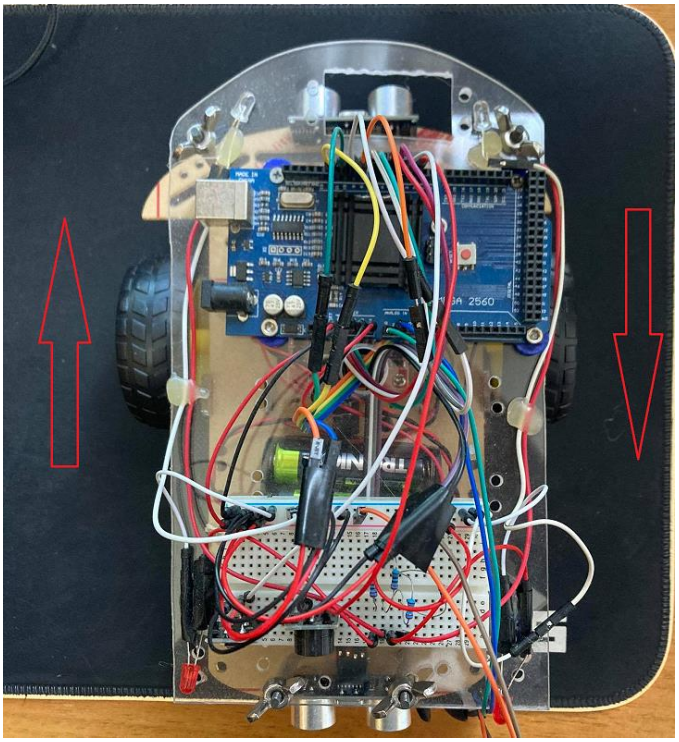
```
digitalWrite(b,0);
digitalWrite(c,0);
digitalWrite(d,1);
delay(10);
}
```



ΕΙΚΟΝΑ 35 ΑΡΙΣΤΕΡΟΣΤΡΟΦΗ ΚΙΝΗΣΗ

Έχοντας ήδη φτιάξει την συνάρτηση ML, η δημιουργία της συνάρτηση MR που αφορά την κίνηση προς τα Δεξιά θα είναι το ακριβώς αντίθετο. Αυτή την φορά θα πρέπει, ο αριστερός τροχός να γυρίζει προς τα μπροστά και ο δεξιός προς τα πίσω. Αυτό θα έχει ως αποτέλεσμα την κυνηγητού αμαξιού προς τα δεξιά.

```
void MR()
{
digitalWrite(a,0);
digitalWrite(b,1);
digitalWrite(c,1);
digitalWrite(d,0);
delay(10);
}
```



ΕΙΚΟΝΑ 36 ΔΕΞΙΟΣΤΡΟΦΗ ΚΙΝΗΣΗ

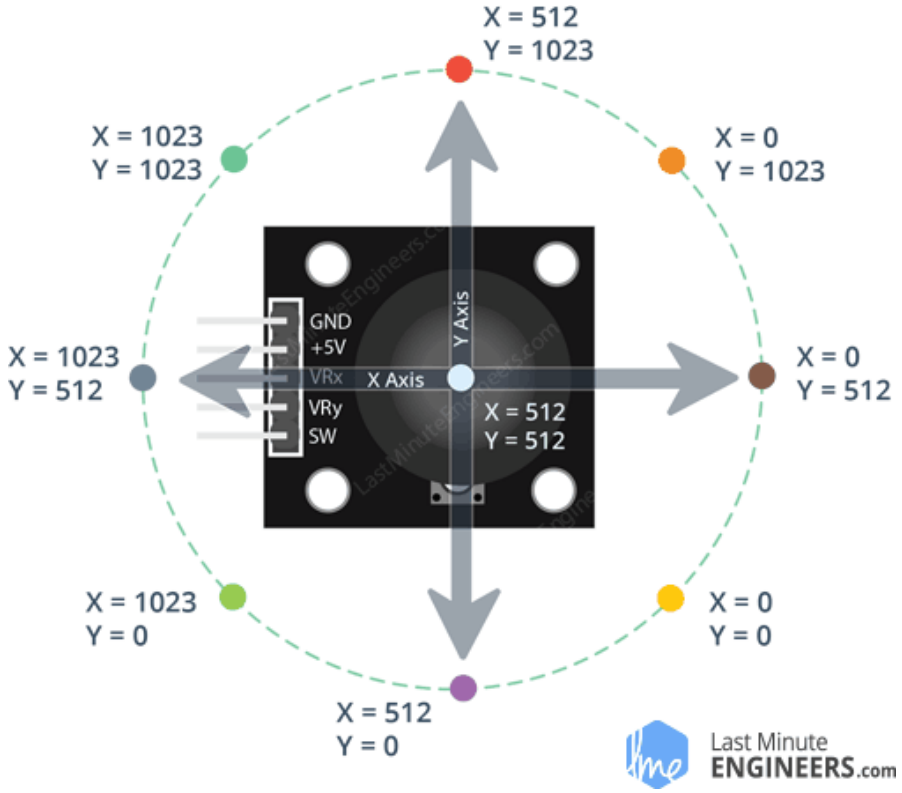
Έχοντας φτιάξει τις 4 κινήσεις, θα χρειαστούμε άλλη μια συνάρτηση που θα χρησιμοποιηθεί για την ακινητοποίηση του αμαξιού. Για να γίνει αυτό ορίζουμε τις 4 ψηφιακές εξόδους a,b,c,d με την τιμή 0. Αρά ο κώδικας της συνάρτησης MS θα είναι έτσι:

```
void MS()
{
digitalWrite(a,0);
digitalWrite(b,0);
digitalWrite(c,0);
digitalWrite(d,0);
delay(10);
}
```

Έχοντας ολοκληρώσει και τις 4 συναρτήσεις που αφορούν την κίνηση αμαξιού, συνεχίζουμε τον κώδικα μέσα στο Void Loop.

Για να μπορέσουμε να ελέγξουμε την κίνηση του αμαξιού ανάλογα με την θέση που βρίσκεται το Joystick θα χρησιμοποιήσουμε την εντολή If και τις συναρτήσεις που δημιουργήσαμε παραπάνω. Στην παρακάτω φωτογραφία φαίνεται η τριγωνομετρία του μοχλού. Έστω ότι, θέλουμε να δώσουμε την εντολή στο αμάξι να κινηθεί ευθεία. Θα πρέπει να κινήσουμε τον μοχλό προς τα πάνω. Σύμφωνα με την εικόνα όταν κινηθεί προς τα πάνω ο μοχλός η τιμή στον άξονα χ θα αυξηθεί ενώ στον y θα παραμείνει μηδέν. Μπορούμε να υπολογίσουμε από την παρακάτω εικόνα ότι η αρχή των αξόνων είναι η τιμή για τον X = 512 και για τον Y επίσης είναι Y=512. Συνεπώς

εάν η τιμή του X πάρει μια τιμή περίπου 700 θέλουμε το αμαξιού να κινηθεί μπροστά.



ΕΙΚΟΝΑ 37 ΔΙΑΓΡΑΜΜΑ JOYSTICK

Εάν η τιμή του X είναι ίση ή πάνω από 700 τότε να κινηθεί το αμαξιού ευθεία (θα καλέσουμε την συνάρτηση MF) αλλιώς να σταματήσει. Για να το πέτυχουμε αυτό θα χρησιμοποιήσουμε τον παρακάτω κώδικα:

```
if(A>=700)
{ if (distance1>=10){
  Serial.println("MF");
  MF();
}
else
{
  MS();
```

Επειδή θέλουμε να αποφύγουμε τον κίνδυνο της σύγκρουσης του αμαξιού με κάποιο εμπόδιο (πχ τοίχο, πάσσαλο κλπ.) προσθέσαμε άλλη μια εντολή if η οποία παίρνει την απόσταση από τον αισθητήρα HC-SR04 (μπροστά αισθητήρας) και φροντίζει να σταματήσει την κίνηση των τροχών εάν η απόσταση του αμαξιού είναι κάτω από 10 cm. Συνεπώς για να γίνει η κίνηση μπροστά του αμαξιού θα πρέπει: Α) ο μοχλός να κινηθεί προς τα πάνω (χ>=700) και να μην υπάρχει εμπόδιο εντός 10 Cm. Εάν οι δυο

παραπάνω παράμετροι είναι αληθές τότε, για όσο έχουμε τον μοχλό προς τα πάνω το αμαξιού θα κινείτε ευθεία.

Με τον ίδιο ακριβώς τρόπο μπορούμε να δημιουργήσουμε και την κίνηση προς τα πίσω. Εάν μοχλός κινηθεί προ τα κάτω και ο Χ πάρει τιμή μικρότερη η ίση του 300 και εάν δεν υπάρχει εμπόδιο στο εντός 10cm στον **ΠΙΣΩ** αισθητήρα, τότε το αμαξιού να κινηθεί προς τα πίσω. Για να το πέτυχουμε αυτό γράφουμε τον παρακάτω κώδικα:

```
if(A<=300)
{ if (distance2>=10) {
Serial.println("MB");
  MB();
}
else
{
  MS();
}
```

Να σημειωθεί πως για να ελέγχουμε την απόσταση χρησιμοποιούμε και τον ανάλογο αισθητήρα. Δηλαδή για τον έλεγχο της απόστασης μπροστά πήραμε τιμές από τον μπροστά αισθητήρα HC-SR04 (distance1) ενώ για τον πίσω πήραμε τιμές από τον πίσω αισθητήρα (distance2).

Για την περιστροφή του αμαξιού προς τα αριστερά θέλουμε ο μοχλός να κινηθεί προς τα αριστερά. Επειδή η κίνηση του αμαξιού γίνεται με επιτόπου περιστροφή δεν χρειαζόμαστε αισθητήρες ασφάλειας. Συνεπώς για όσο ο άξονας Υ είναι μικρότερος η ίσως από 300 θα καλέσουμε την συνάρτηση ML. Αρά ο κώδικας μας θα είναι κάπως έτσι:

```
if(B<=300)
{
  Serial.println("ML");
  ML();
}
```

Αντίστοιχα για την περιστροφή προς τα δεξιά του αμαξιού θέλουμε ο μοχλός να κινηθεί προς τα δεξιά. Συνεπώς για όσο ο άξονας Υ είναι μεγαλύτερος η ίσως από 700 θα καλέσουμε την συνάρτηση MR. Αρά ο κώδικας μας θα είναι κάπως έτσι:

```
if(B>=700)
{
  Serial.println("MR");
  MR
    ();
```

```
}
```

Κλείνοντας των κώδικα μας θα πρέπει να προσθέσουμε και την περίπτωση ο μοχλός θα βρίσκεται στην αρχή των αξόνων, έτσι ώστε το αμαξιού σε περίπτωση που δεν χρησιμοποιείται να είναι κινητοποιημένο. Για να το πέτυχουμε αυτό ουσιαστικά θα πρέπει ο μοχλός να είναι στην αρχή τον αξόνων X και Y.

```
if(A>=450 && A<=550 && B>=450 && B<=550)
{
  Serial.println("MS");
  MS();
}
```

Επειδή ο μοχλός υπάρχει περίπτωση να έχει κάποιο μικρό σφάλμα(DeadZone) είτε μπορεί να το κινήσει κάποιος ελάχιστα κατάλαλος με βάση τις τιμές που δώσαμε παραπάνω έχουμε δώσει το περιθώριο να μην ξεκινάει το αμαξιού χωρίς να είναι αρκετά πάνω, κάτω, αριστερά η δεξιά ο μοχλός.

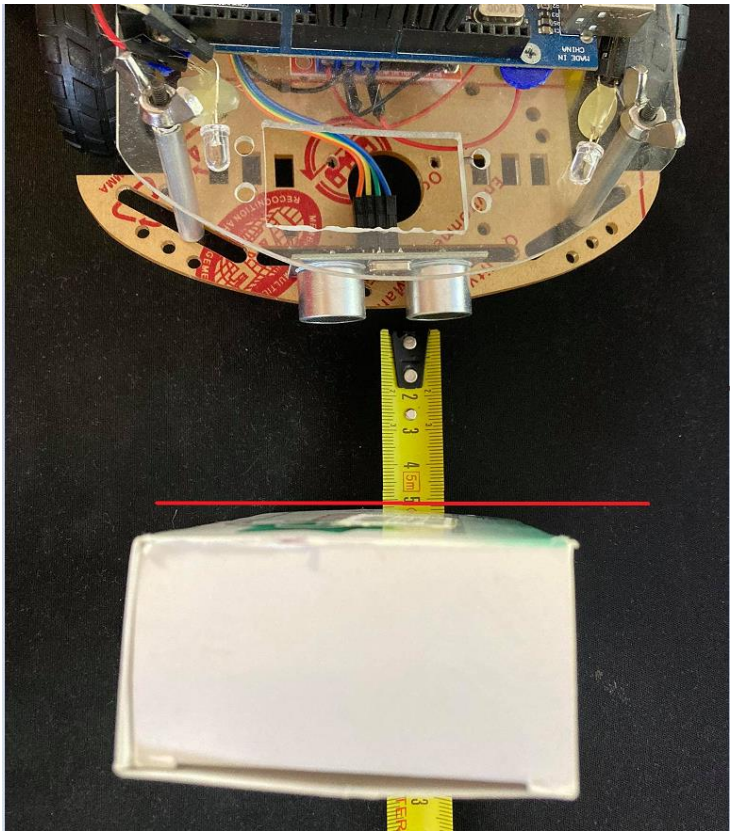
3.4 Έλεγχος σωστής λειτουργίας αναπηρικού αμαξιού

3.4.1 Έλεγχος σωστής κίνησης του αναπηρικού αμαξιού.

Για να ελέγξουμε εάν το αναπηρικό αμάξι κινείται σωστά, ξεκινήσαμε με τις 4 βασικές κινήσεις μπροστά, πίσω, αριστερά και δεξιά (ΕΙΚΟΝΑ . Κάνοντας τον μοχλό μπροστά (προς τα πάνω) θα πρέπει να κινηθεί ευθεία το αναπηρικό αμάξι. Κάνοντας πίσω τον μοχλό θα πρέπει να κάνει πίσω αντίστοιχα. Κάνοντας αριστερά το μοχλό θα πρέπει να κινηθεί αριστερόστροφα και κάνοντας το μοχλό δεξιά να κινηθεί δεξιόστροφα. Έχοντας κάνει όλα τα παραπάνω τότε ο πρώτος έλεγχος ολοκληρώνεται.

3.4.2 Έλεγχος ακρίβειας αισθητήρων HC-SR04

Για να μπορέσουμε να μετρήσουμε την ακρίβεια των αισθητήρων θα πρέπει να τοποθετήσουμε ένα εμπόδιο εσκεμμένα μπροστά από το αναπηρικό αμαξίδιο σε μια απόσταση. Ξεκινάμε λοιπόν την κίνηση προς το εμπόδιο, με το που σταματήσει το αμαξίδιο θα πρέπει να μετρήσουμε την απόσταση από το εμπόδιο. Εάν για παράδειγμα είχαμε βάλει να σταματάει τον κινητήρα στα 5cm και πάρουμε την παρακάτω μέτρηση όπως στην ΕΙΚΟΝΑ 38 τότε είμαστε εντάξει και ο αισθητήρας μας λειτουργεί κανονικά



ΕΙΚΟΝΑ 38 Μέτρηση απόστασης Αμαξιδίου από το εμπόδιο

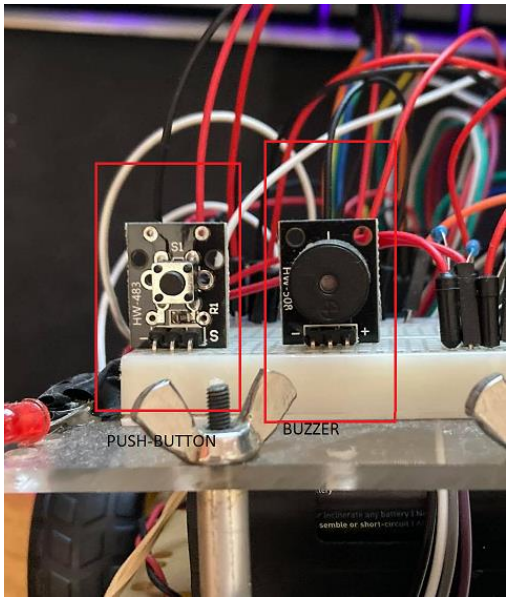
Ένας συμπληρωματικός έλεγχος ακρίβειας βάζουμε εάν εμπόδιο σε απόσταση 10cm από το αμάξι. Στην συνέχεια συνδέουμε το Arduino στον υπολογιστή. Ανοίγουμε την σειριακή οθόνη και βλέπουμε την απόσταση που μετράει ο αισθητήρας. Εάν είναι 10cm τότε ο αισθητήρας μας μετράει σωστά.

3.4.3 Έλεγχος ακινητοποίησης λόγο εμποδίων

Για να γίνει έλεγχος της αυτόματης ακινητοποίησης λόγο εμποδίου θα πρέπει να τοποθετήσουμε στο αναπηρικό αμάξι απέναντι από κάποιο εμπόδιο πχ από ένα τοίχο. Το αμάξι θα πρέπει να ξεκινήσει και μόλις φτάσει πολύ κοντά στον τοίχο θα πρέπει να ακινητοποιηθεί και να μην επιτρέψει στον χειριστή να προχωρήσει άλλο. Συνεπώς ένα το εμπόδιο είναι μπροστά, ο χειρίστης θα μπορεί να κάνει μόνο πίσω, δεξιά και αριστερά. Επίσης θα πρέπει να ακινητοποιεί σε απόσταση περίπου ίση με 5cm ή ανάλογα την απόσταση που βάλαμε στο κώδικα.

3.4.4 Έλεγχος φωτισμού και ηχητικού συναγερμού

Για να γίνει ο έλεγχος του ηχητικού συναγερμού το μόνο που πρέπει να κάνουμε είναι να πατήσουμε το Push/Button παρατεταμένα και να ακούσουμε το Buzzer. Εάν το Buzzer κάνει θόρυβο τότε λειτουργεί σωστά.



ΕΙΚΟΝΑ 49 PUSH BUTTON ΚΑΙ BUZZER

Για να ελέγξουμε ένα λειτουργεί σωστά ο φωτισμός θα πρέπει να αλλάξουμε θέση στον διακόπτη που αφορά τα φωτά. Θα πρέπει να ανάψουν τα LED η να σβήσουν αντίστοιχα.

4 ΣΥΜΠΕΡΑΣΜΑΤΑ - ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Στις μέρες μας η τεχνολογία συμβάλει όλο και περισσότερο στο να κάνει την ζωή μα πιο εύκολη με απείρους τρόπους. Ένας από αυτούς είναι με το έξυπνο καροτσάκι. Με αυτό τον τρόπο ένας άνθρωπος είναι πλέον πιο ανεξάρτητος καθώς πλέον δεν χρειάζεται κάποιον για να τον μετακινεί. Μπορεί να μετακινητέ με ασφάλεια χωρίς να ταλαιπωρείται με την χρήση μόνο ενός μοχλού. Επιπλέον, με τις εξτρά λειτουργίες που παρέχουν κάποια έξυπνα καροτσάκια (Πχ Φορτιστής για κινητό, θέσεις για τα ψώνια, καλαθάκια κλπ.) κάνουν ακόμη πιο άνετη την ζωή αυτών των ανθρώπων. Με την εξέλιξη της τεχνολογίας μπορούμε συνεχώς να κάνουμε όλο και καλύτερη την ζωή αυτών των συνάνθρωπων μας προσθέτοντας όλο και περισσότερα πράγματα και λειτουργίες στο καροτσάκι που για αυτούς είναι η ζωή τους. Στην συνέχεια θα δούμε κάποιες λύσεις που μπορούν να βοηθήσουν σε αυτό.

4.1.1 Bluetooth Χειριστήριο

Στα περισσότερα έξυπνα αναπηρικά καροτσάκια ο μοχλός είναι ενσωματωμένος πάνω στο καροτσάκι. Μία νέα έξυπνη εφαρμογή, είναι ο μοχλός(Joystick) να είναι ασύρματος έτσι ώστε να μπορεί να γίνει έλεγχος για το καροτσάκι από μακριά. Σε περίπτωση που θέλει να μετακινήσει το καροτσάκι η να το φέρει κοντά του εντελώς ανεξάρτητα και χωρίς την βοήθεια κάποιου.

4.1.2 Υπέρυθρες ελέγχου ύψους

Μια από τις καλύτερες βελτιώσεις που μπορούμε να προσθέσουμε στην συγκεκριμένη Εφαρμογή είναι η χρήση αισθητήρων που θα υπολογίζουν το ύψος. Η τοποθέτηση τους μπροστά και πίσω, θα προστατεύσουν το χειρίστη από πιθανή πτώση σε σκάλα η ένα ψηλό πεζοδρόμιο. Ουσιαστικά, μόλις ο αισθητήρας εντοπίσει το κενό αυτόματος θα κόβει την κίνηση προς την συγκεκριμένη κατεύθυνση.

4.1.3 Αποστολή SMS κινδύνου μέσω GSM

Στο χειριστήριο θα μπορούσε να τοποθετηθεί ένα εξτρά κουμπί SOS. Πατώντας παρατεταμένα το κουμπί θα στέλνει ένα SMS κίνδυνου που θα έχει καταχωρηθεί από τον προγραμματιστή με σκοπό να ενημερώσει κάποιο κοντινό πρόσωπο ότι υπάρχει κάποιο πρόβλημα η κάποιος κίνδυνος. Υπάρχουν ειδικά Module για το Arduino οπού μπορεί να τοποθετηθεί κάρτα τηλεφώνου SIM και με τον κατάλληλο προγραμματισμό να σταλεί SMS σε ένα η και περισσότερα τηλεφωνικά νούμερα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Π. ΠΑΠΑΖΟΓΛΟΥ και Σ. Π. ΛΙΩΝΗΣ, Ανάπτυξη Εφαρμογών με το Arduino, Θεσσαλονίκη: ΕΚΔΟΣΕΙΣ ΤΖΙΟΛΑ, 2016.

[2] L. M. ENGINEERS, «LAST MINUTE ENGINEERS,» [Ηλεκτρονικό]. Available: https://lastminuteengineers.com/joystick-interfacing-arduino-processing/. [Πρόσβαση 09 07 2021].

[3] «SUNFOUNDER,» 04 06 2016. [Ηλεκτρονικό]. Available: http://wiki.sunfounder.cc/index.php?title=Buzzer_modules. [Πρόσβαση 07 09 2021].

[4] «TUTORIALS POINT,» [Ηλεκτρονικό]. Available: https://www.tutorialspoint.com/arduino/arduino_dc_motor.htm. [Πρόσβαση 2021 07 09].

[5] WIKIPEDIA, «WIKIPEDIA,» 25 05 2021. [Ηλεκτρονικό]. [Πρόσβαση 2021 07 09].

[6] HANDSONTEC, «HANDSONTEC.COM,» [Ηλεκτρονικό]. Available: http://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf. [Πρόσβαση 09 07 2021].

[7] «ARDUINO.CC,» [Ηλεκτρονικό]. Available: https://www.arduino.cc/en/Guide/Introduction. [Πρόσβαση 09 07 2021].

[8] Dejan, «Ultrasonic Sensor HC-SR04 and Arduino Tutorial,» HOW TO MECHATRONICS, [Ηλεκτρονικό]. Available: https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/. [Πρόσβαση 09 07 2021].

[9] «ARDUinoauroMOTIVE,» [Ηλεκτρονικό]. Available: https://www.ardumotive.com/how-to-use-a-buzzer-en.html. [Πρόσβαση 09 07 2021].

[10] E. EMBEDDED, «Analog Joystick with Arduino,» [Ηλεκτρονικό]. Available: https://exploreembedded.com/wiki/Analog_JoyStick_with_Arduino. [Πρόσβαση 07 09 2021].

[11] «Smart Wheelchair: A Literature Review,» *International Journal of Informatics and Communication Technology (IJ-ICT)*, τόμ. 7, αρ. 2, pp. 63-65, 2018.

[12] R. C. Simpson, «Smart wheelchairs: A literature review,» *The Journal of Rehabilitation Research and Development*, τόμ. 42, αρ. 4, p. 423, 2005.

ΒΙΒΛΙΟΓΡΑΦΙΑ ΕΙΚΟΝΩΝ

Εικόνα 1 ΑΠΛΟ ΑΝΑΠΗΡΙΚΟ ΑΜΑΞΙΔΙΟ / ΕΞΥΠΝΟ ΑΝΑΠΗΡΙΚΟ ΑΜΑΞΙΔΙΟ

https://www.ortopediasilvio.com/en/folding-aluminum-wheelchairs/5356-134940-wheelchair-caneo-b.html#/

https://dealer.swemed.com/50730/wheelchairs-and-walkers/electric-powered-wheelchairs/airwheel-h3s-a-smart-power-wheelchair/

Εικόνα 2 ΑΙΣΘΗΤΗΡΕΣ INFRARED (IR)

https://electrobot.gr/Infrared-IR-sensor-receiver-module-KY-022

Εικόνα 3 ΑΙΣΘΗΤΗΡΑΣ LRF

https://www.alibaba.com/product-detail/Laser-Works-600m-LRF-laser-rangefinder_60504979219.html (GOOGLE)

Εικόνα 4 ΑΙΣΘΗΤΗΡΑΣ HC-SR04

https://www.hellasdigital.gr/electronics/sensors/ultasonic/hc-sr04-ultrasonic-module-distance-measuring-transducer-sensor-for-arduino/

Εικόνα 6 ARDUINO MEGA AT2560

https://makezine.com/2015/03/28/20-projects-celebrate-arduino-day/

Εικόνα 7 ΑΙΣΘΗΤΗΡΑΣ HC-SR04

https://www.hellasdigital.gr/electronics/sensors/ultasonic/hc-sr04-ultrasonic-module-distance-measuring-transducer-sensor-for-arduino/

Εικόνα 8 LED

https://www.hackster.io/rowan07/make-a-simple-led-circuit-ce8308

Εικόνα 9 ACTIVE BUZZER/PASSIVE BUZZER

http://wiki.sunfounder.cc/index.php?title=Buzzer_modules

Εικόνα 10 ΚΙΝΗΤΗΡΑ ΣΥΝΕΧΟΥΣ ΤΑΣΗΣ DC

https://www.tutorialspoint.com/arduino/arduino_dc_motor.htm

Εικόνα 11 ΚΙΝΗΤΗΡΑΣ SERVO

https://circuitdigest.com/article/servo-motor-working-and-basics

Εικόνα 12 ΚΙΝΗΤΗΡΑΣ ΒΗΜΑΤΙΚΟΣ (STEP)

https://www.ato.com/4-wire-bipolar-nema-17-stepper-motor-2-1v-2-1a-1-8-degree-2-phase

Εικόνα 13 ΓΕΦΥΡΑ L298N

https://grobotronics.com/dual-motor-driver-module-l298n.html

Εικόνα 14 PUSH BUTTON

https://arduinomodules.info/ky-004-key-switch-module/

Εικόνα 15 ΜΠΑΤΑΡΙΟΘΗΚΗ 4 ΘΕΣΕΩΝ ΑΑ

https://grobotronics.com/4x-with-wires.html

Εικόνα 16 ΚΙΤ ΣΥΝΑΡΜΟΛΟΓΗΣΗΣ ARDUINO CAR RC

https://www.banggood.com/2-Wheels-Ultrasonic-Smart-Robot-Car-Chassis-Tracking-Car-Kit-For-Arduino-p-1132821.html?rmmds=myorder

Εικόνα 38 ΔΙΑΓΡΑΜΜΑ JOYSTICK

https://lastminuteengineers.com/joystick-interfacing-arduino-processing/

ΠΑΡΑΡΤΗΜΑ Α - ΚΩΔΙΚΑΣ ΟΛΟΚΛΗΡΟΥ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

```
#define SwitchPin 3
int A=0,B=0;
int a=4,b=5,c=6,d=7;
int Buzzer=2;

int trigPin1=11; // BROSTA SENSOR
int echoPin1=10; // BROSTA SENSOR
int trigPin2=9; //PISW SENSOR
int echoPin2=8; //PISW SENSOR

void setup() {
  Serial.begin(9600);
  pinMode(a,OUTPUT);
  pinMode(b,OUTPUT);
  pinMode(c,OUTPUT);
  pinMode(d,OUTPUT);
  pinMode(SwitchPin, INPUT);
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);

}

void loop() {
 //SOUND SESNSOR
long duration1, distance1;
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH);
  distance1 = (duration1*0.034/2);

   if (distance1 >= 500 || distance1 <= 0){
    Serial.println("Out of range");
  }
  else {
    Serial.print ( "APOSTASH BROSTA  ");
    Serial.print ( distance1);
    Serial.println("cm");
  }
```

```
long duration2, distance2;
  digitalWrite(trigPin2, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin2, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin2, LOW);
  duration2 = pulseIn(echoPin2, HIGH);
  distance2= (duration2*0.034/2);

   if (distance2 >= 500 || distance2 <= 0){
    Serial.println("Out of range");
  }
  else {
    Serial.print("APOSTASH PISW  ");
    Serial.print(distance2);
    Serial.println("cm");
  }

  // SOUND1
  if (digitalRead(SwitchPin)==1){
    Serial.println("NOT PRESSED");
    noTone(Buzzer);
  }
   else
   {
  (digitalRead(SwitchPin)==0);
  Serial.println("PRESSED");
  tone(Buzzer,650);
   }
  //JOYSTICK
  A=analogRead(A0);
  B=analogRead(A1);

if(A>=700)
{ if (distance1>=10){
  Serial.println("MF");
  MF();
}
else
{
  MS();
}
  }
```

```
if(A<=300)
{ if (distance2>=10) {

  Serial.println("MB");
  MB();
}
else
{
  MS();
}
}
if(B>=700)
{
  Serial.println("ML");
  ML();
}
if(B<=300)
{
  Serial.println("MR");
  MR
    ();
}
if(A>=450 && A<=550 && B>=450 && B<=550)
{
  Serial.println("MS");
  MS();
}
}
void ML()
{
digitalWrite(a,1);
digitalWrite(b,0);
digitalWrite(c,0);
digitalWrite(d,1);
delay(10);
}
void MR()
{
digitalWrite(a,0);
digitalWrite(b,1);
digitalWrite(c,1);
digitalWrite(d,0);
delay(10);
}
```

```
void MF()
{
digitalWrite(a,1);
digitalWrite(b,0);
digitalWrite(c,1);
digitalWrite(d,0);
delay(10);
}
void MB()
{
digitalWrite(a,0);
digitalWrite(b,1);
digitalWrite(c,0);
digitalWrite(d,1);
delay(10);
}
void MS()
{
digitalWrite(a,0);
digitalWrite(b,0);
digitalWrite(c,0);
digitalWrite(d,0);
delay(10);
}
```

ΠΑΡΑΡΤΗΜΑ Β – ΥΛΙΚΑ ΚΑΙ ΚΟΣΤΟΣ ΥΛΙΚΩΝ

Στον παρακάτω πίνακα αναφέρονται τα εξαρτήματα που χρειαστήκαμε για την κατασκευή του αμαξιδίου. Η αγορά των υλικών πραγματοποιήθηκε 02/2021

ΠΙΝΑΚΑΣ 1 ΚΟΣΤΟΣ ΥΛΙΚΩΝ

ARDUINO AT MEGA 2560	**=20 EUR**
BRIDGE MODULE L298N	=4 EUR
X2 HC-SR04	=1.90 EUR
JOYSTICK	=3 EUR
BUZZER	=1 EUR
X2 DC MOTOR (ΧΩΡΙΣ ΡΟΔΑΚΙΑ)	=3 EUR
X2 LED RED	=0.1 EUR
X2 LED WHITE	=0.1 EUR
X4 RESISTOR (5PCS) 220Ω	=0.2 EUR
PUSH BUTTON MODULE	=1 EUR
BATTERY PACK	=1 EUR
X2 ON/OFF SWITCH	=0.5 EUR
BREADBOARD	=6 EUR
PLEXIGLASS X2	=5 EUR
JUMBER KIT (X50)	=6 EUR
X6 AA ALKALINE	=5 EUR

ΤΕΛΙΚΟ ΚΟΣΤΟΣ	=64,6 EUR

Στον παραπάνω πίνακα η τιμές είναι ενδεικτικές και αναφέρονται τα ροδάκια και οι αποστάτες. Συνολικό κόστος με μονωτική ταινία και μεταφορικά περίπου 70 Ευρώ.

Πολλά από τα παραπάνω εξαρτήματα πωλούνται στο εξωτερικό ως πακέτο (KIT).

Τα 2 KIT που αγοράστηκαν από το εξωτερικό είναι :

https://www.banggood.com/Geekcreit-45-In-1-Sensor-Module-Board-Starter-Kits-Upgrade-Version-For-Arduino-UN0-R3-MEGA2560-Plastic-Bag-Package-p-1137050.html?rmmds=myorder&cur_warehouse=CZ

https://www.banggood.com/2-Wheels-Ultrasonic-Smart-Robot-Car-Chassis-Tracking-Car-Kit-For-Arduino-p-1132821.html?rmmds=myorder&cur_warehouse=CN