

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ  
ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.



Υλοποίηση μικρού αυτόνομου ρομποτικού οχήματος με  
μικροελεγκτή Arduino .

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

της

Μάκρας Αικατερίνης ΑΕΜ 2774

Επιβλέπων: Δρ. Καλόμοιρος Ιωάννης, Αναπληρωτής Καθηγητής

Σέρρες, 2016

## ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ

“ Βεβαιώνω ότι είμαι η συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του τμήματος Μηχανικών Πληροφορικής του Τ.Ε.Ι. Σερρών. ”

# ΠΕΡΙΕΧΟΜΕΝΑ

Υπεύθυνη δήλωση.....σελ. 2

Περίληψη.....σελ. 5

## ΚΕΦΑΛΑΙΟ 1

1.1 Ρομποτική και Ρομπότ.....σελ. 6

1.2. Ιστορική αναδρομή των αυτοκινούμενων Ρομπότ.....σελ. 7

## ΚΕΦΑΛΑΙΟ 2

2.1 Τι είναι ο ARDUINO.....σελ.14

2.2 Μικροελεγκτής , η καρδιά του ARDUINO .....σελ.15

2.3 Εκδόσεις ARDUINO .....σελ.16

2.4 Χαρακτηριστικά του Arduino UNO.....σελ.18

2.4.1 Παράδειγμα εύρεσης του παλμού λειτουργίας duty cycle (κύκλος).....σελ.21

2.4.2 Τροφοδοσία .....σελ.21

2.4.3 Ενσωματωμένα κουμπιά και LED του Arduino UNO.....σελ.22

2.5 Εγκατάσταση περιβάλλοντος Arduino IDE.....σελ.22

## ΚΕΦΑΛΑΙΟ 3

3.1.Το ολοκληρωμένο κύκλωμα οδήγησης  
κινητήρων.....σελ.27

3.1.1 Ιστορικά στοιχεία των ολοκληρωμένων κυκλωμάτων.....σελ.27

3.1.2 Διάκριση ολοκληρωμένων κυκλωμάτων.....σελ.28

3.2 Το ολοκληρωμένο κύκλωμα L293D.....σελ.29

3.2.1 Κινητήρες- εισαγωγή.....σελ.29

3.2.2 Κινητήρας DC.....σελ.30

3.2.3 Συνδεσμολογία για τον έλεγχο δύο κινητήρων με το ολοκληρωμένο  
κύκλωμα.....σελ.31

<b>3.3</b> Κίνηση dc μοτέρ με τη χρήση ολοκληρωμένου κυκλώματος .....	<b>σελ.32</b>
<b>3.3.1</b> H-Bridge.....	<b>σελ.33</b>
<b>3.3.2</b> Συνδεσμολογία Arduino με το ολοκληρωμένο κύκλωμα L293D.....	<b>σελ.34</b>

## **ΚΕΦΑΛΑΙΟ 4**

<b>4.1</b> Εισαγωγή.....	<b>σελ.37</b>
<b>4.2</b> Ο αισθητήρας υπερήχων HC-SR04 .....	<b>σελ.38</b>
<b>4.3</b> Σύνδεση με το Arduino .....	<b>σελ.40</b>

## **ΚΕΦΑΛΑΙΟ 5**

<b>5.1</b> Ηχείο- Sounder.....	<b>σελ.41</b>
<b>5.2</b> Σύνδεση ηχείου στο Arduino .....	<b>σελ.41</b>

## **ΚΕΦΑΛΑΙΟ 6**

<b>6.1</b> Αρχικοποίηση Μεταβλητών.....	<b>σελ.42</b>
<b>6.2</b> Κώδικας που εκτελείται μια φορά .....	<b>σελ.44</b>
<b>6.3</b> Κυρίως πρόγραμμα.....	<b>σελ.45</b>
<b>6.3.1</b> Εντολές εντός κυρίου προγράμματος για την δήλωση του ήχου όταν όχημα πλησιάζει σε εμπόδιο .....	<b>σελ.55</b>
Διάφορες εικόνες που τραβήχτηκαν κατά την διαδικασία υλοποίησης της εργασίας.....	<b>σελ.54</b>
Προβλήματα που παρουσιάστηκαν κατά την υλοποίηση του αυτόνομου ρομποτικού αυτοκινήτου .....	<b>σελ.62</b>
Σχόλια και συμπεράσματα.....	<b>σελ.63</b>
Παράρτημα Υλικών .....	<b>σελ.64</b>
Βιβλιογραφία.....	<b>σελ.68</b>
Ιστοσελίδες.....	<b>σελ.68</b>

## **ΠΕΡΙΛΗΨΗ**

Η παρούσα πτυχιακή εργασία έχει σαν στόχο την ανάπτυξη και την δυνατότητα διδασκαλίας της εκπαιδευτικής ρομποτικής κυρίως με τη χρήση του ARDUINO με αισθητήρες και ήχο πάνω σε ένα αυτοκίνητο που θα είναι αυτόνομο. Στο πλαίσιο αυτής της διπλωματικής εργασίας χρησιμοποιείται ένα ολοκληρωμένο κύκλωμα L293D για την κίνηση των κινητήρων. Με τη βοήθεια του μικροεπεξεργαστή atmega328p (ARDUINO) προγραμματίζουμε την ομαλή λειτουργία του οχήματος. Με τη χρήση των αισθητήρων και των ηχητικών εφέ επιτυγχάνεται η αποφυγή εμποδίων που θα βρίσκονται στο δρόμο του αυτοκινήτου μας.

# ΚΕΦΑΛΑΙΟ 1

## 1.1 Ρομποτική και Ρομπότ

**Ρομποτική** είναι ο σύγχρονος τεχνολογικός κλάδος της αυτοματοποίησης, που έχει ως αντικείμενο τη μελέτη, το σχεδιασμό και τη λειτουργία των ρομπότ, καθώς και την έρευνα για την περαιτέρω ανάπτυξή τους.

**Ρομπότ** είναι μια μηχανική συσκευή η οποία μπορεί να υποκαθιστά τον άνθρωπο σε διάφορες εργασίες. Ένα ρομπότ μπορεί να δράσει κάτω από τον απευθείας έλεγχο ενός ανθρώπου ή αυτόνομα κάτω από τον έλεγχο ενός προγραμματισμένου υπολογιστή. Η λέξη ρομπότ προέρχεται από το σλαβικό *robota* που σημαίνει εργασία.

Η φιλοδοξία του ανθρώπου να δημιουργήσει μηχανές που θα του μοιάζουν τόσο στη μορφή όσο και στη λειτουργία, πρωτοσυναντάται στην Ελληνική Μυθολογία όπου ο Τάλως, ο μυθικός χάλκινος γίγαντας που κατασκεύασε ο Ήφαιστος για να προστατεύει την Κρήτη από τους εισβολείς, αποτελεί το πρώτο ‘‘αυτόματο’’ στην ανθρώπινη ιστορία.

Στη σύγχρονη εποχή, ο όρος καθιερώθηκε με την σημερινή του έννοια, το 1921 από τον Τσέχο θεατρικό συγγραφέα Karel Capek στο έργο του *Rossum's Universal Robots*, όπου σατιρίζει την εξάρτηση της κοινωνίας από τους μηχανικούς εργάτες (ρομπότ) της τεχνολογικής εξέλιξης και που τελικά εξοντώνουν τους δημιουργούς τους.

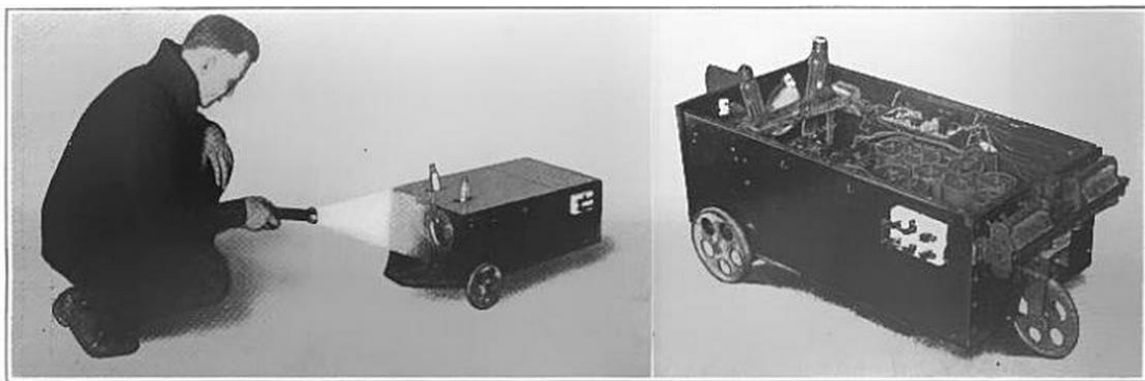
Λίγα χρόνια αργότερα, κατά την δεκαετία του '40, ο Ρώσος συγγραφέας επιστημονικής φαντασίας, Isaak Asimov, συνέλαβε το ρομπότ ως ένα αυτόματο με εμφάνιση ανθρώπου, αλλά απαλλαγμένο από συναισθήματα. Ο όρος ρομποτική χρησιμοποιήθηκε από τον Asimov ως το σύμβολο της επιστήμης που είναι αφιερωμένη στη μελέτη των ρομπότ και διέπονται από τους παρακάτω τρεις βασικούς νόμους:

1. Το ρομπότ δε θα κάνει κακό σε άνθρωπο , ούτε με την αδράνεια του θα επιστρέψει να βλάψει κανείς ανθρώπινο ον .
2. Το ρομπότ πρέπει να υπακούει στις διαταγές που του δίνουν οι άνθρωποι, εκτός αν αυτές οι διαταγές έρχονται σε αντίθεση με τον πρώτο νόμο.
3. Το ρομπότ οφείλει να προστατεύει τη ύπαρξή του, εφόσον αυτό δεν συγκρούεται με τον πρώτο ή το δεύτερο νόμο.

## 1.2. Ιστορική αναδρομή των αυτοκινούμενων Ρομπότ

Στην παράγραφο αυτή θα κάνουμε μια σύντομη ιστορική αναδρομή στην ιστορία των αυτοκινούμενων ρομπότ, η οποία και έχει στιγματίσει την πορεία της τεχνολογίας και της επιστήμης από τα μέσα του 20<sup>ου</sup> αιώνα μέχρι και σήμερα. Η εκτέλεση των κινούμενων ρομπότ προέκυψε με σκοπό την επιτέλεση διάφορων λειτουργιών, από την εκτέλεση απλών καθημερινών εργασιών μέχρι την διενέργεια πολύπλοκων ερευνών, ενώ σήμερα χρησιμοποιούνται και στις πλέον μεγάλης σημασίας διαστημικές αποστολές.

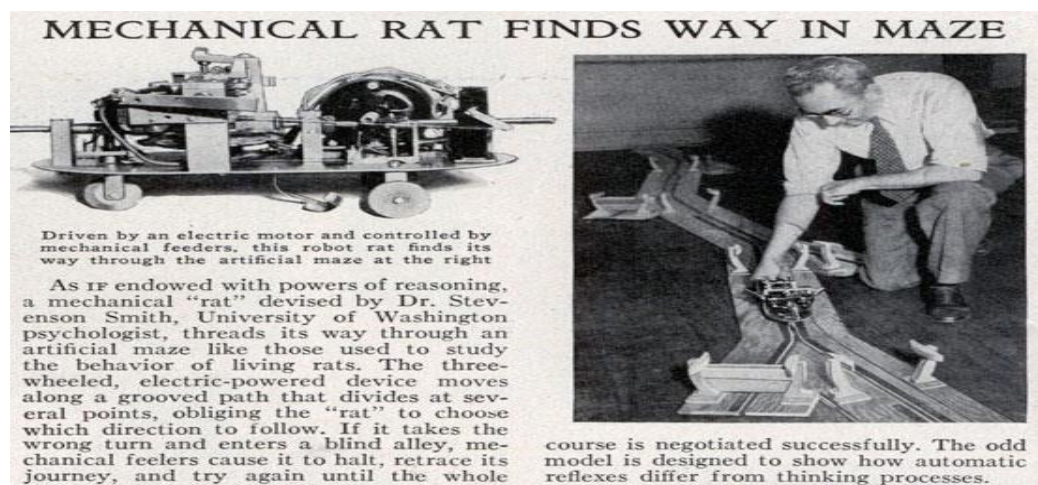
**Electric Dog (1912)** Δημιουργήθηκε από τους ερευνητές John Hammond, Jr και Benjamin Miessner στις ΗΠΑ και αποτέλεσε πρόδρομο όλων των φωτοτροπικών αυτόματων ρομπότ.



The Electric Dog following its luminous master, and a general view of its internal arrangement

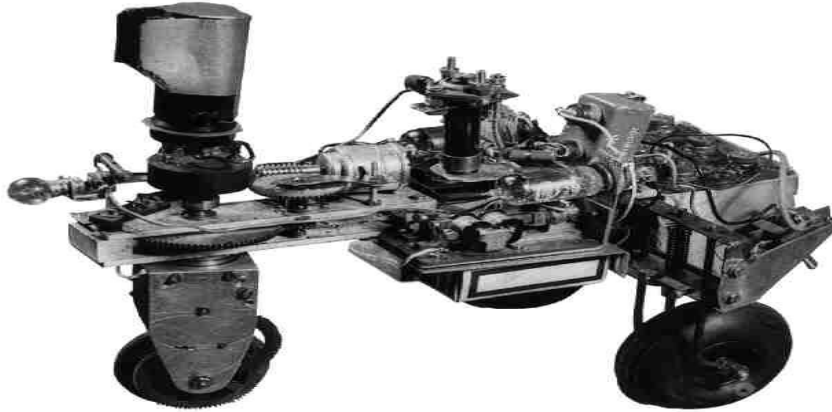
εικ. 1.1 Electric Dog

**Maze Machine Learning - Ross Thomas (1933)** Πρόκειται για έναν μηχανικό αρουραίο ο οποίος δημιουργήθηκε από τον Dr. Stevenson Smith του πανεπιστημίου Ουάσινγκτον.



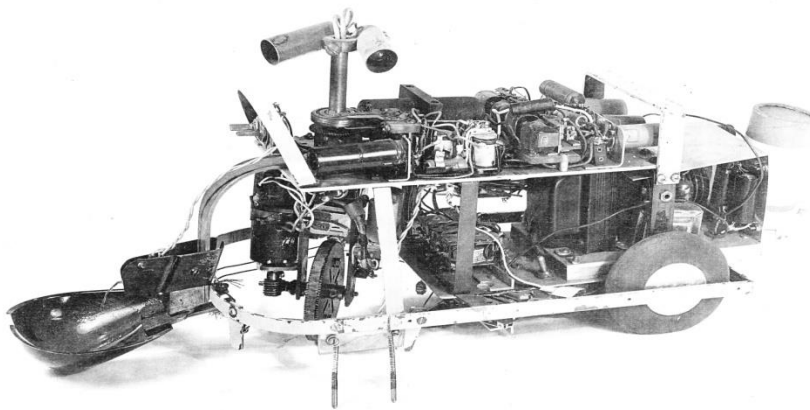
εικ. 1.2 Maze Machine Learning - Ross Thomas

**Elmer & Elsie (1948-1949)** Δύο ρομπότ που κατασκευάστηκαν από τον William Grey Walter και ονομάστηκαν Machina Speculatrix επειδή τους άρεσε να εξερευνούν το περιβάλλον τους, χαρακτηρίστηκαν ως χελώνες λόγω σχήματος και της αργής κίνησής τους.



εικ. 1.3 Elsie

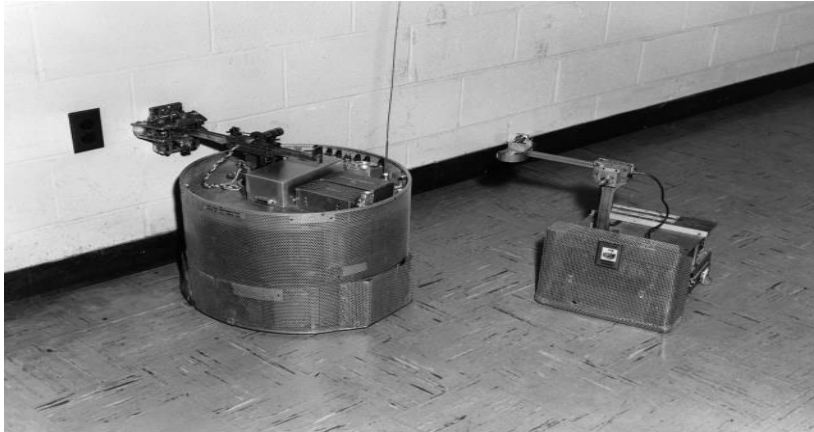
**Squee Electric Squirrel (1951)** Σχεδιάστηκε και κατασκευάστηκε από τον Edmund C. Berkeley. Ήταν ένα μικρό ηλεκτρονικό ρομπότ που περισυνέλεγε ‘καρύδια’ (μπάλες τένις ή γκολφ). Ήταν το πρώτο επιδέξιο ρομπότ υπό αυτόματο έλεγχο και το πρώτο αληθινό ρομπότ.



εικ. 1.4 Squee Electric Squirrel

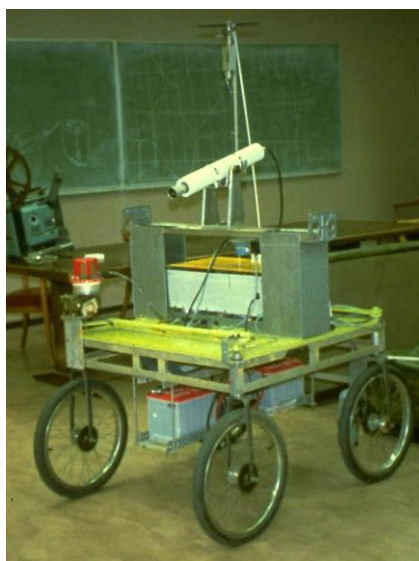


**Beast (1960)** Το Johns Hopkins University ανέπτυξε το ρομπότ αυτό το οποίο ήταν ένα κινητό αυτόνομο που μπορούσε να επιβιώσει σε ένα φυσικό περιβάλλον χωρίς καμιά ανθρώπινη βοήθεια.



εικ.1.5 Beast

**Stanford Cart (1961)** Κατασκευάστηκε αρχικά από τον James L. Adams ως ερευνητικό όχημα για απομακρυσμένες αποστολές στο φεγγάρι. Είχε τέσσερις μικρές ρόδες ποδηλάτου που συνδέονταν με ηλεκτρικούς κινητήρες και τροφοδοτούνταν από μια μπαταρία αυτοκινήτου. Επίσης πραγματοποιούσε λήψεις με μια τηλεοπτική κάμερα με σταθερή θέα προς τα εμπρός.



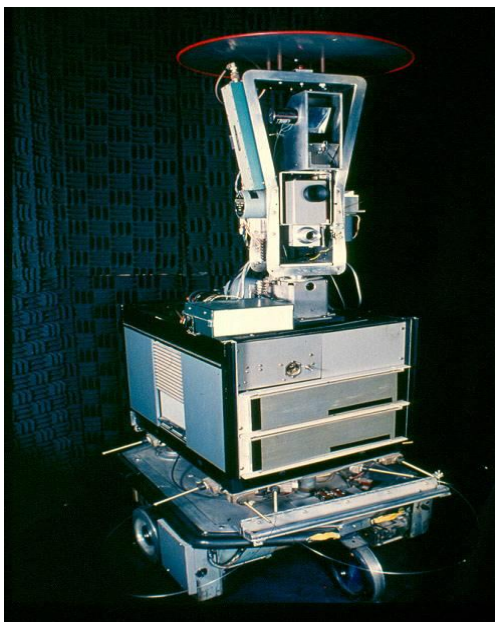
εικ. 1.6 Stanford Cart (αρχική μορφή)

**Stanford Cart (1964- 1971)** Μέχρι το 1966 το ρομπότ αυτό ήταν αχρησιμοποίητο σε ένα εργαστήριο, ώσπου ο Les Ernest βρήκε το δημιουργό του και του μίλησε, αφήνοντας το SAIL (Stanford AI Lab- σίτι για τους ερευνητές στο Στάνφορντ, Τμήμα Επιστήμης Υπολογιστών του οποίου η κύρια εστίαση της έρευνας είναι η Τεχνητή Νοημοσύνη) να χρησιμοποιήσει και να δοκιμάσει την περιήγηση στο δρόμο υπό τον έλεγχο του υπολογιστή χρησιμοποιώντας αναφορές οπτικής επαφής.



εικ 1.7 Stanford Cart (εξελιγμένη μορφή)

**Shakey (1966-1972)** Κατασκευάστηκε μετά από έρευνα που διενεργήθηκε στο ερευνητικό κέντρο του πανεπιστημίου Stanford και ονομάστηκε έτσι εξαιτίας της σπασμωδικής του κίνησης. Ήταν το πρώτο ρομπότ στο οποίο δίνονταν πολύ απλές εντολές (ανοιγόκλεινε πόρτες και διακόπτες) κι εκείνο αντιλαμβανόταν τα απαραίτητα βήματα που έπρεπε να κάνει.



εικ. 1.8 Shakey

**Timbug (1980)** Ίσως ο πρώτος σχεδιασμός που δημοσιεύτηκε για ένα ρομποτικό όχημα που χρησιμοποιούσε αισθητήρες υπερήχων για την ανίχνευση εμποδίων, αποφεύγοντάς τα, κάνοντας τυχαία όπισθεν και αλλάζοντας κατεύθυνση.



εικ. 1.9 Timbug

**Sojourner (1996-1997)** Η NASA στέλνει στον Άρη το Mars Pathfinder μαζί με το ρόβερ του, το Sojourner , έχοντας ως σκοπό την εξερεύνηση της επιφάνειάς του δεχόμενο εντολές από τη Γη. Ήταν το πρώτο ρόβερ που πάτησε στην επιφάνεια του Άρη, ήταν εξοπλισμένο με κάμερες (μπρος και πίσω) και με σύστημα αποφυγής κινδύνων.



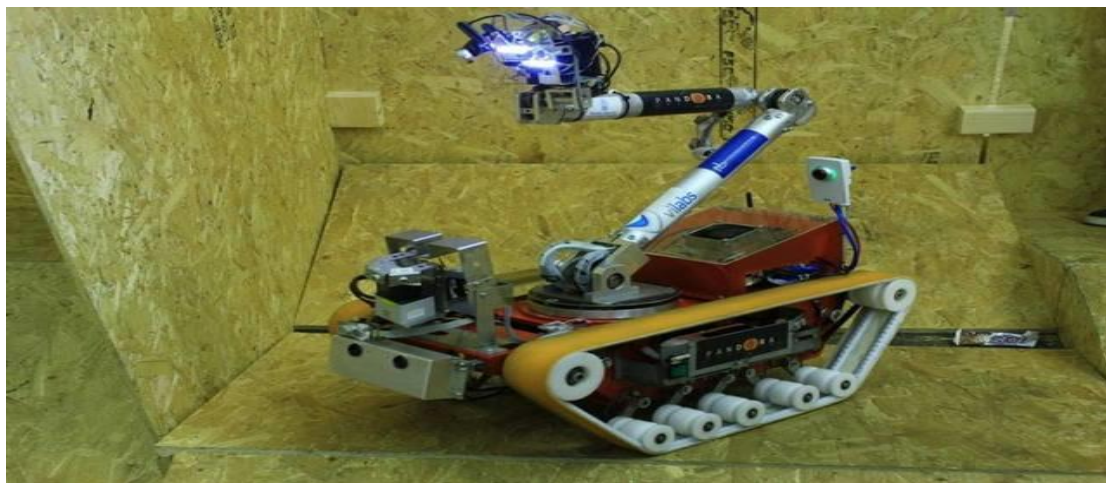
εικ.1.10 Sojourner

**Roomba (2002)** Είναι μια σειρά από αυτόνομα ρομπότ που λειτουργούν σαν ηλεκτρικές σκούπες. Υπό κανονικές συνθήκες στο σπίτι, είναι σε θέση να κινείται αυτόνομα στο πάτωμα αποφεύγοντας τυχόν εμπόδια.



εικ. 1.11 Roomba

**Pandora Robot (2013)** Κατασκευαστής του είναι ο έλληνας φοιτητής του Α.Π.Θ., Άρης Θάλας από τη Λαμία, μέλος της ομάδας Pandora που κατέκτησε τη 2<sup>η</sup> θέση παγκοσμίως στο διαγωνισμό RobocupRescue2013. Ως αντικείμενό του έχει την εξερεύνηση ενός εκ των προτέρων αγνώστου χώρου με απώτερο σκοπό να εντοπίζει πιθανά θύματα μετά από κάποια φυσική καταστροφή.



εικ. 1.12 Pandora Robot

Όλα τα παραπάνω ρομπότ που εξετάσαμε ως αναδρομή στο παρελθόν και στην ιστορία της ρομποτικής, βλέπουμε ότι είναι είτε αυτόνομα είτε χειριζόμενα από τρίτους και χρήσιμα στο να συλλέγουν πληροφορίες ή αντικείμενα. Υπάρχει πληθώρα κατασκευών τέτοιου είδους που για ευνόητους λόγους δεν έχουν αναφερθεί παραπάνω, παρά μόνο παρουσιάστηκαν όσα αποτέλεσαν σταθμό στην ιστορία της τεχνολογίας και της επιστήμης.

## ΚΕΦΑΛΑΙΟ 2

### 2.1 Τι είναι ο ARDUINO

Όπως το περιγράφει ο δημιουργός του, ο Arduino είναι μια «ανοικτού κώδικα» πλατφόρμα «πρωτοτυποποίησης» ηλεκτρονικών βασισμένη σε ευέλικτο και εύκολο στη χρήση hardware και software που προορίζεται για οποιονδήποτε έχει λίγη προγραμματιστική εμπειρία, στοιχειώδεις γνώσεις ηλεκτρονικών και ενδιαφέρεται να δημιουργήσει διαδραστικά αντικείμενα ή περιβάλλοντα.

Στην ουσία, πρόκειται για ένα ηλεκτρονικό κύκλωμα που βασίζεται στον μικροελεγκτή ATmega της Atmel και του οποίου όλα τα σχέδια, καθώς και το software που χρειάζεται για την λειτουργία του, διανέμονται ελεύθερα και δωρεάν ώστε να μπορεί να κατασκευαστεί από τον καθένα. Αφού κατασκευαστεί, μπορεί να συμπεριφερθεί σαν ένας μικροσκοπικός υπολογιστής, αφού ο χρήστης μπορεί να συνδέσει επάνω του πολλαπλές μονάδες εισόδου/εξόδου και να προγραμματίσει τον μικροελεγκτή να δέχεται δεδομένα από τις μονάδες εισόδου, να τα επεξεργάζεται και να στέλνει κατάλληλες εντολές στις μονάδες εξόδου. Μάλιστα κάποιος θα μπορούσε να ισχυριστεί - και θα ήταν ένας αρκετά πετυχημένος παραλληλισμός - ότι λειτουργικά το Arduino μοιάζει πολύ με το NXT Brick των Lego Mindstorms NXT. Άλλωστε η ρομποτική είναι μια από τις πολλές εφαρμογές στις οποίες ο Arduino διαπρέπει.

Ο Arduino βέβαια, δεν είναι ούτε ο μοναδικός, ούτε και ο καλύτερος δυνατός τρόπος για την δημιουργία μιας οποιασδήποτε διαδραστικής ηλεκτρονικής συσκευής. Όμως το κύριο πλεονέκτημά του είναι η τεράστια κοινότητα που το υποστηρίζει και η οποία έχει δημιουργήσει, συντηρεί και επεκτείνει μια ανάλογου μεγέθους online γνωσιακή βάση. Έτσι, παρότι ένας έμπειρος ηλεκτρονικός μπορεί να προτιμήσει διαφορετική πλατφόρμα ή εξαρτήματα ανάλογα με την εφαρμογή που έχει στον νου του, ο Arduino, με το εκτενές documentation, καταφέρνει να κερδίσει όλους αυτούς των οποίων οι γνώσεις στα ηλεκτρονικά περιορίζονται στα όσα λίγα έμαθαν στο σχολείο.

Ακριβώς επειδή απευθύνεται κυρίως σε αρχάριους των ηλεκτρονικών και επειδή, παρά τις αναλυτικότερες οδηγίες που υπάρχουν, δεν έχουν όλοι τις γνώσεις και τα μέσα να κατασκευάσουν μια ηλεκτρονική πλακέτα, κυκλοφορούν έτοιμες, προκατασκευασμένες πλακέτες Arduino που μπορείτε να προμηθευτείτε σε πολύ οικονομική τιμή. Με λίγα χρήματα παραπάνω μάλιστα, οι περισσότεροι προμηθευτές διαθέτουν Arduino Starter Kit, τα οποία, εκτός από το ίδιο ο Arduino, περιέχουν διάφορα άλλα εξαρτήματα και εργαλεία που μπορεί να χρειαστείτε για τις πρώτες σας εφαρμογές (όπως το απαραίτητο καλώδιο USB για την σύνδεση με τον υπολογιστή, ράστερ, καλώδια, LED, διακόπτες, ποτενσιόμετρα, αντιστάσεις, διόδους, τρανζίστορ κ.λπ.).



Εικ.2.1 Arduino UNO

## 2.2 Μικροελεγκτής, η καρδιά του ARDUINO

Ο Arduino βασίζεται στον ATmega328, έναν 8-bit RISC μικροελεγκτή, τον οποίο χρονίζει στα 16MHz. Ο ATmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων:

- a) 2Kb μνήμης SRAM που είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματά σας για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. κατά το runtime. Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στον Arduino σταματήσει ή αν γίνει reset.
- b) 1Kb μνήμης EEPROM η οποία μπορεί να χρησιμοποιηθεί για «ωμή» εγγραφή/ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματά σας κατά το runtime. Σε αντίθεση με την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset οπότε είναι το ανάλογο του σκληρού δίσκου
- c) 32Kb μνήμης Flash, από τα οποία τα 2Kb χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware αυτό που στην ορολογία του Arduino ονομάζεται bootloader είναι αναγκαίο

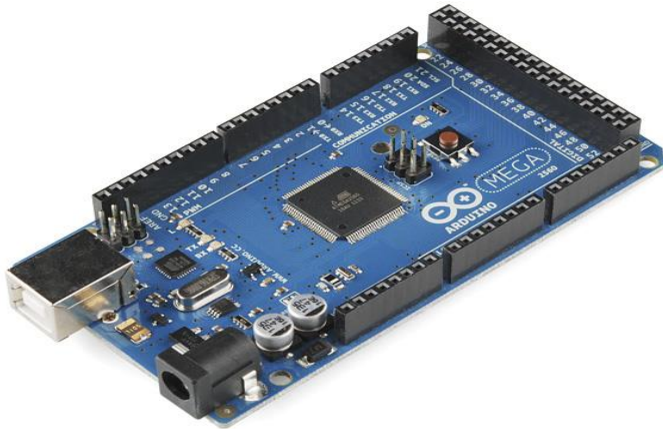
για να εγκαταστήσετε τα δικά σας προγράμματα στον μικροελεγκτή μέσω της θύρας USB, χωρίς δηλαδή να χρειάζεται εξωτερικός hardware programmer. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή σας. Η μνήμη Flash, όπως και η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset. Επίσης, ενώ η μνήμη Flash υπό κανονικές συνθήκες δεν προορίζεται για χρήση runtime μέσα από τα προγράμματά σας, λόγω της μικρής συνολικής μνήμης που είναι διαθέσιμη σε αυτά (2Kb SRAM + 1Kb EEPROM), έχει σχεδιαστεί μια βιβλιοθήκη που επιτρέπει την χρήση όσου χώρου περισσεύει (30Kb μείον το μέγεθος του προγράμματός σας σε μεταγλωττισμένη μορφή).

## 2.3 Εκδόσεις ARDUINO

Στη συνέχεια μπορείτε να δείτε τα γενικά χαρακτηριστικά από τις πιο διαδεδομένες εκδόσεις Arduino που χρησιμοποιούνται σήμερα.

Μικροελεγκτής	ATmega328(arduino UNO)	ATmega2560(arduino mega 2560)	ATmega32u4(arduino lilypad)
Τάση λειτουργίας	5v	5v	3.3v
Τάση εξωτερικής τροφοδοσίας	7-12v	7-12v	3.8v-5v
Όρια τάσης εξωτερικής τροφοδοσίας	6-20v	6-20v	
Ψηφιακές εισοδοι και έξοδοι (μέσω ακροδεκτών)	14(εκ των οποίων οι 6 υποστηρίζουν PWM)	54(εκ των οποίων οι 15 υποστηρίζουν PWM)	9 (εκ των οποίων υποστηρίζει 4 κανάλια PWM)
Αναλογικές εισοδοι(μέσω ακροδεκτών)	6	16	4
Μέγιστο συνεχές ρεύμα ανά ακροδέκτη εισόδου/εξόδου	40mA	40mA	40mA
Μέγιστο ρεύμα για τον ακροδέκτη 3,3v	50mA	50mA	
Flash Memory	32KB(ATmega328)εκ των οποίων τα 0.5 KB χρησιμοποιούνται από τον bootloader	256KB εκ των οποίων τα 8 kB χρησιμοποιούνται από τον bootloader	32KB(ATmega328)εκ των οποίων τα 4 KB χρησιμοποιούνται από τον bootloader
SRAM	2KB	8KB	2.5KB
EEPROM	1KB	4KB	1KB
Ταχύτητα ρολογιού	16MHz	16MHz	8MHz

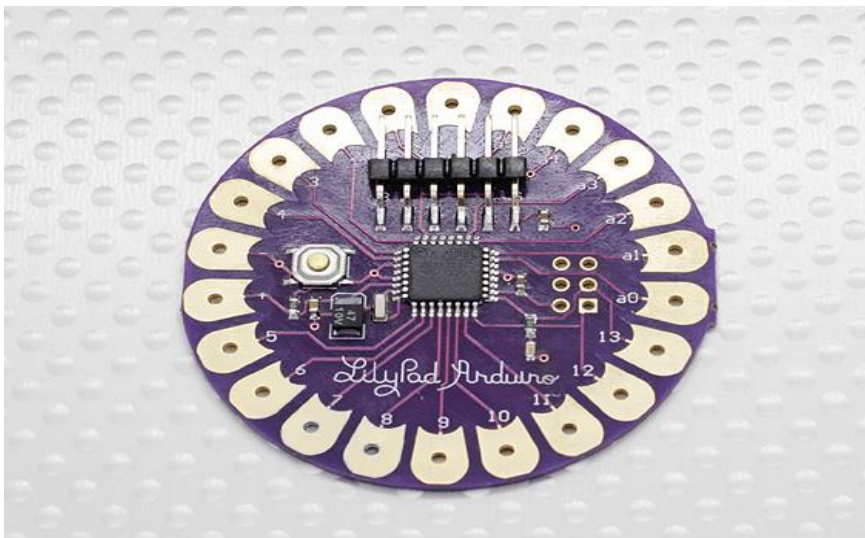




Εικ. 2.2 Arduino Mega 2560

Στην εικόνα 2.2 και σύμφωνα με τα στοιχεία που προβάλλονται στον παραπάνω πίνακα φαίνεται η έκδοση Mega 2560 στην οποία χρησιμοποιείται διαφορετικός τύπος και φυσική μορφή μικροελεγκτή που διαθέτει μεγαλύτερο πλήθος ακροδεκτών σε σχέση με την έκδοση UNO.

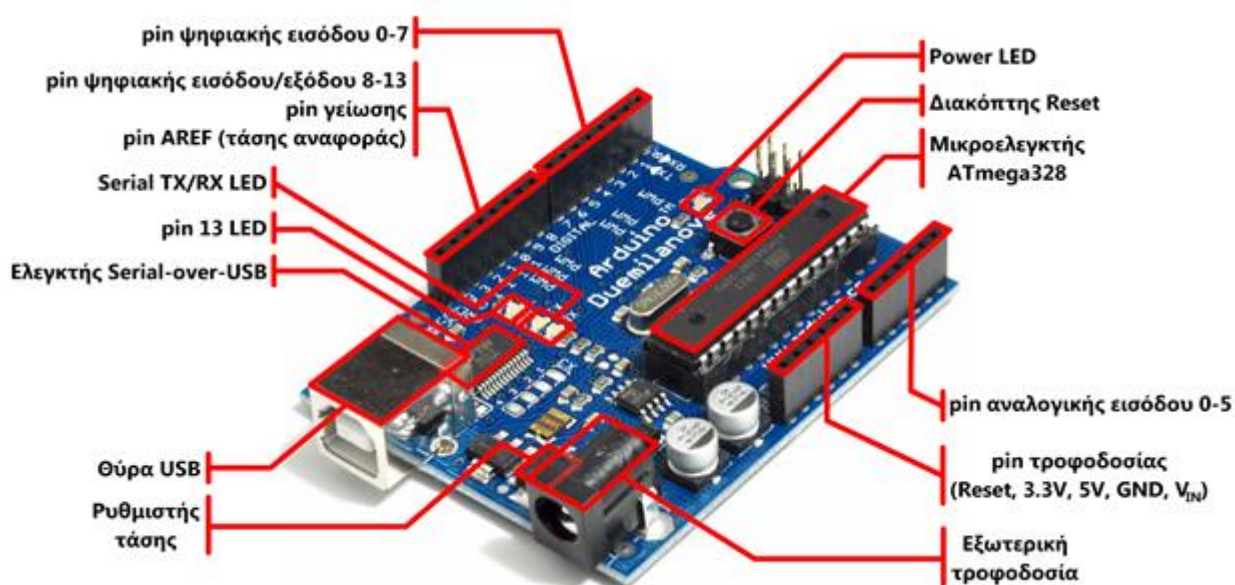
Μια ειδική έκδοση Arduino είναι το Lilypad το οποίο χρησιμοποιείται σε ορισμένες ειδικές εφαρμογές (στον τομέα της υγείας) του οποίου τα χαρακτηριστικά και οι διαφορές από τους άλλους Arduino παρατίθενται στον παραπάνω πίνακα.



Εικ. 2.3 Arduino Lilypad

## 2.4 Χαρακτηριστικά του Arduino UNO

Καταρχήν ο Arduino διαθέτει σειριακό interface. Ο μικροελεγκτής ATmega υποστηρίζει σειριακή επικοινωνία, την οποία ο Arduino προωθεί μέσα από έναν ελεγκτή Serial-over-USB ώστε να συνδέεται με τον υπολογιστή μέσω USB. Η σύνδεση αυτή χρησιμοποιείται για την μεταφορά των προγραμμάτων που σχεδιάζονται από τον υπολογιστή στο Arduino αλλά και για αμφίδρομη επικοινωνία του Arduino με τον υπολογιστή μέσα από το πρόγραμμα την ώρα που εκτελείται.



εικ.2.4 Ο Arduino

Επιπλέον, στην πάνω πλευρά του Arduino βρίσκονται 14 θηλυκά pin, αριθμημένα από 0 ως 13, που μπορούν να λειτουργήσουν ως ψηφιακές εισοδοι και έξοδοι. Λειτουργούν στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί το πολύ 40mA. Ως ψηφιακή έξοδος, ένα από αυτά τα pin μπορεί να τεθεί από το πρόγραμμά σας σε κατάσταση HIGH ή LOW, οπότε ο Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι ρεύμα στο συγκεκριμένο pin. Με αυτόν τον τρόπο μπορείτε λόγω χάρη να ανάψετε και να σβήσετε ένα LED που έχετε συνδέσει στο συγκεκριμένο pin. Αν πάλι ρυθμίσετε ένα από αυτά τα pin ως ψηφιακή είσοδο μέσα από το πρόγραμμά σας, μπορείτε με την κατάλληλη εντολή να διαβάσετε την κατάστασή του (HIGH ή LOW) ανάλογα με το αν η εξωτερική συσκευή που έχετε συνδέσει σε αυτό το pin διοχετεύει ή όχι ρεύμα στο pin (με αυτόν τον τρόπο λόγω χάρη μπορείτε να «διαβάζετε» την κατάσταση ενός διακόπτη).

Μερικά από αυτά τα 14 pin, εκτός από ψηφιακές εισοδοι/έξοδοι έχουν και δεύτερη λειτουργία.

Συγκεκριμένα:

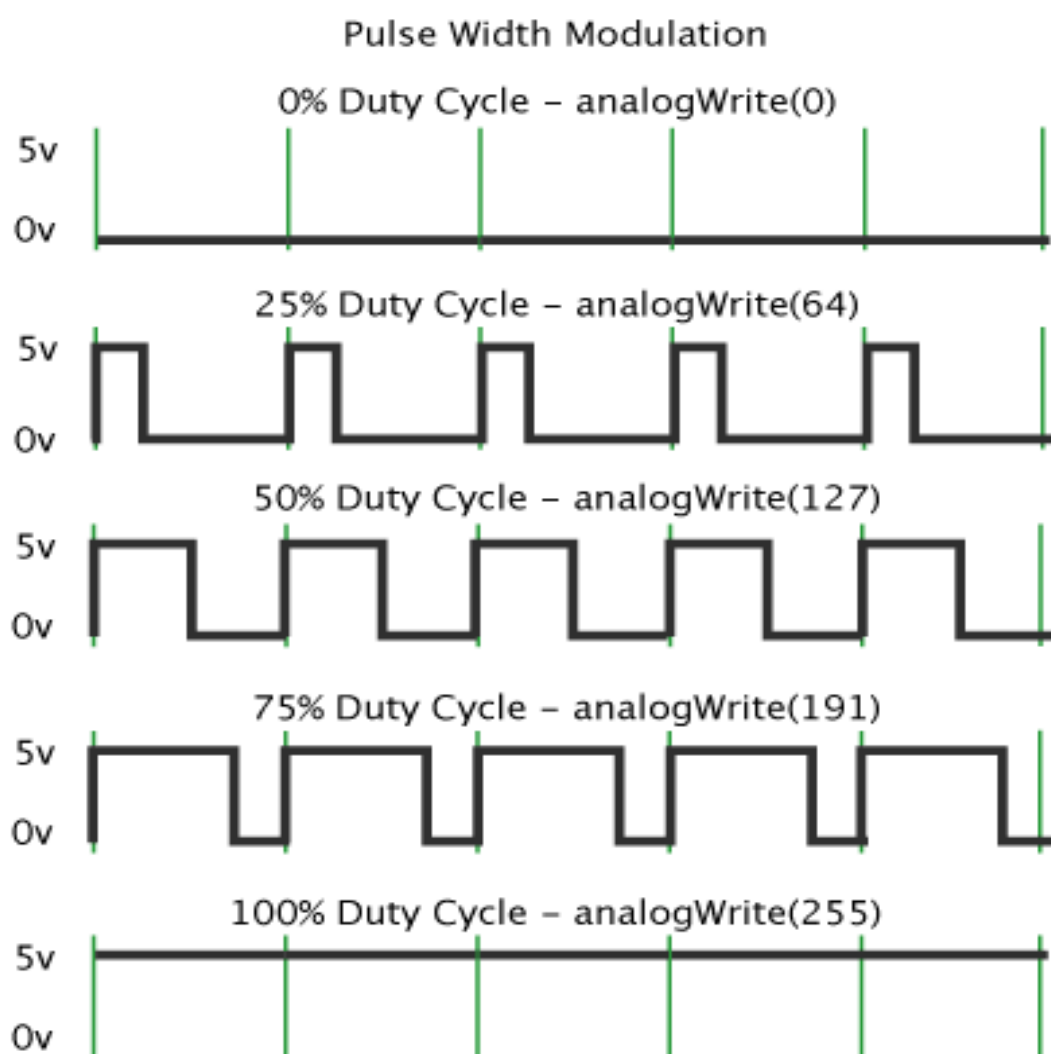
- Τα pin 0 και 1 λειτουργούν ως RX και TX της σειριακής όταν το πρόγραμμά σας ενεργοποιεί την σειριακή θύρα. Έτσι, όταν λόγω χάρη το πρόγραμμά σας στέλνει δεδομένα στην σειριακή, αυτά προωθούνται και στην θύρα USB μέσω του ελεγκτή Serial-Over-USB αλλά και στο pin 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή (π.χ. έναν δεύτερο Arduino στο δικό του pin 1). Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμά σας ενεργοποιήσετε το σειριακό interface, χάνετε 2 ψηφιακές εισόδους/εξόδους.

- Τα pin 2 και 3 λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Με άλλα λόγια, μπορείτε να τα ρυθμίσετε μέσα από το πρόγραμμά σας ώστε να λειτουργούν αποκλειστικά ως ψηφιακές εισοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει \*άμεσα\* και εκτελείται μια συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

- Τα pin 3, 5, 6, 9, 10 και 11 μπορούν να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι με το σύστημα PWM (Pulse Width Modulation), δηλαδή το ίδιο σύστημα που διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων. Έτσι, μπορείτε να συνδέσετε λόγω χάρη ένα LED σε κάποιο από αυτά τα pin και να ελέγξετε πλήρως την φωτεινότητά του με ανάλυση 8bit (256 καταστάσεις από 0-σβηστό ως 255-πλήρως αναμμένο) αντί να έχετε απλά την δυνατότητα αναμμένο-σβηστό που παρέχουν οι υπόλοιπες ψηφιακές εξοδοι. Είναι σημαντικό να καταλάβετε ότι το PWM δεν είναι πραγματικά αναλογικό σύστημα και ότι θέτοντας στην έξοδο την τιμή 127, δεν σημαίνει ότι η έξοδος θα δίνει 2.5V αντί της κανονικής τιμής των 5V, αλλά ότι θα δίνει ένα παλμό που θα εναλλάσσεται με μεγάλη συχνότητα και για ίσους χρόνους μεταξύ των τιμών 0 και 5V.

Όσον αφορά, τώρα, την συγκεκριμένη εργασία, επιτυγχάνουμε διαφορετική ταχύτητα στον κινητήρα όταν αλλάζουμε το πλάτος (το χρόνο) του παλμού. Υπάρχουν πέντε στάνταρ καταστάσεις 0%, 25%, 50%, 75% και 100% (κύκλοι λειτουργίας). Με απλά λόγια όσο μεγαλύτερο χρόνο θέτουμε στο θετικό μέτωπο του παλμού (πλάτος) τόσο μεγαλύτερη ταχύτητα έχουμε στον κινητήρα.

Στο παρακάτω γράφημα οι πράσινες γραμμές αντιπροσωπεύουν μια κανονική χρονική περίοδο. Αυτή η διάρκεια ή η περίοδος είναι το αντίστροφο της συχνότητας PWM. Μέσα από τη βιβλιοθήκη του προγράμματος του Arduino χρησιμοποιούμε το `analogWrite` που έχει κλίμακα από 0-255, με άλλα λόγια μια συχνότητα PWM περίπου 500Hz. Έτσι αν `analogWrite(255)` είναι ένας κύκλος εργασίας 100% (πάντα on), αν `analogWrite(127)` είναι ένας κύκλος 50% (στο μισό χρόνο). Αυτό εξηγείται στο γράφημα, όπως επίσης και για τις υπόλοιπες καταστάσεις.



εικ. 2.5 Pulse Width Modulation (PWM) - γράφημα

## 2.4.1 Παράδειγμα εύρεσης του παλμού λειτουργίας duty cycle (κύκλος)

Έχουμε :

**t=ton+toff** (χρονος της pwm)

**ton** : είναι ο παλμός στα +v

**toff**: είναι ο παλμός στα 0 v

**τύπος : [ton /(ton+toff)] \* 100**

π.χ. v=5v

**duty cycle =75%** τότε

**vout = (75% / 5)\* 100 = 3,75 v** η έξοδος από το Arduino

**περίοδος t=1/f =ton+toff**

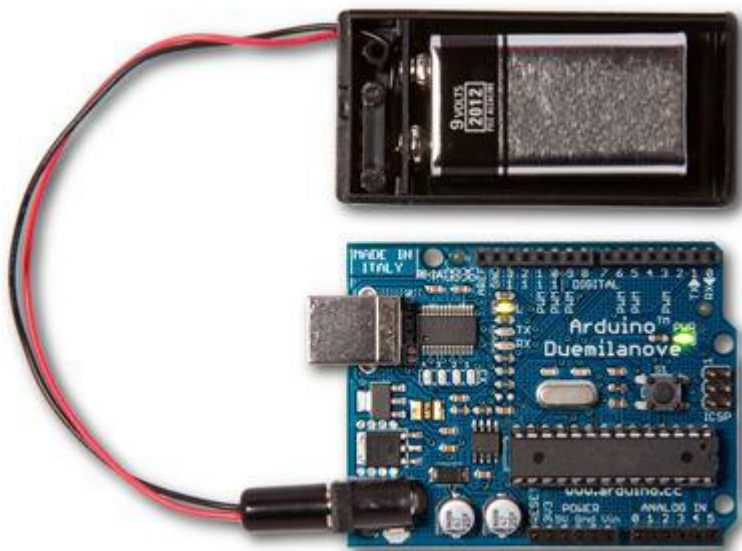
**duty cycle [ton /(ton+toff)] \* 100 =ton / t**

**όταν θετικό παλμός ton = duty cycle \* t**

**όταν 0 παλμός toff =t-ton**

## 2.4.2 Τροφοδοσία

Το Arduino μπορεί να τροφοδοτηθεί με ρεύμα είτε από τον υπολογιστή μέσω της σύνδεσης USB, είτε από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής φινις των 2.1mm (θετικός πόλος στο κέντρο) και βρίσκεται στην κάτω-αριστερή γωνία του Arduino.



εικ.2.6 Τροφοδοσία του Arduino

Για να μην υπάρχουν προβλήματα, η εξωτερική τροφοδοσία πρέπει να είναι από 7 ως 12V και μπορεί να προέρχεται από ένα κοινό μετασχηματιστή του εμπορίου, από μπαταρίες ή οποιαδήποτε άλλη πηγή DC.

### 2.4.3 Ενσωματωμένα κουμπιά και LED του Arduino UNO.

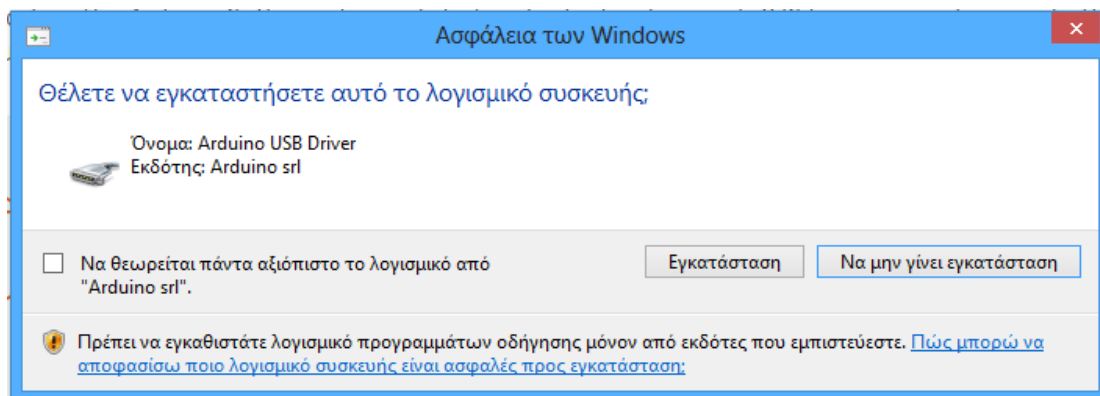
Πάνω στην πλακέτα του Arduino υπάρχει ένας διακόπτης micro-switch και τέσσερα μικροσκοπικά LED επιφανειακής στήριξης. Η λειτουργία του διακόπτη (που έχει τη σήμανση reset) και του ενός LED με τη σήμανση power είναι μάλλον προφανής. Τα δύο LED με τις σημάσεις TX και RX, χρησιμοποιούνται ως ένδειξη λειτουργίας του σειριακού interface καθώς ανάβουν όταν ο Arduino στέλνει ή λαμβάνει αντίστοιχα δεδομένα μέσω USB. Τέλος, υπάρχει το LED με τη σήμανση L. Η βασική δοκιμή λειτουργίας του Arduino είναι να του αναθέσετε να αναβοσβήνει ένα LED.

## 2.5 Εγκατάσταση περιβάλλοντος Arduino IDE

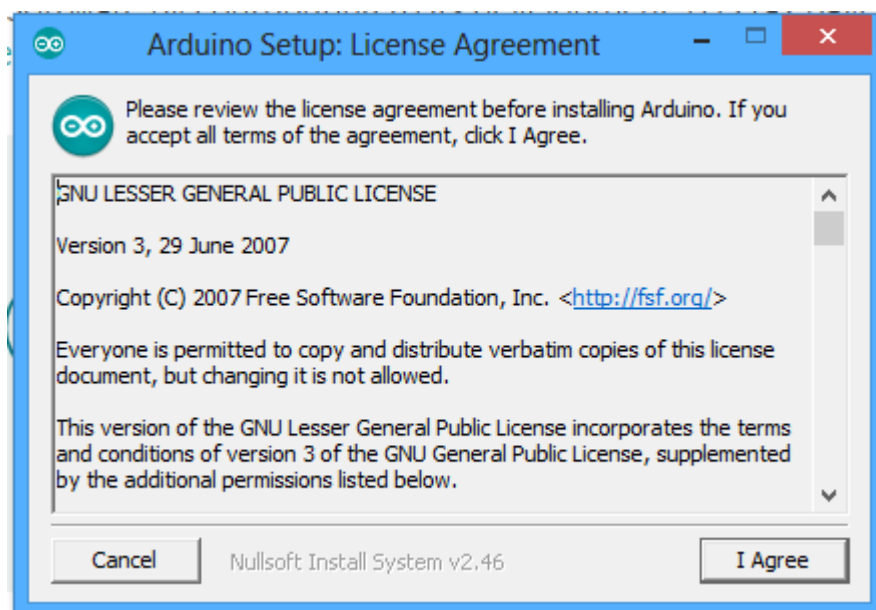
Για να γίνει σωστή εγκατάσταση του προγράμματος, πρέπει να ακολουθηθεί μια σειρά από βήματα, ανάλογα με το λειτουργικό σύστημα που διαθέτει. Στην περίπτωση μας θα εγκατασταθεί σε λειτουργικό σύστημα Windows.

Στην παρούσα πτυχιακή εργασία επιλέχθηκε να χρησιμοποιηθεί η πλακέτα Arduino UNO. Θα χρησιμοποιήσουμε ένα καλώδιο USB για να συνδεθούν πλακέτα και υπολογιστής. Μεταφορτώνουμε δωρεάν την τελευταία έκδοση Arduino-1.6. Από την

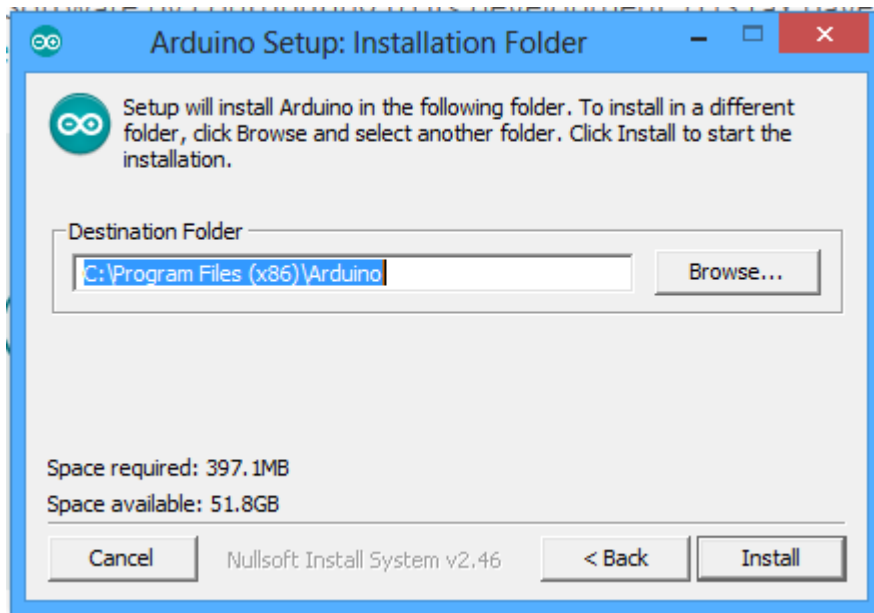
ιστοσελίδα <http://arduino.cc/en/main/software>. Συνδέουμε την πλακέτα Arduino UNO στον υπολογιστή χρησιμοποιώντας το καλώδιο USB. Παρατηρούμε ότι το LED της πλακέτας ανάβει. Αρχίζουμε το κατέβασμα από την σελίδα που προαναφέρθηκε και τα βήματα που θα ακολουθηθούν παρατίθενται στις εικόνες που ακολουθούν.



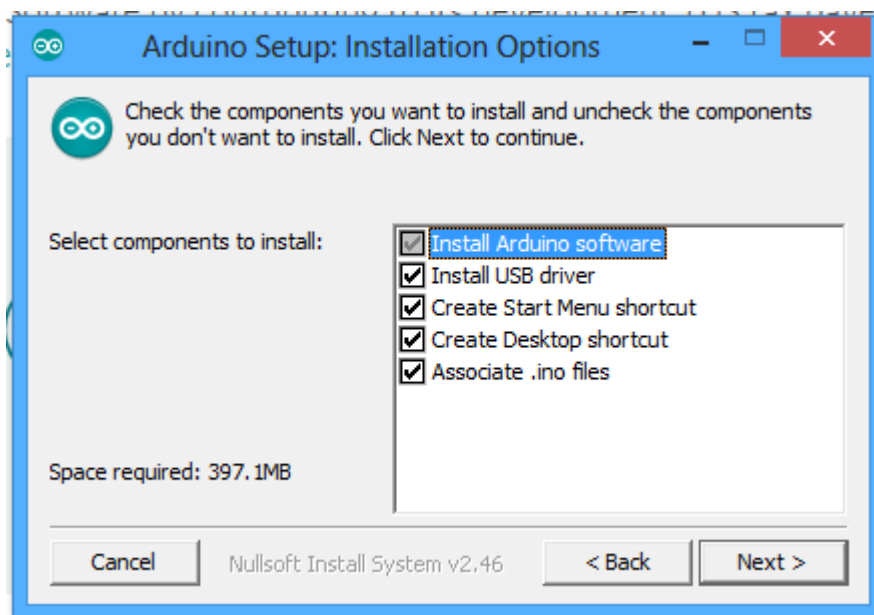
εικ. 2.7 Κάνουμε πρώτα την εγκατάσταση του λογισμικού στον υπολογιστή μας.



εικ 2.8 Αποδεχόμαστε τους όρους.

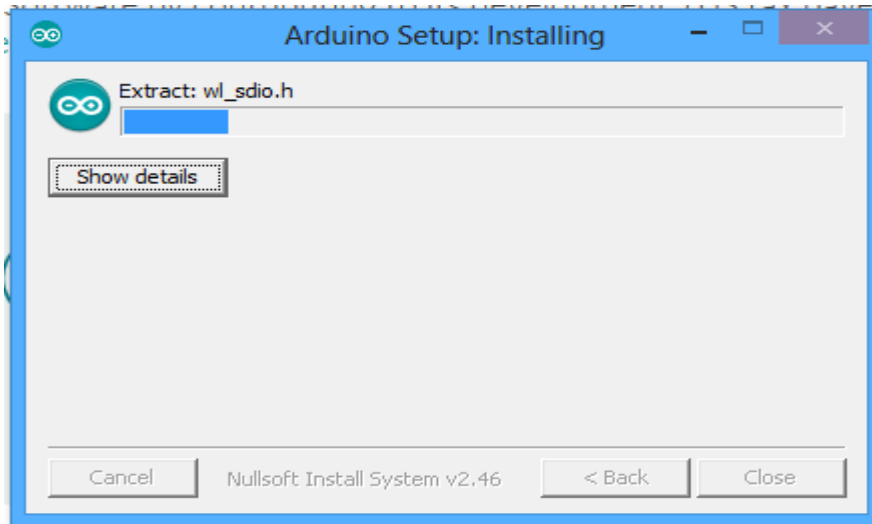


εικ. 2.9 Επιλέγουμε πού θέλουμε να αποθηκευτεί το πρόγραμμα.



εικ.2.10 Επιλέγουμε τα περιεχόμενα που θέλουμε να αποθηκεύσουμε σύμφωνα με αυτά που χρειάζεται ο καθένας.

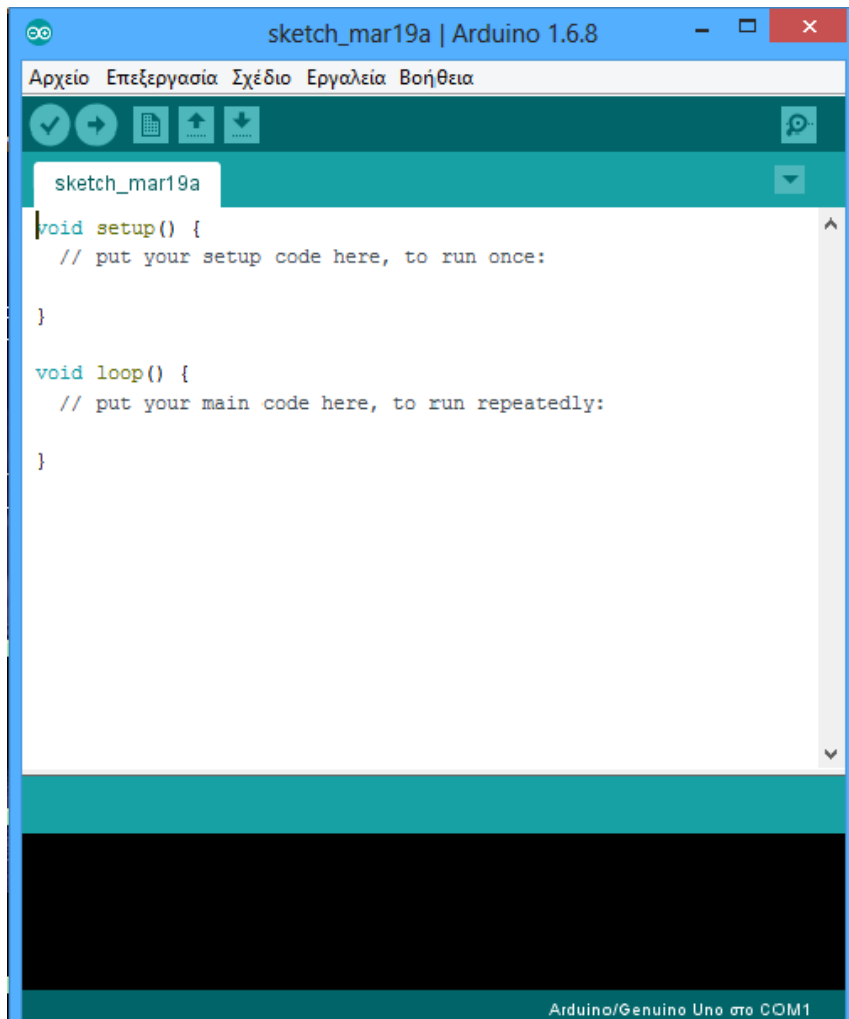




εικ. 2.11 Περιμένουμε να γίνει η εγκατάσταση.



εικ. 2.12 Μόλις γίνει η εγκατάσταση, εμφανίζεται στην επιφάνεια εργασίας, όπου εμείς επιλέξαμε να εμφανιστεί εκεί το εικονίδιο της συντόμευσης. Επάνω σε αυτό πατάμε διπλό κλικ ώστε να ανοίξει το πρόγραμμά μας.



εικ. 2.13 Αυτός είναι ο εργασιακός μας χώρος όπου θα ασχοληθούμε με την συγγραφή του προγραμματισμού του ρομποτικού μας οχήματος.

Το περιβάλλον ανάπτυξης του Arduino περιέχει μια περιοχή επεξεργασίας κειμένου για την συγγραφή κώδικα, μια περιοχή μηνυμάτων, ένα μενού, μια γραμμή εργαλείων και τη βοήθεια. Συνδέεται με το υλικό Arduino UNO για την εργασία μας για τη φόρτωση προγραμμάτων και για να επικοινωνούν μεταξύ τους. Ένα ολοκληρωμένο πρόγραμμα ονομάζεται sketch (όπως αναγράφεται στην εικ.1.13). Έχει δυνατότητες για την αντιγραφή/επικόλληση, αναζήτηση και αντικατάσταση κειμένου. Στην κονσόλα της εικόνας επίσης απεικονίζεται η έξοδος του κειμένου για το περιβάλλον Arduino συμπεριλαμβανομένου τον έλεγχο για τυχόν λάθη που μπορεί να κάνουμε στον κώδικα καθώς τον συγγράφουμε. Τα κουμπιά της γραμμής εργαλείων επιτρέπουν τον έλεγχο και το ανέβασμα των προγραμμάτων, τη δημιουργία νέου sketch και το άνοιγμα ήδη υπάρχοντων sketch. Τελειώνοντας, αποθηκεύουμε τον κώδικά μας.

## **ΚΕΦΑΛΑΙΟ 3**

### **3.1. Το ολοκληρωμένο κύκλωμα οδήγησης κινητήρων**

Τα ψηφιακά κυκλώματα κατασκευάζονται κυρίως με χρήση ολοκληρωμένων κυκλωμάτων (που λέγονται για συντομία ICs- INTEGRATED CIRCUITS). Κάθε IC είναι ένας μικρός κρύσταλλος ημιαγωγού πυριτίου (Si), καλούμενο CHIP. Το CHIP περιλαμβάνει ηλεκτρικά στοιχεία όπως τρανζίστορες, διόδους, αντιστάσεις και πυκνωτές. Τα στοιχεία αυτά είναι συνδεδεμένα μέσα στο CHIP, ώστε να σχηματίζουν ένα ηλεκτρονικό κύκλωμα. Το CHIP τοποθετείται πάνω σε μεταλλικό ή πλαστικό στέλεχος και οι συνδέσεις συγκολλούνται σε εξωτερικά "ποδαράκια" ,έτσι σχηματίζεται το IC. Τα ολοκληρωμένα κυκλώματα διαφέρουν από τα συμβατικά κυκλώματα διακριτών στοιχείων στο ότι τα στοιχεία τους δεν μπορούν να διαχωριστούν ή να αποσυνδεθούν από το κύκλωμα του εσωτερικού του ολοκληρωμένου πακέτου. Η σύνδεση του ολοκληρωμένου με το υπόλοιπο εξωτερικό κύκλωμα γίνεται μόνο με τους εξωτερικούς του ακροδέκτες (ποδαράκια).

Στη φάση κατασκευής τους τα ολοκληρωμένα κυκλώματα (τα οποία ακόμη δεν έχουν ολοκληρωθεί ώστε να λειτουργήσουν) ονομάζονται κύβοι και φτιάχνονται κατά εκατοντάδες πάνω σε πλακίδια. Οι λογικές πύλες με την παρούσα τεχνολογία υλοποιούνται με παθητικά στοιχεία, οπότε τα ολοκληρωμένα κυκλώματα είναι παθητικά. Ολοκληρωμένα κυκλώματα χρησιμοποιούνται σχεδόν σε κάθε στοιχείο ηλεκτρονικού εξοπλισμού που χρησιμοποιείται σήμερα και θεωρούνται επανάσταση στον τομέα της ηλεκτρονικής.

Πλεονεκτήματα των IC:

- 1) Πολύ μικρό μέγεθος
- 2) Χαμηλή τιμή κόστους μαζικής παραγωγής
- 3) Μικρή κατανάλωση ισχύος
- 4) Υψηλή αξιοπιστία λειτουργίας
- 5) Υψηλή ταχύτητα λειτουργίας
- 6) Μείωση εξωτερικών καλωδιακών συνδέσεων

#### **3.1.1 Ιστορικά στοιχεία των ολοκληρωμένων κυκλωμάτων**

Το 1950 είχε εμφανιστεί η κρυσταλλοτρίδος ενώ είχαν κατασκευαστεί οι ηλεκτρονικοί υπολογιστές με λυχνίες κενού και κρυσταλλοτρίδους. Το πρόβλημα των μηχανικών εκείνης της εποχής ήταν ότι δεν υπήρχε αξιόπιστος τρόπος να κατασκευάζονται περίπλοκα και μεγάλα λογικά κυκλώματα, γνωστό και ως "τυραννία των αριθμών". Το 1958 ο Τζακ Κίλμπυ, ερευνητής της εταιρίας Texas

Instrument, ερευνώντας πυρετωδώς τρόπους για τη σμίκρυνση των ηλεκτρικών κυκλωμάτων εφήυρε το ολοκληρωμένο κύκλωμα, εφεύρεση για την οποία βραβεύτηκε με βραβείο Νόμπελ Φυσικής το 2000.

Η τεχνολογία για τη μαζική παραγωγή ολοκληρωμένων κυκλωμάτων ήταν ακόμη ελλιπής, υπήρχαν διάφορα πρακτικά προβλήματα, όπως η σύνδεση των στοιχείων του ολοκληρωμένου. Ο Ρόμπερτ Νόις έλυσε αυτά τα προβλήματα, έτσι ξεκίνησε η μαζική παραγωγή των ολοκληρωμένων κυκλωμάτων. Ο Νόις είναι ένας από τους συνιδρυτές της Intel, μια από τις πιο σημαντικές κατασκευάστριες εταιρείες ολοκληρωμένων κυκλωμάτων στον κόσμο. Το ολοκληρωμένο κύκλωμα αποτέλεσε τομή στην ιστορία των ηλεκτρονικών υπολογιστών.

### 3.1.2 Διάκριση ολοκληρωμένων κυκλωμάτων

- Τα ολοκληρωμένα κυκλώματα διακρίνονται σε τέσσερις (4) κατηγορίες:
- SSI (μικρής κλίμακας ολοκλήρωσης)
  - MSI (μεσαίας κλίμακας ολοκλήρωσης)
  - LSI (μεγάλης κλίμακας ολοκλήρωσης)
  - VLSI (πολύ μεγάλης κλίμακας ολοκλήρωσης)

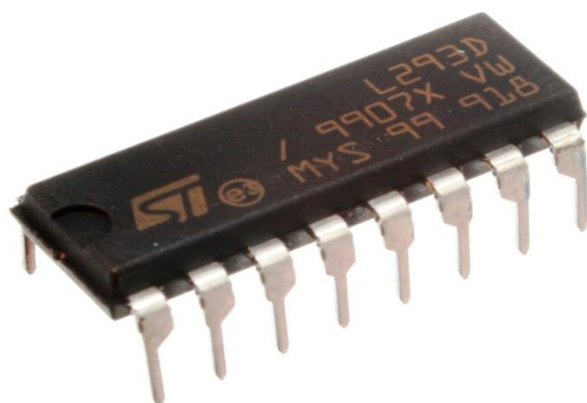
Η διάκρισή τους γίνεται σύμφωνα με τον αριθμό των ολοκληρωμένων κυκλωμάτων από τα οποία αποτελούνται.



εικ. 3.1 Ολοκληρωμένα κυκλώματα

## 3.2 Το ολοκληρωμένο κύκλωμα L293D

Ένα από τα πολλά ολοκληρωμένα κυκλώματα που υλοποιούν H- Bridge είναι και το L293D το οποίο έχει δύο γέφυρες και επομένως μπορεί να ελέγξει μέχρι και δύο dc κινητήρες που θα αναφερθούμε παρακάτω στο κεφάλαιο(3.2.2). Το συγκεκριμένο ολοκληρωμένο έχει ενσωματωμένες διόδους που προστατεύουν τα εσωτερικά του κυκλώματα από εγκλωβισμένα ρεύματα του πηνίου του κινητήρα. Επίσης, το μέγιστο ρεύμα που μπορεί να δώσει σε κάθε κινητήρα είναι 600Ma και μέγιστη συχνότητα PWM σήματος είναι 5kHz. Το ολοκληρωμένο προσφέρει δύο κανάλια (δύο ξεχωριστές H-Bridge) για κάθε κινητήρα αλλά θα μπορούσε να λειτουργήσει και με μόνο ένα κανάλι ενεργό. Επίσης, το L293D είναι λειτουργίας TTL και το CHIP είναι της εταιρείας Texas Instrument.



εικ. 3.2

### 3.2.1 Κινητήρες- εισαγωγή

Οι πιο απλοί και οι πιο συνηθισμένοι μηχανισμοί κίνησης στα ρομπότ είναι οι ηλεκτρονικοί κινητήρες. Ο ηλεκτρονικός κινητήρας είναι μια ηλεκτρομηχανική συσκευή η οποία χρησιμοποιείται για τη μετατροπή της ηλεκτρικής ενέργειας σε κινητική. Η αρχή λειτουργίας του ηλεκτρικού κινητήρα είναι η δύναμη Lorenz η οποία λέει ότι όταν ένας αγωγός διαρρέεται από ηλεκτρικό ρεύμα και βρίσκεται μέσα σε ένα μαγνητικό πεδίο τότε ασκείται πάνω στον αγωγό μια δύναμη (F) η οποία είναι ίση με

$$F= I*\lambda*B$$

όπου I ισούται με την ένταση του ηλεκτρικού ρεύματος, το λ ισούται με το μήκος του αγωγού και το B ισούται με την ένταση του μαγνητικού πεδίου.

Η δύναμη αυτή στέλνει την κατάλληλη συνεχή τάση στον αγωγό του ηλεκτροκινητήρα και αυτός αρχίζει να περιστρέφεται. Ανάλογα με τη διαφορά

δυναμικού που θα παρουσιάζεται στους ακροδέκτες των κινητήρων, αυτοί θα έχουν τη δυνατότητα να περιστρέφονται δεξιόστροφα ή αριστερόστροφα, έτσι το όχημα έχει τη δυνατότητα να κινηθεί ως εξής:

-Επιτόπου στροφή δεξιά, όταν ο δεξιός ηλεκτροκινητήρας περιστρέφεται δεξιόστροφα ενώ ο αριστερός περιστρέφεται αριστερόστροφα.

-Επιτόπου στροφή αριστερά, όταν ο δεξιός ηλεκτροκινητήρας περιστρέφεται αριστερόστροφα ενώ ο αριστερός περιστρέφεται δεξιόστροφα.

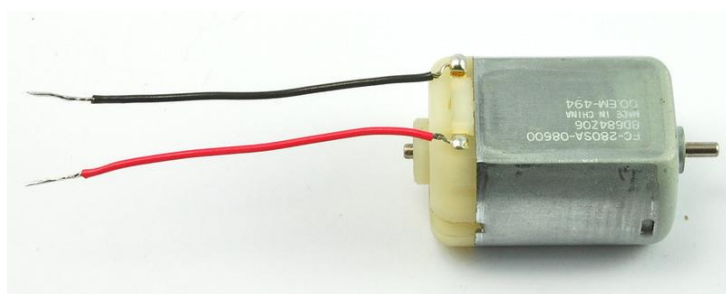
-Κίνηση προς τα μπρός, όταν και οι δύο κινητήρες περιστρέφονται αριστερόστροφα.

- Κίνηση προς τα πίσω, όταν και οι δύο κινητήρες περιστρέφονται δεξιόστροφα.

### 3.2.2 Κινητήρας DC

Οι ηλεκτροκινητήρες προορίζονται για τη χρήση στα 6V. Σε γενικές γραμμές, αυτά τα είδη των κινητήρων μπορούν να λειτουργήσουν σε τάσεις πάνω και κάτω από αυτή την ονομαστική τάση. Έτσι λειτουργούν άνετα στην περιοχή τάσης 3-9V. Οι χαμηλότερες τάσεις ενδέχεται να μην είναι πρακτικές, όπως και οι υψηλότερες τάσεις θα μπορούσαν να επηρεάσουν αρνητικά τη διάρκεια ζωής του κινητήρα.

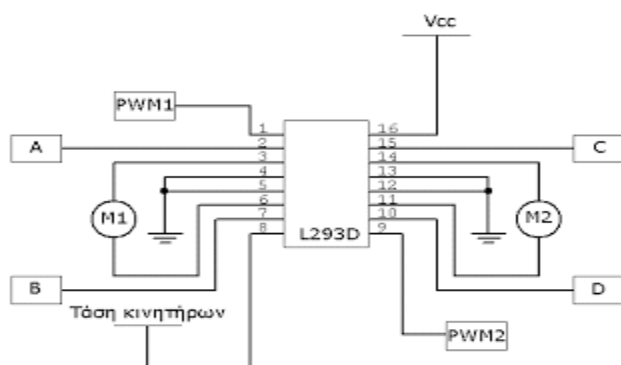
Όσο αφορά λοιπόν την παρούσα εφαρμογή, οι κινητήρες που είναι υπεύθυνοι για την κίνηση των τροχών πρέπει να πληρούν, για τις ανάγκες της εργασίας, τις βασικές προϋποθέσεις. Χρειαζόμαστε συνεχούς περιστροφής κινητήρες οι οποίοι πρέπει να έχουν τάση λειτουργίας της τάξης των 6V και ροπή ικανή να μεταφέρει το βάρος της κατασκευής χωρίς να υπάρχει μεγάλη επιβάρυνση της μπαταρίας.



εικ. 3.3 Κινητήρας DC

Οι κινητήρες στη δική μας κατασκευή υπήρχαν ήδη προσαρτημένοι επάνω στο αυτοκίνητό μας που θα αναφερθούν αναλυτικότερα παρακάτω.

### 3.2.3 Συνδεσμολογία για τον έλεγχο δύο κινητήρων με το ολοκληρωμένο κύκλωμα.



εικ 3.4 Σχηματική αναπαράσταση ολοκληρωμένου κυκλώματος

Στην παραπάνω εικόνα βλέπουμε τη συνδεσμολογία μεταξύ του L293D των κινητήρων (M1 και M2) καθώς και των ακροδεκτών (A, B, C, D, PWM1 και PWM2). Οι ακροδέκτες αυτοί μεταφέρουν σήματα από το μικροελεγκτή προς το ολοκληρωμένο κύκλωμα. Το Vcc είναι η τροφοδοσία που χρειάζεται το ολοκληρωμένο και θα πρέπει να είναι 5V. Η τάση των κινητήρων είναι ανάλογη με το είδος των κινητήρων που θέλουμε να ελέγξουμε, θα πρέπει να προσέξουμε παρόλα αυτά το μέγιστο ρεύμα που μπορεί να δώσει το κάθε κανάλι. Τα σήματα A, B, C, και D καθορίζουν τη λειτουργία των κινητήρων (φρενάρισμα, κίνηση αρνητικής κατεύθυνσης ή θετικής) ενώ τα σήματα PWM1 και PWM2 ρυθμίζουν την ταχύτητα της κίνησης των κινητήρων M1 και M2 αντίστοιχα με την τεχνική διαμόρφωσης εύρους παλμού.

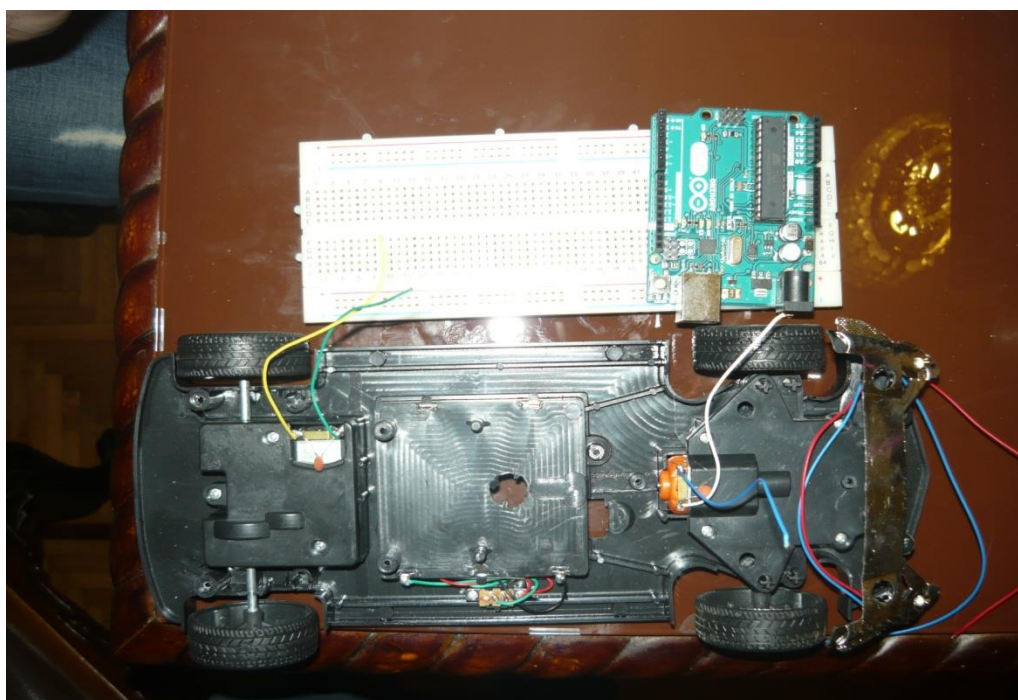
Μέχρι και σε αυτό το σημείο έχουμε αναφερθεί σε δύο πρακτικές. Μία για τον υπολογισμό της ταχύτητας (διαμόρφωση εύρους παλμού) και μία για τον υπολογισμό της φοράς της κίνησης του dc κινητήρα (H-Bridge στο οποίο θα αναφερθούμε στην επόμενη ενότητα). Μπορούμε να χρησιμοποιήσουμε PWM σήματα στις εισόδους έλεγχου του κυκλώματος H-Bridge και να ελέγξουμε πλήρως κατεύθυνση και ταχύτητα. Τα σήματα PWM για τον έλεγχο προέρχονται από τον μικροελεγκτή και έχουν μέγιστη συχνότητα 5kHz λόγω περιορισμού από τον κατασκευαστή του L293D.

#### Έλεγχος θέσης

Για να ελέγξουμε την ακριβή θέση ενός dc κινητήρα χρειαζόμαστε κάποιο αισθητήρα γιατί ο dc κινητήρας είναι μη ελέγξιμος ως προς την θέση. Για να δώσουμε κάποια επιθυμητή θέση μέσω κάποιου σήματος έλεγχου πρέπει να υπολογίσουμε το κλειστό σύστημα κινητήρας - αισθητήρας και με όλες αυτές τις τεχνικές που αναλύσαμε να εφαρμόσουμε τα θεωρήματα της θεωρίας ελέγχου.

### 3.3 Κίνηση dc μοτέρ με τη χρήση ολοκληρωμένου κυκλώματος.

Το τηλεκατευθυνόμενο που θα φτιάξουμε έχει δύο dc μοτέρ, ένα για την κίνηση πίσω κι ένα για την αλλαγή κατεύθυνσης μπροστά. Πριν συνεχίσουμε θα πρέπει να αποσυνδέσουμε το υπάρχον κύκλωμα του τηλεκατευθυνόμενου αφήνοντας αρκετό καλώδιο πίσω για να επιτύχουμε κολλήσεις οι οποίες θα γίνουν σε επόμενη φάση. Δείτε την εικόνα παρακάτω.

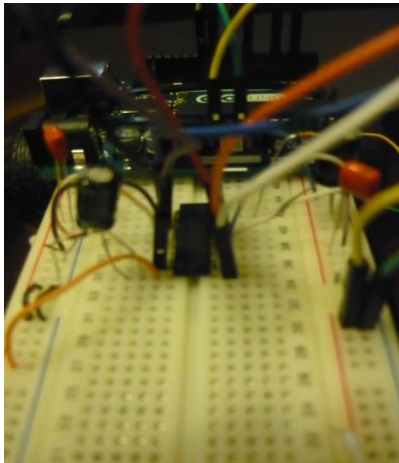


εικ. 3.5 Αποσύνδεση συστήματος τηλεχειρισμού και πλαστικών τμημάτων του αυτοκινήτου

Ενώνουμε τους πόλους μιας μπαταρίας 1.5V AA με τα καλώδια του πίσω μοτέρ (κίνησης) για να βρούμε με ποια πολικότητα το όχημά μας κινείται μπροστά. Βάζουμε ένα αυτοκόλλητο σε αυτό το καλώδιο (που ενώνεται με το + της μπαταρίας) για να το ξεχωρίσουμε. Κάνουμε το ίδιο με το μοτέρ κατεύθυνσης για να δούμε με ποια πολικότητα οι ρόδες στρίβουν αριστερά και δεξιά. Βάζουμε αυτοκόλλητο στο καλώδιο (που ενώνεται με το + της μπαταρίας) που έκανε τις ρόδες να στρίψουν αριστερά.

Το ολοκληρωμένο κύκλωμα L293D το χρησιμοποιούμε για να “οδηγήσουμε” σωστά το μοτέρ μας. Αυτό σημαίνει ότι τα pin του Arduino UNO δεν μπορούν να δώσουν το ρεύμα που χρειάζεται για να κινηθούν τα μοτέρ γι’ αυτό το λόγο πρέπει να χρησιμοποιήσουμε transistor. Το L293D κάνει ακριβώς αυτή τη δουλειά κρύβοντας στο εσωτερικό του τέσσερα transistor.

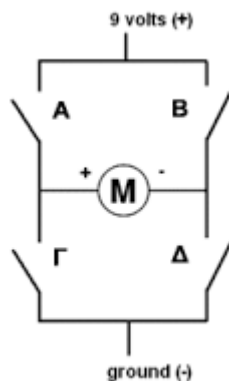




εικ.3.6 L293

Το L293D εκτός του ότι μπορεί να παρέχει το απαιτούμενο ρεύμα στα μοτέρ για να οδηγηθούν σωστά μπορεί να δώσει και τάση μέχρι 35V, αυξάνοντας έτσι την ισχύ  $P=V \cdot I$ . Όμως, αυτό δε σημαίνει ότι τα μοτέρ θα μπορέσουν να αποδώσουν σε τέτοιες τιμές. Πιθανόν να καούν ή να καεί και το ίδιο το ολοκληρωμένο. Γι' αυτό στο pin8 του L293 Motor Power μπορούμε να δώσουμε από 5 μέχρι το πολύ 12V. Το L293D δεν είναι και ό,τι καλύτερο που υπάρχει για την οδήγηση κινητήρων dc καθώς θερμαίνεται όσο ανεβαίνει η ισχύς και μειώνεται η απόδοσή του και δεν μπορεί να δώσει ρεύμα πάνω από 0,5A σε κάθε κανάλι. Ωστόσο είναι φθηνό και έχει εύκολη συνδεσμολογία και το ρεύμα του επαρκεί για την κατασκευή μας.

### 3.3.1 H-Bridge



εικ. 3.7 H-Bridge

Το H-Bridge ονομάζεται και Full-Bridge. Το H-Bridge ονομάζεται έτσι επειδή έχει τέσσερις διακόπτες σε κάθε άκρη του γράμματος H και τον κινητήρα που ελέγχεται

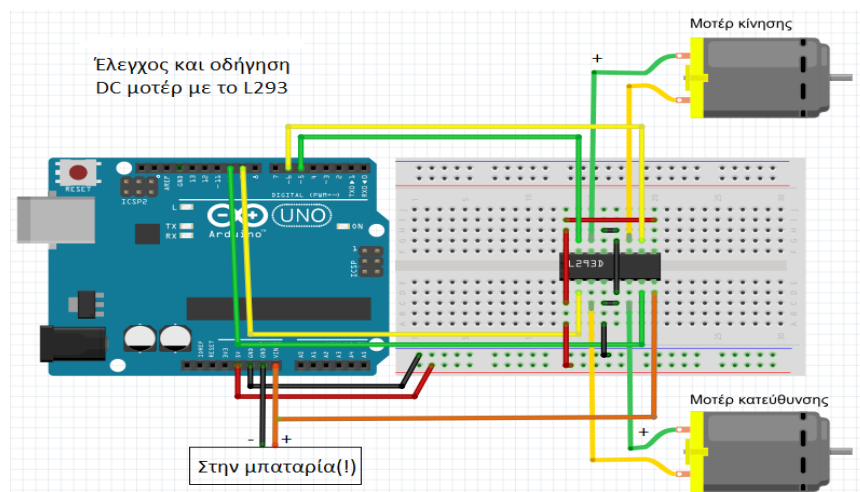
στην οριζόντια γραμμή του όπως στο σχήμα. Βέβαια δεν είναι ακριβώς το γράμμα Η αλλά η ομοιότητα είναι εμφανής. Τα σήματα ελέγχου Α, Β, Γ και Δ στο συγκεκριμένο σχήμα δίνονται από χειροκίνητους διακόπτες. Το κύκλωμα αυτό χρησιμοποιείται για να ελέγχουμε την πορεία του ρεύματος μέσα στον κινητήρα.

- Όταν κλείσουμε τους διακόπτες Α και Δ τότε ρεύμα περνάει από το θετικό στο αρνητικό ακροδέκτη του κινητήρα κάνοντας τον να κινείται προς μια κατεύθυνση.
- Όταν κλείσουμε τους διακόπτες Β και Γ τότε ρεύμα περνάει από τον αρνητικό στο θετικό ακροδέκτη κάνοντας τον να κινείται προς την αντίθετη κατεύθυνση.
- Στις περιπτώσεις όπου κλείσουμε τους διακόπτες Α και Β ή Γ και Δ προκαλούμε φρενάρισμα του κινητήρα.
- Στις περιπτώσεις όπου κλείσουμε τους διακόπτες Α και Γ ή Β και Δ προκαλούμε βραχυκύκλωμα της πηγής τάσεως και είναι ζημιογόνο.

### 3.3.2 Συνδεσμολογία Arduino με το ολοκληρωμένο κύκλωμα L293D

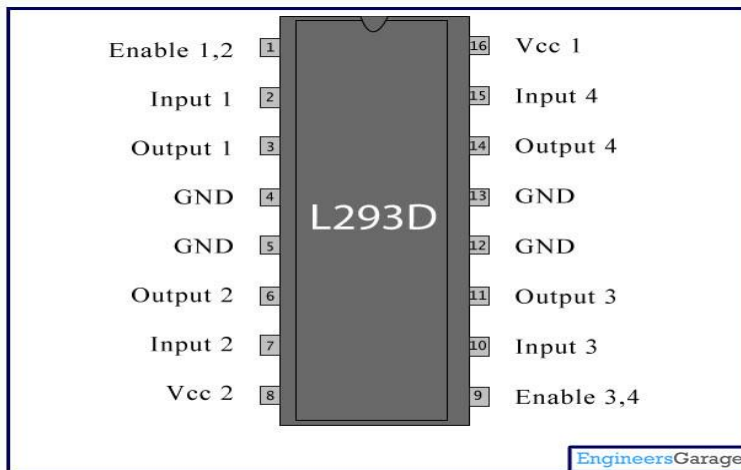
Τα υλικά που θα χρειαστούμε:

1. Arduino UNO
2. L293D
3. Καλώδια (για την ένωση)



εικ. 3.8 Προσομοίωση της σύνδεσης Arduino με το ολοκληρωμένο κύκλωμα

Σταθεροποιούμε αρχικά την πλακέτα Arduino στο Raster( Breadboard) έτσι ώστε να γίνει εύχρηστη και να μη φεύγει από τη θέση της. Στη συνέχεια σταθεροποιούμε το ολοκληρωμένο κύκλωμα L293D έτσι ώστε να συνδέσουμε το Arduino με το ολοκληρωμένο κύκλωμα.

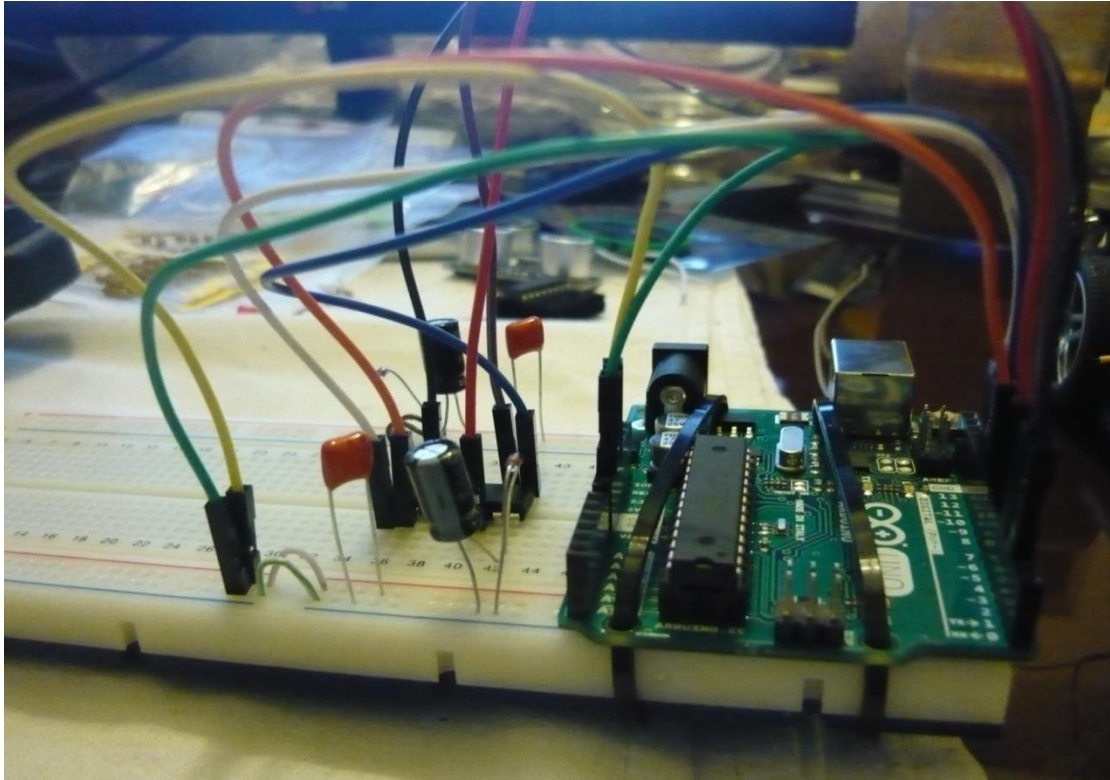


εικ. 3.9 L293D

Όπως βλέπουμε στην εικόνα 3.7, συνδέουμε με καλώδιο τον Arduino με το ολοκληρωμένο κύκλωμα

- το Enable 1,2 με τον υποδοχέα του Arduino ~11 Digital
- το Input 1 με τον υποδοχέα 8
- το Output 1 το αφήνουμε κενό
- το GND 4 στη γείωση του Raster
- το GND 5 επίσης στη γείωση Raster
- το Output 2 το αφήνουμε κενό
- το Input 2 με τον υποδοχέα 7, ο οποίος είναι Digital
- το Vcc 2 στη γείωση του Raster
- το Vcc 1 στη γείωση του Raster
- το Input 4 με τον υποδοχέα 2, ο οποίος είναι Digital
- το Output 4 το αφήνουμε κενό
- το GND 13 στη γείωση του Raster
- το GND 12 στη γείωση του Raster
- το Output 3 το αφήνουμε κενό
- το Input 3 με τον υποδοχέα 4, ο οποίος είναι Digital PWM
- το Enable 3,4 με τον υποδοχέα του Arduino ~10 Digital

Συνδέουμε τα GND του Arduino με τα GND του Raster (γείώσεις). Βάζουμε πυκνωτές και διόδους στα Enable 1,2 και Enable 3,4. Το ένα σκέλος (ποδαράκια) του πυκνωτή μπαίνει στο Enable 1,2 και το άλλο σκέλος στη γείωση. Το ίδιο αντιστοίχως γίνεται και με τη διάοδο, όπου το ένα σκέλος μπαίνει στο Enable 3,4 και το άλλο στη γείωση. Στην περίπτωση του πυκνωτή το σκέλος που μπαίνει στη γείωση, μπαίνει στον αρνητικό ημισφαίριο και στην περίπτωση της διάοδου το σκέλος που μπαίνει στη γείωση, στο θετικό ημισφαίριο της γείωσης του Raster.

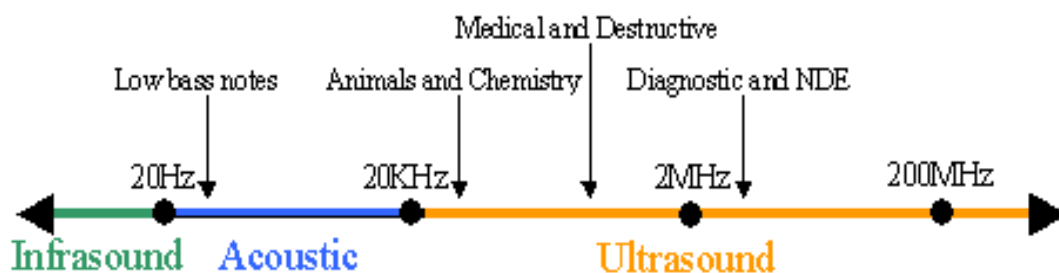


εικ. 3.10 Η σύνδεση στο δικό μας ρομποτικό αυτοκίνητο

## ΚΕΦΑΛΑΙΟ 4

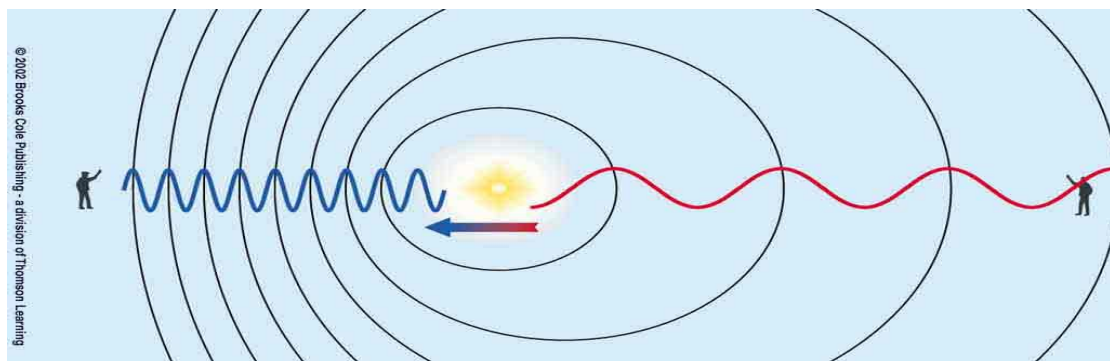
### 4.1 Εισαγωγή

Υπέρηχος ονομάζεται το μηχανικό κύμα με συχνότητα μεγαλύτερη από αυτήν που μπορεί να ακούσει ο άνθρωπος (περίπου 20.000 Hz). Αυτό συμβαίνει καθώς το ανθρώπινο αυτί έχει κάποια όρια και δεν μπορεί να ακούσει πολύ υψηλούς ή χαμηλούς ήχους. Η φυσική του υπέρηχου είναι ίδια με του ήχου (δηλαδή η ταχύτητά του στον αέρα είναι 331,5 m/sec).



εικ. 4.1 Υπέρηχος

Παρόλο πάντως που οι άνθρωποι δεν τους ακούνε, κάποια ζώα μπορούν να τους ακούσουν αλλά και τους χρησιμοποιούν. Χαρακτηριστικά παραδείγματα που μας το δείχνουν αυτό είναι η κίνηση των νυχτερίδων και η σφυρίχτρα που χρησιμοποιείται για τους σκύλους. Οι νυχτερίδες μάλιστα μπορούν να αντιληφθούν το στόχο που κυνηγούν χάρη στο φαινόμενο Doppler.

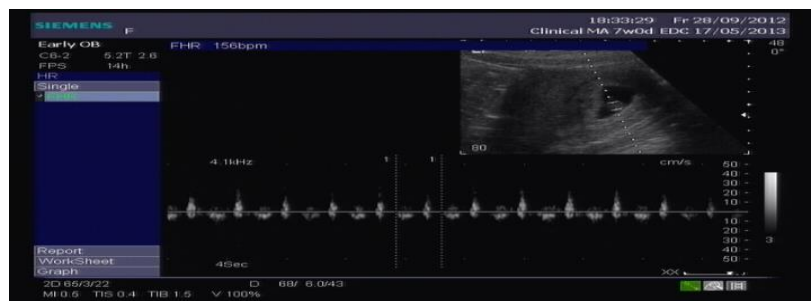


εικ. 4.2 Το φαινόμενο Doppler

Το φαινόμενο Doppler είναι η παρατηρούμενη αλλαγή στη συχνότητα και το μήκος ενός κύματος από παρατηρητή που βρίσκεται σε σχετική κίνηση με την πηγή των κυμάτων. Για κύματα όπως τα ηχητικά, όπως διαδίδονται μέσα σε κάποιο υλικό μέσο, η ταχύτητα τόσο του παρατηρητή όσο και της πηγής, πρέπει να προσδιορίζεται σε σχέση με το μέσο διάδοσης. Το τελικό φαινόμενο Doppler μπορεί επομένως να προκύψει είτε από την κίνηση του παρατηρητή είτε από την κίνηση της πηγής, είτε και των δύο, ως προς το μέσο διάδοσης. Κάθε μια από αυτές τις δύο επιδράσεις

αναλύεται ξεχωριστά.

Τέλος ο υπερήχος χρησιμοποιείται σε συσκευές απεικόνισης του εσωτερικού ενός ανθρώπου, του γήινου φλοιού και άλλων αντικειμένων. Ακόμη σε συσκευές που χρησιμοποιούνται για την ανίχνευση αόρατων ραγισμάτων και αποστάσεων. Μια σημαντική εφαρμογή είναι η απεικόνιση του εμβρύου στις έγκυες γυναίκες.



εικ. 4.3 Απεικόνιση εμβρύου μέσω υπερήχου

## 4.2 Ο αισθητήρας υπερήχων HC-SR04

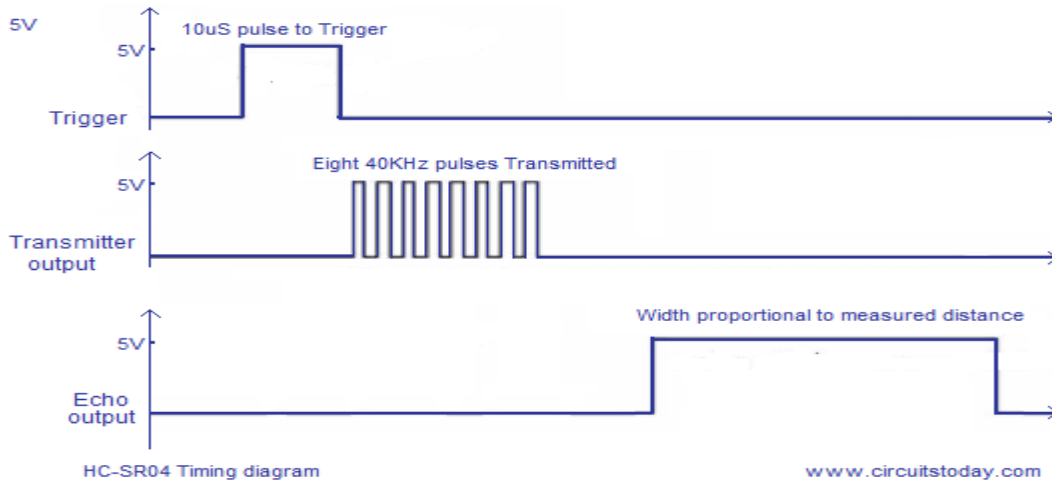
Για τη διαδικασία εντοπισμού και αποφυγής των εμποδίων χρησιμοποιήθηκαν αισθητήρες υπερήχων. Οι αισθητήρες υπερήχων είναι μια φθηνή και αξιόπιστη λύση χωρίς να απαιτούν πολύπλοκα κυκλώματα υποστήριξης. Είναι αρκετά δημοφιλή στην κατηγορία των ρομποτικών οχημάτων.

Οι αισθητήρες υπερήχων λειτουργούν με την ίδια αρχή που λειτουργούν τα ραντάρ και τα σόναρ. Απαρτίζονται από έναν πομπό, ο οποίος δημιουργεί το σήμα και από ένα δέκτη ο οποίος το λαμβάνει. Εκτιμούν την απόσταση ενός στόχου ή ακόμα και την ταχύτητα του στόχου, λαμβάνοντας υπ'όψιν τους την αντανάκλαση ενός ραδιοκύματος ή ενός ηχητικού σήματος επάνω στο στόχο, υπολογίζοντας το χρόνο που έκανε το σήμα για να καλύψει την απόσταση από τον αισθητήρα έως το αντικείμενο και πίσω. Έτσι ο αισθητήρας στέλνει ριπές παλμών υψηλής συχνότητας (40 kHz περίπου), ενώ η κάθε ριπή ταξιδεύει στον αέρα με ταχύτητα 344 m/sec. Τα ηλεκτρονικά του αισθητήρα μετρούν το χρόνο λήψης του σήματος και το μετατρέπουν σε μονάδα μήκους, δεδομένου ότι η ταχύτητα του ήχου επηρεάζεται από τη θερμοκρασία αέρα, οι αισθητήρες υπερήχων περιλαμβάνουν ένα ενσωματωμένο αισθητήρα θερμοκρασίας.



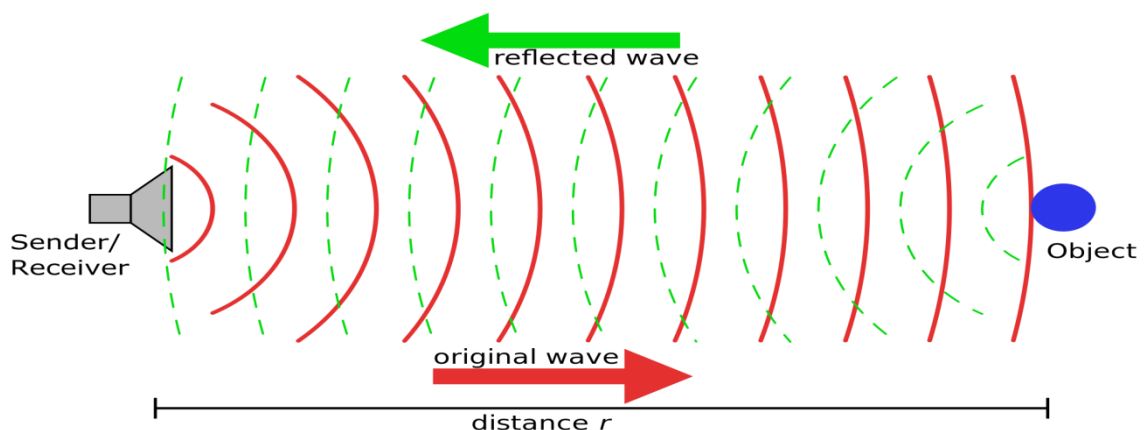
εικ. 4.4 Αισθητήρας HC- SR04

Εφαρμογές τους θα βρούμε σε ένα μεγάλο εύρος τεχνολογιών όπως την ανίχνευση προσέγγισης, την παρουσία ή την απουσία αντικειμένων, την ανίχνευση εμποδίων αυτοματοποιημένα οχήματα, τη μέτρηση απόστασης, τη μέτρηση στάθμης, τη μέτρηση της διεύθυνσης και της ταχύτητας του ανέμου. Η κατανάλωση ισχύος του αισθητήρα είναι της τάξης των 30m A έως 35m A το πολύ, ενώ η τάση λειτουργίας του είναι στα 5V.



εικ 4.5 Χρονοδιάγραμμα του αισθητήρα

Στον Arduino η διαδικασία αυτή (αντιστοίχιση χρόνου σε απόσταση) είναι αρκετά απλή διότι υπάρχει συγκεκριμένη βιβλιοθήκη για τον αισθητήρα HC-SR04 που κάνει την "δύσκολη" δουλειά. Εμείς στον κώδικά μας αρκεί να κάνουμε χρήση μόνο εντολής. Η εντολή αυτή θα μας επιστρέψει την απόσταση σε εκατοστά (cm).



εικ. 4.6 Πομπός και δέκτης (αντανάκλαση κύματος)

## 4.3 Σύνδεση με τον Arduino

Για τη σύνδεση των αισθητήρων με την πλακέτα Arduino χρησιμοποιήθηκαν τέσσερα καλώδια. Η σύνδεση των αισθητήρων με την πλακέτα του Arduino παρατίθεται παρακάτω. Ο αισθητήρας HC-SR04 διαθέτει τέσσερα (4) pin.

-το Vcc που το συνδέουμε στον υποδοχέα των 5V του Arduino

-το Trigger, το οποίο είναι έξοδος από την οποία γίνεται αποστολή σήματος αφού δοθεί τάση και το συνδέουμε με το 3B της πλακέτας (Raster) και τον υποδοχέα 9 του Arduino

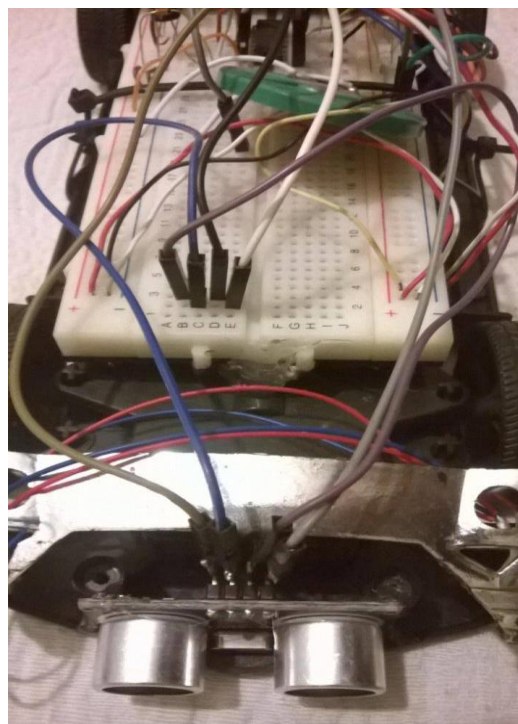
-το Echo, που είναι η είσοδος η οποία λαμβάνει το σήμα, το οποίο έχει σταλεί από πριν και το συνδέουμε με το 2C της πλακέτας (Raster) και τον υποδοχέα 12 του Arduino

-το GND που το συνδέουμε από τη γείωση της πλακέτας (Raster) στη γείωση (-) του Arduino.

Στις εικόνες παρακάτω φαίνεται ακριβώς η σύνδεση την οποία αναλύουμε παραπάνω.



εικ 4.7 Σύνδεση των αισθητήρων με την πλακέτα Arduino



εικ 4.8 Ομοίως με εικ.4.7



## ΚΕΦΑΛΑΙΟ 5

### 5.1 Ηχείο- Sounder

Ένα επίσης απλό εξάρτημα που μπορούμε να συνδέσουμε με τον Arduino μας είναι ένα ηχείο. Δουλεύει ακριβώς όπως ένα LED, με δύο καλώδια, ένα για την πηγή (5V) κι ένα για τη γείωση (GND). Αν του δώσουμε σταθερή τάση (π.χ. 5V) θα μας δώσει σταθερό ήχο, ενώ μεταβλητή τάση (π.χ. αντί στην πηγή (5V), σύνδεσή του με ένα PWM pin στον Arduino και αυξομείωση της τάσης που του παρέχεται θα μας δώσει μεταβλητό ήχο, δηλαδή ηχητικό εφέ.

Εκτός από τα απλά ηχεία υπάρχουν και διατάξεις ηχείων που αλλάζουν τις ιδιότητές τους ανάλογα με την πίεση που τους ασκείται. Η συνδεσμολογία τους είναι ίδια και μπορούμε να τα χρησιμοποιήσουμε για να ανιχνεύσουμε ασκούμενη δύναμη πάνω τους. Στο δικό μας κύκλωμα χρησιμεύει για την ηχητική ενημέρωση κάποιου εμποδίου που συναντά.



εικ. 5.1 Ηχείο

### 5.2 Σύνδεση ηχείου στον Arduino

Στο ηχείο μας εξέχουν δυο καλώδια (+/-). Το καλώδιο (+) το συνδέουμε στη γείωση του Raster που είναι στο αρνητικό ημισφαίριο και το καλώδιο (-) το συνδέουμε στο A21 της πλακέτας Raster το οποίο με τη σειρά του με τη χρήση καλωδίου, συνδέεται στον υποδοχέα 6 του Arduino.

## ΚΕΦΑΛΑΙΟ 6

### 6.1 Αρχικοποίηση Μεταβλητών

```
int Enable1FB = 11; // PWM για κίνηση μπροστά πίσω
int Enable2LR = 10; // PWM για κίνηση δεξιά αριστερά

int Input1FB = 8; //
int Input2FB = 7; // LOW + HIGH κινήτε πίσω HIGH + LOW κινήτε μπροστά

int Input3LR = 4; //
int Input4LR = 2; // HIGH + LOW κινήτε δεξιά LOW + HIGH κινήτε αριστερά

int echoPin1 = 12; // Echo Pin για το μέσω
int trigPin1 = 9; // Trigger Pin για το μέσω
int echoPin2 = A1; // Echo Pin για το αριστερα
int trigPin2 = 3; // Trigger Pin για το μέσω
int echoPin3 = A0; // Echo Pin για το δεξιό
int trigPin3 = 5; // Trigger Pin για το δεξιο

int LEDPin = 13; // Onboard LED
int Altop = 6; // altop

int maximumRange = 50; // Maximum range needed
int minimumRange = 0; // Minimum range needed
long duration1, distance1; // Duration used to calculate distance
long duration2, distance2;
```

**long duration3, distance3;**

Όπως βλέπουμε στο παραπάνω πρόγραμμα, πρώτα αρχίζουμε με τη δήλωση των μεταβλητών των διαφόρων περιφερειακών συσκευών που έχουμε συνδέσει επάνω στον Arduino . Πιο συγκεκριμένα, δηλώνουμε τους ακροδέκτες και τα Enable των κινητήρων όπου Enable 1FB και Enable 2LR τους οποίους συνδέσαμε στους υποδοχείς 11 και 10 αντίστοιχα από τους οποίους τροφοδοτείται με ρεύμα το ολοκληρωμένο κύκλωμα. Το 1FB ενεργοποιεί τον κινητήρα μπρος-πίσω και το 2LR ενεργοποιεί τον κινητήρα δεξιά-αριστερά.

Τα Input 1FB και Input 2FB τα συνδέουμε στους υποδοχείς 8 και 7 αντίστοιχα και ενεργοποιούν την κίνηση μπρος-πίσω. Τα Input3LR και Input4LR τα συνδέουμε στους υποδοχείς 4 και 2 αντίστοιχα και ενεργοποιούν την κίνηση δεξιά-αριστερά.

Για τους αισθητήρες που συνδέσαμε στον Arduino αρχικοποιούμε το echoPin στον υποδοχέα 12,A1,A0 και το trigPin στον υποδοχέα 9,3,5. Αυτά στέλνουν και λαμβάνουν το σήμα.

Το Altop (ηχείο) το οποίο συνδέουμε στον υποδοχέα 6, αρχικοποιεί το ηχείο.

## 6.2 Κώδικας που εκτελείται μια φορά

```
void setup() {  
  // put your setup code here, to run once:  
  pinMode(Enable1FB , OUTPUT); // Θέτουμε το PIN έξοδο  
  pinMode(Enable2LR , OUTPUT); // Θέτουμε το PIN έξοδο  
  pinMode(Input1FB , OUTPUT); // Θέτουμε το PIN έξοδο  
  pinMode(Input2FB , OUTPUT); // Θέτουμε το PIN έξοδο  
  pinMode(Input3LR , OUTPUT); // Θέτουμε το PIN έξοδο  
  pinMode(Input4LR , OUTPUT); // Θέτουμε το PIN έξοδο  
  pinMode(trigPin1 , OUTPUT); // Θέτουμε το PIN έξοδο  
  pinMode(echoPin1 , INPUT); // Θέτουμε το PIN είσοδο  
  pinMode(trigPin2 , OUTPUT); // Θέτουμε το PIN έξοδο  
  pinMode(echoPin2 , INPUT); // Θέτουμε το PIN είσοδο  
  pinMode(trigPin3 , OUTPUT); // Θέτουμε το PIN έξοδο  
  pinMode(echoPin3 , INPUT); // Θέτουμε το PIN είσοδο  
  pinMode(LEDpin , OUTPUT); // Use LED indicator (if required)  
  pinMode(Altop , OUTPUT); // Θέτουμε το PIN έξοδο  
}
```

Όπως εξηγείται στα σχόλια δίπλα από κάθε εντολή, κάποια τα θέτουμε ως εξόδους και κάποια ως εισόδους.

## 6.3 Κυρίως πρόγραμμα

```
void loop() {  
  
  // Μέτρηση της διαδικασίας για να εκπέμψουμε ένα σήμα (μια ριπή) για να  
  μετρήσουμε την απόσταση  
  
  digitalWrite(trigPin1, LOW);  
  
  delayMicroseconds(2);  
  
  
  digitalWrite(trigPin1, HIGH);  
  
  delayMicroseconds(10);  
  
  
  digitalWrite(trigPin1, LOW);  
  
  duration1 = pulseIn(echoPin1, HIGH);  
  
  
  //Calculate the distance (in cm) based on the speed of sound.  
  
  distance1 = duration1/58.2;  
  
  
  // Μέτρηση  
  
  digitalWrite(trigPin2, LOW);  
  
  delayMicroseconds(2);  
  
  
  digitalWrite(trigPin2, HIGH);  
  
  delayMicroseconds(10);  
  
  
  digitalWrite(trigPin2, LOW);  
  
  duration2 = pulseIn(echoPin2, HIGH);
```

```

//Calculate the distance (in cm) based on the speed of sound.
distance2 = duration2/58.2;

// Μέτρηση
digitalWrite(trigPin3, LOW);
delayMicroseconds(2);

digitalWrite(trigPin3, HIGH);
delayMicroseconds(10);

digitalWrite(trigPin3, LOW);
duration3 = pulseIn(echoPin3, HIGH);

//Calculate the distance (in cm) based on the speed of sound.
distance3 = duration3/58.2;

if (distance1 >= maximumRange || distance1 <= minimumRange){
    // Συγκρίνει την απόσταση για μια χρονική στιγμή και αν το εμπόδιο είναι μακριά
    δεν παράγει ήχο

    digitalWrite(LEDPin, HIGH);

    // Σύγκριση της απόστασης για τον ήχο από το ηχείο αν η απόσταση είναι μικρότερη
    από μια σταθερά (δηλ. το όχημα πλησιάζει το εμπόδιο)

if (distance1 < 60){
analogWrite(Altop, 125);
delay(distance1*10);

```

```

analogWrite(Altop, 0);
}
if (distance1 <= 3){
analogWrite(Altop, 125);
}

// Μεγάλη απόστασης
digitalWrite(LEDpin, LOW);
// Κίνηση μπροστά
digitalWrite(Input1FB , HIGH); // HIGH + LOW τότε θα κινηθεί μπροστά
digitalWrite(Input2FB , LOW);
analogWrite(Enable1FB , 200); // Ταχύτητα του μοτόρ
}

else {
//Κίνηση μπροστά πίσω (πίσω)
digitalWrite(Input1FB , LOW); // LOW + HIGH τότε θα κινηθεί πίσω
digitalWrite(Input2FB , HIGH);
analogWrite(Enable1FB , 180); // Ταχύτητα του μοτόρ
delay(80);
digitalWrite(Enable1FB , LOW); // Σταματάω το μοτόρ
delay(500);

digitalWrite(Input3LR , LOW); // HIGH + LOW τότε θα κινηθεί δεξιά
digitalWrite(Input4LR , HIGH);
analogWrite(Enable2LR , 255); // Κίνηση του μοτόρ
delay(2000);

```

```
digitalWrite(Enable1FB , LOW); // Σταματάω το μοτορ  
delay(500);
```

```
digitalWrite(Input1FB , LOW); // LOW + HIGH τότε θα κινηθεί πίσω  
digitalWrite(Input2FB , HIGH);  
analogWrite(Enable1FB , 180); // Ταχύτητα του μοτόρ  
delay(1000);  
digitalWrite(Enable1FB , LOW); // Σταματάω το μοτόρ  
delay(500);
```

```
digitalWrite(Input3LR , HIGH); // HIGH + LOW τότε θα κινηθεί αριστερά  
digitalWrite(Input4LR , LOW);  
digitalWrite(Enable2LR , HIGH); // Επαναφορά του μοτόρ  
delay(200);  
digitalWrite(Enable2LR , LOW);  
delay (1000);
```

*// Συγκρίνει την απόσταση για μια χρονική στιγμή και αν το εμπόδιο είναι μακριά  
δεν παράγει ήχο*

```
digitalWrite(LEDpin, HIGH);
```

*// Σύγκριση της απόστασης για τον ήχο από το ηχείο αν η απόσταση είναι μικρότερη  
από μια σταθερά (δηλ. το όχημα πλησιάζει το εμπόδιο)*

```
if (distance1 < 60){  
analogWrite(Altop, 125);
```



```
delay(distance1*10);  
analogWrite(Altop, 0);  
}  
if (distance1 <= 3){  
analogWrite(Altop, 125);  
}  
}
```

```
if (distance2 >= maximumRange || distance2 <= minimumRange){
```

```
// Συγκρίνει την απόσταση για μια χρονική στιγμή και αν το εμπόδιο είναι μακριά  
δεν παράγει ήχο
```

```
digitalWrite(LEDpin, HIGH);
```

```
// Σύγκριση της απόστασης για τον ήχο από το ηχείο αν η απόσταση είναι μικρότερη  
από μια σταθερά (δηλ. το όχημα πλησιάζει το εμπόδιο)
```

```
if (distance2 < 60){  
analogWrite(Altop, 125);  
delay(distance2*10);  
analogWrite(Altop, 0);  
}  
if (distance2 <= 3){  
analogWrite(Altop, 125);  
}
```

```
// Μεγάλη απόστασης
```

```

digitalWrite(LEDPin, LOW);
// Κίνηση μπροστά

// Μεγάλη απόστασης
digitalWrite(LEDPin, LOW);
// Κίνηση μπροστά

digitalWrite(Input1FB , HIGH); // HIGH + LOW τότε θα κινηθεί μπροστά
digitalWrite(Input2FB , LOW);
analogWrite(Enable1FB , 200); // Ταχύτητα του μοτόρ

}

else {
//Κίνηση μπροστά πίσω (πίσω)
digitalWrite(Input1FB , LOW); // LOW + HIGH τότε θα κινηθεί πίσω
digitalWrite(Input2FB , HIGH);
analogWrite(Enable1FB , 180); // Ταχύτητα του μοτόρ
delay(80);
digitalWrite(Enable1FB , LOW); // Σταματάω το μοτόρ
delay(500);

digitalWrite(Input3LR , LOW); // HIGH + LOW τότε θα κινηθεί δεξιά
digitalWrite(Input4LR , HIGH);
analogWrite(Enable2LR , 255); // Κίνηση του μοτόρ
delay(2000);

digitalWrite(Enable1FB , LOW); // Σταματάω το μοτορ

```

**delay(500);**

**digitalWrite(Input1FB , LOW);** // LOW + HIGH τότε θα κινηθεί πίσω

**digitalWrite(Input2FB , HIGH);**

**analogWrite(Enable1FB , 180);** // Ταχύτητα του μοτόρ

**delay(1000);**

**digitalWrite(Enable1FB , LOW);** // Σταματάω το μοτόρ

**delay(500);**

**digitalWrite(Input3LR , HIGH);** // HIGH + LOW τότε θα κινηθεί αριστερά

**digitalWrite(Input4LR , LOW);**

**digitalWrite(Enable2LR , HIGH);** // Επαναφορά του μοτόρ

**delay(200);**

**digitalWrite(Enable2LR , LOW);**

**delay (1000);**

// Συγκρίνει την απόσταση για μια χρονική στιγμή και αν το εμπόδιο είναι μακριά  
δεν παράγει ήχο

**digitalWrite(LEDpin, HIGH);**

// Σύγκριση της απόστασης για τον ήχο από το ηχείο αν η απόσταση είναι μικρότερη  
από μια σταθερά (δηλ. το όχημα πλησιάζει το εμπόδιο)

**if (distance2 < 60){**

**analogWrite(Altop, 125);**

**delay(distance2\*10);**

**analogWrite(Altop, 0);**

```
}  
  
if (distance2 <= 3){  
analogWrite(Altop, 125);  
}  
}
```

```
if (distance3 >= maximumRange || distance3 <= minimumRange){
```

```
// Σύγκριση της απόστασης για τον ήχο από το ηχείο αν η απόσταση είναι μικρότερη  
από μια σταθερά (δηλ. το όχημα πλησιάζει το εμπόδιο)
```

```
digitalWrite(LEDPin, HIGH);
```

```
// Συγκρίνει την απόσταση για μια χρονική στιγμή και αν το εμπόδιο είναι μακριά δεν  
παράγει ήχο
```

```
if (distance3 < 60){  
analogWrite(Altop, 125);  
delay(distance3*10);  
analogWrite(Altop, 0);  
}  
if (distance3 <= 3){  
analogWrite(Altop, 125);  
}
```

```
// Μεγάλη απόστασης
```

```
digitalWrite(LEDPin, LOW);
```

```
// Κίνηση μπροστά
```

```
digitalWrite(Input1FB , HIGH); // HIGH + LOW τότε θα κινηθεί μπροστά
```

```

digitalWrite(Input2FB , LOW);
analogWrite(Enable1FB , 200); // Ταχύτητα του μοτόρ

}

else {

//Κίνηση μπροστά πίσω (πίσω)

digitalWrite(Input1FB , LOW); // LOW + HIGH τότε θα κινηθεί πίσω
digitalWrite(Input2FB , HIGH);
analogWrite(Enable1FB , 180); // Ταχύτητα του μοτόρ
delay(80);
digitalWrite(Enable1FB , LOW); // Σταματάω το μοτόρ
delay(500);

digitalWrite(Input3LR , LOW); // HIGH + LOW τότε θα κινηθεί δεξιά
digitalWrite(Input4LR , HIGH);
analogWrite(Enable2LR , 255); // Κίνηση του μοτόρ
delay(2000);

digitalWrite(Enable1FB , LOW); // Σταματάω το μοτορ
delay(500);

digitalWrite(Input1FB , LOW); // LOW + HIGH τότε θα κινηθεί πίσω
digitalWrite(Input2FB , HIGH);
analogWrite(Enable1FB , 180); // Ταχύτητα του μοτόρ
delay(1000);
digitalWrite(Enable1FB , LOW); // Σταματάω το μοτόρ

```

```
delay(500);
```

```
digitalWrite(Input3LR , HIGH); // HIGH + LOW τότε θα κινηθεί αριστερά
```

```
digitalWrite(Input4LR , LOW);
```

```
digitalWrite(Enable2LR , HIGH); // Επαναφορά του μοτόρ
```

```
delay(200);
```

```
digitalWrite(Enable2LR , LOW);
```

```
delay (1000);
```

Ο κώδικας παραπάνω χρησιμοποιεί την digitalWrite η οποία γράφει μια υψηλή (HIGH) και μια χαμηλή (LOW) τιμή σε μια ψηφιακή ακίδα. Αν η ακίδα έχει ρυθμιστεί ως έξοδος με τη συνάρτηση pinMode( ), τότε η τάση της θα καθορίσει στην αντίστοιχη τιμή: 5V για HIGH και 0V για LOW.

Επίσης χρησιμοποιεί και την analogWrite η οποία γράφει μια αναλογική τιμή (PWM κύμα) σε μια ακίδα. Μπορεί να χρησιμοποιηθεί για παράδειγμα να ανάψει ένα LED σε διάφορες φωτεινότητες ή να οδηγήσει έναν κινητήρα σε διάφορες ταχύτητες. Μετά από μια κλήση της analogWrite ( ), η ακίδα θα δημιουργήσει ένα σταθερό τετραγωνικό κύμα του καθορισμένου κύκλου λειτουργίας μέχρι την επόμενη κλήση της analogWrite ( ) (ή μια κλήση της digitalWrite( ) ή digitalRead( ) για την ίδια την ακίδα). Η συχνότητα του σήματος PWM είναι περίπου 490 Hz.

Το delay που χρησιμοποιούμε στο πρόγραμμά μας είναι για την παύση/αναμονή για όσο χρόνο προβλέπεται από την κάθε εντολή.

### 6.3.1 Εντολές εντός κυρίου προγράμματος για την δήλωση του ήχου όταν όχημα πλησιάζει σε εμπόδιο

// Σύγκριση της απόστασης για τον ήχο από το ηχείο αν η απόσταση είναι μικρότερη από μια σταθερά (δηλ. το όχημα πλησιάζει το εμπόδιο)

**digitalWrite(LEDpin, HIGH);** // Ανάβει το LED

// Συγκρίνει την απόσταση για μια χρονική στιγμή και αν το εμπόδιο είναι μακριά δεν παράγει ήχο

**if (distance < 60){**

**analogWrite(Altop, 125);**

**delay(distance\*10);**

**analogWrite(Altop, 0);**

**}**

// Συγκρίνει την απόσταση για μια χρονική στιγμή και αν το εμπόδιο είναι κοντά τότε παράγει ήχο

**if (distance <= 3){**

**analogWrite(Altop, 125);**

**}**

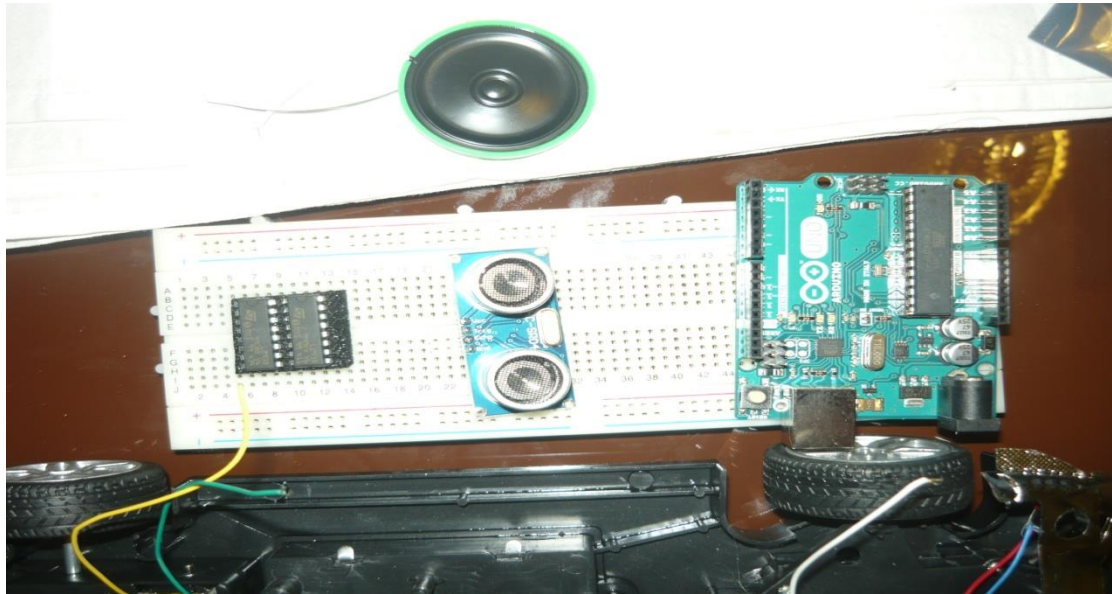
**}**

// Καθυστέρηση για την επόμενη σύγκριση

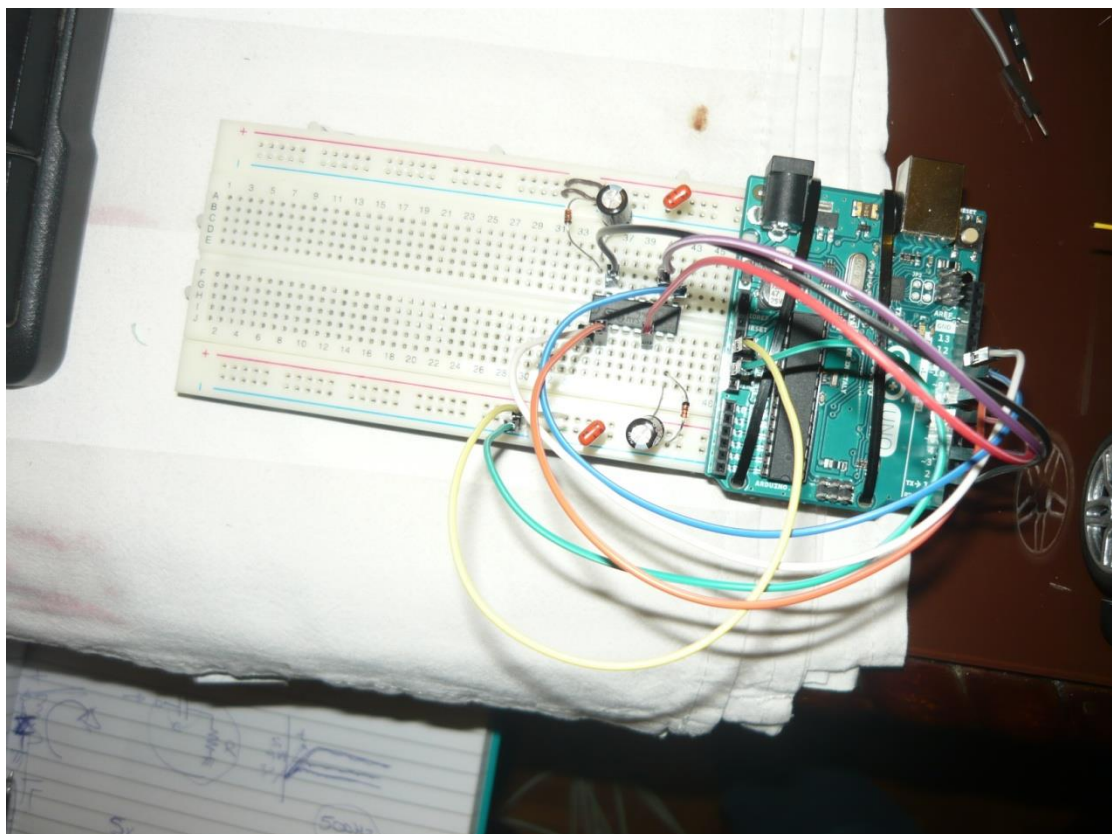
**delay(100);**

**}**

**Διάφορες εικόνες που τραβήχτηκαν κατά την διαδικασία υλοποίησης της εργασίας.**

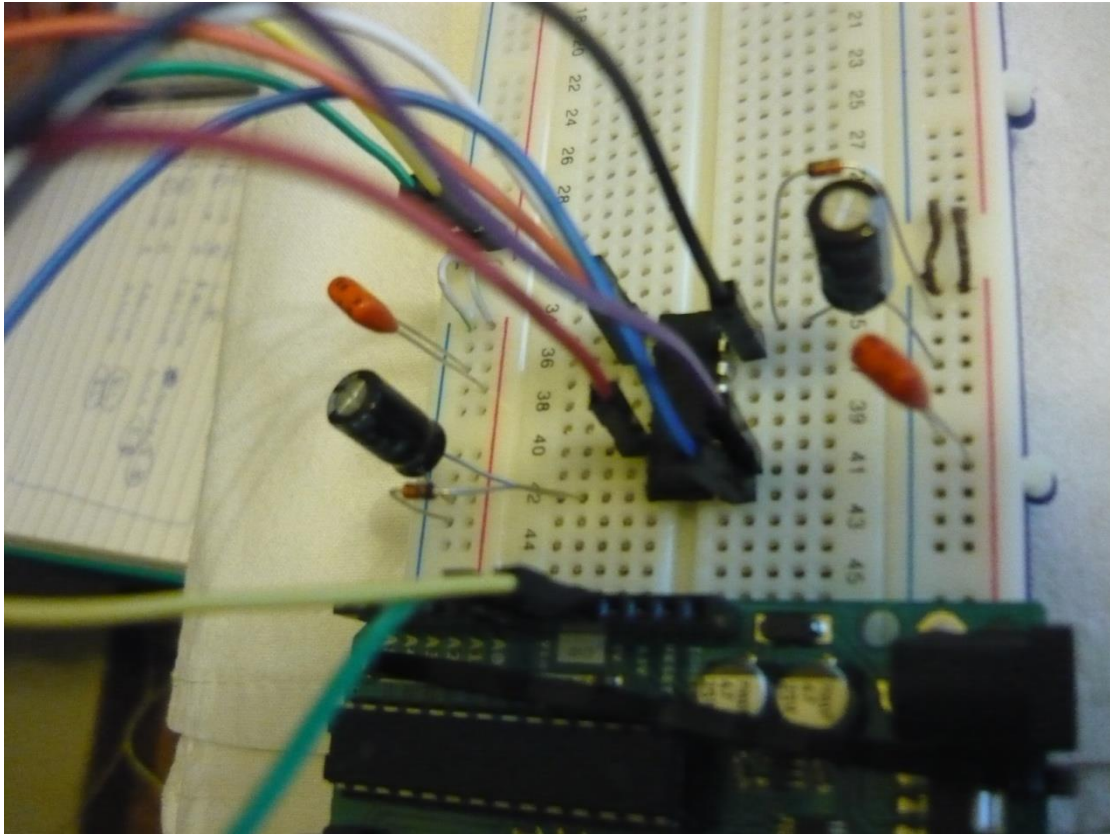


εικ.1 Arduino UNO – L293D – Ηχείο - Αισθητήρας

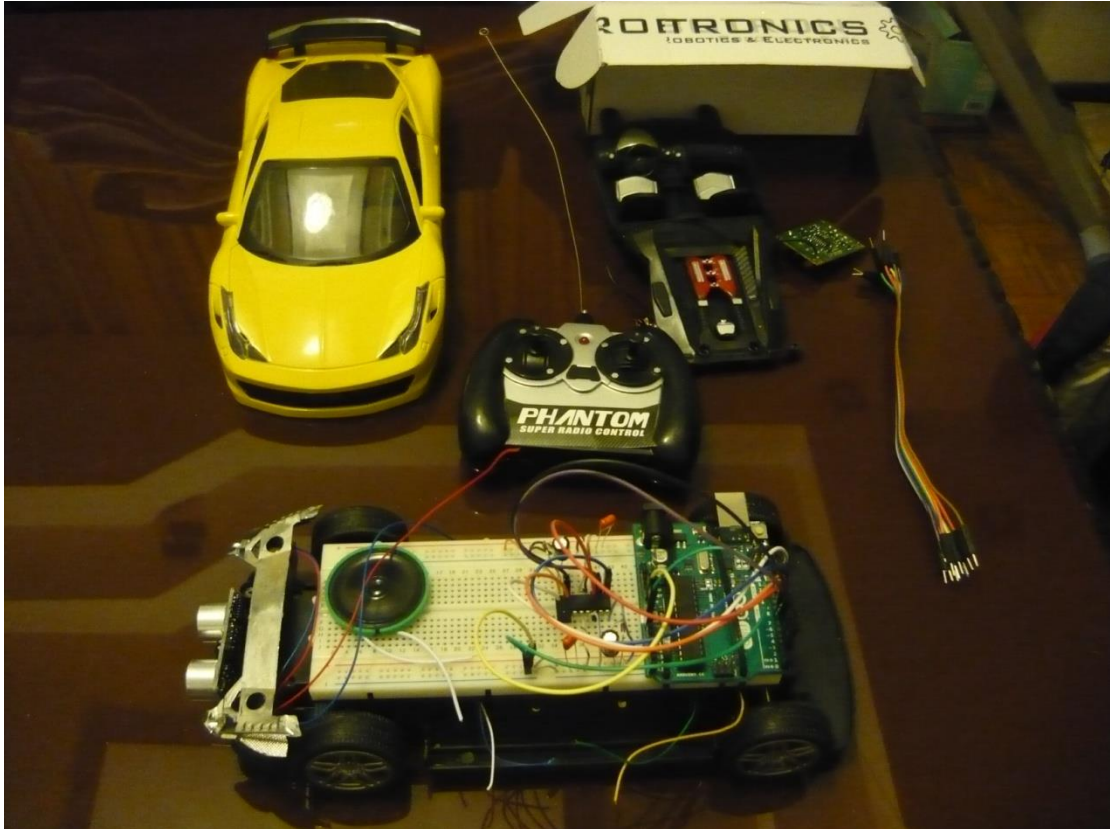


εικ.2 Σύνδεση πρώτης φάσης Arduino με ολοκληρωμένο κύκλωμα

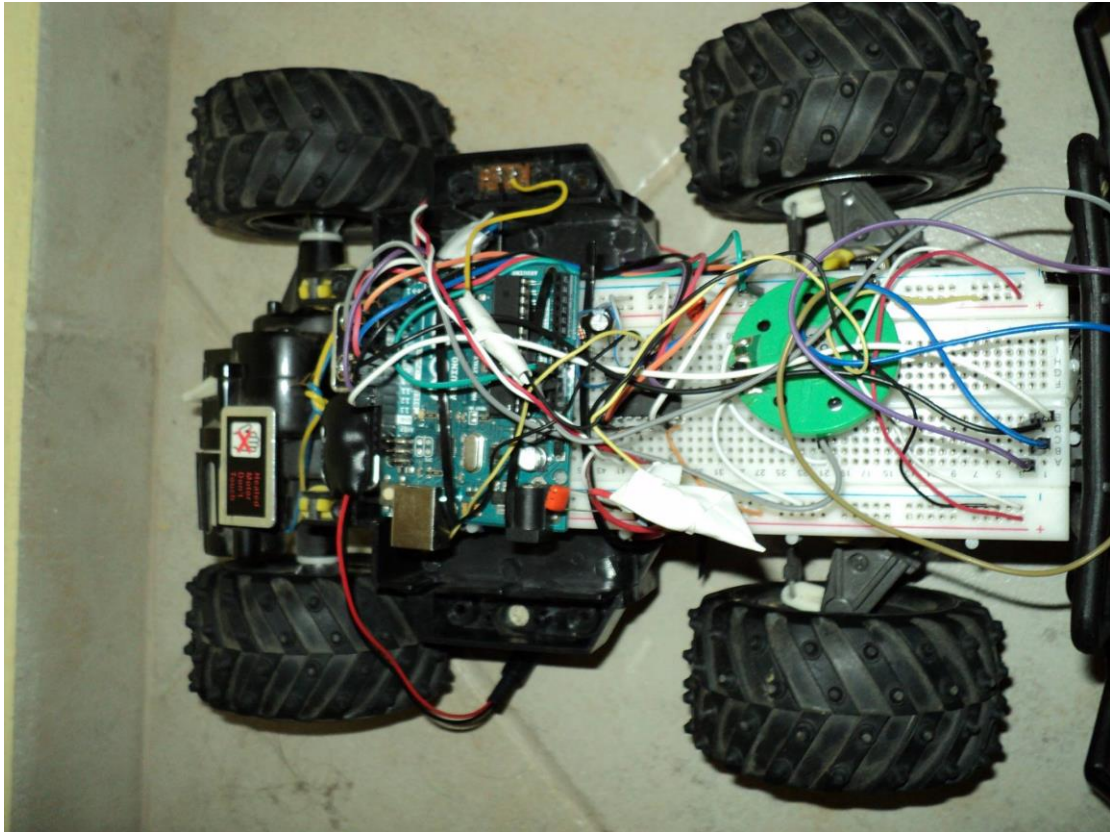




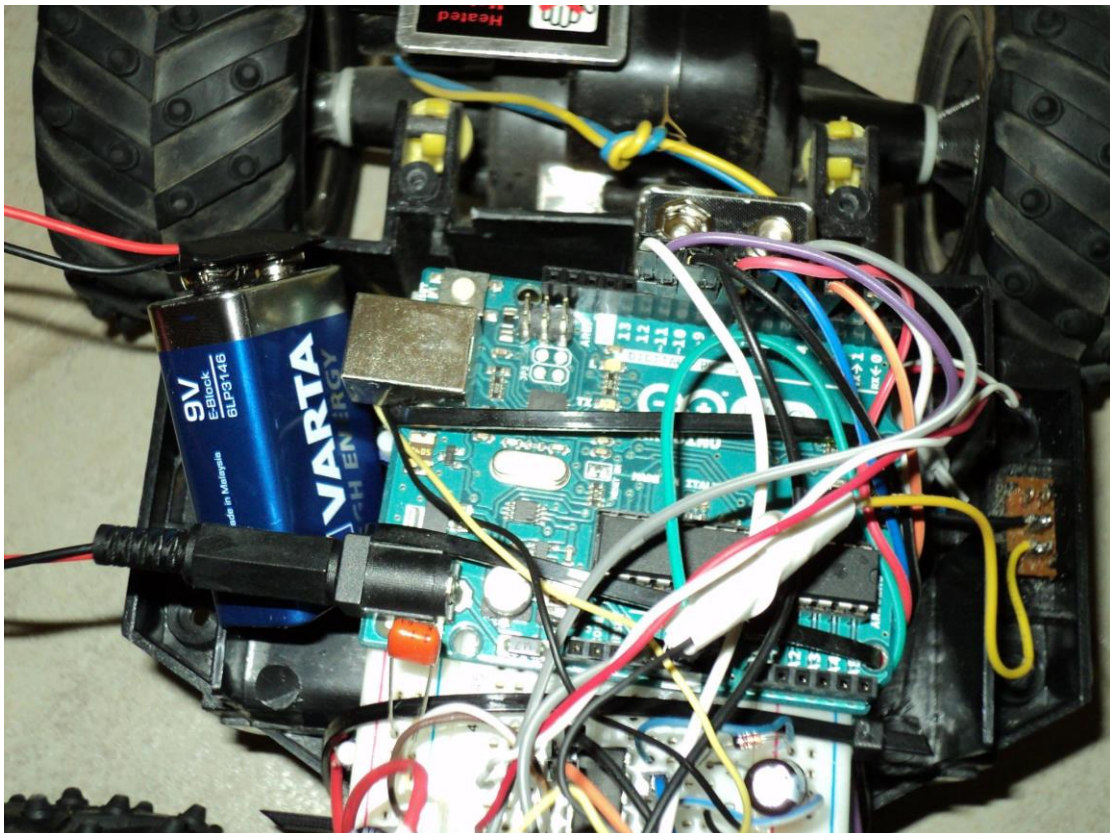
εικ.3 Ομοίως με την εικ.2

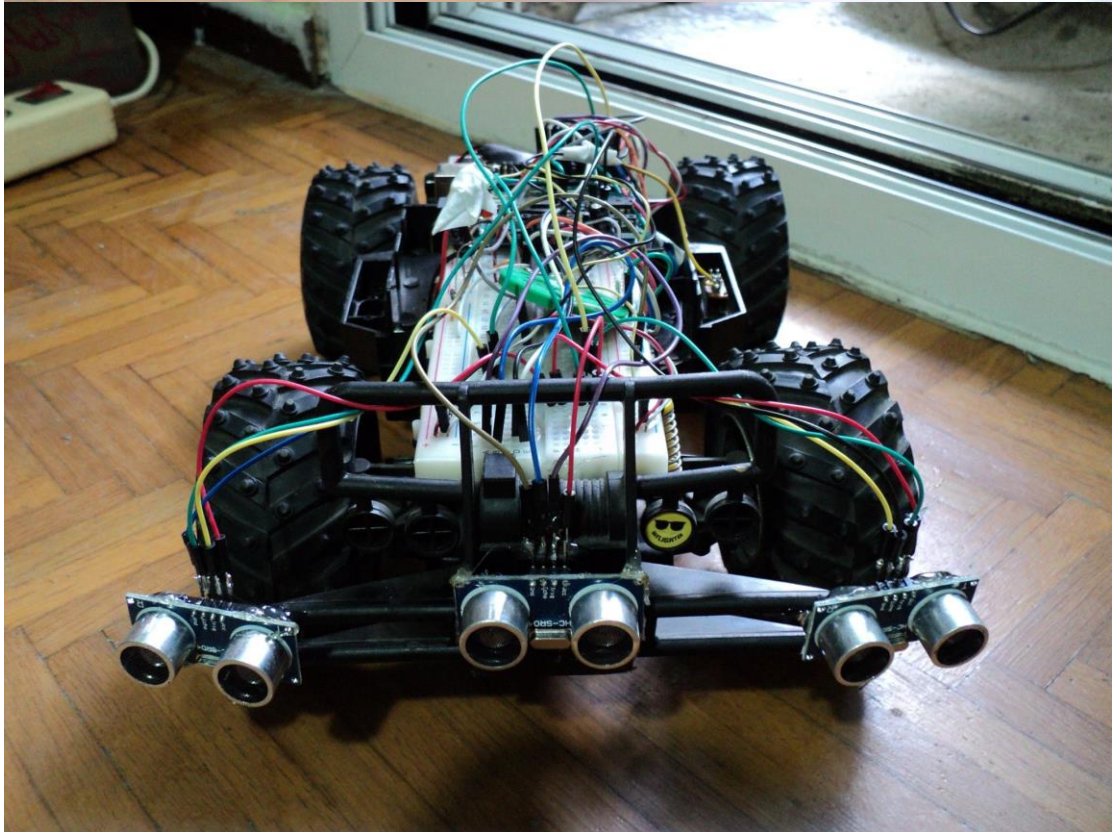
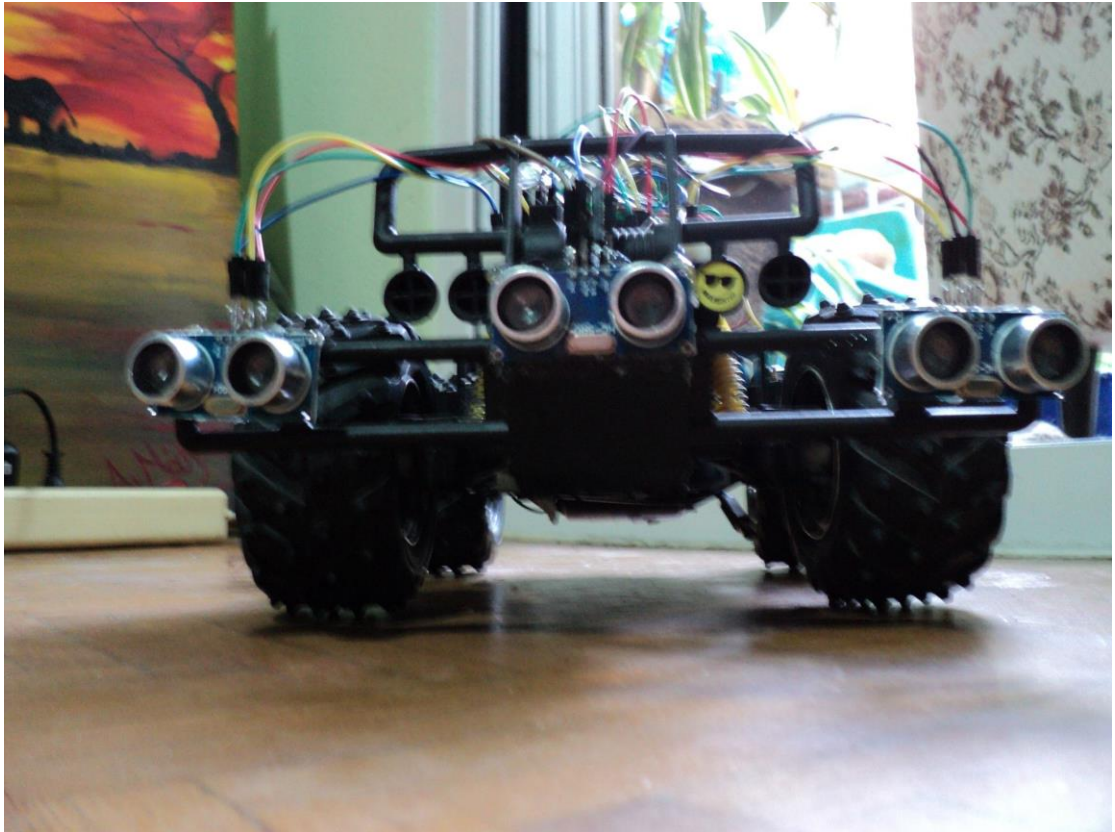


εικ.4 Όλα τα υλικά που αφαιρέσαμε και προσθέσαμε στον Arduino

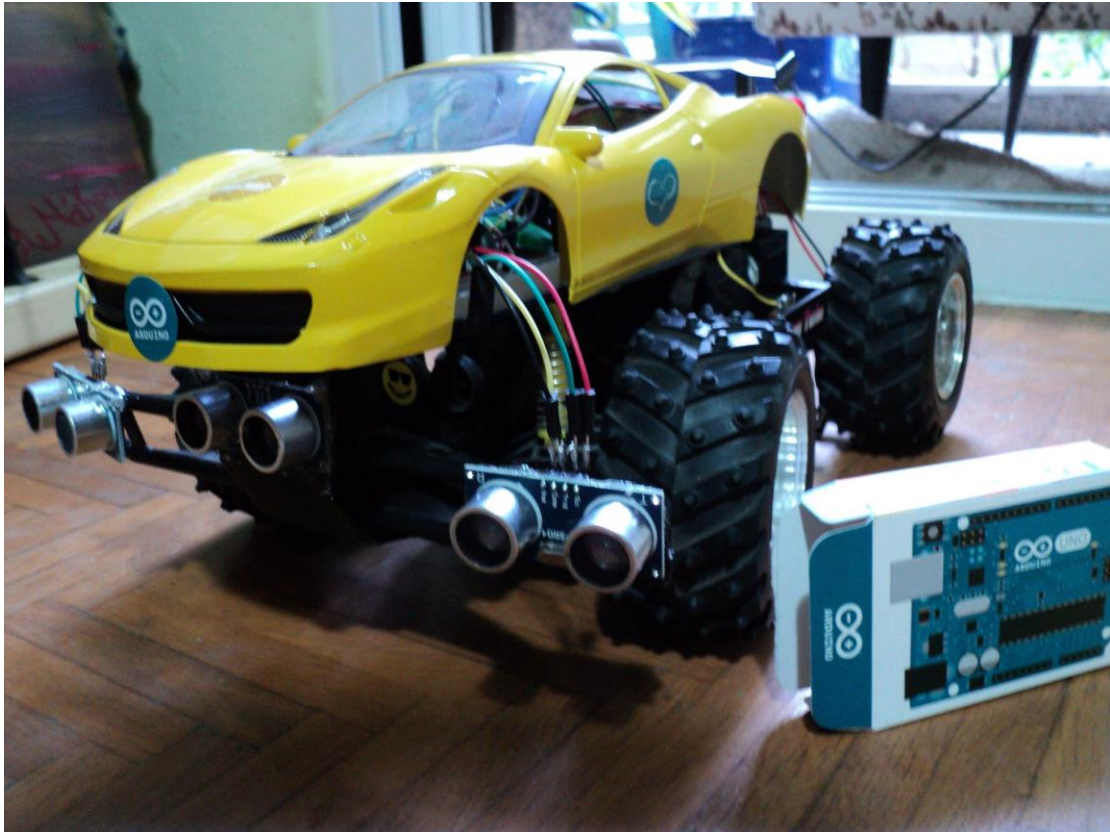


εικ.5









## **Προβλήματα που παρουσιάστηκαν κατά την υλοποίηση του αυτόνομου ρομποτικού αυτοκινήτου**

Κατά τη διαδικασία της υλοποίησης του αυτόνομου ρομποτικού αυτοκινήτου παρουσιάστηκαν κάποια προβλήματα τα οποία και στη συνέχεια αντιμετωπίστηκαν με επιτυχία.

Αρχικά κίνησε ο ένας κινητήρας και συγκεκριμένα ο πίσω, λόγω υπερφόρτωσης, που υπήρχε ήδη προσαρτημένος στο τηλεκατευθυνόμενο αυτοκίνητο που χρησιμοποιήθηκε ως βάση του αυτόνομου. Έτσι, αναγκαστικά έπρεπε να αλλάξει, όπως κι έγινε. Στην συνέχεια έγινε αντιληπτό ότι το πρώτο αμαξίδιο δεν άντεχε το βάρος για την κίνηση προς τα πίσω, το βοηθούσε το καλώδιο της τροφοδοσίας για να κινηθεί ενώ με την μπαταρία δεν μπορούσε και αντικαταστάθηκε ο βραχίονας του αμαξιδίου. Η αλλαγή φαίνεται στις εικόνες τις εργασίας.

Άλλο ένα πρόβλημα παρουσιάστηκε στον τρόπο της σύνταξης του προγράμματος. Ειδικότερα, στο πόσο ταχύτητα θα δοθεί στους κινητήρες (τάση), ώστε να μπορεί να κινηθεί το όχημα και να μην καούν οι κινητήρες, όπως συνέβη αρχικά. Το πρόβλημα αντιμετωπίστηκε επίσης με επιτυχία.

Επιπρόσθετα, μετά από πολλές δοκιμές διαπιστώθηκε ότι η μπαταρία δε διαρκεί πολύ και πρέπει κάθε φορά να αλλάζουν οι ταχύτητες του προγράμματος. Λόγω του ότι πέφτει η τάση γρήγορα και το αμαξίδιο δεν εκπληρώνει σωστά τις εντολές που του δίνονται. Αυτό μπορεί να επιλυθεί με μια επαναφορτιζόμενη μπαταρία ή αλλιώς με εξωτερική τροφοδοσία η οποία θα έχει μακρύ καλώδιο για να κινείται εύκολα στο χώρο.

## **Σχόλια και συμπεράσματα**

Η πτυχιακή αυτή, η οποία ασχολήθηκε με την υλοποίηση ενός αυτόνομου ρομποτικού αυτοκινήτου με αισθητήρες και ήχο, δεν είχε μόνο ως στόχο την καθαυτού υλοποίηση του αυτοκινούμενου ρομπότ αλλά και την περιήγηση του αναγνώστη της στον θαυμαστό κόσμο της Ρομποτικής Τεχνολογίας.

Επιχειρήθηκε αρχικά, μέσα από μια ιστορική αναδρομή στο παρελθόν της τεχνολογίας και της επιστήμης, ο αναγνώστης να γνωρίσει την χρήση και τη λειτουργία της Ρομποτικής. Στη συνέχεια, έγινε παράθεση της πλατφόρμας Arduino και μια πρώτη γνωριμία με αυτήν, αναφέρθηκαν οι διάφοροι τύποι που υπάρχουν και αναλύθηκε ενδελεχώς η πλατφόρμα Arduino UNO, η οποία χρησιμοποιήθηκε στη συγκεκριμένη κατασκευή. Το ολοκληρωμένο κύκλωμα, ο αισθητήρας, ο κινητήρας και το ηχείο ήταν τα περιφερειακά συστήματα τα οποία χρησίμευσαν στην υλοποίηση της κατασκευής, επίσης.

Σαν συμπέρασμα λοιπόν, μπορούμε να πούμε πως το πλεονέκτημα της εν λόγω πλατφόρμας, είναι ότι κάποιος μπορεί να κατασκευάσει ένα ρομπότ, ενώ το κόστος της κατασκευής είναι κατά πολλές φορές χαμηλότερο, σε σχέση με τα συναρμολογούμενα ρομποτικά κιτ της αγοράς. Για να φτιάξει κάποιος ένα ρομπότ με τον Arduino δεν χρειάζεται εξειδικευμένες γνώσεις ηλεκτρονικής, ωστόσο απαιτεί και λίγες γνώσεις προγραμματισμού, οι οποίες καθίστανται σαφείς στην εν λόγω πτυχιακή εργασία.

## Παράρτημα Υλικών

1. Arduino UNO (με το μικροεπεξεργαστή atmega328p)



2. Καλώδια

-USB



-Για τις συνδέσεις υλικών πάνω στο breadboard (πλακέτα)

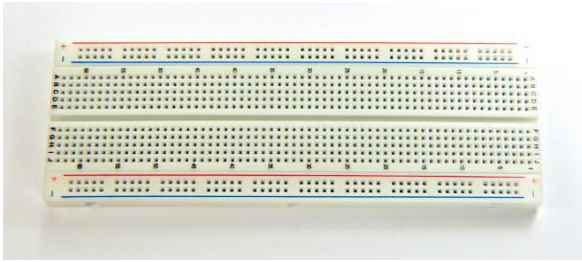


-Μπαταριών





### 3. BreadBoard (πλακέτα)



### 4. Πυκνωτές

- πλαστικοί  $2 \times 0.1 \mu\text{F}$



- πολικότητας  $2 \times 100 \mu\text{F}$



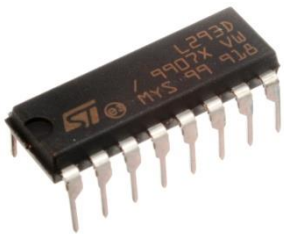
### 5. Δίοδοι $2 \times 1\text{N}4148$



5. Ηχείο (για την παραγωγή ήχου)



6. Ολοκληρωμένο κύκλωμα L293D (για την οδήγηση κινητήρων)



7. 7805 Σταθεροποιητής 5V



8. HC-SR04 αισθητήρας υπερήχων



## 9. Μπαταρίες 2x9V



Τα υλικά τα οποία χρησιμοποιήσαμε δεν ήταν ιδιαίτερα μεγάλου κόστους κι όπως προαναφέραμε στην ενότητα των σχολίων και συμπερασμάτων το κόστος της κατασκευής ενός ρομπότ δεν είναι πλέον υψηλό. Συγκεκριμένα το ρομπότ μας κόστισε ~80 €.

## **Βιβλιογραφία**

1. Παναγιώτης Παπάζογλου, Σπύρος – Πολυχρόνης Λιώνης, Ανάπτυξη εφαρμογών με το Arduino, Ένας πλήρης οδηγός για αρχάριους και προχωρημένους, Εκδόσεις Τζιόλα, 2015
2. John J. Graig, ΕΙΣΑΓΩΓΗ ΣΤΗ ΡΟΜΠΟΤΙΚΗ, Μηχανική & Έλεγχος, Μετάφραση: Νικόλαος Κωφίδης, Εκδόσεις Τζιόλα, 2008
3. ΡΟΜΠΟΤΙΚΗ 3 Έκδοση Ο.Μ Εμίρης, Επίκουρος Καθηγητής Πανεπιστημίου Πειραιά, Εκδόσεις Σελκα-4Μ-Τε
4. ΔΕ Κουπουργιότης Επίκουρος Καθηγητής –ΡΟΜΠΟΤΙΚΗ-ΑΥΤΟΜΑΤΑ, ΔΗΜ.Χ. Βούκαλης, Δρ.Μηχ. Ε.Μ.Π, Ειρήνη Βούκαλη, Μηχανολόγος Ε.Μ.Π., Σύγχρονη Έκδοση

## **Ιστοσελίδες**

1. <http://www.arduino.cc>
2. <http://www.wikipedia.org>
3. <https://deltahacker.gr/arduino-intro/>
4. <http://users.sch.gr/manpoul/docs/arduino/ProgrammingArduino.pdf>
5. [http://www.lib.ntua.gr/gr/el\\_sources/ebooks/kagiapas/CHAPTER11.pdf](http://www.lib.ntua.gr/gr/el_sources/ebooks/kagiapas/CHAPTER11.pdf)
6. [www.citylab.gr](http://www.citylab.gr)
7. <https://sites.google.com/site/texnitinoimosini/home/robot>
8. [hydrobots.gr/index/wp-content/uploads/2013/01/roboticsintro.pdf](http://hydrobots.gr/index/wp-content/uploads/2013/01/roboticsintro.pdf)