



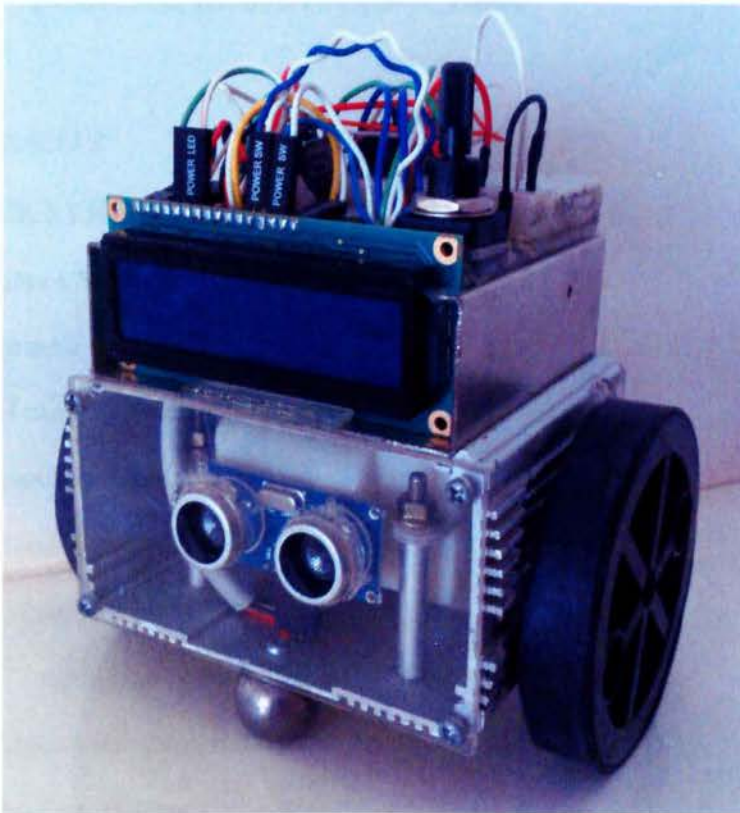
S66
A47

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Αυτοματισμού

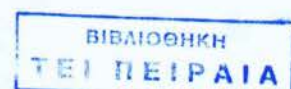
Πτυχιακή εργασία

Θέμα : ΠΡΟΓΡΑΜΜΑΤΙΣΜΕΝΟ ΠΟΛΥΧΡΗΣΤΙΚΟ ΌΧΗΜΑ



ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΜΙΧΑΗΛΣ ΠΑΠΟΥΤΣΙΔΑΚΗΣ

ΦΟΙΤΗΤΗΣ: ΠΑΛΥΒΟΣ ΓΙΩΡΓΟΣ Α.Μ.: 36280



ΑΙΓΑΛΕΩ, ΑΠΡΙΛΙΟΣ 2013

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ ΣΤΗ ΡΟΜΠΟΤΙΚΗ

1.1 Ρομποτική και Ρομπότ	4
1.1.1 Ρομποτική	4
1.1.2 Ρομπότ	4
1.2 Ιστορική αναδρομή των αυτοκινούμενων ρομπότ	5

ΚΕΦΑΛΑΙΟ 2

ΚΑΤΑΣΚΕΥΗ

2.1 Arduino UNO Rev3	27
2.1.1 Ιστορία	27
2.1.2 Η πλακέτα Arduino	28
2.1.3 Είσοδοι – έξοδοι	29
2.1.4 Τροφοδοσία	31
2.1.5 Ενσωματωμένα κουμπιά LED	32
2.1.6 Το περιβάλλον Arduino IDE	32
2.1.7 Γλώσσα προγραμματισμού	34
2.2 Αισθητήρες υπερήχων	37
2.2.1 Εισαγωγή	37
2.2.2 Αισθητήρας Υπερήχων HC-SR04	38
2.2.3 Σύνδεση με Arduino	39
2.3 Κινητήρες	40
2.3.1 Εισαγωγή	40
2.3.2 Κινητήρας DC	42

2.3.3 Οδήγηση κινητήρων	43
2.3.4 Σύνδεση με Arduino	44
2.4 Μπαταρίες	44
2.5 Οθόνη LCD	46
2.5.1 Εισαγωγή	46
2.5.2 Σύνδεση με Arduino	48
2.6 Σενάριο λειτουργίας	49

ΚΕΦΑΛΑΙΟ 3

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΟΥ ΜΙΚΡΟΕΛΕΓΚΤΗ ARDUINO

3.1 Αλγόριθμος κυρίου μέρους	50
3.1.1 Δήλωση μεταβλητών	50
3.1.2 Κυρίως πρόγραμμα	51
3.2 Αλγόριθμος συναρτήσεων	51
3.2.1 Συναρτήσεις sonarF(), sonarR()	51
3.2.2 Συναρτήσεις Start(), Stop()	52
3.2.3 Συνάρτηση Front()	53
3.2.4 Συνάρτηση Back()	53
3.2.5 Συνάρτηση Left()	54
3.2.6 Συνάρτηση Right()	54

ΚΕΦΑΛΑΙΟ 4

4.1 Βιβλιογραφία	55
4.2 Datasheet	56



ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ ΣΤΗ ΡΟΜΠΟΤΙΚΗ

1.1 Ρομποτική και Ρομπότ

1.1.1 Ρομποτική

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.

1.1.2 Ρομπότ

Ένα 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 τη λειτουργία, πρωτοσυναντάται στην Ελληνική μυθολογία. Σύμφωνα με την τελευταία 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 Čapek στο έργο του "R.U.R." (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.

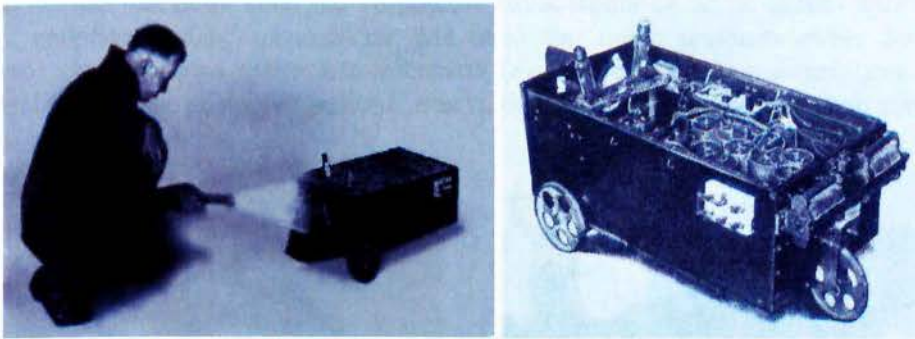
Λίγα χρόνια αργότερα, κατά την δεκαετία του '40, ο ρώσος συγγραφέας επιστημονικής φαντασίας Isaac Asimov συνέλαβε το ρομπότ ως ένα αυτόματο με εμφάνιση ανθρώπου, αλλά απαλλαγμένο από συναισθήματα. Ο όρος ρομποτική χρησιμοποιήθηκε από τον Asimov ως το σύμβολο της επιστήμης που είναι αφιερωμένη στη μελέτη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

- Το ρομπότ δε θα κάνει κακό σε άνθρωπο, ούτε με την αδράνειά του θα επιτρέψει να βλαφτεί ανθρώπινο όν.
- Το ρομπότ πρέπει να υπακούει τις διαταγές που του δίνουν οι άνθρωποι, εκτός αν αυτές οι διαταγές έρχονται σε αντίθεση με τον πρώτο νόμο.
- Το ρομπότ οφείλει να προστατεύει την ύπαρξή του, εφόσον αυτό δεν συγκρούεται με τον πρώτο ή τον δεύτερο νόμο.

1.2 Ιστορική αναδρομή των αυτοκινούμενων Ρομπότ

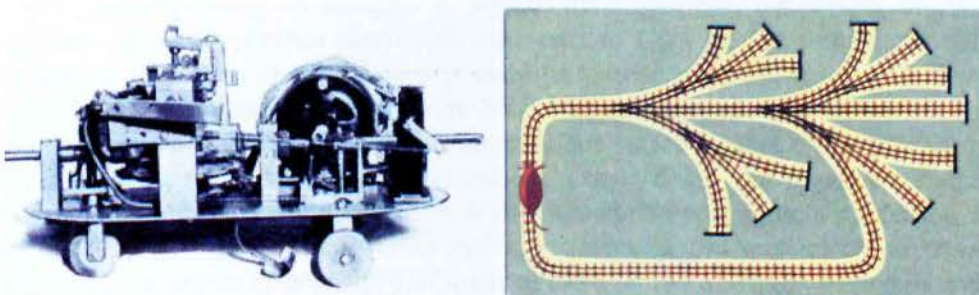
Στην παράγραφο αυτήν, θα κάνουμε μία εκτεταμένη αναφορά στην ιστορία των κινούμενων ρομπότ, η οποία και έχει στιγματίσει την πορεία της τεχνολογίας και της επιστήμης, από τα μέσα του 20ου αιώνα μέχρι σήμερα. Η εξέλιξη των κινούμενων ρομπότ προέκυψε με σκοπό την επιτέλεση διάφορων λειτουργιών, από την εκτέλεση απλών καθημερινών εργασιών, μέχρι τη διενέργεια πολύπλοκων ερευνών, ενώ σήμερα χρησιμοποιούνται και στις πλέον μεγάλης σημασίας διαστημικές αποστολές.

1912 - Electric Dog: Είναι ο πρόγονος όλων των φωτοτροπικών αυτόματων ρομπότ. Σχεδιάστηκε και κατασκευάστηκε στις ΗΠΑ από τους ερευνητές John Hammond, Jr και Benjamin Miessner οι οποίοι εργάζονταν στον τομέα της Ραδιοδυναμικής. Στην πραγματικότητα κατασκευάστηκε από τον τελευταίο. Το Electric dog ήταν εξοπλισμένο με φωτοκύτταρα, έτσι ώστε αν ο 'εχθρός' έστρεφε το φως πάνω του, αυτό αμέσως οδηγούνταν προς το μέρος του.



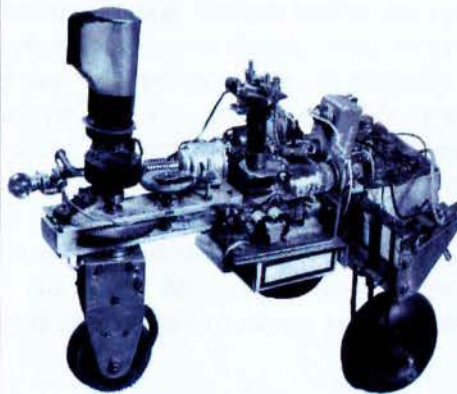
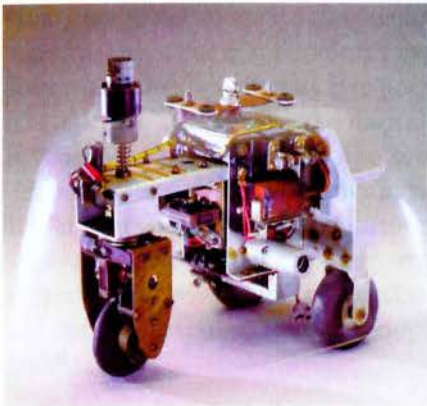
Εικόνα 1.2.1 και 1.2.2: Αριστερά το Electric Dog ακολουθεί ένα φακό και δεξιά η πίσω όψη του ρομπότ

1933 - Maze Machine Learning – Ross Thomas: Ένας μηχανικός "αρουραίος" επινοήθηκε από τον Δρ Stevenson Smith, του Πανεπιστημίου της Ουάσιγκτον. Η ηλεκτρική συσκευή κινούνταν κατά μήκος μιας διαδρομής με αυλάκια που τη χώριζαν σε διάφορα σημεία, υποχρεώνοντας τον "αρουραίο" να επιλέξει ποια κατεύθυνση να ακολουθήσει. Αν έπαιρνε το λάθος δρόμο και έμπαινε σε αδιέξοδο, οι μηχανικές 'κεραίες' του τον ανάγκαζαν να σταματήσει, να ανατρέξει την διαδρομή του και να δοκιμάσει ξανά, μέχρι η όλη πορεία του να ολοκληρωθεί με επιτυχία. Το μοντέλο σχεδιάστηκε για να μας δείξει πώς τα αντανακλαστικά διαφέρουν από τις διαδικασίες σκέψης.



Εικόνα 1.2.3 και 1.2.4: Αριστερά το Ross Thomas και δεξιά ο λαβύρινθος

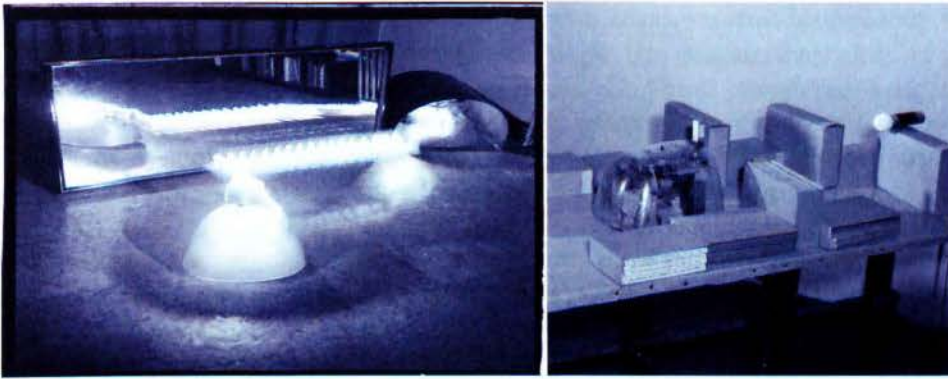
1948–1949 - Elmer και Elsie: Ο William Grey Walter κατασκεύασε τα ρομπότ Elmer και Elsie, δυο αυτόνομα ρομπότ, τα οποία ονομάστηκαν Machina Speculatrix, επειδή τους άρεσε να εξερευνούν το περιβάλλον τους. Επίσης συχνά περιγράφονται και ως χελώνες λόγω του σχήματος τους και του αργού ρυθμού κίνησής τους. Το κάθε ένα από τα ρομπότ Elmer και Elsie ήταν εξοπλισμένο με έναν αισθητήρα φωτός. Αν έβρισκαν κάποια πηγή φωτός κινούνταν προς αυτήν, αποφεύγοντας παράλληλα τα εμπόδια που βρίσκονταν στον δρόμο τους. Επίσης αν το επίπεδο ενέργειάς των μπαταριών τους έπεφτε σε χαμηλά επίπεδα τότε μπορούσαν μέσω του φωτοτροπισμού να βρουν ένα σταθμό φόρτισης και να επαναφορτιστούν. Αυτά τα ρομπότ αποδεικνύαν ότι μία πολύπλοκη συμπεριφορά μπορούσε να προκύψει από έναν απλό σχεδιασμό. Τα ρομπότ ήταν εξοπλισμένα με μια μικρή λάμπα-φλας στο πάνω μέρος τους (κεφάλι) η οποία απενεργοποιούνταν αυτόματα κάθε φορά που το φωτο-κύτταρο λάμβανε ένα επαρκές σήμα φωτός. Όταν συναντούσαν ένα κάτοπτρο ή μια λευκή επιφάνεια, το ανακλώμενο φως από το κεφάλι των ρομπότ ήταν επαρκές για τη λειτουργία ελέγχου απόκρισης του κυκλώματος των ρομπότ στο φως, έτσι ώστε τα ρομπότ λειτουργούσαν με τη δικιά τους αντανάκλαση. Όμως καθώς το έκαναν αυτό, το φως έσβηνε, πράγμα που σήμαινε ότι το ερέθισμα αποκοπτόταν. Έτσι η έλλειψη του ερεθίσματος αποκαθιστούσε το φως, το οποίο και πάλι φαινόταν ως ερέθισμα, και ούτω καθεξής. Τα ρομπότ καθυστερούσαν ως εκ τούτου πριν από έναν καθρέφτη καθώς τρεμόπαιζαν. Με αυτό τον τρόπο αποδεικνύονταν ότι τα ρομπότ είναι ανώτερα από πολλά «έξυπνα» ζώα που συνήθως αντιμετωπίζουν την αντανάκλασή τους με προβληματισμό, σαν να βλέπουν ένα άλλο ζώο μπροστά τους.



Εικόνα 1.2.5 και 1.2.6: Αριστερά το Elmer και δεξιά το Elsie ρομπότ

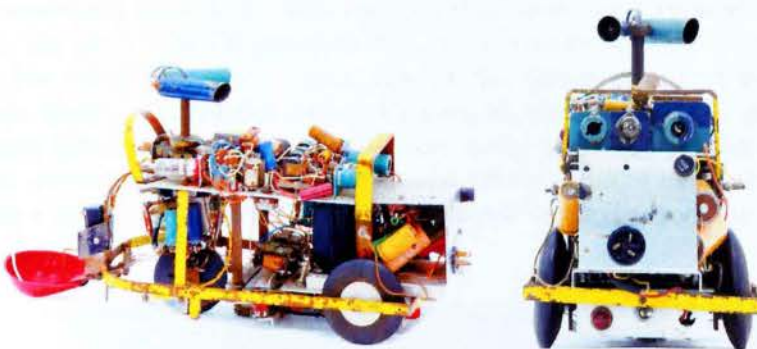
Στη συνέχεια μία από τις χελώνες τροποποιήθηκε (μάλλον η Elmer). Πιο συγκεκριμένα, της είχαν προσθέσει μια απλή και μόνο κυψέλη "εγκέφαλου" και δύο αντανακλαστικά κυκλώματα με τα οποία θα μπορούσε να διδαχθεί απλές συμπεριφορές. Αυτή η χελώνα ονομαζόταν CORA. Σκοπός του Cora ήταν να βρίσκει τρόφιμα και να περιπλανιέται γύρω από ένα σκαμνί που βρίσκονταν στη μέση του δαπέδου. Η εκπαίδευσή του αποτελούνταν από πολύ απλές εντολές. Με αυτές τις εντολές προσπαθούσαν να διδάξουν το ρομπότ ότι ο ήχος μιας σφυρίχτρας σήμαινε εμπόδιο. Έτσι το σφύριγμα αναγνωρίζονταν από το Cora ως ένα αντικείμενο που έπρεπε να αποφύγει. Η εκπαίδευση γινόταν με μια αστυνομική σφυρίχτρα.. Στην Αγγλία, μια αστυνομική σφυρίχτρα είχε δύο νότες που ακούγονται μαζί και κάνουν ένα ιδιαίτερα δυσάρεστο ήχο. Ο Walter προσπάθησε να του διδάξει, ως εκ τούτου, ότι η μία νότα της σφυρίχτρας σήμαινε εμπόδιο και ότι η άλλη νότα σήμαινε τρόφιμα. Για την επίτευξη αυτής της ενέργειας ο Walter προσπάθησε να δημιουργήσει αυτό το αντανακλαστικό έχοντας δύο συντονισμένα κυκλώματα, ένα από τα οποία σχετίζονταν με την ανταπόκριση στην όρεξη και το άλλο με την απόκριση στην αποφυγή. Η μία πλευρά της σφυρίχτρας φυσιόνταν πριν το ρομπότ άγγιζε ένα αντικείμενο έτσι

ώστε να μάθει να το αποφεύγει, ενώ η άλλη πλευρά της σφουρίχτρας φυσιόνταν πριν το ρομπότ διακρίνει φως.



Εικόνα 1.2.7 και 1.2.8: Αριστερά το Cora μπροστά από ένα καθρέπτη και δεξιά μπροστά σε διάφορα εμπόδια

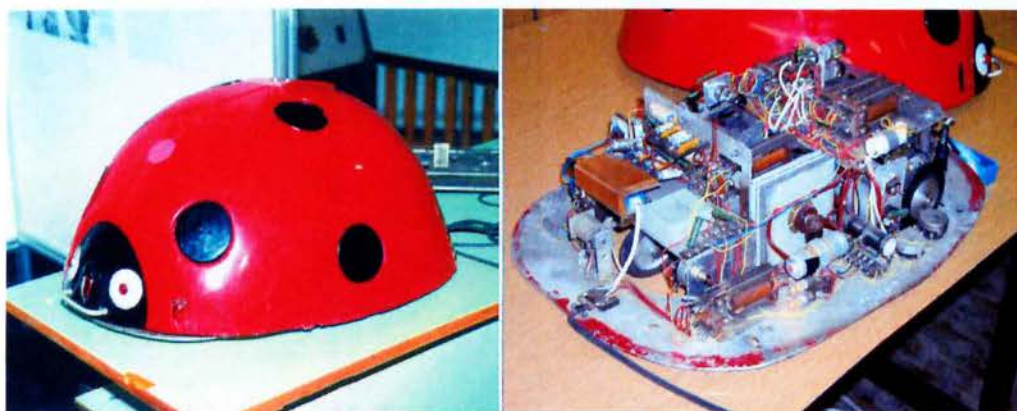
1951 - Squee electric squirrel: Σχεδιάστηκε και κατασκευάστηκε από τον Edmund C. Berkeley. Το Squee ήταν ένα μικρό ηλεκτρονικό ρομπότ σκίουρος που ήταν σε θέση να συλλέξει "καρύδια" (μπάλες τένις ή γκολφ). Ήταν το πρώτο επιδέξιο ρομπότ υπό αυτόματο έλεγχο και το πρώτο αληθινό ρομπότ. Περιείχε τέσσερα αισθητήρια όργανα (δύο φωτοσωλήνες και δύο διακόπτες επαφής), τρία όργανα δράσης (ένας κινητήρας, ένας κινητήρας διεύθυνσης και ένα μοτέρ που ανοίγει και κλείνει τη σέσουλα ή τα "χέρια"), και ένα μικρό εγκέφαλο αποτελούμενο από έξι ρελέ. Το «καρύδι» ήταν μια μπάλα του τένις που φωτιζόνταν από ένα μέλος του κοινού που κατείχε σταθερά ένα φακό πάνω από την μπάλα, δείχνοντας τη στο Squee. Στη συνέχεια, το Squee έπαιρνε την μπάλα στα "χέρια" του, σταματούσε να δίνει προσοχή στο σταθερό φως και επικεντρωνόταν στο να βρει τη θέση στην οποία αναβόσβηνε μια δεύτερη λάμπα, η οποία βρισκόταν πάνω από τη "φωλιά" του. Τότε έπαιρνε το "καρύδι" για την "φωλιά" του, το άφηνε εκεί και στη συνέχεια επέστρεφε στο κνήγι για περισσότερα "καρύδια".



Εικόνα 1.2.9 και 1.2.10: Πλευρική και μπροστινή όψη του Squee electric squirrel

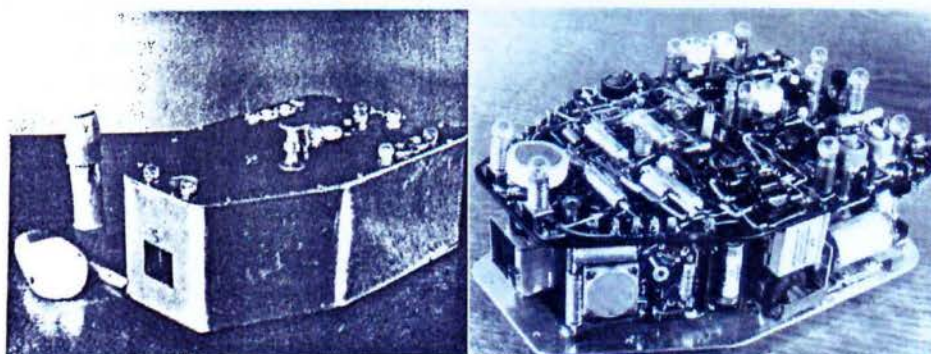
1957 - Katicabogár: Κατασκευάστηκε από τον Daniel Muszka και τον László Kalmár στο Πανεπιστήμιο Szeged της Ουγγαρίας. Το Katicabogár είναι ένα ρομποτικό ζώο το οποίο βρίσκεται στο Informatika Történeti Alapítvány Múzeum. Το Katicabogár

έχει 60 εκατοστά μήκος, 40 εκατοστά πλάτος και 25 εκατοστά ύψος. Έχει αισθητήρες φωτός στα μάτια του, ένα μικρόφωνο, επτά αισθητήρες αφής πάνω στο δέρμα του, μια χωρητική μνήμη και δύο ηλεκτροκινητήρες. Λόγω της δυσκολίας απόκτησης εξαρτημάτων το Katicabogár χρησιμοποιούσε ένα λώρο για την τροφοδοσία του (220V). Επίσης ήταν ικανό να ανταποκριθεί στον ήχο, στο φως και στην αφή. Αν μια πηγή φωτός για παράδειγμα φώτιζε στα μάτια του, αυτό κινούνταν προς την κατεύθυνση του φωτός και αν η πηγή φωτός μετακινούνταν την ακολουθούσε. Επίσης, αν το φως ήταν απενεργοποιημένο ή βρισκονταν πολύ μακριά, τότε το Katicabogár σταμάταγε τη λειτουργία του. Επιπρόσθετα όταν άκουγε ένα οποιοδήποτε ήχο π.χ. ένα μουσικό όργανο τότε αναβόσβηνε τα μάτια του υποδεικνύοντας ότι αντιλαμβάνεται τον ήχο. Στη συνέχεια αν τα ερεθίσματα του φωτός και του ήχου λαμβάνονταν ταυτόχρονα τότε η πασχαλίτσα αντιλαμβάνονταν μόνο τα ηχητικά ερεθίσματα και κινούνταν σε ευθεία γραμμή. Τέλος κάτω από τις κουκκίδες της πασχαλίτσας υπάρχουν διακόπτες. Όταν ένα από αυτούς πατιόνταν ή όταν η πασχαλίτσα άγγιζε ένα αντικείμενο το αποτέλεσμα ήταν να μην αντιδρά στο φως και στον ήχο. Αν όμως στη συνέχεια χαϊδεύαμε τις κουκκίδες αντιδρούσε και πάλι στα ερεθίσματα.



Εικόνα 1.2.11 και 1.2.12: Εξωτερική και εσωτερική εμφάνιση του Katicabogár

1959 - Zemanek: Ήταν ένα συμπαγές αυτόματο που κινείτο σε τρεις τροχούς εκ των οποίων οι κινητήριои ήταν οι δύο πίσω όπου περιλάμβαναν έναν ξεχωριστό κινητήρα ο καθένας. Με την 6V Ni-Cd μπαταρία του, τροφοδοτούσε το σύνολο του εξοπλισμού των δύο κινητήρων, των 19 ρελέ, των 14 τρανζίστορ και των 8 ενδεικτικών λυχνιών του. Επίσης το Zemanek διέθετε έξι κουμπιά του επέτρεπαν τη χειροκίνητη εγκατάσταση των διαφόρων καταστάσεών του. Κάθε φορά που μια δέσμη φωτός έφθανε στο φωτοτρανζίστορ υπήρχε μια ελαφρά καθυστερημένη μετάβαση από τον ένα κινητήρα στο άλλο με αποτέλεσμα το Zemanek να καταγράφει μια ορισειδές κίνηση.

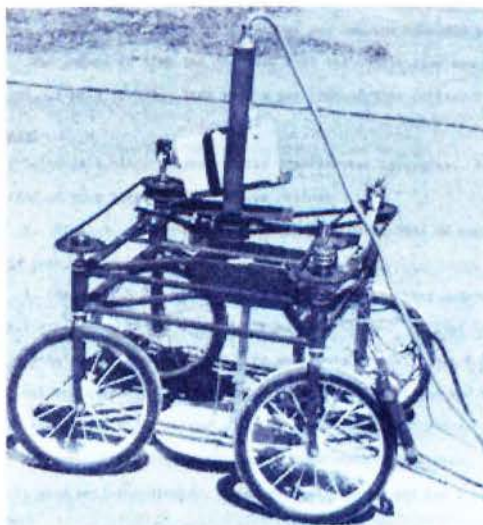


1960 - Beast: Το Johns Hopkins University ανέπτυξε το Beast. Το Beast ήταν ένα κινητό αυτόματο που μπορούσε να επιβιώσει σε ένα φυσικό περιβάλλον χωρίς καμία ανθρώπινη βοήθεια. Έτσι μπορούσε και περιπλανιόταν στις αίθουσες του εργαστηρίου και σε γραφεία καθώς χτυπούσε και τις πόρτες αυτών. Όταν η ενέργεια των μπαταριών του Beast έπεφταν σε χαμηλά επίπεδα, επιδίωκε από μόνο του ηλεκτρικές πρίζες, με ένα ειδικό οπτικό φωτοκύτταρο, όπου συνδεόταν μέσω ενός ειδικού βραχίονα και επαναφορτιζόνταν. Το Beast τροφοδοτούνταν έως ότου οι μπαταρίες φορτιζόνταν πλήρως και στη συνέχεια περιπλανιόταν ξανά μέχρι να έρθει η ώρα της επαναφόρτισης. Τέλος το Beast μπορούσε να αποφύγει εμπόδια, ανθρώπους ακόμα και σκαλιά μέσω των διαφόρων αισθητήρων του.



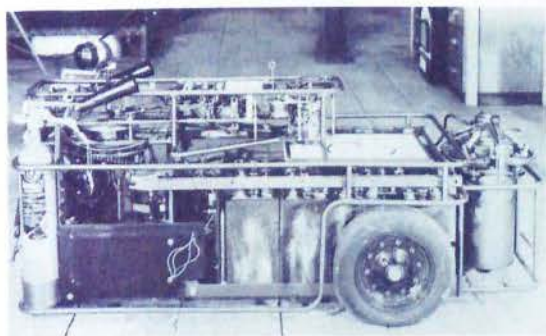
Εικόνα 1.2.15 και 1.2.16: Το Beast κατά την διαδικασία επαναφόρτισής του

1961 - Stanford Cart: Κατασκευάστηκε αρχικά από τον James L. Adams ως ερευνητικό όχημα για απομακρυσμένες αποστολές στο φεγγάρι. Το Stanford Cart 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. Δοκιμές διεξήχθησαν με το Cart να στρίβει τόσο με δύο ρόδες (μπροστινές), όπως ένα αυτοκίνητο, όσο και με τέσσερις, στο οποίο οι τροχοί και η τηλεοπτική κάμερα περιστρέφονταν μαζί. Το Stanford Cart συνδεόταν με ένα πολύ μακρύ καλώδιο σε μια κονσόλα ελέγχου, με οθόνη τηλεόρασης για τον έλεγχο της οδήγησης και της ταχύτητας. Ο Adams διερευνούσε τη δυνατότητα ελέγχου του οχήματος, αποφεύγοντας τα εμπόδια με διάφορους συνδυασμούς τριών συντελεστών: της καθυστέρησης, της επικοινωνίας και της ταχύτητας. Όταν οι εντολές διεύθυνσης καθυστερούσαν στην επικοινωνία, υπήρχε μια τάση για το χειριστή πάνω στο τιμόνι και έχανε τον έλεγχο. Μεταξύ άλλων, ο Adams έδειξε στη διατριβή του ότι με καθυστέρηση επικοινωνίας, που αντιστοιχεί στο ταξίδι μετ' επιστροφής στη Σελήνη (περίπου 2 1/2 δευτερόλεπτα), το όχημα δεν μπορεί να ελεγχθεί με αξιοπιστία εάν ταξιδεύει γρηγορότερα από περίπου 0,3 km/h.



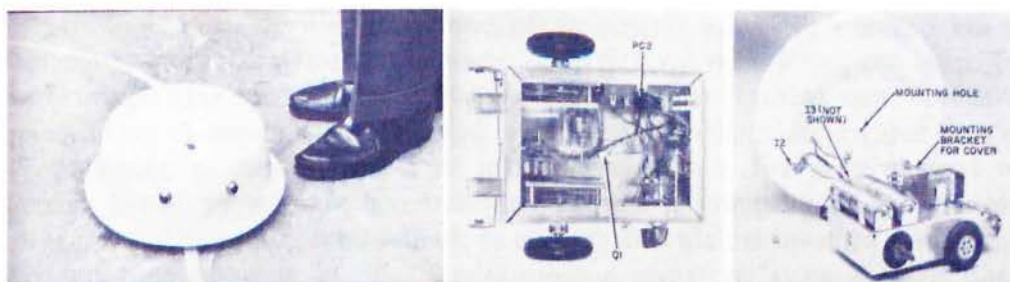
Εικόνα 1.2.17: To Stanford Cart

1962 – Robot Fire Cart: Σχεδιάστηκε και κατασκευάστηκε από τον Meredith Thring στο Queen Mary College. Ήταν ένα πυροσβεστικό ρομπότ που πλοηγούνταν ελεύθερα στο χώρο χρησιμοποιώντας σήματα από μια γυροσκοπική πυξίδα για τη μέτρηση της απόστασης. Όταν εντόπιζε μια εστία φωτιάς, μέσω του φωτοκύτταρου, πήγαινε κατευθείαν προς αυτήν και όταν ο διμεταλλικός διακόπτης ανίχνευε τη θερμότητα της φλόγας, το ρομπότ σταματούσε και τοποθετούσε το ακροφύσιο του πυροσβεστήρα πάνω στη φλόγα για να τη σβήσει. Η ιδέα ήταν να αναπτυχθεί ένα πλήρως αυτόματο ρομπότ "νυχτοφύλακας" που θα μπορούσε να περιπλανιέται γύρω από μια αποθήκη και να κοιτάζει έξω για πυρκαγιά.



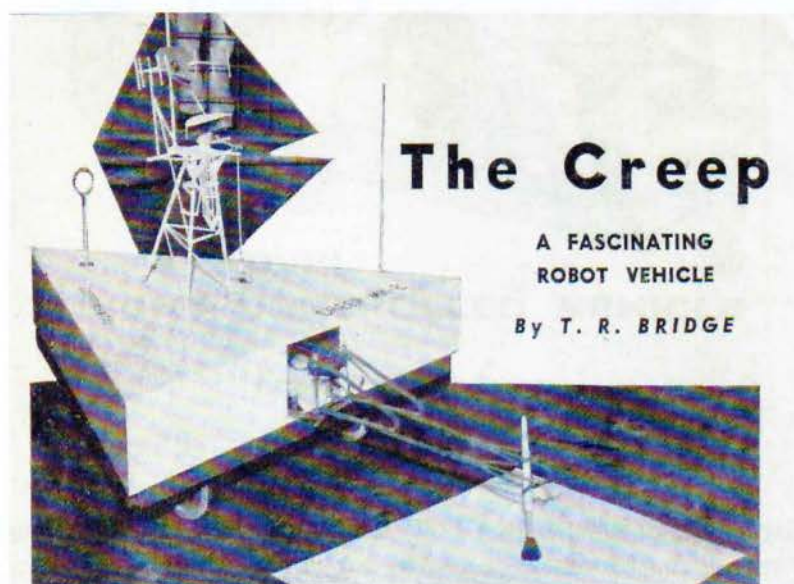
Εικόνα 1.2.18 και 1.2.19: Το Robot Fire Cart απενεργοποιημένο και σε δράση

1962 – Emily: Κατασκευάστηκε από τον Bernard Dickman. Ήταν ένα απλούστερου τύπου ηλεκτρονικό ρομπότ με «one track mind». Δηλαδή αν και ήταν εξοπλισμένο με ένα μόνο "όργανο αίσθησης" (ένα φωτοκύτταρο), με ένα ζεύγος κινητήρων και ένα στοιχειώδη "εγκέφαλο" (ένα τρανζίστορ και ρελέ), ήταν ικανό να επιδειξει μια εξαιρετικά ευφυή συμπεριφορά. Σκοπός του Emily ήταν να ακολουθεί πεισματικά μια λευκή γραμμή, ανεξάρτητα από το πόσες στροφές θα συναντούσε. Αν και ο βηματισμός του έδειχνε ότι ήταν 'υπό την επήρεια αλκοόλ' έφτανε πάντα στο τερματισμό. Τέλος το Emily μπορούσε να ακολουθήσει επ' άπειρον μια δέσμη φωτός (π.χ. φακός) αν την έστρεφες πάνω του.



Εικόνα 1.2.20 και 1.2.21: Αριστερά το Emily ακολουθεί μια άσπρη γραμμή και δεξιά το εσωτερικό του

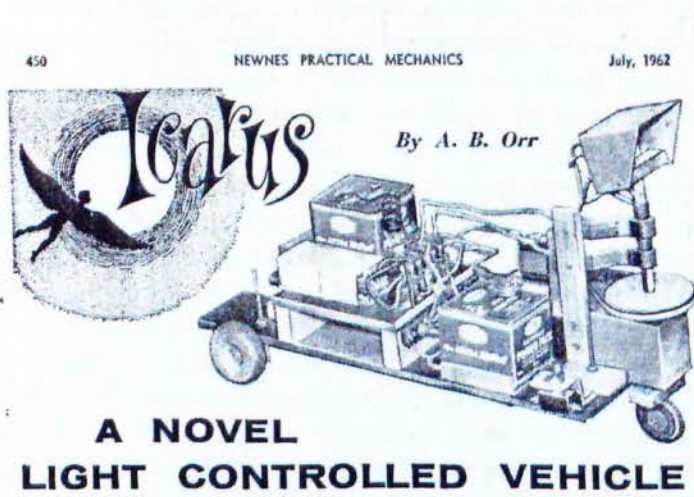
1962 – Creep: Ήταν ένα ασύρματα ελεγχόμενο και προγραμματιζόμενο ρομπότ με ένα βραχίονα και ένα γάντζο που μπορούσε να στρίψει σε γωνία 360 μοιρών. Σχεδιάστηκε από τον T.R. Bridge. Ένα από τα πιο ενδιαφέροντα πράγματα είναι η δυνατότητα εγγραφής ενός προγράμματος και στη συνέχεια η αποστολή του, μέσω του προγράμματος, χωρίς χειροκίνητο έλεγχο. Το σύστημα κατέγραφε τους τόνους σε ένα μαγνητόφωνο και στη συνέχεια αναπαρήγαγε τους τόνους από το μαγνητόφωνο πίσω στον πομπό. Για το σκοπό αυτό ο πομπός ήταν εφοδιασμένος με μία υποδοχή βύσματος, η οποία τροφοδοτούσε την αρπάγη.



Εικόνα 1.2.22: Το Creep

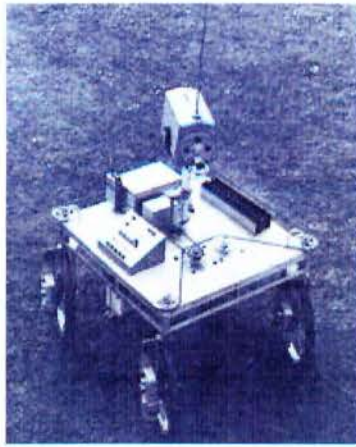
1962 – Icarus: Σχεδιάστηκε από τον A. B. Orr στο Newnes Practical Mechanics τον Ιούλιο του 1962. Μετά από τις μηχανές του Grey Walter αυτό ήταν ίσως το πρώτο δημοσιευμένο σχέδιο για ένα όχημα με φωτοτροπικό πυργίσκο σάρωσης. Σε αυτό το πρωτότυπο μια μονάδα ρελέ χρησιμοποιούνταν για να στρέψει τα πλευρικά φώτα του κατά το σούρουπο. Το Icarus χρησιμοποιούσε το ρελέ για την ενεργοποίηση και την απενεργοποίηση ενός μικρού ηλεκτρικού κινητήρα. Το ρομπότ περιελάμβανε μία μονάδα σάρωσης τοποθετημένη στο ίδιο άξονα με το μπροστινό τιμόνι, ένα απλό τρανζίστορ ενίσχυσης, δύο ρελέ και μπαταρίες. Ένας μικρός ηλεκτροκινητήρας κινούσε τον σαρωτή και το τιμόνι κατά 360 μοίρες. Ένας δεύτερος κινητήρας κινούσε τους πίσω τροχούς μέσω ενός μειωτήρα. Παρακάτω περιγράφεται η λειτουργία του

Icarus. Ξεκινώντας, όταν ενεργοποιούνταν, το ρομπότ παρέμενε σταθερό και ο μπροστινός τροχός με τη μονάδα σάρωσης έψαχναν αργά για ένα φως στο σπίτι. Τα φωτοκύτταρα ήταν εφοδιασμένα με ένα κάλυμμα που τα καθιστούσε κατευθυνόμενα εμποδίζοντας το τυχαίο φως. Μία πηγή φωτός κατόπιν, όπως ένας ισχυρός φακός, ενεργοποιούσε το σαρωτή που επικεντρωνόταν πάνω του. Για να αντιληφτεί τη φωτεινή δέσμη, τα φωτοκύτταρα πατούσαν το ρελέ με συνέπεια το άνοιγμα του κυκλώματος του σαρωτή, προκαλώντας το σαρωτή να εντοπίσει την πηγή φωτός. Την ίδια ώρα όταν πατιόταν το ρελέ λειτουργούσε ο κινητήρας κίνησης στους πίσω τροχούς με αποτέλεσμα το ρομπότ να κινείται προς τα εμπρός σε οποιαδήποτε κατεύθυνση ο σαρωτής όριζε. Το Icarus συνέχιζε να κινείται προς τα εμπρός μέχρι να βρισκόταν εκτός της δέσμης φωτός και η αλληλουχία αναστρεφόταν. Ο σαρωτής όταν έχανε την πηγή φωτός ξανάρχιζε τη σάρωση και το κύκλωμα του κινητήρα άνοιγε και προκαλούσε το ρομπότ να σταματήσει να κινείται προς τα εμπρός. Για την επόμενη σάρωση του, λάμβανε για μια ακόμη φορά το φως και η αλληλουχία επαναλαμβάνονταν. Αυτή η αλληλουχία της λειτουργίας συνεχίζονταν μέχρις ότου ο σαρωτής αντιλαμβανόταν ότι σταματούσε η φωτεινή πηγή μπροστά του. Όταν συνέβαινε αυτό, το ρομπότ κινούνταν άμεσα προς την κατεύθυνση αυτή.



Εικόνα 1.2.23: To Icarus

1963 – Stanford Cart: Ο φοιτητής Paul W. Braisted επινόησε ένα σχέδιο για τη βελτίωση της δυνατότητας ελέγχου του οχήματος Stanford Cart, με την προσθήκη ενός αναλογικού υπολογιστή που λειτουργούσε ως προγνωστικός παράγοντας, δηλαδή λάμβανε υπόψη προηγούμενες εντολές του συστήματος διεύθυνσης. Στη συνέχεια έθετε μια φωτεινή κουκίδα στην οθόνη της τηλεόρασης με τη προβλεπόμενη θέση του ρομπότ, όταν μια τρέχουσα εντολή διεύθυνσης θα άρχιζε να παράγει αποτελέσματα. Με την προσθήκη αυτή το όχημα θα μπορούσε να ελέγχεται στα 8 km/h. Ακόμα υπήρχε ένας θεμελιώδης περιορισμός για τον τηλεχειρισμό που έγκειται στο ότι αν το ταξίδι, κατά τη χρονική καθυστέρηση, ήταν μεγαλύτερο από ότι η απόσταση του οχήματος από ένα αόρατο εμπόδιο, τότε δεν υπήρχε τρόπος για να αποφευχθεί το χτύπημα αυτό.



Εικόνα 1.2.24: Το Stanford Cart σε εξωτερικό περιβάλλον

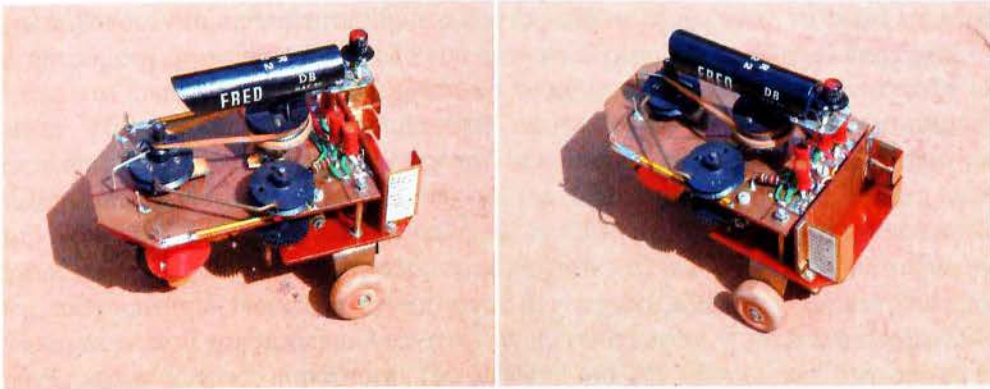
1964 –1971 – Stanford Cart: Μέχρι το 1966 το Stanford Cart ήταν αχρησιμοποίητο σε ένα εργαστήριο, όταν ο Les Earnest, ένας επιστήμονας που είχε δώσει πρόσφατα στο Stanford Cart τεχνητή νοημοσύνη (SAIL), βρήκε και μίλησε με το δημιουργό του Stanford Cart, James Adams, αφήνοντας το SAIL να χρησιμοποιήσει και να δοκιμάσει την περιήγηση στο δρόμο υπό τον έλεγχο του υπολογιστή χρησιμοποιώντας αναφορές οπτικής επαφής. Ωστόσο, οι συνδέσεις ραδιοφώνου και άλλα ηλεκτρονικά που υπήρχαν νωρίτερα είχαν εξαφανιστεί, οπότε προσελήφθη ο διδακτορικός φοιτητής Ηλεκτρολόγων Μηχανικών Rodney Schmidt για να κατασκευάσει ένα χαμηλής ενέργειας πομπό τηλεόρασης και ένα έλεγχο ασύρματης επικοινωνίας για να αναλάβει την οπτική καθοδήγηση του έργου. Η πειραματική λειτουργία άρχισε με ένα χειριστή να ελέγχει το Cart μέσω του υπολογιστή με βάση τις εικόνες της τηλεόρασης που έβλεπε. Χρησιμοποιώντας το KA10 επεξεργαστή, ο Schmidt ήταν τελικά σε θέση να πάρει το Cart για να ακολουθήσει, αυτόματα, μια υψηλής αντίθεσης λευκή γραμμή, κάτω από ελεγχόμενες συνθήκες φωτισμού, με ταχύτητα περίπου 1,3 χιλιόμετρα/ώρα. Ο Schmidt ολοκλήρωσε τη διατριβή του το 1971.



Εικόνα 1.2.25: Το Stanford Cart εφοδιασμένο τεχνητή νοημοσύνη

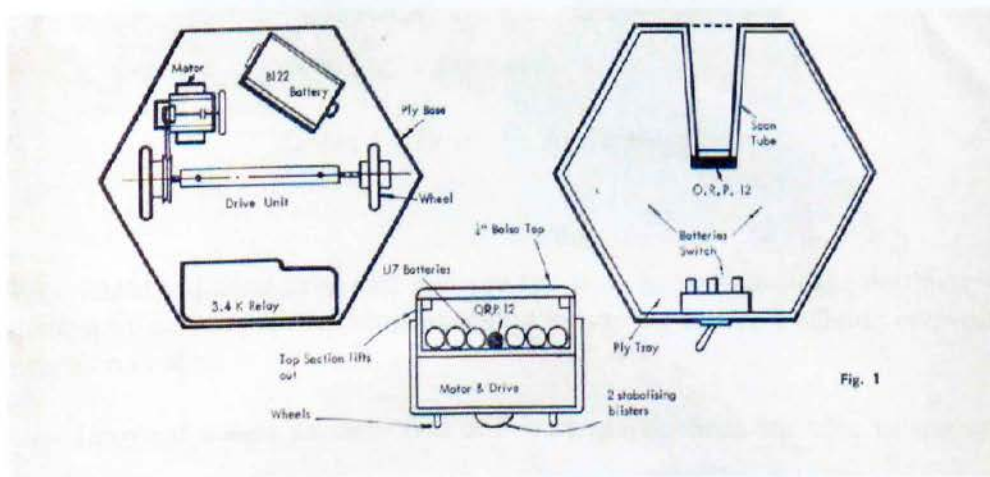
1964 – Fred: Σχεδιάστηκε και κατασκευάστηκε από τον Peter Holland. Πιθανώς το πρώτο δημοσιευμένο σχέδιο για ένα φωτοτροπικό όχημα όπου δεν ακολούθησε

σχεδιαστικά τον Grey Walters, δηλαδή στο να σχεδιάσει ένα πύργο σάρωσης συνδεδεμένο με το μπροστινό τροχό. Αντ' αυτού ο πυργίσκος του Fred περιστρέφονταν ανεξάρτητα από τους τροχούς και ένας αισθητήρας ενσωμάτωνε το επίπεδο φωτός πάνω από τον αριστερό και δεξιό τομέα και κατεύθυνε το τιμόνι προς το φωτεινότερο τομέα.



Εικόνα 1.2.26 και 1.2.27: Δύο όψεις του φωτοτροπικού Fred

1965 – Hexy: Ήταν ένα όχημα αναζήτησης φωτός, σχεδιασμένο από τον G. Draper και δημοσιεύτηκε στο Radio Control Models & Electronics το Μάρτιο του 1965. Το Hexy, αυτή η απλή συσκευή αναζήτησης φωτός, λειτουργούσε από ένα κύκλωμα ηλεκτρονόμων και η λειτουργία σάρωσης του άρχιζε με την αντιστροφή της φοράς κίνησης του κινητήρα μετάδοσης. Ο σαρωτής ήταν συζευγμένος με τους κινητήριους τροχούς, στο κύριο σώμα, με ένα σύστημα από καστανίες και ελέγχονταν από έναν ηλεκτρονόμο που βρίσκονταν στο κύκλωμα του φωτοκυττάρου. Αυτό επιτεύχθηκε πολύ απλά χρησιμοποιώντας ένα μόνο ισχυρό κινητήρα για την μπροστινή ώθηση και σάρωση. Σκοπός του Hexy ήταν η αναζήτηση φωτός. Όταν δεν υπήρχε φως να προσπίπτει στο φωτοκύτταρο, ο σπλισμός του ρελέ ακουμπούσε στην πίσω επαφή και το ρομπότ άρχιζε να σαρώνει. Όταν διακρινόταν φως τότε ο ηλεκτρονόμος αντιστρέφει την λειτουργία του ηλεκτρικού κινητήρα και το ρομπότ κινούνταν ευθεία προς το φως.



Εικόνα 1.2.28: Σχέδιο του οχήματος Hexy

1966-1972 – Shakey: Το Ερευνητικό Κέντρο του Stanford κατασκεύασε και πραγματοποίησε έρευνα πάνω σε ένα ρομπότ που ονομάστηκε Shakey, εξαιτίας της σπασμοδικής του κίνησης. Το Shakey ήταν εξοπλισμένο με κάμερα, αισθητήρα μέτρησης της απόστασης, αισθητήρες επαφής καθώς επίσης και κεραία για ασύρματη επικοινωνία. Το Shakey ήταν το πρώτο ρομπότ που επιχειρηματολογούσε για τις πράξεις του. Αυτό σημαίνει ότι δίνονταν στο ρομπότ πολύ γενικές εντολές, και εκείνο αντιλαμβανόταν τα απαραίτητα βήματα που έπρεπε να κάνει, ώστε να φέρει εις πέρας την ζητούμενη εργασία. Η αποστολή του ήταν να ανοίγει και να κλείνει διακόπτες και πόρτες, και κυρίως να ξέρει πότε πρέπει να το κάνει αυτό και γιατί! Η ανάπτυξη του Shakey οδήγησε σε αρκετά σημαντικά αποτελέσματα που έχουν εκτεταμένες επιπτώσεις στη ρομποτική και την τεχνητή νοημοσύνη, καθώς και την επιστήμη των υπολογιστών γενικότερα. Μερικά από τα πιο αξιοσημείωτα αποτελέσματα, περιλαμβάνουν την ανάπτυξη του αλγόριθμου αναζήτησης, ο οποίος χρησιμοποιείται ευρέως σε pathfinding και σε γραφήματα διάσχισης, τη διαδικασία της αποτύπωσης, το μετασχηματισμό Hough, ο οποίος είναι μια τεχνική εξαγωγής χαρακτηριστικών γνωρισμάτων που χρησιμοποιούνται στην ανάλυση εικόνων, η ψηφιακή επεξεργασία εικόνας και η γραφική παράσταση της μεθόδου για την εύρεση των συντομότερων μονοπατιών.

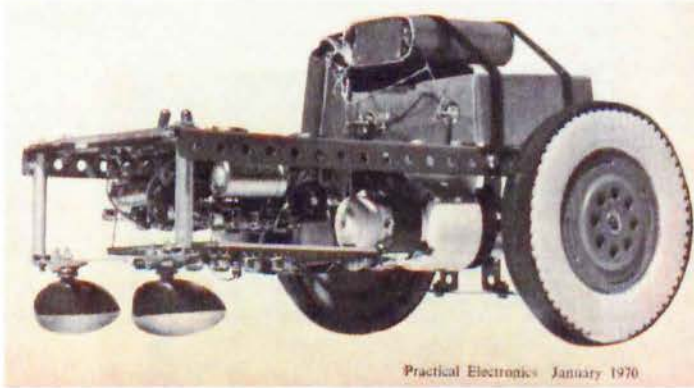


Εικόνα 1.2.29 και 1.2.30: Το Shakey

1969 – EMMA: Κατασκευάστηκε από τον GC Brown. Υπό κανονικές συνθήκες το ρομπότ ήταν σε θέση να εκτελέσει οποιαδήποτε από τις τέσσερις πιθανές ενέργειες. Αυτές είναι οι εξής:

- Προχωρά ευθεία μπροστά (και οι δύο κινητήρες κινούνται προς τα εμπρός).
- Προχωρά προς τα δεξιά (ένας κινητήρας σταματά (ο δεξιά) και ο άλλος κινείται προς τα εμπρός).
- Προχωρά προς τα αριστερά (ένας κινητήρας σταματά (ο αριστερός) και ο άλλος κινείται προς τα εμπρός).
- Σταματά (και οι δύο κινητήρες σταματούν).

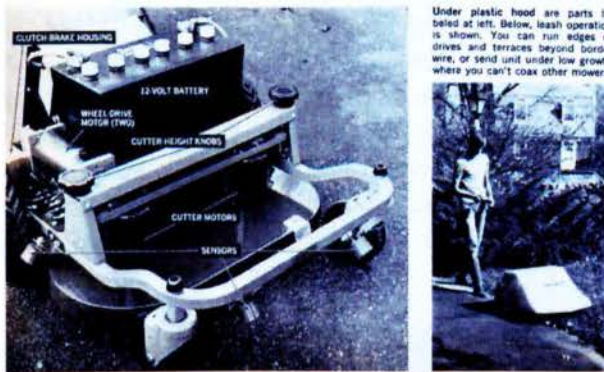
Η αντίστροφη του φωτός ήταν διπλή. Δύο εντελώς ξεχωριστά κανάλια υπήρχαν αριστερά και δεξιά του ρομπότ. Κάθε κανάλι αντλούσε την είσοδο του από ένα φωτο-αισθητήρα και, μετά από κατάλληλη ενίσχυση, είχε άμεσο έλεγχο ως προς τον τρόπο λειτουργίας του αντίθετου καναλιού. Ως εκ τούτου, εάν ένα ισχυρό φως διέγειρε έναν από τους αισθητήρες, το Emma έκανε μια αντανάκλαστική κίνηση μακριά από την πηγή φωτισμού. Επιπλέον το Emma εφοδιαζόταν με ένα σύστημα αίσθησης φορτίου. Αυτό επιτυγχάνονταν με την παρακολούθηση του ρεύματος που αντλούσαν τα μοτέρ κίνησης. Εάν κάποιο προκαθορισμένο επίπεδο ξεπερνούσαν, όπως για παράδειγμα όταν το ρομπότ συναντούσε ένα εμπόδιο, ένα κύκλωμα κατωφλίου θα προκαλούσε φωτιά. Έτσι η έξοδος από αυτό το κύκλωμα τροφοδοτούσε ένα κύκλωμα αποφυγής εμποδίων.



Practical Electronics January 1970

Εικόνα 1.2.31: Το Emma

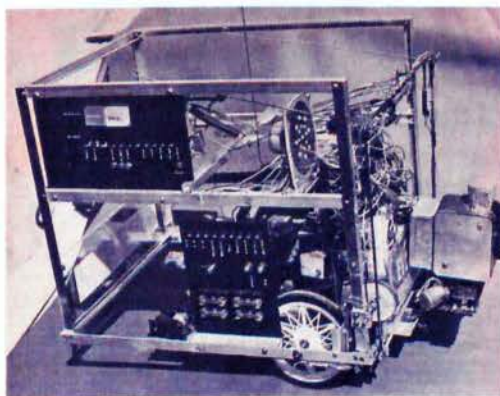
1969 : Το Mowbot ήταν το πρώτο ρομπότ το οποίο κούρευε αυτόματα το γρασίδι.



Εικόνα 1.2.32: Το Mowbot

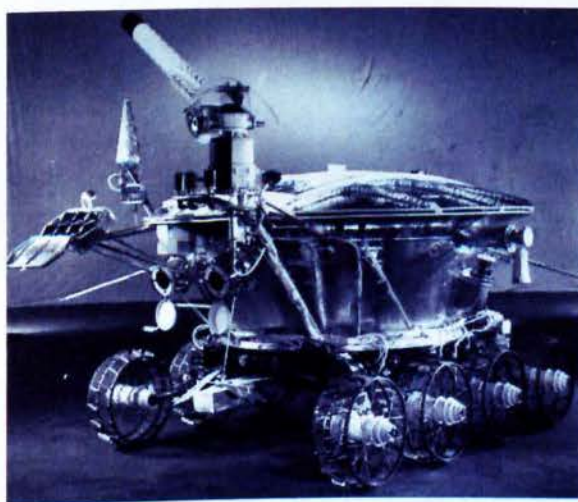
1970 - Merv: Τα αρχικά MERV σημαίνουν Mobile Environmental Response Vehicle. Ο δημιουργός του, Peter Vogel, το κατασκεύασε για να αποδείξει τη θεωρία του περί της τεχνητής νοημοσύνης. «Είναι αλήθεια ότι οι μηχανές είναι ανίκανες για νοημοσύνη, ή μπορούν στην πραγματικότητα να διαθέτουν λογική; » Το MERV είχε την ικανότητα να αυτοσυντηρείται. Περιπλανιόνταν μέχρι να συναντήσει άγνωστο αντικείμενο. Στη συνέχεια πραγματοποιούσε διάφορες δοκιμές σε αυτό το αντικείμενο με σκοπό να ανακάλυψη αν αυτό είναι εχθρικό ή όχι. Αν το αντικείμενο χαρακτηριζόταν ως εχθρικό τότε το MERV απομακρύνονταν. Ο τρόπος που το έκανε ήταν ο εξής:

Αρχικά σταματούσε σε ασφαλή απόσταση από το αντικείμενο. Στη συνέχεια έκανε δοκιμές για να καταλάβει τι είδους αντικείμενο είναι. Για να το επιτύχει αυτό αναζητούσε από την μνήμη του αν αυτό του είχε προκαλέσει "πόνο" παλαιότερα. Τέλος βάση όλων αυτών εφάρμοζε την αντίστοιχη δράση. Για την καθιέρωση των αντικειμένων, δηλαδή για το αν είναι "ασφαλές" ή "εχθρικά" απαιτούνταν ένας μεγάλος αριθμός αισθητήρων που ήταν οι εξής: ένας οπτικός αισθητήρας που εντόπιζε τα αντικείμενα και σταματούσε τρία με τέσσερα εκατοστά μακριά τους, ένας ανιχνευτής φωτεινότητας, ένας αισθητήρας διακρίσεως ηχητικών τόνων για να διαπιστωθεί αν τα αντικείμενα εκπέμπαν «εχθρικούς» ήχους, αισθητήρες θερμότητας, αισθητήρες για τον καθορισμό σκληρότητας των αντικειμένων, αισθητήρες για τον προσδιορισμό των αντικειμένων (αν είναι μεταλλικό ή μη-μεταλλικό) και συσκευές για τον προσδιορισμό του μεγέθους, όσφρησης, και γεύσης.



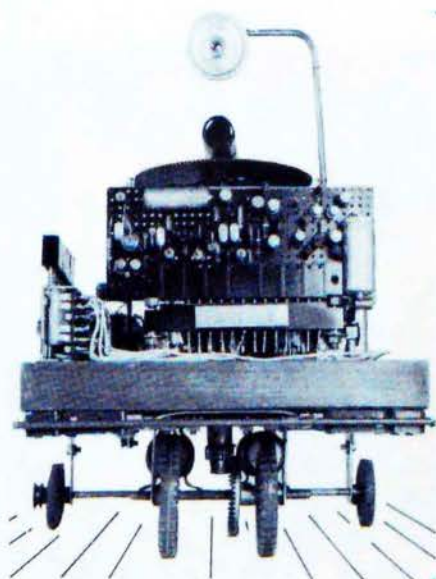
Εικόνα 1.2.33: To Merv

1970 - Lunokhod 1: Η Σοβιετική Ένωση εξερεύνησε την επιφάνεια της Σελήνης, με το Lunokhod 1, ένα σεληνιακό ρόβερ. Το Lunokhod 1 ήταν το πρώτο τηλεκατευθυνόμενο ρομπότ που προσγειώθηκε σε ένα άλλο ουράνιο σώμα. Το διαστημικό σκάφος προσγειώθηκε μαλακά στην επιφάνεια της Σελήνης στη Θάλασσα των Βροχών στις 17 Νοεμβρίου. Το ρόβερ λειτουργούσε κατά τη σεληνιακή ημέρα, σταματώντας περιστασιακά για να επαναφορτίσει τις μπαταρίες του μέσω των ηλιακών συλλεκτών. Τη νύχτα το Lunokhod 1 έκανε αδρανοποίηση μέχρι την επόμενη ανατολή του ηλίου.



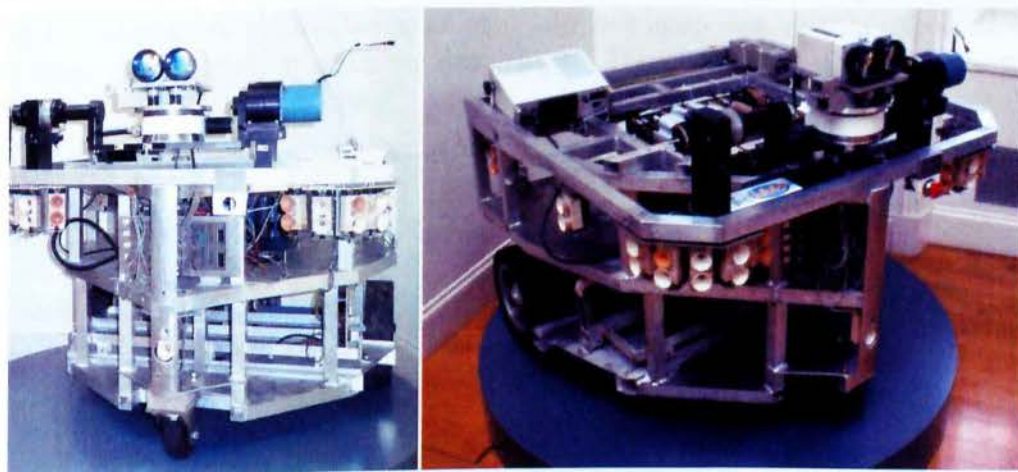
Εικόνα 1.2.34: Το Lunokhod 1

1972 - Cyclops: Ήταν ένα όχημα αναζήτησης φωτός, σχεδιασμένο από τον LC Galitz. Ένα σχέδιο που δημοσιεύθηκε στο The Radio Constructor. Ο Κύκλωπας ήταν ένα από τα τελευταία έργα κατασκευής για έναν πλήρως λειτουργικό κυβερνητικό μοντέλο. Ο Galitz επηρεάστηκε σε μεγάλο βαθμό από τις χελώνες του Grey Walter.



Εικόνα 1.2.35: Μπροστινή όψη του Cyclops

1977 - Hillare: Είναι τα αρχικά από το "Heuristiques Intégrées aux Logiciels et aux Automatismes dans un Robot Évolutif". Σχεδιασμένο το 1977 στο Laboratoire d'Automatique et d'Analyse des Systèmes στη Toulouse, (LAAS), ήταν το πρώτο Γαλλικό αυτόνομο κινητό ρομπότ, εξοπλισμένο με αισθητήρες υπερήχων, 16 αισθητήρες US, κάμερα και τηλεμετρικό laser. Για την επικοινωνία χρησιμοποιούσε ένα σειριακό ράδιο-modem (9600 bauds). Ακόμα περιελάμβανε δύο κινητήριους τροχούς και ένα ελεύθερο, μπαταρίες 24V και τέσσερις επεξεργαστές Intel 80286.



Εικόνα 1.2.36 και 1.2.37: Το κινητό ρομπότ Hillare

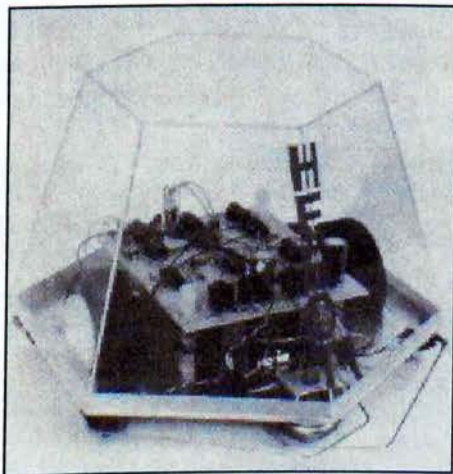
1979 - Stanford Cart: Σε εξέλιξη από το 1967, το Stanford Cart πέρασε με επιτυχία ένα δωμάτιο γεμάτο καρέκλες χωρίς ανθρώπινη παρέμβαση. Ο Hans Moravec ξανακατασκεύασε το Stanford Cart το 1977, εξοπλίζοντάς το με στερεοσκοπική

όραση. Μια τηλεοπτική κάμερα, τοποθετημένη σε μια ράγα στη κορυφή του αμαξιδίου, έπαιρνε εικόνες από πολλές διαφορετικές οπτικές γωνίες και τις αναμετάδιδε σε έναν υπολογιστή. Ο υπολογιστής υπολόγιζε την απόσταση μεταξύ του αμαξιδίου και των εμποδίων στο δρόμο του καθώς χαρτογραφούσε το περιβάλλον του.



Εικόνα 1.2.38: Το εξοπλισμένο Stanford Cart με στερεοσκοπική όραση

1979-80 - HEBOT I: Κατασκευάστηκε από τον John FitzGerald. Ήταν ένα ελεύθερο περιπλανώμενο ρομπότ, που απέφευγε εμπόδια, κατευθυνόταν προς το φως και ακολουθούσε ένα καλώδιο (με εναλλασσόμενο ρεύμα) γύρω από ένα σπίτι. Το HEBOT έκπεμπε έναν ήχο, όταν ανίχνευε φως ή μετά από μια σύγκρουση. Επίσης είχε ένα μικρόφωνο και αναλόγως με το τι λάμβανε μπορούσε να πάει μπρός, πίσω, να κάνει περιστροφή, ή απλά να σταματήσει. Τέλος είχε ένα φορτιστή για την επαναφόρτιση στη «φωλιά» του.



Εικόνα 1.2.39: Το Hebot I

1980 : Το ενδιαφέρον του κοινού γύρω από τα ρομπότ αυξάνεται, και αυτό έχει ως αποτέλεσμα την σταδιακή παραγωγή ρομπότ τα οποία μπορούν να αγοραστούν για οικιακή χρήση. Τα ρομπότ αυτά εξυπηρετούσαν ψυχαγωγικούς ή εκπαιδευτικούς σκοπούς. Παραδείγματα αποτελούν το ρομπότ RB5X και η σειρά HERO (Heathkit Educational RObot).



Εικόνα 1.2.40: Το RB5X

1980 : Η ομάδα του Ernst Dickmanns κατασκεύασε στο Bundeswehr University του Μονάχου, τα πρώτα ρομπότ – αυτοκίνητα, τα οποία είχαν τη δυνατότητα να κινηθούν έως και 90 km/h σε άδειους δρόμους. Η ομάδα του διέθετε ένα Mercedes-Benz με κάμερες και άλλους αισθητήρες. Το αυτοκίνητο ήταν δυνατόν να έχει τον έλεγχο του τιμονιού, του γκαζιού και των φρένων του, μέσω ενός υπολογιστή, με βάση τις εντολές αξιολόγησης που λάμβανε σε πραγματικό χρόνο ανάλογα με τις εκάστοτε ακολουθίες εικόνων. Για λόγους ασφαλείας, αρχικά τα πειράματα πραγματοποιήθηκαν σε δρόμους χωρίς αυτοκίνητα. Από το 1986 τα αυτοκίνητα ρομπότ "VaMoRs" κατάφεραν να οδηγήσουν από μόνα τους, με ταχύτητες έως 96 km/h. Μία από τις μεγαλύτερες προκλήσεις της αυτόνομης οδήγησης με υψηλή ταχύτητα πρόκυπτε μέσα από τις ταχέως μεταβαλλόμενες οπτικές σκηνές του δρόμου. Τότε, οι υπολογιστές ήταν πολύ πιο αργοί από ό,τι είναι σήμερα. Ως εκ τούτου, η εξελιγμένη όραση ήταν αναγκαία για να αντιδράσουν τα αυτοκίνητα σε πραγματικό χρόνο. Η ομάδα του Dickmanns έλυσε το πρόβλημα μέσα από μια καινοτόμο προσέγγιση για την δυναμική όραση. Με τις κινήσεις της πλατφόρμας που μετέφεραν τις κάμερες του συστήματος, επιτρεπόταν να εστιάσουν την προσοχή τους (κάμερες) στα σημαντικότερα στοιχεία της οπτικής εισόδου. Τέλος τα φίλτρα Kalman είχαν επεκταθεί και χρησιμοποιήθηκαν για να επιτευχθεί ισχυρή αυτόνομη οδήγηση ακόμα και σε παρουσία θορύβου και αβεβαιότητας.



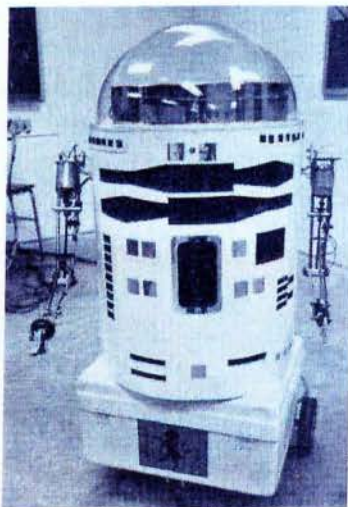
Εικόνα 1.2.41: Το εσωτερικό μιας αυτοκινούμενη Mercedes-Benz

1980 - Timbug 2: Ίσως ο πρώτος σχεδιασμός που δημοσιεύτηκε για ένα ρομποτικό όχημα που χρησιμοποιούσε αισθητήρες υπερήχων για την ανίχνευση εμποδίων. Χρησιμοποιούσε μια δέσμη υπερήχων για την ανίχνευση εμποδίων, αποφεύγοντας τα κάνοντας τυχαία όπισθεν και αλλάζοντας κατεύθυνση. Οι κινητήρες ήταν από ένα μηχανοκίνητο Tank kit. Η ταχύτητα του Timbug-2 δεν ήταν υψηλή, αλλά δεδομένου το ότι οι τροχοί του ήταν περίπου το διπλάσιο της διαμέτρου των γραναζιών κίνησης, το έκανε περίπου δύο φορές πιο γρήγορο από ότι θα ήταν το Tank.



Εικόνα 1.2.42: Μπροστινή όψη του Timbug 2

1980 – Unicorn: Ήταν ένα ρομπότ που είχε βγει στη γραμμή παραγωγής και που μπορούσε να το αγοράσει και να το φτιάξει ο καθένας. Ήταν πλήρως κινητό και είχε τη δυνατότητα να χρησιμοποιεί τους βραχίονες και τα χέρια του. Μπορούσε να ελέγχεται, από ένα σύνδεσμο καλωδίων, από μια κονσόλα ή και σε συνδυασμό με έναν υπολογιστή.



Εικόνα 1.2.43: Το ρομπότ Unicorn

1980-82 - Robart I: Κατασκευάστηκε από τον HR Everett. Του ανατέθηκε η λειτουργία να περιπολεί ένα σπίτι, από δωμάτιο σε δωμάτιο, για τον έλεγχο των ανεπιθύμητων καταστάσεων όπως πυρκαγιά, καπνός, εισβολή, τοξικά αέρια,

πλημμύρες, σεισμούς κλπ. Το ρομπότ περιελάμβανε οπτική ανίχνευση κίνησης, υπέρηχους ανίχνευσης κίνησης και ακοής. Το λογισμικό που περιελάμβανε το Robart I το διατηρούσε σε έναν από τους δύο τρόπους λειτουργίας: τη λειτουργία ειδοποίησης ή την παθητική λειτουργία. Στην παθητική λειτουργία, η πλειοψηφία των αισθητήρων ήταν ενεργοποιημένοι. Μερικές από αυτές τις εισόδους ήταν συνδεδεμένες να προκαλούν μια ειδοποίηση (μετάβαση από την παθητική λειτουργία στη λειτουργία ειδοποίησης), ενώ άλλες θα έπρεπε να αξιολογηθούν πρώτα από το λογισμικό που θα αποφάσιζε αν θα προκαλούσε μια ειδοποίηση στη συνέχεια, εφόσον απαιτούνταν. Και οι δύο λειτουργίες θα μπορούσαν να βρίσκονται σε ισχύ κατά την επαναφόρτιση. Η επαναφόρτιση βέβαια θα μπορούσε να ανασταλεί προσωρινά εάν οι συνθήκες κρίνονταν απαραίτητες. Η παροχή γινόταν με τον εντοπισμό και τη σύνδεση σε ένα επί δαπέδου σταθμό επαναφόρτισης, όταν η τάση της μπαταρίας άρχιζε να εξαντλείται. Οι περιπολίες γίνονταν σε τυχαία χρονικά διαστήματα, με το μεγαλύτερο μέρος του χρόνου που δαπανούνταν να παρέμενε ακίνητο, σε ένα παθητικό τρόπο εντοπισμού εισβολέων, για εξοικονόμηση ενέργειας.



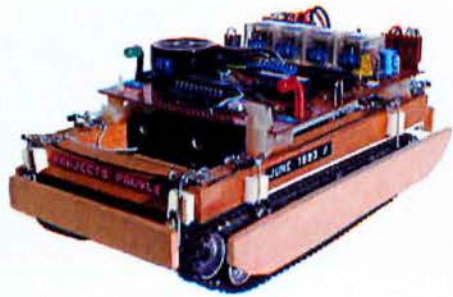
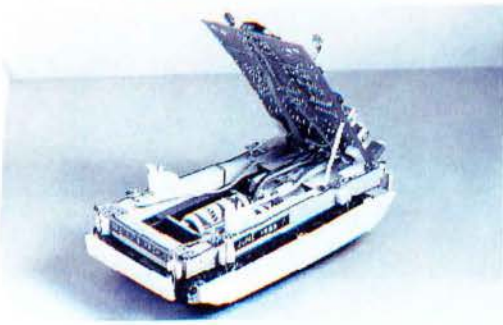
Εικόνα 1.2.44: Το Robart I

1982 – Zeaker: Ήταν ένα μικρό ρομποτικό όχημα που ελέγχονταν από έναν υπολογιστή και συνδεόταν με αυτόν μέσω ενός καλωδίου. Το Zeaker ήταν το πρώτο Βρετανικό μικρό-ρομποτικό όχημα, ελεγχόμενο από υπολογιστή με αισθητήρες αναγνώρισης χτυπημάτων.



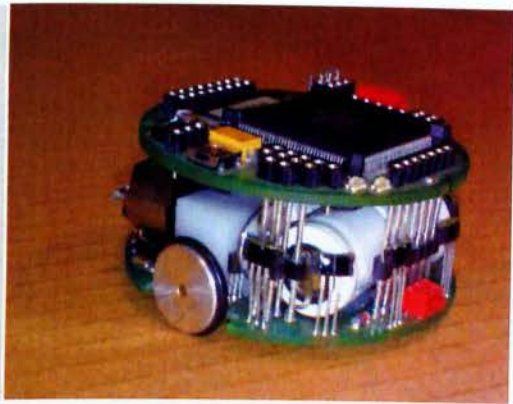
Εικόνα 1.2.45: Το Zeaker

1983 - Prowler: Το πρώτο σχέδιο που δημοσιεύθηκε στη Βρετανία για ένα επεκτάσιμο μικρό όχημα ρομπότ, που ελέγχονταν μέσω ενός καλωδίου από έναν κεντρικό υπολογιστή και είχε αισθητήρες κρούσης για την αποφυγή εμποδίων.



Εικόνα 1.2.46 και 1.2.47: Αριστερά το Prowler με ανοιχτό καπάκι και δεξιά με κλειστό

1991- Khepera: Ο Edo. Franzi, ο André Guignard και ο Francesco Mondada ανέπτυξαν το μικρό αυτόνομο κινούμενο ρομπότ, που είχε ως σκοπό ερευνητικές δραστηριότητες. Το συγκεκριμένο project υποστηρίχτηκε από το εργαστήριο του Πολυτεχνείου της Λωζάννης, LAMIEPFL lab. Αυτό το ρομπότ μπορούσε να ελέγχετε είτε από ένα σειριακό καλώδιο και ως εκ τούτου, από έναν απομακρυσμένο υπολογιστή, ή από τον κώδικα που μπορούσατε να κατεβάσετε απευθείας, σε μία από τις τρεις τράπεζες μνήμης, για αυτόνομη λειτουργία.



Εικόνα 1.2.48 και 1.2.49: Το Khepera από ψηλά και σε πλάγια λήψη

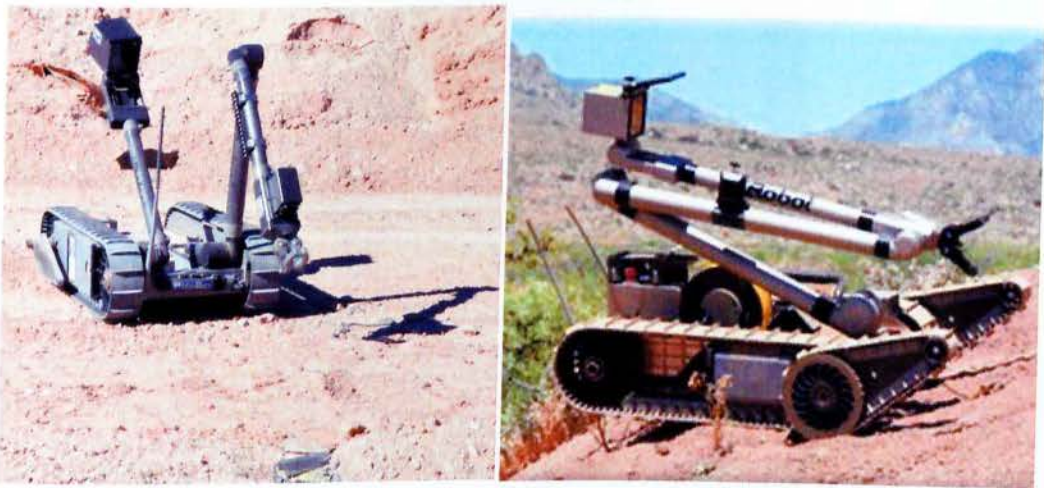
1996–1997 - Sojourner : Η NASA στέλνει στον Άρη το Mars Pathfinder μαζί με το ρόβερ του, το Sojourner, το οποίο είχε ως σκοπό την εξερεύνηση της επιφάνειας του, δεχόμενο εντολές από τη Γη. Το Sojourner ήταν το πρώτο ρόβερ που πάτησε την επιφάνεια του Άρη. Επίσης ήταν εξοπλισμένο με κάμερες μπρός και πίσω καθώς και με σύστημα αποφυγής κινδύνων, το οποίο επέτρεπε στο ρόβερ να βρίσκει αυτόνομα το δρόμο του μέσα στο άγνωστο έδαφος του Άρη. Είχε ηλιακούς συλλέκτες και μια μη επαναφορτιζόμενη μπαταρία, η οποία επέτρεπε περιορισμένες νυχτερινές λειτουργίες. Όταν οι μπαταρίες εξαντλούνταν, μπορούσε να λειτουργήσει μόνο κατά τη διάρκεια της ημέρας. Οι μπαταρίες ήταν λιθίου - θειονυλχλωριδίου (LiSOC12) και μπορούσαν να προσφέρουν 150 watt την ώρα. Επιπλέον, τα 0,22 τετραγωνικά μέτρα ηλιακών κυττάρων μπορούσαν να παράγουν το πολύ 15 Watt, ανάλογα με τις συνθήκες.



Εικόνα 1.2.50 και 1.2.51: Το Sojourner στον Άρη

Ο επεξεργαστής του ήταν ο 80C85 με 2MHz συχνότητα λειτουργίας. Είχε τέσσερις ξεχωριστές κάρτες μνήμης. Είχε μια 64Kbytes RAM της IBM, για τον κεντρικό επεξεργαστή, μια 16 Kbytes PROM (κατασκευάζεται από την Harris), μια 176 Kbytes μη πτητικής αποθήκευσης (κατασκευάζεται από την SEEQ) και μια 512 Kbytes προσωρινής αποθήκευσης δεδομένων (κατασκευάζεται από την Micron). Τα ηλεκτρονικά στεγάστηκαν μέσα σε ένα ηλεκτρονικό κουτί μέσα στο ρόβερ. Η επικοινωνία με το σταθμό βάσης γινόταν με 9600 baud modems.

1999 - PackBot: Παρουσιάζεται το τηλεκατευθυνόμενο στρατιωτικό ρομπότ.



Εικόνα 1.2.52 και 1.2.53: Το PackBot σε δράση

2001 - Swarm-bots: Μοιάζουν με αποικίες εντόμων, και αποτελούνται από έναν μεγάλο αριθμό μικρών ανεξάρτητων ρομπότ, τα οποία αλληλεπιδρούν μεταξύ τους και όλα μαζί εκτελούν πολύπλοκες εργασίες.



Εικόνα 1.2.54: Τα Swarm-bots μέσα σε λαβύρινθο

2002 – Roomba: 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. Διαθέτει ένα σύνολο βασικών αισθητήρων που βοηθούν τις εργασίες, όπως αλλαγή κατεύθυνσης κατά την πρόσκρουση σε εμπόδια, ανίχνευση βρώμικων σημείων στο πάτωμα και την πρόληψη πτώσης του Roomba από τις σκάλες. Χρησιμοποιεί δύο ανεξαρτήτους σε λειτουργία τροχούς για να πραγματοποιεί στροφές 360 μοιρών.



Εικόνα 1.2.55: Το Roomba σε λειτουργία

2004 – Spirit και Opportunity MER-A: Είναι δύο ρομποτικά ρόβερ στον Άρη, που δραστηριοποιούνται, το πρώτο από το 2004-2010 και το δεύτερο από το 2004 μέχρι και σήμερα. Προσγειώθηκαν με επιτυχία στον Άρη τον Γενάρη του 2004, με διαφορά τρεις εβδομάδες το ένα (Spirit) από το άλλο (Opportunity). Το Spirit προσγειώθηκε στην μια πλευρά του πλανήτη και το Opportunity στην άλλη. Το Spirit και το Opportunity έχουν έξι τροχούς, με διαστάσεις 1,5 m ύψος, 2,3 m πλάτος, 1,6 m μήκος και ζυγίζουν 180 κιλά. Οι έξι τροχοί τους στηρίζονται σε ένα φορητό σύστημα που τους επιτρέπει την κινητικότητα σε ανώμαλο έδαφος. Κάθε τροχός έχει το δικό του κινητήρα με την μέγιστη ταχύτητα τους να φτάνει τα 0,18 km/h. Οι ηλιακές συστοιχίες τους παράγουν περίπου 140 watt για έως και τέσσερις ώρες στον Άρη (sol), ενώ οι επαναφορτιζόμενες μπαταρίες ιόντων λιθίου αποθηκεύουν ενέργεια για χρήση τη νύχτα. Επί του σκάφους, ο υπολογιστής που χρησιμοποιούν είναι ένας RAD6000 στα 20 MHz με 128 MB DRAM, 3 MB EEPROM, και 256 MB μνήμη

flash. Η θερμοκρασία λειτουργίας των rover κυμαίνεται από -40°C έως $+40^{\circ}\text{C}$ και μια μεμβράνη χρυσού με ένα στρώμα διοξειδίου του πυριτίου τους παρέχουν μόνωση. Η επικοινωνία εξαρτάται από μια κατευθυνόμενη κεραία χαμηλής απολαβής που επικοινωνεί σε ένα χαμηλό ρυθμό δεδομένων και μιας περιστρεφόμενης, υψηλής απολαβής κεραίας, και οι δύο σε άμεση επαφή με τη Γη. Στόχος των δύο ρόβερ είναι η εξερεύνηση των πετρωμάτων και του εδάφους του Άρη, η αναζήτηση σιδήρου που εμπεριέχουν τα ορυκτά καθώς και ο καθορισμός των γεωλογικών διαδικασιών που έχουν διαμόρφωση την τοπική έκταση.



Εικόνα 1.2.56 και 1.2.57: Αριστερά το Spirit σε εικονική φωτογραφία στον Άρη και δεξιά το Opportunity στις εγκαταστάσεις της NASA

2012 - Curiosity: Το πυρηνοκίνητο Curiosity (ένα εξάτροχο όχημα ενός τόνου) αποτελεί το μεγαλύτερο τροχοφόρο ρομπότ που έχει εκτοξευτεί μέχρι σήμερα στο Διάστημα. Το ρομπότ των δύομισι δισεκατομμυρίων δολαρίων, πέντε φορές μεγαλύτερο από τα δίδυμα ρομπότ Spirit και Opportunity, εκτοξεύτηκε τον Νοέμβριο του 2011 με κύριο αντικειμενικό στόχο να εξετάσει αν ο γειτονικός πλανήτης ήταν ποτέ φιλόξενος για τη ζωή. Το Curiosity προσεδαφίστηκε μέσα στον κρατήρα Gale, το μεγάλο βάθος του οποίου, θα επιτρέψει στο ρομποτικό γεωλόγο να εξετάσει τι κρύβεται κάτω από την απέραντη, άνυδρη έρημο που καλύπτει σήμερα σχεδόν ολόκληρο τον πλανήτη. Τροφοδοτούμενο από μία θερμοηλεκτρική γεννήτρια πλουτωνίου, το ρομπότ θα περάσει στη συνέχεια τουλάχιστον ένα γήινο έτος εξετάζοντας τα πετρώματα με κάμερες, λέιζερ και άλλα όργανα.



Εικόνα 1.2.58: Το Curiosity λίγες μέρες πριν ταξιδέψει στον Άρη

ΚΕΦΑΛΑΙΟ 2

ΚΑΤΑΣΚΕΥΗ

2.1 Arduino UNO Rev3

2.1.1 Ιστορία

Το 2005 στην Ivrea της Ιταλίας κατασκευάζεται μία συσκευή η οποία θα είχε την δυνατότητα να ελέγχει και να αλληλεπιδρά σύμφωνα με το περιβάλλον. Σκοπός των κατασκευαστών ήταν αυτή η συσκευή να κοστίζει λιγότερο σε σχέση με άλλες παρόμοιων δυνατοτήτων. Η ομάδα αποτελούνταν από τους Massimo Banzi, David Cuartielles, Tom Igoe, David Mellis και Gianluca Martino. Το όνομα της συσκευής έχει τις ρίζες του από τον Arduino of Ivrea, έναν βασιλιά της Ιταλίας του ενάτου αιώνα όπου κατοικούσε στην ίδια πόλη. Η συσκευή ονομάστηκε “Arduino” που αντιστοιχούσε σε ένα ιταλικό ανδρικό όνομα και σήμαινε “ισχυρός φίλος”.

Το Arduino αναπτύχθηκε σύμφωνα με την πλατφόρμα Wiring, μία πτυχιακή εργασία του Hernando Barragan από το Interaction Design Institute Ivrea. Είχε ως στόχο να είναι μία ηλεκτρονική εκδοχή της Processing που θα χρησιμοποιούσε ένα περιβάλλον προγραμματισμού δικό της αλλά θα έμοιαζε σχεδιαστικά και συντακτικά με αυτό της Processing. Όχι πολύ καιρό πριν, αυτοί που εργάζονταν πάνω στον τομέα του hardware σήμαινε ότι κατασκεύαζαν κυκλώματα από το μηδέν, χρησιμοποιώντας εκατοντάδες διαφορετικές ηλεκτρονικές διατάξεις όπως αντιστάσεις, πυκνωτές, πηνία, τρανζίστορ, και πολλά άλλα. Τα κύκλωμα αυτά ήταν ενσύρματα με σκοπό να πραγματοποιήσουν συγκεκριμένες εργασίες.

Όταν απαιτούνταν αλλαγές στις εργασίες τους, τότε έπρεπε να γίνουν και κάποιες χρονοβόρες αλλαγές στον σχεδιασμό των κυκλωμάτων, όπως αποσυνγκολλήσεις και συγκολλήσεις των απαιτούμενων διατάξεων, αποσυνδέσεις και συνδέσεις καλωδίων κ.α.. Με την εμφάνιση της ψηφιακής τεχνολογίας και των μικροεπεξεργαστών, αυτές οι λειτουργίες οι οποίες πρώτα έπρεπε να γίνουν με καλώδια και διατάξεις τώρα αντικαταστάθηκαν από τα λογισμικά προγράμματα.

Το λογισμικό είναι πιο εύκολο να τροποποιηθεί από ότι το hardware. Με μερικά πατήματα πλήκτρων, μπορεί να αλλάξει ριζικά η λογική μίας συσκευής και να δημιουργηθούν επιπλέον ακόμη δύο ή τρεις δοκιμαστικές εκδόσεις. Η διαδικασία αυτή θα απαιτούσε το ίδιο χρονικό διάστημα όσο και ο χρόνος που απαιτείται για να κολληθεί ένα ζεύγος αντιστάσεων.

Όπως είναι φυσικό μαζί με τον χρόνο μειώθηκε και το κόστος για τις τροποποιήσεις καθώς και για τις δοκιμαστικές εκδόσεις. Όπως το περιγράφει ο δημιουργός του, το Arduino είναι μία open-source (ανοικτού κώδικα) πλατφόρμα «πρωτοτυποποίησης» ηλεκτρονικών κυκλωμάτων 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.

Το Arduino αποτελείται από δύο κύρια μέρη, την πλακέτα Arduino το οποίο είναι το κομμάτι του hardware πάνω στο οποίο εργάζεται ο κατασκευαστής όταν

πραγματοποιεί μία κατασκευή ενώ το δεύτερο τμήμα είναι το περιβάλλον Arduino IDE, το κομμάτι του λογισμικού που τρέχει στον υπολογιστή. Το IDE χρησιμοποιείται για να δημιουργηθεί ένα sketch (ένα μικρό πρόγραμμα στον υπολογιστή) που φορτώνεται στον μικροελεγκτή της πλακέτας Arduino. Έτσι το sketch λέει στην πλακέτα Arduino τι πρέπει να κάνει.

2.1.2 Η πλακέτα Arduino

Η πλακέτα Arduino είναι ένα μικρό κύκλωμα που περιέχει ένα ολοκληρωμένο σύστημα υπολογιστή χρησιμοποιώντας ένα μικρό chip (ολοκληρωμένο κύκλωμα) που είναι ο μικροελεγκτής. Αυτός ο υπολογιστής είναι τουλάχιστον χίλιες φορές λιγότερο ισχυρός από ένα MacBook, αλλά είναι πολύ φθηνότερος και πολύ χρήσιμος για την κατασκευή ηλεκτρονικών συσκευών. Μάλιστα κάποιος θα μπορούσε να ισχυριστεί ότι λειτουργικά το Arduino μοιάζει πολύ με το NXT Brick των Lego Mindstorms NXT. Άλλωστε η ρομποτική είναι μία από τις πολλές κατηγορίες στις οποίες το Arduino διαπρέπει.

Η ομάδα του Arduino έχει τοποθετήσει σε αυτήν την πλακέτα όλα τα απαραίτητα στοιχεία που απαιτούνται για τον μικροελεγκτή ώστε να μπορεί να λειτουργεί σωστά και να μπορεί να επικοινωνεί με τον υπολογιστή και άλλες σειριακές συσκευές.



Εικόνα 2.1.2.1 και 2.1.2.2: Μπροστινή και πίσω όψη της πλακέτας Arduino.

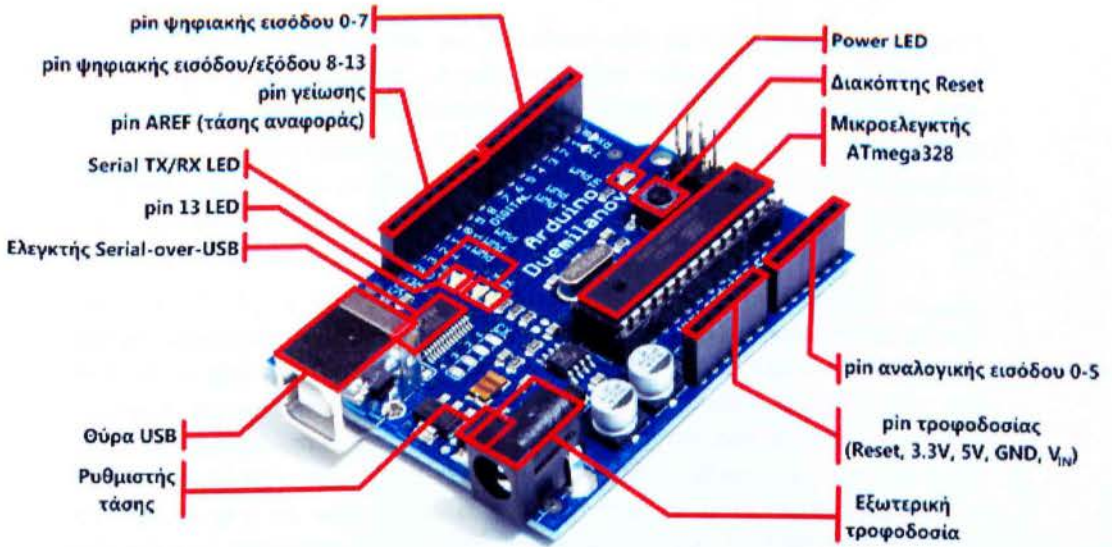
Ειδικότερα μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωση του σε άλλα κυκλώματα. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz (ή κεραμικό αντηχητή σε κάποιες παραλλαγές). Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα bootloader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής.

Γενικά όλες οι πλακέτες είναι προγραμματισμένες μέσω μιας σειριακής σύνδεσης RS-232, αλλά ο τρόπος με τον οποίο αυτό υλοποιείται ποικίλλει ανάλογα με την έκδοση. Οι σειριακές πλακέτες Arduino περιέχουν ένα απλό κύκλωμα αντιστροφής για την μετατροπή ανάμεσα στα σήματα των επιπέδων RS-232 και TTL.

Οι πλακέτες Arduino που κυκλοφορούν σήμερα στην αγορά, συμπεριλαμβανόμενης και της Uno, προγραμματίζονται μέσω USB, εφαρμόζοντας ένα chip προσαρμογέα USB-to-Serial όπως το FTDI FT232.

2.1.3 Είσοδοι - Έξοδοι

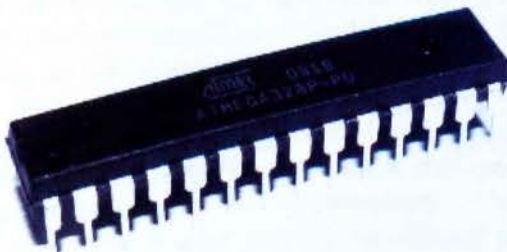
Καταρχήν το Arduino διαθέτει σειριακό interface. Ο μικροελεγκτής ATmega (εικόνα 2.1.3.2) υποστηρίζει σειριακή επικοινωνία, την οποία το Arduino προωθεί μέσα από έναν ελεγκτή Serial-over-USB ώστε να συνδέεται με τον υπολογιστή μέσω USB. Η σύνδεση αυτή χρησιμοποιείται για την μεταφορά των προγραμμάτων 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.



Εικόνα 2.1.3.1: Τα στοιχεία που απαρτίζουν την πλακέτα Arduino.

Επιπλέον, στην πάνω πλευρά του Arduino βρίσκονται 14 θηλυκά pin, αριθμημένα από το 0 έως το 13, που μπορούν να λειτουργήσουν ως ψηφιακές εισόδους και εξόδους. 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 σε κατάσταση HIGH ή LOW, οπότε το Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι ρεύμα στο συγκεκριμένο pin. Με αυτόν τον τρόπο είναι εφικτό να ανάψει και να σβήσει ένα LED που είναι συνδεδεμένο στο συγκεκριμένο pin. Αν πάλι ένα από αυτά τα pin έχει ρυθμιστή ως ψηφιακή είσοδος μέσα από το πρόγραμμά, μπορεί με την κατάλληλη εντολή να διαβαστεί η κατάστασή του (HIGH ή LOW) ανάλογα με το αν η εξωτερική συσκευή που έχει συνδεθεί σε αυτό το pin διοχετεύει ή όχι ρεύμα στο pin.



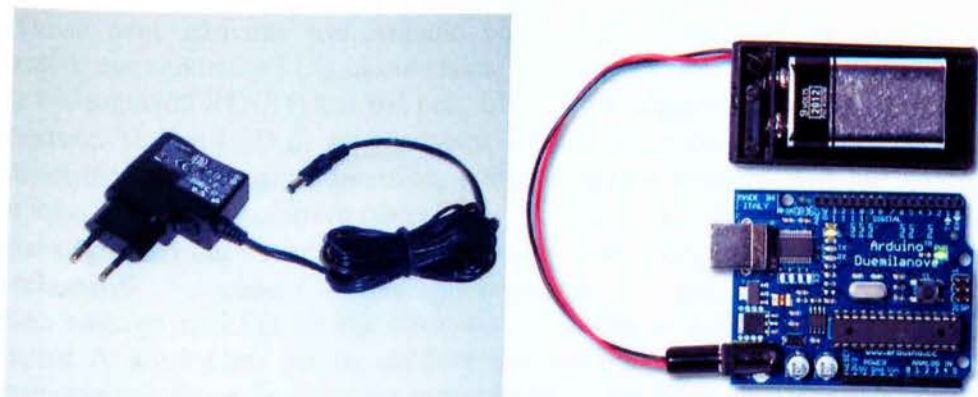
Εικόνα 2.1.3.2: Μικροελεγκτής ATMEGA328P-PU

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 πρόγραμμα ενεργοποιεί την σειριακή θύρα. Έτσι, όταν λόγω χάρη το πρόγραμμα στέλνει δεδομένα στην σειριακή θύρα, 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. ένα δεύτερο Arduino στο δικό του pin 1). Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμα ενεργοποιηθεί το σειριακό interface, χάνονται 2 ψηφιακές εισοδοι/έξοδοι.
- Τα pin 2 και 3 λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Με άλλα λόγια, μπορούν να ρυθμιστούν μέσα από το πρόγραμμα 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, αν συνδεθεί λόγω χάρη ένα LED σε κάποιο από αυτά τα pin, μπορεί να ελεγχθεί πλήρως η φωτεινότητά του με ανάλυση 8bit (256 καταστάσεις από 0-σβηστό ως 255-πλήρως αναμμένο) αντί να υπάρχει απλά η δυνατότητα να είναι απλά αναμμένο ή σβηστό που παρέχουν οι υπόλοιπες ψηφιακές έξοδοι. Το PWM παίρνει ένα εύρος τιμών από το 0 έως το 255. Με λίγα λόγια το PWM δεν είναι πραγματικά αναλογικό σύστημα, έτσι θέτοντας στην έξοδο την τιμή 127, δεν σημαίνει ότι η έξοδος θα παρέχει 2.5V αντί της κανονικής τιμής των 5V, αλλά ότι θα δίνει έναν παλμό που η τάση του θα εναλλάσσεται με μεγάλη συχνότητα και για ίσα χρονικά διαστήματα μεταξύ των τιμών 0V και 5V με σκοπό η μέση τιμή να ισούται με 2,5V.
- Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN, υπάρχει μια ακόμη σειρά από 6 pin, αριθμημένα από το 0 έως το 5. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC (Analog to Digital Converter) που είναι ενσωματωμένο στον μικροελεγκτή. Για παράδειγμα, αν τροφοδοτηθεί ένα από αυτά τα pin με μία τάση η οποία μπορεί να κυμανθεί με ένα ποτενσιόμετρο από 0V ως μία τάση αναφοράς Vref (η οποία αν δεν γίνει κάποια αλλαγή είναι προρυθμισμένη στα 5V), τότε μέσα από το πρόγραμμα μπορεί να «διαβαστεί» η τιμή της θύρας ως ένας ακεραίος αριθμός χωρητικότητας 10-bit, από το 0 (όταν η τάση στο pin είναι 0V) μέχρι το 1023 (όταν η τάση στο pin είναι 5V). Η τάση αναφοράς μπορεί να ρυθμιστεί με μία εντολή όπως για παράδειγμα στα 1.1V. Ένας άλλος τρόπος όπου η τάση αναφοράς μπορεί να δηλωθεί από τον προγραμματιστή είναι τροφοδοτώντας με μία εξωτερική τάση αναφοράς τη θύρα με την σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας. Έτσι, αν τροφοδοτηθεί η θύρα AREF με 3.3V και στην συνέχεια εκτελεσθεί η εντολή να διαβαστεί κάποιο pin αναλογικής εισόδου στο οποίο εφαρμόζετε τάση 1.65V, το Arduino θα επιστρέψει την τιμή 512. Τέλος, καθένα από τα 6 αυτά pin, με κατάλληλη εντολή μέσα από το πρόγραμμα μπορεί να μετατραπεί σε ψηφιακό pin εισόδου/εξόδου όπως τα 14 που βρίσκονται στην απέναντι πλευρά και τα οποία περιγράφηκαν πριν. Σε αυτή την περίπτωση τα pin μετονομάζονται από 0~5 σε 14~19 αντίστοιχα.

2.1.4 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 φισ των 2.1mm (θετικός πόλος στο κέντρο) και βρίσκεται στην κάτω-αριστερή γωνία του Arduino.



Εικόνα 2.1.4.1 και 2.1.4.2: Τροφοδοσία με DC προσαρμογέα ή με αλκαλική μπαταρία

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.

Δίπλα από τα pin αναλογικής εισόδου, υπάρχει μια ακόμα συστοιχία από 6 pin με την σήμανση POWER. Η λειτουργία του καθενός έχει ως εξής:

- Το πρώτο, με την ένδειξη RESET, όταν γειωθεί (σε οποιοδήποτε από τα 3 pin με την ένδειξη GND που υπάρχουν στο Arduino) έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.
- Το δεύτερο, με την ένδειξη 3.3V, μπορεί να τροφοδοτήσει διατάξεις, συσκευές ή αισθητήρες με τάση 3.3V. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial-over-USB και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA.
- Το τρίτο, με την ένδειξη 5V, μπορεί να τροφοδοτήσει και αυτό διάφορες διατάξεις, συσκευές ή αισθητήρες με τάση 5V. Ανάλογα με τον τρόπο τροφοδοσίας του ίδιου του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την θύρα USB (που ούτως ή άλλως λειτουργεί στα 5V), είτε από την εξωτερική τροφοδοσία αφού αυτή περάσει από ένα ρυθμιστή τάσης για να την «κατεβάσει» στα 5V.
- Το τέταρτο και το πέμπτο pin, με την ένδειξη GND, είναι γειώσεις.
- Το έκτο και τελευταίο pin, με την ένδειξη Vin έχει διπλό ρόλο. Σε συνδυασμό με το pin γείωσης δίπλα του, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino, στην περίπτωση που δεν βολεύει να χρησιμοποιηθεί η υποδοχή του φισ των 2.1mm. Αν όμως υπάρχει ήδη συνδεδεμένη εξωτερική τροφοδοσία μέσω του φισ, τότε μπορεί να χρησιμοποιηθεί αυτό το pin για να τροφοδοτήσει εξαρτήματα και συσκευές με

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

- Τα pin 0 και 1 λειτουργούν ως RX και TX της σειριακής όταν το πρόγραμμα ενεργοποιεί την σειριακή θύρα. Έτσι, όταν λόγω χάρη το πρόγραμμα στέλνει δεδομένα στην σειριακή θύρα, 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. ένα δεύτερο Arduino στο δικό του pin 1). Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμα ενεργοποιηθεί το σειριακό interface, χάνονται 2 ψηφιακές εισοδοί/έξοδοι.
- Τα pin 2 και 3 λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Με άλλα λόγια, μπορούν να ρυθμιστούν μέσα από το πρόγραμμα 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, αν συνδεθεί λόγω χάρη ένα LED σε κάποιο από αυτά τα pin, μπορεί να ελεγχθεί πλήρως η φωτεινότητά του με ανάλυση 8bit (256 καταστάσεις από 0-σβηστό ως 255-πλήρως αναμμένο) αντί να υπάρχει απλά η δυνατότητα να είναι απλά αναμμένο ή σβηστό που παρέχουν οι υπόλοιπες ψηφιακές έξοδοι. Το PWM παίρνει ένα εύρος τιμών από το 0 έως το 255. Με λίγα λόγια το PWM δεν είναι πραγματικά αναλογικό σύστημα, έτσι θέτοντας στην έξοδο την τιμή 127, δεν σημαίνει ότι η έξοδος θα παρέχει 2.5V αντί της κανονικής τιμής των 5V, αλλά ότι θα δίνει έναν παλμό που η τάση του θα εναλλάσσεται με μεγάλη συχνότητα και για ίσα χρονικά διαστήματα μεταξύ των τιμών 0V και 5V με σκοπό η μέση τιμή να ισούται με 2,5V.
- Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN, υπάρχει μια ακόμη σειρά από 6 pin, αριθμημένα από το 0 έως το 5. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC (Analog to Digital Converter) που είναι ενσωματωμένο στον μικροελεγκτή. Για παράδειγμα, αν τροφοδοτηθεί ένα από αυτά τα pin με μία τάση η οποία μπορεί να κυμανθεί με ένα ποτενσιόμετρο από 0V ως μία τάση αναφοράς Vref (η οποία αν δεν γίνει κάποια αλλαγή είναι προρυθμισμένη στα 5V), τότε μέσα από το πρόγραμμα μπορεί να «διαβαστεί» η τιμή της θύρας ως ένας ακέραιος αριθμός χωρητικότητας 10-bit, από το 0 (όταν η τάση στο pin είναι 0V) μέχρι το 1023 (όταν η τάση στο pin είναι 5V). Η τάση αναφοράς μπορεί να ρυθμιστεί με μία εντολή όπως για παράδειγμα στα 1.1V. Ένας άλλος τρόπος όπου η τάση αναφοράς μπορεί να δηλωθεί από τον προγραμματιστή είναι τροφοδοτώντας με μία εξωτερική τάση αναφοράς τη θύρα με την σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας. Έτσι, αν τροφοδοτηθεί η θύρα AREF με 3.3V και στην συνέχεια εκτελεσθεί η εντολή να διαβαστεί κάποιο pin αναλογικής εισόδου στο οποίο εφαρμόζετε τάση 1.65V, το Arduino θα επιστρέψει την τιμή 512. Τέλος, καθένα από τα 6 αυτά pin, με κατάλληλη εντολή μέσα από το πρόγραμμα μπορεί να μετατραπεί σε ψηφιακό pin εισόδου/εξόδου όπως τα 14 που βρίσκονται στην απέναντι πλευρά και τα οποία περιγράφηκαν πριν. Σε αυτή την περίπτωση τα pin μετονομάζονται από 0~5 σε 14~19 αντίστοιχα.

την πλήρη τάση της εξωτερικής τροφοδοσίας (7~12V), πριν αυτή περάσει από τον ρυθμιστή τάσης όπως γίνεται με το pin των 5V.

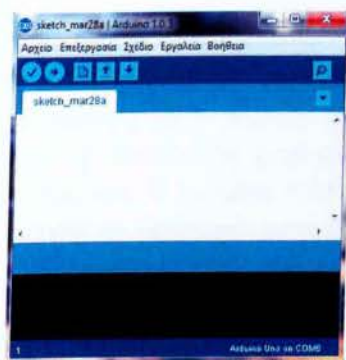
2.1.5 Ενσωματωμένα κουμπιά και LED

Πάνω στην πλακέτα του Arduino υπάρχει ένας διακόπτης micro-switch και τέσσερα μικροσκοπικά LED επιφανειακής στήριξης. Η λειτουργία του διακόπτη (που έχει την σήμανση RESET) και του ενός LED με την σήμανση POWER είναι μάλλον προφανής. Τα δύο LED με τις σημάνσεις TX και RX, χρησιμοποιούνται ως ένδειξη λειτουργίας του σειριακού interface, καθώς ανάβουν όταν το Arduino στέλνει ή λαμβάνει αντίστοιχα δεδομένα μέσω USB. Τα LED αυτά ελέγχονται από τον ελεγκτή Serial-over-USB και συνεπώς δεν λειτουργούν όταν η σειριακή επικοινωνία γίνεται αποκλειστικά μέσω των ψηφιακών pin 0 και 1. Τέλος, υπάρχει το LED με την σήμανση L. Η βασική λειτουργία του LED στην πλακέτα Arduino είναι για να αναβοσβήνει συνήθως για δοκιμαστικό σκοπό. Οι κατασκευαστές του σκέφτηκαν να ενσωματώσουν ένα LED στην πλακέτα, το οποίο σύνδεσαν στο ψηφιακό pin 13. Έτσι, ακόμα και αν δεν έχει συνδέσει τίποτα πάνω στο φυσικό pin 13, αναθέτοντάς του την τιμή HIGH μέσα από το πρόγραμμά, θα ανάψει το ενσωματωμένο LED.

2.1.6 Το περιβάλλον Arduino IDE (integrated development environment)

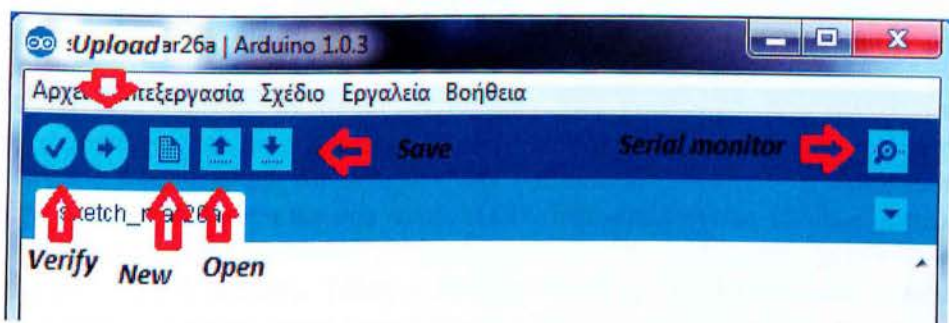
Το Arduino IDE είναι ένα πρόγραμμα βασισμένο σε Java και συγκεκριμένα παρέχει: ένα πρακτικό περιβάλλον για την συγγραφή των προγραμμάτων σας (τα οποία ονομάζονται sketch στην ορολογία του Arduino) με συντακτική χρωματική σήμανση, αρκετά έτοιμα παραδείγματα, μερικές έτοιμες βιβλιοθήκες για προέκταση της γλώσσας και για να χειρίζεστε εύκολα μέσα από τον κώδικά σας τα εξαρτήματα που συνδέετε στο Arduino, τον compiler για την μεταγλώττιση των sketch σας, ένα serial monitor που παρακολουθεί τις επικοινωνίες της σειριακής (USB), αναλαμβάνει να στείλει αλφαριθμητικά της επιλογής σας στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για το debugging των sketch σας και την επιλογή να ανεβάσετε το μεταγλωττισμένο sketch στο Arduino.

Το Arduino IDE είναι αρκετά απλούστερο σε αντίθεση με άλλα περιβάλλοντα ανάπτυξης λογισμικού όπως το Eclipse, το Xcode και το Visual Studio. Κυρίως αποτελείται από έναν editor (κειμενογράφο), έναν compiler, ένα loader και ένα serial monitor. Δεν περιέχει προχωρημένες λειτουργίες όπως debugging ή code completion, δίνει μόνο τη δυνατότητα για μερικές ρυθμίσεις στα «preferences».



Εικόνα 2.1.6.1: Περιβάλλον Arduino IDE

Παρακάτω δίνετε η εικόνα του toolbar του Arduino IDE και η αναλυτική λειτουργία των κουμπιών του toolbar που δίνει άμεση πρόσβαση στις λειτουργίες που χρειάζονται περισσότερο για την ανάπτυξη των εφαρμογών που επιθυμεί να αναπτύξει ο χρήστης.



Εικόνα 2.1.6.2 - Το toolbar του Arduino IDE

Με το κουμπί Verify ο χρήστης μπορεί να μεταγλωττίσει (compile) τον κώδικα που βρίσκεται εκείνη την στιγμή στον editor. Το κουμπί Verify εκτός από τον συντακτικό έλεγχο, αφού μεταγλωττιστεί ο κώδικας στη συνέχεια τον μετατρέπει σε μορφή κατάλληλη για να «φορτωθεί» στον μικροελεγκτή του Arduino.

Το κουμπί New δημιουργεί ένα καινούριο πρόγραμμα διαγράφοντας οτιδήποτε υπάρχει στον editor. Πριν όμως πραγματοποιήσει αυτήν την ενέργεια, δίνει στον χρήστη την ευκαιρία να αποθηκεύσει το υπάρχον sketch (πρόγραμμα) στην περίπτωση που έχει κάνει κάποιες αλλαγές σε αυτό.

Με το κουμπί Open μπορεί ο χρήστης να ανοίξει ένα υπάρχον πρόγραμμα από το σύστημα του.

Το κουμπί Save αποθηκεύει τις αλλαγές που έχουν γίνει στο πρόγραμμα που επεξεργάστηκε στον Editor.

Το κουμπί Upload όπως και το κουμπί Verify μεταγλωττίζει τον υπάρχον κώδικα στον editor. Με τη διαφορά όμως ότι αφού ελέγξει για τυχόν συντακτικά λάθη και το μετατρέπει σε μορφή κατάλληλη για το Arduino, στη συνέχεια θα τον προωθήσει στην θύρα που έχει επιλέξει ο προγραμματιστής από το μενού Tools > Serial Port ώστε να «φορτωθεί» στον μικροελεγκτή.

Ο υπολογιστής μπορεί να επικοινωνήσει με το Arduino μέσω σειριακής σύνδεσης. Επιλέγοντας το κουμπί Serial Monitor ανοίγει ένα serial monitor παράθυρο όπου επιτρέπει να παρακολουθεί ο προγραμματιστής τα δεδομένα που στέλνονται προς το Arduino αλλά και τα δεδομένα που στέλνονται από το Arduino προς τον υπολογιστή.

Επιλέγοντας από το μενού το Sketch πέρα από τις λειτουργίες που αναφέρθηκαν παραπάνω για την εντολή Verify/compile, εμφανίζονται και κάποιες άλλες ενδιαφέρουσες λειτουργίες. Η επιλογή Show Sketch folder είναι μία συντόμευση η οποία ανοίγει σε ένα παράθυρο την διεύθυνση στην οποία το λειτουργικό σύστημα αποθηκεύει τα αρχεία των εφαρμογών. Η επιλογή Add file επιτρέπει στον χρήστη να ανοίξει ένα αρχείο το οποίο μπορεί να βρίσκεται οπουδήποτε μέσα στο σύστημα και να το αποθηκεύσει στον ίδιο φάκελο όπου ανήκει και η εφαρμογή .

Τελευταία είναι η επιλογή Import Library. Ο προγραμματιστής έχει την δυνατότητα να εισάγει στο πρόγραμμα που αναπτύσσει οποιαδήποτε βιβλιοθήκη, είτε αυτή είναι του Arduino είτε κάποια που δημιούργησε ο ίδιος.

Δίπλα από το Sketch στο μενού υπάρχει η επιλογή Tools. Σε αυτή την στήλη του μενού ο προγραμματιστής έχει την δυνατότητα να επιλέξει την θύρα (serial port) με την οποία θα επικοινωνήσει ο υπολογιστής με το Arduino, καθώς επίσης και πια συγκεκριμένη έκδοση Arduino έχει (board). Εκτός όμως από αυτές τις δύο βασικές λειτουργίες υπάρχουν και κάποιες άλλες. Το Auto Format το οποίο μορφοποιεί των κώδικα που βρίσκεται στον editor κατάλληλα για να διαβάζεται ευκολότερα.

Το Archive Sketch το οποίο μετατρέπει την εφαρμογή που αναπτύσσει ο προγραμματιστής σε ένα αρχείο zip και το αποθηκεύει. Ακόμα σε μερικές περιπτώσεις όταν ένα αρχείο ανοιχθεί στο Arduino IDE υπάρχει η περίπτωση να περιέχει χαρακτήρες οι οποίοι δεν είναι ASCII. Έτσι επιλέγοντας το Fix Encoding & Reload ο κώδικας θα ανανεωθεί αντικαθιστώντας τους περιεργους χαρακτήρες σε χαρακτήρες UTF-8 έκδοση. Τέλος η επιλογή Burning Bootloader απευθύνεται σε προχωρημένους χρήστες όπου τους δίνεται η δυνατότητα να χρησιμοποιήσουν την μνήμη που αντιστοιχεί στον bootloader.

2.1.7 Γλώσσα προγραμματισμού

Η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring, μια παραλλαγή της C/C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega, και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++. Για compiler χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc.

Λόγω της καταγωγής της από την C, στην γλώσσα του Arduino χρησιμοποιούνται ουσιαστικά οι ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπων δεδομένων και τους ίδιους τελεστές όπως και στην C. Πέρα από αυτές όμως, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για την διαχείριση του ειδικού hardware του Arduino. Οι πιο σημαντικές από αυτές επεξηγούνται στον πίνακα που ακολουθεί:

Όρισμα	Είδος	Τύπος	Παράμετροι	Περιγραφή
LOW	Σταθερά	int	-	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
HIGH	Σταθερά	int	-	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
INPUT	Σταθερά	int	-	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
OUTPUT	Σταθερά	int	-	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
pinMode	Εντολή	-	(pin, mode)	Καθορίζει αν το συγκεκριμένο ψηφιακό pin θα είναι pin εισόδου ή pin εξόδου ανάλογα με την τιμή που δίνεται στην παράμετρο mode (INPUT ή OUTPUT αντίστοιχα).

digitalWrite	Εντολή	-	(<i>pin, pinstatus</i>)	Θέτει την κατάσταση <i>pinstatus</i> (HIGH ή LOW) στο συγκεκριμένο ψηφιακό <i>pin</i> .
digitalRead	Συνάρτηση	int	(<i>pin</i>)	Επιστρέφει την κατάσταση του συγκεκριμένου ψηφιακού <i>pin</i> (0 για LOW και 1 για HIGH) εφόσον αυτό είναι <i>pin</i> εισόδου.
analogReference	Εντολή	-	(<i>type</i>)	Δέχεται τις τιμές DEFAULT, INTERNAL ή EXTERNAL στην παράμετρο <i>type</i> για να καθορίσει την τάση αναφοράς (V_{ref}) των αναλογικών εισόδων (5V, 1.1V ή η εξωτερική τάση με την οποία τροφοδοτείται το <i>pin</i> AREF αντίστοιχα)
analogRead	Συνάρτηση	int	(<i>pin</i>)	Επιστρέφει έναν ακέραιο από 0 έως 1023, ανάλογα με την τάση που τροφοδοτείται το συγκεκριμένο <i>pin</i> αναλογικής εισόδου στην κλίμακα 0 ως V_{ref} .
analogWrite	Εντολή	-	(<i>pin, value</i>)	Θέτει το συγκεκριμένο ψηφιακό <i>pin</i> σε κατάσταση ψευδοαναλογικής εξόδου (PWM). Η παράμετρος <i>value</i> καθορίζει το πλάτος του παλμού σε σχέση με την περίοδο του παραγόμενου σήματος στην κλίμακα από 0 ως 255 (π.χ. με <i>value</i> 127, το πλάτος του παλμού είναι ίσο με μισή περίοδο).
millis	Συνάρτηση	unsigned long	()	Μετρητής που επιστρέφει το χρονικό διάστημα σε ms από την στιγμή που άρχισε η εκτέλεση του προγράμματος. Λάβετε υπόψη ότι λόγω του τύπου μεταβλητής (unsigned long δηλ. 32bit) θα γίνει overflow σε 2^{32} ms δηλαδή περίπου σε 50 μέρες, οπότε ο μετρητής θα ξεκινήσει πάλι από το μηδέν.
delay	Εντολή	-	(<i>time</i>)	Σταματά προσωρινά την ροή του προγράμματος για <i>time</i> ms. Η παράμετρος <i>time</i> είναι unsigned long (από 0 ως 2^{32}). Σημειώστε ότι παρά την προσωρινή παύση, συναρτήσεις των οποίων η εκτέλεση ενεργοποιείται από interrupt θα εκτελεστούν κανονικά κατά την διάρκεια μιας delay.
attachInterrupt	Εντολή	-	(<i>interrupt, function,</i>	Θέτει σε λειτουργία το συγκεκριμένο <i>interrupt</i> , ώστε να ενεργοποιεί την συνάρτηση <i>function</i> , κάθε φορά που

			<i>triggermode</i>)	ικανοποιείται η συνθήκη που ορίζεται από την παράμετρο <i>triggermode</i> : <ul style="list-style-type: none"> • LOW (ενεργοποίηση όταν η κατάσταση του pin που αντιστοιχεί στο συγκεκριμένο interrupt γίνει LOW) • RISING (όταν από LOW γίνεται HIGH) • FALLING (όταν από HIGH γίνεται LOW) • CHANGE (όταν αλλάξει κατάσταση γενικά)
<code>detachInterrupt</code>	Εντολή	-	<i>(interrupt)</i>	Απενεργοποιεί το συγκεκριμένο <i>interrupt</i> .
<code>noInterrupts</code>	Εντολή	-	<code>()</code>	Σταματά προσωρινά την λειτουργία όλων των <i>interrupt</i>
<code>interrupts</code>	Εντολή	-	<code>()</code>	Επαναφέρει την λειτουργία των <i>interrupt</i> που διακόπηκε προσωρινά από μια εντολή <code>noInterrupts</code> .
<code>Serial.begin</code>	Μέθοδος κλάσης	-	<i>(datarate)</i>	Θέτει τον ρυθμό μεταφοράς δεδομένων του σειριακού interface (σε baud)
<code>Serial.println</code>	Μέθοδος κλάσης	-	<i>(data)</i>	Διοχετεύει τα δεδομένα <i>data</i> για αποστολή μέσω του σειριακού interface. Η παράμετρος <i>data</i> μπορεί να είναι είτε αριθμός είτε αλφαριθμητικό.

Επιπλέον, στην γλώσσα του Arduino κάθε πρόγραμμα αποτελείται από δύο βασικές ρουτίνες ώστε να έχει την γενική δομή:

```
// Ενσωματώσεις βιβλιοθηκών, δηλώσεις μεταβλητών...
```

```
void setup()
```

```
{
  // ...
}
```

```
void loop()
```

```
{
  // ...
}
```

Η βασική ρουτίνα `setup()` εκτελείται μια φορά μόνο κατά την εκκίνηση του προγράμματος ενώ η βασική ρουτίνα `loop()` περιέχει τον βασικό κορμό του προγράμματος και η εκτέλεσή της επαναλαμβάνεται συνέχεια σαν ένας βρόχος `while(true)`.

2.2 Αισθητήρες υπερήχων

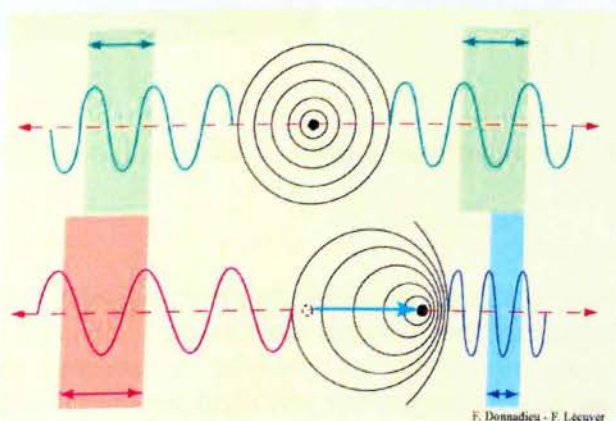
2.2.1 Εισαγωγή

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



Εικόνα 2.2.1.1: Το ηχητικό φάσμα

Παρόλο πάντως που οι άνθρωποι δεν τους ακούνε κάποια ζώα μπορούν και τους ακούν 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 ανιχνεύσουν τον στόχο που κυνηγού χάρη στο φαινόμενο Doppler.



Εικόνα 2.2.1.2: Σχηματική αναπαράσταση φαινομένου Doppler

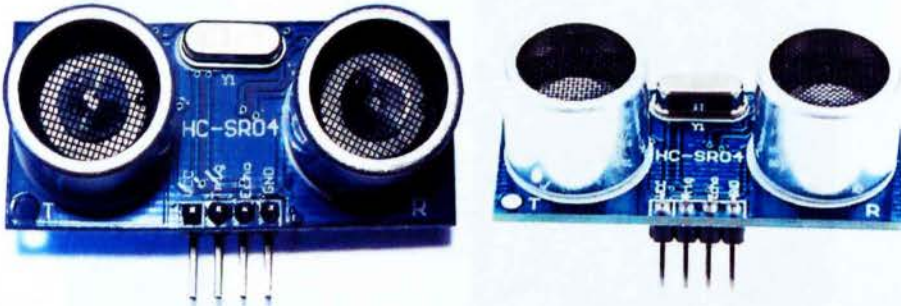
Το φαινόμενο Doppler είναι η παρατηρούμενη αλλαγή στη συχνότητα και το μήκος κύματος ενός κύματος από παρατηρητή που βρίσκεται σε σχετική κίνηση με την πηγή των κυμάτων. Για κύματα όπως τα ηχητικά κύματα, που διαδίδονται μέσα σε κάποιο υλικό μέσο, η ταχύτητα τόσο του παρατηρητή όσο και της πηγής, πρέπει 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. Καθεμιά από αυτές τις δύο επιδράσεις αναλύεται ξεχωριστά.

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

2.2.2 Αισθητήρας υπερήχων HC-SR04

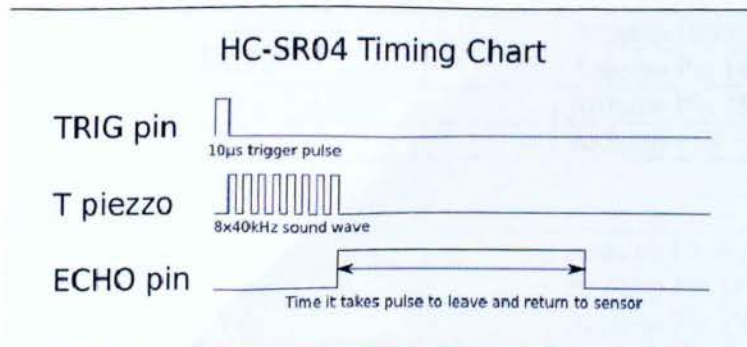
Για την διαδικασία εντοπισμού και αποφυγής των εμποδίων χρησιμοποιήθηκαν αισθητήρες υπερήχων. Οι αισθητήρες υπερήχων είναι μία φθηνή και σχετικά αξιόπιστη λύση χωρίς να απαιτούν πολύπλοκα κυκλώματα υποστήριξης. Είναι αρκετά δημοφιλείς στην κατηγορία των ρομποτικών οχημάτων.

Οι αισθητήρες υπερήχων λειτουργούν με την ίδια αρχή που λειτουργούν τα ραντάρ και τα σόναρ. Απαρτίζονται από ένα 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 υπόψη τους την αντανάκλαση ενός ραδιοκύματος ή ενός ηχητικού σήματος πάνω στο στόχο, υπολογίζοντας τον χρόνο που έκανε το σήμα για να καλύψει την απόσταση από τον αισθητήρα στο αντικείμενο και πίσω. Έτσι ο αισθητήρας στέλνει ριπές παλμών υψηλής συχνότητας (40kHz περίπου), ενώ η κάθε ριπή ταξιδεύει στον αέρα (και σε θερμοκρασία δωματίου 20-25C) με ταχύτητα 344m/s. Τα ηλεκτρονικά του αισθητήρα μετρούν το χρόνο λήψης του σήματος και τον μετατρέπουν σε μονάδα μήκους. Δεδομένου ότι η ταχύτητα του ήχου επηρεάζεται από τη θερμοκρασία αέρα, οι αισθητήρες υπερήχων περιλαμβάνουν έναν ενσωματωμένο αισθητήρα θερμοκρασίας.



Εικόνα 2.2.2.1 και 2.2.2.2: Αισθητήρας υπερήχων HC-SR04

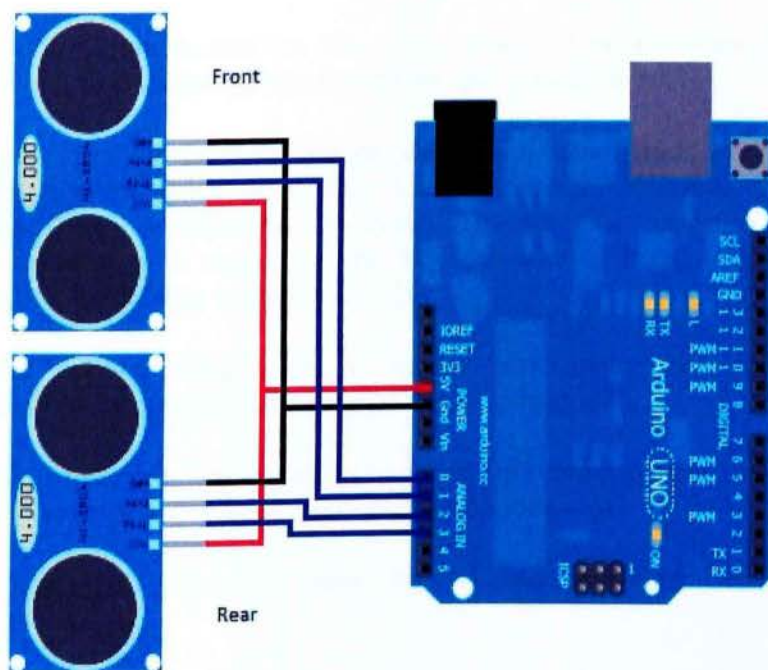
Εφαρμογές τους θα βρούμε σε ένα μεγάλο εύρος τεχνολογιών όπως την ανίχνευση προσέγγισης, την παρουσία ή την απουσία αντικειμένου, την ανίχνευση εμποδίων σε αυτοματοποιημένα οχήματα, τη μέτρηση απόστασης, τη μέτρηση στάθμης, τη μέτρηση της διεύθυνσης και της ταχύτητας του ανέμου κ.τ.λ. Η κατανάλωση ισχύος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



Εικόνα 2.2.2.3: Αρχή λειτουργίας αισθητήρα υπερήχων

2.2.3 Σύνδεση με Arduino

Για την σύνδεση των αισθητήρων με την πλακέτα Arduino χρησιμοποιήθηκαν οχτώ καλώδια. Η σύνδεση των αισθητήρων με την πλακέτα του Arduino γίνεται ως εξής: ο κάθε ένας αισθητήρας HC-SR04 διαθέτει τέσσερα pin (Vcc, Trigger, Echo, GND) εκ των οποίων δυο, τα Echo και Trigger, συνδέονται στα digital pins 14, 15 και 16, 17 του Arduino αντίστοιχα. Στο Vcc pin των αισθητήρων συνδέουμε τάση +5V από το Arduino και στο Gnd pin τους συνδέουμε το Gnd του Arduino. Στην παρακάτω εικόνα φαίνεται αναλυτικά η συνδεσμολογία που ακολουθήθηκε:



Εικόνα 2.2.3.1: Συνδεσμολογία πλακέτας Arduino με αισθητήρες υπερήχων

Στους παρακάτω πίνακες παρατίθενται αναλυτικά οι συνδέσεις των pin (των αισθητήρων) με την πλακέτα του Arduino όπως περιγράφηκαν παραπάνω:

Front HC-SR04

Pin 1	Gnd	Arduino Ground
Pin 2	Echo	Arduino Pin 14
Pin 3	Trig	Arduino Pin 15
Pin 4	Vcc	Arduino +5V

Rear HC-SR04

Pin 1	Gnd	Arduino Ground
Pin 2	Echo	Arduino Pin 16
Pin 3	Trig	Arduino Pin 17
Pin 4	Vcc	Arduino +5V

2.3 Κινητήρες

2.3.1 Εισαγωγή

Οι πιο απλοί και πιο συνηθισμένοι μηχανισμοί κίνησης στα ρομπότ είναι οι ηλεκτρικοί κινητήρες. Ο ηλεκτρικός κινητήρας είναι μία ηλεκτρομηχανική συσκευή, η οποία χρησιμοποιείται για την μετατροπή της ηλεκτρικής ενέργειας σε κινητική. Η αρχή λειτουργίας του ηλεκτρικού κινητήρα είναι η δύναμη Lorentz η οποία λέει ότι, όταν ένας αγωγός διαρρέεται από ηλεκτρικό ρεύμα και βρίσκεται μέσα σε ένα μαγνητικό πεδίο τότε ασκείται πάνω στον αγωγό μία δύναμη η οποία είναι ίση με

$$\mathbf{F} = \mathbf{I} * \lambda * \mathbf{B}$$

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

Έτσι όταν το motor shield στείλει την κατάλληλη συνεχή τάση στον αγωγό του ηλεκτροκινητήρα τότε αυτός θα αρχίζει να περιστρέφεται. Ανάλογα με την διαφορά του δυναμικού που θα παρουσιάζεται στους ακροδέκτες των κινητήρων, αυτοί θα έχουν την δυνατότητα να περιστρέφονται δεξιόστροφα ή αριστερόστροφα. Έτσι το όχημα έχει την 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 μπροστά, όταν και οι δύο κινητήρες περιστρέφονται αριστερόστροφα.
- Κίνηση προς τα πίσω, όταν και οι δύο κινητήρες περιστρέφονται δεξιόστροφα.



Εικόνα 2.3.1.1: Κινητήρας DC

Η λειτουργία του κινητήρα βασίζεται και στη μαγνητική δύναμη μεταξύ ρευματοφόρων αγωγών. Η αλληλεπίδραση μεταξύ των ρευμάτων των αγωγών έχει σαν αποτέλεσμα την εμφάνιση μαγνητικής δύναμης και επομένως και ροπής. Με αυτόν τον τρόπο προκαλείται η περιστροφή του άξονα του κινητήρα. Η αποδιδόμενη ισχύς P ενός κινητήρα (το έργο που παράγεται από τον στρόφαλο (άξονα), ανά μονάδα χρόνου) ισούται με το γινόμενο της ροπής τ επί της γωνιακής ταχύτητας ω του άξονα, δηλαδή:

$$P = \omega * \tau$$

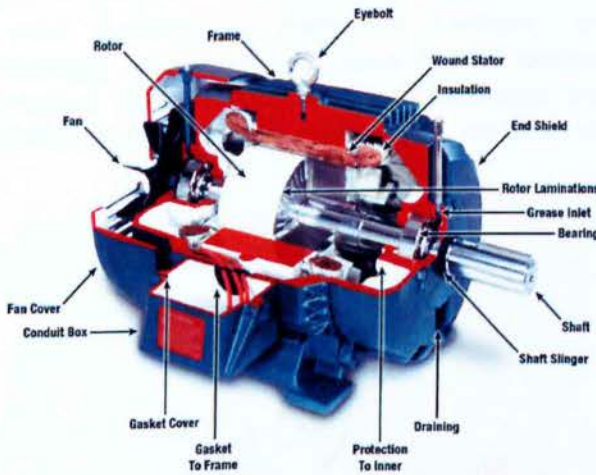
Συνήθως, η γωνιακή ταχύτητα ενός κινητήρα εκφράζεται σε στροφές n ανά μονάδα χρόνου. Άρα η αποδιδόμενη ισχύς σε HP είναι :

$$P = \tau * n / K$$

όπου $K = 1008400$, αν χρησιμοποιούμε για μονάδες μέτρησης oz, in, min , ή $K = 7600$, αν χρησιμοποιούμε Kp, m, sec , ενώ η αποδιδόμενη ισχύς σε kW είναι επίσης :

$$P = \tau * n / K$$

όπου $K = 9550$, αν χρησιμοποιούμε ως μονάδες N, m και min .



Εικόνα 2.3.1.2: Το εσωτερικό ενός ηλεκτροκινητήρα

Ο κάθε κινητήρας μπορεί να αποδώσει έως μία δεδομένη μέγιστη ισχύ, η οποία και ορίζεται από τις τεχνικές προδιαγραφές του. Έτσι όταν το motor shield στείλει την κατάλληλη συνεχή τάση στον αγωγό του ηλεκτροκινητήρα τότε αυτός θα αρχίζει να περιστρέφεται.

Δυστυχώς οι απλοί ηλεκτροκινητήρες δεν δίνουν την δυνατότητα στο όχημα να αλλάξει κατεύθυνση προς τα δεξιά ή αριστερά ενώ βρίσκεται σε κίνηση. Αυτό οφείλεται στο ότι οι απλοί ηλεκτροκινητήρες έχουν την δυνατότητα να περιστρέφονται μόνο με μία σταθερή ταχύτητα ανά λεπτό. Σε αντίθεση οι βηματικοί κινητήρες και οι servo κινητήρες χρησιμοποιώντας ειδικές βιβλιοθήκες του arduino όπως είναι η stepper.h και η servo.h έχουν την δυνατότητα να ρυθμίσουν τις στροφές ανά λεπτό με σκοπό τον καλύτερο έλεγχο του οχήματος.

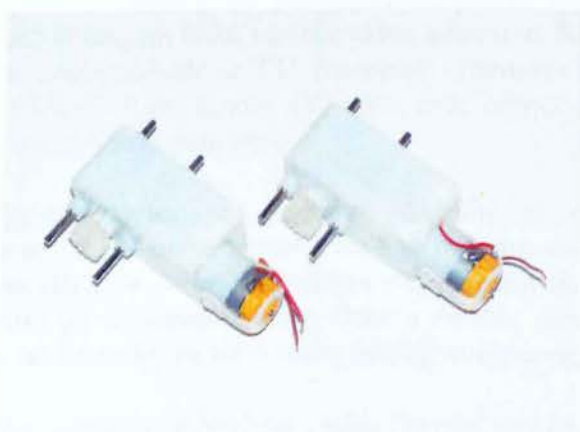
Έτσι για παράδειγμα όταν και οι δύο κινητήρες περιστρέφονται προς την ίδια φορά αλλά με μία διαφορετική ταχύτητα τότε το όχημα θα αναγκαστεί να πραγματοποιήσει μία κίνηση κυκλική και όχι ευθεία, όπου η ακτίνα της θα είναι ανάλογη της διαφοράς των στροφών των κινητήρων. Το μοναδικό μειονέκτημα είναι ότι αυξάνεται κατά πολύ το κόστος της κατασκευής εξαιτίας του ότι οι servo κινητήρες και οι βηματικοί κινητήρες είναι πολύ πιο ακριβοί ενώ απαιτούν και πιο πολύπλοκα συστήματα οδήγησης.

2.3.2 Κινητήρας DC

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

Όσον 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 υπάρξει μεγάλη επιβάρυνση της μπαταρίας.

Οι κινητήρες που χρησιμοποιήθηκαν για την κίνηση του οχήματος είναι συνεχούς τάσης με κιβώτιο ταχυτήτων. Ο ρόλος του κιβωτίου ταχυτήτων είναι να πολλαπλασιάζει κατά βούληση τη ροπή του κινητήρα, ώστε ανάλογα να επιτυγχάνεται καλύτερη επιτάχυνση ή μεγαλύτερη ταχύτητα κίνησης. Οι συγκεκριμένοι κινητήρες είναι ενδεδειγμένοι ως μηχανισμοί κίνησης τροχών, σε ρομποτικές εφαρμογές όπου απαιτείται η μεταφορά φορτίου με σχετικά μεγάλο βάρος.



Εικόνα 2.3.2.1: Κινητήρες DC με κιβώτιο ταχυτήτων

Στη συνέχεια παρουσιάζονται αναλυτικά τα τεχνικά χαρακτηριστικά των κινητήρων που χρησιμοποιήθηκαν για την κίνηση του ρομπότ.

Τεχνικά Χαρακτηριστικά:

Σχέση μετάδοσης : 100:1

Ονομαστική ταχύτητα περιστροφής : 140 rpm

Ένταση λειτουργίας : 60 mA

Τάση λειτουργίας : 6V

Τύπος κινητήρων : συνεχούς ρεύματος

2.3.3 Οδήγηση κινητήρων

Για την οδήγηση των δυο κινητήρων του οχήματος, χρησιμοποιήθηκε ο μικροελεγκτής SN754410 H-Bridge Motor. Είναι ένας τετραπλός οδηγός υψηλής τάσης (τύπου H) που είναι σχεδιασμένος να παρέχει αμφίδρομα ρεύματα πάνω από 1A και τάσεις από 4.5V ως 36V.



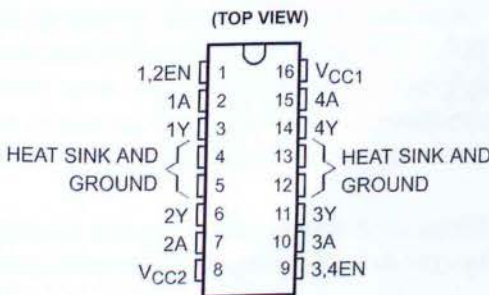
Εικόνα 2.3.3.2: μικροελεγκτής SN754410 H-Bridge

Η συσκευή είναι σχεδιασμένη να μεταφέρει επαγωγικά φορτία σε ρελέ και διπολικούς κινητήρες καθώς και άλλα υψηλής τάσης φορτία σε διάφορες εφαρμογές. Όλες οι εισόδους του είναι συμβατές με TTL (transistor - transistor logic) και χαμηλού επιπέδου λογικής CMOS. Κάθε έξοδος (Y) είναι ένας οδηγός με μια αντίσταση Darlington και μια ψευδό-Darlington πηγή.

Οι οδηγοί ενεργοποιούνται σε ζευγάρια. Δηλαδή οι οδηγοί 1 και 2 ενεργοποιούνται με το 1,2 EN και οι οδηγοί 3 και 4 να ενεργοποιούνται με το 3,4 EN. Όταν η είσοδος είναι HIGH οι αντίστοιχοι οδηγοί ενεργοποιούνται και οι έξοδοι τους δουλεύουν συμφασικά με τις εισόδους τους. Όταν η είσοδος είναι LOW, οι οδηγοί απενεργοποιούνται και μπαίνουν σε κατάσταση υψηλής αντίστασης.

Με τις κατάλληλες εισόδους δεδομένων, κάθε ζευγάρι των οδηγών σχηματίζουν ένα πλήρες H (ή γέφυρα) το οποίο είναι κατάλληλο για την εγκατάσταση κινητήρων. Μια ξεχωριστή παροχή ρεύματος (VCC1) παρέχεται για την λογικές εισόδους και την ελαχιστοποίηση της επαγωγής του ρεύματος. Η παροχή (VCC2) χρησιμοποιείται για τα κυκλώματα εξόδου.

SN754410 QUADRUPLE HALF-H DRIVER



FUNCTION TABLE
(each driver)

INPUTS†		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high-level, L = low-level
X = irrelevant

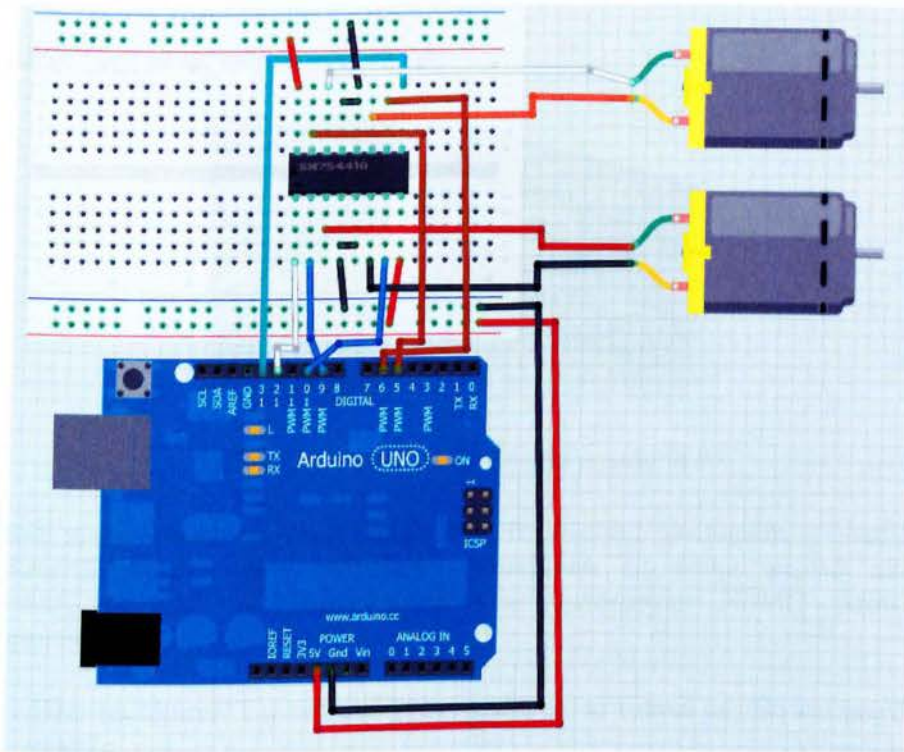
Z = high-impedance (off)

† In the thermal shutdown mode, the output is in a high-impedance state regardless of the input levels.

Εικόνα 2.3.3.1: Πίνακας λειτουργίας για κάθε οδηγό του H-Bridge Motor Driver

2.3.4 Σύνδεση με Arduino

Παρακάτω παρουσιάζετε η γραφική απεικόνιση των συνδέσεων των δύο κινητήρων με την πλακέτα του Arduino μέσω του μικροελεγκτή SN754410. Τα χρώματα των καλωδίων είναι όμοια με αυτά που χρησιμοποιήθηκαν στην κατασκευή του ρομπότ.



Εικόνα 2.3.4.1: Συνδεσμολογία πλακέτας Arduino με κινητήρες DC

2.4 Μπαταρίες

Για την τροφοδοσία της κατασκευής χρησιμοποιήθηκαν επαναφορτιζόμενες μπαταρίες Νικελίου – Υδριδίου Μετάλλου (Ni-MH). Οι συγκεκριμένες μπαταρίες αποτελούν τις πιο δημοφιλείς επαναφορτιζόμενες μπαταρίες, και έχουν αντικαταστήσει τις μπαταρίες Νικελίου – Καδμίου (Ni-Cd) καθώς έχουν μεγαλύτερη πυκνότητα ενέργειας. Παράγονται σε τυπικά μεγέθη AAA, AA κτλ.

Θεωρούνται εν γένει καλοί αντικαταστάτες των απλών αλκαλικών μπαταριών σε πολλές περιπτώσεις. Οι κυψέλες των μπαταριών αυτών είναι της τάξεως των 8,7V. Επίσης επαναφορτίζεται πλήρως έως και 1.000 φορές, χωρίς να απαιτείται πλήρης αποφόρτιση πριν την επαναφόρτιση. Τέλος, με την τεχνολογία νικελίου-υδριδίου μετάλλου οι μπαταρίες μπορούν να επαναφορτιστούν πολλές φορές, και το γεγονός αυτό καθιστά σχεδόν περιττή την αντικατάστασή τους.

Η μπαταρία που χρησιμοποιήθηκε είναι μεγέθους 9V και χωρητικότητας 200 mAh και φαίνεται στην επόμενη εικόνα, ενώ οι τεχνικές προδιαγραφές της παρουσιάζονται παρακάτω.

Τεχνικές Προδιαγραφές.

Τάση μπαταρίας : 9V

Χωρητικότητα : 200 mAh

Χημική σύνθεση : Νικελίου-υδριδίου μετάλλου

Διάρκεια ζωής εκτός λειτουργίας : 2 έτη



Εικόνα 2.4.1: Επαναφορτιζόμενη μπαταρία 9V

Όπως προανέφερα υπάρχουν δύο βασικές τεχνολογίες μπαταριών, η Νικελίου – Υδριδίου Μετάλλου και η Νικελίου – Καδμίου. Ας δούμε μερικά από τα πλεονεκτήματα και τα μειονεκτήματα των μπαταριών Ni-MH όπου και χρησιμοποιούνται.

- Πλεονεκτήματα: Έχουν υψηλή χωρητικότητα σε τυπικά μεγέθη και φορτίζονται εύκολα
- Μειονεκτήματα: Ακριβότερες από τις Ni-Cd, αποφορτίζονται πολύ γρήγορα σε κατάσταση αδράνειας και δεν μπορούν να χρησιμοποιηθούν σε συστήματα με μεγάλες απαιτήσεις ενέργειας.

Η επιλογή της τροφοδοσίας έγινε με βάση τα datasheet του Arduino, όπου απαιτείται τάση εισόδου από 7V ως 20V DC. Προτεινόμενη τροφοδοσία είναι από 9V ως 12V. Για λόγους οικονομίας, όμως, επιλέγουμε την χαμηλότερη δυνατή τροφοδοσία.

Επιδιώκουμε, λοιπόν, τροφοδοσία της τάξης των 9V, και για τον λόγο αυτό χρησιμοποιήθηκε 1 μπαταρία 9V τύπου Ni-MH. Παρακάτω παρουσιάζεται ένας φορτιστής για συγκεκριμένου τύπου μπαταρίες.



Εικόνα 2.4.2: Φορτιστής μπαταριών 9V

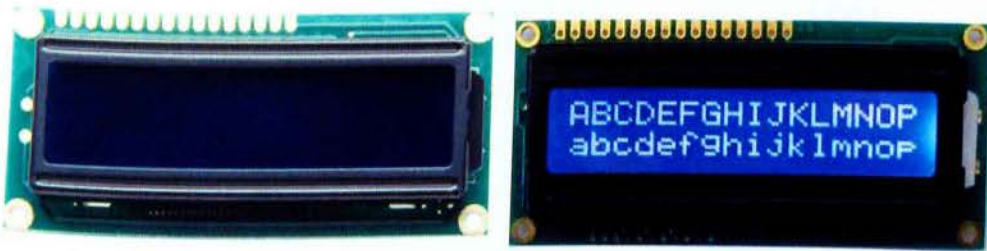
2.5 Οθόνη LCD

2.5.1 Εισαγωγή

Η οθόνη LCD (υγρών κρυστάλλων) είναι μια ηλεκτρονική μονάδα και την συναντάμε σε ένα ευρύ φάσμα εφαρμογών. Μια 16x2 LCD οθόνη είναι πολύ βασική μονάδα και χρησιμοποιείται πολύ συχνά σε διάφορες συσκευές και κυκλώματα.

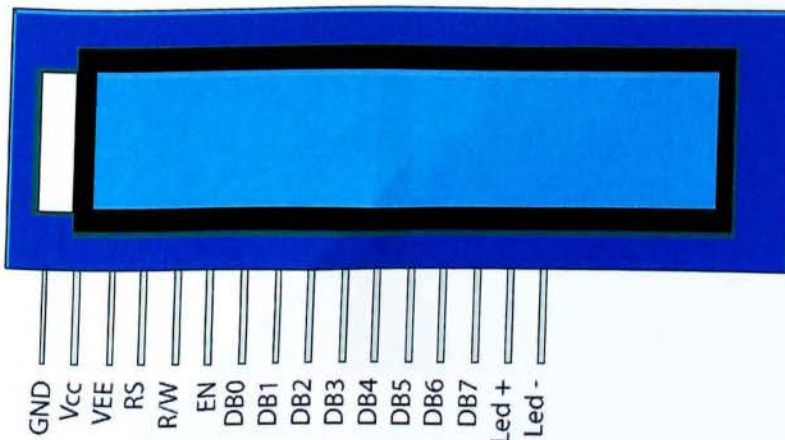
Αυτές οι μονάδες προτιμώνται έναντι των seven segment display και άλλων πολλών LEDs. Οι λόγοι που είναι τόσο δημοφιλής είναι διότι: οι οθόνες LCD είναι οικονομικές, εύκολα προγραμματιζόμενες, δεν έχουν περιορισμό στην εμφάνιση ειδικών και απλών χαρακτήρων (σε αντίθεση με τις seven segment display), εμφανίζουν κινούμενα σχέδια και ούτω καθεξής.

Μια οθόνη 16x2 LCD σημαίνει ότι μπορεί να εμφανίσει 16 χαρακτήρες ανά γραμμή και υπάρχουν 2 τέτοιες γραμμές. Σε αυτή την οθόνη LCD κάθε χαρακτήρας εμφανίζεται σε μήτρα 5x7 pixel. Αυτή η οθόνη LCD έχει δύο μητρώα, αυτά της διοίκησης και των δεδομένων.



Εικόνα 2.5.1.2 και 2.5.1.3: Οθόνη LCD 16x2 με μπλε φωτισμό απενεργοποιημένη και σε λειτουργία

Το μητρώο αποθηκεύει τις εντολές που δόθηκαν στην οθόνη LCD. Μέσω μιας εντολής η οθόνη LCD μπορεί να κάνει ένα προκαθορισμένο έργο όπως, μια αρχικοποίηση, την εκκαθάριση της οθόνης, την αλλαγή της θέσης του δρομέα, τον έλεγχο της οθόνης κλπ. Το μητρώο αποθηκεύει τα δεδομένα που θα εμφανίζονται στην οθόνη LCD. Τα δεδομένα είναι η τιμή ASCII του χαρακτήρα που πρόκειται να εμφανίζεται στην οθόνη LCD.



Εικόνα 2.5.1.4: Διάγραμμα pin οθόνης

Παρακάτω ακολουθεί ένας πίνακας με την περιγραφή της λειτουργίας του κάθε pin της οθόνης ξεχωριστά:

Lcd Pins	Λειτουργία	Όνομα
1	Ground (0V)	Γείωση
2	Τάση τροφοδοσίας 5V (4.7V - 5.3V)	Vcc
3	Ρύθμιση της αντίθεσης μέσω μιας μεταβλητής αντίστασης	V ₀
4	Επιλέγει καταχωρητή εντολών όταν δεν παίρνει τάση/ Επιλέγει καταχωρητή δεδομένων όταν παίρνει τάση	Επιλέξτε Εγγραφή
5	Χωρίς τάση γράφει στο μητρώο/ Με τάση διαβάζει από το μητρώο	Ανάγνωση / Εγγραφή
6	Στέλνει τα δεδομένα στις πινέζες δεδομένων, όταν αλλάζει κατάσταση ο παλμός, από υψηλός σε χαμηλό	Ενεργοποίηση
7	8-bit δεδομένων	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Τάση 5V οπίσθιου φωτισμού	Led +
16	Γείωση 0V οπίσθιου φωτισμού	Led-

Για την ολοκλήρωση της σύνδεσης, της οθόνης με την πλακέτα του Arduino χρησιμοποιήθηκε ένα ποτενσιόμετρο 1 K Ω , για την ρύθμιση της αντίθεσης της οθόνης LCD. Το ποτενσιόμετρο δεν είναι τίποτα άλλο παρά μια μεταβλητή αντίσταση. Αποτελείται από αγωγική πλάκα σχήματος Ω , πάνω στην οποία γυρίζει, με τη βοήθεια ενός τροφέα, μια επαφή. Ανάλογα με την απόσταση της επαφής από την είσοδο του ρεύματος στο ποτενσιόμετρο μεταβάλλεται και η αντίσταση.



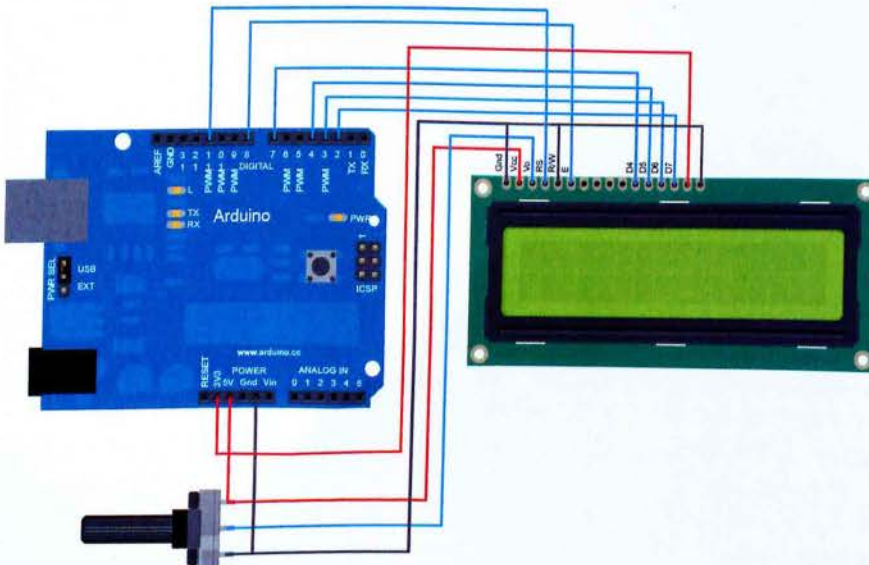
2.5.2.1: Ποτενσιόμετρο

2.5.2 Σύνδεση με Arduino

Παρακάτω παρατίθενται αναλυτικά ο πίνακας με τις συνδέσεις των pin της οθόνης LCD με τις ψηφιακές εισόδους/εξόδους του Arduino, καθώς και η γραφική απεικόνιση των συνδέσεων αυτών.

Lcd Pins	Arduino Digital inputs/outputs	Όνομα
1	Ground (0V)	Gnd
2	5V (4.7V - 5.3V)	Vcc
3	Ποτενσιόμετρο	Vo
4	Pin 11	Rs
5	Ground (0V)	R/W
6	Pin 8	E
7	Δεν χρησιμοποιείτε	DB0
8	Δεν χρησιμοποιείτε	DB1
9	Δεν χρησιμοποιείτε	DB2
10	Δεν χρησιμοποιείτε	DB3
11	Pin 7	DB4
12	Pin 4	DB5
13	Pin 3	DB6
14	Pin 2	DB7
15	3.3V	Led +
16	Ground (0V)	Led -

Παρακάτω παρουσιάζετε η γραφική απεικόνιση των συνδέσεων της πλακέτας του Arduino με την οθόνη LCD και το ποτενσιόμετρο.



Εικόνα 2.5.2.2: Συνδεσμολογία πλακέτας Arduino με οθόνη και ποτενσιόμετρο

2.6 Σενάριο λειτουργίας

Όπως έχουμε ήδη αναφέρει, η διαδικασία αποφυγής εμποδίων θα πραγματοποιηθεί με τη χρήση αισθητήρων εκπομπής και λήψης υπερήχων, η τεχνολογία των οποίων έχει παρουσιαστεί στην παράγραφο 2.2.

Οι πομποδέκτες υπερήχων Front και Rear, έχουν ως αρμοδιότητα τον εντοπισμό των εμποδίων. Όταν εντοπιστεί ένα, τότε το ρομπότ σταματάει πηγαίνει ελάχιστα όπισθεν και έπειτα στρίβει αριστερά. Στην περίπτωση που κατά την κίνηση του όπισθεν βρει εμπόδιο πίσω του, ο αισθητήρας Rear, τότε το ρομπότ κάνει περιστροφή μέχρι ο μπροστινός αισθητήρας να βρει κενό χώρο μπροστά του και να διαφύγει. Αν στη συνέχεια ο μπροστινός αισθητήρας δεν βρίσκει εμπόδια μπροστά του τότε το αυτοκινούμενο πηγαίνει μπροστά, στρίβει δεξιά, έπειτα αριστερά και αν σε όλη αυτή την κίνηση δεν εντοπιστεί εμπόδιο συνεχίζει την πορεία του ευθεία.



Εικόνα 2.6.1: Το αυτοκινούμενο ρομπότ σε λειτουργία

ΚΕΦΑΛΑΙΟ 3

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΟΥ ΜΙΚΡΟΕΛΕΓΚΤΗ ARDUINO

3.1 Αλγόριθμος κυρίου μέρους

3.1.1 Δήλωση μεταβλητών

```
const int motorLgnd=5;
const int motorLpwr=6;
const int motorRgnd=9;
const int motorRpwr=10;
const int leftEn=12;
const int rightEn=13;
const int sonarFrec=14;
const int sonarFtrig=15;
const int sonarRrec=16;
const int sonarRtrig=17;
```

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(11,8,7,4,3,2);
```

```
void setup(){
  pinMode(motorLpwr,OUTPUT);
  pinMode(motorLgnd,OUTPUT);
  pinMode(motorRpwr,OUTPUT);
  pinMode(motorRgnd,OUTPUT);
  pinMode(leftEn,OUTPUT);
  pinMode(rightEn,OUTPUT);
  pinMode(sonarFrec,INPUT);
  pinMode(sonarFtrig,OUTPUT);
  pinMode(sonarRrec,INPUT);
  pinMode(sonarRtrig,OUTPUT);
  Serial.begin(9600);
  Start();
}
```

Σε αυτή την παράγραφο πραγματοποιούμε τη δήλωση μεταβλητών. Πιο συγκεκριμένα δηλώνουμε τους ακροδέκτες και τα enable των κινητήρων (όπου motorLgnd είναι ο ακροδέκτης της γείωσης του αριστερού κινητήρα), όπως επίσης και τους ακροδέκτες των δύο αισθητήρων (όπου sonarFtrig είναι ο ακροδέκτης trig του μπροστινού αισθητήρα). Τα δηλώνουμε ως constant γιατί θέλουμε να έχουν σταθερή τιμή πάντα μέσα στο πρόγραμμα και να μην αλλάζουν παρά μόνο αν το θελήσουμε εμείς. Επίσης ο αριθμός κάθε μεταβλητής παριστάνει το ψηφιακό pin εξόδου που είναι το κάθε ένα συνδεδεμένο με το Arduino. Στη συνέχεια δηλώνουμε την οθόνη που είναι συνδεδεμένη στα pin 22,8,7,4,3,2 του Arduino. Τέλος στη συνάρτηση setup() θέτουμε τα pin του Arduino σε εξόδους, εκτός αυτών των sonarFrec και sonarRrec, ενεργοποιούμε την σειριακή επικοινωνία με την οθόνη του προγράμματος και ξεκινάμε το ρομπότ.

3.1.2 Κυρίως πρόγραμμα

```
if (sonarF()>20)
  Front(60+sonarF(),0.6);
if (sonarF()>20){
  Back(80,0.06);
  Stop(0.3);
  Right(80,0.55); }
if (sonarF()>20){
  Left(80,0.6);
  Stop(0.3);
  Left(80,0.55); }
if (sonarF()>20)
  Right(80,0.6);
else if(sonarF()<20 && sonarR()>20) {
  Back(80,0.6);
  Left(80,0.6); }
else if(sonarF()<20 && sonarR()<20)
  while (sonarF()>20) Left(80,0.6);
}
```

Στο κυρίως πρόγραμμα προγραμματίζετε η κύρια λειτουργία του αυτοκινούμενου. Ξεκινώντας εξετάζουμε εάν ο μπροστινός αισθητήρας είναι σε απόσταση μεγαλύτερη από 20 εκατοστά. Εάν αυτό αληθεύει τότε, αν ο αισθητήρας βρίσκει ακόμα σε απόσταση μεγαλύτερη από 20 εκατοστά πηγαίνει στιγμιαία πίσω, σταματάει για 0.3 milliseconds και στη συνέχεια πάει δεξιά με ταχύτητα 100 για χρόνο 0.55 milliseconds. Στη συνέχεια πραγματοποιείτε ξανά ο μπροστινός έλεγχος και στη συνέχεια στρίβει αριστερά, σταματά και ξαναπηγαίνει αριστερά. Συνεχίζοντας ξανακάνει μπροστινό έλεγχο και αν αληθεύει στρίβει δεξιά. Στην περίπτωση που κάποια από τις παραπάνω συνθήκες δεν αληθεύει τότε πηγαίνει πίσω και μετά αριστερά εάν βέβαια δεν υπάρχει εμπόδιο πίσω στα 20 εκατοστά. Στην αντίθετη περίπτωση στρίβει αριστερά μέχρις ότου ο μπροστινός αισθητήρας να βρει κενό μπροστά και συνεχίζει από την αρχή την παραπάνω διαδικασία.

3.2 Αλγόριθμος συναρτήσεων

3.2.1 Συναρτήσεις sonarF(), sonarR()

```
float sonarF(){
  digitalWrite(sonarFtrig,LOW);
  delayMicroseconds(2);
  digitalWrite(sonarFtrig,HIGH);
  delayMicroseconds(5);
  digitalWrite(sonarFtrig,LOW);
  return pulseIn(sonarFrec,HIGH)/58.7734;
}

float sonarR(){
  digitalWrite(sonarRtrig,LOW);
  delayMicroseconds(2);
  digitalWrite(sonarRtrig,HIGH);
  delayMicroseconds(5);
  digitalWrite(sonarRtrig,LOW);
  return pulseIn(sonarRrec,HIGH)/58.7734;
}
```

Οι συναρτήσεις `sonarF()` και `sonarR()` είναι πανομοιότυπες και για αυτό θα εξηγήσουμε την λειτουργία μιας εκ των δυο παραγράφων του προγράμματος. Επιλέγουμε την πρώτη από τις δύο παραγράφους. Στην πρώτη γραμμή του προγράμματος δηλώνουμε το `sonar` και το ορίζουμε ένα `float` αριθμό για τον λόγο ότι δέχεται όλους τους θετικούς αριθμούς. Στη συνέχεια θέτουμε το `pin trig` του `sonar` σε κατάσταση χαμηλή δηλαδή να αρχίζει χωρίς να στέλνει σήμα, να περιμένει 2 `microsecond` και εν συνεχεία να στέλνει ένα παλμό για 5 `microseconds` και να σταματάει. Τέλος η συνάρτηση επιστρέφει την απόσταση μεταξύ του `sonar` και του τοίχου. Ο αριθμός 58.7734 προκύπτει από τον πολλαπλασιασμό του 29, που είναι ο αριθμός των `microseconds` ανά εκατοστό (η ταχύτητα του ήχου είναι 340 m/s) επί το 2 που είναι η απόσταση που διανύει το σήμα για να πάει και να έρθει πάλι πίσω.

3.2.2 Συναρτήσεις `Start()`, `Stop()`

```
void Start(){
    digitalWrite(leftEn,HIGH);
    digitalWrite(rightEn,HIGH);
    lcd.clear();
    lcd.begin(16,2);
}
void Stop(float time){
    if(time>0){
        analogWrite(motorLpwr,0);
        analogWrite(motorLgnd,0);
        analogWrite(motorRpwr,0);
        analogWrite(motorRgnd,0);
        delay(time*1000);
    }
    else{
        analogWrite(motorLpwr,0);
        analogWrite(motorLgnd,0);
        analogWrite(motorRpwr,0);
        analogWrite(motorRgnd,0);
    }
}
```

Ξεκινώντας με την συνάρτηση `Start()`, ενεργοποιούμε τα `enable pin` των κινητήρων και εν συνεχεία καθαρίζουμε την οθόνη και αρχικοποιούμε τις γραμμές και τις στήλες της. Παρακάτω δηλώνουμε τη συνάρτηση `Stop()` η οποία δέχεται μια `float` τιμή στην παράμετρο `time`. Αν η τιμή αυτή είναι μεγαλύτερη από το 0 τότε σταματάνε και οι 2 κινητήρες για όσο χρονικό διάστημα έχει δοθεί στην παράμετρο `time`. Σε αντίθετη περίπτωση οι κινητήρες θα σταματήσουν για κάποια `miliseconds`.

3.2.3 Συνάρτηση Front()

```
void Front(int spd,float time){
  if(spd>255) spd=255;
  if(time>0){
    analogWrite(motorLpwr,spd);
    analogWrite(motorLgnd,0);
    analogWrite(motorRpwr,spd);
    analogWrite(motorRgnd,0);
    delay(time*1000);
  }
  else{
    analogWrite(motorLpwr,spd);
    analogWrite(motorLgnd,0);
    analogWrite(motorRpwr,spd);
    analogWrite(motorRgnd,0);
  }
}
```

Σε αυτή την παράγραφο δηλώνουμε τη συνάρτηση Front() βάζοντάς της δύο παραμέτρους, την παράμετρο spd για τον έλεγχο της ταχύτητας των κινητήρων και την παράμετρο time η οποία χρησιμεύει για τον χρόνο που θα κάνει delay το πρόγραμμα μια οποιαδήποτε εντολή κίνησης του έχουμε δώσει π.χ. πήγαινε μπροστά, πίσω, αριστερά, δεξιά ή σταμάτα για κάποιο συγκεκριμένο χρονικό διάστημα.

3.2.4 Συνάρτηση Back()

```
void Back(int spd,float time){
  lcd.clear();
  lcd.home();
  lcd.print("Rear:");
  lcd.print(sonarR());
  lcd.print(" cm");
  lcd.setCursor(0,1);
  lcd.print("Rear");
  lcd.noCursor();
  if(spd>255) spd=255;
  if(time>0){
    analogWrite(motorLpwr,0);
    analogWrite(motorLgnd,spd);
    analogWrite(motorRpwr,0);
    analogWrite(motorRgnd,spd);
    delay(time*1000);
  }
  else{
    analogWrite(motorLpwr,0);
    analogWrite(motorLgnd,spd);
    analogWrite(motorRpwr,0);
    analogWrite(motorRgnd,spd);
  }
}
```

Η συνάρτηση Back() όπως και η συναρτήσεις Front(), Left() και Right() είναι σχεδόν πανομοιότυπες μεταξύ τους. Η διαφορά τους έγκειται στο γεγονός ότι στη συνάρτηση Front() δεν χρησιμοποιήσαμε την οθόνη σε αντίθεση με τις άλλες τρεις συναρτήσεις στις οποίες εμφανίζουμε, αναλόγως με το τι κίνηση πραγματοποιεί κάθε

φορά το ρομπότ, αντίστοιχα τις λέξεις Rear, Left ή Right. Με αυτό τον τρόπο μπορούμε να καταλάβουμε τη λειτουργία και των υπολοίπων δύο συναρτήσεων.

3.2.5 Συνάρτηση Left()

```
void Left(int spd,float time){
  lcd.clear();
  lcd.home();
  lcd.print("Front:");
  lcd.print(sonarF());
  lcd.print(" cm");
  lcd.setCursor(0,1);
  lcd.print("Left");
  lcd.noCursor();
  if(spd>255) spd=255;
  if(time>0){
    analogWrite(motorLpwr,0);
    analogWrite(motorLgnd,spd);
    analogWrite(motorRpwr,spd);
    analogWrite(motorRgnd,0);
    delay(time*1000);
  }
  else{
    analogWrite(motorLpwr,0);
    analogWrite(motorLgnd,spd);
    analogWrite(motorRpwr,spd);
    analogWrite(motorRgnd,0);
  }
}
```

3.2.6 Συνάρτηση Right()

```
void Right(int spd,float time){
  lcd.clear();
  lcd.home();
  lcd.print("Front:");
  lcd.print(sonarF());
  lcd.print(" cm");
  lcd.setCursor(0,1);
  lcd.print("Right");
  lcd.noCursor();
  if(spd>255) spd=255;
  if(time>0){
    analogWrite(motorLpwr,spd);
    analogWrite(motorLgnd,0);
    analogWrite(motorRpwr,0);
    analogWrite(motorRgnd,spd);
    delay(time*1000);
  }
  else{
    analogWrite(motorLpwr,spd);
    analogWrite(motorLgnd,0);
    analogWrite(motorRpwr,0);
    analogWrite(motorRgnd,spd);
  }
}
```

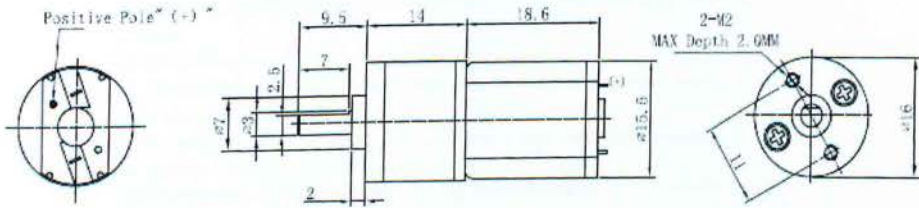
ΚΕΦΑΛΑΙΟ 4

4.1 Βιβλιογραφία

1. <http://www.arduino.cc/>
2. <http://grobotronics.com>
3. <http://www.frc.ri.cmu.edu>
4. <http://users.sch.gr>
5. <https://el.wikipedia.org>
6. <http://users.sch.gr/>
7. <http://www.piperidis.co.nr/>
8. <http://www.engineersgarage.com>
9. <http://deltahacker.gr>
10. <http://itp.nyu.edu/physcomp/>
11. <http://cyberneticzoo.com>
12. <http://www.grobot.gr/>
13. <http://www.forbes.com/>
14. <http://www.mechatronics.gr/>
15. <http://www.datasheetcatalog.com/>
16. <http://fritzing.org/>
17. <http://davidbuckley.net>
18. <http://www.stanford.edu>
19. <http://www.computerhistory.org>
20. <http://www.computerhistory.org>

4.2 Datasheet

OUTLINE



1. STANDARD OPERATING CONDITIONS:

- 1.1 Rated Voltage; DC 6V
- 1.2 Operating Temperature: $-10 \sim +60$ °C
- 1.3 Direction of Rotation:
Counter Clockwise when viewing motor output side with positive voltage applied to positive (red) terminal.

2. MEASURING CONDITIONS:

- 2.1 Rated Voltage; DC 6V
- 2.2 Environmental temperature: 21 °C
- 2.3 Environmental humidity: 65%RH
- 2.4 Motor Position: Motor to be held with shaft horizontally

3. ELECTRICAL CHARACTERISTICS

(At initial stage after 30 seconds run-in)

- 3.1 No Load
 - 3.1.1 No Load Speed: $147.6 \pm 10\%$ rpm
 - 3.1.2 No load Current: 0.06A (0.12A max)
- 3.2 On Load
 - 3.2.1 Rated Torque: 0.3kg.cm
 - 3.2.2 Rated Current: 0.19A (0.285A max)
 - 3.2.3 Rated Speed: $118.8 \pm 10\%$ rpm

4. MECHANICAL CHARACTERISTICS:

- 4.1 End Play: ≤ 0.4 mm
- 4.2 Dimension crawling: See <OUTLINE>

5. ASSEMBLING MOTOR:

030 6V 15000 $\pm 10\%$ rpm

PERFORMANCE CURVE

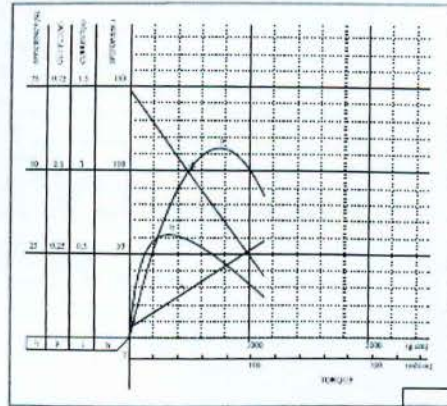
MODEL: EM46A2304R0600150

VOLTAGE: 6 VDC Constant

P/C NO: _____

DATE: 06-11-2010

REMARKS: _____



Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) If the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2.

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

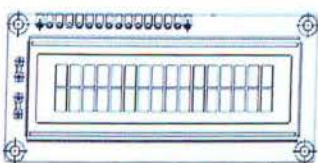
Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
Measuring Angle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

16 x 2 Character LCD



FEATURES

- 5 x 8 dots with cursor
- Built-in controller (KS 0066 or Equivalent)
- + 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A,K (LED)
- N.V. optional for + 3V power supply

MECHANICAL DATA

ITEM	STANDARD VALUE	UNIT
Module Dimension	80.0 x 36.0	mm
Viewing Area	66.0 x 16.0	mm
Dot Size	0.56 x 0.66	mm
Character Size	2.96 x 5.56	mm

ABSOLUTE MAXIMUM RATING

ITEM	SYMBOL	STANDARD VALUE			UNIT
		MIN.	TYP.	MAX.	
Power Supply	VDD-VSS	-0.3	-	7.0	V
Input Voltage	VI	-0.3	-	VDD	V

NOTE: VSS = 0 Volt, VDD = 5.0 Volt

ELECTRICAL SPECIFICATIONS

ITEM	SYMBOL	CONDITION	STANDARD VALUE			UNIT	
			MIN.	TYP.	MAX.		
Input Voltage	VDD	VDD = +5V	4.7	5.0	5.3	V	
		VDD = +3V	2.7	3.0	5.3	V	
Supply Current	IDD	VDD = 5V	-	1.2	3.0	mA	
Recommended LC Driving Voltage for Normal Temp. Version Module	VDD - V0	-20 °C	-	-	-	V	
		0 °C	4.2	4.8	5.1		
		25 °C	3.8	4.2	4.6		
		50 °C	3.5	4.0	4.4		
		70 °C	-	-	-		
LED Forward Voltage	VF	25 °C	-	4.2	4.6	V	
LED Forward Current	IF	25 °C	Array	-	130	260	mA
			Edge	-	20	40	
EL Power Supply Current	IEL	Vol = 110VAC;400Hz	-	-	5.0	mA	

BIBΛΙΟΘΗΚΗ
ΤΕΙ ΠΕΙΡΑΙΑ