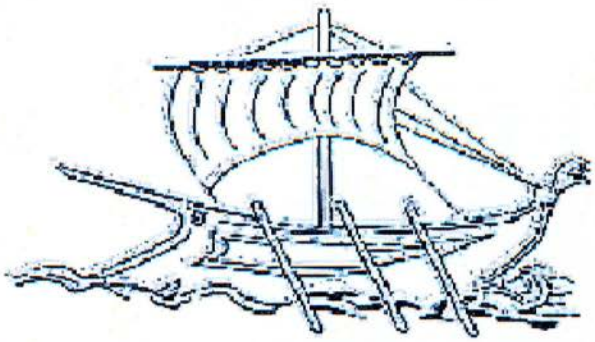


ΑΤΕΙ ΠΕΙΡΑΙΑ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Χρήση Ρομποτικού Οχήματος για Προσδιορισμό Αντικειμένων στον Χώρο

Παναγιώτης Αθανασίου

Εισηγητής: Ιωάννης Έλληνας

ΑΙΓΑΛΕΩ

ΝΟΕΜΒΡΙΟΣ 2013

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Χρήση Ρομποτικού Οχήματος για Προσδιορισμό Αντικειμένων στον Χώρο

Παναγιώτης Αθανασίου

Α.Μ. 38095

Εισηγητής:
Ιωάννης Έλληνας

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό των μικροελεγκτών και πώς οι εφαρμογές τους εξυπηρετούν την καθημερινότητα σε πολλούς τομείς.

Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου, τον οποίο θα ήθελα να ευχαριστήσω.

Ακόμα θα ήθελα να ευχαριστήσω τον κ. Δράμπαλο για τις πολύτιμες συμβουλές του καθώς και τον συμφοιτητή μου, Βασίλη Μπραντίνο για την συνεχή βοήθεια του.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου που θα ήθελε να τελειώσω τις σπουδές μου σε λιγότερο από πέντε χρόνια.

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία ασχολείται με την ανάπτυξη, σχεδιασμό, κατασκευή και προγραμματισμό ενος ρομποτικού οχήματος με σκοπό την ανάδειξη των οφελών που προκείπτουν απο την χρήση τέτοιου είδους τεχνολογίας. Το ρομπότ θα εχει την ικανότητα να μεταδίδει ζωντανό βίντεο στον τελικό χρήστη και θα ειναι σε θέση να ανιχνεύει αντικείμενα στο περιβάλλοντα χώρο, να εστιάζει την κάμερα πάνω σε αυτά μετρώντας παράλληλα και την αποστασή τους απο αυτό. Επιπρόσθετα, θα μπορεί να ανιχνεύσει διαφορετικές πηγές υπέρυθρης ακτινοβολίας κατα επιλογή του χρήστη.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Επιστήμη Ηλεκτρονικών Υπολογιστών και Ρομποτική

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Bluetooth, Arduino, Robot, Sensor, Processing

ΠΕΡΙΕΧΟΜΕΝΑ

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

dc = direct current = συνεχές ρεύμα

ir = infrared = υπέρυθρες

hex = hexadecimal = δεκαεξαδικός

ms = millisecond = χιλιοστά του δευτερολέπτου

nm = nanometers = νανόμετρα

rpm = revolutions per minute = στροφές ανά λεπτό

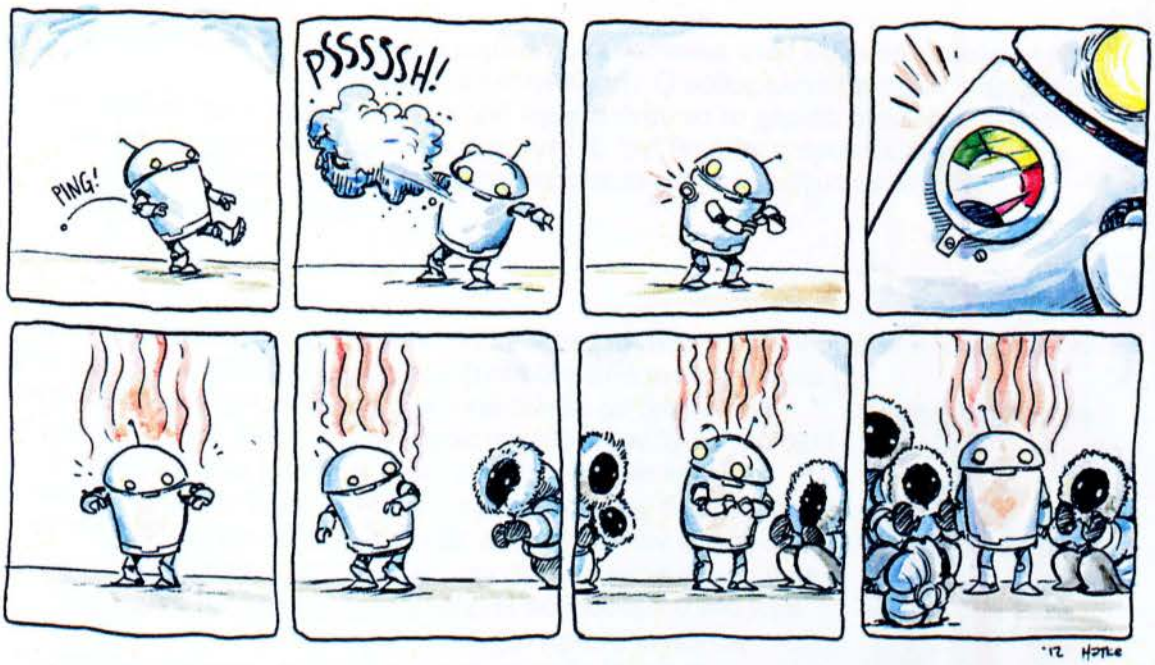
ide = integrated development environment = ολοκληρωμένο περιβάλλον ανάπτυξης

gnd = ground = γείωση

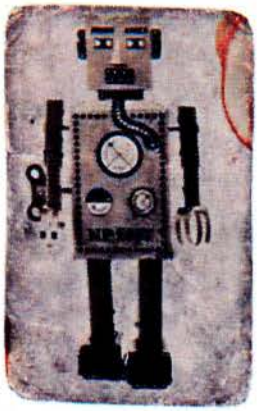
v = voltage = τάση

1.1 Εισαγωγή

Σε αυτό το κεφάλαιο γίνεται μια αναδρομή στο παρελθόν και στην ιστορία των ρομπότ και πώς επηρέασαν την ζωή μας απο τότε μέχρι σήμερα.



1.2 Ιστορική αναδρομή - Ρομπότ στην Αρχαιότητα



Η ρομποτική είναι μία από τις πιο σύγχρονες επιστήμες. Κι αυτό, γιατί δεν θα μπορούσε να αναπτυχθεί χωρίς την ύπαρξη των ηλεκτρονικών υπολογιστών και της υψηλής τεχνολογίας. Παρόλα αυτά, η ιδέα ενός μηχανήματος που θα μπορούσε να αντικαταστήσει τον άνθρωπο σε διάφορες εργασίες είναι πιο παλιά από ό,τι φαντάζεστε. Οι ρίζες της βρίσκονται στα βάθη των αρχαίων χρόνων!

Το πρώτο ρομπότ το συναντούμε στην ελληνική μυθολογία και συγκεκριμένα στην Κρήτη. Ο μύθος έλεγε ότι ο Δίας παρήγγειλε στον Ήφαιστο να κατασκευάσει ένα "ον" προκειμένου να το χαρίσει στον Μίνωα, βασιλιά της Κρήτης. Το έργο που θα επιτελούσε αυτό το "ον" θα ήταν η προστασία τής Κρήτης. Έτσι λοιπόν ο Ήφαιστος πήρε χαλκό και κασσίτερο και με το μείγμα έφτιαξε ένα μπρούτζινο ανθρωπόμορφο "ον".

Για να του δώσει όμως ζωή, ώστε να επιτελέσει το σκοπό για τον οποίον προοριζόταν, ο Ήφαιστος του έβαλε μία φλέβα που ξεκινούσε από τον αυχένα και κατέληγε στους αστραγάλους, όπου και έκλεινε με ένα χάλκινο καρφί. Μέσα στη φλέβα αυτή ο Ήφαιστος έβαλε τον "ιχώρ", το αίμα των αθανάτων. Και εγένετο Τάλως, που στην αρχαία κρητική διάλεκτο σημαίνει "ήλιος". Ο Τάλως προσεφέρθη ως δώρο από τον Δία στον Μίνωα και αμέσως ανέλαβε υπηρεσία. Ήταν ψηλός 30 μέτρα, βάδιζε με ταχύτητα 245 χλμ. την ώρα και μέσα στην ημέρα έκανε το γύρο της Κρήτης τρεις φορές. Η παρουσία του στο νησί είχε διπλό στόχο. Αφ΄ενός προστάτευε την Κρήτη, πετώντας τεράστιους βράχους σε όποιο πλοίο προσπαθούσε να προσεγγίσει και όποιοι κατάφερναν να αποβιβασθούν, τους έσφιγγε στα χέρια του και τους έκαιγε. Όποιος κάτοικος της Κρήτης συναντιώταν μαζί του, για να μην θεωρηθή εχθρός, έκανε το σινιάλο-σύνθημα: σήκωνε το χέρι του με ανοικτά το μικρό του δάχτυλο και το δείκτη, σχηματίζοντας το κερασφόρο χέρι (δύο κέρατα). Αφ΄ετέρου, ο Τάλως κρατούσε στα χέρια του τις πλάκες με τους νόμους του Μίνωα και φρόντιζε για την εφαρμογή τους. Κάτω λοιπόν από τη διαρκή προστασία του Τάλω, οι μινωικοί Κρήτες, λαός κατ' εξοχήν ειρηνικός, μπορούσαν να ασχοληθούν απερίσπαστοι με τις καθημερινές τους ασχολίες. Κάποια ημέρα στον ορίζοντα φάνηκε ένα καράβι.

Ο Τάλως ετοιμάστηκε να υπερασπισθεί το έδαφος του νησιού. Το καράβι πλησιάζει, είναι η "Αργώ" με τους αργοναύτες, τον Ιάσονα και τη Μήδεια. Ο Τάλως αρχίζει να εκσφενδονίζει βράχους εναντίον τους και τότε η Μήδεια κάνει τα μάγια της. Ο Τάλως χάνει "τα λογικά του", αρχίζει να τρέχει στα βράχια, ο αστράγαλός του χτυπάει κάπου, το καρφί φεύγει από τη θέση του και ο "ιχώρ όμοιος με αναλυτό (λιωμένο) μολύβι" χύνεται στο έδαφος. Ο Τάλως, νεκρός πια, βυθίζεται στη θάλασσα.

Στα ιατρικά λεξικά, "ιχώρ" λέγεται το δύσοσμο και υποπράσινο υγρό που παράγεται κατά τη νέκρωση και αποσύνθεση των ιστών. Δύσοσμοι όμως είναι και οι ατμοί του θειικού οξέως (H2S04), το οποίο χρησιμοποιείται στην κατασκευή συσσωρευτών. Μία στήλη-συσσωρευτής θα μπορούσε να είχε τοποθετηθή, από τον αυχένα μέχρι τον αστράγαλο. Σύμφωνα με τον Απολλόδωρο, κάποιοι περιέγραφαν τον Τάλω ως "ταύρο με δύο "κέρατα" (μήπως κεραίες;)! Ίσως αυτές οι κεραίες να βοηθούσαν τον Μίνωα να δίνει τις εντολές του στον Τάλω. Είναι πιθανόν.

Στην πόλη της Φαιστού βρέθηκαν νομίσματα στα οποία εικονίζεται ο Τάλως.

Είναι πολύ πιθανό, αυτός ο μύθος να αποτέλεσε την έμπνευση για πολλούς μηχανικούς της αρχαιότητας, που προσπάθησαν να κατασκευάσουν «έξυπνες» μηχανές, που θα έλυναν τα χέρια των ανθρώπων της εποχής. Μία από τις πρώτες τέτοιες μηχανές θεωρείται η πετομηχανή, ένα ιπτάμενο αντικείμενο -στο οποίο βέβαια δεν έμπαιναν άνθρωποι- που μπορούσε να διανύσει μέχρι και απόσταση 200 μέτρων.

Αρκετά χρόνια μετά, κάπου στο 1.200 μ.Χ., ο Άραβας Αλ Τζαζάρι δημιούργησε το πρώτο ρομπότ με μορφή ανθρώπου. Το ρόμποτ αυτο επαιζε μουσική και συγκεκριμένα, τύμπανο.

Δυστυχώς τα ακριβή σχέδια του Αλ Τζαζάρι δεν έχουν σωθεί στις μέρες μας. Έχουμε όμως τα σχέδια ενός άλλου σπουδαίου καλλιτέχνη και μηχανικού, του Λεονάρντο Ντα Βίντσι. Ο Ντα Βίντσι (1452-1519) σχεδίασε ένα ανθρωπόμορφο ρομπότ που έμοιαζε με ιππότη με πανοπλία και μπορούσε να κουνά τα χέρια και το κεφάλι του.

Σε σχέση με τα ρομπότ της σημερινής εποχής, οι μηχανές της αρχαιότητας φαντάζουν παιδικές. Σίγουρα, όμως, ήταν η έμπνευση για τις σπουδαίες εφευρέσεις που έχουν ήδη γίνει, αλλά και γι' αυτές που θα γίνουν στο μέλλον.

1.3 Η ιστορία των ρομπότ

Η επιστήμη της ρομποτικής είναι μία νέα επιστήμη, που αναπτύσσεται όμως με ραγδαίους ρυθμούς. Έτσι, τα πρώτα σύγχρονα ρομπότ, που φτιάχτηκαν κάπου στα 1950, δεν έχουν καμία σχέση με τα ρομπότ που χρησιμοποιούμε στις μέρες μας και σίγουρα, ούτε με αυτά που θα έχουμε στο μέλλον.

Αξίζει όμως να γνωρίσουμε τα ρομπότ των προηγούμενων δεκαετιών. Κάθε ένα από αυτά ήταν η έμπνευση αλλά και μία πολύτιμη πηγή πληροφοριών, ώστε να κατασκευαστούν ακόμα πιο εξελιγμένα μοντέλα που θα μπορούσαν να πραγματοποιήσουν ακόμα πιο πολύπλοκες εργασίες.

UNIMATE 1961 («Γιούνιμεϊτ»):

Πρόκειται για το πρώτο ρομπότ που χρησιμοποιήθηκε στη βιομηχανία και συγκεκριμένα, στην κατασκευή αυτοκινήτων. Το Unimate μετέφερε σιδερένια εξαρτήματα στο σκελετό του αυτοκινήτου που κατασκευαζόταν. Ήταν μία εργασία πολύ επικίνδυνη για τους ανθρώπους, καθώς εισέπνεαν τοξικά αέρια, ενώ μπορεί να τραυματίζονταν αν δεν ήταν αρκετά προσεκτικοί. Εύκολα, λοιπόν, καταλαβαίνει κανείς πόσο σημαντική ήταν αυτή η εφεύρεση για χιλιάδες εργάτες στις αυτοκινητοβιομηχανίες!

RANCHO ARM 1963 («Ράντσο αρμ»):

Το συγκεκριμένο ρομπότ είναι μία από τις πρώτες προσπάθειες των επιστημόνων του νοσοκομείου Λος Αμίγκος στην Καλιφόρνια να δημιουργήσουν τεχνητά μέλη για άτομα με ειδικές ανάγκες. Και τα κατάφεραν. Το Rancho Arm, χάρη στις 6 αρθρώσεις του, είχε ευελιξία όμοια με αυτή του ανθρώπινου χεριού και οι κινήσεις του ελέγχονταν μέσω ηλεκτρονικού υπολογιστή.

SHAKEY 1970 («Σέικι»):

Το Shakey είναι το πρώτο κινούμενο ρομπότ που μπορούσε να πραγματοποιεί λογικές διεργασίες. Δημιουργήθηκε από τους επιστήμονες του τεχνολογικού ινστιτούτου SRI (Stanford Research Institute) στις ΗΠΑ. Μπορούσε να μετακινείται στο χώρο και να αναγνωρίζει για ποιο λόγο κάνει μία ενέργεια. Η αποστολή του ήταν να ανοίγει και να κλείνει διακόπτες και πόρτες, και κυρίως να ξέρει πότε πρέπει να το κάνει και γιατί! Επίσης χρησιμοποιήθηκε και σαν τηλεοπτική κάμερα, αλλά και σαν αισθητήρας ακτίνων lazer.

DANTE I 1992 («Δάντης»):

Είναι το ρομπότ που επέτρεψε στον άνθρωπο να μελετήσει για πρώτη φορά από πολύ κοντά τον κρατήρα ενός ενεργού ηφαιστείου . Δημιουργήθηκε από την επιστημονική ομάδα του πανεπιστημίου Κάρνεγκι Μέλον στις ΗΠΑ. Αν και η πρώτη του αποστολή απέτυχε (στόχος ήταν η εξερεύνηση του ηφαιστείου στο βουνό Έρεβος στην Ανταρκτική), η δεύτερη στέφθηκε με επιτυχία. Ο Δάντης καταφέρε να μπει στον κρατήρα του ηφαιστείου του βουνού Σπουρ στην Αλάσκα και να συγκεντρώσει πολύτιμα ευρύματα. Στη μορφή θύμιζε αράχνη, ώστε να μπορεί να σκαρφαλώνει στις δύσβατες επιφάνειες ενός κρατήρα. Ο Dante ήταν εξοπλισμένος με αισθητήρες και υπερσύγχρονες κάμερες και μας έδωσε πολύτιμες πληροφορίες για ένα περιβάλλον, που ο άνθρωπος δεν θα μπορούσε να επισκεφθεί ποτέ, εξαιτίας των υψηλών θερμοκρασιών και των δύσκολων συνθηκών.

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο αναλύονται οι 3 λειτουργίες του ρομπότ καθώς και μία συνοπτική περιγραφή ορισμένων προβλημάτων στον λογισμικό που παρουσιάστηκαν κάτα την συγγραφή του κώδικα.

Το ρομποτ υποστηρίζει 3 λειτουργίες.

- Χειροκίνητη λειτουργία (Manual Mode)
- Λειτουργία για εντοπισμό αντικειμένων που περιβάλλουν το ρομπότ (Detection Mode)
- Λειτουργία για εντοπισμό υπέρυθρων πηγών ακτινοβολίας (Detection Mode)

2.2 Χειροκίνητη Λειτουργία:

Με αυτήν την λειτουργία επιτυγχάνεται ο χειρισμός του ρομπότ χειροκίνητα, μέσω ηλεκτρονικού υπολογιστή και με χρήση της ασύρματης τεχνολογίας bluetooth.

Σε αυτό το στάδιο δημιουργήθηκε και το πρόγραμμα με γραφικό περιβάλλον για τον χρήστη (GUI).

Σημείωση:

Σχετικά με το bluetooth...

Το κύκλωμα bluetooth παρέχει την δυνατότητα ασύρματης σειριακής επικοινωνίας μεταξύ συσκευών.

Το κύκλωμα του bluetooth που χρησιμοποιήθηκε έχει εμβέλεια 10 μέτρων (Class 2).

Ο χειρισμός του οχήματος θα πρέπει να γίνεται από απόσταση μικρότερη των 10 μέτρων, καθώς, σε αντίθετη περίπτωση η επικοινωνία με το όχημα μπορει να χαθεί.

Ο χρήστης μπορεί να κινήσει το ρομπότ προς όλες τις κατευθύνσεις, χρησιμοποιώντας συγκεκριμένα πλήκτρα από το πληκτρολόγιο του ηλεκτρονικού υπολογιστή. Παράλληλα μπορεί να ελέγξει την κατεύθυνση της κάμερας που φέρει το ρομπότ, με την χρήση ενός σερβοκινητήρα (servo), καθώς και την ταχύτητα του οχήματος.

Για να υπάρξει επικοινωνία μεταξύ του οχήματος και του ηλεκτρονικού υπολογιστή, προϋπόθεση είναι, ο ηλεκτρονικός υπολογιστής να ενσωματώνει κύκλωμα bluetooth.

Πρέπει να γίνει ταυτοποίηση των στοιχείων, απο την μεριά του ηλεκτρονικού υπολογιστή, εισάγοντας τον κατάλληλο κωδικό (PIN code), ώστε να επιτευχθεί η αντιστοίχιση και να δημιουργηθεί το ζεύγος οχήματος - ηλεκτρονικού υπολογιστή (paired)

Κατά την εκκίνηση του προγράμματος, γίνεται αυτόματα η σύνδεση του ρομπότ με τον εκάστοτε ηλεκτρονικό υπολογιστή. Απαιτείται κάποιο χρονικό περιθώριο (< 3 sec) για την εγκατάσταση της σύνδεσης με το όχημα.

Καθώς η σύνδεση με το ρομπότ πραγματοποιήθηκε, ο χρήστης είναι πλέον σε θέση, να ελέγξει το όχημα.

Ο χρήστης καλείται να πατήσει τα πλήκτρα λειτουργίας (Π.Λ) (καθορισμένα απο τον κατασκευαστή) ώστε να κινήσει το όχημα στο χώρο.

Καθώς ένα πλήκτρο (Π.Λ) πατιέται, δημιουργείται ένας ASCII κωδικός - συγκεκριμένος για κάθε πλήκτρο. Το πρόγραμμα συγκρατεί τον ASCII κωδικό αυτόν, και στην συνέχεια δημιουργεί έναν ακέραιο αριθμό - συγκεκριμένο για κάθε ASCII κωδικό.

Η ίδια διαδικασία γίνεται και για την απελευθέρωση κάθε πλήκτρου.

Τέλος, ο ακέραιος αυτός αριθμός που δημιουργήθηκε, αποστέλλεται μέσω του κυκλώματος bluetooth του ηλεκτρονικού υπολογιστή, στο κύκλωμα bluetooth του οχήματος, για επεξεργασία απο το ρομπότ.

2.2.1 Λειτουργία ελέγχου ταχύτητας του οχήματος

Για να επιτευχθεί κάτι τέτοιο, χρησιμοποιώντας έναν ρυθμιστή/σύρτη (Slider), αναγκαία κρίνεται η μέτρηση των συντεταγμένων του κέρσορα του ποντικιού (Mouse). Καθώς μετακινούμε το ρυθμιστή (Slider) - πάνω ή κάτω - αλλάζουν οι συντεταγμένες 'Y' του κέρσορα. Το πρόγραμμα συγκρατεί την τιμή των συντεταγμενών 'Y' και τις αποστέλλει μέσω του κυκλώματος bluetooth του ηλεκτρονικού υπολογιστή, στο κύκλωμα bluetooth του οχήματος, για επεξεργασία.

ΠΑΡΑΤΗΡΗΣΕΙΣ:

Το όχημα για την κίνηση των κινητήρων του, αντιλαμβάνεται τιμές 0 (ελάχιστο) - 255 (μέγιστο), όπου το '0' είναι η ακινητοποίηση των κινητήρων και '255' η μέγιστη δυνατή ταχύτητα στους κινητήρες του (μέγιστο rpm). Για αυτόν τον λόγο πραγματοποιείται μια αναλογική βαθμονόμηση (Calibration) σε σχέση με την αντίστοιχη τιμή που στάλθηκε από τον ηλεκτρονικό υπολογιστή με την χρήση του ρυθμιστή (Slider) ταχύτητας. Έτσι, όταν στέλνεται στο ρόμποτ η συντεταγμένη "Y' = 350 θα γίνει η κατάλληλη βαθμονόμισή της για να ανήκει στο διάστημα 0 - 255. Στην εν λόγω πτυχιακή εργασία, θα γίνει χρήση ως μέγιστης τιμής της ταχύτητας των κινητήρων, η τιμή '110' έναντι της '255', για λόγους που διευκρινίζονται στην ενότητα 3.14.

Η αποστολή δεδομένων για την κίνηση του ρομπότ στον χώρο, πρέπει να γίνεται μια φορά. Καθώς ο χρήστης πατά ένα πλήκρο (Π.Λ) για την κίνηση του οχήματος, η παρατετάμενη χρήση του (μόνιμα πατημένο) έχει σαν αποτέλεσμα απώλεια δεδομένων (εντολών). Αξίζει να σημειωθεί οτι για τον έλεγχο των δεδομένων εισόδου, προτιμήθηκε η χρήση της δομής επανάληψης 'while' έναντι της 'if', καθώς με την χρήση της δεύτερης, δεν υπήρχε σταθερή τάση στους κινητήρες. Η δομή επανάληψης 'if' ικανοποιούνταν μια φορά σε κάθε κύκλο του προγράμματος, ταχύτατα μεν, αλλά οι κινητήρες, είχαν μια περισσότερο σπασμωδική κίνηση αντι για μιάς πιο ομαλής (κίνησης) που προκαλεί η χρήση της δομής επανάληψης 'while'.

2.3 Λειτουργία Εντοπισμού Αντικειμένων που Περιβάλλουν το Ρομπότ (Detection Mode)

Η λειτουργία αυτή, ενεργοποιείται μέσα απο το γραφικό περιβάλλον χρήστη, πατώντας το κατάλληλο κουμπί με το ποντίκι (Mouse).

Σκοπός αυτής της λειτουργίας είναι η ανίχνευση αντικειμένων που βρίσκονται κοντά και περιμετρικά του οχήματος. Αυτό επιτυνχάνεται με την χρήση αισθητήρων υπερήχων.

Ο αισθητήρας υπερήχων εκπέμπει ανά τακτά χρονικά διαστήματα ένα σήμα - υψηλής συχνότητας ηχητικό κύμα - . Αυτό το σήμα ανακλάται στις επιφάνειες των αντικειμένων και ένα μέρος του, επιστρέφει πίσω στον αισθητήρα. Ο χρόνος που κάνει το σήμα απο την στιγμή που εκπέμφθηκε, μέχρι την επιστροφή του στον αισθητήρα, είναι καθοριστικός γιατί με αυτόν τον τρόπο μπορούμε να υπολογίσουμε την απόσταση του αντικειμένου από τον αισθητήρα.

Όταν ένας αισθητήρας εντοπίσει ένα αντικείμενο, το όχημα πρέπει να ανταποκριθεί και να στραφεί προς το αντικείμενο αυτό. Με αυτόν τον τρόπο η κάμερα που φέρει το ρομπότ στο μπροστινό μέρος του, θα εστιάσει στο αντικείμενο και θα αναμεταδώσει ζωντανό βίντεο και ήχο προς τον τελικό χρήστη.

Επιπλέον υπάρχει η δυνατότητα εμφάνισης της απόστασης του κοντινότερου αντικειμένου στον χρήστη, μέσω του γραφικού περιβάλλοντος χρήστη (GUI). Μόλις ανιχνευθεί το κοντινότερο, προς το ρομπότ, αντικείμενο, ο επεξεργαστής του ρομπότ στέλνει μέσω του κυκλώματος bluetooth την απόσταση του αντικειμένου στο πρόγραμμα, όπου και εμφανίζεται.

2.3.1 Λειτουργία ελέγχου απόστασης εντοπισμού αντικειμένων

Όπως προαναφέρθηκε, για να επιτευχθεί κάτι τέτοιο, γίνεται η χρήση ενός ρυθμιστή / σύρτη (Slider), ώστε να γίνει η μέτρηση των συντεταγμένων του κέρσορα του ποντικιού (Mouse). Καθώς μετακινούμε το ρυθμιστή (Slider) - δεξιά ή αριστερά - αλλάζουν οι συντεταγμένες 'X' του κέρσορα. Το πρόγραμμα συγκρατεί την τιμή των συντεταγμένων 'X' και τις αποστέλλει μόνο σε κάθε απελευθέρωση του αριστερού πλήκτρου του ποντικιού, μέσω του κυκλώματος bluetooth του ηλεκρονικού υπολογιστή στο κύκλωμα bluetooth του οχήματος, για επεξεργασία απο το ρομπότ.

Έτσι υπάρχει η δυνατότητα αυξομείωσης της μέγιστης απόστασης, μέσα στην οποία θα γίνεται η ανίχνευση των αντικειμένων από τους αισθητήρες.

2.3.2 Αναλυτικότερα

Το ρομπότ φέρει μια φωτοδίοδο πράσινου χρώματος (Green LED) , μία χρώματος μπλέ (Blue LED) και μια κόκκινου χρώματος (Red LED).

Η φωτοδίοδος πράσινου χρώματος (Green LED) υποδηλώνει οτι το ρομπότ βρίσκεται σε αυτόματη λειτουργία εντοπισμού αντικειμένων (Detection mode) και δεν έχει ανιχνεύσει κανένα αντικείμενο μέχρι στιγμής.

Η φωτοδίοδος μπλέ χρώματος (Blue LED) υποδηλώνει οτι ένα αντικείμενο ανιχνεύθηκε και οτι αρχίζει η περιστροφή του οχήματος, προς αυτό.

Η φωτοδίοδος κόκκινου χρώματος (Red LED) άναβει κάθε φορά που το όχημα λαμβάνει τον παλμό σύνδεσης απο το πρόγραμμα χρήστη.

2.3.3 Παλμός σύνδεσης

Το όχημα μπορεί και αναγνωρίζει πότε είναι συνδεδεμένο με το πρόγραμμα χρήστη.

Το πρόγραμμα χρήστη λαμβάνει κάθε 2 sec έναν παλμό άπο το όχημα, και έπειτα με την σειρά του, αποστέλλει στο όχημα έναν απαντητικό παλμό, επιτρέποντας στο ρομπότ να αναγνωρίζει οτι η σύνδεση δεν έχει χαθεί. Άμα η σύνδεση χαθεί, η OLED οθόνη του όχηματος θα εμφανίσει ένα σχετικό μήνυμα.

Το ρομπότ είναι προγραμματισμένο να τείθεται εκτός λειτουργίας άμα χαθεί η επικοινωνία του με το πρόγραμμα χρήστη, και να επαναφέρει όλες του τις ρυθμίσεις, στην αρχική τους κατάσταση.

ΠΑΡΑΤΗΡΗΣΕΙΣ:

Ο εντοπισμός αντικειμένων είναι μια αυτόματη λειτουργία του οχήματος.

Θα πρέπει να υπολογιστούν τα σενάρια εντοπισμού περισσότερων αντικειμένων από του ενός έτσι ώστε το ρόμποτ, να είναι σε θέση να αντιληφθεί το πιο κοντινό σε αυτό, αντικείμενο.

Με αυτόν τον τρόπο άμα υπάρχουν στον περιβάλλοντα χώρο πολλά αντικείμενα, ανιχνεύσιμα από τους αισθητήρες υπερήχων, το όχημα θα στραφεί πρός το αντικείμενο που είναι πιό κοντά προς αυτό, δηλαδή στο αντικείμενο που το σήμα έκανε λιγότερη ώρα να ανακλαστεί στην επιφάνεια του και να επιστρέψει πίσω στον αισθητήρα.

Θα πρέπει να γίνει κατανοητό οτι άμα ένας αισθητήρας διεγερθεί απο το ανακλώμενο σήμα και το ρομπότ αρχίσει να στρέφεται προς την κατεύθυνση αυτού του αισθητήρα, κατά την περιστροφή του στον χώρο, οι αισθητήρες αλλάζουν προσανατολισμό και τα αντικείμενα βρίσκονται σε διαφορετικές θέσεις ως πρός το όχημα.

Για την αποστολή των δεδομένων, με σκοπό τον καθορισμό της μέγιστης απόστασης εντοπισμού αντικειμένων χρησιμοποιούνται ακέραιοι αριθμοί. Στην προκειμένη περίπτωση, οι συντεταγμένες 'Y' του κέρσορα είναι μεταξύ των τιμών των συντεταγμένων 'X' που αποστέλλονται για τον καθορισμό της ταχύτητας το οχήματος. Για αυτόν τον λόγο γίνεται αναλογική βαθμονόμηση (Calibration) μέσα στο ίδιο το πρόγραμμα, για την αντίστοιχη τιμή 'X' που πρόκειται να αποσταλλεί, και εφόσον βαθμονομηθεί σε έναν ακέραιο που δεν ειναι μεταξύ των τιμών για τον έλεγχο της ταχύτητας, αποστέλλεται στο ρομπότ.

2.4 Λειτουργία Εντοπισμού Υπέρυθρων Πηγών Ακτινοβολίας (IR Mode)

Η λειτουργία αυτή, ενεργοποιείται μέσα απο το γραφικό περιβάλλον χρήστη, πατώντας το κατάλληλο κουμπί με το ποντίκι (Mouse).

Σκοπός αυτής της λειτουργίας ειναι η ανίχνευση πηγών υπέρυθρης ακτινοβολίας. Αυτό επιτυνχάνεται με την χρήση ενός αισθητήρα υπερύθρων.

Η μέγιστη απόσταση εντοπισμού είναι τα 400 cm. Το ρομπότ είναι σε θέση να αναγνωρίζει και να εντοπίζει μια συγκεκριμένη πηγή υπερύθρων κάθε φορά.

Καθώς ο αισθητήρας υπερύθρων είναι ευαίσθητος σε μεγάλο εύρος μήκος κύματος (700 nm – 1 mm) ο διαχωρισμός των πηγών υπερύθρων δεν είναι δυνατός με τον εντοπισμό του μήκους κύματος της ακτινοβολίας που εκμπέμπει κάθε πηγή.

Για αυτόν τον λόγο κάθε πηγή στέλνει έναν 16αδικό 4ψήφιο κωδικό (HEX), ανά τακτά χρονικά διαστήματα των 25ms . Το ρομπότ μπορεί να αναγνωρίζει αυτούς τους κωδικούς και κατ'επέκταση την ταυτότητα της κάθε πηγής. Αξίζει να σημειωθεί οτι η συχνότητα αναμετάδοσης θα πρέπει να είναι μεγαλύτερη των 25ms. Οποιαδήποτε αλλαγή στον ρυθμό αναμετάδοσης για τιμές < 25ms θα εχει σαν αποτέλεσμα να μην αναμεταδίδεται ο 16αδικός αριθμός που εχει οριστεί στο πρόγραμμα αλλά 1 byte πληροφορίας 'FFFFFFFF'.

Όταν ο αισθητήρας εντοπίσει την εκάστοτε πηγή, το όχημα πρέπει να ανταποκριθεί και να στραφεί προς το αντικείμενο αυτό. Με αυτόν τον τρόπο η κάμερα που φέρει το ρομπότ στο μπροστινό μέρος του, θα εστιάσει στο αντικείμενο και θα αναμεταδώσει ζωντανό βίντεο και ήχο προς τον τελικό χρήστη.

Επιπλέον υπάρχει η δυνατότητα εμφάνισης της απόστασης της πηγής απο το όχημα, στον χρήστη, μέσω του γραφικού περιβάλλοντος χρήστη (GUI). Μόλις ανιχνευθεί η πηγή υπερύθρων, ο επεξεργαστής του ρομπότ στέλνει μέσω του κυκλώματος bluetooth την απόσταση του αντικειμένου στο πρόγραμμα, όπου και εμφανίζεται.

2.4.1 Αναλυτικότερα

Όπως, προαναφέρθηκε στην ενότητα 2.3.2, το ρομπότ φέρει μια φωτοδίοδο πράσινου χρώματος (Green LED) , μία χρώματος μπλέ (Blue LED) και μια κόκκινου χρώματος (Red LED).

Η φωτοδίοδος πράσινου χρώματος (Green LED) υποδηλώνει οτι το ρομπότ βρίσκεται σε αυτόματη λειτουργία εντοπισμού αντικειμένων (Detection mode) και δεν εχει ανιχνεύσει κανένα αντικείμενο μέχρι στιγμής.

Η φωτοδίοδος μπλέ χρώματος (Blue LED) υποδηλώνει οτι ένα αντικείμενο ανιχνεύτηκε και οτι αρχίζει η περιστροφή του οχήματος, προς αυτό.

Η φωτοδίοδος κόκκινου χρώματος (Red LED) ανάβει κάθε φορά που το όχημα λαμβάνει τον παλμό σύνδεσης απο το πρόγραμμα χρήστη.

ΠΑΡΑΤΗΡΗΣΕΙΣ:

Η βιβλιοθήκη <IRremote.h> υποστηρίζει την αποστολή δεδομένων με την χρήση υπερύθρων μόνο από το ψηφιακό ακροδέκτη '3' (Digital Pin 3) της πλακέτας. Αυτό σημαίνει οτι για την αποστολή δεδομένων από 2 πομπούς είναι απαραίτητη η χρήση 2 πλακετών arduino ή η κατασκευή ενός κυκλώματος οδήγησης για κάθε πομπό (IR driver). Αντί για αυτό, έγινε η χρήση ενός ηλεκτρονικού διακόπτη (Relay), ώστε να είναι εφικτή η διάκριση των σημάτων που αποστέλλονται, και απο ποιόν πομπό αποστέλλονται κάθε φορά.

Με το πάτημα ενός κουμπιού (Button), που βρίσκεται πάνω στο breadboard shield, γίνεται η επιλεκτική αποστολή των δεδομένων σε κάθε πομπό ξεχωριστά. Ο πομπός των 980nm αποστέλλει τον δεκαεξαδικό αριθμό 0x7A4C, ενώ αντίστοιχα ο πομπός των 880nm τον δεκαεξαδικό αριθμό 0x50DC. Με κάθε αλλαγή της καταστάσης του κουμπιού απο low σε high (0v - 5v) γίνεται η αλλαγή της θέσης στον διακόπτη (Relay), καθώς και η εναλλαγή των δεδομένων προς αποστολή, απο τον ψηφιακό ακροδέκτη '3'. Με αυτόν τον τρόπο πετυχαίνουμε την αποστολή του αριθμού 0x7A4C απο τον πομπό των 980nm και την αποστολή του αριθμού 0x50DC απο τον πομπό των 880nm με την χρήση μιας πλακέτας arduino.

Η μέγιστη απόσταση εντοπισμού της πηγής υπερύθρων (IR) απο το ρομπότ, ειναι προγραμματισμένη στα 400cm, παρόλο που το ρομπότ είναι σε θέση να εντοπίσει μια πηγή (IR) σε μεγαλύτερη απόσταση. Αυτό κρίθηκε απαραίτητο γιατί ο αισθητήρας υπερήχων μπορεί να ανιχνεύσει σε απόσταση <= 4 μέτρων, άρα το όχημα δεν θα μπορούσε να μετρήσει την απόσταση απο την πηγή (IR) αμα η τελευταία υπερέβαινε τα 4 μέτρα.

Ο αισθητήρας IR που φέρει το ρομπότ, ανιχνεύει ακτινοβολία μόνο όταν μια πηγή ειναι απέναντι του. Αυτό επιτυγχάνεται με την κατασκευή ενός 'housing' απο πλαστικό και την χρήση αφρώδους υλικού.

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο αναλύονται τα υλικά που χρησιμοποιήθηκαν για την κατασκευή του ρομπότ και γίνεται μια συνοπτική περιγραφή για το κύκλωμα του καθώς και για τις δυσκολίες που παρουσιάστηκαν κατα την κατασκευή του.

3.2 Συνοπτικά για το κύκλωμα του ρομπότ

Βάρος: 950 γραμμάρια

Χρόνος αναμονής: > 1 μήνας

Η παροχή ενέργειας στο ρομπότ, γίνεται με την χρήση μιας μπαταρίας τάσεως 12 volt και χωρητικότητας 6800 μιλιαμπερορίων (6800mah). Να σημειωθεί οτι δεν ειναι δυνατόν να προσδιοριστεί ο ακριβής χρόνος λειτουργίας του οχήματος, καθώς η κατανάλωση (και κατ'έπεκταση ο χρόνος λειτουργίας) ποικίλει ανάλογα με ποια λειτουργία είναι ενεργοποιημένη, την ταχύτητα των κινητήρων, αμα γίνεται χρήση ή οχι της ασύρματης κάμερας κ.α.

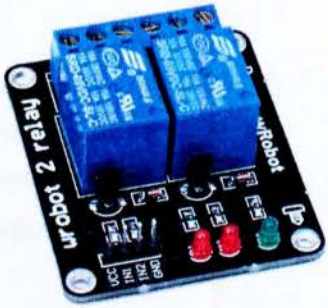
Η λειτουργία εντοπισμού αντικειμένων (Detection Mode) είναι η πιο απαιτητική σε θέμα κατανάλωσης, καθώς γίνεται συνεχής χρήση και των 7 αισθητήρων.

Με αυτήν την μπαταρία τροφοδοτούνται:

- η ασύρματη κάμερα
- οι 2 DC κινητήρες
- οι πλακέτες 'arduino'

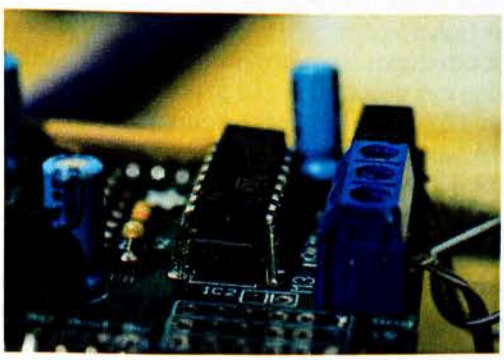
Τα υπόλοιπα υλικά, τροφοδοτούνται μέσω της πλακέτας με τάση της τάξεως των 5 volt

Για πρακτικούς λόγους, έγινε η χρήση ενος ηλεκτρονικού διακόπτη (relay) για την ενεργοποίηση ή απενεργοποίηση της ασύρματης κάμερας, τόσο για λόγους κατανάλωσης όσο για την άσκοπη χρήση της. Ο χρήστης μπορεί να ελέγξει αυτήν την λειτουργία μέσω του γραφικού περιβάλλοντος - πρόγραμμα.



Το ρομπότ είναι προγραμματισμένο να κάνει χρήση 3 φωτοδιόδων (LEDs) σε συνθήκες χαμηλού φωτισμού, κάνοντας εφικτή την μετάδοση βίντεο με φως. Η χρήση των φωτοδιόδων (LEDs) αυτών, γίνεται μόνο όταν η κάμερα είναι ενεργοποιημένη.

Η πλακέτα των κινητήρων ενισχύθηκε με 2 επιπλέον L293(NE) τοποθετώντας το κάθε ολοκληρωμένο κύκλωμα (Microchip) πάνω στα ήδη υπάρχοντα ολοκληρωμένα κυκλώματα (Microchips) της πλακέτας (Συνδεσμολογία piggyback).



Αυτό έγινε με σκοπό την αύξηση του ρεύματος που μπορεί να διαχειριστεί η πλακέτα και κατ'επέκταση οι κινητήρες. Το αποτέλεσμα είναι η διπλάσια αύξηση της 'δύναμης' των κινητήρων.Η θερμοκρασία των ολοκληρωμένων κυκλωμάτων αυξήθηκε, γι'αυτό κρίθηκε απαραίτητη η χρήση ενός ανεμιστήρα (Fan) ώστε να μειωθεί η θερμοκρασία τους να μην οδηγηθούν στην καταστροφή τους.

3.3 Arduino

Το Arduino είναι μια υπολογιστική πλατφόρμα βασισμένη σε μια απλή μητρική πλακέτα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη C++ με κάποιες μετατροπές). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, SuperCollider.

Πλατφόρμα

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.

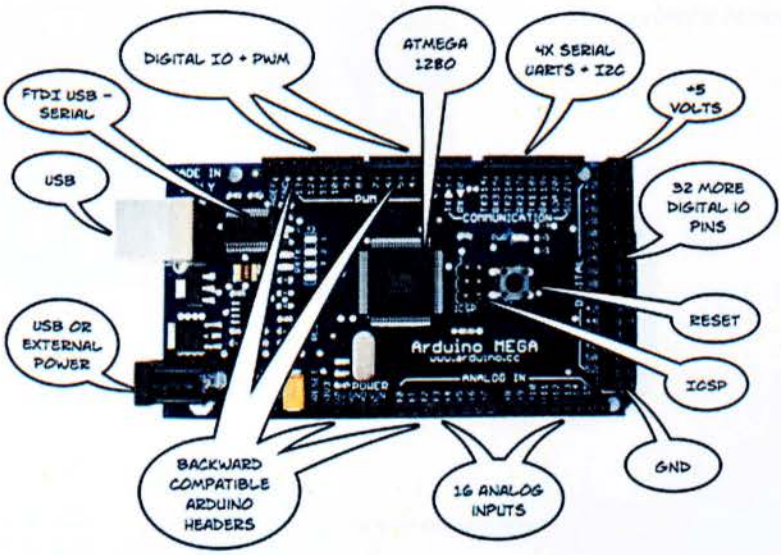
Οι περισσότερες πλακέτες Arduino που κυκλοφορούν σήμερα στην αγορά, προγραμματίζονται μέσω USB, εφαρμόζοντας ένα τσίπ προσαρμογέα USB-to-serial όπως το FTDI FT232.

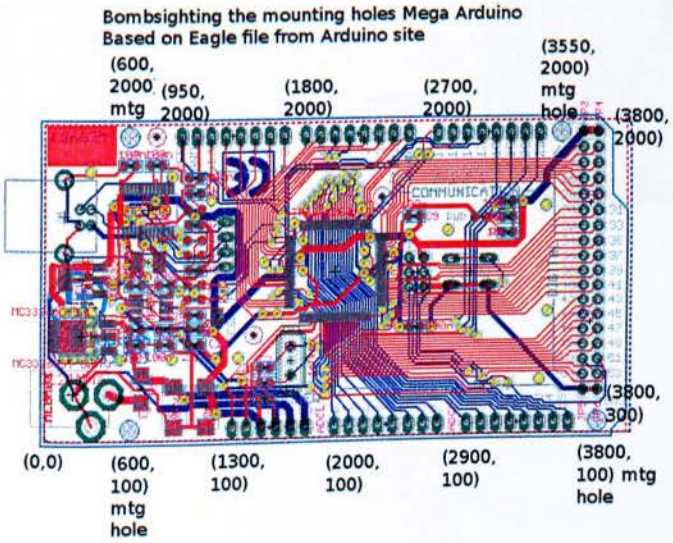
Η πλακέτα του Arduino έχει εκτεθειμένες τις περισσότερες επαφές εισόδου/εξόδου για χρήση με άλλα κυκλώματα. Το Diecimila, για παράδειγμα, παρέχει 14 ψηφιακές επαφές εισόδου/εξόδου, από τις οποίες οι 6 μπορούν να παράξουν σήματα PWM, και 6 αναλογικές εισόδους. Αυτές οι επαφές είναι διαθέσιμες στην κορυφή της πλακέτας μέσω θηλυκών συνδέσεων μεγέθους 0,1 ιντσών.

Διάφορες plug-in πλακέτες εφαρμογών γνωστές σαν "shields" είναι, επίσης, διαθέσιμες στο εμπόριο.

Λογισμικό

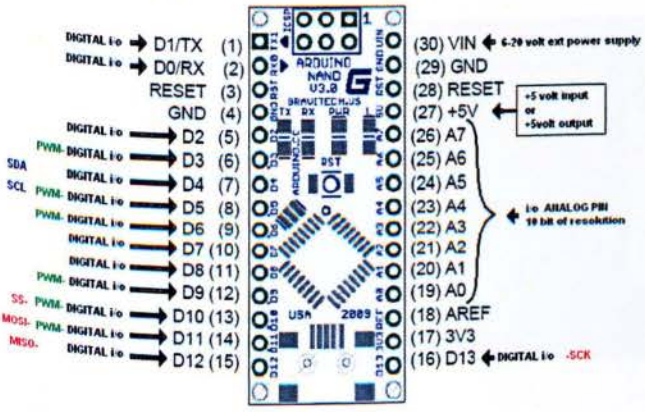
Το IDE του Arduino είναι γραμμένο σε Java και μπορεί να τρέξει σε πολλαπλές πλατφόρμες. Περιλαμβάνει επεξεργαστή κώδικα (επεξεργαστή κειμένου με διάφορα εύχρηστα εργαλεία) και μεταγλωττιστή, και έχει την ικανότητα να φορτώνει εύκολα το πρόγραμμα μέσω σειριακής θύρας από τον υπολογιστή στην πλακέτα.

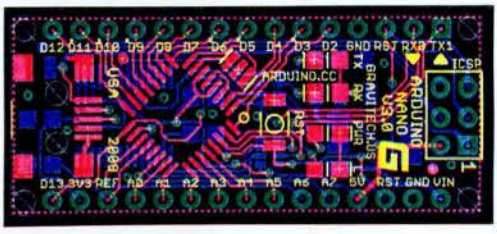




3.3.1 Arduino Nano

Το arduino nano είναι μια πλακέτα arduino με 8 αναλογικές εισόδους/ εξόδους και 14 ψηφιακούς ακροδέκτες.
Η συχνότητα λειτουργίας του είναι 16Mhz



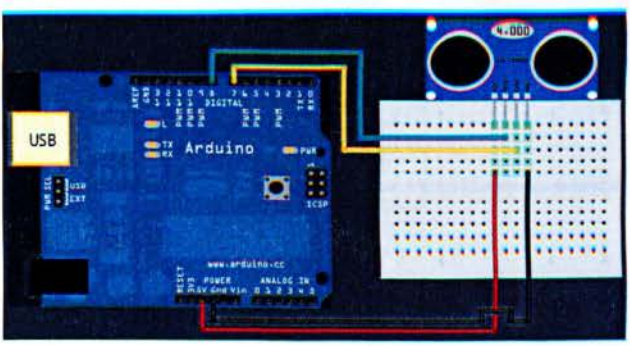


3.4 Αισθητήρας υπερήχων

Σαν υπέρηχο καθορίζουμε εκείνο το κύμα το οποίο βρίσκεται πάνω από την μέγιστη συχνότητα που μπορεί να ακούσει το ανθρώπινο αυτί.

Οι αισθητήρες υπερήχων λειτουργούν με την ίδια αρχή που λειτουργούν τα ραντάρ και τα σόναρ. Εκτιμούν την απόσταση ενός στόχου λαμβάνοντας υπόψη τους την αντανάκλαση ενός ραδιοκύματος ή ενός ηχητικού σήματος πάνω στο στόχο. Δημιουργούν υψηλής συχνότητας κύματα και χρησιμοποιώντας το επιστρεφόμενο σήμα καθορίζουν την απόσταση ή ακόμα και την ταχύτητα του στόχου. Για να το επιτύχουν αυτό χρησιμοποιούν τον χρόνο που έκανε το σήμα για να καλύψει την απόσταση από τον αισθητήρα στο αντικείμενο και πίσω. Εφαρμογές τους θα βρούμε σε ένα μεγάλο εύρος τεχνολογιών από την μέτρηση της διεύθυνσης και της ταχύτητας του ανέμου έως και την απεικονιστική ιατρική.

Αισθητήρας υπερήχων HC-SR04 για τον υπολογισμό απόστασης. Η απόσταση που μπορεί να υπολογίσει είναι από 2εκ. έως 400εκ. με ακρίβεια ενός εκατοστού.



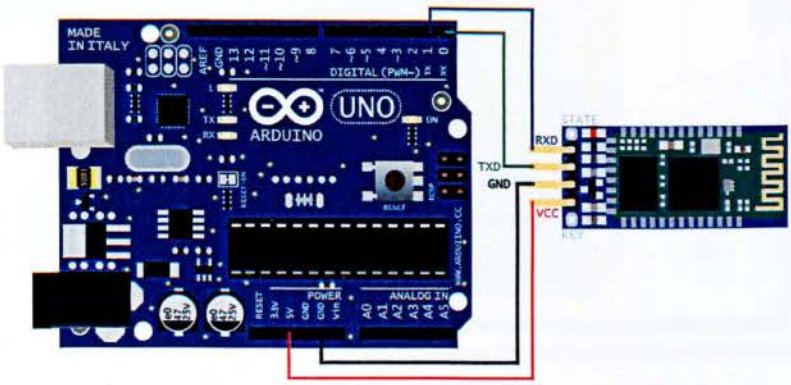
Ο αισθητήρας έχει 4 ακροδέκτες.

Vcc = Τάση 5 Volt
GND = Γείωση
Trigger = Σήμα για την ενεργοποίηση του αισθητήρα
Echo = Αποτελέσματα προς επεξεργασία (χρόνος)

3.5 Bluetooth

Το Bluetooth είναι ένα βιομηχανικό πρότυπο.Πρόκειται για μια ασύρματη τηλεπικοινωνιακή τεχνολογία μικρών αποστάσεων, η οποία μπορεί να μεταδώσει σήματα μέσω μικροκυμάτων σε ψηφιακές συσκευές. Επομένως το Bluetooth είναι ένα πρωτόκολλο το οποίο παρέχει προτυποποιημένη, ασύρματη επικοινωνία ανάμεσα σε PDA, κινητά τηλέφωνα, φορητοί υπολογιστές, προσωπικοί υπολογιστές, εκτυπωτές, καθώς και ψηφιακές φωτογραφικές μηχανές ή ψηφιακές κάμερες, μέσω μιας ασφαλούς, φθηνής και παγκοσμίως διαθέσιμης χωρίς ειδική άδεια ραδιοσυχνότητας μικρής εμβέλειας. Από τεχνικής άποψης το Bluetooth είναι ένα πρωτόκολλο ασύρματης δικτύωσης σε φυσικό επίπεδο, υποεπίπεδο MAC και, προαιρετικά, υποεπίπεδο LLC.

Το Bluetooth, χρησιμοποιεί συχνότητες στα 2,45 GHz για την μετάδοση πληροφοριών και η περιοχή εμβέλειας μιας Bluetooth συσκευής είναι συνήθως γύρω στα 10 μέτρα, αν και ειδικές Bluetooth συσκευές μπορούν να καλύψουν μέχρι και 100 μέτρα.



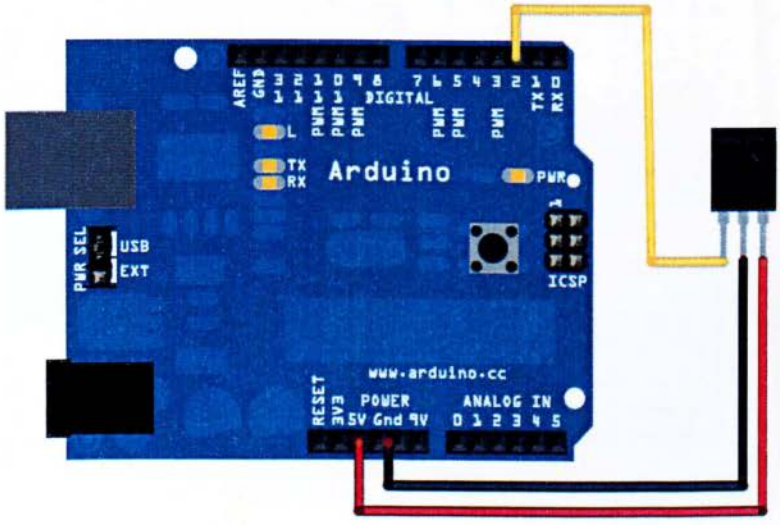
Ο αισθητήρας έχει 4 ακροδέκτες.

Vcc = Τάση 5 Volt
GND = Γείωση
RXD = Λαμβανόμενα δεδομένα
TXD = Εκπεμπόμενα δεδομένα

3.6 Αισθητήρας υπερύθρων

Η υπέρυθρη ακτινοβολία ή υπέρυθρες ακτίνες είναι τμήμα του φάσματος της ηλεκτρομαγνητικής ακτινοβολίας. Στο φάσμα τοποθετούνται ως μικρότερη συχνότητα στην προέκταση της κόκκινης ορατής ακτινοβολίας.

Το μήκος κύματός τους κυμαίνεται από το 1 χιλιοστό έως τα 700 νανόμετρα, όπου ξεκινά το ορατό φάσμα. Συνήθως εκπέμπονται από όλα τα σώματα που έχουν κάποια θερμοκρασία. Τα σώματα με τη μεγαλύτερη θερμοκρασία εκπέμπουν περισσότερες υπέρυθρες.

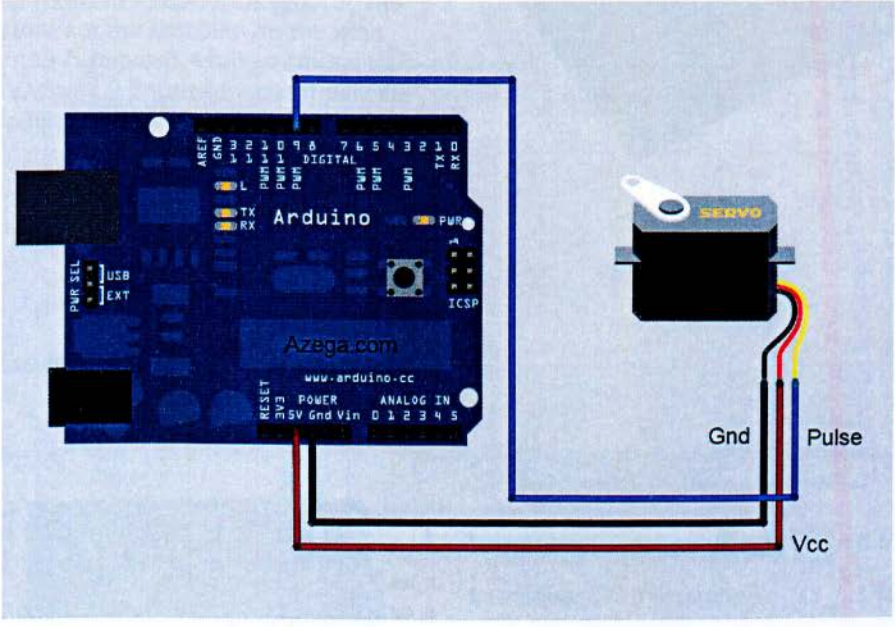


Ο αισθητήρας υπερύθρων έχει 3 ακροδέκτες.

Vcc = Τάση 5 Volt
GND = Γείωση
Data = Δεδομένα

3.7 Σερβοκινητήρας

Η λέξη "σέρβο" από τη λατινική λέξη "servus" που σημαίνει υπηρέτης. Ο σερβοκινητήρας μπορεί να θεωρηθεί ότι είναι ένας κινητήρας που ανάλογα με τις ανάγκες, μετακινεί τον αξονα του σε ενα καθορισμένο σημείο δίνοντας του την κατάλληλη συχνότητα στην ακροδέκτη 'pulse'

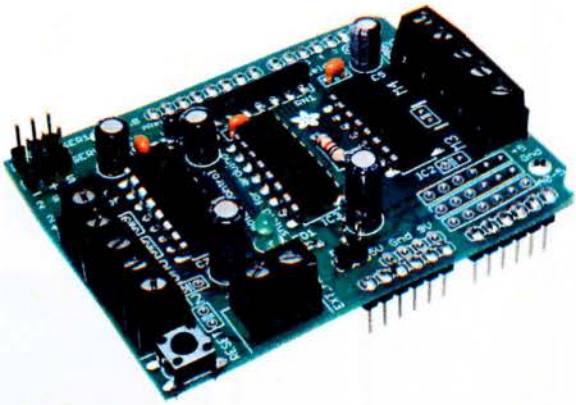


Ο σερβοκινητήρας έχει 3 ακροδέκτες.

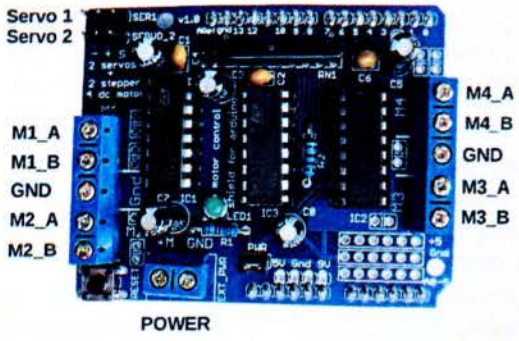
Vcc = Τάση 5 Volt
GND = Γείωση
Pulse = Σήμα εισόδου / (ή και εξόδου)

3.8 Πλακέτα οδήγησης κινητήρων (Motor Shield)

Το Motor Shield της adafruit βασίζεται στο L298, η οποία είναι μια διπλή πλήρη γέφυρα οδηγός σχεδιάστηκε για να οδηγεί επαγωγικά φορτία, όπως ηλεκτρονόμοι, πηνία, DC και βηματικών κινητήρων.Η πλακέτα επιτρέπει την οδήγηση τεσσάρων κινητήρων συνεχούς ρεύματος με μια πλακέτα Arduino, ελέγχοντας την ταχύτητα και την κατεύθυνση του κάθε κινητήρα ξεχωριστά. Μπορεί επίσης να γινει έλεγχος 2 βηματικών κινητήρων και 2 σερβοκινητήρων.



Συνδεσμολογία :



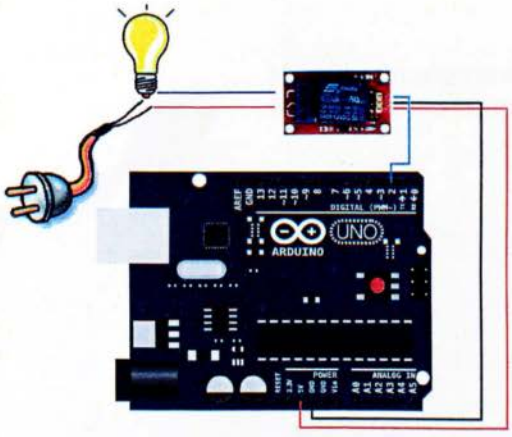
1 κινητήρας DC στης υποδοχές M4_A & M4_B

1 κινητήρας DC στις υποδοχές M3_A & M3_B

1 σερβοκινητήρας στην υποδοχή servo 1

Τροφοδοσία 12V στην υποδοχή POWER

3.9 Ηλεκτρονικός διακόπτης (Relay)

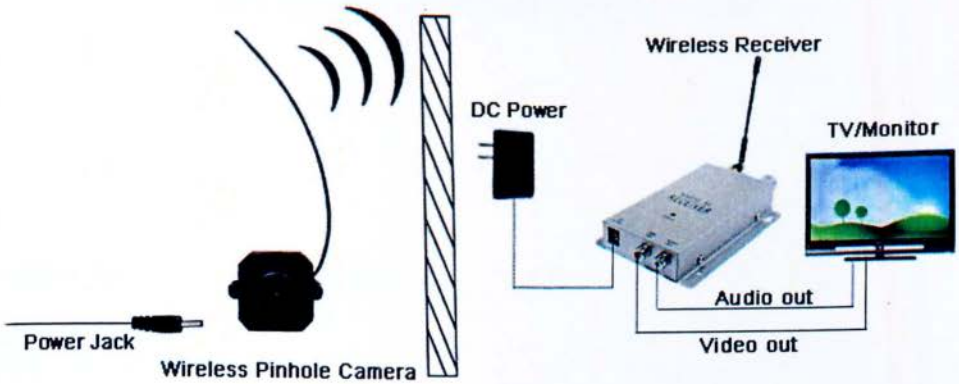


Ο ηλεκτρονόμος, ρελέ (relay) ή ρελές είναι ένας ηλεκτρικός διακόπτης που ανοίγει και κλείνει ένα ηλεκτρικό κύκλωμα κάτω από τον έλεγχο ενός άλλου ηλεκτρικού κυκλώματος. Συνήθως ένας ηλεκτρονόμος αποτελείται από περισσότερες από μία ελεγχόμενες επαφές. Ο ρόλος τους είναι να βοηθούν στον έλεγχο των αυτοματισμών, για παράδειγμα βοηθούν στην ενεργοποίηση/απενεργοποίηση βοηθητικών κυκλωμάτων όπως ενδεικτικές λυχνίες.

Όταν ηλεκτρικό ρεύμα διαρρέει το πηνίο του ηλεκτρονόμου, το παραγόμενο μαγνητικό πεδίο έλκει έναν οπλισμό που είναι μηχανικά συνδεδεμένος σε μια κινούμενη επαφή. Έτσι, η κινούμενη επαφή είτε συνδέεται με μια σταθερή επαφή είτε αποσυνδέεται από τη σταθερή επαφή. Μόλις το ηλεκτρικό ρεύμα στο πηνίο διακοπεί, ο οπλισμός επιστρέφει στη θέση ηρεμίας του εξαιτίας μιας δύναμης επαναφοράς, που είναι ίση με το ήμισυ της μαγνητικής. Η δύναμη επαναφοράς παρέχεται συνήθως από ένα ελατήριο.

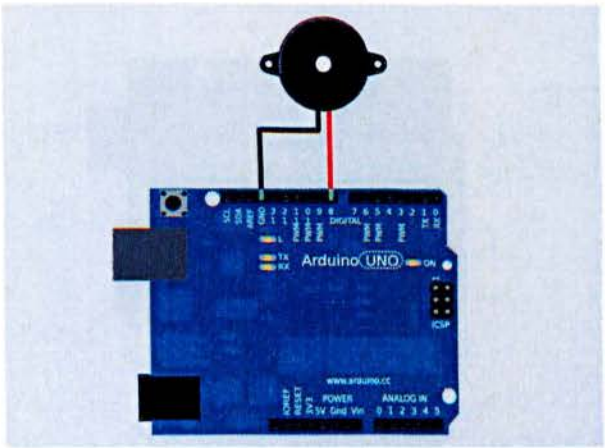
3.10 Ασύρματη κάμερα

Η ασύρματη κάμερα εκπέμπει εικόνα
(και ήχο) σε ενα δέκτη, μέσω bluetooth



3.11 Ηχείο

Το **ηχείο** (*speaker*) αποτελεί μία διάταξη/συσκευή, η οποία έχει σκοπό τη μετατροπή της λαμβανόμενης ηλεκτρικής ενέργειας (εισερχόμενο σήμα) σε ακουστική ενέργεια, δηλαδή σε στιγμιαίες μεταβολές πίεσης του ατμοσφαιρικού αέρα (διαμήκη κύματα), οι οποίες αντιστοιχούν σε όσο το δυνατόν περισσότερο φυσικό και αληθοφανή ήχο.

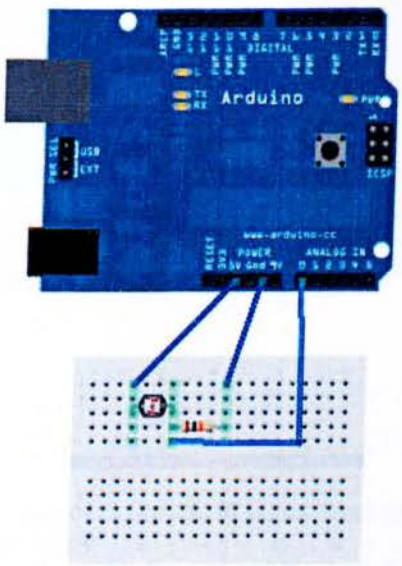


Ο ηχείο (buzzer) έχει 2 ακροδέκτες.

Data = Σήμα εισόδου
GND = Γείωση

3.12 Αισθητήρας φωτός

Το φωτοκύτταρο αυτό, αλλάζει την εσωτερική του αντίσταση, ανάλογα με την ποσότητα του φωτός που είναι εκτεθειμένη πάνω του.

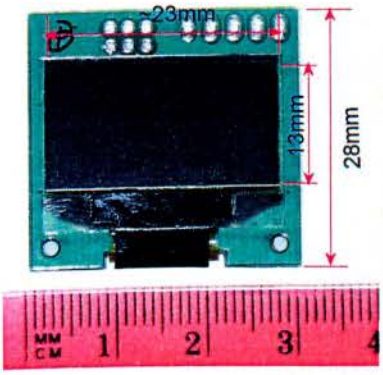


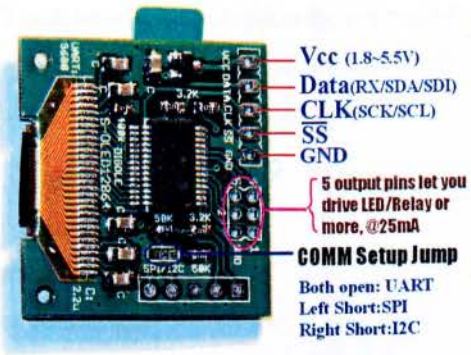
Ο φωτοκύτταρο έχει 2 ακροδέκτες.

Data = Αναλογικά δεδομένα εξόδου (πρέπει να γίνει η κατάλληλη γείωση)
Vcc = Τάση 5 Volt

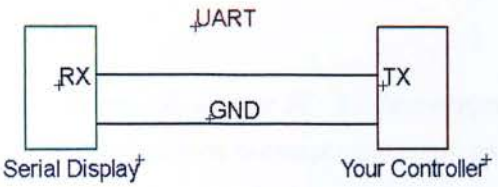
3.13 Οθόνη OLED

Μια οθόνη OLED είναι μια συσκευή φωτεινής ένδειξης κατασκευασμένη από οργανικά, ημιαγώγιμα υλικά. Ο όρος OLED αναφέρεται στην οργανική φωτοδίοδο και αποτελεί ακρωνύμιο των αγγλικών λέξεων «Organic Light Emitting Diode». Το λεπτό στοιχείο φωτοεκπομπής διαφέρει από τη συμβατική, ανόργανη δίοδο φωτοεκπομπής (LED) ως προς τη μικρότερη πυκνότητα ρεύματος και φωτός.





Η οθόνη συνδέθηκε με τον μικροεπεξεργαστή μεσω πρωτόκολλου *UART*, όπως απεικονίζεται παρακάτω (εικ.3.13)



Εικόνα 3.13

Vcc = Τάση 5 Volt
GND = Γείωση
Data = Δεδομένα εισόδου (RX)

3.14 Κάτι πήγε στράβα....

Υπήρξαν αρκετά προβλήματα κατά την κατασκευή του ρομπότ. Τόσο σε θέμα υλικού (Hardware) όσο σε θέμα λογισμικού (Software).

Η πλακέτα "arduino" διαθέτει έναν σταθεροποιητή/ρυθμιστή τάσης (voltage regulator) για την ελλάτωση της τάσης σε 5 volt, καθώς ο επεξεργαστής και πολλά από τα περιφεριακά του οχήματος, χρειάζονται 5 volt τάση για να λειτουργήσουν σωστά. Ο ρυθμιστής τάσης μετατρέπει ένα μέρος την τάσης εισόδου σε θερμοκρασία ώστε στην έξοδο του να απορρέει σταθερή τάση 5 volt.

Η χρήση της μπαταρίας τάσης 12 volt που χρησιμοποιήθηκε είχε σαν αποτέλεσμα την υπερβολική αύξηση της θερμοκρασίας του ρυθμιστή τάσης με αποτέλεσμα να καταστρέψει ένα μέρος της πλακέτας που είναι υπεύθυνο για την τροφοδοσία της. Κάτα συνέπεια, το όχημα δεχόταν ενέργεια μόνο με την χρήση του σειριακού καλωδίου (USB) και όχι απο την εξωτερική τροφοδοσία. Το πρόβλημα διορθώθηκε με την χρήση ενός 7805 ρυθμιστή τάσης, ο οποίος πλέον τοποθετήθηκε στο σασί του οχήματος για καλύτερη διάχυση της θερμότητας.

Οι κινητήρες του οχήματος είναι της τάξεως των 3V - 6V. Δίνοντας στους κινητήρες την μέγιστη ταχύτητα τους, το αποτέλεσμα ήταν οι επαφές στο εσωτερικό των κινητήρων να αλλοιωθούν και οι κινητήρες να υπολειτουργούν ή να πάψουν να λειτουργούν. Το πρόβλημα αντιμετωπίστηκε αποσυναρμολογώντας τους κινητήρες και επισκευάζοντας τις εσωτερικές επαφές που είχαν λυγίσει απο την υψηλή τάση.

Ένα άλλο πρόβλημα που αντιμετωπίστηκε ήταν ο θόρυβος (Noise) που προκαλούσαν οι κινητήρες κατά την λειτουργία τους, με αποτέλεσμα να υπάρχει συχνά αν οχι πάντα, επανεκκίνηση του επεξεργαστή του ρομπότ, και κατ'επέκταση η απώλεια της σειριακής επικοινωνίας. Το πρόβλημα παρουσιαζόταν στις υψηλές ταχύτητες των κινητήρων και όταν αυτοί δούλευαν για 5-10 δευτερόλεπτα. Το πρόβλημα αντιμετωπίστηκε με την χρήση 6 πυκνωτών (Capacitors) χωρητικότητας 0.1μF (0.1uF - 100nF).

Εικόνα 3.14

Οι πυκνωτές τοποθετήθηκαν ένας μεταξύ των 2 ακροδεκτών του κινητήρα, και οι άλλοι δυο μεταξύ του κάθε ακροδέκτη και του κινητήρα, ξεχωριστά, όπως απεικονίζεται παραπάνω. (εικ.3.14)

Πρόβλημα παρουσιάστηκε επίσης κατά την χρήση της πλακέτας των κινητήρων (Motor shield). Με την χρήση πολυμέτρου διαπιστώθηκε οτι τα τερματικά (Terminals) των κινητήρων δεν έδιναν ίδια τάση αλλά υπήρχε μια διαφορά τάσης της τάξεως του 1.5volt μεταξύ των δυο τερματικών (Terminals). Με αυτόν τον τρόπο το όχημα αντί να πάει ευθεία έκανε ανοιχτούς κύκλους μεγάλης διαμέτρου. Η δοκιμή έγινε για συχνότητες 1Khz, 8Khz, 16Khz, και 64Khz.

Αξίζει να σημειωθεί οτι όσο πιο υψηλή συχνότητα (frequency) δίνεται στους κινητήρες, τόσο πιο πολύ 'δύναμη' έχουν οι κινητήρες αλλά ο παραγόμενος θόρυβος αυξάνεται.

Το ενδεχόμενο για ελλατωματική πλακέτα των κινητήρων καταρρίφθηκε όταν έγινε δοκιμή πάνω σε άλλη πλακέτα 'arduino', και συγκεκριμένα του μοντέλου 'UNO'. Διαφοροποιώντας την συνδεσμολογία υπήρξε ορατά πιο καλό αποτέλεσμα.

Το όχημα φέρει 3 Led (White LED) παράλληλα συνδεδεμένα, που ενεργοποιούνται αυτόματα σε συνθήκες χαμηλού φωτισμού και όταν η κάμερα είναι ενεργοποιημένη. Ωστόσο κατά την ενεργοποίηση τους, τα LEDs τρεμοέπαιζαν, με αποτέλεσμα να μην υπάρχει σταθερός φωτισμός (η φωτεινότητα των LEDs αυξομειωνόταν.)

Οι απαιτήσεις για ρεύμα είναι 60mA (3 LEDs x 20mA)

Γι'αυτόν τον λόγο έγινε παροχή ρεύματος και απο ένα δεύτερο ακροδέκτη. Έτσι, κατά την ενεργοποιήση των LEDs δίνεται ρεύμα 40mA απο τον έναν ακροδέκτη και 40mA απο τον άλλον.

(40mA X 2 = 80mA)

Ο ανεμιστήρας για την ψύξη των ολοκληρωμένων κυκλωμάτων των κινητήρων λειτουργεί με τάση 12Volt. Παρόλο αυτά, οι στροφές του ανεμιστήρα (rpm) (και κατ'επέκταση ο παραγόμενος θόρυβος κατά την λειτουργία του), ήταν αρκετές για τις απαιτήσεις της ψύξης του συστήματος. Γι'αυτό τροφοδοτήθηκε με 5 volt έναντι των 12 volt. Με αυτόν τον τρόπο μειώθηκαν και οι στροφές του ανεμιστήρα και ο παραγόμενος θόρυβος, ενώ η απόδοση της ψύξης έμεινε σχεδόν αμετάβλητη.

ΠΑΡΑΤΗΡΗΣΗ

Αρχικά, το πρόγραμμα σχεδιάστηκε για να μπορεί να ελέγχει την διαθεσιμότητα της σύνδεσης μέσω της συνάρτησης Try / Catch. 'Οταν υπήρχε σειριακή επικοινωνία μέσω καλωδίου usb, το πρόγραμμα ανταποκρινόταν όπως θα έπρεπε. Ο παλμός σύνδεσης δημιουργήθηκε καθώς η χρήση της συνάρτησης Try / Catch δεν ήταν δυνατό να χρησιμοποιηθεί μαζί με την χρήση της τεχνολογίας bluetooth. Το πρόγραμμα όταν επιχειρούσε να στείλει δεδομένα και η αποστολή αποτύγχανε (λόγω χαμένης σύνδεσης), η συνάρτηση Try / Catch δεν είχε τα επιθυμητά αποτελέσματα με αποτέλεσμα το πρόγραμμα να σταματήσει να ανταποκρίνεται.

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο παρατίθεται ο κώδικας που γράφτηκε για να λειτουργήσει το όχημα σε συνεργασία με το πρόγραμμα χρήστη.

4.2 Τα μέρη του λογισμικού

Το λογισμικό που υλοποιήθηκε, αποτελείται απο 3 μέρη.

*(*)**Το πρώτο** απευθύνεται στον τελικό χρήστη και είναι το πρόγραμμα μέσω του οποίου γίνεται η αποστολή των δεδομένων για τον χειρισμό του robot, καθώς και για την πληροφόρηση του χρήστη για την κατάσταση του ρομπότ.*

*(*)**Το δεύτερο** απευθύνεται στο ίδιο το όχημα, με το οποιο γίνεται η επεξεργασία των δεδομένων, ο έλεγχος των αισθητήρων και των κινητήρων dc / servo*

Αξίζει να σημειωθεί, οτι το ρομπότ έχει 2 μικροελεγκτές/εγκεφάλους. Ο πρώτος εγκέφαλος ελέγχει τις κύριες λειτουργίες του ρομπότ (κάμερα, πλακέτα κινητήρων, επικοινωνία με τον ηλεκτρονικό υπολογιστή), και ο δεύτερος, την οθόνη, την διαθεσιμότητα της σύνδεσης (παλμός σύνδεσης) και την λειτουργία υπερύθρων

Η λειτουργία υπερύθρων ανατέθηκε αποκλειστικά στον δεύτερο εγκέφαλο, καθώς οι βιβλιοθήκες της δεν ηταν συμβατές με τις βιβλιοθήκες της λειτουργίας υπερήχων.

Έγινε η απόπειρα παραμετροποίησης των timers στα αρχεία ".cpp" στις βιβλιοθήκες <IRremote> , <NewPing>, ώστε να γίνει εφικτός ο έλεγχος αντίστοιχα, στους αισθητήρες υπερύθρων και υπερήχων απο τον ίδιο μικροελεγκτή, ωστόσο υπήρχαν πολλά προβλήματα στον συγχρονισμό και οι αισθητήρες έπαυαν να λειτουργούν και να ανταποκρίνονται στον κώδικα.*

*(*)**Το τρίτο** απευθύνεται στις πηγές υπέρυθρης ακτινοβολίας, ποιά σήματα θα στέλνονται, με ποιά συχνότητα θα εκπέμπονται και απο ποιά πηγή.*

ΣΗΜΕΙΩΣΗ:

Λόγω της συνεχούς αναβάθμησης του λογισμικού και την επίλυση μικρολαθών (Bugs), το παρόν πρόγραμμα, ένδεχεται να διαφέρει απο το τελικό.

Παρακάτω παρατίθενται τα 4 μέρη του κώδικα.

IRIS v2.6 - *Πρόγραμμα χρήστη (Mac os x)*

IRIS MEGA - *Πρόγραμμα μικροελεγκτή Mega*

IRIS NANO - *Πρόγραμμα μικροελεγκτή Nano*

IRIS UNO - *Πρόγραμμα μικροελεγκτή Uno - πηγές υπέρυθρων*

(*) Ο κώδικας δεν απεικονίζεται στην κανονική στοίχιση του

4.2.1 IRIS v2.6 :

```
/* ##############################################################################################
   #                                                                                            #
   #     Technological Education Institute of Piraeus                                           #
   #     Department of Electronic Computer Systems                                              #
   #     250 Thivon & P.Ralli Ave.                                                              #
   #     12244 Egaleo, Athens, Greece                                                           #
   #     Tel. : +30 (210) 538-1110                                                              #
   #                                                                                            #
   #                                                                                            #
   #     by Panagiotis Athanasiou                                                               #
   #     E-mail: panos.ips@gmail.com                                                            #
   #                                                                                            #
   #     IRIS - Processing v2.6                                                                 #
   #     Last Modified: 3 September 2013 @ 01:26                                                #
   #                                                                                            #
   ############################################################################################## */


import processing.serial.*;
import java.awt.Frame;
import processing.core.PApplet;  // Importing libraries


PFrame F;
Functions S;

Serial port;                          // New serial connection

public void setup() {
  frame.setTitle("IRIS v2.6");                          // Current version
  println(Serial.list());                               // Display available ports
  port = new Serial(this, Serial.list()[5], 9600);   // New serial connection on port #5     -> (Serial(this,
Serial.list()[X], 9600);) , where "X" is selected port
  port.buffer(1);
  size(680, 400);                                       // Window size (680*400)
  PFrame f = new PFrame();
}

int valueU = 0;                 // Variable for UP arrow
int valueD = 0;                 // Variable for DOWN arrow
int valueL = 0;                 // Variable for LEFT arrow
int valueR = 0;                 // Variable for RIGHT arrow
int valueCR = 0;                // Variable for camera LEFT arrow
int valueCL = 0;                // Variable for camera RIGHT arrow
int valueB = 70;                // Variable for detection button
int valueBC = 70;               // Variable for camera button rectangle
int valueMY = 50;               // Variable for Y coordinate
int valueMX = 95;               // Variable for X coordinate
int valueBD = 150;              // Variable for distance rectangle
int valueBI = 70;               // Variable for IR button
int valueBIObj1 = 70;           // Variable for object1 button
int valueBIObj2 = 70;           // Variable for object2 button
int valueBIR = 200;             // Variable for 'RUN' button
int valueText = 150;            // Variable for textfield 'Detection Distance'
int valueTextC = 150;           // Variable for textfield 'Camera'
int valueIntData = 0;           // Variable to convert float to integer
int valueIntDText = 0;          // Variable to convert float to integer and display it on 'distance' textfield
int valueIntSText = 0;          // Variable to convert float to integer and display it on 'current speed' textfield
int var = 0;                    // Variable for sending codes
int objD;                       // Distance from the object
int wait = 0;                   // Counter
int start;                      // Variable for connection pulse
int sTemp;                      // Temporary input
int offScreen = 0;              // Variable for offline screen
long waitCR;                    // Counter
long waitC;                     // Counter
long waitD;                     // Counter
long splash = 0;                // "Processing.." text - Time to establish serial connection
float valueFloatData = 0;       // Variable to store float numbers
float valueFloatDText = 0;      // Variable to store float numbers
float valueFloatSText = 0;      // Variable to store float numbers
boolean send = false;           // Flag - Distanse change
boolean sendString = false;     // Flag - Mouse Released
boolean state, pState;          // Communication Flags
String direction = "";          // Variable for direction
String valueCS = "OFF" ;        // Variable for Camera status

public void draw() {

  state = false;                              // Variables
  state = keyPressed;
  splash = millis();                          // Timer
  waitC = millis();

  background(228, 231, 237);                  // Background color
```

```
if (start == 0) {                          // Connection pulse
  port.write(222);
  start = 1;
}

while (port.available() > 0) {             // Wait for sensor data
  sTemp = port.read();
  port.write(222);
  waitCR = millis();
  println("> Connection Pulse <");

  if (sTemp != 223) {
    objD = sTemp;
  }
}

waitD = waitC - waitCR;
if (waitD > 5000) {
offScreen = 1;
}

if (offScreen == 1) {
  background(208, 211, 217);                // Background color
  fill(28, 31, 27);
  textSize(22);
  text("CONNECTION LOST", 235, 110);
  textSize(16);
  text("Please, restart the application ...", 210, 150);
  text("( make sure that IRIS is in range and turned on )", 145, 170);
  delay(1000);
}
else {
  if (sendString == false) {
    if (splash > 2500) {                    // Give 2 seconds to make
      fill(28, 31, 27);                     // sure that the serial
      text("Ready", 620, 390);              // connection has been established -
    }                                       // Data doesn't transmitted instantly
    else {                                  // after launching the program
      fill(28, 31, 27);
      text("Preparing...", 591, 390);
    }
  }

  if (sendString == true && wait < 23) {    // Show "Sending" text after changing
    fill(28, 31, 27);                       // detection distance
    text("Sending...", 600, 390);
    wait++;
    send = false;
  }
  else {
    sendString = false;
    wait = 0;
  }

  fill (133, 142, 154);
  rect(40, 45, 170, 170);                   // Direction frame

  fill (valueU, 76, 76);
  triangle(125, 50, 100, 100, 150, 100);    // UP arrow

  fill (valueD, 76, 76);
  triangle(100, 160, 125, 210, 150, 160);   // DOWN arrow

  fill (valueL, 76, 76);
  triangle(95, 105, 45, 130, 95, 155);      // LEFT arrow

  fill (valueR, 76, 76);
  triangle(155, 105, 205, 130, 155, 155);   // RIGHT arrow

  fill (228, 240, 255);
  rect(285, 45, 150, 60);                   // Camera frame

  fill (valueCL, 76, 76);
  triangle(345, 50,295, 75, 345, 100);      // Camera RIGHT arrow

  fill (valueCR, 76, 76);
  triangle(375, 50, 425, 75, 375, 100);     // Camera RIGHT arrow

  fill (50, 150, 80);
  rect(510, 45, 100, 170);                  // Speed rectangle

  fill (50, valueB, 80);
  rect(40, 300, 20, 20);                    // Detection button

  fill (50,valueBD, 80);
  rect(90, 300, 120, 20);                   // Distance rectangle

  fill (50, valueBC, 80);
  rect(40, 330, 20, 20);                    // Camera button

  fill (50, valueBI, 80);
```

```
    rect(40, 360, 20, 20);                    // IR button

    fill (50, valueBIObj2, 80);
    rect(90, 360, 30, 20);                    // IR Obj2 button

    fill (50, valueBIObj1, 80);
    rect(140, 360, 30, 20);                   // IR Obj1 button

    fill (200, valueBIR, 200);
    rect(195, 360, 40, 20);                   //  RUN button

    fill(255, 100, 153);
    text("Obj1", 90, 355);                    // Textfield 'Obj1'

    fill(255, 100, 153);
    text("Obj2", 140, 355);                   // Textfield 'Obj2'

    fill(255, 100, 153);
    text("Direction", 95, 25);                // Textfield 'Direction'

    fill(255, 100, 153);
    text("Camera", 335, 25);                  // Textfield 'Camera'

    fill(255, 102, 153);
    text("Speed", 542, 25);                   // Textfield 'Speed'

    fill(255, 100, 153);
    text("U:", 20, 315);                      // Textfield 'U:'

    fill(255, 100, 153);
    text("C:", 20, 345);                      // Textfield 'C:'

    fill(255, 100, 153);
    text("IR:", 20, 375);                     // Textfield 'C:'

    fill(255, 100, 153);
    text("RUN", 202, 375);                    // Textfield 'RUN'

    fill(valueTextC, 102, 153);
    text("Camera: " + valueCS, 578, 340);     // Information textfield 'Camera'

    fill(255, 102, 153);
    text("Current Speed: " + valueIntSText + "/10", 530, 360);            // Information textfield 'Current
Speed'

    fill(valueText, 102, 153);
    text("Detection Distance:   " + valueIntDText + " cm", 490, 300);     // Information textfield 'Detection
Distance'

    fill(255, 102, 153);

    if (objD == 0) {
      fill(150, 102, 153);
      text("Object Distance:     --", 510, 320);                          // Object's distance information
    }
    else if (objD <= 10) {
      fill(255, 102, 153);
      text("Object Distance:     "+ objD + " cm", 510, 320);
    }
    else if (objD <= 100) {
      fill(255, 102, 153);
      text("Object Distance:   "+ objD + " cm", 510, 320);
    }
    else {
      fill(255, 102, 153);
      text("Object Distance: "+ objD + " cm", 510, 320);
    }

    if (valueBI != 255 && valueB != 255) {

      if (keyPressed == true && (keyCode == UP || keyCode == DOWN || key == 'w'|| key == 's')) {            //
Loop for the UP and the DOWN arrow
        fill(76, 120, 154);
        text("Moving "+ direction, 75, 250);
      }
      else if (keyPressed == true && (keyCode == LEFT || keyCode == RIGHT || key == 'a'|| key == 'd')) {    //
Loop for the LEFT and the RIGHT arrow
        fill(76, 120, 154);
        text("Turning"+ direction, 85, 250);
      }

      if (keyPressed == true && (keyCode == UP || key == 'w')) {          // Change color on UP arrow
        valueU = 255;
        direction = "Forward";
      }
      else {
        valueU = 76;
      }

      if (keyPressed == true && (keyCode == DOWN|| key == 's')) {         // Change color on DOWN arrow
        valueD = 255;
```

```
      direction = "Backwards";
    }
    else {
      valueD = 76;
    }

    if (keyPressed == true && (keyCode == LEFT || key == 'a')) {        // Change color on LEFT arrow
      valueL = 255;
      direction = " Left";
    }
    else {
      valueL = 76;
    }

    if (keyPressed == true && (keyCode == RIGHT || key == 'd')) {       // Change color on RIGHT arrow
      valueR = 255;
      direction = " Right";
    }
    else {
      valueR = 76;
    }
  }
  else {
    if (keyPressed == true && (keyCode == UP || keyCode == DOWN || keyCode == LEFT || keyCode == RIGHT || key ==
'w' || key == 's' || key == 'a' || key == 'd')) {
      fill(76, 120, 154);
      text("Please, enable Manual Mode", 40, 250);
    }
  }

  if (keyPressed == true && key == '4') {                               // Move camera left - Change color on
camera LEFT arrow
    valueCL = 255;
    var = 5;
    port.write(var);
    println(var);
    delay(50);
  }
  else {
    valueCL = 76;
  }

  if (keyPressed == true && key == '6') {                               // Move camera right - Change color on
camera RIGHT arrow
    valueCR = 255;
    var = 6;
    port.write(var);
    println(var);
    delay(50);
  }
  else {
    valueCR = 76;
  }

  if (keyPressed == true && key == '5') {                           // Set servo at 90 degrees
    var = 9;
    port.write(var);
    println(var);
  }
  else if (keyPressed == true && key == 'o') {                      // Set maximum / minimum speed when 'p' or
'o' keys are pressed
    valueMY = 50 ;
    port.write(valueMY);
  }
  else if (keyPressed == true && key == 'p') {
    valueMY = 210;
    port.write(valueMY);
  }

  valueFloatData = map(valueMX, 95, 205, 10, 20);                   // Rearrange float numbers to setpoints
  valueFloatDText = map(valueMX, 95, 205, 20, 90);
  valueFloatSText = map(valueMY, 50, 210, 2, 10);

  valueIntData = PApplet.parseInt(valueFloatData);                          // Float to integer
  valueIntDText = PApplet.parseInt(valueFloatDText);
  valueIntSText = PApplet.parseInt(valueFloatSText);

  line(510, valueMY, 610, valueMY);                                 // Draw line with mouseY coordinate
  line(valueMX, 300, valueMX, 320);

  if(mouseX > 510 && mouseX < 610) {                                // Get mouseX coordinate when the mouse
    if(mouseY >= 50 && mouseY <= 210) {                             // pointer is inside the rectangle
      if(mousePressed == true) {                                    // dimensions AND mouse is pressed
        println("Speed "+ valueIntSText);                           // and save this information
        println("Coordinates: " + mouseX + "," + mouseY);           // to 'valueMY' value
        valueMY = mouseY;                                           // to 'valueMY' value
        sendString = true;
        port.write(valueMY);
      }
    }
  }
```

```
if(mouseX >= 95 && mouseX <= 205 && valueB == 255) {              // Set detection distance
  if(mouseY >= 300 && mouseY <= 320) {
    if(mousePressed == true) {
      println("Distance "+ valueIntData);
      println("Coordinates: " + mouseX + "," + mouseY);
      valueMX = mouseX;
      send = true;
    }
  }
}

if (mousePressed == true && mouseX >=40 && mouseX <=60) {         // Detection button
  if (mouseY >= 300 && mouseY <= 320) {
    println("Coordinates: " + mouseX + "," + mouseY);
    if (valueB == 255) {
      valueB = 70;
      valueBD = 150;
      valueText = 150;
      println("Detecting OFF");
      port.write(8);
    }
    else {
      valueB = 255;
      valueBD = 200;
      valueText = 255;
      valueBI = 70;
      valueBIObj2 = 70;
      valueBIObj1 = 70;
      valueBIR = 200;
      port.write(24);                                             // Disable IR mode
      delay(10);
      println("Detecting ON");
      port.write(7);
    }
  }
  delay(190);
}

if (mousePressed == true && mouseX >=40 && mouseX <=60) {        // Camera button
  if (mouseY >= 330 && mouseY <= 350) {
  println("Coordinates: " + mouseX + "," + mouseY);
    if (valueBC == 255) {
      valueBC = 70;
      valueTextC = 150;
      println("Camera OFF");
      port.write(22);
      valueCS = "OFF";
    }
    else {
      valueBC = 255;
      valueTextC = 255;
      println("Camera ON");
      port.write(21);
      valueCS = "ON";
    }
  }
  delay(200);
}

if (mousePressed == true && mouseX >=40 && mouseX <=60) {         // IR button
  if (mouseY >= 360 && mouseY <= 380) {
    println("Coordinates: " + mouseX + "," + mouseY);
    if (valueBI == 255) {
      valueBI = 70;
      valueBIObj2 = 70;
      valueBIObj1 = 70;
      valueBIR = 200;
      valueText = 150;
      port.write(24);
      println("Infrared OFF");
    }
    else {
      valueBI = 255;
      valueBIObj2 = 255;
      valueText = 255;
      valueBIR = 255;
      valueB = 70;
      valueBD = 150;
      port.write(8);                                              // Disable detect mode
      delay(10);
      println("Infrared ON");
      port.write(23);
    }
  }
  delay(190);
}

if (valueBI == 255) {
  if (mousePressed == true && mouseX >=90 && mouseX <=120) {      // IR Object 1 button
    if (mouseY >= 360 && mouseY <= 380) {
```

```
println("Coordinates: " + mouseX + "," + mouseY);
valueBIObj2 = 255;
valueBIObj1 = 70;
println("Object 1 Selected..");
port.write(25);
}
delay(200);
}

if (mousePressed == true && mouseX >=140 && mouseX <=170) {     // IR Object 2 button
  if (mouseY >= 360 && mouseY <= 380) {
    println("Coordinates: " + mouseX + "," + mouseY);
    valueBIObj2 = 70;
    valueBIObj1 = 255;
    println("Object 2 Selected..");
    port.write(26);
  }
  delay(200);
}

if (mouseX >=195 && mouseX <=235 && mouseY >= 360 && mouseY <= 380) {        // IR RUN
  valueBIR = 220;
  if (mousePressed == true)  {
    println("Coordinates: " + mouseX + "," + mouseY);
    println("RUN..");
    port.write(27);
    delay(200);
  }
}
else {
  valueBIR = 255;
}
}

/* ######################################################
   #                                                    #
   #    Communication with arduino.                     #
   #    Variable 'var' has different values depending   #
   #    on which key is pressed or which key has been   #
   #    released.                                       #
   #                                                    #
   ###################################################### */


 if (valueBI != 255 && valueB != 255) {
   if(state != pState && (key == 'w'|| keyCode == UP)) {      // This code generates
     pState = state;                                          // an int number only ONCE and sent it via
     if (valueU == 255) {                                     // serial port to destination
       var = 1;                                               // This code is for function "Forward"
       port.write(var);
       println(var);                                          // The code Generates an int number when a key is
pressed
     }
     else {                                                   // and another number (int var = 0;) when the
       var = 0;                                               // key is released
       port.write(var);
       println(var);
     }
   }

   if(state != pState && (key == 's'|| keyCode == DOWN)) {    // Same for function "Backwards"
     pState = state;
     if (valueD == 255) {
       var = 2;
       port.write(var);
       println(var);
     }
     else {
       var = 0;
       port.write(var);
       println(var);
     }
   }

   if(state != pState && (key == 'a'|| keyCode == LEFT)) {    // Same for function "Left"
     pState = state;
     if (valueL == 255) {
       var = 3;
       port.write(var);
       println(var);
     }
     else {
       var = 0;
       port.write(var);
       println(var);
     }
   }

   if(state != pState && (key == 'd'|| keyCode == RIGHT)) {   // Same for function "Right"
     pState = state;
```

```
        if (valueR == 255) {
          var = 4;
          port.write(var);
          println(var);;
        }
        else {
          var = 0;
          port.write(var);
          println(var);
        }
      }
    }
  }   // END IF
}     // END main window

public void mouseReleased() {                                    // Send detection distance data
  if(send == true) {                                           // to serial port
    port.write(valueIntData);
    sendString = true;
  }
}

/* ##########################################################
   #                                                        #
   #    Create new window for Functions. [ help ]           #
   #    This window will open as the main programs starts.  #
   #    It can't be closed until the primary window exits.  #
   #                                                        #
   ########################################################## */


public class PFrame extends Frame {
  public PFrame() {
  setBounds(250, 300, 290, 420);
  S = new Functions();
  add(S);
  S.init();
  show();
  }
}

public class Functions extends PApplet {

  public void setup() {

    size(280, 400);
  }


  public void draw() {

    background(228, 231, 237);

    fill(254, 74, 75);
    text("Functions", 110, 15);
    textSize(12);

    text("1) ' w ' key or \u2191 : Forward", 10, 50);              // Functions - Textfield
    text("2) ' s ' key or \u2193 : Backwards", 10, 70);
    text("3) ' a ' key or \u2190 : Left", 10, 90) ;
    text("4) ' d ' key or \u2192 : Right", 10, 110);
    text("5) ' 4 ' key : Camera Left", 10, 130);
    text("6) ' 6 ' key : Camera Right", 10, 150);
    text("7) ' 5 ' key : Reset Camera Positon", 10, 170);
    text("8) ' o ' key : Minimum Speed", 10, 190);
    text("9) ' p ' key : Maximum speed", 10, 210);
    text("10) D Button : Enable/Disable Ultrasonic Mode", 10, 230);
    text("11) C Button : Enable/Disable Camera", 10, 250);
    text("12) IR Button : Enable/Disable Infrared Mode", 10, 270);
    text("\u2022 Use speed box to adjust the speed.", 10, 320);
    text("\u2022 Use detection box to adjust the", 10, 340);
    text("measurement distance.", 22, 355);
    text("(Only with ultrasonic mode enabled)", 22, 370);

  } // END - second window


} //END



  static public void main(String[] passedArgs) {
    String[] appletArgs = new String[] { "IRIS_v2_6" };
    if (passedArgs != null) {
      PApplet.main(concat(appletArgs, passedArgs));
    } else {
      PApplet.main(appletArgs);
    }
  }
}
```

4.2.2 IRIS MEGA :

```
/* ##############################################################################################
   #                                                                                            #
   #     Technological Education Institute of Piraeus                                           #
   #     Department of Electronic Computer Systems                                              #
   #     250 Thivon & P.Ralli Ave.                                                              #
   #     12244 Egaleo, Athens, Greece                                                           #
   #     Tel. : +30 (210) 538-1110                                                              #
   #                                                                                            #
   #                                                                                            #
   #     by Panagiotis Athanasiou                                                               #
   #     E-mail: panos.ips@gmail.com                                                            #
   #                                                                                            #
   #     Bluetooth Module:  "IRIS"  |  Passcode: 8052                                           #
   #                                                                                            #
   #     IRIS - Arduino Mega Board - Firmware v2.6  (Mega 2560)                                 #
   #     Last Modified: 1 September 2013 @ 19:34                                                #
   ############################################################################################## */


#include <AFMotor.h>
#include <Servo.h>
#include <NewPing.h>
#include <LED.h>


#define echoPinR          30        // Set up - ultrasonic sensors
#define triggerPinR       31
#define echoPinMR         32
#define triggerPinMR      33
#define echoPinBR         34
#define triggerPinBR      35
#define echoPinB          36
#define triggerPinB       37
#define echoPinBL         42
#define triggerPinBL      43
#define echoPinML         44
#define triggerPinML      45
#define echoPinL          46
#define triggerPinL       47
#define irModeOn          22        // Set up - ir receiver pin
#define cameraS           48        // Set up - relay pin
#define photocell         A0        // Set up - photocell pin (analog)
#define responsePinObj1   A2        // Set up - responseObj1 pin  (analog)
#define responsePinObj2   A3        // Set up - responseObj2 pin  (analog)


int distance = 20;                  // Variable for detection distance
int distanceIR = 400;               // Variable for detection distance in IR mode
int motorSpeed = 70 ;               // Motor speed 70/110
int sTemp;                          // Variable for sensors loop
int photocellD;                     // Photocell sensor data
int distData;                       // Data to send to processing
int senB;                           // Variable for sensor 1 (back)
int senBL;                          // Variable for sensor 2 (back-left)
int senBR;                          // Variable for sensor 3 (back-right)
int senML;                          // Variable for sensor 4 (middle-left)
int senMR;                          // Variable for sensor 5 (middle-right)
int senL;                           // Variable for sensor 6 (left)
int senR;                           // Variable for sensor 7 (right)
int mode = 1;                       // Variable for obj1/obj2 (Processing)
int bStop = 15;                     // 15 cm distance - backwards
int enabled = 6;                    // Variable that holds array[x]
int responseObj1 = 0;               // Variable for response data (analog)
int responseObj2 = 0;               // Variable for response data (analog)
int temp;                           // Variable to calculate minimum distance
int pos;                            // Varieble for servo
int i;                              // Variable for 'for' loop
int input;                          // Variable for Serial.read();
int writeOnLoop = 0;                // Variable for writing to port while "Runing"
int started = 0;                    // Variable for 'while' loops
int offTone = 0;                    // Variable for connection test
unsigned long waitC;                // Variable for idle
unsigned long waitB;                // Variable for idle
unsigned long waitCS;               // Variable for idle
Servo camera;                       // Variable for camera
boolean cameraStatus = false;       // Camera status
LED ledS = LED (24);                // Led to indicate if sensors are workinkg - 'detection mode' led
LED ledD = LED (26);                // Led to indicate the robot has began to detecting - 'sensor enable' led
LED ledP = LED (28);                // Led working with photocell sensor
LED ledP40ma = LED(29);             // +20mA on digital pin 28, working with photocell sensor
LED connLed = LED (53);             // Led that indicates connection

AF_DCMotor motorL(3, MOTOR12_64KHZ);         // Create motor #1 (left), 64KHz pwm
AF_DCMotor motorR(4, MOTOR12_64KHZ);         // Create motor #2 (right), 64KHz pwm


void setup() {
```

```
Serial.begin(9600);                 //
Serial1.begin(14400);
motorL.setSpeed(motorSpeed);        // Set speed to motor Left
motorR.setSpeed(motorSpeed);        // Set speed to motor Right
pinMode(photocell, INPUT);          // Set photocell pin mode
pinMode(responsePinObj1, INPUT);    // Set responsePin pin mode
pinMode(responsePinObj2, INPUT);    // Set responsePin pin mode
pinMode(cameraS, OUTPUT);           // Set relay mode
pinMode(irModeOn, OUTPUT);          // Set photocell irModeOn pin mode
digitalWrite(cameraS, HIGH);        // Set relay off
digitalWrite(irModeOn, LOW);        // Set infrared pin off
camera.attach(9);                   // Servo at 9 pin
camera.write(75);                   // Servo default position
ledS.off();                         // Set Sensor led
ledD.off();                         // Set detecting led
ledP.off();                         // Set lights led
ledP40ma.off();                     // Set lights led +40mA
tone(A15,800);                      // Tone
delay(20);                          //
tone(A15,950);                      //
delay(20);                          //
tone(A15,800);                      //
delay(20);                          //
noTone(A15);                        //
delay(4000);                        //

}


void loop() {

input = Serial.read();                                  // Read from serial connection

waitC = millis();

waitCS = waitC - waitB;
if (waitCS > 1000) {
waitB = millis();
Serial.write(223);
}

if (input == 222) {
Serial1.write(1);
connLed.on();
delay(1);
connLed.off();
delay(80);
connLed.on();
delay(1);
connLed.off();
}

NewPing stopBL(triggerPinBL, echoPinBL, bStop);         // Create stop[X] sensors for backwards movement
NewPing stopBR(triggerPinBR, echoPinBR, bStop);
NewPing stopB(triggerPinB, echoPinB, bStop);

while (input == 1) {                                    // If key is 'UP' or 'w'
sTemp = Serial.read();
motorL.setSpeed(motorSpeed);
motorR.setSpeed(motorSpeed);
motorL.run(FORWARD);
motorR.run(FORWARD);

waitC = millis();

waitCS = waitC - waitB;
if (waitCS > 1000) {
waitB = millis();
Serial.write(223);
}

if (sTemp == 222) {
Serial1.write(1);
connLed.on();
delay(1);
connLed.off();
delay(80);
connLed.on();
delay(1);
connLed.off();
}

responseObj1 = analogRead(responsePinObj1);
responseObj2 = analogRead(responsePinObj2);

if (responseObj1 >= 255 && responseObj2 >= 255 && offTone == 0) {     // Test connection
motorL.run(RELEASE);
motorR.run(RELEASE);
break;
}
```

```
photocellD = analogRead(photocell);                 // Check natural lighting

if (photocellD < 20 && cameraStatus == true) {
ledP.on();
ledP40ma.on();
}
else {
ledP.off();
ledP40ma.off();
}

if(sTemp >= 50 && sTemp <= 210) {                   // Set motors speed
motorSpeed = sTemp;                                 // accorting to received
motorSpeed = map(motorSpeed, 50, 210, 50, 110);     // data
motorL.setSpeed(motorSpeed);                        //
motorR.setSpeed(motorSpeed);
}

if (sTemp == 0) {
motorL.run(RELEASE);
motorR.run(RELEASE);
break;
}
}

while (input == 2) {                                // If key is 'DOWN' or 's'
sTemp = Serial.read();
motorL.setSpeed(70);
motorR.setSpeed(70);
Serial1.write(1);

unsigned int senStopBL = stopBL.ping();
unsigned int senStopBR = stopBR.ping();
delay(5);
unsigned int senStopB = stopB.ping();

senBL = senStopBL / US_ROUNDTRIP_CM;
senBR = senStopBR / US_ROUNDTRIP_CM;
senB = senStopB / US_ROUNDTRIP_CM;

if (senBL != 0 || senBR != 0 || senB != 0) {        // If sensors have detected an object < 15 cm
motorL.run(RELEASE);
motorR.run(RELEASE);
}
else {
motorL.run(BACKWARD);
motorR.run(BACKWARD);
}

waitC = millis();

waitCS = waitC - waitB;
if (waitCS > 1000) {
waitB = millis();
Serial.write(223);
}

if (sTemp == 222) {
Serial1.write(1);
connLed.on();
delay(1);
connLed.off();
delay(80);
connLed.on();
delay(1);
connLed.off();
}

responseObj1 = analogRead(responsePinObj1);
responseObj2 = analogRead(responsePinObj2);

if (responseObj1 >= 255 && responseObj2 >= 255 && offTone == 0) {     // Test connection
motorL.run(RELEASE);
motorR.run(RELEASE);
break;
}

photocellD = analogRead(photocell);                 // Check natural lighting

if (photocellD < 20 && cameraStatus == true) {
ledP.on();
ledP40ma.on();
}
else {
ledP.off();
ledP40ma.off();
}

if(sTemp >= 50 && sTemp <= 210) {                   // Set motors speed
motorSpeed = sTemp;                                 // accorting to received
motorSpeed = map(motorSpeed, 50, 210, 70, 110);     // data
```

```
}

if (sTemp == 0) {
motorL.run(RELEASE);
motorR.run(RELEASE);
break;
}
}

while (input == 3) {                                    // If key is 'LEFT' or 'a'
sTemp = Serial.read();
motorL.setSpeed(70);
motorR.setSpeed(70);
motorL.run(BACKWARD);
motorR.run(FORWARD);
Serial1.write(1);

waitC = millis();

waitCS = waitC - waitB;
if (waitCS > 1000) {
waitB = millis();
Serial.write(223);
}

if (sTemp == 222) {
Serial1.write(1);
connLed.on();
delay(1);
connLed.off();
delay(80);
connLed.on();
delay(1);
connLed.off();
}

responseObj1 = analogRead(responsePinObj1);
responseObj2 = analogRead(responsePinObj2);

if (responseObj1 >= 255 && responseObj2 >= 255 && offTone == 0) {     // Test connection
motorL.run(RELEASE);
motorR.run(RELEASE);
break;
}

photocellD = analogRead(photocell);                 // Check natural lighting

if (photocellD < 20 && cameraStatus == true) {
ledP.on();
ledP40ma.on();
}
else {
ledP.off();
ledP40ma.off();
}

if(sTemp >= 50 && sTemp <= 210) {                   // Set motors speed
motorSpeed = sTemp;                                 // accorting to received
motorSpeed = map(motorSpeed, 50, 210, 70, 110);     // data
}

if (sTemp == 0) {
motorL.run(RELEASE);
motorR.run(RELEASE);
break;
}
}

while (input == 4) {                                    // If key is 'RIGHT' or 'd'
sTemp = Serial.read();
motorL.setSpeed(70);
motorR.setSpeed(70);
motorL.run(FORWARD);
motorR.run(BACKWARD);
Serial1.write(1);

waitC = millis();

waitCS = waitC - waitB;
if (waitCS > 1000) {
waitB = millis();
Serial.write(223);
}

if (sTemp == 222) {
Serial1.write(1);
connLed.on();
delay(1);
connLed.off();
delay(80);
connLed.on();
```

```
delay(1);
connLed.off();
}

responseObj1 = analogRead(responsePinObj1);
responseObj2 = analogRead(responsePinObj2);

if (responseObj1 >= 255 && responseObj2 >= 255 && offTone == 0) {     // Test connection
motorL.run(RELEASE);
motorR.run(RELEASE);
break;
}

photocellD = analogRead(photocell);                                   // Check natural lighting

if (photocellD < 20 && cameraStatus == true) {
ledP.on();
ledP40ma.on();
}
else {
ledP.off();
ledP40ma.off();
}

if(sTemp >= 50 && sTemp <= 210) {                                     // Set motors speed
motorSpeed = sTemp;                                                   // accorting to received
motorSpeed = map(motorSpeed, 50, 210, 70, 110);                       // data
}

if (sTemp == 0) {
motorL.run(RELEASE);
motorR.run(RELEASE);
break;
}
}

if (input == 21) {
digitalWrite(cameraS, LOW);
cameraStatus = true;
}
else if (input == 22) {
digitalWrite(cameraS, HIGH);
cameraStatus = false;
}
                                                                      // read servo's current position
pos = camera.read();

if(input == 6 && pos < 140) {                                         // when the '6' key is pressed
pos = pos + 7;                                                        // and the servos' position is less
camera.write(pos);                                                    // than 145 degress add one degree in
delay(5);                                                             // servos current positon
Serial1.write(1);
}                                                                     // same for input = 5 - '4' key on computer
else if (input == 5 && pos > 15) {
pos = pos - 7;
camera.write(pos);
delay(5);
Serial1.write(1);
}
else if (input == 9) {                                                // Servo's default position
camera.write(75);
}

if(input >= 50 && input <= 210) {                                     // Set motors speed
motorSpeed = input;                                                   // accorting to received
motorSpeed = map(motorSpeed, 50, 210, 70, 110);                       // data
motorL.setSpeed(motorSpeed);
motorR.setSpeed(motorSpeed);
}

photocellD = analogRead(photocell);
//Serial.println(photocellD);   ** DEBUG **

if (photocellD < 20 && cameraStatus == true) {
ledP.on();
ledP40ma.on();
}
else {
ledP.off();
ledP40ma.off();
}

responseObj1 = analogRead(responsePinObj1);
responseObj2 = analogRead(responsePinObj2);

if (responseObj1 >= 255 && responseObj2 >= 255 && offTone == 0) {     // Test connection
digitalWrite(cameraS, HIGH);
Serial1.write(2);
distance = 20;
motorSpeed = 70;
tone(A15,500);
```

```
delay(50);
noTone(A15);
delay(20);
tone(A15,400);
delay(50);
noTone(A15);
offTone = 1;
}
else if (responseObj1 < 255 && responseObj2 < 255 && offTone == 1) {
tone(A15,800);
delay(20);
tone(A15,950);
delay(20);
tone(A15,800);
delay(20);
noTone(A15);
offTone = 0;
}

while (input == 7) {                                    // If D button has been clicked on processing

if (input == 7 && started == 0) {
Serial1.write(3);
tone(A15,900);
delay(20);
noTone(A15);
waitB = millis();
started = 1;
}

sTemp = Serial.read();                                  // Look for input from serial connection
ledD.on();                                              // Turn on led if mode = detection  // Blink when sensors = 0
-    keep scanning
motorL.setSpeed(70);
motorR.setSpeed(70);

waitC = millis();

waitCS = waitC - waitB;
if (waitCS > 1000) {
waitB = millis();
Serial.write(223);
}

if (sTemp == 222) {
Serial1.write(1);
connLed.on();
delay(1);
connLed.off();
delay(80);
connLed.on();
delay(1);
connLed.off();
}

if(sTemp >= 10 && sTemp <= 20) {                        // Set detention distance
distance = sTemp;                                       // according to received
distance = map(distance, 10, 20, 20, 90);               // data - map function to
delay(20);                                              // calibrate distance detection
}

if (sTemp == 21) {
digitalWrite(cameraS, LOW);
cameraStatus = true;
}
else if (sTemp == 22) {
digitalWrite(cameraS, HIGH);
cameraStatus = false;
}
                                        // Read servo's current position

pos = camera.read();

if(sTemp == 6 && pos < 140) {           // When the '6' key is pressed
pos = pos + 7;                          // and the servos' position is less
camera.write(pos);                      // than 140 degress add one degree in
delay(5);                               // servos current positon
Serial1.write(1);
}
else if (sTemp == 5 && pos > 15) {      // Same for input = 5 - '4' key on computer
pos = pos - 7;
camera.write(pos);
delay(5);
Serial1.write(1);
}
else if (sTemp == 9) {
camera.write(75);                   // Servo's default position
}

if(sTemp >= 50 && sTemp <= 210) {               // Set motors speed
motorSpeed = sTemp;                             // accorting to received
motorSpeed = map(motorSpeed, 50, 210, 70, 110); // data
```

```
}

photocellD = analogRead(photocell);

if (photocellD < 20 && cameraStatus == true) {
ledP.on();
ledP40ma.on();
}
else {
ledP.off();
ledP40ma.off();
}

NewPing sB(triggerPinB, echoPinB, distance);        // Set the new detention
NewPing sBL(triggerPinBL, echoPinBL, distance);     // distance to sensors
NewPing sBR(triggerPinBR, echoPinBR, distance);
NewPing sML(triggerPinML, echoPinML, distance);
NewPing sMR(triggerPinMR, echoPinMR, distance);
NewPing sL(triggerPinL, echoPinL, distance);
NewPing sR(triggerPinR, echoPinR, distance);

unsigned int SDB = sB.ping();          // Get ping time from sensor - back
delay(5);                              // delay for power consumption
unsigned int SDBL = sBL.ping();        // Get ping time from sensor - back left
unsigned int SDBR = sBR.ping();        // Get ping time from sensor - back right
delay(5);
unsigned int SDML = sML.ping();        // Get ping time from sensor - middle left
unsigned int SDMR = sMR.ping();        // Get ping time from sensor - middle right
delay(5);
unsigned int SDL = sL.ping();          // Get ping time from sensor left
unsigned int SDR = sR.ping();          // Get ping time from sensor right

senB = SDB / US_ROUNDTRIP_CM;          // Convert ping time to distance -
senBL = SDBL / US_ROUNDTRIP_CM;        // Save these values to  Sen[X] variables.
senBR = SDBR / US_ROUNDTRIP_CM;
senML = SDML / US_ROUNDTRIP_CM;
senMR = SDMR / US_ROUNDTRIP_CM;
senL = SDL / US_ROUNDTRIP_CM;
senR = SDR / US_ROUNDTRIP_CM;

if (senB == 0 && senBL == 0 && senBR == 0 && senML == 0 && senMR == 0 && senL == 0 && senR == 0) {       // Check
if sensors don't detect anything
enabled = 7;
}
else {

int *convert[7] = {&senB, &senBL, &senBR, &senML, &senMR, &senL, &senR};                                 // If
sensor = 0  -> distance = 100
temp = 100;

for(int i = 0; i <= 6; i++) {
if(*(convert[i]) == 0) {
*convert[i] = temp;
}
}

int *sensor[7] = {&senB, &senBL, &senBR, &senML, &senMR, &senL, &senR};                                  // Find the
sensor which has detected an item closer than the others
distData = 100;

for(int i = 0; i <= 6; i++) {
if(*(sensor[i]) < distData) {
distData = *(sensor[i]);
enabled = i;
}
}
}

if (enabled < 7 && enabled >=0 ) {
ledD.off();                            // Turn off led if robot has detected an item
ledS.blink(100);
Serial.write(223);
Serial1.write(1);
Serial.write(distData);

if (enabled == 0) {                    // The sensor B has detected an object closer
motorL.run(FORWARD);
delay(5);
motorR.run(BACKWARD);
delay(1400);
motorL.run(RELEASE);
motorR.run(RELEASE);
}
else if (enabled == 1) {               // The sensor BL has detected an object closer
motorL.run(BACKWARD);
delay(5);
motorR.run(FORWARD);
delay(1250);
motorL.run(RELEASE);
motorR.run(RELEASE);
}
```

```
else if (enabled == 2) {          // The sensor BR has detected an object closer
motorL.run(FORWARD);
delay(5);
motorR.run(BACKWARD);
delay(1250);
motorL.run(RELEASE);
motorR.run(RELEASE);
}
else if (enabled == 3) {          // The sensor ML has detected an object closer
motorL.run(BACKWARD);
delay(5);
motorR.run(FORWARD);
delay(1000);
motorL.run(RELEASE);
motorR.run(RELEASE);
}
else if (enabled == 4) {          // The sensor MR has detected an object closer
motorL.run(FORWARD);
delay(5);
motorR.run(BACKWARD);
delay(1000);
motorL.run(RELEASE);
motorR.run(RELEASE);
}
else if (enabled == 5) {          // The sensor L has detected an object closer
motorL.run(BACKWARD);
delay(5);
motorR.run(FORWARD);
delay(800);
motorL.run(RELEASE);
motorR.run(RELEASE);
}
else if (enabled == 6) {          // The sensor R has detected an object closer
motorL.run(FORWARD);
delay(5);
motorR.run(BACKWARD);
delay(800);
motorL.run(RELEASE);
motorR.run(RELEASE);
}
Serial.write(223);
Serial1.write(1);                 // Time to wait = Total -> (Total + 50ms delay)
delay(2000);
Serial.write(0);
}
else if (enabled == 7) {            // All sensors are out of range
delay(100);
}

responseObj1 = analogRead(responsePinObj1);
responseObj2 = analogRead(responsePinObj2);

if (responseObj1 >= 255 && responseObj2 >= 255 && offTone == 0) {   // Test connection
ledD.off();
Serial1.write(2);
started = 0;
break;
}

if (sTemp == 8) {                   // When the D button is pressed again
ledD.off();                         // turn off the 'sensor enable' led
motorL.setSpeed(motorSpeed);
motorR.setSpeed(motorSpeed);
tone(A15,900);
delay(20);
noTone(A15);
Serial1.write(2);
started = 0;                      // exit from while loop if the button renctagle is clicked
break;
}                                   // END while
}

while (input == 23) {               // IR mode

if (input == 23 && started == 0) {
Serial1.write(4);
tone(A15,900);
delay(20);
noTone(A15);
started = 1;
}

sTemp = Serial.read();
ledD.on();
motorL.setSpeed(80);
motorR.setSpeed(80);
digitalWrite(irModeOn, HIGH);

waitC = millis();

waitCS = waitC - waitB;
```

```
if (waitCS > 1000) {
waitB = millis();
Serial.write(223);
}

if (sTemp == 222) {
Serial1.write(1);
connLed.on();
delay(1);
connLed.off();
delay(80);
connLed.on();
delay(1);
connLed.off();
}

if (sTemp == 25) {                    // Object selection (Processing)
mode = 1;
Serial1.write(5);
tone(A15,900);
delay(20);
noTone(A15);
}
else if (sTemp == 26) {
mode = 2;
tone(A15,900);
delay(20);
noTone(A15);
Serial1.write(6);
}

if (sTemp == 21) {
digitalWrite(cameraS, LOW);
cameraStatus = true;
}
else if (sTemp == 22) {
digitalWrite(cameraS, HIGH);
cameraStatus = false;
}

pos = camera.read();                      // Read servo's current position

if(sTemp == 6 && pos < 140) {             // when the '6' key is pressed
pos = pos + 7;                            // and the servos' position is less
camera.write(pos);                        // than 145 degress add one degree in
delay(5);                                 // servos current positon
Serial1.write(1);
}
else if (sTemp == 5 && pos > 15) {        // Same for input = 5 - '4' key on computer
pos = pos - 7;
camera.write(pos);
delay(5);
Serial1.write(1);
}
else if (sTemp == 9) {
camera.write(75);                       // Servo's default position
}

if(sTemp >= 50 && sTemp <= 210) {
motorSpeed = sTemp;
motorSpeed = map(motorSpeed, 50, 210, 70, 110);
}

photocellD = analogRead(photocell);

if (photocellD < 20 && cameraStatus == true) {
ledP.on();
ledP40ma.on();
}
else {
ledP.off();
ledP40ma.off();
}

if (sTemp == 27) {                                          // IR RUN
Serial1.write(7);                                           // analog pins - set'0'
delay(800);
waitC = millis();
responseObj1 = analogRead(responsePinObj1);
responseObj2 = analogRead(responsePinObj2);

while (responseObj1 < 255 && responseObj2 < 255) {          // No detect
motorL.run(BACKWARD);
motorR.run(FORWARD);
waitB = millis();

if ((waitB - waitC > 1000) && (waitB - waitC < 2000) && writeOnLoop == 0) {
Serial.write(223);
Serial1.write(1);
writeOnLoop = 1;
```

```
}

if ((waitB - waitC >= 2000) && writeOnLoop == 1) {
Serial.write(223);
Serial1.write(1);
writeOnLoop = 0;
}

if (waitB - waitC > 2400) {
motorL.run(RELEASE);
motorR.run(RELEASE);
break;
}

responseObj1 = analogRead(responsePinObj1);
responseObj2 = analogRead(responsePinObj2);

if ((responseObj1 >= 255 && mode == 1) || (responseObj2 >= 255 && mode == 2)) {
motorL.run(RELEASE);
motorR.run(RELEASE);
writeOnLoop = 0;
break;
}
}

if ((responseObj1 >= 255 && mode == 1) || (responseObj2 >= 255 && mode == 2)) {                // If IR found
NewPing sBIR(triggerPinB, echoPinB, distanceIR);                                               // set measure
distance
unsigned int SDBIR = sBIR.ping();
senB = SDBIR / US_ROUNDTRIP_CM;
distData = senB;
if (distData > 0) {
ledD.off();
ledS.blink(100);
Serial.write(distData);                                    // then write this distance to serial port
motorL.run(FORWARD);                                       // drive motors
delay(5);
motorR.run(BACKWARD);
Serial.write(223);
Serial1.write(1);
delay(1400);
motorL.run(RELEASE);                                       // release motors
motorR.run(RELEASE);
//sTemp = 0;                                                 // end exit while loop
}
}                                                          // END IF
}

responseObj1 = analogRead(responsePinObj1);
responseObj2 = analogRead(responsePinObj2);

if (responseObj1 >= 255 && responseObj2 >= 255) {          // Test connection
ledD.off();
digitalWrite(irModeOn, LOW);
Serial1.write(2);
writeOnLoop = 0;
started = 0;
break;
}

if (sTemp == 24) {                                         // EXIT
ledD.off();                                                // Set motorspeed
motorL.setSpeed(motorSpeed);
motorR.setSpeed(motorSpeed);
digitalWrite(irModeOn, LOW);
tone(A15,900);
delay(20);
noTone(A15);
Serial1.write(2);
started = 0;
break;
}
}                                                                  // END while

}                                                                  // END
```

4.2.3 IRIS NANO :

```
/* ##########################################################################################################
   #
   #     Technological Education Institute of Piraeus                                                       #
   #     Department of Electronic Computer Systems                                                          #
   #     250 Thivon & P.Ralli Ave.                                                                          #
   #     12244 Egaleo, Athens, Greece                                                                       #
   #     Tel. : +30 (210) 538-1110                                                                          #
   #                                                                                                        #
   #                                                                                                        #
   #     by Panagiotis Athanasiou                                                                           #
   #     E-mail: panos.ips@gmail.com                                                                        #
   #                                                                                                        #
   #     Bluetooth Module:  "IRIS"  |  Passcode: 8052                                                       #
   #                                                                                                        #
   #     IRIS - Arduino Nano Board - Firmware v1 (ATmega328 / Nano 3.0)                                     #
   #     Last Modified: 3 August 2013 03:26                                                                 #
   ########################################################################################################## */

#define _Digole_Serial_UART_
#include <DigoleSerial.h>
#include <Wire.h>
#include <IRremote.h>

#define irRecv       4      // Set up reveiver pin
#define readyPin     5      // Set up ready pin
#define obj1         A4     // Set up response obj 1 (analog)
#define obj2         A3     // Set up response obj 2 (analog)

DigoleSerialDisp disp(&Serial, 14400);

int i = 0;
int online = 0;
int splash = 0;
int uScr = 0;
int sObj1 = 0;
int sObj2 = 0;
int resetMode = 0;
int irData;
int decCode;
int ready;
int func;
unsigned long waitD;
unsigned long wait;
unsigned long waitC;
String infrared = "Infrared Mode";
String ultrasonic = "Ultrasonic Mode";
String manual = "Manual Mode";
String tObj1 = "Target: Object 1";
String tObj2 = "Target: Object 2";


IRrecv irrecv(irRecv);      // Create reveiver object

decode_results results;     // Incoming data


void setup() {
Serial.begin(14400);
disp.begin();
irrecv.enableIRIn();
pinMode(readyPin, INPUT);
pinMode(irRecv, INPUT);
analogWrite(obj1, 0);
analogWrite(obj2, 0);
disp.clearScreen();
disp.setLCDColRow(16,2);
disp.disableCursor();
disp.displayConfig(0);
disp.displayStartScreen(0);
disp.clearScreen();
disp.setFont(10);
disp.drawStr(0,0,"Booting up...");
delay(1000);
disp.clearScreen();
disp.setFont(10);

for (i = 0; i >= -16; i--) {
disp.drawStr(0, i, "Arduino Mega board:");
disp.drawStr(0, i + 1, "Controller ATmega2560");
disp.drawStr(0, i + 2, "Flash Memory  256 KB ");
disp.drawStr(0, i + 3, "8 KB by bootloader  ");
disp.drawStr(0, i + 4, "Clock Speed   16 MHz");
disp.drawStr(0, i + 5, "                    ");
disp.drawStr(0, i + 6, "Arduino Nano board:");
disp.drawStr(0, i + 7, "Controller  ATmega328");
disp.drawStr(0, i + 8, "Flash Memory  32 KB  ");
```

```
disp.drawStr(0, i + 9, "2 KB by bootloader ");
disp.drawStr(0, i + 10, "Clock Speed  16 MHz");
disp.drawStr(0, i + 11, "                   ");
disp.drawStr(0, i + 12, "Bluetooth v2.0");
disp.drawStr(0, i + 13, "@ 2.4GHz, Class 2");
disp.drawStr(0, i + 14, "Ultrasonic HC-SR04");
disp.drawStr(0, i + 15, "Infrared:          ");
disp.drawStr(0, i + 16, "Frequency  38Khz");
disp.drawStr(0, i + 17, "Camera @ 2.4 Ghz");
disp.drawStr(0, i + 18, "Motors @ 64Khz  ");
disp.drawStr(0, i + 19, "Power:             ");
disp.drawStr(0, i + 20, "12 Volt DC 6800 mAh");
disp.drawStr(0, i + 21, "Baud Rate @ 9600 bps");
disp.drawStr(0, i + 22, "UART Baud @ 14400 bps");
delay(100);
}

delay(1500);
waitC = millis();
disp.clearScreen();
disp.setFont(10);
disp.setColor(1);
disp.drawStr(1,2,"Project - IRIS v2.6");
disp.drawStr(1,4,"by Panos Athanasiou");
delay(1000);

}

void loop() {

wait = millis();

if (Serial.available() > 0) {
waitC = millis();
}

waitD = wait - waitC;
waitD = abs(waitD);
if (waitD > 5000 && waitD < 16500) {
analogWrite(obj1, 255);
analogWrite(obj2, 255);
disp.clearScreen();
disp.drawStr(6, 0, "IRIS");
disp.setPrintPos(0, 1, _TEXT_);
disp.setFont(10);
disp.drawStr(2, 2, "Connection offline");
delay(700);
disp.setMode('~');
disp.drawBox(4, 15, 122, 12);
delay(700);
online = 0;
splash = 0;
}

if (waitD >= 18500 && splash == 0) {
disp.clearScreen();
disp.setFont(10);
disp.setColor(1);
disp.drawStr(1,2,"Project - IRIS v2.6");
disp.drawStr(1,4,"by Panos Athanasiou");
delay(1000);
splash = 1;
online = 0;
}

func = Serial.read();
if (func == 1 && online == 0 && waitD <= 5000) {
disp.clearScreen();
disp.drawStr(6, 0, "IRIS");
disp.setPrintPos(0, 1, _TEXT_);
disp.setFont(10);
disp.setMode('~');
disp.drawBox(4, 15, 120, 12);
disp.drawStr(2, 2, "Connection online!");
analogWrite(obj1, 0);
analogWrite(obj2, 0);
delay(2500);
disp.clearScreen();
disp.drawStr(6, 0, "IRIS");
disp.setFont(10);
disp.setPrintPos(0, 3, _TEXT_);
disp.print(manual);
//disp.setPrintPos(0, 6, _TEXT_);
//disp.print(cameraStatus);
online = 1;
uScr = 0;
delay(1000);
}
else if (func == 3 && uScr == 0 && waitD <= 5000) {
disp.clearScreen();
disp.drawStr(6, 0, "IRIS");
```

```
disp.setFont(10);
disp.setPrintPos(0, 3, _TEXT_);
disp.print(ultrasonic);
//disp.setPrintPos(0, 6, _TEXT_);
//disp.print(cameraStatus);
delay(1000);
uScr = 1;
}
else if (func == 2 && waitD <= 5000) {
disp.clearScreen();
disp.drawStr(6, 0, "IRIS");
disp.setFont(10);
disp.setPrintPos(0, 3, _TEXT_);
disp.print(manual);
//disp.setPrintPos(0, 6, _TEXT_);
//disp.print(cameraStatus);
delay(1000);
uScr = 0;
}


ready = digitalRead(readyPin);

while (ready == HIGH) {

if (resetMode == 0) {
disp.clearScreen();
disp.drawStr(6, 0, "IRIS");;
disp.setFont(10);
disp.setPrintPos(0, 3, _TEXT_);
disp.print(infrared);
disp.setPrintPos(0, 4, _TEXT_);
disp.print(tObj1);
//disp.setPrintPos(0, 6, _TEXT_);
//disp.print(cameraStatus);
delay(1000);
sObj1 = 1;
sObj2 = 0;
resetMode = 1;
waitC = millis();
}

wait = millis();

if (Serial.available() > 0) {
waitC = millis();
}

waitD = wait - waitC;
waitD = abs(waitD);
if (waitD > 5000) {
analogWrite(obj1, 255);
analogWrite(obj2, 255);
//online = 0;
//splash = 0;
break;
}

func = Serial.read();

if (func == 7) {
analogWrite(obj1, 0);
analogWrite(obj2, 0);
delay(25);
}

if (func == 5 && sObj1 == 0 && waitD <= 5000) {
disp.clearScreen();
disp.drawStr(6, 0, "IRIS");;
disp.setFont(10);
disp.setPrintPos(0, 3, _TEXT_);
disp.print(infrared);
disp.setPrintPos(0, 4, _TEXT_);
disp.print(tObj1);
//disp.setPrintPos(0, 6, _TEXT_);
//disp.print(cameraStatus);
delay(1000);
sObj1 = 1;
sObj2 = 0;
}
else if (func == 6 && sObj2 == 0 && waitD <= 5000) {
disp.clearScreen();
disp.drawStr(6, 0, "IRIS");
disp.setFont(10);
disp.setPrintPos(0, 3, _TEXT_);
disp.print(infrared);
disp.setPrintPos(0, 4, _TEXT_);
disp.print(tObj2);
//disp.setPrintPos(0, 6, _TEXT_);
//disp.print(cameraStatus);
delay(1000);
```

```
sObj1 = 0;
sObj2 = 1;
}

if (irrecv.decode(&results)) {          // Decode data
//Serial.println(results.value, HEX);
irrecv.resume();
}

if (results.value == 0x7A4C) {          // Obj1
analogWrite(obj2, 0);
analogWrite(obj1, 255);
}

if (results.value == 0x50DC) {          // Obj2
analogWrite(obj1, 0);
analogWrite(obj2, 255);
}


results.value = 0;

ready = digitalRead(readyPin);

if (ready == LOW) {
analogWrite(obj1, 0);
analogWrite(obj2, 0);
sObj1 = 0;
sObj2 = 0;
uScr = 0;
resetMode = 0;
break;

}
}                                       // END while

}
```

4.2.4 IRIS UNO :

```
/* ##########################################################################################################
   #                                                                                                        #
   #      Technological Education Institute of Piraeus                                                      #
   #      Department of Electronic Computer Systems                                                         #
   #      250 Thivon & P.Ralli Ave.                                                                         #
   #      12244 Egaleo, Athens, Greece                                                                      #
   #      Tel. : +30 (210) 538-1110                                                                         #
   #                                                                                                        #
   #                                                                                                        #
   #      by Panagiotis Athanasiou                                                                          #
   #      E-mail: panos.ips@gmail.com                                                                       #
   #                                                                                                        #
   #                                                                                                        #
   #      IRIS - IR Sender Firmware v1.1                                                                    #
   #      Last Modified: 3 March 2013 03:26                                                                 #
   #                                                                                                        #
   ########################################################################################################## */

#include <IRremote.h>
#define relay 2
#define buttonPin 4


int state = HIGH;
int pState = LOW;
int button;
long time = 0;          // the last time the output pin was toggled
long debounce = 200;    // the debounce time, increase if the output flickers
IRsend irsend;

void setup() {

pinMode(3, OUTPUT);
pinMode(relay, OUTPUT);
pinMode(button, INPUT);
digitalWrite(relay, LOW);
irsend.enableIROut(38);
irsend.mark(0);
}

void loop() {

button = digitalRead(buttonPin);
if (button == HIGH && pState == LOW && millis() - time > debounce) {
if (state == HIGH) {
state = LOW;
}
else {
state = HIGH;
}
time = millis();
}

digitalWrite(relay, state);

pState = button;

if (state == LOW) {
irsend.sendRC5(0x7A4C, 16); // 7A4C HEX code - Wrobot module
}
else if (state == HIGH) {
irsend.sendRC5(0x50DC, 16); // 50DC HEX code -  module
}
delay(25);

}
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

nxtepalchioua.weebly.com

el.wikipedia.org

www.digole.com

images.google.com

grobot.gr

amazon.de

benhatke.com

instructables.com

arduino.cc

processing.org

unipi.gr

hys.teipir.gr

code.google.com