



Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Αυτοματισμού

Πτυχιακή Εργασία

Θέμα:

**«Υπερηχητικό αυτοκίνητο με ασύρματη – αυτόματη
διαχείριση συσκευών»**

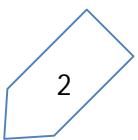


ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΠΑΠΟΥΤΣΙΔΑΚΗΣ ΜΙΧΑΛΗΣ

ΦΟΙΤΗΤΗΣ: Καττές Νικόλαος

A.M.:35152

Αιγάλεω, Μάιος 2013



Περιεχόμενα

Πρόλογος	5
Κεφάλαιο 1: Μηχατρονική	6
1.1 Φιλοσοφία της Μηχατρονικής	6
1.2 Ιστορία	7
1.3 Ορισμοί Μηχατρονικής	7
1.4 Αντικείμενο	7
1.5 Μηχατρονικό Σύστημα	8
1.6 Παραδείγματα Μηχατρονικών Συστημάτων	8
1.7 Εφαρμογές	10
1.8 Γενικότερες Εφαρμογές	11
1.9 Παραλλαγές Μηχατρονικής	11
Κεφάλαιο 2: Η Κατασκευή	14
2.1 Η Λειτουργία	14
2.2 Τα εξαρτήματα	14
2.2.1 Ο Μικροελεγκτής	14
2.2.2 Οι σερβοκινητήρες	16
2.2.3 Αισθητήρες απόστασης	19
2.2.4 Αισθητήρας θερμοκρασίας	20
2.2.5 Ασύρματη πρίζα	21
2.2.6 Reed Relay	23
2.2.7 Οθόνη LCD 2x16 5VDC	23
2.2.8 RGB ταινία LED	24
2.2.9 LDR (Light Dependent Resistor)	25
2.2.10 Transistors	25
2.2.11 Τροφοδοσία σερβοκινητήρων	26
2.2.12 Γενική τροφοδοσία	27
Κεφάλαιο 3: Ο Προγραμματισμός	29
Κεφάλαιο 4: Μελλοντικές Βελτιώσεις	50
4.1 Επικοινωνία μέσω Bluetooth	50

4.2 Αναπαραγωγή ήχου	52
4.3 Σύνδεση μέσω WiFi	53
4.4 Παρακολούθηση χώρου	54

Βιβλιογραφία	57
---------------------	----

Πρόλογος

Στόχος αυτής της πτυχιακής καθώς και του παρόντος βιβλίου, είναι να δείξει πώς επιτυγχάνεται ο έλεγχος σχεδόν οποιουδήποτε συστήματος, με σκοπό την αυτοματοποίηση και τη διευκόλυνση της ζωής ή των εργασιών μας.

Έτσι λοιπόν, με γνώμονα την τεχνογνωσία, θα εξετάσουμε και θα αναλύσουμε έως ενός ορισμένου σημείου, τα ηλεκτρονικά εξαρτήματα καθώς και τον κώδικα που χρησιμοποιούμε, προκειμένου να κατανοήσουμε σε βάθος τον τρόπο υλοποίησης αυτής της εργασίας.

Θα ξεκινήσουμε όμως, κάνοντας μια μικρή περίληψη – αναδρομή στο γνωσιακό υπόβαθρο που κρύβεται πίσω από τέτοιες κατασκευές, τη μηχανική.

Κεφάλαιο 1

Μηχατρονική

Ο όρος **Μηχατρονική** είναι ένας σύγχρονος νεολογισμός που υποδηλώνει τον συνδυασμό των επιστημών της Μηχανολογίας, Ηλεκτρονικής - Ηλεκτρολογίας και Πληροφορικής.

Μηχατρονική: = Μηχανολογία + Ηλεκτρονική + Πληροφορική

Εφάμιλλος όρος για την Μηχατρονική είναι η Τεχνική Κυβερνητική - Technical Cybernetics.

Σπανίως χρησιμοποιείται και ο όρος Μηχανοτρονική ή Ηλεκτρομηχανολογικοί Αυτοματισμοί.

Η Μηχατρονική θεωρείται εμπλουτισμός των κατά βάση μηχανολογικών συστημάτων με ηλεκτρονικά εξαρτήματα που αρκετά συχνά εμπεριέχουν λογισμικό, δηλαδή:

Μηχατρονική είναι η συνεργεία τριών επιστημών

- Μηχανολογία
- Ηλεκτρονική/Ηλεκτρολογία
- Πληροφορική

με σκοπό την δημιουργία συστημάτων που να απλοποιούν την παραγωγή.

1.1 Φιλοσοφία της Μηχατρονικής

Στην φιλοσοφία της Μηχατρονικής, ο ενσωματωμένος υπολογιστής ελέγχου είναι το κεντρικό στοιχείο, και ο πυρήνας της τεχνολογίας η οποία την καθιστά την Μηχατρονική ένα μοναδικό τομέα. Ψηφιακά και αναλογικά κυκλώματα, μαζί με επενεργητές και επιστημονικά όργανα περιβάλλουν άμεσα τον υπολογιστή ελέγχου και λειτουργούν προσαρμοστικά μεταξύ του υπολογιστή και του ελεγχόμενου φυσικού συστήματος. Τα χαρακτηριστικά που διαφοροποιούν το κάθε σύγχρονο μηχανικό σύστημα, καθορίζονται σε μεγάλο βαθμό από την εφευρετικότητα και αποτελεσματικότητα του ενσωματωμένου σε αυτό λογισμικό. Τα παρεμβαλλόμενα στοιχεία υποστηρίζουν το λογισμικό αυτό παρέχοντας του τις τρέχουσες πληροφορίες από το ελεγχόμενο σύστημα και μεταφράζοντας τις εντολές του σε ενεργή παροχή διαμορφωμένης ισχύος.

1.2 Ιστορία

Η Μηχατρονική επικεντρώνεται στη μηχανική, την ηλεκτρονική, την μηχανολογία συστημάτων ελέγχου, των ηλεκτρονικών υπολογιστών, τη μοριακή μηχανική (από ναυοχημεία και Βιολογία), η οποία και σε συνδυασμό μεταξύ των, να καταστήσει την παραγωγή:

- απλούστερη,
- πιο οικονομική,
- αξιόπιστη και ευέλικτη.

Ο όρος "Μηχανοτρονική" επινοήθηκε για πρώτη φορά από τον Tetsuro Mori, ανώτερο μηχανικό της ιαπωνικής εταιρείας Yaskawa, το 1969. Η Μηχατρονική εναλλακτικά, μπορεί να αναφέρεται και ως η Επιστήμη των "Ηλεκτρομηχανολογικών συστημάτων" ή λιγότερο συχνά ως η Επιστήμη "ελέγχου και του αυτοματισμού της μηχανικής".

Το 1982 επιτρέπεται από την εταιρεία η ελεύθερη χρήση του όρου.

Η Μηχατρονική αποτελεί το άμεσο εκείνο υπόβαθρο για την έρευνα στο τεχνικό τομέα της Κυβερνητικής. Σημαντικές φυσιογνωμίες και χρονολογίες σταθμοί στην Κυβερνητική και κατ' επέκταση στην Μηχατρονική υπήρξαν το 1936 από τον Άλαν Τούρινκ το 1948 από τον Νόρμπερτ Βίνερ και Μόρθυ (Morthy), με τις μηχανές ψηφιακού ελέγχου, που αρχικά αναπτύχθηκαν το 1946 ο Τηλεχειρισμός το 1951 από τον Γκερτζ (Goertz) καθώς και η ανώνυμη εταιρεία *Bedford Associates* που αναπτύχθηκε το 1968.

1.3 Ορισμοί Μηχατρονικής

Επί του παρόντος υπάρχουν διάφοροι ορισμοί της Μηχατρονικής, ανάλογα με την περιοχή ενδιαφέροντος. Ειδικότερα, η UNESCO ορίζει για την Μηχατρονική ότι είναι:

"Η συνεργιακή ολοκλήρωση της μηχανολογίας με την ηλεκτρονική και τον ευφυή υπολογιστή ελέγχου στον σχεδιασμό και την κατασκευή των προϊόντων και διαδικασιών."

Ωστόσο ένας πιο ενδιαφέρων ορισμός είναι ότι Μηχατρονική είναι: ***"Η Μελέτη και κατασκευή των ευφυών μηχανικών συστημάτων."***

Κάτω από αυτή τη θεώρηση, η Μηχατρονική μπορεί να ερμηνευθεί ως ***"Η εφαρμογή πολύπλοκης διαδικασίας λήψης αποφάσεων κατά τη λειτουργία φυσικών συστημάτων."***

1.4 Αντικείμενο

Η Μηχατρονική όπως προαναφέρθηκε, πρόκειται να συγχωνεύσει τις πιο πάνω επιστήμες και να περιγράψει αντί διάφορων προτύπων ένα γενικό ολιστικό Μηχατρονικό σύστημα.

Τα συστήματα της Μηχατρονικής έχουν το στόχο να μετατρέψουν με την τεχνολογία που τα διέπει:

- Επεξεργαστές
- Ενεργοποιητές
- Αισθητήρες κτλ

την μορφή της ενέργειας αλλά και των υλικών, την μεταφορά τους και την περαιτέρω επεξεργασία τους καθώς και τη μεταφορά ή/και αποθήκευση των πληροφοριών.

1.5 Μηχατρονικό Σύστημα

Ένα Μηχατρονικό σύστημα αποτελείται κυρίως από μηχανισμούς

- Κίνησης,
- Ελέγχου και
- Αισθητήρες.

Η παραδοσιακή Μηχανική αποτελείται μόνο από μηχανισμούς και ενεργοποιητές, και προαιρετικά μπορεί να ενσωματωθεί ο έλεγχος. Η Μηχατρονική ενσωματώνει όλες τις απαιτούμενες προϋποθέσεις για έλεγχο κλειστού βρόχου και ως εκ τούτου και τους ανάλογους αισθητήρες

Ένα Μηχατρονικό σύστημα είναι ένα σύστημα το οποίο ενσωματώνει την ψηφιακή επεξεργασία σήματος και την έκδοση του σήματος αυτού σε ένα τελικό σημείο δράσης μέσω ενός ενεργοποιητή, δημιουργώντας κινήσεις ή ενέργειες σχετικά με το σύστημα. Είναι ένα ολοκληρωμένο σύστημα με αισθητήρες, μικροεπεξεργαστές και ελεγκτές.

Τα συστήματα Μηχατρονικής μπορούν να διαιρεθούν έτσι σε **ομάδες λειτουργίας**, να διαμορφωθούν σε εκείνους τους βρόχους αυτόματου ελέγχου και να αποτελέσουν μέρος των ενότητων με τα μηχανικά - ηλεκτρικά - μαγνητικά - θερμικά - οπτικά στοιχεία τους και την τεχνολογία αισθητήρων, με σκοπό τη συλλογή των μετρημένων μεταβλητών της επιβλέπουσας κατάστασης, την ενεργοποίηση την κανονικοποίηση και τον έλεγχο καθώς επίσης και επεξεργασία και την πληροφορική στην επεξεργασία δεδομένων.

1.6 Παραδείγματα Μηχατρονικών Συστημάτων

- 1) Χειρισμός/συστήματα ρομπότ
- 2) Ενόητες εργαλειομηχανών
- 3) Ψηφιακές φωτογραφικές μηχανές

- 4) Κίνηση και έλεγχος φορέα CD/DVD Player
- 5) Ανεμογεννήτριες
- 6) Αντιολισθητικά συστήματα
- 7) Ηλεκτρονικά προγράμματα σταθερότητας οχημάτων

Ειδικά για τα δύο τελευταία παραδείγματα, τα ηλεκτρονικά και το λογισμικό αντικαθιστούν τα μικρότερης ακρίβειας, πιο ευπαθή και πιο ακριβά αναλογικά μηχανικά συστήματα με ψηφιακό ηλεκτρονικό έλεγχο, όπως τα συστήματα αντιεμπλοκής πέδησης (ABS) τα συστήματα ελέγχου μείγματος, προπορείας, σπινθρισμού (ECU) και τα συστήματα ελέγχου ολίσθησης (ASP/ESP) στα αυτοκίνητα. Παγκοσμίως η μηχαντρονική είναι αντικείμενο ειδίκευσης μηχανολόγων ή μηχανικών παραγωγής.

Παρόμοιες σημερινές κατασκευές βρίσκουμε στους παρακάτω συνδέσμους. Ωστόσο καμία από αυτές τις κατασκευές δεν περιλαμβάνει το σύνολο των δυνατοτήτων της παρούσας εργασίας.

Αποφυγή εμποδίων με χρήση υπερηχητικού αισθητήρα:

1. <http://www.instructables.com/id/Clusterbot/>
2. <http://www.youtube.com/watch?v=TmR5C1drLio>
3. <http://www.youtube.com/watch?v=-6966OYBvBQ>
4. <http://www.youtube.com/watch?v=Nvz02bf1IQ4>
5. <http://www.youtube.com/watch?v=EywxNUAm3iY>
6. <http://www.youtube.com/watch?v=wkAZItadw9U>
7. <http://www.youtube.com/watch?v=LyXm0aabdnM>

Έλεγχος θερμοκρασίας με χρήση του αισθητήρα θερμοκρασίας DS18B20:

1. <http://www.krekr.nl/content/arduino-nanode-thermostat-alpha/>
2. <http://www.modsbyus.com/diy-arduino-thermostat/>
3. <http://www.robertoizerillo.com/wordpress/?p=74>

1.7 Εφαρμογές

Η απλούστερη εφαρμογή αφορά στην δυναμική ανάλυση ενός μηχανικού συστήματος και τον (ενεργό, ημί - ενεργό ή παθητικό) έλεγχό του.

Οι πιο σημαντικές εφαρμογές της Μηχατρονικής είναι η ρομποτική, τα συστήματα μεταφορών, συστήματα παραγωγής, μηχανές CNC, και οι βιομηχαντικές νανομηχανές. Η τελειότερη όμως εφαρμογή της Μηχατρονικής είναι το Ρομπότ.

Πρέπει να σημειωθεί ότι η Ρομποτική είναι κλάδος της Μηχατρονικής.

Ρομποτική είναι η τέχνη του σχεδιασμού και της κατασκευής επαναπρογραμματιζομένων στοιχείων - συσκευών ευέλικτων και ικανών να εκτελούν διάφορες λειτουργίες. Το επίπεδο του αυτοματισμού είναι πολύ πιο ευέλικτο και δείχνει τις μελλοντικές τάσεις στην υπόλοιπη μηχατρονική.

Η εφαρμογή των μηχατρονική στη μεταφορά λαμβάνει χώρα κατά το σχεδιασμό των ενεργητικών μηχανισμούς (π.χ. ενεργός αναστολή), τους κραδασμούς ελέγχου, μηχανισμούς σταθεροποίησης και αυτόνομη πλοήγηση.

Στην κατασκευή, η Μηχατρονική έχει χρησιμοποιηθεί για μοντέλα διακριτών κατά περίπτωση συστημάτων και έχει υποβάλει αίτηση για το βέλτιστο σχεδιασμό των γραμμών παραγωγής, καθώς και τη βελτιστοποίηση των υφιστάμενων διαδικασιών. Επίσης, έχει συμβάλει στην αυτοματοποίηση των γραμμών παραγωγής και τη δημιουργία της έννοιας της ευέλικτης κατασκευής.

Μηχατρονική είναι η ιστορία του ψηφιακού ελέγχου μηχανών. Σε αυτό το θέμα τις τελευταίες εξελίξεις είναι οι εξής: της ανάλυσης, ανίχνευσης και ελέγχου των κραδασμών και της θερμοκρασίας στην εργαλεία κοπής, των μεθόδων διάγνωσης και εργαλεία κοπής ταχεία προτυποποίηση, EDM λείζερ και σύνθεση.

Στο πεδίο αυτό γίνεται σύντομη εισαγωγή στην προσομοίωση δυναμικών μηχανικών συστημάτων, στον έλεγχο κατασκευών και ιδιαίτερα στην χρήση μεθόδων **Ανάλυσης και Μοντελοποίησης**:

- ευφυούς ελέγχου και συγκεκριμένα, σε μεθόδους οι οποίες στηρίζονται σε
- ασαφή λογική,
- νευρωνικά δίκτυα
- συναφείς υβριδικές τεχνικές.
- γενετικούς αλγορίθμους

Η βασική γνώση της τεχνικής δυναμικής καθώς και η δυνατότητα τουλάχιστον χρήσης ηλεκτρονικού υπολογιστή θεωρούνται αναγκαία .

Καθώς τα συστήματα αυτά σπάνια πληρούν τις προϋποθέσεις μιας μελέτης , το μαθηματικό μοντέλο που χρησιμοποιείται είναι πολύπλοκο (μη-γραμμικό), έχει ατέλειες κτλ. Για αυτο και χρησιμοποιούμε τα προαναφερθέντα ευφυή συστήματα ελέγχου .

1.8 Γενικότερες εφαρμογές

- Αυτοματισμού, και στον τομέα της ρομποτικής
- Σερβοϋδραυλική μηχανική
- Αισθητήρες και συστήματα ελέγχου
- Αυτοκίνητο Βιομηχανίες, στη σχεδίαση των υποσυστημάτων, όπως η αντι-εμπλοκή κατά την πέδηση
- Μηχανικών Ηλεκτρονικών Υπολογιστών, του σχεδιασμού μηχανισμών, όπως οι οδηγοί δίσκων

1.9 Παραλλαγές Μηχατρονικής

Μια παραλλαγή του αναδυόμενου αυτού τομέα είναι η *biomechatronics* - Βιομηχατρονική, σκοπός της οποίας είναι η ενσωμάτωση μηχανικών μερών με ένα ανθρώπινο ον, συνήθως με τη μορφή των αποσπώμενων συσκευών όπως *exoskeleton*. Αυτή είναι η "πραγματική ζωή" έκδοση του *cyberware*.

Η Βιομηχατρονική είναι η εφαρμογή της μηχατρονικής για την επίλυση των προβλημάτων των βιολογικών συστημάτων, και ιδίως την ανάπτυξη νέων τύπων προθέσεων, χειρουργικών προσομοιωτών, τον έλεγχο της θέσης των ιατρικών πράξεων (π.χ. καθετήρες), αναπηρικές πολυθρόνες και χειρουργικές τηλεχειρισμός

Επίσης η νανομηχανική είναι ένας τομέας που έχει επωφεληθεί από τις εξελίξεις στη μηχανική. Ένα πολύ προφανές παράδειγμα είναι η ανάπτυξη του σκληρού δίσκου.

Κεφάλαιο 2

Η Κατασκευή

Η κατασκευή είναι σίγουρα το δυσκολότερο κομμάτι της πτυχιακής. Απαιτεί γνώσεις, προσωπική προσπάθεια και χρόνο. Οι ελλειπείς γνώσεις οδηγούν σε λάθος αποφάσεις. Έτσι λοιπόν θα εξετάσουμε τη λειτουργία, τα εξαρτήματα που χρησιμοποιούμε καθώς και τι επιτυγχάνουμε με τη χρήση καθενός από αυτά.

2.1 Η Λειτουργία

Η βάση είναι ένα μικρό αυτοκίνητο στο οποίο θα βάλουμε κινητήρες με αρκετή δύναμη ροπής, θα προσαρμόσουμε αισθητήρες και θα βάλουμε την κατάλληλη τροφοδοσία. Ο στόχος είναι να κινείται αποφεύγοντας τα εμπόδια μπροστά ή πίσω του – όταν κάνει όπισθεν. Σε όλη τη διάρκεια της λειτουργίας του μετρά τη θερμοκρασία και τη δείχνει μέσω της RGB ταινίας LED η οποία αλλάζει χρώματα σε καθορισμένες στάθμες. Αν η θερμοκρασία ξεπεράσει τους 25° C, τότε ενεργοποιεί μέσω ασύρματης πρίζας, ένα σύστημα ψύξης, π.χ. ανεμιστήρα. Όταν ο χώρος δροσίσει τότε απενεργοποιείται το σύστημα ψύξης. Τέλος η οθόνη LCD μας δείχνει τις απαραίτητες πληροφορίες.

2.2 Τα εξαρτήματα

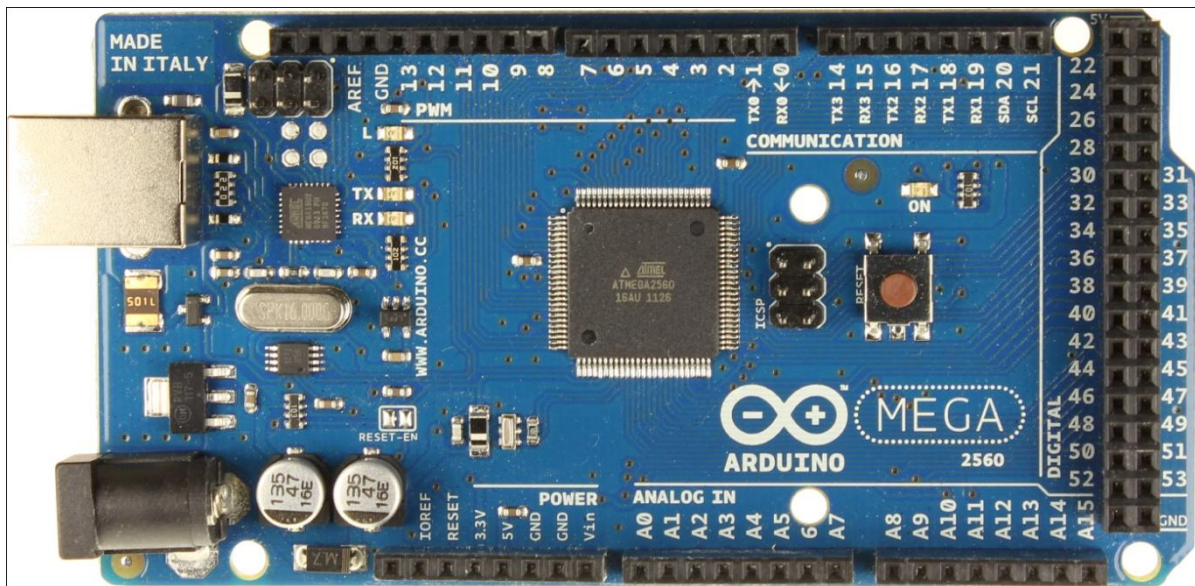
Έχοντας πάντα στο νου την επιθυμητή λειτουργία της εργασίας αυτή, ας δούμε ένα ένα όλα τα στοιχεία που συνθέτουν την κατασκευή μας.

2.2.1 Ο Μικροελεγκτής

Ο μικροελεγκτής είναι ένας τύπος επεξεργαστή, ουσιαστικά μια παραλλαγή μικροεπεξεργαστή, ο οποίος μπορεί να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα, λόγω των πολλών ενσωματωμένων υποσυστημάτων που διαθέτει. Χρησιμοποιείται ευρύτατα σε όλα τα ενσωματωμένα συστήματα ελέγχου χαμηλού και μεσαίου κόστους, όπως αυτά που χρησιμοποιούνται σε αυτοματισμούς, ηλεκτρονικά καταναλωτικά προϊόντα (από ψηφιακές

φωτογραφικές μηχανές έως παιχνίδια), ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα.

Ο μικροελεγκτής είναι η καρδιά της κατασκευής μας. Χρησιμοποιούμε τον Arduino Mega 2560 που περιέχει τον 8-bit ATmega2560, 54 ψηφιακές εισόδους – εξόδους (εκ των οποίων οι 14 μπορούν να χρησιμοποιηθούν ως PWM) και 16 αναλογικές εξόδους. Η σύνδεση γίνεται μέσω USB και η τροφοδοσία είτε μέσω AC – DC adapter, είτε μέσω μπαταριών.



Τεχνικά Χαρακτηριστικά:	
Μικροελεγκτής:	ATmega2560
Τάση λειτουργίας:	5V
Τάση εισόδου(συστήνεται):	7-12V
Τάση εισόδου (όρια):	6-20V
Ψηφιακά I/O Pins:	54 (εκ των οποίων οι 14 μπορούν να παρέχουν PWM ως έξοδο)
Αναλογικά Pins εισόδου:	16
DC ρεύμα ανά I/O Pin:	40 mA
DC ρεύμα για το 3.3V Pin:	50 mA
Flash Memory:	256 KB of which 8 KB used by bootloader
SRAM:	8 KB
EEPROM:	4 KB
Συχνότητα ρολογιού:	16 MHz

2.2.2 Οι σερβοκινητήρες

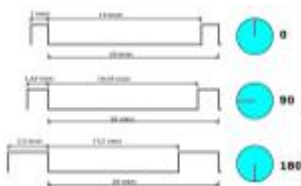
Οι σερβοκινητήρες είναι μικρές συσκευές που έχουν έναν άξονα ο οποίος μπορεί να περιστρέφεται σε συγκεκριμένη γωνία (0 έως 180 μοίρες) που είναι ελεγχόμενη από τον χρήστη. Ο έλεγχος ενός σερβοκινητήρα γίνεται με την βοήθεια παλμών που επαναλαμβάνονται κάθε 20 msec (ή $1/0,02 = 50$ Hz). Το μήκος του παλμού ρυθμίζει και την γωνία του άξονα. Τέτοιες συσκευές υπάρχουν σε τηλεκατευθυνόμενα αεροπλανάκια, αυτοκίνητα και φυσικά σε robots.



Οι σερβοκινητήρες έχουν έναν κινητήρα, ένα ποτενσιόμετρο και ένα κύκλωμα ελέγχου που μετρά συνεχώς την τιμή του ποτενσιόμετρου. Το ποτενσιόμετρο είναι συνδεδεμένο στον άξονα. Καθώς ο άξονας περιστρέφεται αλλάζει και η τιμή του ποτενσιόμετρου. Έτσι το κύκλωμα ελέγχου γνωρίζει σε ποια θέση βρίσκεται ο άξονας. Βλέποντας το σήμα ελέγχου το κύκλωμα μετακινεί τον άξονα στην κατάλληλη θέση.



Οι σερβοκινητήρες έχουν τρία καλώδια. Δύο τροφοδοσίας και ένα του σήματος ελέγχου. Η μορφή του σήματος ελέγχου φαίνεται παρακάτω. Οι τιμές των παλμών δεν είναι πάντα ίδιες για όλους του σερβοκινητήρες (κυμαίνονται όμως στα παρακάτω πλαίσια).



Στην κατασκευή μας χρησιμοποιούμε τους εξής σερβοκινητήρες:

- **Σερβοκινητήρας κίνησης – για τους πίσω τροχούς:**
Servo SpringRC SM-S4303R 360°



Τεχνικά χαρακτηριστικά:	
Τάση λειτουργίας:	4.8 – 6.0VDC
Μέγιστη ταχύτητα λειτουργίας:	60 – 70 RPM (4.8-6.0VDC αντίστοιχα)
Ροπή:	3.3 – 4.8 kg/cm (4.8-6.0VDC αντίστοιχα)
Κατασκευή:	4 πλαστικά γρανάζια + 1 μεταλλικό
	Διπλό ρουλεμάν
Διαστάσεις:	42 x 39.5 x 22.5mm
Μήκος καλωδίου:	30cm
Βάρος:	44g
Datasheet:	http://www.sparkfun.com/datasheets/Robotics/servo-360_e.pdf

- **Σερβοκινητήρας στροφής – για τους μπροστινούς τροχούς:**
Servo SpringRC SM-S3317B

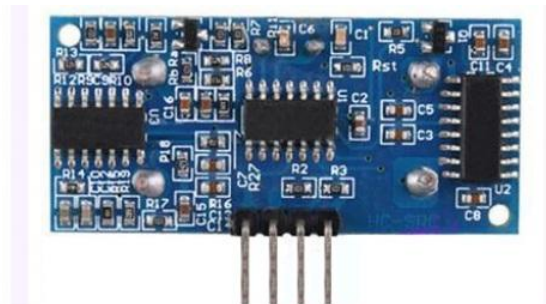


Τεχνικά χαρακτηριστικά:	
Τάση λειτουργίας:	4.8 – 6.0VDC
Μέγιστη ταχύτητα λειτουργίας:	0.14sec/60° στα 4.8VDC έως 0.11sec/60° στα 6VDC
Ροπή:	2.1 kg/cm στα 4.8VDC έως 2.5kg/cm στα 6VDC
Κατασκευή:	4 πλαστικά γρανάζια + 1 μεταλλικό
	Ένα ρουλεμάν
Γωνία περιστροφής:	120° (±60°)
Διαστάσεις:	31.3x16.5x28.6mm
Μήκος καλωδίου:	20cm
Βάρος:	20g

2.2.3 Αισθητήρες απόστασης

Απαραίτητη προϋπόθεση για να επιτευχθεί η αυτοματοποίηση του συστήματός μας, δηλαδή να κινείται χωρίς τον δικό μας έλεγχο, είναι οι αισθητήρες απόστασης. Καταλήγουμε στην επιλογή των υπερηχητικών αισθητήρων HC-SR04 λόγω του χαμηλού κόστους, καλής αξιοπιστίας και μεγάλης εμβέλειας.

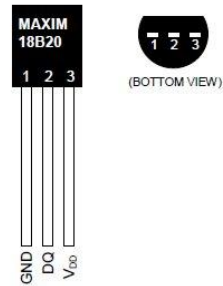
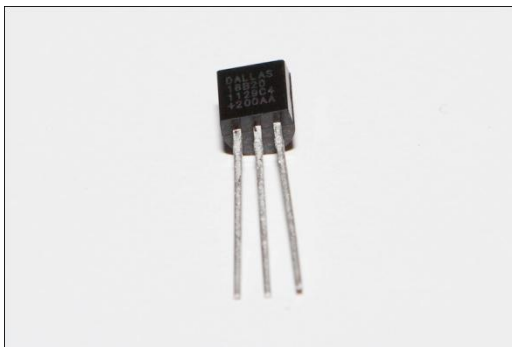
Ο αισθητήρας HC-SR04 διαθέτει 4 pins (Vcc, Trigger, Echo, GND) εκ των οποίων 2, τα Trigger και Echo, συνδέονται στα digital pins του Arduino. Στο Vcc pin συνδέουμε +5V από το Arduino. Η αρχή λειτουργίας του είναι απλή, το Trigger pin στέλνει ένα παλμό (HIGH) μικρής διάρκειας (~10μS) και το σήμα ανάκλασης από ένα αντικείμενο που επιστρέφει πίσω, εγείρει το Echo pin (HIGH). Η χρονική διάρκεια από τη μετάδοση μέχρι και τη λήψη του παλμού είναι γνωστή, οπότε η απόσταση υπολογίζεται εύκολα, αφού η ταχύτητα μετάδοσης είναι κι αυτή γνωστή και ισούται με 340m/s (ταχύτητα ήχου).



Τεχνικά χαρακτηριστικά:	
Τάση λειτουργίας:	5 VDC
Ρεύμα λειτουργίας:	15mA
Συχνότητα λειτουργίας:	40Hz
Ελάχιστη εμβέλεια:	2cm
Μέγιστη εμβέλεια:	400cm
Γωνία μέτρησης:	15°
Ακρίβεια:	0.3cm
Διαστάσεις:	45x20x15mm
Datasheet:	http://www.micropik.com/PDF/HCSR04.pdf http://jaktek.com/wp-content/uploads/2011/12/HC-SR04.pdf

2.2.4 Αισθητήρας θερμοκρασίας

Εφόσον η μέτρηση της θερμοκρασίας είναι σημαντική για τον σωστό έλεγχο και απεικόνιση του συστήματός μας, θα επιλέξουμε έναν αξιόπιστο και ακριβή αισθητήρα, τον DS18B20.



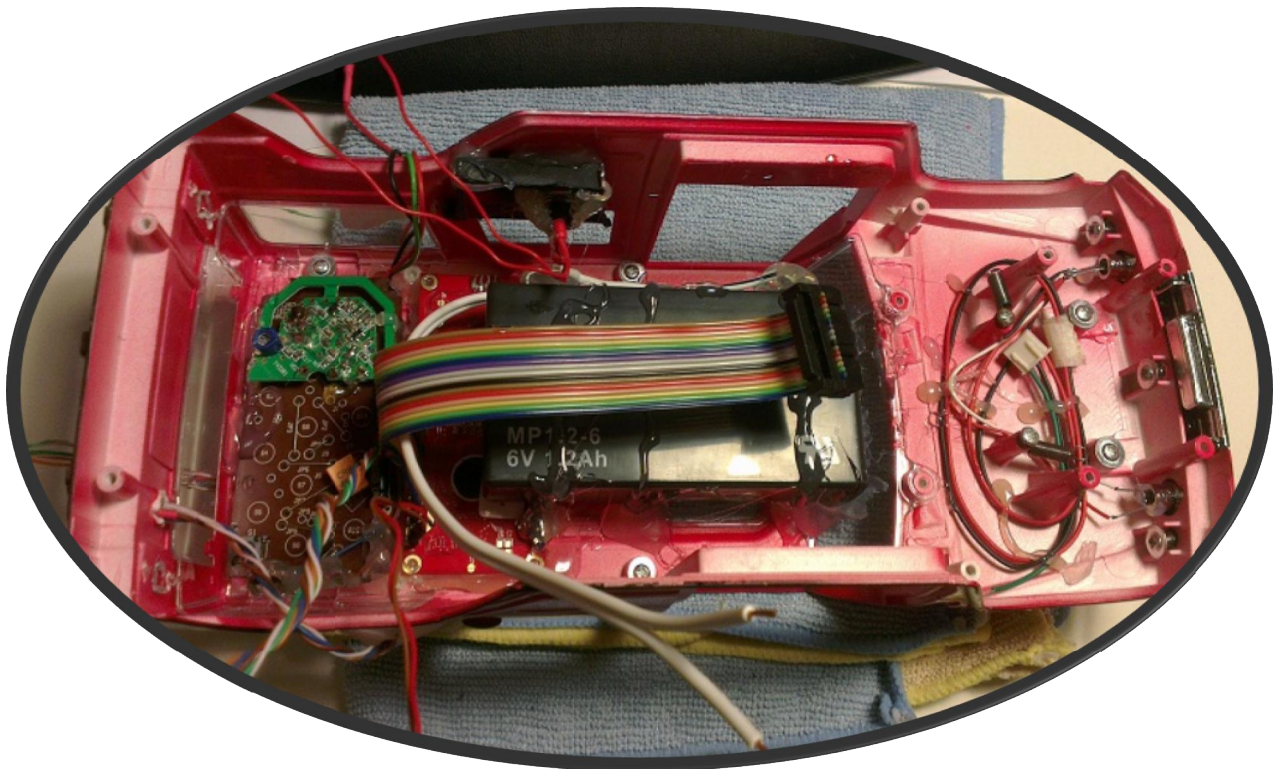
Τεχνικά χαρακτηριστικά:	
Τάση λειτουργίας:	3 – 5.5 VDC
Ενεργό Ρεύμα:	1mA
Ακρίβεια:	±0.5°C για μετρήσεις από -10°C έως +85°C
	±2°C για μετρήσεις από -55°C έως +125°C
Datasheet:	http://grobotronics.com/files/datasheets/ds18b20.pdf

2.2.5 Ασύρματη πρίζα

Για να ελέγχεται ασύρματα η συσκευή ψύξης, χρησιμοποιούμε μία ασύρματη πρίζα με τηλεχειριστήριο.

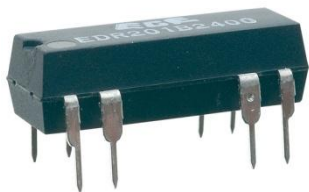


Το τηλεχειριστήριο θα το ανοίξουμε και θα κολλήσουμε καλώδια στο ON και OFF προκειμένου να το τοποθετήσουμε μέσα στο αυτοκινητάκι μας.



2.2.6 Reed Relay

Στη συνέχεια ενώνουμε τα καλώδια του τηλεχειριστηρίου με Reed Relay λόγω μικρής κατανάλωσης, για να επιτύχουμε τον έλεγχο μέσω του Arduino.



Τεχνικά χαρακτηριστικά Reed Relay:	
Τάση λειτουργίας:	5 VDC
Ρεύμα Λειτουργίας:	10mA
Μέγιστο Ρεύμα:	1A

2.2.7 Οθόνη LCD 2x16 5VDC

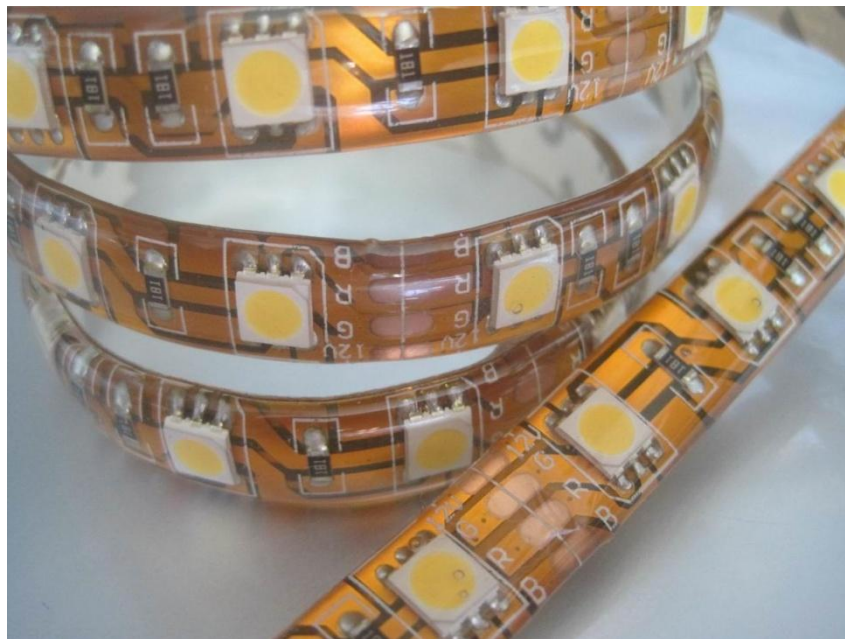
Η οθόνη LCD είναι απαραίτητη για την επικοινωνία του αυτοκινήτου με τον χρήστη. Αρχικά μας δείχνει τη θερμοκρασία, μετά δείχνει πότε γίνεται ο έλεγχος της πρίζας για να επιτευχθεί η σωστή κωδικοποίηση, στη συνέχεια μας δείχνει τις αποστάσεις από κάθε αισθητήρα HC-SR04 και τέλος γράφει «Πολύ ζέστη», στην περίπτωση που η θερμοκρασία ξεπεράσει τους 25°C.



2.2.8 RGB ταινία LED

Στο πίσω μέρος του αυτοκινήτου βλέπουμε 5εκ. LEDταινίας που αλλάζουν χρώματα ανάλογα τη θερμοκρασία. Συγκεκριμένα:

Χρώματα ανάλογα τη θερμοκρασία:	
<5°C	Λευκό
5-10°C	Μπλε
10-15°C	Γαλάζιο
15-20°C	Μωβ
20-25°C	Πράσινο
25-30°C	Πορτοκαλί
>30	Κόκκινο



2.2.9 LDR (Light Dependent Resistor)

Η LDR έχει την ιδιότητα ότι αλλάζει την αντίστασή της ανάλογα με το φως. Την έχουμε τοποθετήσει στο καπό του αυτοκινήτου για να μετράει τη φωτεινότητα του δωματίου.



2.2.10 Transistors

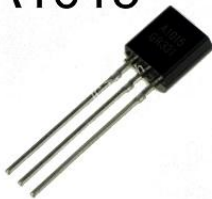
Για την ενεργοποίηση του κόκκινου Backlight της LCD οθόνης, της RGB ταινίας LED και των μπροστινών φώτων, χρησιμοποιούμε transistors ως διακόπτες.

C945



Τεχνικά χαρακτηριστικά C945:	
Τύπος transistor:	NPN
Μέγιστη Τάση:	50 VDC
Μέγιστο Ρεύμα:	100mA
Datasheet:	http://www.datasheetcatalog.org/datasheet/philips/2PC945.pdf

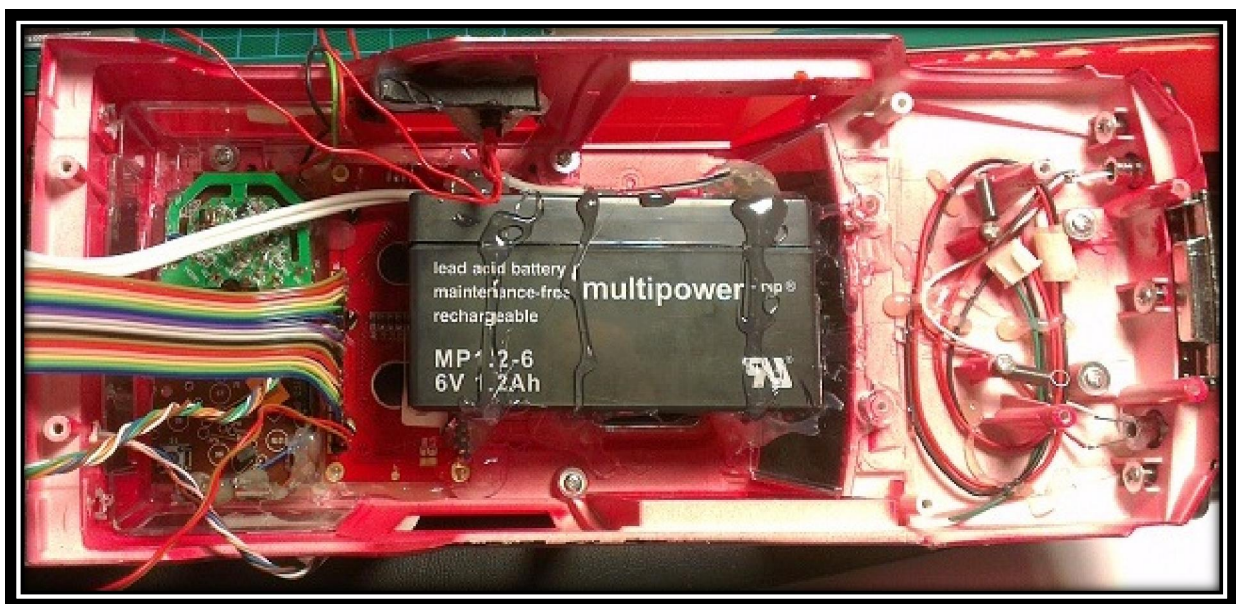
A1015



Τεχνικά χαρακτηριστικά A1015:	
Τύπος transistor:	PNP
Μέγιστη Τάση:	50 VDC
Μέγιστο Ρεύμα:	150mA
Datasheet:	http://www.datasheetcatalog.org/datasheets/50/157039_DS.pdf

2.2.11 Τροφοδοσία σερβοκινητήρων

Οι σερβοκινητήρες που χρησιμοποιούμε δεν έχουν τεράστια κατανάλωση ρεύματος, όμως κατά την εκκίνησή τους ως γνωστόν, μπορεί να τραβήξουν έως και 3 φορές περισσότερο από το ρεύμα λειτουργίας τους. Επιβάλλεται λοιπόν η πηγή ενέργειας να μπορεί να αντέξει τέτοια μεγάλη κατανάλωση. Επιλέγουμε λοιπόν μία μπαταρία μολύβδου. Δυστυχώς λόγω περιορισμών χώρου και βάρους, δε μπορούμε να βάλουμε μία μεγάλη (π.χ. 3.8Ah). Θα περιοριστούμε στην 6V 1.2Ah, την οποία θα κολλήσουμε στον ουρανό του αυτοκινήτου.



2.2.12 Γενική τροφοδοσία

Η ταινία LED RGB για να λειτουργήσει χρειάζεται τάση 12V. Επομένως δε μπορούμε να επιλέξουμε χαμηλότερη τάση. Επιλέγουμε λοιπόν να βάλουμε 10 επαναφορτιζόμενες μπαταρίες AA Ni-MH 1.2V. Σχεδόν όλες όμως οι Ni-MH έχουν το μειονέκτημα ότι μετά από λίγες εβδομάδες χάνουν μεγάλο μέρος της ενέργειάς τους. Αν δηλαδή αφήσουμε αχρησιμοποίητο το αυτοκίνητο για ένα ή δύο μήνες, μετά δε θα λειτουργεί γιατί οι μπαταρίες θα έχουν αποφορτιστεί!

Υπάρχει όμως μία μικρή ποικιλία Ni-MH που χρησιμοποιεί νέα τεχνολογία με αποτέλεσμα να διατηρεί το 90% της αρχικής ενέργειας για 6 μήνες και το 85% για ένα χρόνο. Μία τέτοια μπαταρία είναι η Panasonic Evolta. Βάζουμε λοιπόν 10 Panasonic Evolta AA Ni-MH 1.2V 2050Ah.



Κεφάλαιο 3

Ο Προγραμματισμός

Τα εξαρτήματα του προηγούμενου κεφαλαίου δε μπορούν να κάνουν τίποτα χωρίς τον ανάλογο προγραμματισμό. Έτσι λοιπόν σε αυτό το κεφάλαιο θα εξετάσουμε τον «κώδικα» που απαιτείται για να δουλέψει το σύστημά μας.

Προτού ξεκινήσουμε, να αναφέρουμε η γλώσσα προγραμματισμού δεν είναι Assembly (ευτυχώς), αλλά μία άλλη που μοιάζει λίγο με C++ και Java. Μπορείτε να την κατεβάσετε από εδώ:

<http://arduino.cc/en/Main/Software>

Για να γίνει κατανοητή η εξήγηση, θα είναι στο αριστερό μέρος με έντονα γράμματα ο κώδικας και στο δεξί ο σχολιασμός.

```
#include <NewPing.h>
#include <Servo.h>
#include <LiquidCrystal.h>
#include <OneWire.h>
```

Στην αρχή του προγράμματος εισάγουμε τις βιβλιοθήκες (libraries) που θέλουμε να περιληφθούν.

Στην επιστήμη υπολογιστών καλούμε **βιβλιοθήκη** (library) μια συλλογή από έτοιμα υποπρογράμματα που χρησιμοποιείται για την ανάπτυξη λογισμικού. Οι βιβλιοθήκες περιέχουν υποβοηθητικό κώδικα και δεδομένα, παρέχοντας, με αυτόν τον τρόπο, υπηρεσίες σε προγράμματα. Αυτό επιτρέπει τον διαμοιρασμό και τη χρήση του κώδικα και των δεδομένων με αρθρωτό τρόπο. Η έννοια της βιβλιοθήκης είναι αναπόσπαστο τμήμα του δομημένου προγραμματισμού και αναπτύχθηκε παράλληλα με αυτόν.

Η βιβλιοθήκη <NewPing.h> περιέχει το κομμάτι του κώδικα που χρειαζόμαστε για την επικοινωνία με τους υπερηχητικούς αισθητήρες HC-SR04.

Η βιβλιοθήκη <Servo.h> περιέχει το κομμάτι του κώδικα που χρειαζόμαστε για την επικοινωνία με τους σερβοκινητήρες.

Η βιβλιοθήκη <LiquidCrystal.h> περιέχει το κομμάτι του κώδικα που χρειαζόμαστε για την επικοινωνία με την LCD οθόνη.

Η βιβλιοθήκη <OneWire.h> περιέχει το κομμάτι του κώδικα που χρειαζόμαστε για την επικοινωνία με τον αισθητήρα θερμοκρασίας DS18B20.

Οι παραπάνω βιβλιοθήκες καθώς και πολλές άλλες βρίσκονται στην ιστοσελίδα:

<http://arduino.cc/en/Reference/Libraries>

```
#define Trigger_mprosta_aristera 27
#define Echo_mprosta_aristera 26
#define Trigger_mprosta_dexia 29
#define Echo_mprosta_dexia 28
#define Trigger_piso 31
#define Echo_piso 30
#define MAX_DISTANCE 150
```

Εδώ δηλώνουμε ποια pins (digital inputs/outputs) αντιστοιχούν σε ποιο αισθητήρα απόστασης. Θυμόμαστε από την επεξήγηση των HC-SR04 ότι πρέπει να συνδέσουμε τα pins "Trigger" και "Echo" στον μικροελεγκτή. Αυτό ακριβώς κάνουμε. Π.χ. για τον πίσω αισθητήρα έχουμε συνδέσει τα pins 30 και 31.

Επιπλέον ορίζουμε ποια είναι η μέγιστη απόσταση που θέλουμε να μετρείται – 150cm.

```
NewPing sonar_mprosta_aristera(Trigger_mprosta_aristera,
Echo_mprosta_aristera, MAX_DISTANCE);
NewPing sonar_mprosta_dexia(Trigger_mprosta_dexia,
Echo_mprosta_dexia, MAX_DISTANCE);
NewPing sonar_piso(Trigger_piso, Echo_piso, MAX_DISTANCE);
```

Δείχνουμε πόσους υπερηχητικούς αισθητήρες έχουμε και τους ονομάζουμε. Π.χ. «Μπροστά Δεξιά».

```
int apostasi_mprosta_aristera=100;
int apostasi_mprosta_dexia=100;
int apostasi_piso=100;
```

Δηλώνουμε τις μεταβλητές που θα χρησιμοποιήσουμε για τη μέτρηση των αποστάσεων ως ακέραιους αριθμούς (int) που παίρνουν τιμές από -32.768 έως 32.767 και βάζουμε αρχική τιμή το 100.

```
int pin13=13;
```

Ο μόνος λόγος αυτής της δήλωσης είναι ότι το pin13 είναι κατασκευαστικά συνδεδεμένο με ένα led το οποίο εγώ προσωπικά δε θέλω να ανάβει λόγω άσκοπης κατανάλωσης ενέργειας. Ελάχιστη μεν, μη μηδενική δε. Έτσι λοιπόν πιο κάτω θα βρούμε μία εντολή που θα το σβήνει.

```
int red=37;  
int green=44;  
int blue=45;
```

Αντιστοίχιση pins
σε χρώματα της
οθόνης LCD.

```
float therm;
```

Μεταβλητή αποθήκευσης της
θερμοκρασίας. Επιλέγουμε τον
τύπο "float" επειδή η
θερμοκρασία μετριέται με
δεκαδικά ψηφία.

```
int red_led=8;  
int blue_led=7;  
int green_led=9;
```

Αντιστοίχιση pins σε
χρώματα της RGB
ταινίας LED.

```
int trofodosia_6V=6;  
int Sustima_thermokrasias_ON=50;  
int Sustima_thermokrasias_OFF=51;
```

Για την τροφοδοσία από την 6V
μπαταρία μολύβδου έχουμε βάλει ένα
μικρό ρελέ με 5V πηνίο. Αυτό θα
ενεργοποιηθεί όταν εμείς δώσουμε 5V
στο pin 6.

Επιπλέον εισάγουμε ότι το ασύρματο
σύστημα ελέγχου θερμοκρασίας (το
τηλεχειριστήριο που
αποσυναρμολογήσαμε, θα ελέγχεται
από τα pins 50 και 51.


```
int voithitikos_metritis  
_gia_sustima_thermokrasias=0;
```

Αυτός ο βοηθητικός μετρητής, θα μας βοηθήσει στο πρόγραμμα για να ελέγχουμε αν έχει ενεργοποιηθεί το σύστημα ελέγχου θερμοκρασίας. Για παράδειγμα όταν έχουμε 26°C, ανάβει αυτόματα η συσκευή ψύξης. Μόλις πέσει κάτω από 25°C, αν ο βοηθητικός μετρητής είναι 1(που σημαίνει ότι έχει ενεργοποιηθεί το σύστημα ψύξης και λόγω αυτού δρόσισε), τότε θα σβήσει αυτόματα.

```
int fos=0;  
int Fota=35;
```

Στη μεταβλητή fos θα αποθηκεύεται η φωτεινότητα του δωματίου(από την LDR).

Τα μπροστινά φώτα συνδέονται στο pin 35.

```
Servo servokinisi;  
Servo servostrofi;
```

Εδώ δηλώνουμε ότι έχουμε 2 σερβοκινητήρες και τους ονομάζουμε.

```
OneWire ds(32);
```

Αντιστοιχίζουμε το pin32 με τον αισθητήρα θερμοκρασίας.

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

Για τον έλεγχο της LCD οθόνης δηλώνουμε:

- Τέσσερα pins για τη μεταφορά δεδομένων
→5, 4, 3, 2
- Ένα pin ενεργοποίησης (enable) → 11
- Ένα pin για τον έλεγχο της μνήμης των δεδομένων (Register Select) → 12

```
void setup()  
{
```

Το "void setup" είναι το κομμάτι στο οποίο δηλώνουμε ποια pins είναι είσοδοι, ποια έξοδοι και γενικότερα ό,τι θέλουμε να εκτελεστεί κατά την έναρξη της λειτουργίας για μία μόνο φορά.

```
  lcd.begin(16, 2);
```

Δείχνουμε ότι η οθόνη που έχουμε συνδέσει έχει 16 στήλες και 2 γραμμές.

```
  Serial.begin(2400);
```

Εδώ δηλώνουμε ότι θέλουμε να έχουμε σειριακή επικοινωνία του μικροελεγκτή με τον ηλεκτρονικό υπολογιστή έτσι ώστε να μας δείχνει στην οθόνη τα δεδομένα που θα επιλέξουμε παρακάτω. Απαραίτητη προϋπόθεση για να δούμε τα δεδομένα, είναι να ανοίξουμε τη "Σειριακή Οθόνη" από το μενού του λογισμικού του Arduino. Επιπλέον ρυθμίζουμε την ταχύτητα μετάδοσης των δεδομένων στα 2400bits/second.

```
  servokinisi.attach(22);
```

```
  servostrofi.attach(23);
```

Ο σερβοκινητήρας στις πίσω ρόδες (κίνηση) ενώνεται στο pin22 και ο σερβοκινητήρας στις μπροστινές ρόδες (στροφή) ενώνεται στο pin23.

```
  pinMode(pin13, OUTPUT);
```

```
  digitalWrite(pin13, LOW);
```

Εδώ είναι το σημείο που δηλώνουμε ότι το pin13 είναι έξοδος και επειδή όπως είπαμε παραπάνω είναι συνδεδεμένο με ένα led, το σβήνουμε.

```
pinMode(trofodosia_6V, OUTPUT);  
digitalWrite(trofodosia_6V,LOW);
```

trofodosia_6V = έξοδος
Για σιγουριά το κάνουμε LOW, δηλαδή
αν για κάποιο λόγο (δυσλειτουργία ή
κάτι άλλο) είναι ενεργοποιημένη η 6V
τροφοδοσία, τώρα απενεργοποιείται.

```
pinMode(red_led, OUTPUT);  
pinMode(blue_led, OUTPUT);  
pinMode(green_led, OUTPUT);  
pinMode(red, OUTPUT);  
pinMode(green, OUTPUT);  
pinMode(blue, OUTPUT);  
pinMode(Sustima_thermokrasias_ON, OUTPUT);  
pinMode(Sustima_thermokrasias_OFF, OUTPUT);
```

Τα pins που αντιστοιχούν
στα χρώματα κόκκινο,
μπλε, πράσινο της LCD
οθόνης και της ταινίας
RGB, καθώς και αυτά που
ελέγχουν το σύστημα
ελέγχου θερμοκρασίας

```
digitalWrite(red,HIGH);
```

Το transistor A1015 που χρησιμοποιούμε για το
άναμμα/σβήσιμο του κόκκινου backlight της LCD
οθόνης, είναι PNP. Άρα όταν δίνουμε 5V (HIGH) το
φως σβήνει. Εδώ λοιπόν απλώς επιβεβαιώνουμε
ότι είναι σβηστό, προτού το ανάψουμε.

```
lcd.clear();
```

Καθαρίζουμε την LCD οθόνη (πρέπει να είναι κενή
προτού αρχίσουμε να γράφουμε κάτι, γιατί αλλιώς
θα μπερδευτούν τα παλιά γραφόμενα με τα νέα.

```
digitalWrite(red,LOW);  
digitalWrite(blue,HIGH);  
digitalWrite(green,HIGH);
```

Ενεργοποιούμε όλα τα
χρώματα με αποτέλεσμα να
φαίνεται άσπρη ο οθόνη.

```
therm=thermokrasia());
```

Καλείται η συνάρτηση `thermokrasia()` η οποία όπως θα δούμε παρακάτω μετράει τη θερμοκρασία και την αποδίδει στη μεταβλητή `therm`.

```
for (int o=1; o<11; o++)
```

```
{
```

```
lcd.print(" ");
```

```
lcd.print(therm);
```

```
lcd.print("*C");
```

```
therm=thermokrasia());
```

```
xrwma_analoga_ti_thermokrasia());
```

```
delay(1000);
```

```
lcd.clear();
```

```
}
```

```
digitalWrite(red_led, LOW);
```

```
digitalWrite(blue_led, LOW);
```

```
digitalWrite(green_led, LOW);
```

Γίνονται 10 μετρήσεις της θερμοκρασίας και κάθε φορά φαίνεται στην οθόνη.

Επίσης η συνάρτηση `xrwma_analoga_ti_thermokrasia()` αποδίδει διαφορετικό χρώμα κάθε φορά στην RGB ταινία LED αναλόγως τη θερμοκρασία.

Σβήνει η RGB ταινία LED.

```
lcd.print("Elegchos prizas..");
```

```
delay(4000);
```

```
digitalWrite(Sustima_thermokrasias_OFF,HIGH);
```

```
delay(2000);
```

```
digitalWrite(Sustima_thermokrasias_OFF,LOW);
```

```
delay(2000);
```

Κάθε φορά που η πρίζα βγαίνει από το ρεύμα και μετά ξαναμπάινει, πρέπει να πατάμε από το τηλεχειριστήριο ένα πλήκτρο ON ή OFF και συγχρόνως το κουμπί "Learn" από την πρίζα για να συγχρονιστεί. Ουσιαστικά δηλαδή αυτό κάνουμε εδώ. Μόλις λουπόν δείξει η οθόνη "Elegchos prizas..", περιμένουμε 4" και πατάμε το "Learn".

```
lcd.clear();  
lcd.print("Parakalw balte ti sosti Apostasi!");  
delay(2000);  
for (int CursorPosition = 0; CursorPosition < 17; CursorPosition++)  
{  
  lcd.scrollDisplayLeft();  
  delay(400);  
}
```

Θέλουμε η οθόνη να δείξει το μήνυμα «Παρακαλώ βάλτε τη σωστή Απόσταση». Επειδή όμως οι χαρακτήρες είναι περισσότεροι από 16 που έχει η οθόνη, δημιουργούμε ένα βρόχο επανάληψης για να μετακινούνται τα γράμματα με ταχύτητα 400milliseconds και να δούμε όλο το μήνυμα.

```
delay(5000);  
lcd.clear();  
lcd.print("M.A M.D PISO");
```

Περιμένουμε 5" και μετά αφού καθαρίζουμε την οθόνη, γράφουμε το μήνυμα: "M.A M.D PISO". Αυτό το κάνουμε διότι μετά θα αρχίσουμε να μετράμε τις αποστάσεις και πρέπει να αναγνωρίζουμε τι δείχνει ο κάθε αισθητήρας.
M.A = Μπροστά Αριστερά
M.D = Μπροστά Δεξιά
PISO = Πίσω

```

while (apostasi_mprosta_aristera!=34)
{
  delay(100);

  apostasi_mprosta_aristera = sonar
  _mprosta_aristera.ping_cm();

  delay(100);

  apostasi_mprosta_dexia = sonar
  _mprosta_dexia.ping_cm();

  Serial.print("Apostasi: ");
  Serial.print(apostasi_mprosta_aristera);
  Serial.println(" cm");
  Serial.print("\t");
  Serial.print("\t");
  Serial.println(apostasi_mprosta_dexia);
  servokinisi.write(90);
  servostrofi.write(90);
  nikelcdprint();
  therm=thermokrasia_xoris_kathisterisi();
  xrwma_analoga_ti_thermokrasia();
}

```

Εδώ βάζουμε άλλο έναν βρόχο επανάληψης ο οποίος ουσιαστικά κρατά ακίνητο το αυτοκίνητο μέχρι να εισαχθεί η σωστή απόσταση η οποία είναι 34 εκατοστά.

Μέχρι να γίνει αυτό γίνονται συνεχώς μετρήσεις οι οποίες εκτυπώνονται τόσο στην LCD οθόνη, όσο και στη σειριακή οθόνη του ηλεκτρονικού υπολογιστή.

Επεξήγηση εντολών:

`servokinisi.write(90)` = σταματά τον σερβοκινητήρα που κινεί το αυτοκίνητο. Κάτω από 90 είναι όπισθεν και πάνω από 90 είναι μπροστινή κίνηση.

`servostrofi.write(90)` = ρυθμίζει τον σερβοκινητήρα που στρίβει το αυτοκίνητο, στις 90° που κατασκευαστικά είναι ευθεία.

`niklcdprint()` = συνάρτηση που θα την δούμε αναλυτικά πιο κάτω και δείχνει στην LCD οθόνη, τη σωστή απόσταση. Δηλαδή την απόσταση του μπροστά αριστερά αισθητήρα, την τοποθετεί κάτω από το M.A κλπ.

`thermokrasia_xoris_kathisterisi()` = συνάρτηση που μετράει τη θερμοκρασία χωρίς ενδιάμεση καθυστέρηση.

```

Serial.println(" Xekiname ");
lcd.clear();
lcd.print(" Ready ");
digitalWrite(green,LOW);
digitalWrite(blue,LOW);
delay(2500);
lcd.clear();
lcd.print(" Set ");
digitalWrite(green,HIGH);
delay(2500);
lcd.clear();
lcd.print(" GO ");
digitalWrite(red,HIGH);
digitalWrite(blue,LOW);
digitalWrite(trofodosia_6V,HIGH);
delay(2500);

lcd.clear();
lcd.print("M.A M.D Piso");
}

```

Όταν εισαχθεί η σωστή απόσταση, τότε η εκτέλεση του προγράμματος θα συνεχιστεί.

Αμέσως μετά λοιπόν η οθόνη θα γίνει κόκκινη και θα γράψει: **"Ready"**.
2,5'' μετά, η οθόνη θα γίνει κίτρινη και θα γράψει: **"Set"**.

Άλλα 2,5'' μετά, η οθόνη θα γίνει πράσινη και θα γράψει: **"GO"**.
Αυτό το εμπνεύστηκε από τους αγώνες ταχυτήτων!

Επιπρόσθετα, ενεργοποιείται το ρελέ της 6V τροφοδοσίας, καθαρίζει η οθόνη και γράφει "M.A M.D Piso" για να είναι έτοιμη η οθόνη μας για την εκκίνηση του αυτοκινήτου και τις επόμενες μετρήσεις.

```

void loop()
{
  Serial.println("kinisi Mprosta ");
  Serial.println(apostasi_mprosta_aristera);
  Serial.print("\t");
  Serial.print("\t");
  Serial.println(apostasi_mprosta_dexia);

```

Από αυτό το σημείο αρχίζει το κυρίως πρόγραμμά μας. Ξεκινά με την εκτύπωση των αποστάσεων στη σειριακή οθόνη του ηλεκτρονικού υπολογιστή.

```

delay(50);
apostasi_mprosta_aristera = sonar_mprosta_aristera.ping_cm();
delay(50);
apostasi_mprosta_dexia = sonar_mprosta_dexia.ping_cm();
servokinisi.write(150);
servostrofi.write(90);
therm=thermokrasia_xoris_kathisterisi();
xrwma_analoga_ti_thermokrasia();
fos=analogRead(1);
Serial.println("\t");
Serial.println("fos ");
Serial.println(fos);

```

Το σημαντικότερο κομμάτι του προγράμματος!

Το αυτοκίνητο κινείται ευθεία μπροστά, μετράει συνεχώς τις αποστάσεις από τους δύο μπροστινούς αισθητήρες, αλλά και τη θερμοκρασία, σύμφωνα με την οποία αποδίδει στο κατάλληλο χρώμα στην RGB ταινία LED. Ταυτόχρονα διαβάζει και τυπώνει στη σειριακή οθόνη, τη φωτεινότητα του δωματίου.


```

if (fos<400)
{
  digitalWrite(Fota, LOW);
}
else
{
  digitalWrite(Fota, HIGH);
}

```

Αρχίζουν οι συνθήκες όπως λέμε στον προγραμματισμό.

Αν το φως του δωματίου είναι αρκετό, τότε τα φώτα να είναι σβηστά.
Αλλιώς τα φώτα ανάβουν.

```

if ((therm>25)&&(voithitikos_metritis_gia_sustima_thermokrasias==0))
{
  servokinisi.write(90);
  servostrofi.write(90);
  lcd.clear();
  lcd.print("  Polu Zesti");
  delay(5000);
  digitalWrite(Sustima_thermokrasias_ON, HIGH);
  delay(2000);
  digitalWrite(Sustima_thermokrasias_ON, LOW);
  delay(1000);
  voithitikos_metritis_gia_sustima_thermokrasias=1;
  lcd.clear();
  lcd.print("M.A M.D  Piso");
}

```

Αν η θερμοκρασία ανέβει πάνω από 25°C και ο βοηθητικός μετρητής για το σύστημα θερμοκρασίας είναι 0, τότε σταματάει το αυτοκίνητο να κινείται, στην οθόνη εμφανίζεται το μήνυμα «Πολύ Ζέστη» και ενεργοποιείται η πρίζα στην οποία έχουμε συνδέσει το σύστημα ψύξης.

Αμέσως μετά κάνουμε τον βοηθητικό μετρητή "1", καθαρίζουμε την οθόνη και ξαναγράφουμε:
M.A M.D Piso
για να συνεχιστεί η λειτουργία του αυτοκινήτου.

```

if ((therm<=25)&&(voithitikos_metritis_gia_sustima_thermokrasias==1))
{
servokinisi.write(90);
servostrofi.write(90);
lcd.clear();
lcd.print(" Drosise Ligaki");
delay(5000);
digitalWrite(Sustima_thermokrasias_OFF, HIGH);
delay(2000);
digitalWrite(Sustima_thermokrasias_OFF, LOW);
delay(1000);
voithitikos_metritis_gia_sustima_thermokrasias=0;
lcd.clear();
lcd.print("M.A M.D Piso");
}

```

Αν η θερμοκρασία πέσει κάτω από 25°C και ο βοηθητικός μετρητής για το σύστημα θερμοκρασίας είναι 1 (που σημαίνει ότι είναι ενεργοποιημένο το σύστημα ψύξης), τότε σταματάει το αυτοκίνητο να κινείται, στην οθόνη εμφανίζεται το μήνυμα «Δρόσιση Λιγάκι» και απενεργοποιείται η πρίζα στην οποία έχουμε συνδέσει το σύστημα ψύξης.

Αμέσως μετά κάνουμε τον βοηθητικό μετρητή "0", καθαρίζουμε την οθόνη και ξαναγράφουμε:
M.A M.D Piso
για να συνεχιστεί η λειτουργία του αυτοκινήτου.

```
niklcdprint();
```

Εμφάνιση των αποστάσεων στην οθόνη.

```

if ((apostasi_mprosta_aristera<29)&&(apostasi_mprosta_aristera>3))
{
for (int x=0; x<7; x++)
{
Serial.println("Opisthen aristera ");
Serial.println(apostasi_mprosta_aristera);
Serial.print("\t");
Serial.print("\t");
Serial.println(apostasi_mprosta_dexia);
servokinisi.write(40);
servostrofi.write(115);
delay(100);
apostasi_mprosta_aristera = sonar
_mprosta_aristera.ping_cm();
delay(100);
apostasi_mprosta_dexia = sonar
_mprosta_dexia.ping_cm();
delay(100);
apostasi_piso = sonar_piso.ping_cm();
niklcdprint();
if (((apostasi_piso<21) && (apostasi_piso>2)))
{
break;
}
}
}
}

```

Αν η απόσταση από τον μπροστά αριστερό αισθητήρα γίνει λιγότερο από 25εκατοστά και περισσότερο από 3εκατοστά, τότε εμφανίζεται στη σειριακή οθόνη το μήνυμα «Όπισθεν αριστερά» μαζί με τις αποστάσεις. Το αυτοκίνητο ξεκινά να κινείται με κατεύθυνση πίσω αριστερά.

Αυτό είναι σε ένα βρόχο επανάληψης για 6 φορές. Έτσι θα κάνει όπισθεν για αρκετό χρόνο που ελπίζουμε ότι θα έχει αποφύγει ό,τι βρισκόταν μπροστά του. Αν όχι θα ξαναμπεί στο βρόχο.

Αν όμως, οποιαδήποτε στιγμή συναντήσει εμπόδιο και πίσω, βγαίνει από το βρόχο (και συνεχίζει ευθεία μπροστά).

Σημείωση: Θεωρητικά δε χρειαζόταν στη συνθήκη να είναι η απόσταση μεγαλύτερη από 3 εκατοστά, όμως πρακτικά ο αισθητήρας HC-SR04, όταν ξεπερνά τα 2μέτρα περίπου, αρχίζει να δείχνει ορισμένες φορές "1" ή "2" εκατοστά απόσταση και πάνω από 3-4μέτρα, δείχνει "0". Οπότε το πρόγραμμα θα νόμιζε ότι υπάρχει εμπόδιο!

```

else if ((apostasi_mprosta_dexia<29)&&(apostasi_mprosta_dexia>3))
{
for (int y=0; y<7; y++)
{
Serial.println("Opisthen dexia ");
Serial.println(apostasi_mprosta_aristera);
Serial.print("\t");
Serial.print("\t");
Serial.println(apostasi_mprosta_dexia);
servokinisi.write(40);
servostrofi.write(65);
delay(50);

apostasi_mprosta_aristera = sonar_mprosta_aristera.ping_cm();
delay(50);
apostasi_mprosta_dexia = sonar_mprosta_dexia.ping_cm();
delay(40);
apostasi_piso = sonar_piso.ping_cm();
niklcdprint();
if (((apostasi_piso<21) && (apostasi_piso>2)))
{
break;
}
}
}
}
}

```

Ακριβώς το ίδιο με πριν αλλά για τη δεξιά μεριά.

Η μόνη διαφορά εδώ είναι ότι έχουμε μειώσει τον χρόνο. Αυτό χρησιμεύει σε περίπτωση που βρεθεί σε κάποια περίεργη γωνία με ίσες αποστάσεις αριστερά δεξιά. Ενώ σε μία τέτοια περίπτωση δε θα μπορούσε να ξεκολλήσει, τώρα μπορεί γιατί θα πάει περισσότερο χρόνο πίσω αριστερά, απ' ότι πίσω δεξιά, με αποτέλεσμα να αλλάξουν οι γωνίες.

Εδώ τελειώνει το πρόγραμμά μας! Παρακάτω θα δούμε τις συναρτήσεις.

```

void nikelcdprint() {
  lcd.setCursor(0, 1);
  lcd.print(apostasi_mprosta_aristera);
  lcd.print(" ");
  lcd.setCursor(5, 1);
  lcd.print(apostasi_mprosta_dexia);
  lcd.print(" ");
  lcd.setCursor(11, 1);
  lcd.print(apostasi_piso);
  lcd.print(" ");
}

```

Η συνάρτηση nikelcdprint() τοποθετεί τον κέρσορα στην κατάλληλη θέση για να γραφτεί σωστά η απόσταση.
Π.χ. Η εντολή lcd.setCursor(5, 1) τοποθετεί τον κέρσορα ακριβώς κάτω από το M.D (που έχουμε αναφέρει ότι είναι το μπροστά δεξιά).

```

float thermokrasia() {
  byte o;
  byte present = 0;
  byte type_s;
  byte data[12];
  byte addr[8];
  float celsius;

  if ( !ds.search(addr)) {
    ds.reset_search();
    delay(50);
  }

  ds.reset();
  ds.select(addr);
  ds.write(0x44,1);
  delay(800);
  present = ds.reset();
  ds.select(addr);
  ds.write(0xBE);

  for ( o = 0; o < 9; o++) {
    data[o] = ds.read();
  }
  unsigned int raw = (data[1] << 8) | data[0];
  if (type_s) {
    raw = raw << 3;
    if (data[7] == 0x10) {
      raw = (raw & 0xFFF0) + 12 - data[6];
    }
  } else {

```

Το περιεχόμενο της συνάρτησης thermokrasia() βρίσκεται στα παραδείγματα του λογισμικού του Arduino για τον One Wire αισθητήρα θερμοκρασίας, που είναι αυτός που χρησιμοποιούμε. Ουσιαστικά μετράει ψηφιακά, δηλαδή σε bits τη θερμοκρασία.

<http://playground.arduino.cc/Learning/OneWire>

```

byte cfg = (data[4] & 0x60);
if (cfg == 0x00) raw = raw << 3;
else if (cfg == 0x20) raw = raw << 2;
else if (cfg == 0x40) raw = raw << 1;
}
celsius = (float)raw / 16.0;
Serial.print(" Temperature = ");
Serial.print(celsius);
Serial.print(" Celsius, ");
return celsius;
}

```

```

void xrwma_analoga_ti_thermokrasia()
{
  if (therm<=5)
  {
    digitalWrite(red_led, HIGH);
    digitalWrite(blue_led, HIGH);
    digitalWrite(green_led, HIGH);
  }
  else if (therm>5 && therm<=10)
  {
    digitalWrite(red_led, LOW);
    digitalWrite(blue_led, HIGH);
    digitalWrite(green_led, LOW);
  }
  else if (therm>10 && therm<=15)
  {
    digitalWrite(red_led, LOW);
    digitalWrite(blue_led, HIGH);
    digitalWrite(green_led, HIGH);
  }
  else if (therm>15 && therm<=20)
  {
    digitalWrite(red_led, HIGH);
    digitalWrite(blue_led, HIGH);
    digitalWrite(green_led, LOW);
  }
  else if (therm>20 && therm<=25)
  {
    digitalWrite(red_led, LOW);
    digitalWrite(blue_led, LOW);
    digitalWrite(green_led, HIGH);
  }
}

```

Η συνάρτηση `xrwma_analoga_ti_thermokrasia()` αλλάζει το χρώμα της RGB ταινίας LED σύμφωνα με τη θερμοκρασία.

Στη σελίδα 23 υπάρχει ο αναλυτικός πίνακας.

```

else if (therm>25 && therm<=30)
{
digitalWrite(red_led, HIGH);
digitalWrite(blue_led, LOW);
digitalWrite(green_led, HIGH);
}
else if (therm>30)
{
digitalWrite(red_led, HIGH);
digitalWrite(blue_led, LOW);
digitalWrite(green_led, LOW);
}
}

```

```

float thermokrasia_xoris_kathisterisi()
{
byte o;
byte present = 0;
byte type_s;
byte data[12];
byte addr[8];
float celsius;

if (!ds.search(addr)) {
ds.reset_search();
delay(50);
}

ds.reset();
ds.select(addr);
ds.write(0x44,1);

present = ds.reset();
ds.select(addr);
ds.write(0xBE);

for ( o = 0; o < 9; o++) {
data[o] = ds.read();
}

unsigned int raw = (data[1] << 8) | data[0];
if (type_s) {
raw = raw << 3;
if (data[7] == 0x10) {

```

Η συνάρτηση thermokrasia_xoris_kathisterisi() είναι ίδια με την συνάρτηση thermokrasia().

Η μόνη διαφορά είναι ότι δεν περιλαμβάνεται στο μέσο της η καθυστέρηση των 800ms που έχει η αρχική συνάρτηση και αυτό μας χρησιμεύει για τις περιπτώσεις που θέλουμε γρήγορη μέτρηση αλλά και να μην καθυστερήσει ο συνολικός βρόχος.

```
    raw = (raw & 0xFFF0) + 12 - data[6];  
  }  
} else {  
  byte cfg = (data[4] & 0x60);  
  if (cfg == 0x00) raw = raw << 3;  
  else if (cfg == 0x20) raw = raw << 2;  
  else if (cfg == 0x40) raw = raw << 1;  
}  
celsius = (float)raw / 16.0;  
Serial.print(" Temperature = ");  
Serial.print(celsius);  
Serial.print(" Celsius, ");  
return celsius;  
}
```

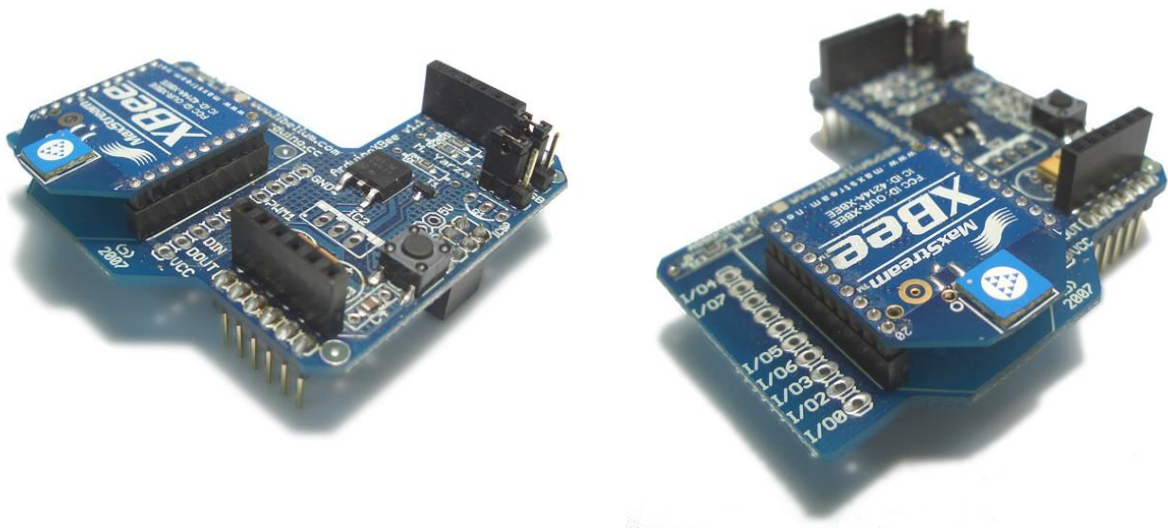

Κεφάλαιο 4

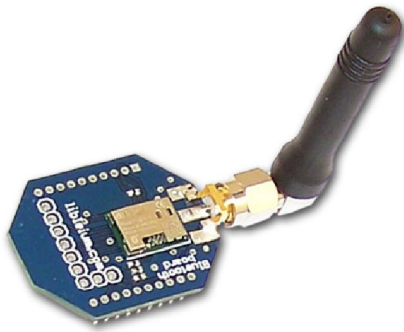
Μελλοντικές Βελτιώσεις

Η συνεχώς εξελισσόμενη τεχνολογία μας εξοπλίζει με καινούριες ιδέες και μας δίνει τη δυνατότητα να τις υλοποιήσουμε. Ασφαλώς αυτό απαιτεί τις ανάλογες θυσίες – προσωπική προσπάθεια, χρόνο και φυσικά υλικούς πόρους. Έτσι και στην παρούσα εργασία δεν πρέπει να παραμελήσουμε το γεγονός ότι θα μπορούσαμε να κάνουμε σημαντικές βελτιώσεις. Ας εξετάσουμε κάποιες από αυτές.

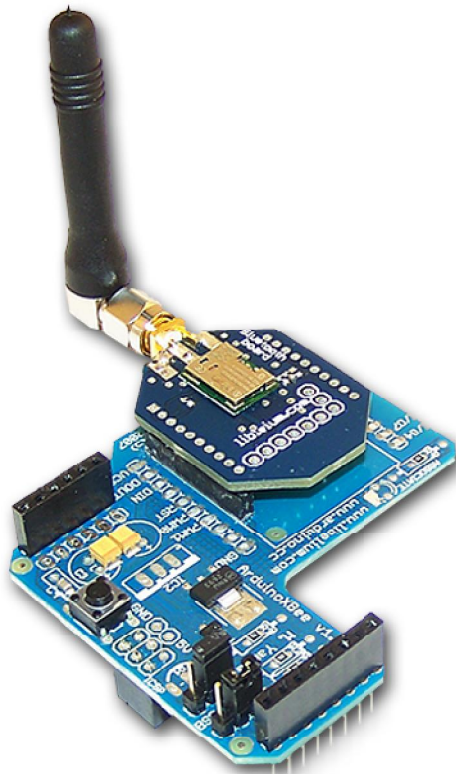
4.1 Επικοινωνία μέσω Bluetooth

Μία μικρή βελτίωση είναι η χρήση ενός Bluetooth module. Χρησιμοποιώντας λοιπόν το android τηλέφωνό μας θα μπορούσαμε να ενεργοποιήσουμε ή απενεργοποιήσουμε τη λειτουργία του αυτοκινήτου, να το κατευθύνουμε ή ακόμα και να επιλέξουμε να ενεργοποιήσει/απενεργοποιήσει το σύστημα ψύξης. Για να το κάνουμε αυτό θα χρειαστούμε το Arduino XBee shield και μία Bluetooth **κεραία**.





Μόλις τα συνδέσουμε μεταξύ τους θα έχουν αυτή τη μορφή:



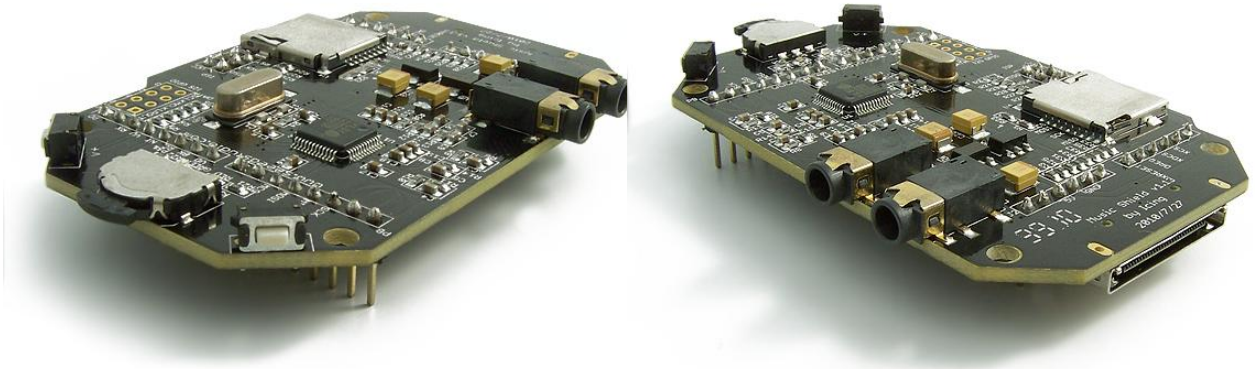
Ένα παράδειγμα τέτοιας σύνδεσης βρίσκουμε στο παρακάτω link:

<http://www.mobot.es/MobotBTCar.html>

4.2 Αναπαραγωγή ήχου

Βάζοντας το music shield πάνω στον Arduino και συνδέοντας κάποια μικρά φορητά ηχεία (portable), μπορούμε να ηχογραφήσουμε μηνύματα και να κάνουμε το αυτοκίνητό μας να “μιλάει”. Για παράδειγμα, να ακούγονται τα μηνύματα που εμφανίζονται στην LCD οθόνη ή/και να παίζει το αγαπημένο μας κομμάτι καθώς κινείται.

Άλλη μία εφαρμογή είναι να προσαρμόσουμε ένα μικρόφωνο και να όταν λείπουμε από το σπίτι ακουστεί κάποιος δυνατός θόρυβος, να ξεκινήσει κάποια ηχογραφημένη συνομιλία, ώστε να φαίνεται ότι υπάρχει κόσμος!



Παραδείγματα ηχείων:



4.3 Σύνδεση μέσω WiFi

Με το Arduino WiFi Shield μπορούμε να ελέγχουμε το αυτοκίνητό μας μέσω WiFi και γενικότερα να μεταδίδουμε δεδομένα.



Σχετικά παραδείγματα:

<http://blog.tkjelectronics.dk/2011/02/wifi-controlled-rc-car-with-the-arduino/>

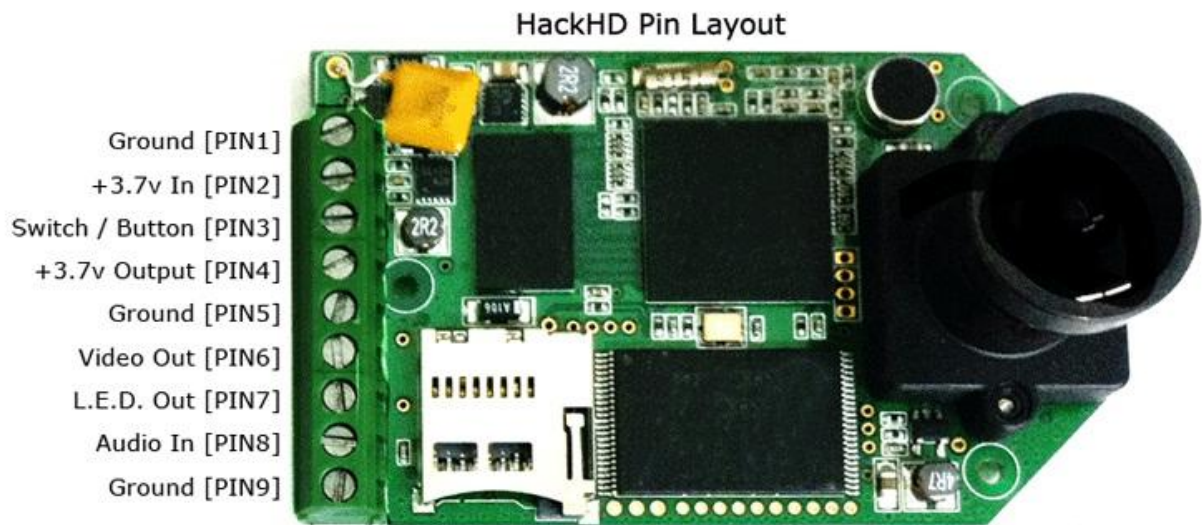
<http://www.youtube.com/watch?v=BSYBrEvtSis&list=PL1523240F4C28F4EA>

4.4 Παρακολούθηση Χώρου

Τοποθετώντας μία κάμερα, θα μπορούμε να βλέπουμε ό,τι βλέπει το αυτοκίνητό μας. Το βίντεο μπορεί είτε να αποθηκεύεται σε microSD card, είτε να το βλέπουμε μέσω internet, χρησιμοποιώντας σύνδεση WiFi.

Παράδειγμα κάμερας:

<http://hackhd.com/>



Σχετικά παραδείγματα:

<http://cpvrc.wordpress.com/2012/11/23/episode-2-hackhd-on-my-arduino-rc-racer/#>

Συνοψίζοντας μπορούμε να πούμε ότι οι προοπτικές βελτίωσης είναι σχεδόν αμέτρητες. Υπό αυτό το πρίσμα διακρίνουμε πόσα λίγα έχουμε κάνει σε αυτή την πτυχιακή εργασία. Αλλά δεν απογοητεύομαστε γιατί ήταν μία μεγάλη εμπειρία που μας έδωσε τις απαιτούμενες γνώσεις για την κατασκευή/εφαρμογή συστημάτων μηχανικής καθώς και τις βάσεις για περαιτέρω ανάπτυξη.

Βιβλιογραφία:

1. <http://el.wikipedia.org>
2. <http://arduino.cc>
3. <http://alexkaltsas.wordpress.com/2012/03/18/avr-gcc-%CF%83%CE%B5%CF%81%CE%B2%CE%BF%CE%BA%CE%B9%CE%BD%CE%B7%CF%84%CE%AE%CF%81%CE%B5%CF%82-%CE%BC%CE%BF%CE%BD%CF%84%CE%B5%CE%BB%CE%B9%CF%83%CE%BC%CE%BF%CF%8D-aka-servos/>
4. <http://www.arduino.gr/2013/01/water-tank-level-display.html>
5. <http://www.cooking-hacks.com>