



**ΑΕΙ ΠΕΙΡΑΙΑ ΤΕΧΝΟΛΟΓΙΚΟΥ ΤΟΜΕΑ**

***Σχολή Τεχνολογικών Εφαρμογών***

***Τμήμα Ηλεκτρολόγων Μηχανικών Τ.Ε.***

## **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Διερεύνηση Και Αξιοποίηση Των Δυνατοτήτων  
Της Ηλεκτρονικής Πλατφόρμας Arduino Για  
Κατασκευή Επίγειου Radar**



**Λάγαρης Δημήτριος  
Α.Μ.: 32097**

Υπεύθυνος Καθηγητής  
Ζαχμάνογλου Άρτεμις

Αιγάλεω 2017

## Πίνακας περιεχομένων

<b>1. Πρόλογος.....</b>	<b>3</b>
<b>2. Arduino.....</b>	<b>4</b>
2.1. Εισαγωγή .....	4
2.2. Μικροελεγκτές .....	4
2.3. Εφαρμογές Μικροελεγκτών .....	5
2.4. Ιστορία Arduino .....	6
2.5. Πλεονεκτήματα Arduino .....	7
2.6. Εκδόσεις Arduino .....	8
2.7. Χαρακτηριστικά Arduino UNO .....	9
2.8. Arduino UNO .....	11
2.9. Processing.....	18
<b>3. Radar - Σερβοκινητήρας – Αισθητήρας υπερήχων .....</b>	<b>21</b>
3.1. Radar.....	21
3.2. Σερβοκινητήρας.....	23
3.3. Αισθητήρας υπερήχων .....	32
<b>4. Ανοιχτός Κώδικας .....</b>	<b>36</b>
4.1. ΕΕΛ/ΛΑΚ.....	36
4.2. Λογισμικό Ανοιχτού Κώδικα – Ελεύθερο Λογισμικό.....	36
4.3. Παράθεση Κώδικα Κατασκευής .....	39
<b>5. Παρουσίαση και διαδικασία κατασκευής.....</b>	<b>49</b>
5.1. Υλικά που χρησιμοποιήθηκαν.....	49
5.2. Συνδεσμολογία κατασκευής .....	58
5.3. Φωτογραφικό υλικό .....	60
<b>6. Επίλογος.....</b>	<b>65</b>
<b>7. Βιβλιογραφία - Πηγές .....</b>	<b>66</b>
7.1. Βιβλιογραφία.....	66
7.2. Ηλεκτρονικές Πηγές.....	66

# 1. ΠΡΟΛΟΓΟΣ

---

Αντικείμενο της παρούσας πτυχιακής εργασίας καθίσταται η διερεύνηση και η αξιοποίηση των δυνατοτήτων της ηλεκτρονικής πλατφόρμας Arduino για την δημιουργία και κατασκευή ενός επίγειου radar. Στη διαδικασία δημιουργίας και κατασκευής του radar είναι απαραίτητη η γνώση προγραμματιστικής γλώσσας του μικροελεγκτή Arduino, της Wiring C, η οποία αποτελεί παραλλαγή της γλώσσας C++ καθώς και η λειτουργία των μικροσυσκευών που χρησιμοποιήθηκαν σε επίπεδο υλικού.

Το Arduino αποτέλεσε πραγματικά μια μικρή επανάσταση στο χώρο της τεχνολογίας επιτρέποντας στον καθένα, με εξειδικευμένες γνώσεις ή όχι να αναπτύξει πολύ εύκολα, αρκετά γρήγορα και με πολύ χαμηλό κόστος την όποια εφαρμογή θελήσει. Αυτός ο single-board μικροελεγκτής, δηλαδή μια μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή και πολλές εισόδους και εξόδους αποτελεί πλέον το πιο διαδεδομένο δημιουργικό εργαλείο. Χωρίς καμία αμφιβολία, το Arduino είναι το εισιτήριο για κάποιον που ξεκινά για πρώτη φορά στο χώρο των μικροελεγκτών, των ενσωματωμένων συστημάτων ή γενικότερα στον χώρο των σύγχρονων εφαρμογών.

## 2. Arduino

---

### 2.1. Εισαγωγή

Αποτελεί πραγματικότητα το γεγονός ότι, το μεγαλύτερο μέρος του πληθυσμού της γης έχει απολαύσει το παιχνίδι με τηλεκατευθυνόμενο όχημα, διαθέτει πλυντήριο ρούχων στο σπίτι τους, κλιματιστικό για να δροσίζονται το καλοκαίρι ή ακόμα και συναγερμό στο σπίτι ή στο αυτοκίνητο τους για την παροχή ασφάλειας.

Όλα τα παραπάνω αποτελούν παραδείγματα που αποδεικνύουν ότι, μέσα στην καθημερινότητα του ατόμου υπάρχουν κρυμμένοι υπολογιστές, που έχουν καταφέρει να διεισδύσουν στην ζωή, έτσι ώστε να καθιστούν απλούστερες ορισμένες λειτουργίες. Ένας υπολογιστής είναι μια ηλεκτρονική συσκευή που μπορεί να λάβει δεδομένα στην είσοδο του, να τα επεξεργαστεί, και να δώσει τα αποτελέσματα του σε κάποια έξοδο του.

Οπότε όλες οι παραπάνω συσκευές που έχουν αναφερθεί βασίζονται σε τέτοιους «υπολογιστές», μόνο που αυτοί οι υπολογιστές ονομάζονται μικροελεγκτές, επειδή είναι μικροί, ενσωματωμένοι στις συσκευές και διαφοροποιημένοι από τους κανονικούς υπολογιστές γραφείου.

### 2.2. Μικροελεγκτές

Οι μικροελεγκτές είναι ένας τύπος επεξεργαστή, δηλαδή μια άλλη εκδοχή μικροεπεξεργαστή, ο οποίος μπορεί να λειτουργήσει με λιγότερα 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, ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα.

Κανείς δεν θα μπορούσε να αντιτεθεί στην άποψη ότι, η καρδιά ενός κλασικού υπολογιστή γραφείου είναι ο μικροεπεξεργαστής που διαθέτει, και η βασική του λειτουργία είναι η εκτέλεση όλων των εντολών. Ο μικροεπεξεργαστής όμως, δεν μπορεί να λειτουργήσει μόνος του, χρειάζεται την μητρική πλακέτα, την μνήμη ram, τον σκληρό δίσκο, την οθόνη κτλ. Δηλαδή χρειάζεται ένα σύνολο από υποσυστήματα και περιφερειακά για να μπορέσει να γίνει λειτουργικός και να αξιοποιηθεί στον έπακρο βαθμό από τους χρήστες.

Αντίθετα, ένας μικροελεγκτής είναι ένα «ολοκληρωμένο» σύστημα το οποίο εκτός από την επεξεργασία δεδομένων, διαθέτει και εισόδους και εξόδους και μνήμη. Μια βασική διαφορά των μικροελεγκτών από τους υπολογιστές γραφείου είναι ότι οι μικροελεγκτές έχουν την δυνατότητα να παίρνουν πληροφορίες από το φυσικό περιβάλλον μέσω αισθητήρων και μπορούν να ελέγχουν φυσικά αντικείμενα όπως led, κινητήρες κτλ, ενώ αντίθετα ο προσωπικός υπολογιστής γραφείου δεν μπορεί να ανταποκριθεί στις παραπάνω λειτουργίες.

Χαρακτηριστικό παράδειγμα των προαναφερθέντων αποτελεί ένα τυπικό κλιματιστικό δωματίου, όπου μέσω ενός τηλεχειριστήριου, ο χρήστης ενεργοποιεί και καθορίζει την επιθυμητή θερμοκρασία που θέλει και στην συνέχεια ενεργοποιείται ο μικροελεγκτής που μέσω ενός αισθητήρα θερμοκρασίας ελέγχει συνεχώς την θερμοκρασία του χώρου στο οποίο βρίσκεται και με βάση την τιμή που εντοπίζει ενεργοποιεί ή απενεργοποιεί τον συμπιεστή και τον ανεμιστήρα της κεντρικής μονάδας έτσι ώστε να επιτυγχάνεται η επιθυμητή θερμοκρασία στον χώρο.

### 2.3. Εφαρμογές Μικροελεγκτών

Οι μικροελεγκτές χρησιμοποιούνται σε μεγάλο αριθμό εφαρμογών, οι οποίοι συνοψίζονται στους εξής τομείς:

- Βιομηχανικός έλεγχος
- Παιχνιδομηχανές
- Έξυπνα σπίτια
- Απομακρυσμένος έλεγχος και παρακολούθηση χώρου
- Πληροφορική
- Συστήματα αυτόματων συναλλαγών
- Μουσικά εφέ και επεξεργασία ήχου
- Εκπαιδευτικά Ιδρύματα

## 2.4. Ιστορία Arduino

Το 2003 στην πόλη Ιβρέα της Ιταλίας (λίγο έξω από το Τορίνο) ένας φοιτητής στα πλαίσια μια εργασίας ανέπτυξε το Wiring Project. Το project αυτό ήταν η ανάπτυξη ενός απλού συστήματος βασισμένου σε μικροελεγκτή που να προσφέρεται για εύκολη και γρήγορη ανάπτυξη εφαρμογών. Το Arduino ήταν μια επέκταση του wiring project με σκοπό να χρησιμοποιηθεί από ανθρώπους μη σχετικούς με την τεχνολογία. Με αυτό το στόχο το Arduino αποτέλεσε ταυτόχρονα και το καλύτερο ξεκίνημα για φοιτητές και μηχανικούς. Το Arduino πήρε το όνομα του από τον Arduino που ήταν στρατιωτικός διοικητής της Ιβρέα και αργότερα βασιλιάς της Ιταλίας.

Το Arduino είναι μια πλατφόρμα ανάπτυξης έργων (φυσικού υπολογισμού), ανοιχτού κώδικα. Η πλατφόρμα περιλαμβάνει μία απλή πλακέτα μικροελεγκτή (τα αντίστοιχα σχέδια είναι δωρεάν) μαζί με ένα περιβάλλον ανάπτυξης προγραμμάτων (επίσης δωρεάν) για τον προγραμματισμό του μικροελεγκτή. Το Arduino χρησιμοποιεί μικροελεγκτές Atmel AVR ενώ υπάρχουν διάφορες παραλλαγές με διαφορετική έκδοση του μικροελεγκτή που διαθέτουν περισσότερες ή λιγότερες δυνατότητες. Η γλώσσα που χρησιμοποιείται για τον προγραμματισμό του μικροελεγκτή είναι η Wiring C, η οποία αποτελεί παραλλαγή της γνωστής γλώσσας C++ ενώ μέσα από την προσθήκη βιβλιοθηκών υποστηρίζονται περισσότερες λειτουργίες και δυνατότητες. Το περιβάλλον ανάπτυξης που χρησιμοποιεί το Arduino είναι δανεισμένο από τη γλώσσα Processing.

## 2.5. Πλεονεκτήματα Arduino

Τα βασικότερα πλεονεκτήματα του Arduino είναι:

- **Χαμηλό Κόστος.** Μπορεί κάποιος να αγοράσει για παράδειγμα το Arduino Uno R3 με κόστος μικρότερο από 12€ (eBay) και δωρεάν μεταφορικά. Επίσης, για ευκολία στους αρχάριους χρήστες υπάρχουν τα λεγόμενα starter kit που περιλαμβάνουν εκτός από την πλακέτα και μία πλειάδα από αισθητήρες, καλώδια, διακόπτες, οθόνες, led κτλ που κοστίζουν ελάχιστα πάλι με δωρεάν μεταφορικά. Επίσης για τον προγραμματισμό του, το περιβάλλον ανάπτυξης είναι ανοιχτού κώδικα και μπορεί να το κατεβάσει κάποιος δωρεάν από το επίσημο site του Arduino. (<https://www.arduino.cc/en/Main/Software>)
- **Συμβατότητα.** Το λογισμικό του Arduino εκτελείται σε Windows, Linux και Mac Os
- **Απλότητα.** Το περιβάλλον ανάπτυξης είναι αρκετά απλό ώστε να χρησιμοποιείται εύκολα και από αρχάριους χρήστες.
- **Επεκτάσιμο λογισμικό ανοιχτού κώδικα.** Τα προγράμματα του Arduino είναι ανοιχτού κώδικα που σημαίνει ότι οποιοσδήποτε μπορεί να βρει τμήματα πηγαίου κώδικα, να τα μελετήσει, να τα χρησιμοποιήσει και να τα τροποποιήσει σύμφωνα με τις ανάγκες του. Επίσης, μέσα από την προσθήκη βιβλιοθηκών C++ μπορούν να προστεθούν δυνατότητες στο περιβάλλον ανάπτυξης. Κάποιος πεπειραμένος χρήστης μπορεί να γράψει τον δικό του κώδικα χαμηλού επιπέδου και να τον καταστήσει διαθέσιμο μέσω βιβλιοθηκών, ενώ αντίθετα οι αρχάριοι χρήστες μπορούν να προσθέσουν υποστήριξη για νέες διατάξεις (αισθητήρες, κτλ) χωρίς να χρειάζεται να γνωρίζουν τις χαμηλού επιπέδου λεπτομέρειες του προγραμματισμού των διατάξεων αυτών. Ακόμα και μέσα από την κοινότητα μπορούν να πάρουν πολύτιμη βοήθεια για το project που θέλουν. (<http://forum.arduino.cc/>)

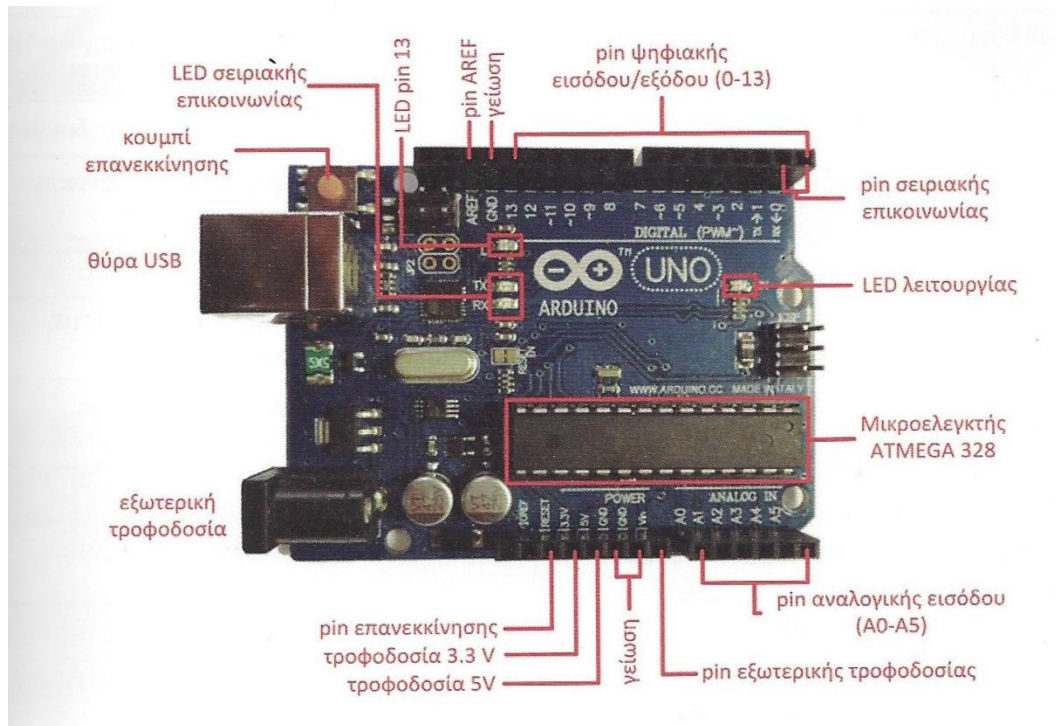
## 2.6. Εκδόσεις Arduino

Υπάρχουν πολλές εκδόσεις και παραλλαγές του Arduino. Κάθε board υποστηρίζει συγκεκριμένο τύπο μικροελεγκτή και διαθέτει διαφορετικό σύνολο ακροδεκτών για εισόδους/εξόδους. (Αναφέρονται μόνο οι Official Arduino boards, modules, shields, kits. <https://www.arduino.cc/en/Main/Products>)

Entry Level	Enhanged Features	Internet of things	Wearable	3D Printing
Arduino UNO	Arduino Mega	Arduino Yun	Arduino Gemma	Materia 101
Arduino 101	Arduino Zero	Arduino MKR100	Lilypad Arduino USB	
Arduino Pro	Arduino Due	Arduino Ethernet Shield	Lilypad Arduino Main Board	
Arduino Pro Mini	Arduino Proto Shield	Arduino GSM Shield	Lilypad Arduino Simple	
Arduino Micro		Arduino WiFi Shield 101	Lilypad Arduino Simple Snap	
Arduino Nano				
Arduino Starter Kit				
Arduino Basic Kit				
Arduino Motor Shield				



## 2.7. Χαρακτηριστικά Arduino UNO



Εικόνα.1

Ακολουθεί η αποτύπωση των βασικών χαρακτηριστικών του Arduino UNO.

Μικροελεγκτής	ATmega328
Τάση λειτουργίας	5V
Τάση εξωτερικής τροφοδοσίας	7-12V
Όρια εξωτερικής τροφοδοσίας	6-20V
Ψηφιακές εισοδοί/έξοδοι (μέσω ακροδεκτών)	14 (εκ των οποίων 6 υποστηρίζουν PWM)
Αναλογικές εισοδοί (μέσω ακροδεκτών)	6
Μέγιστο συνεχές ρεύμα ανά ακροδέκτη εισόδου/εξόδου	40mA
Μέγιστο ρεύμα για τον ακροδέκτη 3,3V	50mA
Flash Memory	32KB (ATmega328) εκ των οποίων τα 0.5KB χρησιμοποιούνται από τον bootloader
SRAM	2KB (ATmega328)
EEPROM	1KB (ATmega328)
Ταχύτητα ρολογιού	16MHz

Επίσης, παρακάτω παρουσιάζονται τα βασικά χαρακτηριστικά του board συνοπτικά και συγκεντρωμένα σε πίνακα παρότι αναφέρονται και στην Εικόνα.1 πιο πάνω.

<b>Χαρακτηριστικά</b>	<b>Περιγραφή</b>
Κουμπί επανεκκίνησης	Το γνωστό κουμπί reset που αρχικοποιεί ξανά τη λειτουργία board
LED σειριακής επικοινωνίας	Φωτοβολεί όταν γίνεται ανταλλαγή δεδομένων με τον υπολογιστή
LED pin 13	Ενσωματωμένο Led για δυνατότητα άμεσης δοκιμής
Pin AREF	Προσδιορισμός τάσης αναφοράς για ανάγνωση αναλογικών σημάτων
Γείωση	Ακροδέκτης GND
Pin ψηφιακής εισόδου/εξόδου (0-13)	Βασικοί ακροδέκτες ψηφιακής επικοινωνίας με εξωτερικές συσκευές και κυκλώματα
Pin σειριακής επικοινωνίας	Πρόσβαση στα σήματα εκπομπής και λήψης σειριακής επικοινωνίας με τον υπολογιστή
LED λειτουργίας	Ένδειξη λειτουργίας του board
Μικροελεγκτής	Η καρδιά του συστήματος (επεξεργασία εντολών)
Pin αναλογικής εισόδου	Διάβασμα αναλογικών σημάτων
Pin εξωτερικής τροφοδοσίας	Πρόσθετη δυνατότητα τροφοδοσίας
Τροφοδοσία 5V	Τροφοδοσία εξωτερικών κυκλωμάτων 5V
Τροφοδοσία 3,3V	Τροφοδοσία εξωτερικών κυκλωμάτων 3,3V
Pin επανεκκίνησης	Προγραμματιζόμενο reset
Εξωτερική τροφοδοσία	Τροφοδοσία από μπαταρία
Θύρα USB	Επικοινωνία με τον υπολογιστή και παροχή τροφοδοσίας

Είναι σημαντικό να επισημανθεί η εστίαση στο Arduino UNO (πιο συγκεκριμένα στο Arduino UNO R3) διότι αποτελεί το κεντρικό θέμα ανάλυσης της παρούσας πτυχιακής εργασίας. Συνεπώς, είναι δυνατό να θεωρηθεί ως αντικείμενο μελέτης, με σκοπό την πλήρη παρουσίαση του μαγικού κόσμου που προσφέρει.

## 2.8. Arduino UNO

Σημαντικό βήμα της παρούσας πτυχιακής εργασίας είναι η εστίαση σε όλα τα σημεία ενός Arduino UNO, έτσι ώστε με το πέρας ολόκληρης της πτυχιακής εργασίας ο αναγνώστης να είναι σχεδόν έτοιμος να πάρει ένα Arduino και να ξεκινήσει το δικό του project.

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

Επιπλέον, στην πάνω πλευρά του Arduino βρίσκονται 14 θηλυκά pin, αριθμημένα από 0 ως 13, που μπορούν να λειτουργήσουν ως ψηφιακές είσοδοι και έξοδοι. Λειτουργούν στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί το πολύ 40mA. Ως ψηφιακή έξοδος, ένα από αυτά τα pin μπορεί να τεθεί από το πρόγραμμά σε κατάσταση HIGH ή LOW, οπότε το Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι ρεύμα στο συγκεκριμένο pin. Με αυτόν τον τρόπο είναι εφικτό λόγω χάρη το άνοιγμα και το σβήσιμο ενός LED που έχει συνδεθεί στο συγκεκριμένο pin. Αν πάλι ρυθμιστεί ένα από αυτά τα pin ως ψηφιακή είσοδος μέσα από το πρόγραμμά σας, πραγματοποιείται η ανάγνωση της κατάστασης του με την κατάλληλη εντολή (HIGH ή LOW) ανάλογα με το αν η εξωτερική συσκευή που έχει συνδεθεί σε αυτό το pin διοχετεύει ή όχι ρεύμα στο pin (με αυτόν τον τρόπο εντοπίζεται και η κατάσταση ενός διακόπτη). Μερικά από αυτά τα 14 pin, εκτός από ψηφιακές είσοδοι/έξοδοι έχουν και δεύτερη λειτουργία.

Συγκεκριμένα:

- Τα pin 0 και 1 λειτουργούν ως RX και TX της σειριακής όταν το πρόγραμμά σας ενεργοποιεί την σειριακή θύρα. Έτσι, όταν το πρόγραμμά σας αποστέλλει δεδομένα στην σειριακή, αυτά προωθούνται και στην θύρα USB μέσω του ελεγκτή Serial-Over-USB αλλά και στο pin 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή (π.χ. ένα δεύτερο Arduino στο δικό του pin 1). Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμά ενεργοποιηθεί το σειριακό interface, χάνονται 2 ψηφιακές είσοδοι/έξοδοι.
- Τα pin 2 και 3 λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Με άλλα λόγια, μπορούν να ρυθμιστούν ώστε να λειτουργούν αποκλειστικά ως ψηφιακές είσοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει άμεσα

και εκτελείται μια συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

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

Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN, εντοπίζεται μια ακόμη σειρά από 6 pin, αριθμημένα από το 0 ως το 5. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC (Analog to Digital Converter) που είναι ενσωματωμένο στον μικροελεγκτή. Για παράδειγμα, μπορεί να τροφοδοτηθεί ένα από αυτά με μια τάση, η οποία μπορεί να κυμανθεί με ένα ποτενσιόμετρο από 0V ως μια τάση αναφοράς  $V_{ref}$  που αν δεν πραγματοποιηθεί κάποια αλλαγή είναι προρυθμισμένη στα 5V. Τότε, μέσω του προγράμματος πραγματοποιείται η ανάγνωση της τιμής του pin ως ένα ακέραιο αριθμό ανάλυσης 10-bit, από 0 (όταν η τάση στο pin είναι 0V) μέχρι 1023 (όταν η τάση στο pin είναι 5V). Η τάση αναφοράς μπορεί να ρυθμιστεί με μια εντολή στο 1.1V, ή σε κάποια άλλη τάση (μεταξύ 2 και 5V) τροφοδοτώντας εξωτερικά με αυτή την τάση το pin με την σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας. Έτσι, αν τροφοδοτηθεί το pin AREF με 3.3V και στην συνέχεια επιχειρηθεί η ανάγνωση κάποιου pin αναλογικής εισόδου στο οποίο εφαρμόζεται τάση 1.65V, το Arduino θα επιστρέψει την τιμή 512. Τέλος, καθένα από τα 6 αυτά pin, με κατάλληλη εντολή μέσα από το πρόγραμμα μπορεί να μετατραπεί σε ψηφιακό pin εισόδου/εξόδου όπως τα 14 που βρίσκονται στην απέναντι πλευρά και τα οποία περιεγράφηκαν πριν. Σε αυτή την περίπτωση τα pin μετονομάζονται από 0 ~ 5 σε 14 ~ 19 αντίστοιχα.

Το Arduino μπορεί να τροφοδοτηθεί με ρεύμα είτε από τον υπολογιστή μέσω της σύνδεσης USB, είτε από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής φισ των 2.1mm (θετικός πόλος στο κέντρο) και βρίσκεται στην κάτω-αριστερή γωνία του Arduino. Για να μην υπάρχουν προβλήματα, η εξωτερική τροφοδοσία πρέπει να είναι από 7 ως 12V και μπορεί να προέρχεται από ένα κοινό μετασχηματιστή του εμπορίου, από μπαταρίες ή οποιαδήποτε άλλη πηγή DC. Δίπλα από τα pin αναλογικής εισόδου, υπάρχει μια ακόμα συστοιχία από 6 pin με την σήμανση POWER.

Η λειτουργία του καθενός έχει ως εξής:

- Το πρώτο, με την ένδειξη RESET, όταν γειωθεί (σε οποιοδήποτε από τα 3 pin με την ένδειξη GND που υπάρχουν στο Arduino) έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.
- Το δεύτερο, με την ένδειξη 3.3V, μπορεί να τροφοδοτήσει τα εξαρτήματά σας με τάση 3.3V. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial-over-USB και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA.
- Το τρίτο, με την ένδειξη 5V, μπορεί να τροφοδοτήσει τα εξαρτήματά σας με τάση 5V. Ανάλογα με τον τρόπο τροφοδοσίας του ίδιου του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την θύρα USB (που ούτως ή άλλως λειτουργεί στα 5V), είτε από την εξωτερική τροφοδοσία αφού αυτή περάσει από ένα ρυθμιστή τάσης για να την «φέρει» στα 5V.
- Το τέταρτο και το πέμπτο pin, με την ένδειξη GND, είναι φυσικά γειώσεις.
- Το έκτο και τελευταίο pin, με την ένδειξη Vin έχει διπλό ρόλο. Σε συνδυασμό με το pin γείωσης δίπλα του, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino, στην περίπτωση που δεν σας βολεύει να χρησιμοποιήσετε την υποδοχή του φικ των 2.1mm. Αν όμως έχετε ήδη συνδεδεμένη εξωτερική τροφοδοσία μέσω του φικ, μπορείτε να χρησιμοποιήσετε αυτό το pin για να τροφοδοτήσετε εξαρτήματα με την πλήρη τάση της εξωτερικής τροφοδοσίας (7 ~ 12V), πριν αυτή περάσει από τον ρυθμιστή τάσης όπως γίνεται με το pin των 5V.

Πάνω στην πλακέτα του Arduino υπάρχει ένας διακόπτης micro-switch και τέσσερα μικροσκοπικά LED επιφανειακής στήριξης. Η λειτουργία του διακόπτη (που έχει την σήμανση RESET) και του ενός LED με την σήμανση POWER είναι μάλλον προφανής. Τα δύο LED με τις σημάνσεις TX και RX, χρησιμοποιούνται ως ένδειξη λειτουργίας του σειριακού interface, καθώς ανάβουν όταν το Arduino στέλνει ή λαμβάνει (αντίστοιχα) δεδομένα μέσω USB. Σημειώστε ότι τα LED αυτά ελέγχονται από τον ελεγκτή Serial-over-USB και συνεπώς δεν λειτουργούν όταν η σειριακή επικοινωνία γίνεται αποκλειστικά μέσω των ψηφιακών pin 0 και 1. Τέλος, υπάρχει το LED με την σήμανση L. Η βασική δοκιμή λειτουργίας του Arduino είναι να αναβοσβήνει ένα LED. Για να πραγματοποιηθεί αυτό από την πρώτη στιγμή χωρίς να συνδεθεί τίποτα πάνω στο Arduino, οι κατασκευαστές σκέφτηκαν να ενσωματώσουν ένα LED στην πλακέτα, το οποίο σύνδεσαν στο ψηφιακό pin 13. Έτσι, ακόμα και αν δεν έχει συνδεθεί τίποτα πάνω στο φυσικό pin 13, αναθέτοντάς του την τιμή HIGH μέσα από το πρόγραμμά θα ανάψει αυτό το ενσωματωμένο LED. Τέλος τα ICSP pins είναι για τον προγραμματισμό του Arduino απευθείας με τον ISP/ICSP programmer. Επίσης, τα 6 pins AREF είναι για το USB system, η διασύνδεση USB έχει τον δικό του μικροελεγκτή, που για το Arduino Uno, ATmega16u2 είναι AFAICT.

Το Arduino IDE, παρέχει ότι χρειάζεται ένας χρήστης για την κατασκευή του Arduino, η τελευταία έκδοση του οποίου ( Arduino IDE) διατίθεται σε ένα από τα δημοφιλέστερα λειτουργικά συστήματα.

Το Arduino IDE είναι βασισμένο σε Java και συγκεκριμένα παρέχει:

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

Για τα δύο τελευταία χαρακτηριστικά βέβαια, το Arduino πρέπει να έχει συνδεθεί σε μια από τις θύρες USB του υπολογιστή και, λόγω του ελεγκτή Serial-over-USB, θα πρέπει να αναγνωριστεί από το λειτουργικό σας σύστημα ως εικονική σειριακή θύρα.

Για την σύνδεση απαιτείται ένα καλώδιο USB από Type A σε Type B, όπως αυτό των εκτυπωτών. Για την αναγνώριση από το λειτουργικό χρειάζεται η εγκατάσταση του οδηγού του FTDI chip (δηλαδή του ελεγκτή Serial-over-USB) ο οποίος υπάρχει στον φάκελο drivers του Arduino IDE. Την τελευταία έκδοση αυτού του οδηγού μπορεί κάποιος να προμηθευτεί για κάθε λειτουργικό σύστημα από το site της FTDI (<http://www.ftdichip.com/Drivers/VCP.htm>)

Στους τελευταίους πυρήνες του Linux υπάρχει εγγενής υποστήριξη του συγκεκριμένου ελεγκτή. Αν όλα έγιναν σωστά, το κεντρικό παράθυρο του Arduino IDE θα εμφανιστεί όταν εκτελεστεί και στο μενού Tools → Serial Port θα πρέπει να εμφανίζεται η εικονική σειριακή θύρα (συνήθως COM# για τα Windows, /dev/ttyusbserial## για το MacOS και /dev/ttyusb## για το Linux). Ο χρήστης οφείλει να επιλέξει αυτή την εικονική θύρα και στην συνέχεια τον τύπο του Arduino που τον ενδιαφέρει (Arduino Duemilanove w/ ATmega328) από το μενού Tools → Board. Το Arduino είναι πλέον έτοιμο να δεχτεί τα sketch. Τέλος αν εμφανιστεί οποιοδήποτε πρόβλημα οι αναλυτικές οδηγίες για κάθε λειτουργικό σύστημα είναι διαθέσιμες στη διεύθυνση <http://arduino.cc/en/Guide/HomePage>.

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

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

Όρισμα	Είδος	Τύπος	Παράμετροι	Περιγραφή
LOW	Σταθερά	int	-	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
HIGH	Σταθερά	int	-	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
INPUT	Σταθερά	int	-	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
OUTPUT	Σταθερά	int	-	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
pinMode	Εντολή	-	( <i>pin, mode</i> )	Καθορίζει αν το συγκεκριμένο ψηφιακό <i>pin</i> θα είναι <i>pin</i> εισόδου ή <i>pin</i> εξόδου ανάλογα με την τιμή που δίνεται στην παράμετρο <i>mode</i> (INPUT ή OUTPUT αντίστοιχα).
digitalWrite	Εντολή	-	( <i>pin, pinstatus</i> )	Θέτει την κατάσταση <i>pinstatus</i> (HIGH ή LOW) στο συγκεκριμένο ψηφιακό <i>pin</i> .
digitalRead	Συνάρτηση	int	( <i>pin</i> )	Επιστρέφει την κατάσταση του συγκεκριμένου ψηφιακού <i>pin</i> (0 για LOW και 1 για HIGH) εφόσον αυτό είναι <i>pin</i> εισόδου.
analogReference	Εντολή	-	( <i>type</i> )	Δέχεται τις τιμές DEFAULT, INTERNAL ή EXTERNAL στην παράμετρο <i>type</i> για να καθορίσει την τάση αναφοράς ( $V_{ref}$ ) των αναλογικών εισόδων (5V, 1.1V ή η εξωτερική τάση με την οποία τροφοδοτείται το <i>pin</i> AREF αντίστοιχα)
analogRead	Συνάρτηση	int	( <i>pin</i> )	Επιστρέφει έναν ακέραιο από 0 έως 1023, ανάλογα με την τάση που τροφοδοτείται το συγκεκριμένο <i>pin</i> αναλογικής εισόδου στην κλίμακα 0 ως $V_{ref}$ .
analogWrite	Εντολή	-	( <i>pin, value</i> )	Θέτει το συγκεκριμένο ψηφιακό <i>pin</i> σε κατάσταση ψευδοαναλογικής εξόδου (PWM). Η παράμετρος <i>value</i> καθορίζει το πλάτος του παλμού σε σχέση με την περίοδο του παραγόμενου σήματος στην

κλίμακα από 0 ως 255 (π.χ. με *value* 127, το πλάτος του παλμού είναι ίσο με μισή περίοδο).

millis	Συνάρτηση	unsigned long	()	Μετρητής που επιστρέφει το χρονικό διάστημα σε ms από την στιγμή που άρχισε η εκτέλεση του προγράμματος. Λάβετε υπόψη ότι λόγω του τύπου μεταβλητής (unsigned long δηλ. 32bit) θα γίνει overflow σε 2 <sup>32</sup> ms δηλαδή περίπου σε 50 μέρες, οπότε ο μετρητής θα ξεκινήσει πάλι από το μηδέν.
delay	Εντολή	-	(time)	Σταματά προσωρινά την ροή του προγράμματος για <i>time</i> ms. Η παράμετρος <i>time</i> είναι unsigned long (από 0 ως 2 <sup>32</sup> ). Σημειώστε ότι παρά την προσωρινή παύση, συναρτήσεις των οποίων η εκτέλεση ενεργοποιείται από interrupt θα εκτελεστούν κανονικά κατά την διάρκεια μιας delay.
attachInterrupt	Εντολή	-	(interrupt,function,triggermode)	Θέτει σε λειτουργία το συγκεκριμένο <i>interrupt</i> , ώστε να ενεργοποιεί την συνάρτηση <i>function</i> , κάθε φορά που ικανοποιείται η συνθήκη που ορίζεται από την παράμετρο <i>triggermode</i> : <ul style="list-style-type: none"> <li>• LOW (ενεργοποίηση όταν η κατάσταση του pin που αντιστοιχεί στο συγκεκριμένο interrupt γίνει LOW)</li> <li>• RISING (όταν από LOW γίνει HIGH)</li> <li>• FALLING (όταν από HIGH γίνει LOW)</li> <li>• CHANGE (όταν αλλάξει κατάσταση)</li> </ul>
detachInterrupt	Εντολή	-	(interrupt)	Απενεργοποιεί το συγκεκριμένο <i>interrupt</i> .
noInterrupts	Εντολή	-	()	Σταματά προσωρινά την λειτουργία όλων των interrupt
interrupts	Εντολή	-	()	Επαναφέρει την λειτουργία των interrupt που διακόπηκε προσωρινά από μια εντολή



noInterrupts.

Serial.begin	Μέθοδος κλάσης	-	( <i>datarate</i> )	Θέτει τον ρυθμό μεταφοράς δεδομένων του σειριακού interface (σε baud)
Serial.println	Μέθοδος κλάσης	-	( <i>data</i> )	Διοχετεύει τα δεδομένα <i>data</i> για αποστολή μέσω του σειριακού interface. Η παράμετρος <i>data</i> μπορεί να είναι είτε αριθμός είτε αλφαριθμητικό.

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

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

```
void setup()
{
  // ...
}
```

```
void loop()
{
  // ...
}
```

```
// Υπόλοιπες συναρτήσεις...
```

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

Αν και πρόκειται μόνο για τις πιο βασικές λειτουργίες της γλώσσας του Arduino, με αυτές και λίγες βασικές γνώσεις C είναι εφικτή η δημιουργία ενός sketch ακόμα και για κάποιο αρκετά περίπλοκο project.

Επιπλέον παρέχονται μαζί και κάποιες βιβλιοθήκες που οι εξής:

- **EEPROM:** Ανάγνωση και εγγραφή σε μόνιμη αποθήκευση
- **Ethernet:** Εφαρμόζεται για την σύνδεση με τα διαδίκτυο αλλά χρησιμοποιώντας Arduino Ethernet Shield.
- **Wire:** Η διεπαφή δύο καλωδίων (TWI/I2C) έχει την δυνατότητα αποστολής και λήψης των δεδομένων εκτός των συσκευών και των αισθητήρων.
- **Firmata:** Χρησιμοποιείται στην επικοινωνία με τις εφαρμογές του υπολογιστή, που χρησιμοποιείται ένα τυποποιημένο τμηματικό πρωτόκολλο.
- **Software Serial:** Εφαρμόζεται για την τμηματική ανακοίνωση σχετικά με οποιοσδήποτε ψηφιακές εισόδους.
- **Liquid Crystal:** Εφαρμόζεται για τον έλεγχο επιδείξεων υγρού κρυστάλλου (LCDs οθόνες)
- **Servo:** Εφαρμόζεται για τον έλεγχο των servo μηχανών.
- **Stepper:** Εφαρμόζεται για τον έλεγχο των stepper κινητήρων.

## 2.9. Processing

Το Processing αποτελεί μια γλώσσα προγραμματισμού ανοικτού κώδικα (όπως και το Arduino) ταυτόχρονα όμως και ένα πρόγραμμα για ανθρώπους που επιθυμούν να προγραμματίσουν εικόνα και ήχο. Το 2001 δύο απόφοιτοι του πανεπιστημίου MIT ο Benjamin Fry και ο Casey Reas ξεκίνησαν την ανάπτυξη της γλώσσας Processing πάνω σε Java. Οι δύο αυτές εξέχουσες προσωπικότητες οραματίζονταν το πρόγραμμα Arduino με την ίδια λογική, η οποία επεκτείνεται στην δυνατότητα εκμάθησης του προγραμματισμού από αρχάριους χρήστες και στην παροχή ενός επαγγελματικού εργαλείου παραγωγής πολυμεσικών εφαρμογών. Το περιβάλλον του Processing είναι γραμμένο σε Java. Τα προγράμματα που είναι γραμμένα σε Java και Processing, τρέχουν γρηγορότερα από προγράμματα που βασίζονται σε scripting γλώσσες, κάτι το οποίο είναι πολύ σημαντικό για εφαρμογές με γραφικά. Οι διαφορές μεταξύ του Processing και της Java είναι οι βιβλιοθήκες γραφικών του Processing και το απλοποιημένο στυλ προγραμματισμού που δεν προϋποθέτει οι χρήστες να καταλαβαίνουν πιο εξειδικευμένες έννοιες όπως κλάσεις, αντικείμενα, ή animation (ενώ εξακολουθούν να είναι διαθέσιμες για προχωρημένους χρήστες). Τέτοιες τεχνικές λεπτομέρειες πρέπει να είναι ειδικά προγραμματισμένες σε Java, αλλά ολοκληρωμένες σε Processing, κάνοντας τα προγράμματα μικρότερα και ευκολότερα να διαβαστούν.

Επίσης στην processing υπάρχουν βιβλιοθήκες, οι οποίες είναι και αυτές ανοιχτού κώδικα και είναι οι εξής:

- **Arduino:** Επιτρέπει την επικοινωνία, τον έλεγχο και τον καθορισμό του μικροελεγκτή Arduino.
- **Serial:** Επιτρέπει το read και το write δεδομένων από εξωτερικές συσκευές, με περιορισμό ένα byte κάθε φορά. Επιπλέον, παρέχει την δυνατότητα σε δύο υπολογιστές να λάβουν και να αποστέλλουν δεδομένα και τέλος προσφέρει ευελιξία καθώς μπορεί να χρησιμοποιήσει μικροελεγκτές σαν συσκευές εισόδου και εξόδου.
- **Network:** Επιτρέπει την ανάγνωση και εγγραφή δεδομένων σε όλες τις μηχανές στο διαδίκτυο. Ακόμη, προσφέρει την δημιουργία client-server.
- **Netscape Javascript:** Επιτρέπει την διασύνδεση προγραμμάτων σε γλώσσα Java με την Processing.
- **Video:** Επιτρέπει την αναπαραγωγή αρχείων βίντεο, την χρήση αρχείων βίντεο από συσκευές και την δημιουργία βίντεο από προγράμματα που τρέχουν στον υπολογιστή.
- **Minim:** Επιτρέπει την χρήση αρχείων ήχου μέσω προγραμμάτων JavaSound API, Javazoom's και MP3SPI.
- **PDF Export:** Επιτρέπει την δημιουργία αρχείων PDF κατευθείαν μέσω του processing. Μέσα στα αρχεία είναι δυνατόν να συμπεριληφθούν γραφικά σε οποιαδήποτε κλίμακα ακόμα και σε πολύ υψηλή ανάλυση και μπορούν να μετατραπούν δεδομένα 3D σε 2D, με απαραίτητη την χρήση της βιβλιοθήκης DXF με σκοπό την εξαγωγή των συγκεκριμένων δεδομένων.
- **DXF Export:** Επιτρέπει την δημιουργία DXF αρχείων ( .dxf είναι αρχεία autocad) και τον σχεδιασμό σχημάτων χρησιμοποιώντας τις συναρτήσεις beginRaw() και endRaw().

Το περιβάλλον προγραμματισμού του Processing αποτελείται από ένα κειμενογράφο όπου πραγματοποιείται η καταγραφή του επιθυμητού κώδικα, η δημιουργία ενός πεδίου μηνυμάτων, η κατασκευή καρτελών για την διατήρηση πολλαπλών ανοιχτών αρχείων και η εκμετάλλευση της γραμμής εργαλείων, και του μενού. Στο περιβάλλον του Processing τα προγράμματα που δημιουργούνται ονομάζονται sketch και αποθηκεύονται στο sketchbook που αποτελεί έναν φάκελο του υπολογιστή. Για την εκτέλεση των προγραμμάτων μπορεί να δημιουργηθεί αρχείο πηγαίνοντας στο μενού File->Export Application όπου επιλέγεται το λειτουργικό σύστημα στο οποίο θα τρέχει το πρόγραμμά και πατώντας την ένδειξη Export. Το Processing έχει τρεις διαφορετικούς τρόπους προγραμματισμού (modes) ώστε να γίνει εφικτή η ανάπτυξη των προγραμμάτων (sketches) σε διάφορες πλατφόρμες. Τα τρία modes είναι το Java, το JavaScript και το Android. Το Java mode δημιουργεί εφαρμογές της Java. Το JavaScript παράγει εφαρμογές που τρέχουν μέσω HTML5 και WebGL, και το Android mode δημιουργεί εφαρμογές οι οποίες τρέχουν σε Smartphone με λειτουργικό σύστημα Android όπως και σε ταμπλέτες με λειτουργικό σύστημα Android.

## 3. Radar / Σερβοκινητήρας / Αισθητήρας υπερήχων

---

### 3.1. Radar

Ο ραδιοεντοπιστής ή γνωστότερο με το διεθνές όνομα Radar που προέρχεται από συντόμευση των αγγλικών λέξεων "RADio Detection And Ranging" σημαίνει ανίχνευση με ηλεκτρομαγνητικά κύματα και μέτρηση αποστάσεως. Αποτελεί ένα βασικό ηλεκτρονικό σύστημα ηλεκτρομαγνητικού εντοπισμού, παρακολούθησης ακίνητων και κινητών στόχων, σε αποστάσεις και συνθήκες φωτισμού απαγορευτικές για τον απευθείας οπτικό εντοπισμό, δηλαδή με το ανθρώπινο μάτι ή και οπτικά όργανα. Η μεγάλη αξία του ραντάρ οφείλεται στις σημαντικές δυνατότητες ανίχνευσης και παρακολούθησης στόχων σε μεγάλες αποστάσεις και με μεγάλη ακρίβεια. Η αρχή λειτουργίας του ραντάρ βασίζεται στην εκπομπή και λήψη (επιστροφών) των ηλεκτρομαγνητικών κυμάτων μετά από ανάκλαση σε κάποιο αντικείμενο.

Από το 1920 μια σημαντική ανακάλυψη έρχεται στο προσκήνιο, χάρη στον εφευρέτη Albert W. Hull. Πρόκειται για την πρώτη λυχνία magnetron. Η λειτουργία της βασίζεται στην παραγωγή υψηλής ισχύος ηλεκτρομαγνητικών σημάτων μικροκυμάτων. Καθώς τα σήματα εξέρχονται από μία πηγή, δύναται να χρησιμοποιηθούν προκειμένου να μεταδώσουν σήματα σε μεγάλες αποστάσεις. Η εφεύρεση αυτή συνέβαλε σε μεγάλο βαθμό στην εξέλιξη της ασύρματης επικοινωνίας. Αυτό, όπως ήταν φυσικό οδήγησε στην βελτίωση των τεχνικών οι οποίες μετατρέποντας τα βασικά στοιχεία του σήματος που εκπέμπεται, επιτυγχάνεται η μετάδοση της πληροφορίας από ένα σημείο σε κάποιο άλλο. Οι αποστάσεις επικοινωνίας αυξάνονται ενώ η ποιότητα των σημάτων γίνεται καλύτερη. Το 1904 ο Γερμανός Christian Hülsmeyer κατοχύρωσε με δίπλωμα ευρεσιτεχνίας ένα μέσο χρήσης του ραντάρ, αλλά η εφεύρεση του προσέλκυσε πολύ μικρό ενδιαφέρον. Όμως στα τέλη της δεκαετίας του 1930, η απειλή της επίθεσης του αέρα αύξησε την εργασία σε αυτή την τεχνολογία και έτσι ερευνητικές ομάδες σε τουλάχιστον οκτώ χώρες όπως: η Γαλλία, η Γερμανία, η Ιταλία, η Ιαπωνία, η Ολλανδία, η Σοβιετική Ένωση, το Ηνωμένο Βασίλειο και οι Ηνωμένες Πολιτείες ενδιαφέρθηκαν για την κατασκευή ραντάρ. Ακόμη και πριν από το ξέσπασμα του πολέμου, η Βρετανία είχε κατασκευάσει ένα σύστημα ραντάρ αεράμυνας που λεγόταν Chain Home. Στις Ηνωμένες Πολιτείες, οι ερευνητές στο Rad Lab του MIT και το Ναυτικό Ερευνητικό Εργαστήριο των ΗΠΑ έσπευσαν στην δημιουργία ραντάρ.

Το Ραντάρ το οποίο ουσιαστικά "βλέπει" με ραδιοκύματα καθορίζει την απόσταση σε ένα αντικείμενο με τη μέτρηση του χρόνου που απαιτείται για ένα ραδιοφωνικό σήμα να ταξιδέψει από έναν πομπό προς το αντικείμενο και να επιστρέψει. Τέτοιες μετρήσεις μπορούν να μετατραπούν σε γραμμές και θέσεις (LOP) που αποτελείται από κύκλους με ακτίνα ίση με την απόσταση προς το αντικείμενο. Τα θαλάσσια ραντάρ χρησιμοποιούν κατευθυντήριες κεραιές, επιτυγχάνοντας τον προσανατολισμό ενός αντικειμένου. Το ραντάρ δικαίως χαρακτηρίζεται το κατεξοχήν μέσο σύνθεσης της εικόνας της ναυτιλιακής καταστάσεως. Σήμερα, το ραντάρ δεν έχει μόνο βελτιωθεί, αλλά χρησιμοποιείται και για διάφορους άλλους σκοπούς. Τα συστήματα ραντάρ, ανάλογα με την πλατφόρμα/φορέα και τον επιθυμητό χώρο ραδιοεντοπισμού διακρίνονται σε:

- **Ραντάρ Ανιχνεύσεως Αέρα:** Εγκαθίστανται στο έδαφος (σε βουνοκορφές ή κοντά σε αεροδρόμια) και σε πλοία. Τα συγκεκριμένα ραντάρ στοχεύουν στην ανίχνευση του εναέριου χώρου σε μεγάλες αποστάσεις και σε μεγάλα ύψη. Με το Ραντάρ ανιχνεύσεως αέρα εξασφαλίζεται ο έλεγχος της εναέριας κυκλοφορίας για την εξασφάλιση της δυνατότητας προσανατολισμού των αεροσκαφών, και ο εντοπισμός εχθρικών αεροσκαφών σε μεγάλες αποστάσεις.
- **Ραντάρ Ανιχνεύσεως Επιφάνειας:** Τα ραντάρ ανιχνεύσεως επιφάνειας ή αλλιώς ραντάρ ναυσιπλοΐας, εγκαθίστανται σε ακτές ή σε πλοία και ανιχνεύουν την επιφάνεια της θάλασσας. Εντοπίζουν όμως και τον εναέριο χώρο, αλλά σε μικρό ύψος. Επιπλέον, ανιχνεύουν στερεά αντικείμενα από σχετικά αγωγίμο υλικό (στόχους), που βρίσκονται στην επιφάνεια της θάλασσας ή σε μικρό ύψος και παρέχουν ακριβείς πληροφορίες αποστάσεων και διοπτεύσεων των στόχων που εντοπίζουν. Η ανίχνευση των στόχων επιτυγχάνεται ανεξάρτητα από τις συνθήκες ορατότητας και σε αποστάσεις μεγαλύτερες από αυτές του ορατού ορίζοντα. Τα ραντάρ ανιχνεύσεως επιφάνειας χρησιμοποιούνται για να διασφαλίζουν την ασφαλή ναυσιπλοΐα, τον εντοπισμό εχθρικών ή μη πλοίων και την ύπαρξη ή μη αντικειμένων (στόχων) στην επιφάνεια της θάλασσας.
- **Ραντάρ Ελέγχου Προσγειώσεως Αεροσκαφών:** Εγκαθίστανται κοντά σε διαδρόμους αεροδρομίων ή σε αεροπλανοφόρα πλοία. Έχουν μικρή εμβέλεια, αλλά μεγάλη ακρίβεια και παρέχουν πληροφορίες αποστάσεως, ύψους, κατευθύνσεως διαδρόμου προσγειώσεως και ίχνους καθόδου. Οι πληροφορίες αυτές μεταδίδονται από το χειρίστη του ραντάρ με ραδιοτηλέφωνο στα αεροσκάφη και έτσι εξασφαλίζεται ασφαλής προσγείωση, ακόμη και σε περίπτωση εντελώς ανύπαρκτης ορατότητας.
- **Ραντάρ Μετρήσεως Ταχύτητας:** Με τα συγκεκριμένα ραντάρ επιτυγχάνεται η ακριβής μέτρηση της ταχύτητας οχημάτων στους αυτοκινητόδρομους και

διαπιστώνεται η τήρηση ή μη του ορίου ταχύτητας, καθώς και η μέτρηση της ταχύτητας των πλοίων σε θαλάσσιες περιοχές που ισχύει συγκεκριμένο όριο ταχύτητας.

- **Ραντάρ Ανίχνευσης Κίνησης:** Ο ρόλος των συγκεκριμένων ραντάρ έγκειται στην ανίχνευση κινήσεων σε σταθερούς χώρους και στην ενεργοποίηση συναγερμών, φώτων και διαφόρων άλλων μηχανισμών.
- **Ραντάρ Ελέγχου Πυρός:** Τα ραντάρ ελέγχου πυρός αποτελούν μέρος των διάφορων τύπων οπλικών συστημάτων, και παρέχουν τα απαραίτητα στοιχεία βολής και διόρθωσης της κατευθύνσεως κίνησης ορισμένων τύπων τηλεκατευθυνόμενων βλημάτων.
- **Υψομετρικά Ραντάρ:** Εγκαθίστανται σε αεροσκάφη και εξασφαλίζουν ακριβή μέτρηση του ύψους πτήσεως τους.
- **Μετεωρολογικά Ραντάρ:** Εξασφαλίζεται ο έγκαιρος εντοπισμός και η παρακολούθηση των επερχομένων καταιγίδων και κυκλώνων.

Αντικείμενο της παρούσας εργασίας αποτελεί το ραντάρ ανίχνευσης κίνησης, που παρέχει αρκετές εφαρμογές στην καθημερινή ζωή. Το συγκεκριμένο ραντάρ εντοπίζεται σε συναγερμούς κατοικίας που ενεργοποιούν τον φωτισμό ή τον γενικό συναγερμό, στα καινούργια αυτοκίνητα (το parking assistant είναι μία τέτοια τεχνολογία). Επιπλέον, εμφανίζονται καινούριες οικιακές ηλεκτρικές σκούπες που σκουπίζουν μόνες τους και έχουν ένα τέτοιο σύστημα έτσι ώστε να αποφεύγουν εμπόδια και σε πολλά καταστήματα υπάρχουν τέτοια συστήματα που ενεργοποιούνται φώτα και απενεργοποιούνται μόλις δεν υπάρχει κίνηση έτσι ώστε να εξοικονομούν ενέργεια. Επομένως, η μελέτη και η κατασκευή ενός ραντάρ στα πλαίσια της παρούσας πτυχιακής εργασίας είναι δυνατό να αποτελέσουν ένα ισχυρό κίνητρο και ερέθισμα σε άλλους φοιτητές ή μηχανικούς που ενδιαφέρονται για το αντικείμενο, διότι στο μέλλον πεδίο έρευνας θα είναι τέτοιου είδους αισθητήρες, για να ελέγχονται όλες οι λειτουργίες μέσων υπολογιστών με σκοπό την κατανάλωση λιγότερης ενέργειας.

### 3.2. Σερβοκινητήρας

Στον τομέα του ελέγχου κίνησης και ειδικότερα σε εφαρμογές ελέγχου θέσεως ταχύτητας και ροπής άξονα χρησιμοποιούνται εδώ και αρκετά χρόνια ειδικοί κινητήρες που λέγονται σερβοκινητήρες. Η επιτυχία στην δημιουργία των σερβοκινητήρων έγκειται στην δυνατότητα εκτός από την απλή ρύθμιση των στροφών και της ταχύτητας που γίνονται και στους συνήθεις κινητήρες με ρυθμιστές στροφών και inverters να γίνεται πραγματικός έλεγχος της θέσης του άξονα του κινητήρα (positioning).

Η παραπάνω δυνατότητα γίνεται εφικτή μέσω της προσθήκης ενός συγκεκριμένου αισθητηρίου στο σώμα του κινητήρα (resolver ή encoder) που ελέγχει με συγκεκριμένες διαδικασίες την ταχύτητα και τη θέση του άξονα του κινητήρα. Με την εξέλιξη αυτή η τεχνολογία πέρασε από τον κλασσικό έλεγχο της ταχύτητας (speed control) στο συνολικό έλεγχο της κίνησης (motion control). Αυτή η εφαρμογή αποτέλεσε κομβικό σημείο στην εξέλιξη της ρομποτικής. Οι σερβοκινητήρες δεν μπορούν να επιτελέσουν την ουσιαστική λειτουργία τους εκτελώντας μεμονωμένες κινήσεις, αλλά αποτελούν το βασικό στοιχείο ενός συνολικού συστήματος αυτοματισμού που επιτελεί motion control. Στο συνολικό σερβοσύστημα εντάσσονται εκτός από το σερβοκινητήρα, η μονάδα ελέγχου που στην πλειοψηφία των περιπτώσεων είναι ένα PLC και ο σερβοενιχυτής (servodrive) που αποτελεί τον ενδιάμεσο κρίκο μεταξύ της μονάδας ελέγχου και του σερβοκινητήρα. Οι σερβοκινητήρες είναι οι κινητήρες που χρησιμοποιούνται στα Συστήματα Αυτομάτου Ελέγχου, χωρίς φυσικά να ταυτίζονται με τους κοινούς κινητήρες παρόλο που μοιάζουν κατασκευαστικά. Οι σερβοκινητήρες διαφέρουν από τους άλλους κινητήρες στο ότι ενσωματώνουν ένα σύστημα ανάδρασης το οποίο χρησιμοποιείται σε συνδυασμό με έναν σερβομηχανισμό οδήγησης με σκοπό να ελεγχθεί είτε η ροπή είτε η ταχύτητα. Ο σερβομηχανισμός οδήγησης συνίσταται στον σερβοενιχυτή (servodrive) που αποτελεί ένα ξεχωριστό στοιχείο του σερβοσυστήματος. Ωστόσο, το σύστημα ανάδρασης είναι ενσωματωμένο στο σερβοκινητήρα. Οι περισσότεροι σερβοκινητήρες διαθέτουν μια ενσωματωμένη παλμογεννήτρια (encoder ή pulse generator), αυξητικού ή απόλυτου τύπου (incremental ή absolute type). Αν η παλμογεννήτρια είναι αυξητικού τύπου τότε παράγουν ένα συγκεκριμένο αριθμό παλμών ανά περιστροφή του άξονα του σερβοκινητήρα, ενώ αν είναι απόλυτου θέσεως τότε δίνουν στην έξοδό τους συγκεκριμένο αριθμό ως αποτέλεσμα μιας ακολουθίας bit. Και στις δύο περιπτώσεις τα σήματα μπορούν να μεταφερθούν στη μονάδα ελέγχου, όπου με τον κατάλληλο προγραμματισμό να επιτευχθεί ο βέλτιστος έλεγχος του σερβοκινητήρα. Πιο συγκεκριμένα, ο έλεγχος αυτός μπορεί να αφορά είτε στη θέση περιστροφής του άξονα ως προς τις 360 μοίρες που συνιστούν μια πλήρη περιστροφή, είτε στην ταχύτητα περιστροφής του άξονα, είτε στην αναπτυσσόμενη σε αυτόν ροπή. Αν εξετάσει κανείς τους σερβοκινητήρες από τη σκοπιά των ηλεκτροκινητήρων, θα διαπιστώσει ότι, από λειτουργική άποψη κύριο γνώρισμα τους είναι η ικανότητά τους να αναπτύσσουν μεγάλες επιταχύνσεις στην περίπτωση της πλήρους ακινησίας, όταν δηλαδή παρουσιάζεται μικρή ροπή αδράνειας και μεγάλη ροπή στρέψης.



Για την επίτευξη των προηγούμενων λειτουργιών πρέπει να τηρούνται οι ακόλουθες προϋποθέσεις:

1. Ο ρότορας να έχει μεγάλο μήκος και μικρή διάμετρο.
2. Να υπάρχουν περιελίξεις αντισταθμίσεως οι οποίες επιτρέποντας ανάπτυξη μεγαλύτερων ρευμάτων αυξάνουν τη ροπή στρέψης.
3. Για μικρής ισχύος κινητήρες προβλέπεται μόνιμος μαγνήτης μέσα στους πόλους του οποίου και γύρω από ένα μόνιμο στέλεχος (όπως στα όργανα κινητού πλαισίου) περιστρέφεται το τύλιγμα του ρότορα.
4. Να είναι μειωμένη η σταθερά χρόνου L.R του τυλίγματος του ρότορα.

Η επιλογή ενός σερβοκινητήρα γίνεται έχοντας υπόψη ότι η ισχύς του θα πρέπει να καλύπτει την ισχύ του φορτίου (ωφέλιμη) και τις τριβές (απώλειες) της διάταξης. Πέραν αυτού ο σερβοκινητήρας πρέπει να λειτουργεί στις επιθυμητές ταχύτητες και συνάμα να παρέχει την απαραίτητη επιτάχυνση στο ρότορα και στο φορτίο. Οι σερβοκινητήρες διακρίνονται σε ηλεκτρικούς AC και DC, σε πνευματικούς και υδραυλικούς.

Οι μονοφασικοί σερβοκινητήρες συνεχούς ρεύματος διακρίνονται στους εξής τέσσερις τύπους:

1. Ο πρώτος είναι αυτός που τα τυλίγματα του στάτορα τροφοδοτούνται από πηγή σταθερής τάσεως ή ρεύματος, ενώ το τύλιγμα του ρότορα από μια τάση ελέγχου. Οι σερβοκινητήρες αυτοί είναι γνωστοί σαν ελεγχόμενοι από το ρότορα. Σε αυτούς τους σερβοκινητήρες αν διατηρούμε σταθερή την τάση ελέγχου  $V_e$  η ροπή στρέψης μικραίνει γραμμικά σε συνάρτηση με την αύξηση της γωνιακής ταχύτητας  $\omega$  του κινητήρα.
2. Ο δεύτερος τύπος σερβοκινητήρα είναι ο ελεγχόμενος από το στάτορα. Σ' αυτόν τον τύπο το τύλιγμα του ρότορα τροφοδοτείται από μια πηγή σταθερής τάσεως ή ρεύματος ενώ το τύλιγμα του στάτορα από μια τάση ελέγχου. Σε αυτούς τους σερβοκινητήρες η ροπή στρέψης είναι ανεξάρτητη από τη γωνιακή ταχύτητα του στάτορα και εξαρτάται μόνο από τη σταθερά  $K$  και το ρεύμα του στάτορα. Ωστόσο αν το μαγνητικό υλικό εργάζεται στον κόρο η ροπή στρέψης επηρεάζεται και από τη γωνιακή ταχύτητα του στάτορα και μάλιστα σε πολύ μεγάλες γωνιακές ταχύτητες η ροπή μικραίνει γιατί αυξάνει πάρα πολύ η ανηλεκτρεγερτική δύναμη.
3. Ο τρίτος τύπος είναι ο σερβοκινητήρας με τα τυλίγματα στάτορα και ρότορα σε σύνδεση σειράς: Οι σερβοκινητήρες αυτοί έχουν διπλό τύλιγμα στο στάτορα έτσι που το καθένα να συνδέεται σε σειρά με το τύλιγμα του ρότορα

με τη βοήθεια ηλεκτρονόμων. Η ροπή στρέψης του κινητήρα μεταβάλλεται εκθετικά και εξαρτάται από τα μεγέθη του ρεύματος ελέγχου και της γωνιακής ταχύτητας. Είναι πολύ μεγάλη κατά την εκκίνηση οπότε η γωνιακή ταχύτητα είναι μικρή, ενώ μικραίνει απότομα όταν η γωνιακή ταχύτητα μεγαλώνει, χρησιμοποιείται κυρίως εκεί όπου απαιτείται μεγάλη ροπή κατά την εκκίνηση (όπου έχουμε περιστροφή μαζών) αφού η γραμμικότητα δεν παίζει κανένα ρόλο.

4. Ένας ιδιαίτερα σημαντικός τύπος σερβοκινητήρας είναι αυτός με μόνιμο μαγνήτη. Ο σερβοκινητήρας του τύπου αυτού έχει αντί για τυλίγματα στάτορα, μόνιμο μαγνήτη, ενώ ο ρότορας έχει κανονικό τύλιγμα μέσα από το οποίο ελέγχεται ο κινητήρας. Ο κινητήρας αυτός μοιάζει πολύ με τους ασύγχρονους κινητήρες παράλληλης διέγερσης και λόγω του μικρού όγκου του χρησιμοποιείται σε Συστήματα Αυτομάτου Ελέγχου πάνω σε αεροπλάνα. Τέλος, ο μικρός όγκος του κινητήρα επιτυγχάνεται μέσω ειδικού κράματος μόνιμου μαγνήτη.

Οι μαγνήτες που χρησιμοποιούνται στους σερβοκινητήρες κατηγοριοποιούνται στους παρακάτω βασικούς τύπους:

- Οι κεραμικοί μαγνήτες οι οποίοι αποτελούνται από οξειδίο του σιδήρου και καρβίδιο του βαρίου ή του στροντίου. Οι κεραμικοί μαγνήτες χρησιμοποιούνται σε κινητήρες μικρής σχετικά ισχύος για μη ενεργοβόρες διαδικασίες ελέγχου.
- Οι μαγνήτες AlNiCo, δηλαδή οι μαγνήτες αλουμινίου, νικελίου, κοβαλτίου, οι οποίοι είναι δυνατόν να περιέχουν ίχνη από σίδηρο, χαλκό και τιτάνιο. Σήμερα δε περιλαμβάνονται στο σχεδιασμό των καινούργιων κινητήρων εξαιτίας του υψηλού κόστους και της σχετικά εύκολης απομαγνήτισης τους σε συνθήκες ανοικτού κυκλώματος.
- Οι μαγνήτες Σαμαρίου Κοβαλτίου, οι οποίοι λόγω του υψηλού κόστους χρησιμοποιούνται μόνο σε εφαρμογές στις οποίες η υψηλή θερμοκρασία και η αντοχή σε διάβρωση αποτελούν κρίσιμες παραμέτρους.
- Οι μαγνήτες Νεοδύμιου Σιδήρου Βορίου (NdFeB) είναι οι πιο σύγχρονη γενιά μαγνητών. Οι εξαιρετικές μαγνητικές τους ιδιότητες τους καθιστούν παράλληλα κατάλληλους και για συμπαγείς κατασκευές σε χρήσεις που απαιτούν μικρότερα κόστη κατασκευής. Σημαντικό μειονέκτημα τους είναι ότι διαβρώνονται πολύ εύκολα.

Οι σερβοκινητήρες εναλλασσομένου ρεύματος μπορούν να είναι διφασικοί ή τριφασικοί. Οι διφασικοί αποτελούνται από δύο τυλίγματα στο στάτη με τέτοια

τοποθέτηση, ώστε να παρουσιάζουν διαφορά φάσεως 90 μοιρών και το ρότορα. Το ένα τύλιγμα ονομάζεται τύλιγμα αναφοράς και τροφοδοτείται από μια εναλλασσόμενη τάση σταθερής τιμής, ενώ το άλλο τύλιγμα είναι τύλιγμα ελέγχου και τροφοδοτείται από την τάση ελέγχου. Όταν λοιπόν εφαρμοστούν αυτές οι τάσεις στα τυλίγματα τότε δημιουργείται στρεφόμενο μαγνητικό πεδίο από τα δύο ρεύματα που διαρρέουν τα τυλίγματα και ο ρότορας περιστρέφεται.

Ο ρότορας είναι φτιαγμένος από χάλκινες ράβδους που βραχυκυκλώνουν μεταξύ τους (βραχυκυκλωμένος δρομέας). Η ταχύτητα και η διεύθυνση περιστροφής καθορίζονται από το πλάτος και τη φάση της τάσης ελέγχου. Οι κινητήρες εναλλασσόμενου ρεύματος παρουσιάζουν μεγάλη ροπή στρέψης για μικρές γωνιακές ταχύτητες. Επιπρόσθετα, η σχέση μεταξύ ροπής και γωνιακής ταχύτητας είναι όμοια με αυτήν των σερβοκινητήρων συνεχούς ρεύματος που ελέγχονται από το ρότορα, δηλαδή η ροπή στρέψης μικραίνει γραμμικά σε συνάρτηση με την αύξηση της γωνιακής ταχύτητας.

Η λειτουργία ανάδρασης σε έναν σερβοκινητήρα αποσκοπεί στο συνεχή έλεγχο των εντολών θέσης και ταχύτητας που δίνονται προς τον κινητήρα. Αυτό επιτυγχάνεται από τον ενισχυτή του σερβοσυστήματος που αποτελεί και το σύστημα οδήγησης του σερβοκινητήρα (servodrive). Ο «σερβοενισχυτής» αποτελεί τον ενδιάμεσο σταθμό μεταξύ μονάδας ελέγχου και σερβοκινητήρα. Στην πραγματικότητα το servodrive είναι ένα ειδικού τύπου inverter, το οποίο όμως χρησιμοποιείται αποκλειστικά για έλεγχο σερβοκινητήρων και γι' αυτό έχει σχεδιαστεί έτσι ώστε να επιτελεί μια πολύ ειδική λειτουργία. Η λειτουργία των ενισχυτών του σερβοκινητήρα (servodrives) πιο συγκεκριμένα, αποσκοπεί στο να διατηρεί σταθερές τις απαιτούμενες στροφές, να διατηρεί σταθερή τη ροπή σε όλη την περιοχή στροφών του κινητήρα, αλλά ταυτόχρονα να δίνει τη δυνατότητα της βηματικής κίνησης με απόλυτο έλεγχο των δύο προηγούμενων παραμέτρων. Αναλυτικότερα, τα συστήματα οδήγησης σερβοκινητήρων (servodrives) επιτελούν τρεις βασικές λειτουργίες:

1. Τον έλεγχο της ανάδρασης του «σερβοσυστήματος».
2. Τον έλεγχο του κινητήρα.
3. Τη μετατροπή ισχύος.

Ο έλεγχος ενός «σερβοσυστήματος» συνίσταται στη ρύθμιση της ταχύτητας και της θέσης ενός κινητήρα. Η ρύθμιση αυτή βασίζεται σε ένα σήμα ανάδρασης. Το βασικότερο «σερβοκύκλωμα» είναι το κύκλωμα της ταχύτητας, του οποίου ο ρόλος είναι η παραγωγή μιας εντολοδότησης για τη ροπή με σκοπό την ελαχιστοποίηση του σφάλματος μεταξύ της αρχικής εντολοδότησης για την ταχύτητα και της ταχύτητας που λαμβάνεται από το σήμα ανάδρασης. Λόγω ότι στους σερβοκινητήρες κατά κανόνα απαιτείται να υπάρχει και έλεγχος θέσης συνήθως προστίθεται ένα κύκλωμα ελέγχου θέσης σε σειρά με το κύκλωμα ελέγχου ταχύτητας. Στα κυκλώματα ελέγχου

ενός σερβοσυστήματος σημαντική παράμετρος είναι και η ρύθμιση της έντασης του σήματος ανάδρασης. Αν αυτή η ρύθμιση γίνει σε ψηλή στάθμη τότε αφενός μεν τα αποτελέσματα είναι καλύτερα, αφετέρου το συνολικό σερβοσύστημα καθίσταται πιο ασταθές. Για το λόγο αυτό χρησιμοποιούνται φίλτρα που συνδέονται σε σειρά με τα κυκλώματα ελέγχου ταχύτητας. Ο έλεγχος του κινητήρα συνίσταται στην παραγωγή μιας ροπής κινητήρα που να ανταποκρίνεται στην εντολοδότηση ροπής που στέλνεται από το κύκλωμα ελέγχου του σερβοσυστήματος.

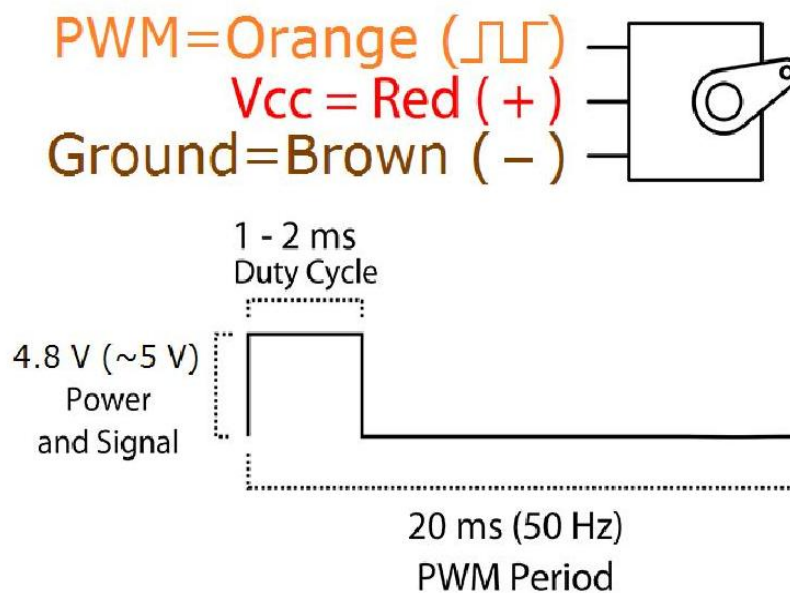
Στους κινητήρες που φέρουν ψήκτρες (κατά κύριο λόγο είναι οι κινητήρες συνεχούς ρεύματος) ο έλεγχος του κινητήρα συνίσταται απλά στον έλεγχο του ρεύματος των τυλιγμάτων καθώς η ροπή του κινητήρα είναι περίπου ανάλογη του ρεύματος των τυλιγμάτων. Τα κυκλώματα ελέγχου ρεύματος είναι όμοια με αυτά του ελέγχου ταχύτητας με τη μόνη διαφορά ότι ενεργούν σε ψηλότερες συχνότητες. Ένα κύκλωμα ελέγχου ρεύματος λαμβάνει την εντολοδότηση ρεύματος και τη συγκρίνει με την τιμή του ρεύματος που λαμβάνεται από το σήμα ανάδρασης παράγοντας μια έξοδο, δηλαδή ένα σήμα ρύθμισης τάσης. Αν ο κινητήρας χρειάζεται να δουλέψει με μεγαλύτερη ροπή τότε αυξάνεται η εφαρμοζόμενη σ' αυτόν τάση μέχρι να επιτευχθεί το επιθυμητό ρεύμα τυλιγμάτων. Τέλος, αναφορικά με τη μετατροπή της ταχύτητας υπάρχουν αλγόριθμοι που στηρίζονται στην ικανότητα της πηγής ισχύος να παράγει το ρεύμα εκείνο που θα ικανοποιεί τις απαιτήσεις που προκύπτουν από τα κυκλώματα ελέγχου ταχύτητας και θέσης.

Η σύνδεση του σερβοενισχυτή με το συνολικό σερβοσύστημα γίνεται συνήθως με μια σειρά θυρών επικοινωνίας που διαθέτει ο ειδικός αυτός inverter. Μια θύρα μπορεί να χρησιμοποιείται για τη σύνδεση του servodrive με την παλμογεννήτρια του σερβοσυστήματος που συνιστά το τοπικό αισθητήριο του στο ελεγχόμενο σημείο της παραγωγικής ή κατασκευαστικής διαδικασίας. Μια άλλη θύρα μπορεί να συνδέεται με Η/Υ, μέσω του οποίου μπορεί να προγραμματίζεται ο σερβοενισχυτής, έτσι ώστε με τον καθορισμό συγκεκριμένων παραμέτρων να επιτυγχάνεται ο βέλτιστος έλεγχος του κινητήρα. Σε περίπτωση που το ο κεντρικός έλεγχος του σερβοσυστήματος υποστηρίζεται από PLC ή DCS, τότε ο σερβοενισχυτής μπορεί να διαθέτει κι άλλες θύρες για την εισαγωγή σημάτων από το PLC ή το DCS. Υπάρχει και η περίπτωση οι σχεδιαστικές επιλογές να απαιτούν τη μεταφορά τοπικών σημάτων μέσω άλλων αισθητηρίων, τερματικών ή ακόμα και μπουτόν στο servodrive χωρίς αυτά προηγουμένως να περνάνε από το PLC ή το DCS. Και σε αυτήν την περίπτωση χρειάζεται ο σερβοενισχυτής να είναι εφοδιασμένος με κατάλληλο αριθμό θυρών.

Η δυνατότητα που προσφέρουν οι σερβοκινητήρες στα συστήματα ελέγχου κίνησης με τον παράλληλο έλεγχο ταχύτητας και θέσης με πολύ μεγάλη ακρίβεια και χωρίς περιορισμούς μηχανικής ισχύος (δηλαδή αναγκαίας ροπής) τους καθιστά κατάλληλους για μια τεράστια γκάμα βιομηχανικών εφαρμογών. Το μεγαλύτερο μέρος από αυτές αναφέρεται σε εξελιγμένη αυτοματοποίηση κατασκευαστικών διαδικασιών και σε μεταφορά και συσκευασία υλικών και προϊόντων. Αναλυτικότερα οι εφαρμογές που συναντούν οι σερβοκινητήρες στη βιομηχανία είναι οι ακόλουθες:

- Σε ρομποτικά συστήματα όλων των ειδών. Τα ρομποτικά συστήματα έχουν την δυνατότητα να υποστηρίζουν μεταξύ άλλων, εξελιγμένες εργαλειομηχανές κατεργασιών υλικών και μηχανές συγκόλλησης και βαφής μετάλλων, αλλά και συστήματα συναρμολόγησης σε κατασκευαστικές βιομηχανίες. Στην κατεργασία των υλικών κατέχουν εξέχοντα ρόλο στην αναβάθμιση των παραδοσιακών εργαλειομηχανών σε εργαλειομηχανές CNC. Χρησιμοποιούνται στις κοπές μετάλλων οποιασδήποτε μορφής. Βρίσκουν ιδιαίτερες εφαρμογές σε τριαξονικά συστήματα κοπής, σε σύγχρονες κοπές εν κινήσει πολλαπλών σταθμών και μπορούν να επιτύχουν κοπές εν κινήσει με συγχρονισμό μέχρι 12 αξόνων. Ιδιαίτερη εφαρμογή συναντούν στις μηχανές συρματοουργίας.
- Οι σερβοκινητήρες διευρύνουν την εφαρμογή τους στον κατασκευαστικό τομέα. Πιο συγκεκριμένα, στο βιομηχανικό τομέα χρησιμοποιούνται σε μηχανές που αναλαμβάνουν την επεξεργασία χαρτιού, ξύλου, μαρμάρου. Ιδιαίτερα σημαντικό ρόλο παίζουν στις μηχανές παραγωγής πλαστικών προϊόντων (extruders, μηχανές blow, κ.λ.π), ενώ ειδικές εφαρμογές βρίσκουν σε διάφορες άλλες βιομηχανίες, όπως στην καπνοβιομηχανία και στην τυπογραφία.
- Στον τομέα της μεταφοράς και συσκευασίας υλικών και προϊόντων οι σερβοκινητήρες βρίσκουν μαζική εφαρμογή. Ειδικότερα, χρησιμοποιούνται σε εγκιβωτιστικά συστήματα, σε παλετοποιητικά συστήματα διαφόρων προϊόντων και σε συστήματα pick and place. Ακόμα χρησιμοποιούνται σε μηχανές συσκευασίας, σε καρτονέττες και σε ετικέτες. Τέλος, χρησιμοποιούνται σε γεμιστικά μηχανήματα χύδην, στερεών και υγρών προσόντων.

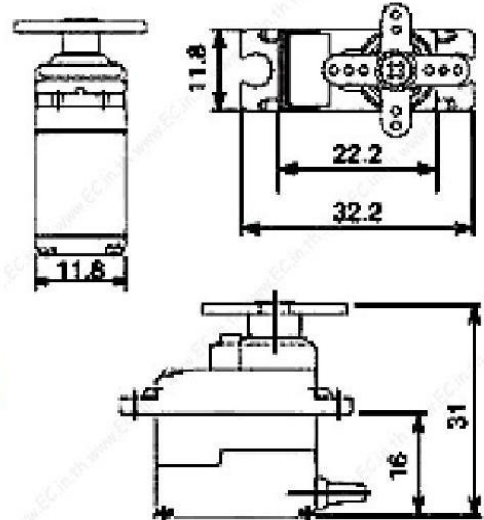
Τα παραπάνω δεδομένα καθίστανται οικεία σε όσους ασχολούνται αποκλειστικά με τους σερβοκινητήρες (επαγγελματίες μηχανικούς), ενώ για την παρούσα εργασία τα πράγματα είναι πιο εύκολα και απλά. Δηλαδή, ο συγκεκριμένος σερβοκινητήρας που αναλύεται είναι μικρός και ελαφρύς αλλά με μεγάλη ισχύ εξόδου, μπορεί να περιστρέφει περίπου 180 μοίρες (90 σε κάθε κατεύθυνση) και λειτουργεί όπως ακριβώς και τα κανονικά σέρβο απλά είναι σε μικρότερες διαστάσεις (για τις ανάγκες προσομοιώσεις). Ο μελετώμενος σερβοκινητήρας απευθύνεται σε αρχάριους χρήστες που θέλουν να έχουν ένα σερβομοτέρ με συνεχή κίνηση χωρίς να χρειάζεται να ασχοληθούν με την οικοδόμηση ενός ελεγκτή κινητήρα με ανάδραση και κιβώτιο ταχυτήτων. Επίσης, λειτουργεί με παλμό έτσι ώστε να ελεγχτεί η θέση του βραχίονα του, και ο κατασκευαστής ενημερώνει ακριβώς τον χρήστη, ερχόμενος ο σερβοκινητήρας με την παρακάτω οδηγία:



Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, "-90" (~1 ms pulse) is all the way to the left.

Τα χαρακτηριστικά του αναφέρονται στο δεύτερο φυλλάδιο το οποίο συνοδεύει το σερβομοτέρ, που είναι το κάτωθι:

## SG90 9 g Micro Servo



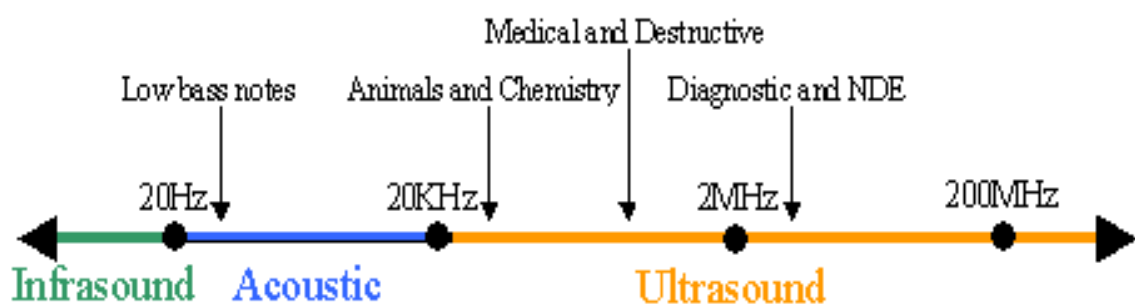
Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but *smaller*. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

### Specifications

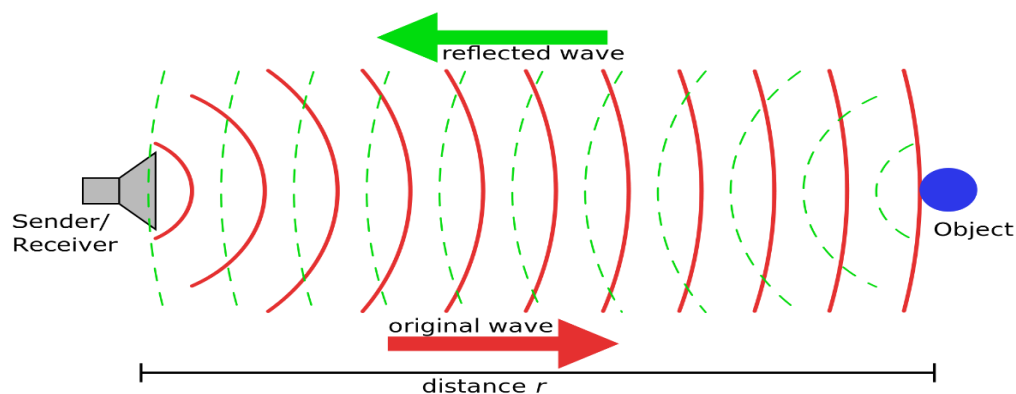
- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10  $\mu$ s
- Temperature range: 0 °C – 55 °C

### 3.3. Αισθητήρας υπερήχων

Υπέρηχος (ultrasound) ονομάζεται το μηχανικό κύμα με συχνότητα μεγαλύτερη από αυτήν που μπορεί να ακούσει ο άνθρωπος (Το ανθρώπινο αυτί δε μπορεί να ακούσει ήχους από πηγή που κάνει περισσότερες από 20.000 ταλαντώσεις σε ένα δευτερόλεπτο περίπου 20.000 Hz). Δηλαδή ο υπέρηχος είναι, ένας ήχος τόσο υψηλός που δεν μπορεί να γίνει αντιληπτός με την ακοή, καθώς το ανθρώπινο αυτί έχει όρια και δεν μπορεί να ακούσει πολύ ψηλούς ή χαμηλούς ήχους.



Όπως φαίνεται και στην παραπάνω εικόνα, οι υπέρηχοι βρίσκονται πάνω απ' τις ακουστικές συχνότητες, έτσι δεν μπορεί να τους ακούσει ο άνθρωπος, για την ακρίβεια το ανθρώπινο αυτί. Παρόλο που το ανθρώπινο αυτί δεν τους ακούει, υπάρχουν κάποια ζώα που μπορούν να τους ακούν αλλά και να τους χρησιμοποιούν. Χαρακτηριστικά παραδείγματα που αποδεικνύουν την προηγούμενη διατύπωση είναι η κίνηση των νυχτερίδων, η σφυρίχτρα που χρησιμοποιείται για τους σκύλους και οι συνομιλίες μεταξύ δελφινιών. Αισθητήρες υπερήχων συναντάμε σε πολλές εφαρμογές στην ιατρική, στην πλοήγηση σκαφών/πλοίων ακόμα και στα αυτοκίνητα, τα λεγόμενα park sensors. 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 ηχητικού σήματος ή ενός ραδιοκύματος πάνω στο στόχο.





Η αρχή λειτουργίας των αισθητήρων υπερήχων είναι ότι δημιουργούν υψηλής συχνότητας ηχητικά κύματα και χρησιμοποιώντας το επιστρεφόμενο ηχητικό σήμα καθορίζουν την απόσταση ή 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. Στο Arduino τώρα, ο HC-SR04 είναι ένας αισθητήρας απόστασης που διαθέτει έναν πομπό και ένα δέκτη υπερήχων όπως και τέσσερις ακροδέκτες και λειτουργεί με υπερήχους. Οι δύο ακραίοι ακροδέκτες χρησιμοποιούνται για την τροφοδοσία, VCC που συνδέεται στο pin 5Volt, και η γείωση, GND που συνδέεται στο pin GND. Και οι δύο κεντρικοί ακροδέκτες, ο ένας είναι η ενεργοποίηση του αισθητήρα (Trig) και ο άλλος η έξοδος (Echo) που συνδέονται σε ψηφιακά pin. Η διαδικασία είναι απλή και για να ξεκινήσει χρειάζεται ένα παλμό στο pin ενεργοποίησης, τότε ο πομπός στέλνει μία ακολουθία υπερήχων που ταξιδεύουν στον χώρο και μόλις βρεθεί κάποιο αντικείμενο το ανακλά και επιστρέφει προς τον αισθητήρα, μόλις λάβει την ανάκλαση τότε στο pin εξόδου αποστέλλεται ένας παλμός HIGH που η διάρκεια αυτού ταυτίζεται με το χρόνο που μεσολαβεί από την αποστολή των υπερήχων μέχρι τη λήψη της ανάκλασής τους.

Η ανάρτηση της εξόδου πραγματοποιείται με τη συνάρτηση pulseIn() που ορίζεται από το pin της λήψης και το επίπεδο του παλμού HIGH ή LOW. Η συνάρτηση από το pin της λήψης παίρνει έναν παλμό με το επιθυμητό επίπεδο HIGH ή LOW και στη συνέχεια επιστρέφει μία ακέραια τιμή τύπου LONG που δείχνει τη διάρκεια του παλμού σε msec στην έξοδο του αισθητήρα. Η διάρκεια αυτή, είναι ο χρόνος που μεσολαβεί από την αποστολή της ακολουθίας υπερήχων μέχρι τη λήψη της ανάκλασής τους. Με άλλα λόγια, είναι ο χρόνος που χρειάζονται οι υπέρηχοι για να διανύσουν δύο φορές την απόσταση μεταξύ του αισθητήρα και του κοντινότερου αντικειμένου. Οπότε, αν duration είναι η τιμή που επιστρέφει ο αισθητήρας και distance η ζητούμενη απόσταση, με δεδομένα από την φυσική έχουμε δεδομένη την ταχύτητα του ήχου  $340\text{m/s} = 0,034\text{cm}/\mu\text{s}$  και τον τύπο της ταχύτητας  $u=s/t$  άρα:

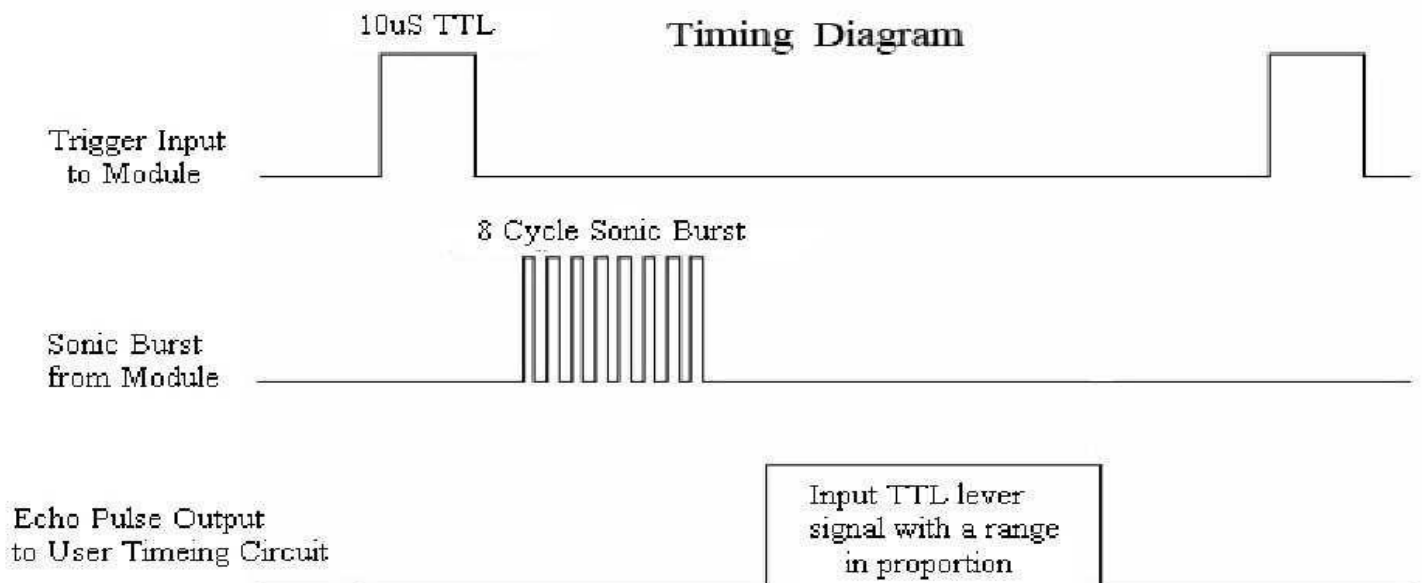
$$0,034 = \frac{\text{distance} * 2}{\text{duration}} \Leftrightarrow \text{distance} = \frac{0,034 * \text{duration}}{2}$$

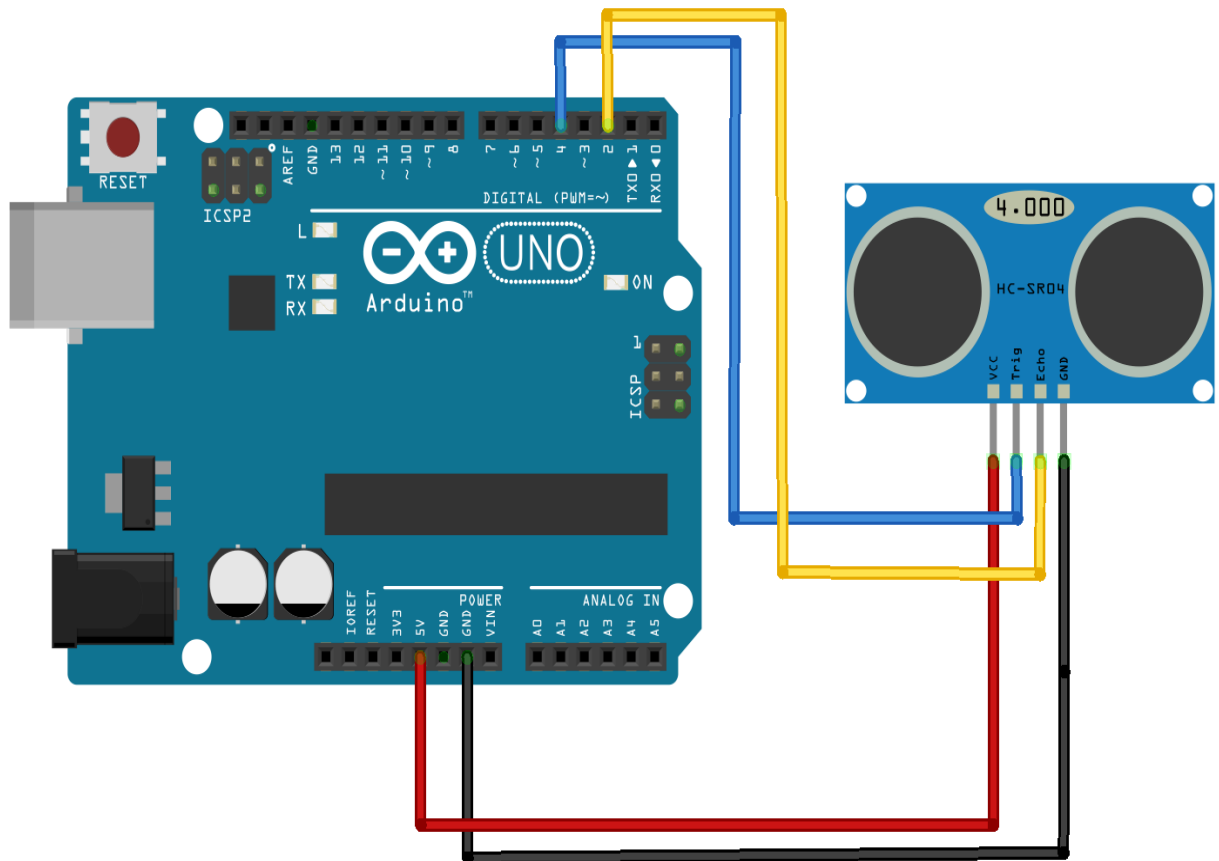
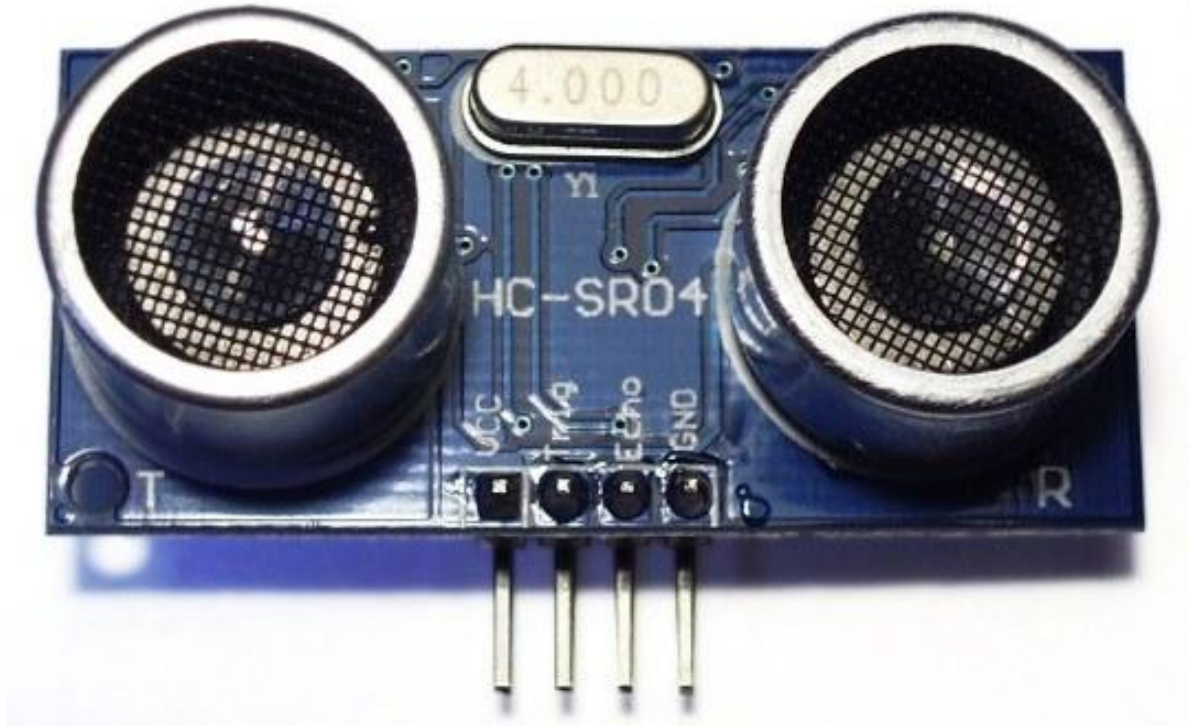
Ο συγκεκριμένος αισθητήρας υπερήχων παρέχει ένα εύρος μέτρησης από 2cm ως 400cm με ακρίβεια που αγγίζει τα 3mm.

Τα απαιτούμενα κριτήρια για την ορθή λειτουργία του είναι:

1. Χρησιμοποίηση παλμού ο οποίος να έχει τουλάχιστον 10μs διάρκεια.
2. Το εξάρτημα αυτόματα στέλνει 8 παλμούς συχνότητας 40kHz και ανιχνεύει αν υπάρχει παλμός που επέστρεψε.
3. Η απόσταση λαμβάνεται υπόψιν σύμφωνα με το χρόνο του σήματος στη αποστολή και λήψη του σήματος. Εξαρτάται βεβαίως από την ταχύτητα που ήχου που ανέρχεται στα 340 m/s.

Η τάση λειτουργίας του εξαρτήματος είναι 5V και παρέχεται από τον μικροελεγκτή. Το ρεύμα λειτουργίας είναι 15mA και η συχνότητα λειτουργίας είναι 40Hz. Η γωνία μέτρησης που επιτυγχάνει είναι οι 15 μοίρες. Οι διαστάσεις του εξαρτήματος είναι (45\*20\*15mm).





Made with  Fritzing.org

## 4. Ανοιχτός Κώδικας

---

### 4.1. ΕΕΛ/ΛΑΚ

Η Εταιρία Ελεύθερου Λογισμικού / Λογισμικού Ανοιχτού Κώδικα (ΕΕΛ/ΛΑΚ) είναι μια αστική μη κερδοσκοπική εταιρία που ιδρύθηκε την 28 Φεβρουαρίου 2008 στην Ελλάδα από 26 Πανεπιστήμια και Ερευνητικά Κέντρα, με αντικείμενο το ελεύθερο λογισμικό καθώς και το λογισμικό ανοιχτού κώδικα στο χώρο της εκπαίδευσης, του δημόσιου τομέα και των επιχειρήσεων στην Ελλάδα.

### 4.2. Λογισμικό Ανοιχτού Κώδικα – Ελεύθερο Λογισμικό

Στον χώρο της πληροφορικής και των ηλεκτρονικών υπολογιστών, με τον όρο λογισμικό ανοικτού κώδικα (Open Source Software) εννοείται το λογισμικό του οποίου ο πηγαίος κώδικας ανατίθεται σε τρίτον πρόσωπο για να τον ερευνήσει. Κατά καιρούς έχουν εμφανιστεί αρκετές διαφορετικές άδειες χρήσης σχεδιασμένες να συνοδεύουν λογισμικό ανοικτού κώδικα. Ορισμένες από αυτές επιτρέπουν στους χρήστες να τροποποιήσουν τον κώδικα ή να τον αξιοποιήσουν σε άλλες εφαρμογές.

Το Ελεύθερο Λογισμικό, όπως ορίζεται στον Ορισμό του Ελεύθερου Λογισμικού (Free Software Foundation), καθίσταται το λογισμικό που μπορεί να χρησιμοποιηθεί, αντιγραφεί, μελετηθεί, τροποποιηθεί και αναδιανεμηθεί χωρίς περιορισμό. Το ελεύθερο λογισμικό αποτελεί ένα εναλλακτικό μοντέλο ανάπτυξης και χρήσης λογισμικού που βασίζεται στην ελεύθερη διάθεση του πηγαίου κώδικα, και παρέχει τη δυνατότητα αλλαγών ή βελτιώσεων, ώστε να καλύπτονται όλες οι πιθανές ανάγκες του χρήστη.

Η ελευθερία από τέτοιους περιορισμούς αποτελεί βασικό προνόμιο στην ιδέα του Ελεύθερου Λογισμικού και αντιτίθεται στα λογισμικά εκείνα που περιορίζουν τον χρήστη σε βασικές λειτουργίες όπως ο κρυφός πηγαίος κώδικας, η περιορισμένη λειτουργία, η απαγόρευση κάποιας χρήσης του προγράμματος όπως της επαγγελματικής, η απαγόρευση μετάδοσης σε τρίτους, και όχι το εμπορικό λογισμικό το οποίο μπορεί να πωληθεί και να αναδιανεμηθεί με σκοπό το κέρδος. Ο όρος Ελεύθερο Λογισμικό δεν αναφέρεται στην τιμή της διανομής του λογισμικού, την οποία μάλιστα επιτρέπεται να χρεώνει ο κάθε διανομέας, εάν το επιθυμεί, αλλά

στα δικαιώματα του χρήστη κατά την χρήση του λογισμικού μετά την απόκτηση του. Ωστόσο, η συντριπτική πλειοψηφία των Ελεύθερων Λογισμικών διανέμεται δωρεάν.

Το Ελεύθερο Λογισμικό μπορεί να χαρακτηριστεί και ως ανοιχτό λογισμικό ή λογισμικό ανοιχτού κώδικα, χωρίς όμως να ταυτίζονται οι δυο αυτές έννοιες.

**Σύμφωνα με τον Ρίτσαρντ Στάλλμαν, ιδρυτή του Ιδρύματος Ελεύθερου Λογισμικού, δεν είναι κάθε λογισμικό ελεύθερο μόνο και μόνο επειδή είναι ανοιχτού κώδικα.**

Τα οφέλη χρήσης του Ελεύθερου λογισμικού / λογισμικού ανοιχτού κώδικα είναι:

- Απόλυτα νόμιμο λογισμικό το οποίο διατίθεται χωρίς κόστος και έχει πολύ μικρότερο κόστος συντήρησης
- Λογισμικό που ενσωματώνει άμεσα τις πιο σύγχρονες τεχνολογικές εξελίξεις
- Συχνές ενημερώσεις με νέες δυνατότητες
- Μεγάλη και προσιτή ευρύτητα ανάπτυξης και υποστήριξης του λογισμικού
- Πληθώρα πρόσθετων προγραμμάτων με εύκολη και δωρεάν εγκατάσταση
- Σταθερότητα και ασφάλεια
- Ασφαλέστερο και πιο αξιόπιστο λογισμικό σε σχέση με ιδιόκτητο λογισμικό που μπορεί να εντοπιστεί από το διαδίκτυο
- Δυνατότητα εξερεύνησης και εκμάθησης του τρόπου λειτουργίας του λογισμικού και προσαρμογής στις διαφορετικές ανάγκες των χρηστών

Το κόστος άδειας χρήσης των εφαρμογών ελεύθερου λογισμικού είναι τις περισσότερες φορές μηδενικό. Ειδικότερα, δεν πραγματοποιούνται αγορές αδειών χρήσεις και παρέχεται απεριόριστος αριθμός εγκαταστάσεων. Η χρήση ανοιχτού κώδικα δεν περιορίζει τον οργανισμό ή τον απλό χρήστη σε μια σχέση εξάρτησης από εταιρίες. Επιπλέον, λόγω ότι η διανομή, η διόρθωση σφαλμάτων και η ανάπτυξη του λογισμικού Ελεύθερου Λογισμικού / Λογισμικού Ανοιχτού Κώδικα μπορεί να γίνει από οποιαδήποτε καταρτισμένη ομάδα, δημιουργείται ένα περιβάλλον έντονου ανταγωνισμού, ο οποίος συμβάλλει στην διατήρηση ενός χαμηλού κόστους και υψηλών υπηρεσιών υποστήριξης. Ακόμη αξίζει να σημειωθεί ότι, το Ελεύθερο λογισμικό περιβάλλεται από μια μεγάλη κοινότητα χρηστών και προγραμματιστών, οι οποίοι αποσκοπούν μέσω της συνεργασίας στη συνεχή βελτίωση του λογισμικού, ανταλλάσσοντας γνώσεις και ιδέες. Σήμερα, λειτουργεί ένα παγκόσμιο ανοιχτό δίκτυο προγραμματιστών, οι οποίοι διορθώνουν και εξελίσσουν τον κώδικα των προγραμμάτων, παρουσιάζοντας ταχύτατα νέες βελτιωμένες εκδόσεις λογισμικού. Επομένως, συμβάλλουν καθημερινά στην δημιουργία τεχνολογικά εξελιγμένων κοινών αγαθών.

Η λογική της ανάπτυξης του λογισμικού ανοιχτού κώδικα είναι τέτοια έτσι ώστε να αποφεύγονται αρνητικές εκπλήξεις και σφάλματα μέσω της πολλαπλής δοκιμής. Ο κώδικας ερευνάται από μεγάλο πλήθος ανθρώπων, άρα τα πιθανά κενά ασφαλείας εντοπίζονται και διορθώνονται με μεγάλη ταχύτητα. Η υποστήριξη σε περίπτωση εμφάνισης προβλημάτων μπορεί να διορθωθεί άμεσα. Το Διαδίκτυο αποτελεί το βασικό τρόπο εντοπισμού και πρόσβασης στο διαθέσιμο Ελεύθερο Λογισμικό. Η σχέση του Διαδικτύου και του Ελεύθερου Λογισμικού / Λογισμικού Ανοιχτού Κώδικα υπήρξε εξ αρχής στενή και αμφίδρομη. Η εξάπλωση του Ελεύθερου Λογισμικού / Λογισμικού Ανοιχτού Κώδικα έχει στηριχθεί στην ευρεία χρήση του Διαδικτύου και η διαδικασία ανάπτυξης και λειτουργίας του Διαδικτύου βασίζεται, κατά κύριο λόγο, σε Ελεύθερο Λογισμικό / Λογισμικό Ανοιχτού Κώδικα. Ακόμη, και το Διαδίκτυο νέας γενιάς αναπτύσσεται με ανοιχτό λογισμικό. Το Λογισμικό Ανοιχτού Κώδικα, με συνεχείς βελτιώσεις και αξιόλογη φιλικότητα προς το χρήστη, κερδίζει διαρκώς ένηθρους υποστηρικτές παγκοσμίως. Αξίζει να αναφερθεί ότι, στην εκπαίδευση, στη δημόσια διοίκηση και στις επιχειρήσεις, οι ενδιαφερόμενοι ενημερώνονται και αποκτούν ιδιαίτερα ελκυστικά εργαλεία, αξιόπιστα, σταθερά στη λειτουργία, και απαλλαγμένα από το υψηλό κόστος απόκτησης και συνεχούς αναβάθμισης που απαιτούν τα κλειστά λογισμικά. Συνεπώς, αυξάνεται σημαντικά το ποσοστό των πόρων που διατίθενται στην τεχνική υποστήριξη με σημαντικά οφέλη για την τοπική και εθνική οικονομία.

### 4.3. Παράθεση Κώδικα Κατασκευής

#### 1. Κώδικας πλατφόρμας Arduino

```
// Δημιουργήθηκε από τον Λάγαρη Δημήτριο για την πτυχιακή εργασία του
```

```
// Βιβλιοθήκη για τις εντολές Servo
```

```
#include <Servo.h>
```

```
// Ορίζω τα pins για Trig and Echo του Ultrasonic Sensor
```

```
const int trigPin = 8;
```

```
const int echoPin = 7;
```

```
// Ορίζω μεταβλητές για την διάρκεια και την απόσταση
```

```
long duration;
```

```
int distance;
```

```
// Δημιουργώ ένα αντικείμενο servo για να ελέγγω τον σερβοκινητήρα
```

```
Servo myServo;
```

```
void setup() {
```

```
  pinMode(trigPin, OUTPUT); // Ορίζω το trigPin ως έξοδο
```

```
  pinMode(echoPin, INPUT); // Ορίζω το echoPin ως είσοδο
```

```
  Serial.begin(9600);
```

```
  myServo.attach(9); // Ορίζω το pin του σερβοκινητήρα
```

```
}
```

```
void loop() {
```

```
  // Περιστρέφω τον σερβοκινητήρα από 0 ως 180 μοίρες
```

```
  for(int i=0;i<=180;i++){
```

```
    myServo.write(i);
```

```
    delay(30);
```

```
distance = calculateDistance();// Καλώ μία συνάρτηση για τον υπολογισμό της απόστασης που μετράται από τον Ultrasonic sensor για κάθε μοίρα
```

```
Serial.print(i); // Στέλνω την τρέχουσα μοίρα στην Serial Port
```

```
Serial.print(","); // Στέλνω πρόσθετο χαρακτήρα που χρειάζεται για μετά για το Processing IDE
```

```
Serial.print(distance); // Στέλνω την τιμή της απόστασης στην Serial Port
```

```
Serial.print("."); // Στέλνω πρόσθετο χαρακτήρα που χρειάζεται για μετά για το Processing IDE
```

```
}
```

```
// Περιστρέφω τον σερβοκινητήρα από 180 ως 0 μοίρες
```

```
for(int i=180;i>0;i--){
```

```
myServo.write(i);
```

```
delay(30);
```

```
distance = calculateDistance();
```

```
Serial.print(i);
```

```
Serial.print(",");
```

```
Serial.print(distance);
```

```
Serial.print(".");
```

```
}
```

```
}
```

```
// Συνάρτηση για τον υπολογισμό της απόστασης που μετράται από τον Ultrasonic sensor
```

```
int calculateDistance(){
```

```
digitalWrite(trigPin, LOW);
```

```
delayMicroseconds(2);
```



```
// Ορίζω το trigPin σε HIGH κατάσταση για 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH); // Διαβάζω το echoPin, επιστρέφει τον χρόνο
ταξιδιού του ηχητικού κύματος σε microseconds

distance= duration*0.034/2;

return distance;
}
```

**Τέλος κώδικα πλατφόρμας Arduino.**

## 2. Κώδικας πλατφόρμας Processing

```
/* Γραφικό κομμάτι του Arduino Radar από τον Λάγαρη Δημήτριο */  
  
import processing.serial.*; // Εισάγω βιβλιοθήκη για την σειριακή επικοινωνία  
  
import java.awt.event.KeyEvent; // Εισάγω βιβλιοθήκη για την ανάγνωση των  
δεδομένων από την serial port  
  
import java.io.IOException;  
  
Serial myPort; // Ορίζω Object Serial  
  
// Καθορίζω μεταβλητές  
  
String angle="";  
  
String distance="";  
  
String data="";  
  
String noObject;  
  
float pixsDistance;  
  
int iAngle, iDistance;  
  
int index1=0;  
  
int index2=0;  
  
PFont orcFont;  
  
void setup() {  
  
  size (1280, 720); // Τοποθετώ την ανάλυση της οθόνης  
  
  smooth();  
  
  myPort = new Serial(this,"COM3", 9600); // Ξεκινώ την σειριακή επικοινωνία  
  
  myPort.bufferUntil('.'); // Διαβάζω τα δεδομένα από την serial port  
  
  orcFont = loadFont("AnonymousPro-25.vlw");  
  
}  
  
void draw() {
```

```
fill(98,245,31);

textFont(orcFont);

// Προσομοιώνω τη θαμπάδα κινήσεων και την αργή εξασθένιση της
κινούμενης γραμμής

noStroke();

fill(0,4);

rect(0, 0, width, height-height*0.065);

fill(98,245,31); // Πράσινο χρώμα

// Καλώ την συνάρτηση για την δημιουργία του radar

drawRadar();

drawLine();

drawObject();

drawText();

}

void serialEvent (Serial myPort) { // Ξεκινάω να διαβάζω τα δεδομένα από την
Serial Port

data = myPort.readStringUntil('.');

data = data.substring(0,data.length()-1);

index1 = data.indexOf(","); // Βρίσκω τον χαρακτήρα ',' και τοποθετώ σε αυτόν
την μεταβλητή "index1"

angle= data.substring(0, index1); // Διαβάζω τα δεδομένα από την θέση "0" ως
τη θέση της μεταβλητής index1 ή την τιμή της γωνίας που στέλνει το Arduino
μέσω της Serial Port

distance= data.substring(index1+1, data.length()); // Διαβάζω τα δεδομένα από
την θέση "index1" ως το τέλος ή την τιμή της απόστασης

// Μετατρέπω της αλφαριθμητικές (string) μεταβλητές σε ακέραιους (integer)

iAngle = int(angle);
```

```
iDistance = int(distance);  
  
}  
  
void drawRadar() {  
    pushMatrix();  
  
    translate(width/2,height-height*0.074); // Μετακινώ τις αρχικές συντεταγμένες  
    σε νέα θέση  
  
    noFill();  
  
    strokeWeight(2);  
  
    stroke(98,245,31);  
  
    // Σχεδιάζω τις καμπύλες  
  
    arc(0,0,(width-width*0.0625),(width-width*0.0625),PI,TWO_PI);  
    arc(0,0,(width-width*0.27),(width-width*0.27),PI,TWO_PI);  
    arc(0,0,(width-width*0.479),(width-width*0.479),PI,TWO_PI);  
    arc(0,0,(width-width*0.687),(width-width*0.687),PI,TWO_PI);  
  
    // Σχεδιάζω τις ευθείες με γωνία  
  
    line(-width/2,0,width/2,0);  
  
    line(0,0,(-width/2)*cos(radians(30)),(-width/2)*sin(radians(30)));  
    line(0,0,(-width/2)*cos(radians(60)),(-width/2)*sin(radians(60)));  
    line(0,0,(-width/2)*cos(radians(90)),(-width/2)*sin(radians(90)));  
    line(0,0,(-width/2)*cos(radians(120)),(-width/2)*sin(radians(120)));  
    line(0,0,(-width/2)*cos(radians(150)),(-width/2)*sin(radians(150)));  
  
    line((-width/2)*cos(radians(30)),0,width/2,0);  
  
    popMatrix();  
  
}  
  
void drawObject() {  
    pushMatrix();
```

```
    translate(width/2,height-height*0.074);// Μετακινώ τις αρχικές συντεταγμένες
    σε νέα θέση

    strokeWeight(9);

    stroke(255,10,10); // Κόκκινο χρώμα

    pixsDistance = iDistance*((height-height*0.1666)*0.025); // Καλύπτω την
    απόσταση από τον αισθητήρα, από cm σε pixels

    // Περιορίζω την εμβέλεια σε 40 cm

    if(iDistance<40){

        // Σχεδιάζω το αντικείμενο σύμφωνα με την γωνία και την απόσταση

        line(pixsDistance*cos(radians(iAngle)),-pixsDistance*sin(radians(iAngle)),(width-
        width*0.505)*cos(radians(iAngle)),-(width-width*0.505)*sin(radians(iAngle)));

    }

    popMatrix();

}

void drawLine() {

    pushMatrix();

    strokeWeight(9);

    stroke(30,250,60);

    translate(width/2,height-height*0.074);// Μετακινώ τις αρχικές συντεταγμένες
    σε νέα θέση

    line(0,0,(height-height*0.12)*cos(radians(iAngle)),-(height-
    height*0.12)*sin(radians(iAngle))); // Σχεδιάζω την γραμμή σύμφωνα με την
    γωνία

    popMatrix();

}

void drawText() { // Σχεδιάζω το κείμενο στην οθόνη

    pushMatrix();

    if(iDistance>40) {
```

```
noObject = "Out of Range";
}
else {
noObject = "In Range";
}
fill(0,0,0);
noStroke();
rect(0, height-height*0.0648, width, height);
fill(98,245,31);
textSize(25);

text("10cm",width-width*0.3854,height-height*0.0833);
text("20cm",width-width*0.281,height-height*0.0833);
text("30cm",width-width*0.177,height-height*0.0833);
text("40cm",width-width*0.0729,height-height*0.0833);
textSize(40);
text("Object: " + noObject, width-width*0.875, height-height*0.0277);
text("Angle: " + iAngle + " °", width-width*0.48, height-height*0.0277);
text("Distance: ", width-width*0.26, height-height*0.0277);
if(iDistance<40) {
text("    " + iDistance + " cm", width-width*0.225, height-height*0.0277);
}
textSize(25);
fill(98,245,60);

translate((width-width*0.4994)+width/2*cos(radians(30)),(height-
height*0.0907)-width/2*sin(radians(30)));
```

```
rotate(-radians(-60));  
text("30°",0,0);  
  
resetMatrix();  
  
translate((width-width*0.503)+width/2*cos(radians(60)),(height-  
height*0.0888)-width/2*sin(radians(60)));  
  
rotate(-radians(-30));  
  
text("60°",0,0);  
  
resetMatrix();  
  
translate((width-width*0.507)+width/2*cos(radians(90)),(height-  
height*0.0833)-width/2*sin(radians(90)));  
  
rotate(radians(0));  
  
text("90°",0,0);  
  
resetMatrix();  
  
translate(width-width*0.513+width/2*cos(radians(120)),(height-  
height*0.07129)-width/2*sin(radians(120)));  
  
rotate(radians(-30));  
  
text("120°",0,0);  
  
resetMatrix();  
  
translate((width-width*0.5104)+width/2*cos(radians(150)),(height-  
height*0.0574)-width/2*sin(radians(150)));  
  
rotate(radians(-60));  
  
text("150°",0,0);  
  
popMatrix();  
  
}
```

**Τέλος κώδικα πλατφόρμας Processing.**

Στους κώδικες, έχουν προστεθεί αρκετά σχόλια έτσι ώστε να καταλαβαίνει ο κάθε ενδιαφερόμενος την σκέψη και την γραφή αυτού του προγράμματος, για να μελετήσει, αντιγράψει ή και να χρησιμοποιήσει τον κώδικα, που συνδυάζει αρκετά ενδιαφέροντα πράγματα μαζί. Και στους δύο κώδικες ότι υπάρχει μετά από τις δύο πλάγιες γραμμές ( // ) είναι σχόλιο και δεν επηρεάζουν τον κώδικα, δηλαδή είναι σαν να μην υπάρχουν για το λογισμικό, απλά συμβάλλουν στην διευκόλυνση των αναγνωστών. Γι' αυτό και ο κώδικας αυτός ίσως περιέχει κάποιες παρανοήσεις.



## 5. Παρουσίαση και διαδικασία κατασκευής

---

### 5.1. Υλικά Που Χρησιμοποιήθηκαν

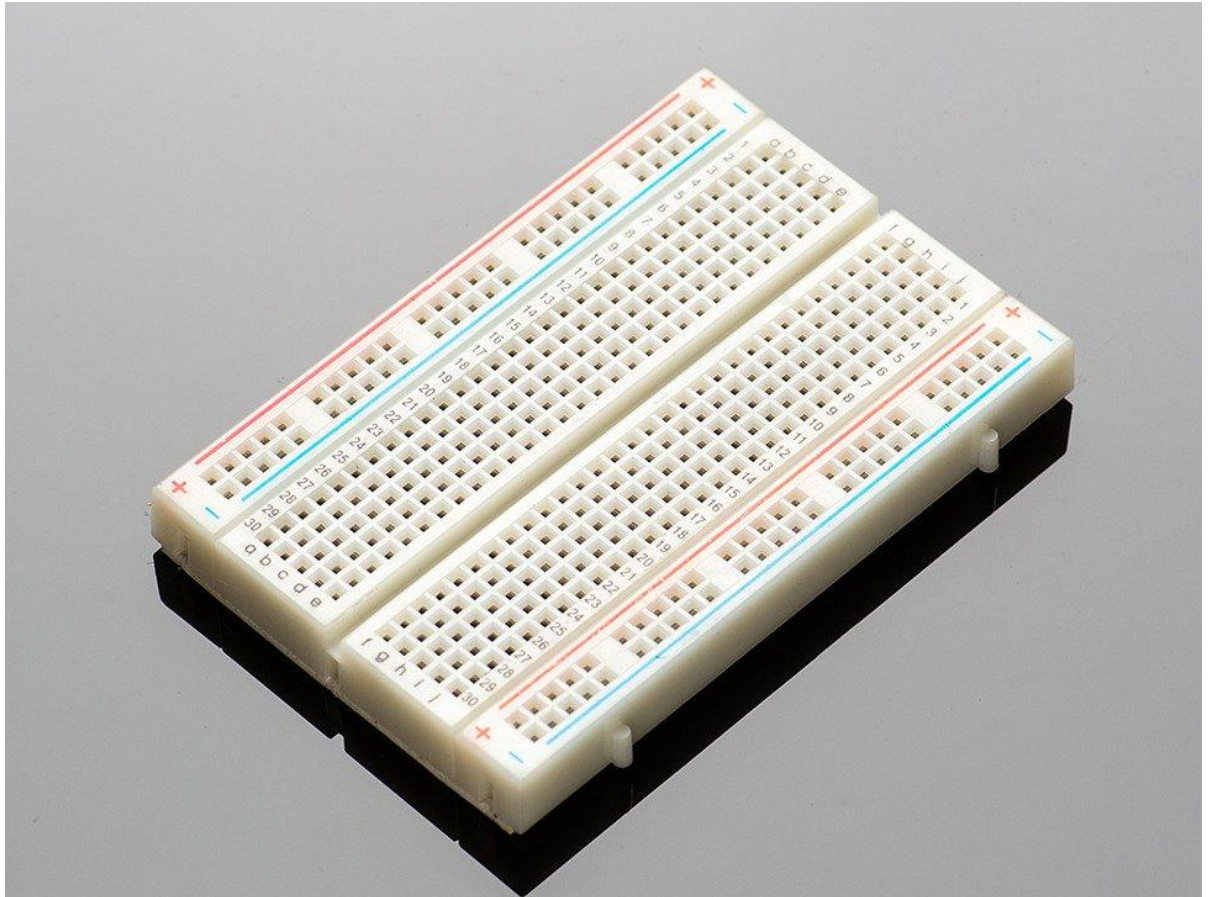
#### A. Καλώδια – Αγωγοί



Η παραπάνω εικόνα αποτυπώνει τα καλώδια, τα οποία συνδέουν τα διάφορα εξαρτήματα και μεταφέρουν τις πληροφορίες που χρειάζεται έτσι ώστε να επιτευχθεί το επιθυμητό αποτέλεσμα.

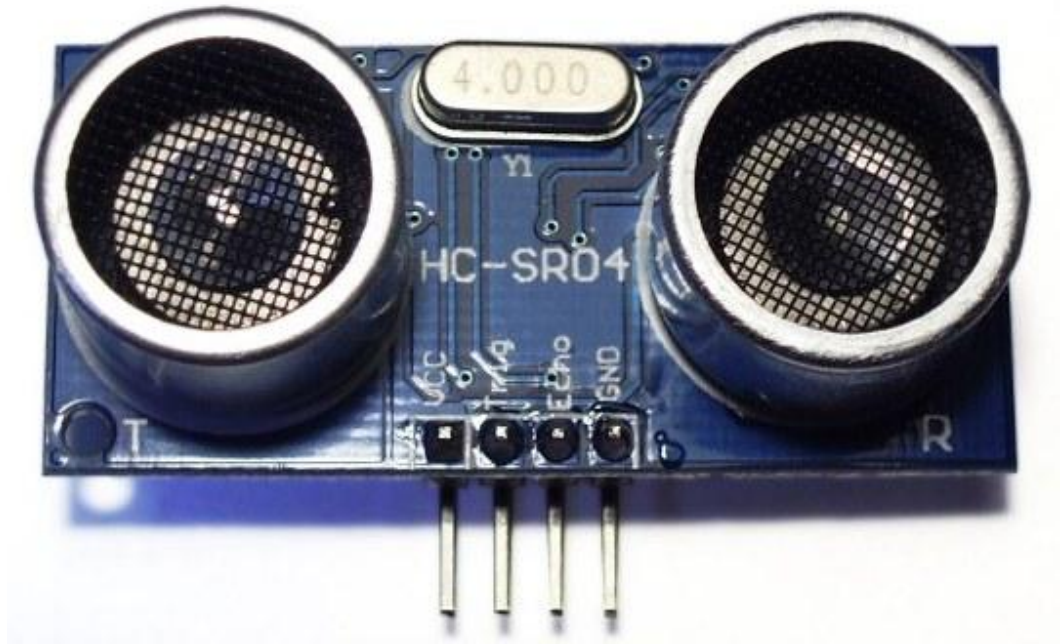
**B. Arduino UNO R3**

Στη συνέχεια, παρουσιάζεται πλατφόρμα του μικροελεγκτή μαζί με τον programmer που έχει αναλυθεί σε προγενέστερο στάδιο.

**C. Breadboard**

Σε αυτό το σημείο προβάλλεται το breadboard, δηλαδή η πλακέτα που διευκολύνει την δημιουργία κυκλωμάτων με ταχύ ρυθμό σε αξιόπιστο βαθμό χωρίς την χρήση κολλητηρίου.

**D. HC-SR04 + Bracket**



Φυσικά δεν θα ήταν δυνατό να παραλειφθεί ο αισθητήρας υπερήχων που χρησιμοποιήθηκε στην κατασκευή μαζί με το bracket του.

**Ε. Tower Pro Micro Servo SG90**

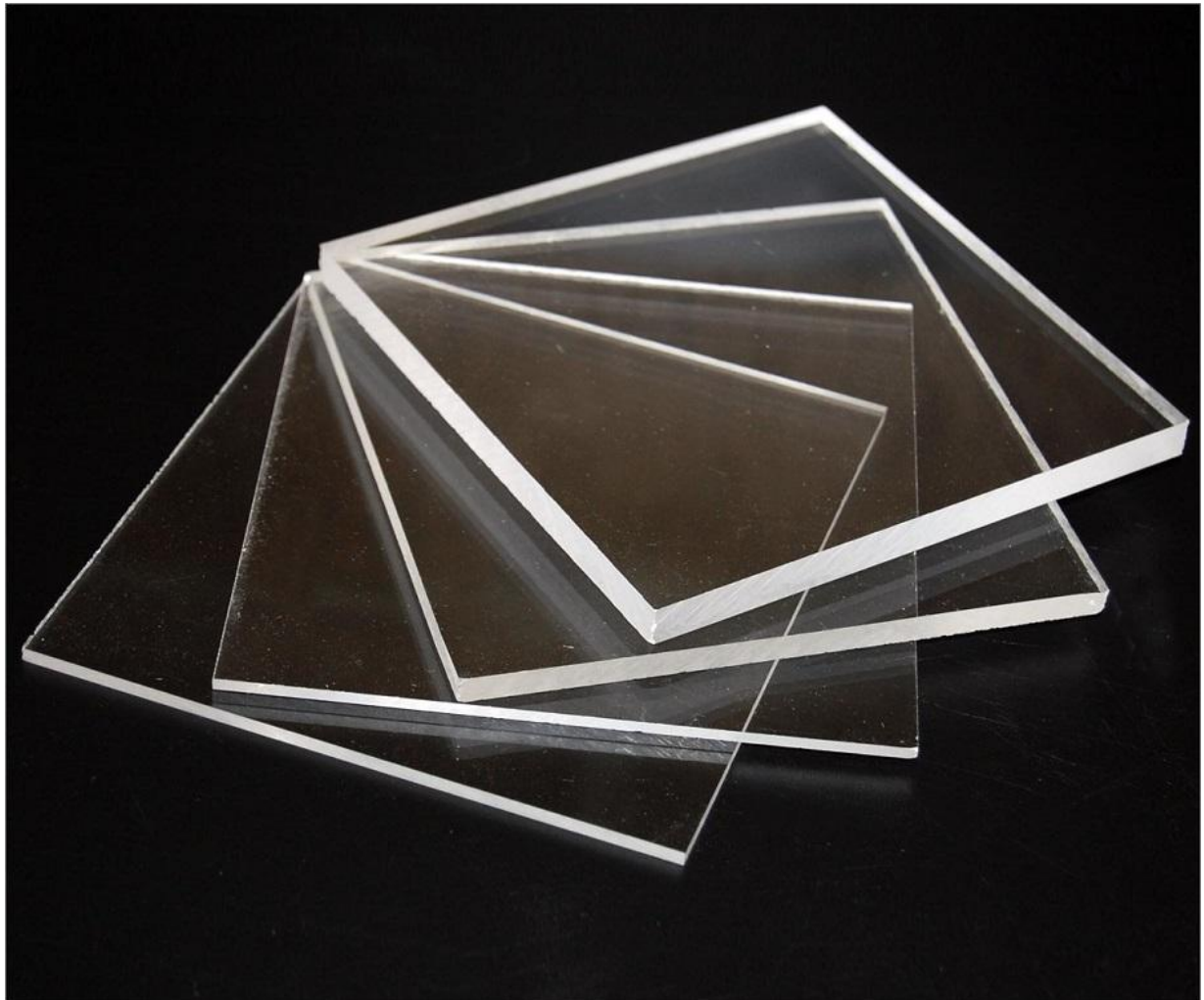


Σημαντικό ρόλο παίζει και ο σερβοκινητήρας που χρησιμοποιήθηκε στην κατασκευή.

**Ε. Ξύλινη κατασκευή (κουτί για την προστασία κατασκευής και χρηστών)**



Ένα από τα πιο σημαντικά τμήματα της κατασκευής καθίσταται το ξύλινο κουτί, το οποίο περιέχει μαζεμένα όλα τα εξαρτήματα της κατασκευής. Προσφέρει προστασία της κατασκευής αλλά και του χρήστη.

**G. Plexi-Glass**

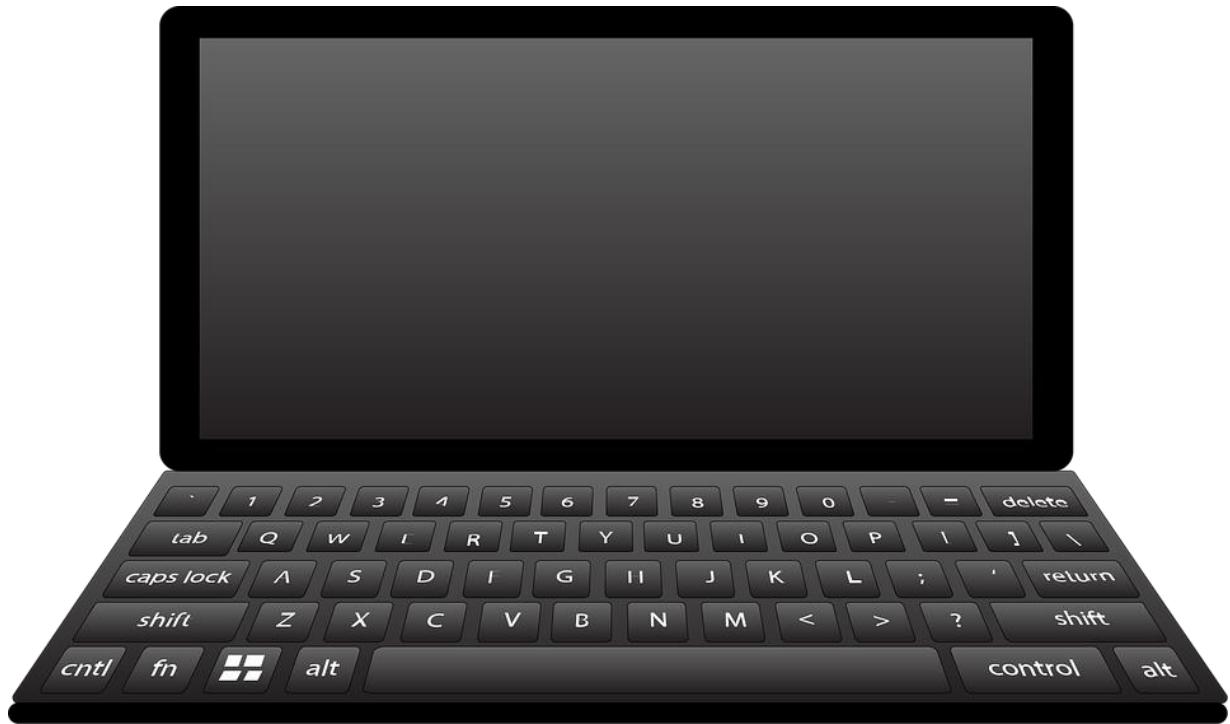
Το αισθητικό κομμάτι της κατασκευής στηρίζεται στο plexiglass, ένα διαφανές άκαμπτο θερμοπλαστικό γυαλί, που είναι τοποθετημένο στο πάνω μέρος, της ξύλινης κατασκευής. Το συγκεκριμένο υλικό τοποθετήθηκε λόγω της διαφάνειας που προσφέρει με στόχο την εύκολη κατανόηση των υλικών της κατασκευής και παράλληλα λόγω της ανθεκτικότητας του σε σχέση με το γυαλί που είναι επίσης διαφανές υλικό. Επίσης, το plexiglass είναι θερμομονωτικό και είναι δύσκολο να υποστεί φθορά όπως συμβαίνει με το γυαλί.

### Η. Οθόνη (για αναπαράσταση γραφικών)



Ακολούθως, παρουσιάζεται μια απλή οθόνη για με σκοπό την αναπαραγωγή του γραφικού τμήματος της κατασκευής.

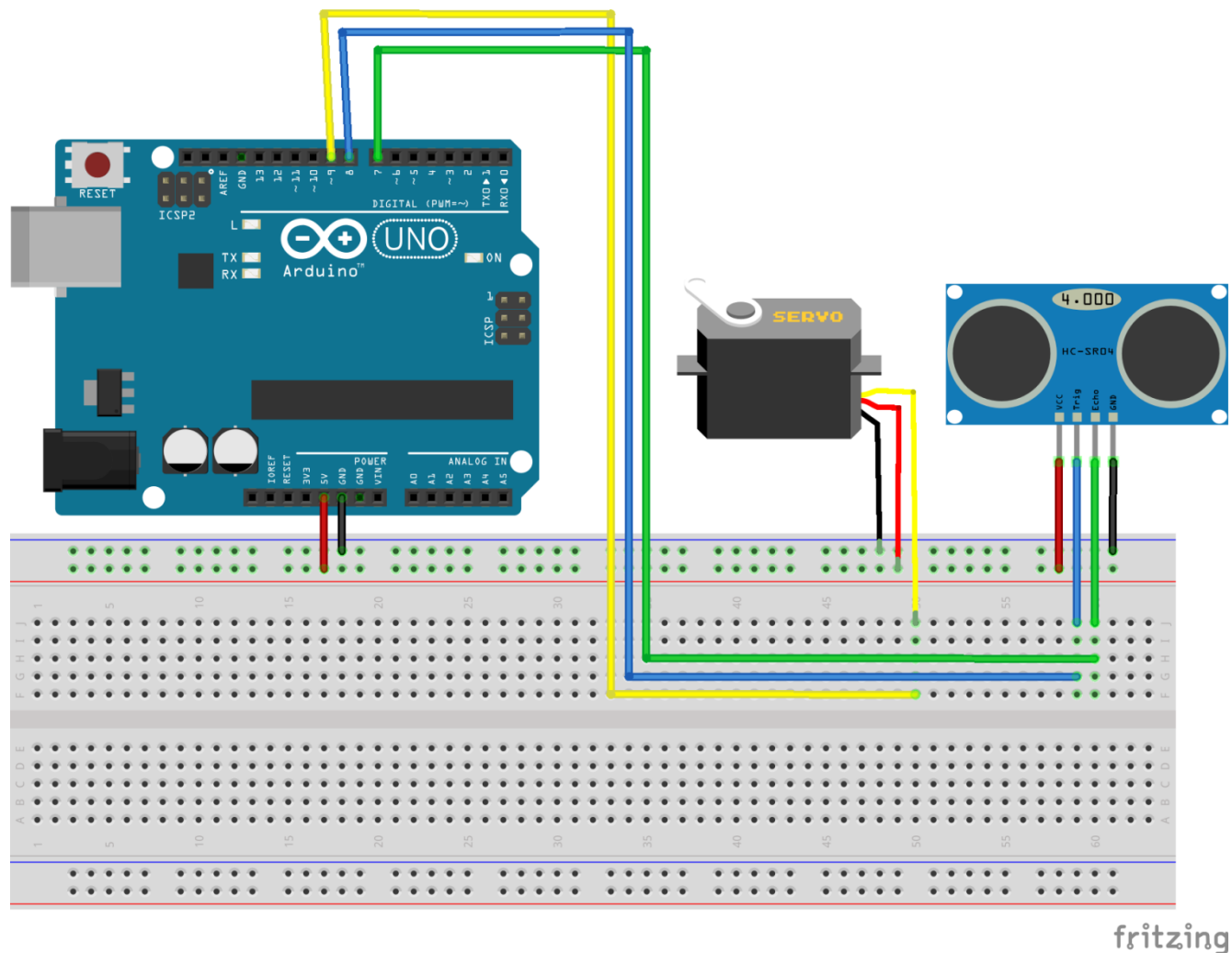


**I. Ηλεκτρονικός Υπολογιστής γραφείου (για την συγγραφή του κώδικα)**

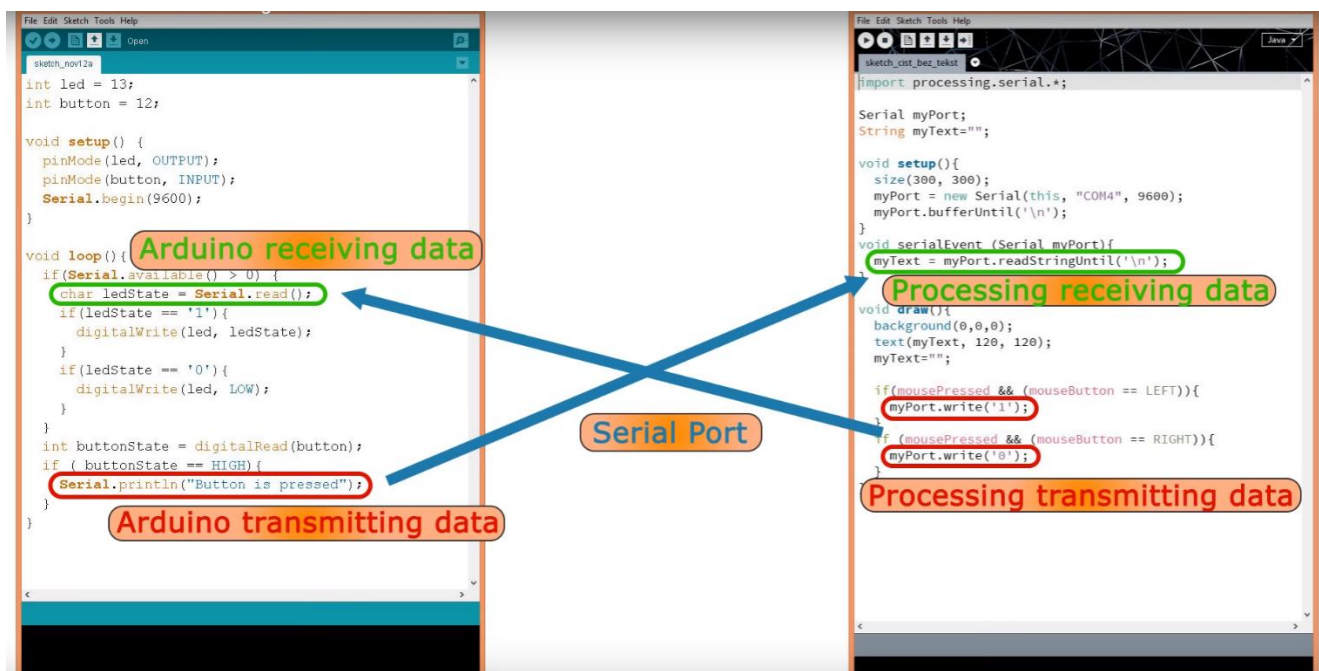
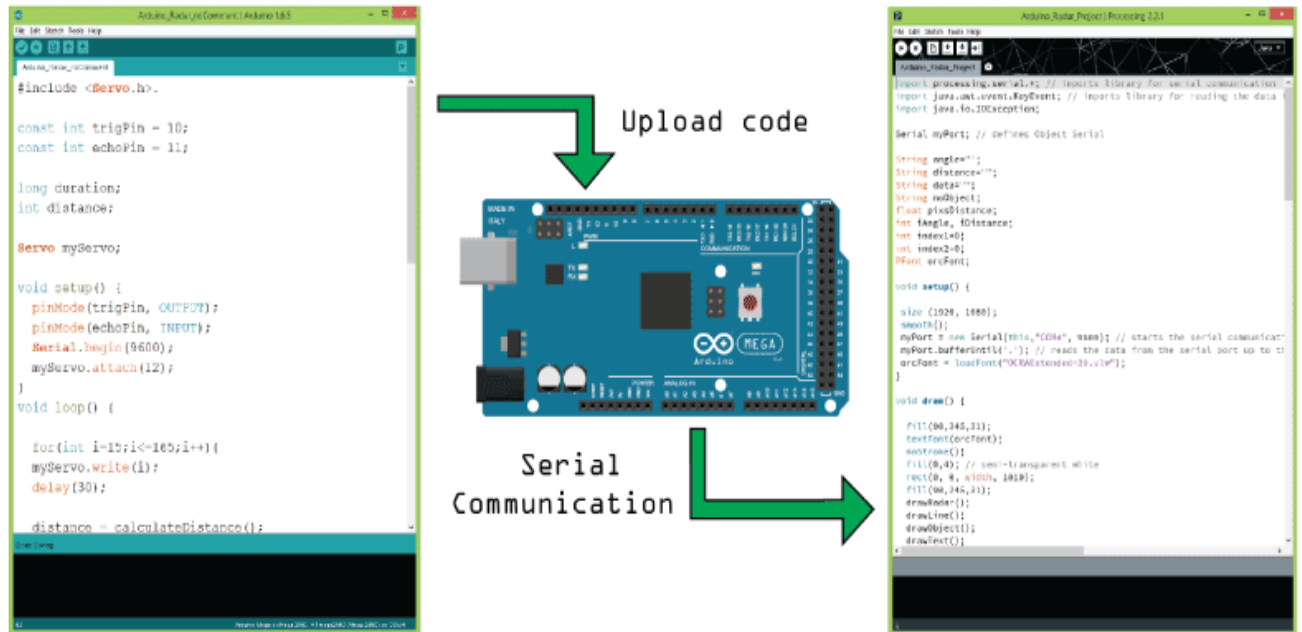
Ολοκληρώνοντας το κεφάλαιο των υλικών είναι σημαντικό να τονιστεί η χρησιμότητα του ηλεκτρονικού υπολογιστή με στόχο την λειτουργία του λογισμικού Arduino, το οποίο ήταν αναγκαίο για την συγγραφή του κώδικα της κατασκευής.

## 5.2. Συνδεσμολογία κατασκευής

Στην συγκεκριμένη ενότητα παρουσιάζεται ο τρόπος σύνδεσης των υλικών με την βοήθεια του breadboard. Τέλος, αξίζει να αναφερθεί ότι, το κύκλωμα σχεδιάστηκε στο πρόγραμμα **fritzing** το οποίο είναι και αυτό ανοιχτού κώδικα και μάλιστα δωρεάν και καλύπτει όλα τα λειτουργικά συστήματα, δηλαδή Windows, Mac Os και Linux.



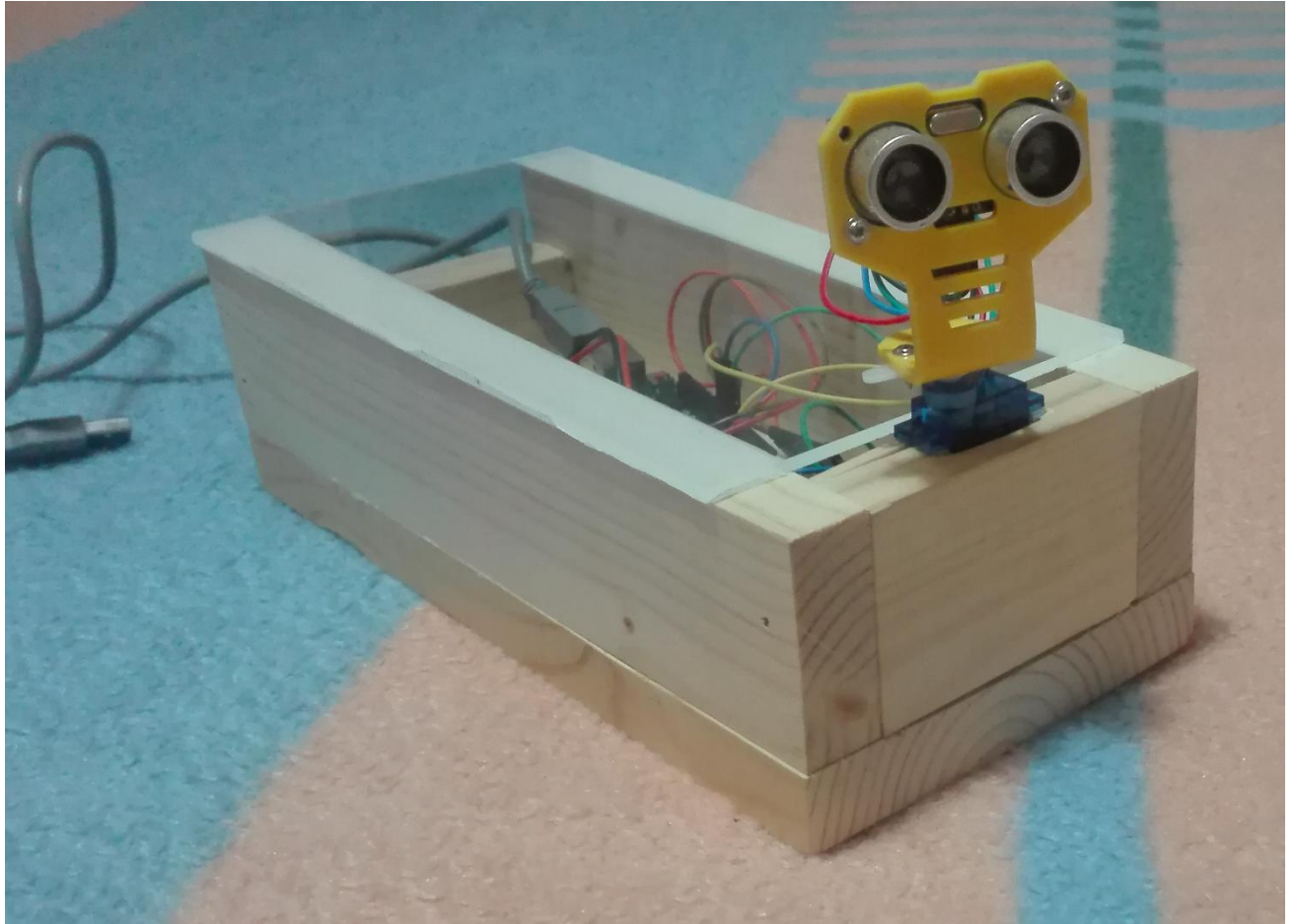
Παρακάτω παρουσιάζονται βοηθητικά γραφήματα που δείχνουν πως αλληλοεπιδρούν οι κώδικες μεταξύ τους, του Arduino και του Processing έτσι ώστε να επιτευχθεί το σωστό αποτέλεσμα του μικροελεγκτή.



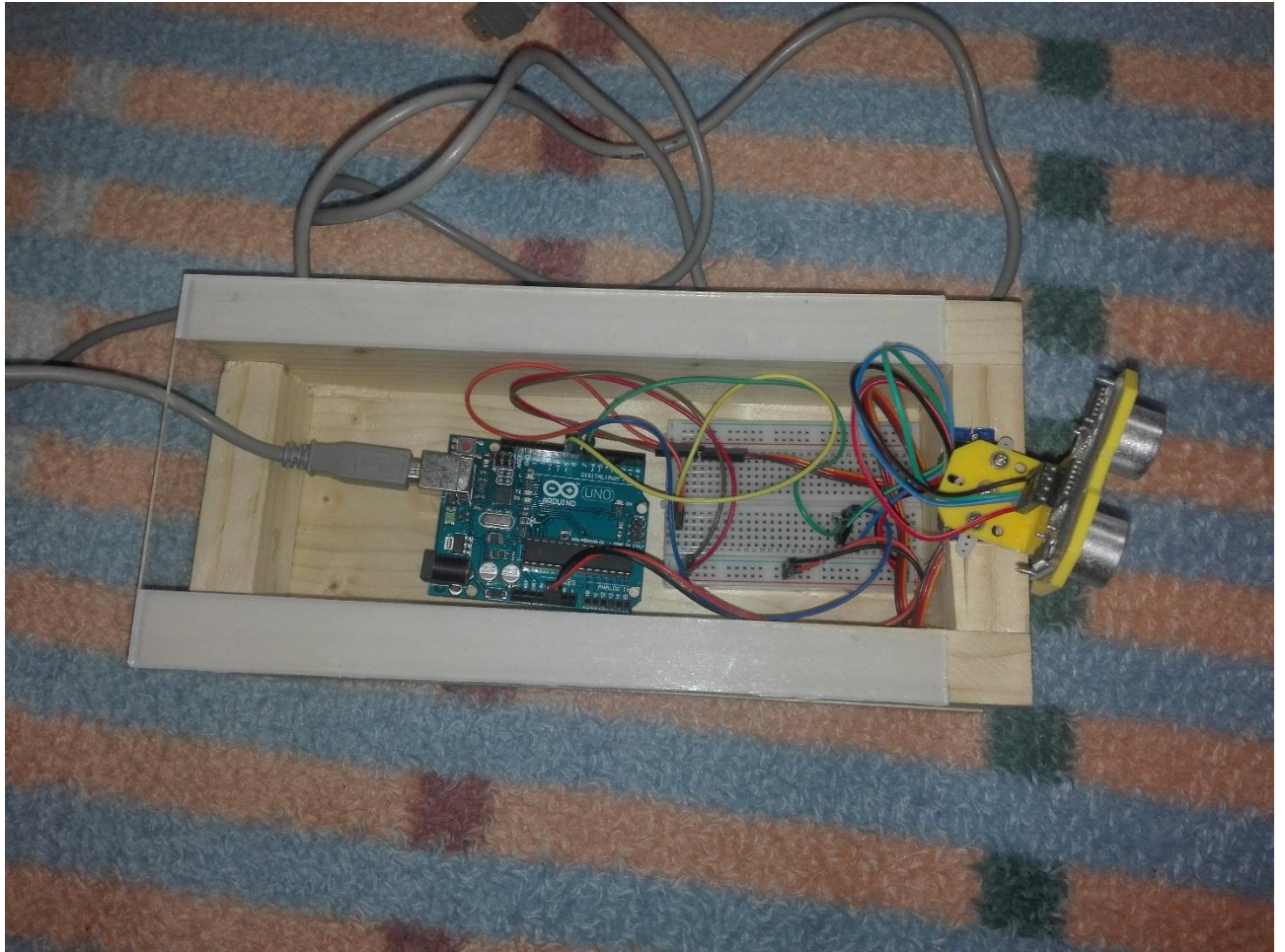
Σε αυτό το σημείο θεωρείται απαραίτητο να σημειωθεί ότι, υπάρχουν εντολές που μεταφέρουν και λαμβάνουν αντίστοιχα τα δεδομένα, μεταξύ των δύο «λογισμικών» που προσεγγίζονται.

### 5.3. Φωτογραφικό Υλικό

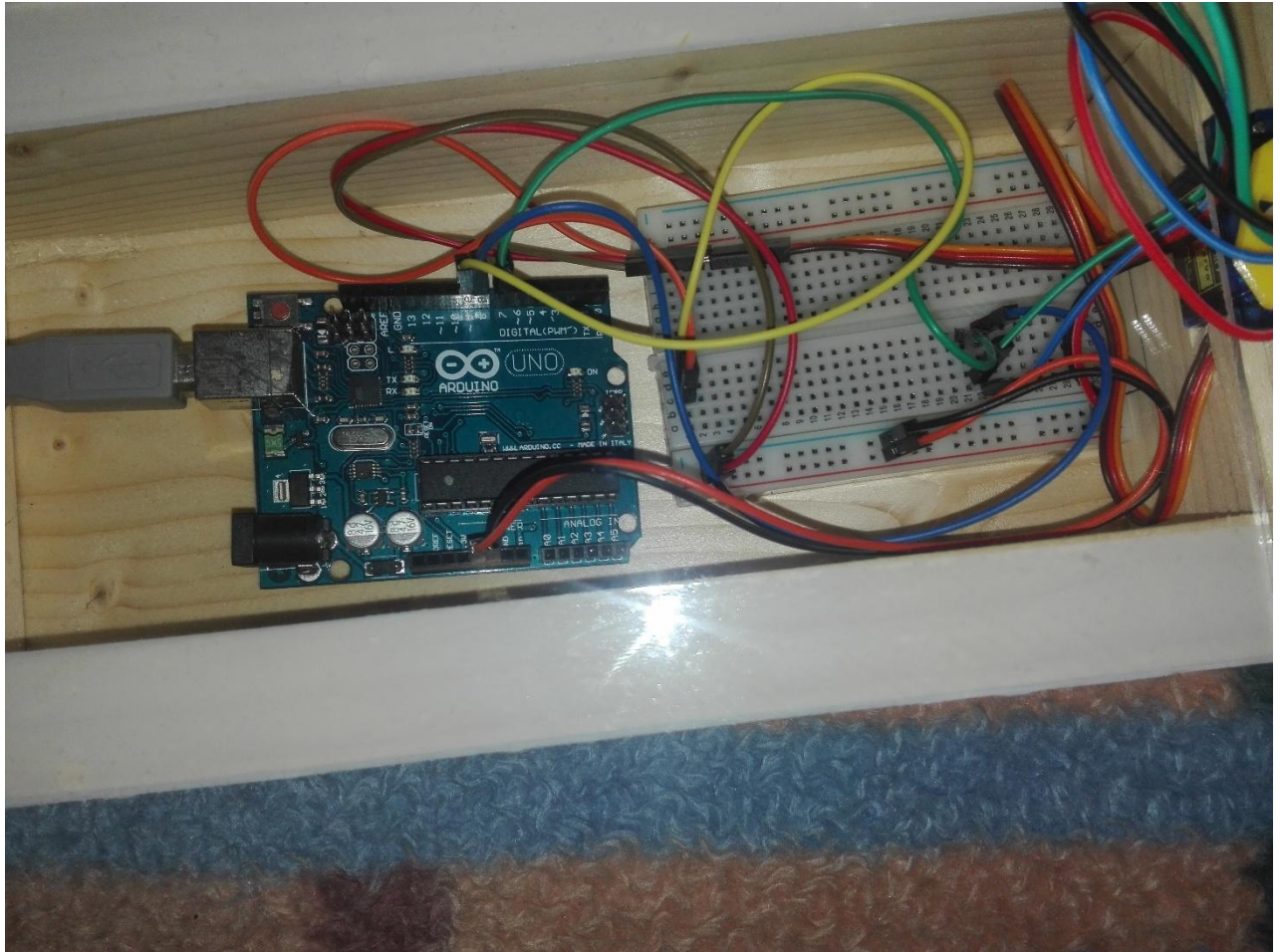
Παρακάτω υπάρχουν μερικά στιγμιότυπα από την κατασκευή ολοκληρωμένη και μερικά στιγμιότυπα από το γραφικό κομμάτι της κατασκευής.



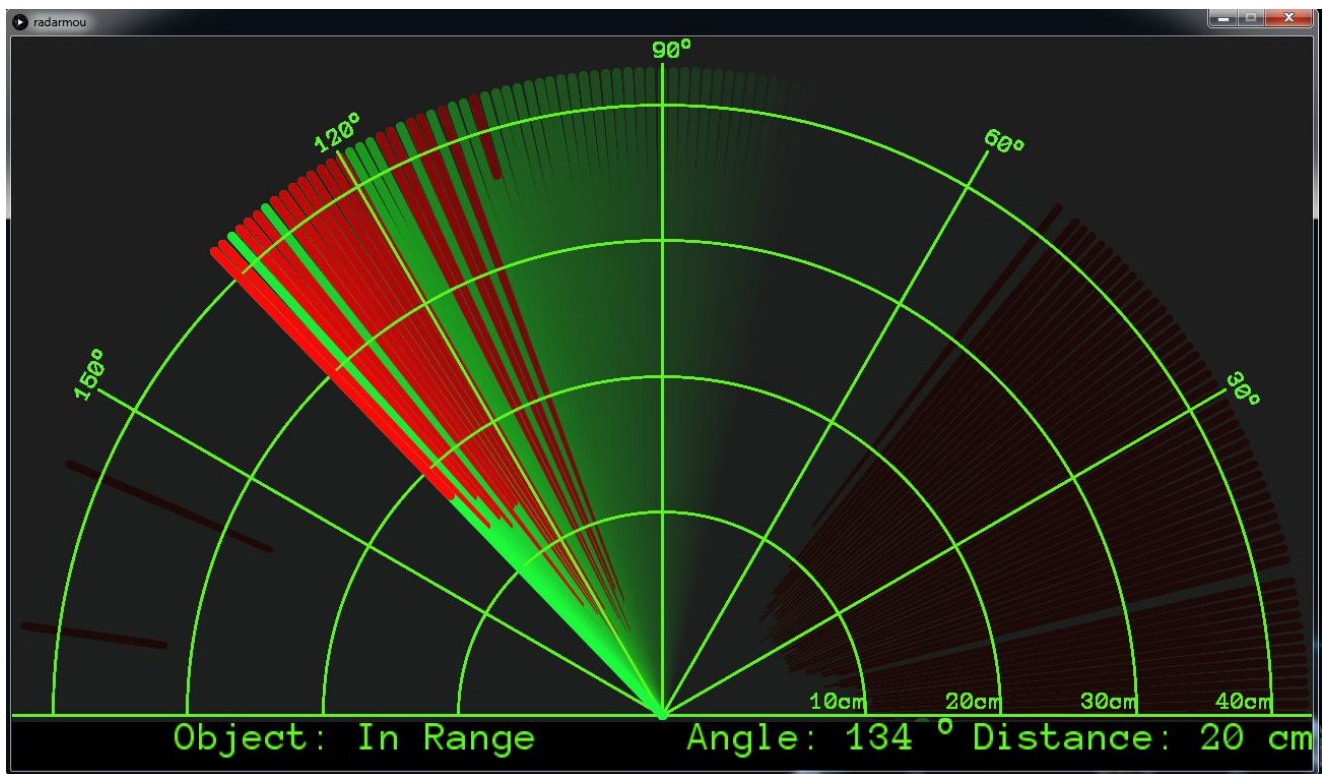
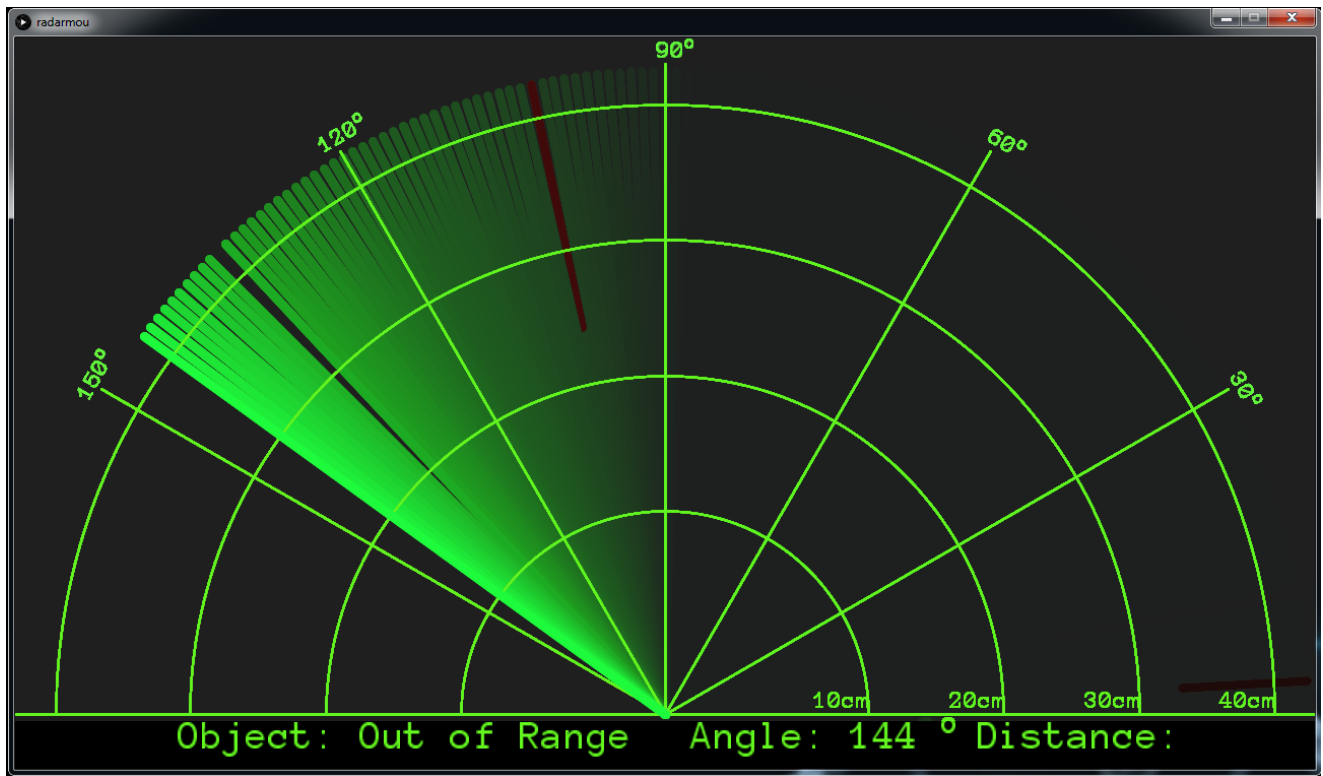
Παρουσιάζεται μια ολοκληρωμένη εικόνα της κατασκευής.

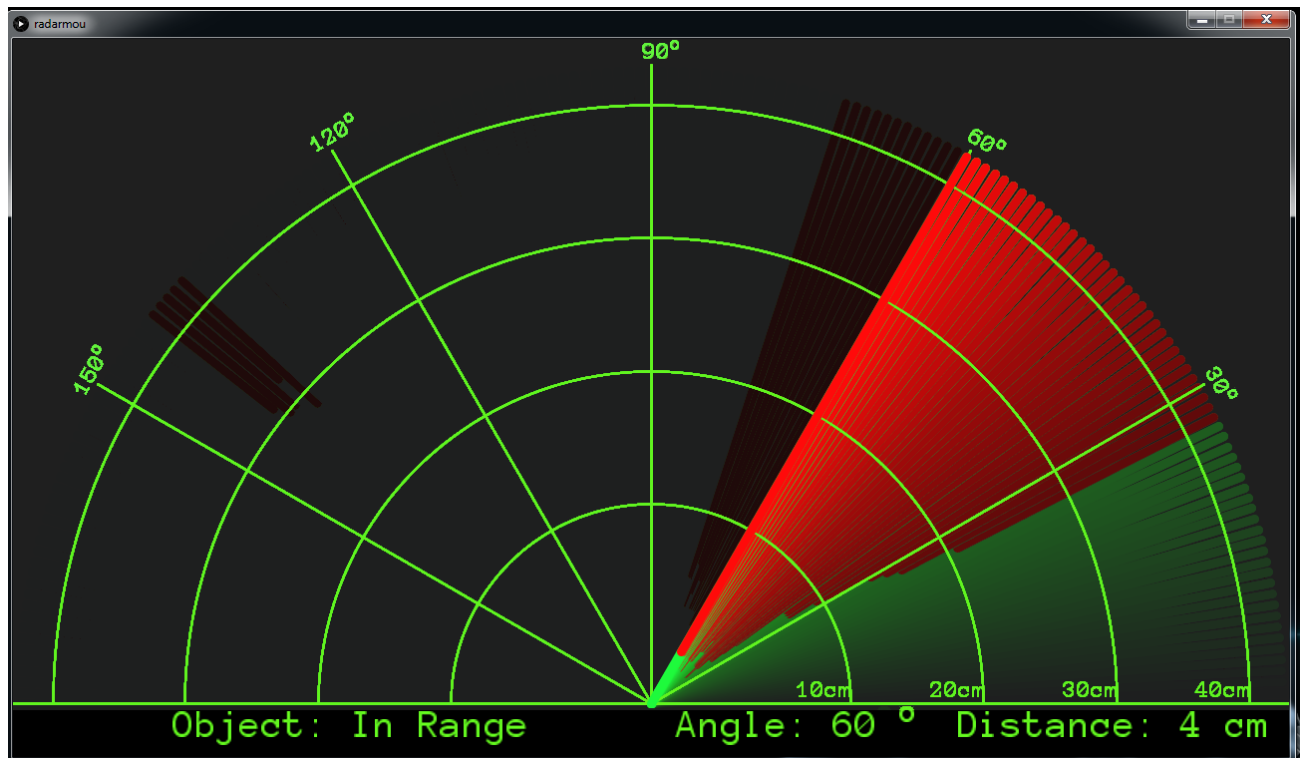


Σε αυτό το σημείο προβάλλεται το πάνω μέρος της κατασκευής, που μέσα από το plexiglass διακρίνεται η πλατφόρμα arduino υπο, το breadboard με την καλωδίωση και τα εξαρτήματα συνδεδεμένα μεταξύ τους.



Μία πιο κοντινη λήψη των ανωτέρω με εστίαση στην καλωδίωση και το πόσο σημαντικό και εύκολο είναι το breadboard.





Προβάλλονται τρία στιγμιότυπα από το γραφικό κομμάτι της κατασκευής, το πρώτο στιγμιότυπο δείχνει ότι δεν υπάρχει κάποιο αντικείμενο κοντά στην περιοχή που ελέγχει το radar, το δεύτερο δείχνει ότι το radar έχει εντοπίσει κάποιο αντικείμενο σε μακρινή απόσταση 20cm και τέλος το τρίτο στιγμιότυπο δείχνει ότι το radar έχει εντοπίσει κάποιο αντικείμενο σε κοντινή απόσταση και συγκεκριμένα 4cm.



## 6. ΕΠΙΛΟΓΟΣ

---

Με αφετηρία όλα τα προηγούμενα δεδομένα γίνεται κατανοητό ότι, η παρούσα κατασκευή αποτελεί αντικείμενο μίας πτυχιακής εργασίας συνεπώς δεν είναι δυνατό να συσχετιστεί με μία επαγγελματική κατασκευή εμπορίου. Επομένως, αναδεικνύονται οι δυνατότητες που μπορεί να έχουν τα ενσωματωμένα συστήματα που κρύβουν τεχνογνωσία. Βέβαια, η κατασκευή δεν μπορεί να θεωρηθεί αλάνθαστη διότι χρίζει περαιτέρω διερεύνησης σε αρκετά σημεία, γεγονός που συνδέεται με την σκοπιμότητα της πτυχιακής εργασίας που αφορά την πρόκληση ενός ερεθίσματος, έτσι ώστε να δουν με άλλο μάτι το Arduino οι νέοι αλλά και οι παλιοί μηχανικοί. Εξάλλου, ολόκληρη η πτυχιακή εργασία δεν είναι τίποτε άλλο παρά μονάχα μια μικρή εισαγωγή στα ενσωματωμένα συστήματα, τα οποία αποτελούν πλέον σημαντικό μέρος των εκπαιδευτικών προγραμμάτων που είναι προσανατολισμένα στην τεχνολογία. Η προσέγγισή τους γίνεται μέσω εκπαιδευτικών μικροελεγκτών, όπως το Arduino.

Στόχος της παρούσας πτυχιακής είναι η κατανόηση των ενσωματωμένων συστημάτων δηλαδή μικρά σε μέγεθος υπολογιστικά συστήματα, που ο άμεσος ρόλος τους δεν είναι η επεξεργασία δεδομένων. Εξυπηρετούν αυτοματισμούς και βρίσκονται πλέον σε κάθε τύπο ηλεκτρονικής συσκευής καθημερινής χρήσης, όπως κάμερες, οικιακές συσκευές, συσκευές γραφείου. Απαντώνται σε συστήματα μετρήσεων, σε παιχνίδια, σε βιομηχανικούς αυτοματισμούς, σε αυτοκίνητα και όλα τα μέσα μεταφοράς, σε ηλεκτρονικά όργανα και τηλεπικοινωνιακά συστήματα. Τα ενσωματωμένα συστήματα αποτελούν την ταχύτερα αναπτυσσόμενη βιομηχανία και αλλάζουν την καθημερινότητα, καθώς διεισδύουν παντού. Αποτελούν τη βάση της τεχνολογίας που αναφέρεται ως διαδίκτυο πραγμάτων (Internet of Things).

Συντελικά πρέπει να σημειωθεί ότι με την χρήση αυτού του κυκλώματος και με τον κατάλληλο προγραμματισμό, κάποιος που έχει γνώση μπορεί να δημιουργήσει και να προγραμματίσει τέτοιου είδους κυκλώματα δημιουργώντας πατέντες φθηνού κόστους που θα μπορούν να εκπληρώσουν πληθώρα αναγκών.

Η δημιουργία αυτής της εργασίας δίνει ένα έναυσμα για την απαρχή παρόμοιων κυκλωμάτων, τα οποία αναβαθμίζουν την τεχνολογία και είναι ευκολονόητη η λειτουργία τους όχι μόνο από γνώστες προγραμματισμού αλλά και από ανθρώπους οι οποίοι δεν είναι επαγγελματίες του χώρου.

## 7. ΒΙΒΛΙΟΓΡΑΦΙΑ - ΠΗΓΕΣ

---

### 7.1. Βιβλιογραφία

[1] **Ανάπτυξη Εφαρμογών με το Arduino** του Παναγιώτης Παπάζογλου και Σπύρος-Πολυχρόνης Λιώνης, *Εκδόσεις Τζιόλα*

[2] **30 Arduino Projects for the Evil Genius** by Simon Monk, *publishing by The McGraw-Hill*

[3] **Arduino Cookbook, 2<sup>nd</sup> Edition** by Michael Margolis, *publishing by O'Reilly Media*

[4] **Εισαγωγή στους μικροελεγκτές PIC** του Ιωάννης Καλόμοιρος από το ΤΕΙ Κεντρικής Μακεδονίας Σχολή Τεχνολογικών Εφαρμογών Τμήμα Μηχανικών Πληροφορικής Τ.Ε.

### 7.2. Ηλεκτρονικές Πηγές

[1] < [www.arduino.cc](http://www.arduino.cc) >

[2] < <http://wiring.org.co> >

[3] < [www.processing.org](http://www.processing.org) >

[4] < <http://fritzing.org> >

[5] < <https://deltahacker.gr> >

[6] < <http://www.metadosi-ischios.gr> >

[7] < <https://ellak.gr> >