



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS



## Project: Arduino Sonar Radar

**Γκίνη Βασιλική: up1100847**

**Μάζαρης Νικόλαος: up1100928**

**Μπανάκος Βασίλειος: up1100950**

## **ΠΕΡΙΛΗΨΗ**

Η εργασία αυτή περιγράφει την δημιουργία ενός sonar radar με την χρήση του συστήματος arduino και του περιβάλλοντος γραφικών Processing 4. Με την συμβολή της συνδεσμολογίας των ηλεκτρονικών στοιχείων (ultrasonic sensor, servo motor) και των προγραμμάτων που υλοποιήθηκαν στα δύο περιβάλλοντα (ArduinOS και Processing 4 , επιτεύχθηκε ο στόχος, ο οποίος ήταν ο έλεγχος ύπαρξης εμποδίων στον χώρο και η γραφική απεικόνιση των δεδομένων στον υπολογιστή.

## **ΠΡΟΛΟΓΟΣ**

Η αναφορά αυτή αποτελεί προϊόν της τελειοποίησης της συγκεκριμένης εργασίας στο μάθημα <<Εισαγωγή στην επιστήμη του ηλεκτρολόγου μηχανικού>> , του τμήματος ηλεκτρολόγων μηχανικών και τεχνολογίας υπολογιστών του Πανεπιστημίου Πατρών, με υπεύθυνο καθηγητή τον Κύριο Αθανάσιο Σκόδρα.

## **ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ**

|        |                              |       |
|--------|------------------------------|-------|
| 1.     | Σκοπός.....                  | Σελ 1 |
| 2.     | Εργαλεία - Budget.....       | Σελ 1 |
| 3.     | Μεθοδολογία - Υλοποίηση..... | Σελ 1 |
| 3.1.   | Περιγραφή Μεθόδων.....       | Σελ 1 |
| 3.2.   | Συνδεσμολογία.....           | Σελ 2 |
| 3.3.   | Προγραμματισμός.....         | Σελ 3 |
| 3.3.1. | ArduinOS.....                | Σελ 3 |
| 3.3.2. | Processing.....              | Σελ 4 |
| 4.     | Αποτελέσματα.....            | Σελ 5 |
| 5.     | Συμπεράσματα.....            | Σελ 6 |
| 6.     | Βιβλιογραφία - Αναφορές..... | Σελ 7 |

## **1) ΣΚΟΠΟΣ**

Σκοπός της εργασίας μας είναι η δημιουργία ενός sonar radar με την καλύτερη δυνατή ακρίβεια χρησιμοποιώντας ηλεκτρονικά στοιχεία συμβατά με funduino. Παράλληλα, στόχος μας ήταν η εξοικείωση με την λειτουργία και την συνδεσμολογία του arduino και του processing, αλλά και ο πειραματισμός με το περιβάλλον του ArduinoOS. Επίσης, έγινε επαφή με διάφορα προγράμματα μοντελοποίησης κυκλωμάτων Arduino. Τέλος, σκοπός μας ήταν η ολοκλήρωση της εργασίας μέσα στα κατάλληλα χρονικά περιθώρια.

## **2) ΕΡΓΑΛΕΙΑ - BUDGET**

- Funduino UNO Rev - 17 €
- Servo Motor - 4 €
- Ultrasonic Sensor - 2 €
- Breadboard - 3 €
- Jump Wires - 2 €
- Ξύλο για κατασκευή περιβλήματος - 10 €
- Μπογιά - 2 €
- Αφρώδης ταινία διπλής όψεως 6 €

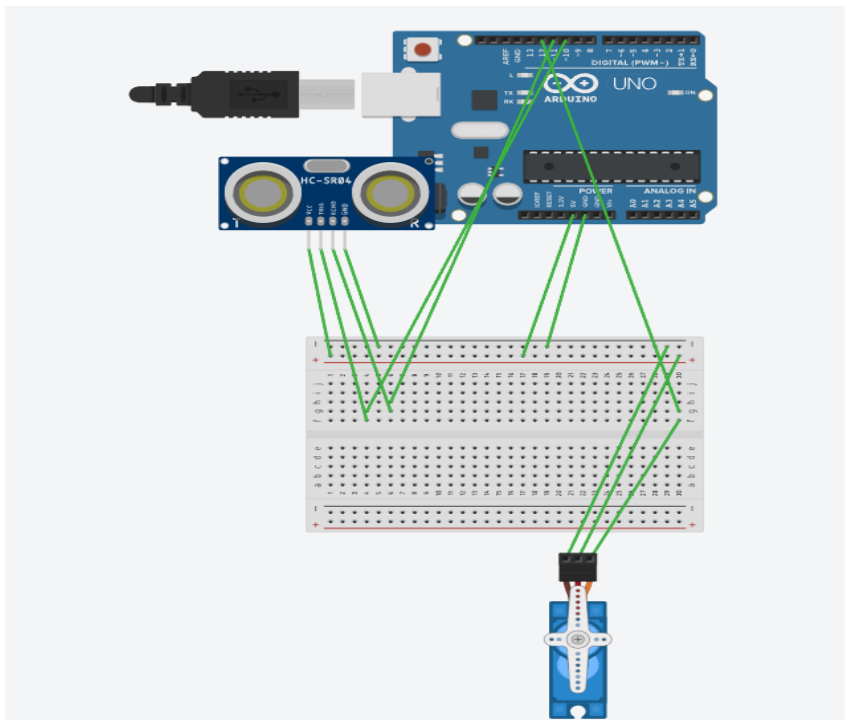
## **3) ΜΕΘΟΔΟΛΟΓΙΑ - ΥΛΟΠΟΙΗΣΗ**

### **3.1) ΠΕΡΙΓΡΑΦΗ ΜΕΘΟΔΩΝ**

Η υλοποίηση της εργασίας ξεκίνησε με προσωπική έρευνα από όλα τα μέλη της ομάδας πάνω στο αντικείμενο. Έγινε μελέτη τόσο του υλικού που μας προσφέρθηκε από τον διδάσκοντα καθηγητή, αλλά και εξωτερικών πηγών στο διαδίκτυο. Συγκεκριμένα, έγινε εξοικείωση με την συνδεσμολογία τόσο του arduino, όσο και της ειδικής γλώσσας προγραμματισμού ArduinoOS. Στην συνέχεια, λόγω ανάγκης υλοποίησης γραφικών που αναπαριστούν την κίνηση του radar, αποφασίστηκε η χρήση του Processing 4. Έπειτα, ξεκίνησε η υλοποίηση της εργασίας. Αρχικά, συνδέθηκε το arduino με τα υπόλοιπα ηλεκτρονικά στοιχεία (breadboard, servo motor, ultrasonic sensor, jump wires) και πραγματοποιήθηκαν μετρήσεις του συστήματος ώστε να δημιουργηθεί ξύλινο περίβλημα από ειδικό (για αισθητικούς λόγους). Εν συνεχεία, δημιουργήθηκε ο κώδικας στο περιβάλλον Arduino IDE, ο οποίος πραγματοποιεί την στρέψη του servo motor, αλλά και την εύρεση εμποδίου και τον υπολογισμό της απόστασής του από το σύστημα. Επιπρόσθετα, για την γραφική απεικόνιση της λειτουργίας του radar, υλοποιήθηκε κώδικας σε Processing 4, ο οποίος μέσω σειριακών δεδομένων που λαμβάνονται μέσω σειριακής πύλης από το arduino, μοντελοποιεί τις ενδείξεις του ultrasonic sensor. Τέλος, πραγματοποιήθηκαν δοκιμές και προστέθηκαν τελευταίες λεπτομέρειες για την ολοκληρωμένη και σωστή επίτευξη της εργασίας μας.

### **3.2) ΣΥΝΔΕΣΜΟΛΟΓΙΑ**

Ο ultrasonic sensor έχει τέσσερα καλώδια: VCC, Trig, Echo και GND. Το VCC καλώδιο συνδέεται με τον ακροδέκτη των 5V του arduino και το GND καλώδιο συνδέεται με τον GND ακροδέκτη του arduino, δηλαδή με την γείωση. Τα άλλα δύο καλώδια, (Trig και Echo) συνδέονται σε δύο οποιουσδήποτε ακροδέκτες του arduino. Το Trig καλώδιο στέλνει υπερηχητικό κύμα από τον πομπό, ενώ το Echo καλώδιο επιτρέπει την ανάγνωση του ανακλασμένου σήματος. Χρησιμοποιώντας την ταχύτητα του ήχου (δηλαδή των υπερηχητικών κυμάτων) και τον χρόνο που χρειάστηκε το υπερηχητικό κύμα για να φύγει από τον αισθητήρα και να γυρίσει σε αυτόν, υπολογίζουμε την απόσταση των εμποδίων στον κώδικα. Ο servo motor έχει τρία καλώδια: ένα καλώδιο ισχύος (συνήθως είναι κόκκινο), ένα καλώδιο γείωσης (συνήθως είναι μαύρο ή καφέ) και ένα καλώδιο σήματος (συνήθως είναι κίτρινο ή πορτοκαλί). Το καλώδιο ισχύος συνδέεται με τον ακροδέκτη 5V στο arduino, το καλώδιο γείωσης συνδέεται με τον GND ακροδέκτη στο arduino και το καλώδιο σήματος συνδέεται σε έναν ακροδέκτη του arduino που υποστηρίζει λειτουργία PWM. Η λειτουργία αυτή, χρησιμοποιείται για τον έλεγχο της έντασης του σήματος. Οι παραπάνω συνδέσεις γίνονται μέσω του breadboard, το οποίο βοηθάει στην πιο εύκολη συνδεσμολογία των ηλεκτρονικών στοιχείων, δίνοντας παραπάνω ακροδέκτες. Όλες οι οριζόντιες γραμμές ακροδεκτών στον breadboard συνδέονται μεταξύ τους. Έτσι, για να συνδέσουμε δύο καλώδια μεταξύ τους, αρκεί να τα συνδέσουμε με ακροδέκτες που βρίσκονται στην ίδια οριζόντια γραμμή. Τέλος, συνδέουμε το καλώδιο που δέχεται ρεύμα από τον ακροδέκτη 5V του arduino σε έναν ακροδέκτη με θετικό πρόσημο στο breadboard και συνδέουμε το καλώδιο από την γείωση του arduino σε έναν ακροδέκτη με αρνητικό πρόσημο στο breadboard.[4][5][6]



**Συνδεσμολογία Arduino και ηλεκτρονικών στοιχείων[2]**

### **3.3) ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ**

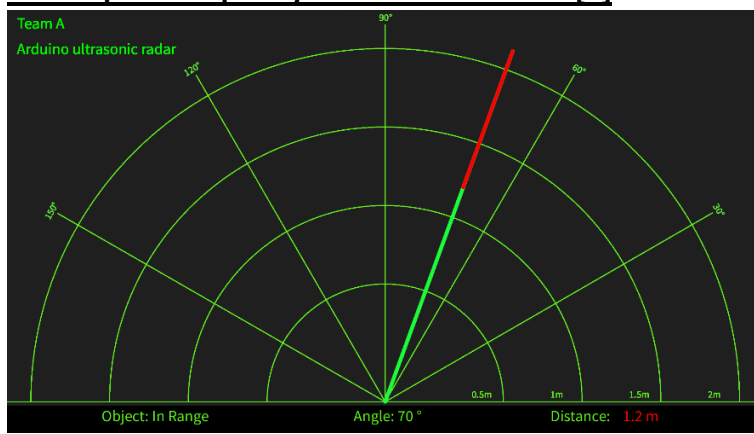
#### **3.2.1) ArduinOS:**

Το κομμάτι του ArduinOS, αποτελείται από δύο βασικές συναρτήσεις: την setup και την loop και μία βοηθητική συνάρτηση την find\_dist για τον υπολογισμό της απόστασης. Αρχικά, με την εκτέλεση του ArduinOS, ανοίγει μία σειριακή πύλη (COM3). Με την πρώτη συνάρτηση του κώδικα, την setup, υλοποιείται η επικοινωνία μεταξύ άλλων περιβαλλόντων (πχ.laptop), δίνεται η δυνατότητα δημιουργίας δεδομένων, τα οποία έπειτα χρησιμοποιούνται στο κομμάτι του processing, συνδέονται ο servo motor και ο αισθητήρας καθώς και ενεργοποιούνται τα διάφορα στοιχεία και πινάκια του. Εν συνεχεία, η συνάρτηση loop χρησιμοποιεί δύο for loops, τα οποία λαμβάνει 180 μοίρες από δεξιά προς τα αριστερά και από αριστερά προς τα δεξιά. Για κάθε μοίρα που προσλαμβάνει στέλνει ένα μήνυμα στην πύλη με την γωνία και την απόσταση σε μορφή: “γωνία,απόσταση.”. Με αυτόν τρόπο δεν σταματά ποτέ η διαδικασία loop και συνεχίζει επ’ αόριστον να λαμβάνει πληροφορίες. Τέλος, υπάρχει η συνάρτηση find\_dist, η οποία χρησιμοποιείται για την εύρεση της απόστασης των εμποδίων. Πιο συγκεκριμένα, δέχεται τα δεδομένα από τον sensor motor για τον χρόνο και έπειτα με βάση τον τύπο  $x = v/(t/2)$  όπου  $x$  : μετατόπιση  $v$  : ταχύτητα ήχου και  $t/2$  : χρόνος διάδοσης κύματος - ο χρόνος που δέχεται ως πληροφορία ο κώδικας είναι ο χρόνος μεταφοράς του κύματος μέχρι να βρεθεί ξανά στην αρχική του θέση και έτσι, πρέπει να ληφθεί ο μισός χρόνος, ο οποίος αποτυπώνει τον χρόνο μέχρι την συνάντηση του εμποδίου.[1]

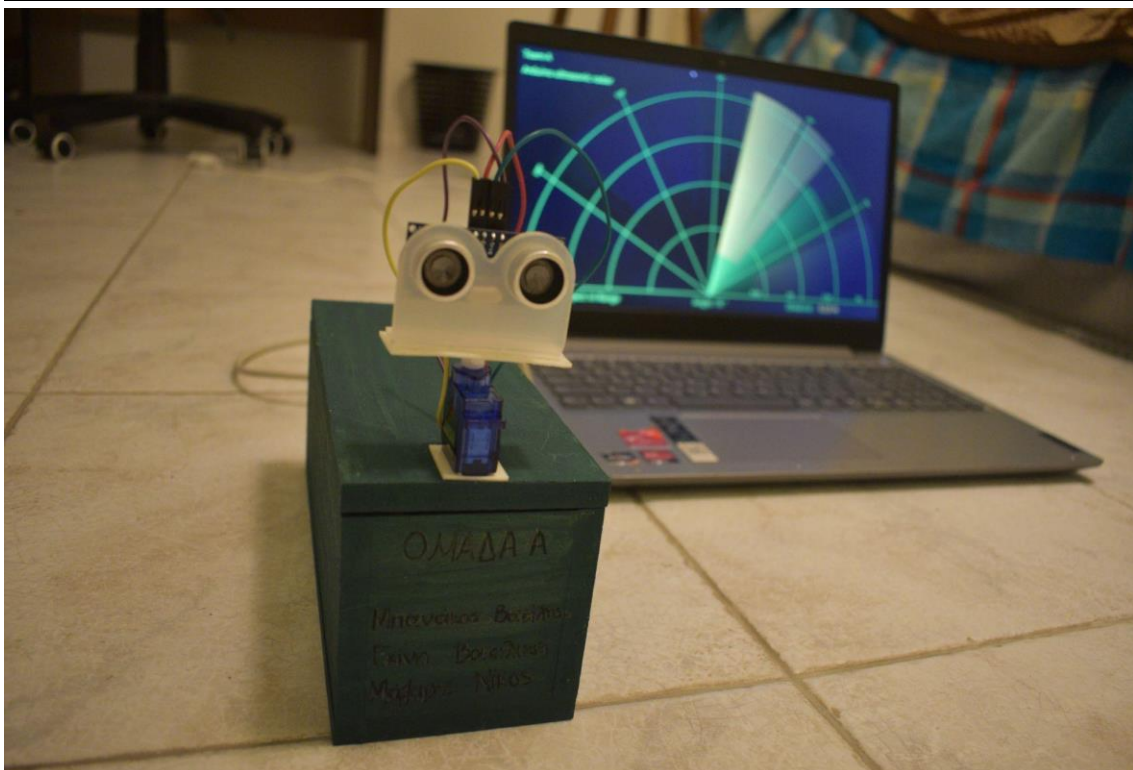
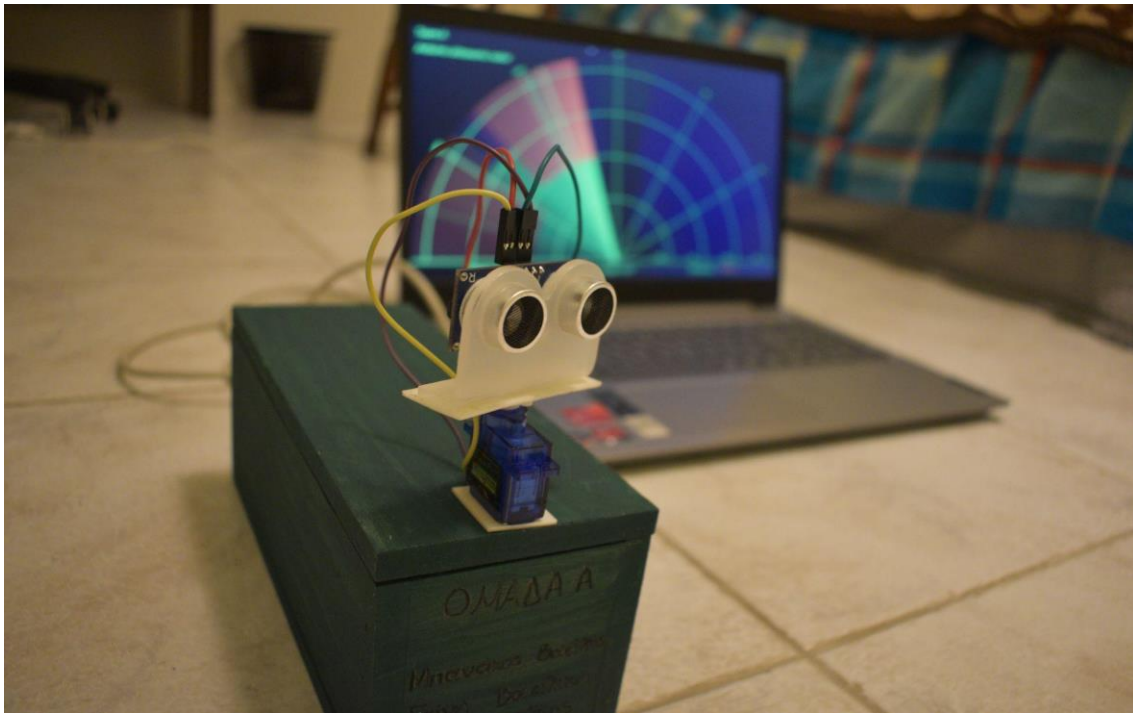
### 3.2.2) Processing

Η χρήση του Processing είναι αναγκαία για την αναπαράσταση των δεδομένων που προλαμβάνει και δημιουργεί το σύστημα, σε ένα γραφικό περιβάλλον radar. Αυτό το κομμάτι του κώδικα αποτελείται από τις συναρτήσεις `setup`, `draw`, `serialEvent`, `BasicRadarStructure`, `FindObject`, `Line` και `Text`. Η πρώτη εκτελεί την διαδικασία της ενεργοποίησης του παραθύρου, αλλά και την δυνατότητα εκμετάλλευσης των σειριακών δεδομένων από το `arduino`. Η συνάρτηση `draw`, οριοθετεί τα πλαίσια του βασικού μέρους του παραθύρου στους τομείς των χρωμάτων αλλά και της διαφοροποίησης της γραμμής των `texts`, και του `radar`, και τελικά καλεί όλες τις συναρτήσεις υπεύθυνες για την τελική αναπαράσταση. Η `serialEvent`, δέχεται για κάθε μοίρα του `servo`, τα δεδομένα από την σειριακή πύλη και τα διαχωρίζει δημιουργώντας δύο βασικά δεδομένα-μεταβλητές για το πρόγραμμα, τις μοίρες του `radar` και την απόσταση του αντικειμένου. Η συνάρτηση `BasicRadarStructure`, είναι υπεύθυνη για τον σχεδιασμό των ημικυκλίων αλλά και των γραμμών που αντιπροσωπεύουν την θέση του `radar` και δεδομένου μέτρου γωνίες. Επίσης η συνάρτηση `FindObject` μετατρέπει την απόσταση που αντικείμενο που ανίχνευσε ο αισθητήρας σε `pixel`, και ανάλογα αυτής, δημιουργεί μια γραμμή με αντίστοιχο χρώμα και μήκος, πάνω στην ήδη υπάρχουσα γραμμή της εκάστοτε μοίρας (ώστε να δείξει την ύπαρξη του αντικειμένου). Η `Line` είναι υπεύθυνη για την συνεχή δημιουργία της πράσινης γραμμής που απεικονίζει την γωνία που βρίσκεται το `arduino` την συγκεκριμένη χρονική στιγμή. Τέλος, η `Text`, τοποθετεί στο περιβάλλον κάθε είδους `text` που υπάρχει, συμπεριλαμβανομένης της συνεχούς ένδειξης της απόστασης (`text`, έλεγχος, χρώμα) αλλά και των μοιρών.[3]

#### Αναπαράσταση ενός frame από το radar[3]



#### 4) ΑΠΟΤΕΛΕΣΜΑΤΑ



Φωτογραφίες τελικού αποτελέσματος



## **5) ΣΥΜΠΕΡΑΣΜΑΤΑ**

Αρχικά, είναι σημαντικό να αναφερθεί ότι για την υλοποίηση του project μας, χρειάστηκαν πολλοί και διαφορετικοί πόροι. Συγκεκριμένα, ήταν αναγκαία η χρήση του διαδικτύου για έρευνα και εκτενή μελέτη του θέματος από πολλές πηγές. Ήταν αναγκαίος ο συνδυασμός και η συνεργασία δύο διαφορετικών IDEs (ArduinOS και Processing), κάτι το οποίο αποτέλεσε δυσκολία, καθώς ήταν η πρώτη φορά που ήρθαμε αντιμέτωποι μια τέτοια κατάσταση. Ταυτόχρονα, το γεγονός ότι το Processing χρησιμοποιεί σειριακή πύλη προερχόμενη από το Arduino και δέχεται σειριακά δεδομένα, καθώς και το γεγονός ότι το Processing είναι γραμμένο σε Java, αποτέλεσαν πρόκληση. Επίσης, απαιτήθηκαν διάφορες δοκιμές και μετρήσεις για την κατάληξη στον σωστό τύπο της αναλογίας απόστασης (σε εκατοστά) και pixels.



## **ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΑΙ ΑΝΑΦΟΡΕΣ**

1. [https://howtomechatronics.com/projects/arduino-radar-project/?utm\\_content=cmp-true](https://howtomechatronics.com/projects/arduino-radar-project/?utm_content=cmp-true)
2. <https://www.tinkercad.com/dashboard>
3. <https://processing.org/>
4. [https://www.tutorialspoint.com/arduino/arduino\\_dc\\_motor.htm](https://www.tutorialspoint.com/arduino/arduino_dc_motor.htm)
5. <https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>
6. <https://docs.arduino.cc/learn/electronics/servo-motors>