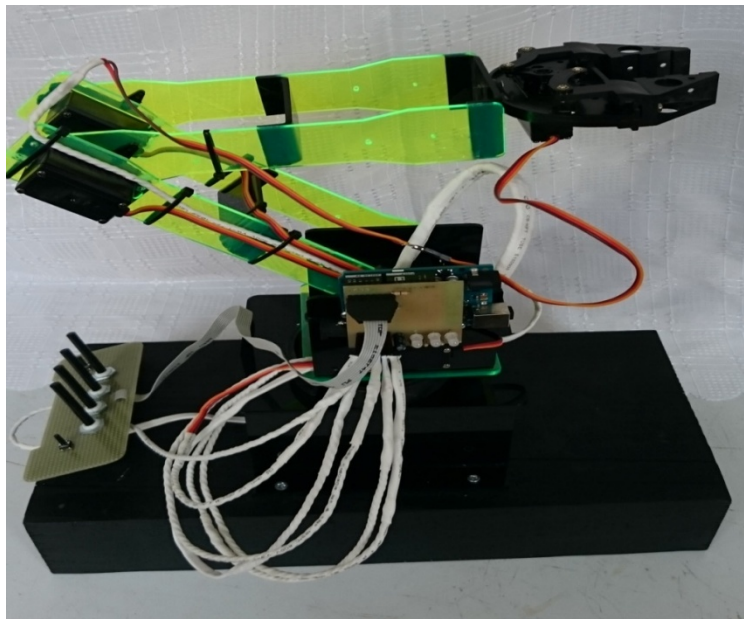


**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ  
ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

*Κατασκευή πρότυπου ρομποτικού βραχίονα 4<sup>ων</sup> βαθμών ελευθερίας και έλεγχος του μέσω της πλατφόρμας Arduino*



**Φοιτητές:**

*Μεντεσιδης Χαράλαμπος Α.Μ. 2489    Νικηφορίδης Ιωάννης Α.Μ. 2238*

*Επιβλέπων Καθηγητής: Καζαρλής Σπυρίδων*

**Σέρρες 2015-2016**

*Κατασκευή πρότυπου ρομποτικού βραχίονα 4<sup>ων</sup> βαθμών  
ελευθερίας και έλεγχος του μέσω της πλατφόρμας  
Arduino*

## **ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ**

Η εκπόνηση της παρούσας πτυχιακής εργασίας με τίτλο «Κατασκευή πρότυπου ρομποτικού βραχίονα 4<sup>ov</sup> βαθμών ελευθερίας και έλεγχός του μέσω της πλατφόρμας Arduino» εκπονήθηκε από τους Μεντεσίδη Χαράλαμπο και Νικηφορίδη Ιωάννη, υπό την επίβλεψη του καθηγητή κ. Καζαρλή Σπυρίδωνα, και προετοιμάστηκε προσωπικά από την συγγραφική ομάδα για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Μηχανικών Πληροφορικής ΤΕ. Βεβαιώνεται ότι οι προαναφερθέντες φοιτητές είναι οι συγγραφείς αυτής της πτυχιακής εργασίας και κάθε βοήθεια από πηγές, από τις οποίες έγινε χρήση δεδομένων και πληροφοριών, αναφέρεται πλήρως στην βιβλιογραφία.

## ***ΕΥΧΑΡΙΣΤΙΕΣ***

Θα θέλαμε να ευχαριστήσουμε θερμά τον καθηγητή κ. Καζαρή Σπυρίδωνα κυρίως για την εμπιστοσύνη που μας έδειξε, αλλά και για την υπομονή που έκανε κατά τη διάρκεια υλοποίησης της πτυχιακής εργασίας. Παράλειψη θα ήταν να μην αναφέρουμε την πολύτιμη βοήθεια του και καθοδήγηση του, για την επίλυση διάφορων θεμάτων. Τέλος, θα θέλαμε να απευθύνουμε ευχαριστίες στους γονείς μας, οι οποίοι στήριξαν τις σπουδές μας με διάφορους τρόπους, φροντίζοντας για την καλύτερη δυνατή μόρφωσή μας και στους φίλους μας για τη στήριξη, τη συμπαράσταση και την κατανόησή τους.

## ΠΕΡΙΕΧΟΜΕΝΑ

1	Πρόλογος.....	7
2	Ρομποτικοί βραχίονες.....	9
2.1	Ρομποτική.....	9
2.1.1.	Ιστορική αναδρομή.....	9
2.1.2.	Τομείς εφαρμογής.....	11
3	Δομή των ρομπότ.....	12
4	Πλεονεκτήματα - Μειονεκτήματα των ρομπότ.....	14
4.1.1	Πλεονεκτήματα των ρομπότ.....	14
4.1.2	Μειονεκτήματα των ρομπότ.....	14
4.2	Αρθρωτό ρομπότ.....	15
4.2.1	Αρθρώσεις σε ρομποτικούς βραχίονες.....	15
4.3	Βαθμοί κινητικότητας και βαθμοί ελευθερίας.....	15
4.3.1	Χώρος Εργασίας.....	16
4.3.2	ΩφέλιμοΦορτίο-Επαναληψιμότητα -Ακρίβεια.....	16
5	Κινηματική θεωρία.....	18
5.1	Εισαγωγή στην κινηματική θεωρία.....	18
5.2	Ευθύ κινηματικό πρόβλημα.....	19
5.3	Αντίστροφο κινηματικό πρόβλημα.....	20
6	Arduino.....	21
6.1	Αναλυτική παρουσίαση του μικροελεγκτή arduino.....	21
6.1.1	Ορισμός.....	21
6.1.2	Δυνατότητες.....	21
6.1.3	Χαρακτηριστικά του arduino.....	22
6.1.4	Πλεονεκτήματα.....	22
6.2	Υλικό πλατφόρμας.....	23
6.2.1	Shields.....	24
6.2.2	Μνήμη.....	24
6.2.3	Είσοδοι – Έξοδοι.....	25
6.2.4	Τροφοδοσία.....	26
6.2.5	Ενσωματωμένα LED.....	27
6.3	Ολοκληρωμένο περιβάλλον ανάπτυξης.....	28
7	Κινητήρες servo.....	30

7.1.1	Ορισμός.....	30
7.1.2	Πλεονεκτήματα - Μειονεκτήματα.....	30
7.2	TowerProMG996R - MG995 .....	31
8	Σχεδιασμός – Μελέτη και βήματα κατασκευής του ρομποτικού βραχίονα.....	32
8.1	Εισαγωγή.....	32
8.2	Σχέδια κατασκευής του βραχίονα .....	32
8.3	Μελέτη και βήματα .....	34
8.4	Κατασκευή πλακέτας shield.....	38
8.5	Κατασκευή τηλεχειριστηρίου .....	41
9	Πρόγραμμα.....	44
10	Ανάλυση του τρόπου λειτουργίας του βραχίονα.....	55
11	Πειράματα .....	57
12	Προβλήματα - Βελτιώσεις.....	58
13	Επίλογος.....	62
14	Κοστολόγιο .....	63
15	Βιβλιογραφία.....	64

# 1 Πρόλογος

Η επιστήμη της ρομποτικής πραγματεύεται την σχεδίαση κατασκευή και λειτουργία μηχανισμών που προσομοιώνουν ανθρώπινα μέλη, ανθρώπινες δυνατότητες, αυτοματοποιημένα συστήματα ή και ανθρωπόμορφες κατασκευές που σκοπό έχουν να αναλάβουν διαδικασίες και αποστολές που κανονικά θα απαιτούσαν την παρουσία ανθρώπου. Οι μηχανισμοί αυτοί μπορούν να εκτελούν λειτουργίες με μεγαλύτερη ταχύτητα, με μεγαλύτερη δύναμη ή ισχύ, με μεγαλύτερη ακρίβεια, με μεγαλύτερη επαναληψιμότητα, με μικρότερες πιθανότητες λάθους και ενδεχομένως σε περιβάλλοντα που θα μπορούσαν να είναι επικίνδυνα για τον άνθρωπο.

Σήμερα, η ρομποτική αποτελεί σημαντικό μέρος της ίδιας της ζωής του ανθρώπου καθώς έχει συμβάλει στην καλυτέρευση της, όταν υπάρχει κάποια ιδιαιτερότητα, με τη δημιουργία ρομποτικών οργάνων, όπως όργανα ακοής, ρομποτικά χέρια και πόδια. Η συμβολή της, όμως, σε τομείς όπως η ιατρική, η εκπαίδευση, η βιομηχανία, η εξερεύνηση του διαστήματος, η οικιακή εργασία, αλλά και πολλοί άλλοι, κάνει φανερό πόσο συνυφασμένη είναι πλέον η ρομποτική με την ανθρώπινη ζωή. Σε ορισμένες περιπτώσεις η χρήση της έχει εξαλείψει εντελώς την ανθρώπινη εργασία, με απλά λόγια γίνεται αντικατάσταση του ανθρώπινου δυναμικού με ρομπότ. Πέρα από τα αρνητικά που αφορούν το κατά πόσο οι άνθρωποι μπορούν να βρίσκουν πλέον δουλειά, αυτό που εντυπωσιάζει είναι η πρόοδος της επιστήμης της ρομποτικής, γιατί πράγματα που παλαιότερα έμοιαζαν απίθανα να πραγματοποιηθούν, τώρα είναι πλέον εφικτά. Ένα από τα πολλά παραδείγματα που μπορεί να παρατεθεί αφορά τον κλάδο της ιατρικής, όπου ορισμένες εγχειρήσεις γίνονται αποκλειστικά από ρομπότ.

Σκοπός της παρούσας πτυχιακής εργασίας είναι η σχεδίαση και η κατασκευή ενός απλού ρομποτικού βραχίονα 4<sup>ov</sup> βαθμών ελευθερίας (περιστροφή βάσης, περιστροφή άκρων, άνοιγμα – κλείσιμο αρπάγης) με έτοιμους μηχανισμούς κινητήρων servo μοντελισμού και ελαφριά υλικά, όπως αλουμίνιο, plexiglass ή EPOfoam.

Οι σερβομηχανισμοί είναι έτοιμα RCservos που χρησιμοποιούνται στον μοντελισμό και οδηγούνται με σήμα PWM (Pulse Width Modulation, Διαμόρφωση Εύρους Παλμού), όπου η διάρκεια του θετικού παλμού (dutycycle) καθορίζει και την γωνία στροφής του servo. Τα servo οδηγούνται με ένα καλώδιο τριών ακροδεκτών, εκ των οποίων οι δύο αποτελούν την τροφοδοσία και ο τρίτος μεταφέρει το PWM σήμα. Οι τέσσερις σερβομηχανισμοί του βραχίονα ελέγχονται από μία πλατφόρμα arduino μέσω της οποίας μπορεί να εκτελεί κινήσεις κατευθυνόμενος από τον χειριστή ή να εκτελεί συγκεκριμένες ακολουθίες κινήσεων εκτελώντας με αυτόν τον τρόπο ένα «πρόγραμμα». Είναι υψηλής ροπής για να κινούν τον

βραχίονα και τροφοδοτούνται από ανεξάρτητο τροφοδοτικό 5V που έχει το απαραίτητο ρεύμα στους σερβομηχανισμούς. Τέλος, ο βραχίονας είναι δυνατόν να εκτελεί απλές κινήσεις μετακίνησης ελαφριών αντικειμένων, τα οποία θα βρίσκονται σε προκαθορισμένες θέσεις.



## **2 Ρομποτικοί βραχιόνες**

### **2.1 Ρομποτική**

Η ρομποτική είναι ο σύγχρονος τεχνολογικός κλάδος της αυτοματοποίησης, που έχει ως αντικείμενο τη σχεδίαση, τη σύλληψη, τη θεωρία, την κατασκευή και τις εφαρμογές των ρομπότ καθώς και την έρευνα για την περαιτέρω ανάπτυξή τους. Το ρομπότ είναι μηχανική συσκευή που δεν έχει απαραίτητα τη συμπεριφορά ή τη μορφή του ανθρώπου, αλλά που μπορεί να υποκαθιστά τον άνθρωπο σε διάφορες εργασίες. Ένα ρομπότ μπορεί να δράσει κάτω από τον απευθείας έλεγχο ενός ανθρώπου ή αυτόνομα κάτω από τον έλεγχο ενός προγραμματισμένου υπολογιστή. Επίσης μπορούν να χρησιμοποιηθούν για την αντικατάσταση του ανθρώπου στην εκτέλεση μιας εργασίας, η οποία είτε είναι δύσκολη είτε επικίνδυνη για να γίνει απευθείας από αυτόν και η οποία αφορά τόσο στο φυσικό επίπεδο, όσο και στο επίπεδο λήψης αποφάσεων.

Ο συγγραφέας βιβλίων επιστημονικής φαντασίας Isaac Asimov χρησιμοποίησε για πρώτη φορά τη λέξη "ρομποτική" για να περιγράψει την τεχνολογία των ρομπότ. Έθεσε κάποιους κανόνες για τη λειτουργία των ρομπότ που ισχύουν μέχρι σήμερα.

#### **Κανόνες του Isaac Asimov:**

- Ένα ρομπότ δεν επιτρέπεται ενεργητικά, ή λόγω απραξίας του να βλάψει ένα ανθρώπινο ον.
- Το ρομπότ πρέπει να υπακούει ένα ανθρώπινο ον, εκτός αν οι εντολές που δέχεται αντιτίθενται στον Πρώτο Νόμο.
- Το ρομπότ πρέπει να προστατεύει την ύπαρξή του, αρκεί αυτό να μην συγκρούεται με τον πρώτο και τον δεύτερο νόμο.

#### **2.1.1. Ιστορική αναδρομή**

Ένα από τα πρώτα ρομπότ που αναφέρονται στην ιστορία του ανθρώπινου πολιτισμού είναι ο Τάλως από την ελληνική μυθολογία. Ο Τάλως ήταν μυθικός φύλακας ο οποίος ήταν επιφορτισμένος με το καθήκον να επιτηρεί την εφαρμογή των νόμων στην Κρήτη κουβαλώντας τους μαζί του που ήταν γραμμένοι σε χάλκινες πλάκες. Ήταν φτιαγμένος από χαλκό και είχε γιγάντιο ανθρωπόμορφο σώμα. Όσον αφορά την προέλευσή του, υπάρχουν πολλές εκδοχές. Η πιο γνωστή είναι από τον Απολλόδωρο που αναφέρει πως κατασκευάστηκε από τον θεό Ήφαιστο και τον χάρισε στο βασιλιά Μίνωα για να προστατεύει την Κρήτη. Ο

Πλάτων τον θεωρεί υπαρκτό πρόσωπο και αδελφό του Ροδάμανθου. Ο Απολλώνιος ο Ρόδιος λέει πως ήταν δώρο του Δία στην Ευρώπη, η οποία με την σειρά της τον χάρισε στο γιο της Μίνωα. Μεταγενέστερα ο Ι. Κακριδής, εκφράζει την άποψη ότι ο Τάλως ήταν ηλιακή θεότητα που αργότερα μεταπλάστηκε σε ήρωα βασιζόμενος στο ότι ο Ησύχιος αναφέρει πως η λέξη «ταλῶς» σήμαινε ήλιος στην Κρήτη.

Η επιστήμη της ρομποτικής στη σύγχρονη εποχή αναπτύσσεται με ραγδαίους ρυθμούς. Τα σημερινά ρομπότ δεν έχουν καμία σχέση με τα πρώτα ρομπότ που φτιάχτηκαν κοντά στο 1950. Τα σημαντικότερα ρομπότ των προηγούμενων δεκαετιών είναι τα εξής:

**UNIMATE 1961:** Πρόκειται για το πρώτο βιομηχανικό ρομπότ που χρησιμοποιήθηκε συγκεκριμένα στην κατασκευή αυτοκινήτων. Το Unimate ήταν υπεύθυνο για την μεταφορά σιδερένιων εξαρτημάτων στο σκελετό του αυτοκινήτου που κατασκευαζόταν. Για τους ανθρώπους ήταν μία εργασία πολύ επικίνδυνη, καθώς μπορεί να τραυματίζονταν αν δεν ήταν αρκετά προσεκτικοί, ενώ εισέπνεαν και τοξικά αέρια.

**RANCHO ARM 1963:** Αποτελεί μία από τις πρώτες προσπάθειες για την εφαρμογή των ρομπότ στον κλάδο της ιατρικής. Πιο συγκεκριμένα το RanchoArm χρησιμοποιήθηκε στο νοσοκομείο Λος Αμίγκος στην Καλιφόρνια για την δημιουργία τεχνητών μελών για άτομα με ειδικές ανάγκες.

**SHAKEY 1970 :** Το Shakey είναι το πρώτο ρομπότ που μπορεί να κινείται. Έχει δημιουργηθεί από επιστήμονες του τεχνολογικού ινστιτούτου SRI (Stanford Research Institute) στις ΗΠΑ. Μπορούσε να μετακινείται στο χώρο και να αναγνωρίζει τον λόγο για τον οποίο κάνει μία ενέργεια.

**DANTE I 1992:** Δημιουργήθηκε από την επιστημονική ομάδα του πανεπιστημίου Κάρνεγκι Μέλον στις ΗΠΑ . Το ρομπότ αυτό επέτρεψε στον άνθρωπο να μελετήσει για πρώτη φορά από πάρα πολύ κοντά τον κρατήρα ενός ενεργού ηφαιστείου . Ο DanteI μπόρεσε να μπει στον κρατήρα του ηφαιστείου του βουνού Σπουρ στην Αλάσκα και να συλλέξει πολύτιμα ευρήματα. Στη μορφή θύμιζε αράχνη, για να μπορεί να σκαρφαλώνει στις επιφάνειες ενός κρατήρα που είναι δύσβατες. Ο Dante είχε στην διάθεση του αισθητήρες και υπερσύγχρονες κάμερες και μπορούσε να φέρει πολύτιμες πληροφορίες για ένα περιβάλλον, που ο άνθρωπος δεν θα είχε την δυνατότητα να επισκεφθεί ποτέ, εξαιτίας των υψηλών θερμοκρασιών και των δύσκολων συνθηκών.

### **2.1.2. Τομείς εφαρμογής**

Η χρήση των ρομπότ πραγματοποιείται σε αρκετούς παραγωγικούς τομείς και κυρίως στη βιομηχανία. Η μορφή των βιομηχανικών ρομπότ μοιάζει με αυτήν ενός ανθρώπινου βραχίονα με αρθρώσεις (ώμο, αγκώνα, καρπό) και παλάμη (αρπάγη/δαγκάνα, δάκτυλα). Με γνώμονα το είδος της εργασίας που πρέπει να εκτελέσει ένα ρομπότ γίνεται η επιλογή του τύπου της κίνησής του (γραμμική, κυλινδρική, σφαιρική, αρθρωτή). Τα βιομηχανικά ρομπότ είναι κατάλληλα για επαναλαμβανόμενες εργασίες: φόρτωμα/ξεφόρτωμα μηχανών, συναρμολόγηση, κ.α.

Άλλοι τομείς εφαρμογής της ρομποτικής τεχνολογίας είναι η εξερεύνηση του διαστήματος, οι αγροτικές εφαρμογές, η ιατρική, το στρατό, η έρευνα και διάσωση κ.τ.λ. Η ρομποτική τεχνολογία στους τομείς αυτούς, παρά την ύπαρξη πρωτοτύπων, είναι ακόμη σε πρωταρχικό στάδιο. Τα αίτια για το γεγονός αυτό μπορούν να αναζητηθούν στην έλλειψη βαθύτερης κατανόησης των μηχανισμών ελέγχου που επιτρέπουν στον άνθρωπο να χειρίζεται επιδέξια μια ποικιλία αντικειμένων στην καθημερινότητά του. Για παράδειγμα, ο επιτυχής χειρισμός ενός ποτηριού αποτελεί μια καθημερινή ενέργεια ενός ανθρώπου, χωρίς ο ίδιος να είναι σε θέση να εξηγήσει τις λεπτομέρειες και την αλληλουχία των ενεργειών που είχαν ως αποτέλεσμα την επιτυχή έκβαση της δράσης του.

### 3 Δομή των ρομπότ

Τα δύο βασικά στοιχεία που αποτελούν ένα ρομποτικό σύστημα είναι το μηχανικό μέρος και ο ελεγκτής.

#### **Μηχανικό μέρος**

Με τον όρο μηχανικό μέρος ενός ρομποτικού συστήματος αναφερόμαστε στο σύνολο των βραχιόνων του το περιλαμβάνει. Ένα ρομποτικό σύστημα μπορεί να διαθέτει έναν ή περισσότερους βραχίονες ανάλογα με την εργασία για την οποία έχει σχεδιαστεί (Patel, 2006). Κάθε βραχίονας αποτελείται από τα εξής μέρη:

**Βάση:** Αποτελεί τον κορμό του ρομποτικού συστήματος και είναι στερεωμένη στο περιβάλλον εργασίας αυτού. Έτσι σε αυτήν συνδέεται μία αλυσίδα συνδέσμων και αρθρώσεων που καταλήγει στο τελικό εργαλείο δράσης.

**Συνδέσμους:** Είναι στερεά και ως συνήθως μεταλλικά σώματα που συγκροτούν τον σκελετό του ρομποτικού συστήματος.

**Αρθρώσεις:** Είναι μηχανισμοί που επιτρέπουν τη σχετική κίνηση μεταξύ των συνδέσμων. Διακρίνονται σε γραμμικές που επιτρέπουν την κίνηση κατά μήκος ενός άξονα, και σε περιστροφικές που επιτρέπουν την κίνηση γύρω από τον άξονά τους.

**Κινητήρες:** Κάθε άρθρωση χρειάζεται τουλάχιστον ένα κινητήρα. Ο τύπος του κινητήρα μπορεί να είναι υδραυλικός, ηλεκτρικός (βηματικός, σερβοκινητήρας) ή πνευματικός.

**Αισθητήρια:** Είναι στοιχεία υλικού που μπορούν να συλλέγουν πληροφορίες για τον έλεγχο της θέσης του ρομπότ. Για τη συλλογή αυτών των πληροφοριών χρησιμοποιούνται διάφοροι τύποι αισθητήρων, από απλά ποτενσιόμετρα και ταχύμετρα μέχρι ψηφιακοί οπτικοί κωδικοποιητές θέσης (encoders).

**Εργαλείο δράσης:** Είναι το τελικό άκρο του κάθε βραχίονα του ρομποτικού συστήματος, δηλαδή ένα μηχανικό εξάρτημα κατάλληλα σχεδιασμένο για την εκτέλεση της εργασίας για την οποία έχει προγραμματιστεί το συγκεκριμένο σύστημα. Έτσι, ένα εργαλείο δράσης μπορεί να ποικίλλει από έναν βιομηχανικό συγκολλητή μετάλλων, μία "αρπάγη" για τη μεταφορά αντικειμένων μέχρι ένα λεπτό χειρουργικό εργαλείο.

### ***Ελεγκτής***

Η ηλεκτρονική μονάδα που μας δίνει τη δυνατότητα να προγραμματίσουμε ένα ρομποτικό σύστημα ονομάζεται ελεγκτής και αποτελείται από:

**Ηλεκτρονικά (Hardware):** Περιλαμβάνουν έναν μικροελεγκτή ή υπολογιστή, στον οποίο γίνεται η αποθήκευση του προγράμματος που πρόκειται να εκτελεστεί, τα ηλεκτρονικά επικοινωνίας μεταξύ του ελεγκτή, του μηχανικού μέρους και του εξωτερικού περιβάλλοντος του ρομποτικού συστήματος (interface) και τους ενισχυτές ισχύος που ενισχύουν τα σήματα ελέγχου στο επίπεδο που απαιτείται ώστε οι κινητήρες να κινούν τις αρθρώσεις.

**Λογισμικό (Software):** Για τη δημιουργία των κατάλληλων σημάτων ελέγχου, σύμφωνα με κάποιον αλγόριθμο, λαμβάνοντας υπόψη διάφορες μεταβλητές όπως τη θέση, το φορτίο και την ταχύτητα του ρομπότ είναι υπεύθυνο κυρίως το λογισμικό. Το λογισμικό μπορεί να περιλαμβάνει και διάφορα προγράμματα που είναι βοηθητικά για τον έλεγχο της λειτουργίας του ρομπότ, τον προγραμματισμό του και την ενημέρωση του χρήστη με διαγνωστικά μηνύματα

## **4 Πλεονεκτήματα - Μειονεκτήματα των ρομπότ**

### **4.1.1 Πλεονεκτήματα των ρομπότ**

Τα σύγχρονα ρομπότ παρουσιάζουν κάποια πλεονεκτήματα σε σχέση με τον άνθρωπο, μερικά από τα οποία είναι τα εξής:

1. Ευελιξία
2. Ευφυΐα και ειδικούς αισθητήρες
3. Ταχύτητα και ακρίβεια
4. Μεγαλύτερη δύναμη
5. Γρήγορη ανάπτυξη – ολοκλήρωση
6. Οικονομικές λύσεις
7. Σταθερή ποιότητα προϊόντος και παραγωγικότητα
8. Ακρίβεια
9. Επαναληψιμότητα
10. Η απόδοση των ρομπότ είναι ανεξάρτητη από τον αριθμό των επαναλήψεων εκτέλεσης μιας εργασίας
11. Μείωση κόστους
12. Αύξηση της παραγωγικότητας
13. Απαλλαγή ανθρώπου από επικίνδυνες και ανθυγιεινές εργασίες

### **4.1.2 Μειονεκτήματα των ρομπότ**

1. Δεν μπορούν να ανταπεξέλθουν σε εργασίες που απαιτούν νοημοσύνη.
2. Δεν μπορούν να ανταπεξέλθουν σε εργασίες που εκτελούνται σε αβέβαιο περιβάλλον.
3. Μείωση θέσεων εργασίας σε ανιδίκευτο και χαμηλά ειδικευόμενο προσωπικό που δεν αντισταθμίζεται από την δημιουργία νέων θέσεων.

## **4.2 Αρθρωτό ρομπότ**

Μια στρεφόμενη βάση πάνω στην οποία είναι σωστά τοποθετημένα τρία σταθερά μέλη (σύνδεσμοι), που με την σειρά τους συνδέονται με στροφικές αρθρώσεις, συναπαρτίζουν ένα αρθρωτό ρομπότ, που κατά γενική ομολογία, η κινητική του διάταξη εμφανίζει πολλά κοινά με το χέρι του ανθρώπου. Το εργαλείο, δηλαδή η αρπάγη που, φυσικά, είναι προσαρμοσμένο κατ' αναλογία της παλάμης, μέσω του καρπού συνδέεται με τον πήχη. Ακόμη, ο πήχης συνδέεται με τον άνω βραχίονα με την βοήθεια του «αγκώνα» και τέλος, ο άνω βραχίονας συνδέεται με το σταθερό μέρος, τη βάση, μέσω του «ώμου». Δεν είναι λίγες οι φορές που έχει παρατηρηθεί στην άρθρωση του «ώμου» να υπάρχει σε οριζόντιο, βέβαια, επίπεδο και μία περιστροφική κίνηση.

### **4.2.1 Αρθρώσεις σε ρομποτικούς βραχίονες**

Οι περισσότεροι ρομποτικοί βραχίονες, βασίζονται σε βραχίονες που συνδέονται μεταξύ τους με αρθρώσεις ενός βαθμού ελευθερίας. Οι ρομποτικοί βραχίονες, για να έχουν την ευελιξία που προσδοκούμε, σχεδιάζονται συνήθως με αρκετές (από τρεις μέχρι έξι) αρθρώσεις, τοποθετημένες κατάλληλα έτσι ώστε να επιτρέπουν στους επενεργητές να κινούνται με ευκολία.

Πιο συγκεκριμένα, οι αρθρώσεις αυτές είναι οι: Α) Περιστροφική άρθρωση (revolute joint), η οποία υλοποιεί σχετική περιστροφική κίνηση γύρω από τον άξονά της. Η μεγάλη ταχύτητα είναι το κύριο χαρακτηριστικό της και αυτό την καθιστά ως την πιο κοινή άρθρωση σε ρομποτικούς βραχίονες. Β) Γραμμικές αρθρώσεις που διαχωρίζονται σε ολίσθησης (sliding) και πρισματικές (prismatic) οι οποίες επιτρέπουν την ελεγχόμενη μετακίνηση προς μία κατεύθυνση. Το στοιχείο που τις χαρακτηρίζει είναι η μεγάλη ακρίβειά τους. Για αυτόν τον λόγο χρησιμοποιούνται κυρίως σε διατάξεις συναρμολόγησης.

## **4.3 Βαθμοί κινητικότητας και βαθμοί ελευθερίας**

Η αναφορά στους βαθμούς κινητικότητας και στους βαθμούς ελευθερίας έχει ως σκοπό να επισημανθεί η διαφορά ανάμεσά τους. Αρχικά, από την μία μεριά, ο αριθμός των βαθμών κινητικότητας είναι σταθερός, αλλά και ίσος πάντα με βάση τον αριθμό των αρθρώσεων του κάθε βραχίονα (πρισματικών ή/και περιστροφικών). Από την άλλη μεριά, οι βαθμοί ελευθερίας του βραχίονα είναι ανάλογοι με το έργο που καλείται να εκτελέσει. Σε γενικές γραμμές, για

να επιτευχθεί η τοποθέτηση και ο προσανατολισμός ενός αντικειμένου στον χώρο των τριών διαστάσεων απαιτείται η χρήση έξι βαθμών ελευθερίας, δηλαδή τριών για την τοποθέτηση στο χώρο ενός σημείου του αντικειμένου και τριών για τον προσανατολισμό του αντικειμένου στο σύστημα συντεταγμένων αναφοράς. Ο ρομποτικός βραχίονας, ο οποίος έχει έξι βαθμούς κινητικότητας είναι φανερό ότι μπορεί να αντιμετωπίσει με επιτυχία ένα τέτοιο έργο, όπως και κάθε άλλο έργο στο οποίο απαιτούνται έξι βαθμοί ελευθερίας.

### **4.3.1 Χώρος Εργασίας**

Ως χώρος εργασίας ορίζεται ο τρισδιάστατος χώρος τον οποίο μπορεί να σαρώνει η άκρη του ρομποτικού μηχανισμού. Το μέγεθος και η γεωμετρική μορφή του χώρου αυτού εξαρτώνται από την κατασκευαστική δομή του ρομπότ, κάτι που θα γίνει φανερό και στη συνέχεια.

### **4.3.2 Ωφέλιμο Φορτίο - Επαναληψιμότητα - Ακρίβεια**

Από τα πιο σημαντικά μεγέθη ενός βραχίονα είναι το ωφέλιμο φορτίο, η επαναληψιμότητα και η ακρίβεια. Πιο συγκεκριμένα τα παραπάνω μεγέθη αναφέρονται στα εξής:

**Ωφέλιμο Φορτίο:** είναι το βάρος που μπορεί να μεταφέρει το άκρο του βραχίονα. Ως σημείο εφαρμογής του βάρους θεωρείται η φλάντζα του καρπού. Το προδιαγραφόμενο αυτό φορτίο δεν είναι σταθερό και εξαρτάται από την ταχύτητα με την οποία πρόκειται να κινηθεί ο καρπός.

**Επαναληψιμότητα:** εκφράζει τη δυνατότητα του βραχίονα να γυρίσει στο ίδιο σημείο μετά από αρκετές επαναλήψεις και δίνεται ως εύρος μέσα στο οποίο ο βραχίονας θα τερματίσει την κίνηση. Η απόκλιση οφείλεται στο ότι κατά τη λειτουργία του το ρομπότ είναι δυνατό να χάσει λίγο από τη μέτρηση της θέσης με αποτέλεσμα να μη μπορεί να επιστρέψει στη συγκεκριμένη θέση μετά από ορισμένους κύκλους λειτουργίας. Δεδομένου ότι στις συνήθεις βιομηχανικές εφαρμογές οι επιθυμητές κινήσεις διδάσκονται στο ρομπότ αντιλαμβάνεται κανείς τη σπουδαιότητα της επαναληψιμότητας.

**Ακρίβεια:** είναι η ικανότητα του ρομπότ να πηγαίνει ακριβώς στη θέση που του έχει δοθεί εντολή να πάει. Η ακρίβεια εξαρτάται κυρίως από τη διακριτότητα των εξαρτημάτων ελέγχου, τη μηχανολογική σύνδεση των μελών του και το ελάχιστο επιτρεπόμενο σφάλμα που επιβάλλει η ευστάθεια της λειτουργίας των σέρβο. Η ακρίβεια επηρεάζεται από το είδος και το



μέγεθος του εκάστοτε φορτίου, σε αντίθεση με την επαναληψιμότητα, γι' αυτό και ορισμένοι κατασκευαστές προδιαγράφουν μόνο την τελευταία.

## 5 Κινηματική θεωρία

### 5.1 Εισαγωγή στην κινηματική θεωρία

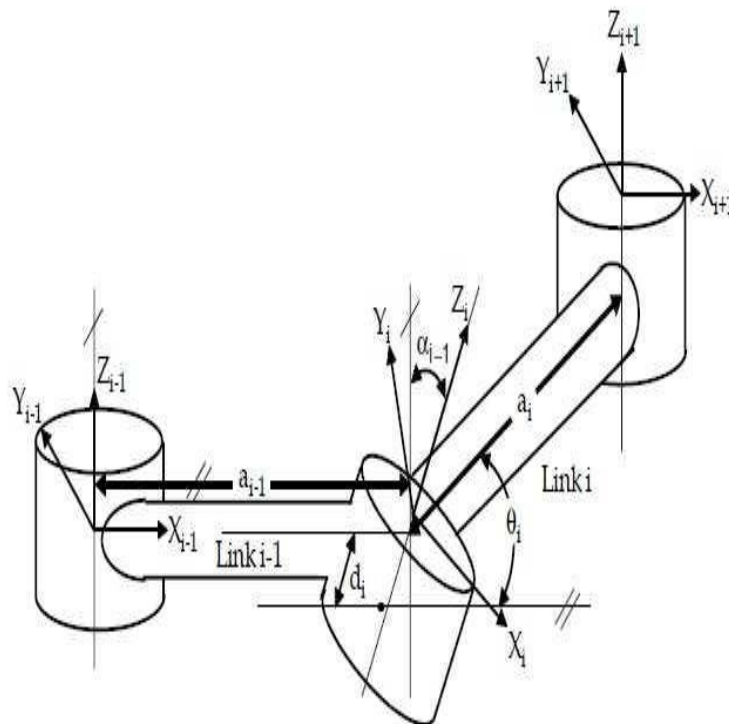
Η Κινητική ή Κινηματική (Kinematics, απ'το ελληνικό κινεῖν) είναι κλάδος της μηχανικής που περιγράφει την κίνηση των σωμάτων αδιαφορώντας για τη μάζα τους ή τις αιτίες, δυνάμεις, που προκαλούν την κίνησή τους. Επίκεντρο του ενδιαφέροντος της κινηματικής είναι η θέση και όλα τα παράγωγα των μεταβλητών της (όπως η ταχύτητα, η επιτάχυνση).

Στην κινηματική ανάλυση του βραχίονα ενδιαφερόμαστε για την επίλυση δύο προβλημάτων, του ευθέως και του αντίστροφου κινηματικού προβλήματος. Στην περίπτωση της εύρεση της θέσης και του προσανατολισμού του άκρου του βραχίονα ως προς την βάση του, όταν γνωρίζουμε την θέση κάθε άρθρωσης συνίσταται το ευθύ κινηματικό πρόβλημα, ενώ το αντίστροφο κινηματικό πρόβλημα στην εύρεση της θέσης και του προσανατολισμού της κάθε άρθρωσης, σε σχέση με την βάση του. Ένω η λύση του ορθού κινηματικού προβλήματος είναι μοναδική, η λύση του ανάστροφου κινηματικού προβλήματος δεν είναι, διότι η κινηματική εξίσωση περιέχει τριγωνομετρικές συναρτήσεις.

Απαραίτητη προϋπόθεση για την μετακίνηση του άκρου του βραχίονα από μία θέση σε μία άλλη είναι η γνώση των γωνιών των αρθρώσεων που αντιστοιχούν σε αυτήν. Επιλύοντας το αντίστροφο κινηματικό πρόβλημα μπορούν να υπολογιστούν οι γωνίες αυτές. Τέλος η γεωμετρία του βραχίονα είναι αυτή που καθορίζει τον βαθμό δυσκολίας επίλυσης των προβλημάτων αυτών.

## 5.2 Ευθύ κινηματικό πρόβλημα

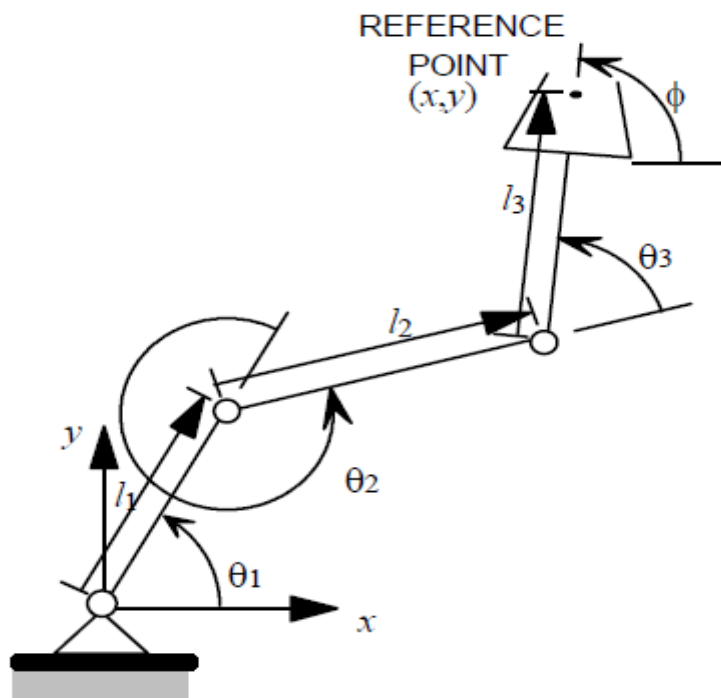
Η πιο κοινή μέθοδος για την περιγραφή του κινηματικού προβλήματος ενός ρομπότ είναι η μεθοδολογία Denavit - Hartenberg που χρησιμοποιεί τέσσερις παραμέτρους. Σε κάθε άρθρωση επισυνάπτεται ένα σύστημα συντεταγμένων και καθορίζει τις DH παραμέτρους. Με την μέθοδο αυτή, γίνεται η επιλογή των συστημάτων συντεταγμένων με τρόπο συγκεκριμένο, τα οποία προσαρμόζονται σε κάθε σύνδεσμο του ρομποτικού βραχίονα και εκτελούν τα βήματα του αλγόριθμου. Τέλος, γίνεται με αυτόν τον τρόπο και ο προσδιορισμός της θέσης, αλλά και του προσανατολισμού του άκρου του ρομποτικού βραχίονα ως προς το ακίνητο σύστημα συντεταγμένων.



5.2. Συντεταγμένες σε γενικού τύπου βραχίονα

### 5.3 Αντίστροφο κινηματικό πρόβλημα

Στο αντίστροφο κινηματικό πρόβλημα μας δίνεται ο προσανατολισμός και η θέση του εργαλείου της τελικής δράσης. Επίσης το αντίστροφο κινηματικό πρόβλημα αναφέρεται στον προσδιορισμό των μεταβλητών των αρθρώσεων του ρομπότ. Τέλος η χρήση μη γραμμικών αλγεβρικών εξισώσεων είναι απαραίτητη για την επίλυση αυτού του προβλήματος.



5.3. Επίπεδο ρομπότ τριών βαθμών ελευθερίας

## 6 *Arduino*

### 6.1 *Αναλυτική παρουσίαση του μικροελεγκτή arduino*

#### 6.1.1 *Ορισμός*

Το arduino είναι ένας single-board μικροελεγκτής, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους. Το arduino μπορεί να διαβάσει το περιβάλλον λαμβάνοντας σήματα από διάφορους αισθητήρες και μπορεί να το επηρεάσει ελέγχοντας φώτα, μηχανισμούς και άλλα ανάλογα με το πώς έχει προγραμματιστεί. Η πλακέτα αυτή μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Η γλώσσα προγραμματισμού Wiring είναι αρκετά εύκολη στη σύνταξη και διατίθεται σε πλατφόρμες Linux, MAC και Windows με άδεια χρήσης GPL. Αυτό, όμως, που κάνει το Arduino ακόμα πιο σημαντικό είναι ότι όλο το κύκλωμα της πλακέτας διατίθεται με άδεια χρήσης CreativeCommons, πράγμα που σημαίνει ότι ο καθένας μπορεί να κατασκευάσει τη δική του πλακέτα όπως αυτός επιθυμεί.



4.1.1.

#### 6.1.2 *Δυνατότητες*

Ο arduino αν και μικροσκοπικός (7x5 cm) οι δυνατότητες που προσφέρει είναι πάρα πολλές. Μπορούμε να το χρησιμοποιήσουμε σε εφαρμογές ρομποτικής και γενικότερα σε αυτοματισμούς καταφέροντας έτσι πάρα πολλά όπως: την κίνηση servo, stepper και DC κινητήρων, τη λήψη πληροφοριών από διάφορους αισθητήρες (θερμοκρασίας, υγρασίας, υπέρυθρων κ.α.), την αμφίδρομη σειριακή επικοινωνία μεταξύ Arduino και PC χρησιμοποιώντας γλώσσες προγραμματισμού (όπως Java και python), όπως επίσης την αναπαραγωγή και αντίληψη ήχων. Η πλακέτα arduino μέχρι αυτή τη στιγμή διατίθεται σε 12 βασικές παραλλαγές (Duemilanove, Diecimila, Nano, Mega, Bluetooth, LilyPad, Mini,

MiniUSB, Pro, ProMini, Serial και SerialSS), οι οποίες αναφέρονται σε διαφορετικές χρήσεις η κάθε μια, ανάλογα με τις ανάγκες της εφαρμογής μας.

### 6.1.3 Χαρακτηριστικά του *arduino*

Microcontroller:	ATmega328
Τάση λειτουργίας:	5V
Τάση εισόδου:	7-12V
Τάση εισόδου (όριο):	6-20V
Ψηφιακοί ακροδέκτες I/O:	14 (6 PWM έξοδοι)
Αναλογικοί ακροδέκτες εισόδου:	6
Ισχύς συνεχόμενου ρεύματος ανά ακροδέκτη	40 mA
Ισχύς συνεχόμενου ρεύματος για ακροδέκτη τάσης 3.3V	50mA
FlashMemory:	32 KB(εκ των οποίων 0,5 KB που χρησιμοποιούνται από τον bootloader)
Μνήμη SRAM	2 KB
Μνήμη EEPROM	1 KB
Ταχύτητα ρολογιού	16 MHz

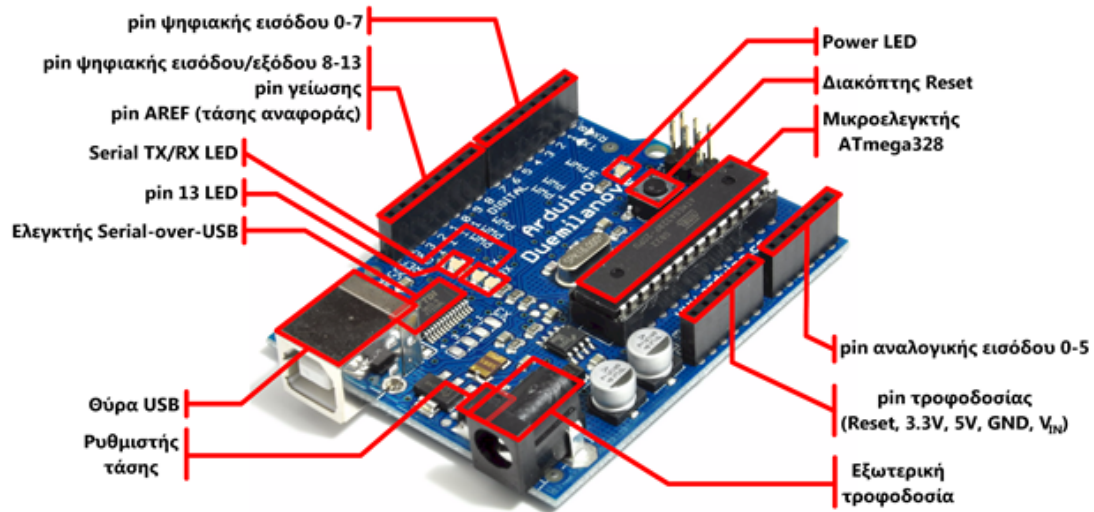
### 6.1.4 Πλεονεκτήματα

Μερικά από τα βασικότερα πλεονεκτήματα της πλατφόρμας *arduino* είναι τα εξής:

- *Οικονομική:* Η πλατφόρμα *Arduino* αποτελεί οικονομική λύση διότι είναι φθηνότερη. Επιπλέον, είναι αρχιτεκτονικά ανοιχτή και μπορεί ο οποιοσδήποτε να την αναπτύξει από μόνος του.
- *Μεταφέρσιμη:* σε σχέση με τις υπάρχουσες πλατφόρμες στο εμπόριο η πλατφόρμα *Arduino* παρέχει πλήρη μεταφερσιμότητα με αποτέλεσμα να μπορεί να προγραμματιστεί στα περισσότερα λειτουργικά συστήματα.
- *Επεκτάσιμη:* το υλικό και το λογισμικό της πλατφόρμας *Arduino* είναι ανοιχτά και ελεύθερα για όλους. Καθημερινά, χιλιάδες υποστηρικτές του ελεύθερου λογισμικού

αναπτύσσουν διάφορες βιβλιοθήκες για την υποστήριξη της πλατφόρμας. Παράλληλα, τόσο η αρχιτεκτονική όσο και το υλικό της πλατφόρμας εξελίσσονται συνεχώς.

## 6.2 Υλικό πλατφόρμας



4.2.

Η πλακέτα Arduino απαρτίζεται από έναν cmos οχτώ ψηφίων AtmelAVR μικροελεγκτή (στις πιο νέες εκδόσεις: ATmega328 και Atmega168 και στις πιο παλαιές εκδόσεις: Atmega8) και εξαρτήματα που την συμπληρώνουν, τα οποία διευκολύνουν τον χρήστη στον προγραμματισμό, αλλά και στο να μπορέσει να ενσωματωθεί σε περαιτέρω κυκλώματα. Ένας γραμμικός ρυθμιστής με τάση 5V, αλλά και ένας κρυσταλλικός ταλαντωτής 16MHz περιλαμβάνονται σε όλες τις πλακέτες. Ο μικροελεγκτής προγραμματίζεται ήδη από τη στιγμή της κατασκευής του με ένα bootloader, έχοντας ως σκοπό το να μην είναι απαραίτητος εξωτερικός προγραμματιστής.

Σε κάθε πλακέτα Arduino ο μικροελεγκτής προγραμματίζεται, αλλά και περιέχει έναν ειδικό φορτωτή εκκίνησης, που μπορεί στην ενσωματωμένη μνήμη Flash της πλακέτας να απλοποιήσει κατά κύριο λόγο την διαδικασία της τοποθέτησης των προγραμμάτων. Αυτό, έχει ως αποτέλεσμα να μην κρίνεται απαραίτητο να υπάρχει κάποια εξωτερική συσκευή προγραμματισμού μικροελεγκτών AVR. Ο προγραμματισμός στις πιο πρόσφατες πλακέτες Arduino γίνεται δια της θύρας USB, την οποία διαθέτουν, ενώ παρατηρείται ότι χρειάζεται και η βοήθεια από ένα κύκλωμα μετατροπής από USB σε σειριακή όπως για παράδειγμα το FTDI FT232.

Σε γενικές γραμμές, ο προγραμματισμός όλων των πλακετών γίνεται μέσα από μια σειριακή σύνδεση RS-232, όμως σημειώνεται ποικιλία στον τρόπο υλοποίησης του κατ' αναλογία της έκδοσής του. Τέλος, ένα απλό κύκλωμα αντιστροφής χρησιμεύει στην μετατροπή μεταξύ των σημάτων των επιπέδων RS-232 και TTL.

### 6.2.1 *Shields*

Τα Arduino Shields είναι πλακέτες επέκτασης του Arduino τα οποία συνδέονται στα Header του Arduino. Τα shields μπορούν να παρέχουν έλεγχο στα motors, GPS, Ethernet, LCD εικόνας ή breadboarding (προτυποποίησης).

### 6.2.2 *Μνήμη*

Ο μικροελεγκτής Atmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων:

- Μνήμη SRAM των 2Kb η οποία αποτελεί την ωφέλιμη μνήμη που χρησιμοποιούν τα προγράμματά για να αποθηκεύουν πίνακες μεταβλητές, κ.λ.π. κατά το runtime. Όταν η παροχή ρεύματος στο Arduino σταματήσει ή αν γίνει reset αυτή η μνήμη χάνει τα δεδομένα της, ακριβώς όπως συμβαίνει και σε έναν υπολογιστή. Επειδή η μνήμη SRAM είναι πολύ περιορισμένη, αν τελειώσει, το πρόγραμμα μπορεί να δυσλειτουργεί. Έτσι εάν θέλουμε να ελεγχθεί εάν η μνήμη SPAM έχει εξαντληθεί, θα πρέπει οι σειρές ή τα σχόλια ή άλλες δομές δεδομένων του προγράμματος να λιγοστεψούν. Εάν στη συνέχεια το πρόγραμμα λειτουργεί σωστά, κατά πάσα πιθανότητα η μνήμη SRAM έχει εξαντληθεί. Έτσι για την αντιμετώπιση αυτού του προβλήματος αν υπάρχουν μεγάλοι πίνακες, τότε θα πρέπει να αντικατασταθούν με μικρότερους τύπους δεδομένων.
- Μνήμη EEPROM των 1Kb η οποία μπορεί να θεωρηθεί κάτι ανάλογο με έναν σκληρό δίσκο καθώς χρησιμοποιείται για την εγγραφή/ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματά κατά το runtime. Τέλος σε αντίθεση με την SRAM, δεν χάνει τα περιεχόμενά της με reset ή απώλεια τροφοδοσίας.
- Μνήμη Flash των 32Kb, εκ των οποίων τα 2Kb χρησιμοποιούνται από το firmware του Arduino που είναι ήδη εγκαταστημένο από τον κατασκευαστή του. Το firmware αυτό (που στην ορολογία του Arduino ονομάζεται bootloader) μας επιτρέπει την εγκατάσταση των προγραμμάτων μας στον μικροελεγκτή μέσω μιας USB θύρας, χωρίς να χρειάζεται εξωτερικός hardware programmer. Τα 30Kb της μνήμης χρησιμοποιούνται για την αποθήκευση των προγραμμάτων (αφού πρώτα



μεταγλωττιστούν στον υπολογιστή μας). Τέλος η μνήμη Flash, όπως και η EEPROM δεν χάνει τα περιεχόμενά της με reset ή απώλεια τροφοδοσίας.

### 6.2.3 Είσοδοι – Έξοδοι

Στην πλακέτα Arduino διατίθεται και σειριακό interface. Η σειριακή επικοινωνία υποστηρίζεται από τον μικροελεγκτή Atmega, η οποία προωθείται από την πλακέτα Arduino μέσω ενός ελεγκτή Serial-over-USB, με σκοπό να είναι εφικτή η σύνδεση διαμέσου USB με τον ηλεκτρονικό υπολογιστή. Η σύνδεση αυτή αποδεικνύει τη χρησιμότητά της καθώς είναι υπεύθυνη και για την μεταφορά των προγραμμάτων, τα οποία σχεδιάζονται, από τον ηλεκτρονικό υπολογιστή στην πλακέτα Arduino, αλλά και για την επικοινωνία μεταξύ της πλακέτας Arduino με τον ηλεκτρονικό υπολογιστή, που είναι αμφίδρομη, την στιγμή της εκτέλεσης του προγράμματος.

Ακόμη, δεκατέσσερα θηλυκά pin υπάρχουν στο επάνω μέρος της πλακέτας Arduino, στα οποία η αρίθμηση είναι από το μηδέν έως το δεκατρία και έχουν την δυνατότητα να λειτουργούν ως ψηφιακές εισοδοι, αλλά και έξοδοι. Η λειτουργία τους γίνεται στα 5V και το κάθε ένα είναι ικανό να διαθέτει ή φυσικά να δεχτεί το πολύ 40 mA ρεύματος. Σε ορισμένα από αυτά τα δεκατέσσερα pin σημειώνεται και μία δεύτερη λειτουργία, εκτός από την λειτουργία τους ως ψηφιακές εισοδοι και έξοδοι. Πιο συγκεκριμένα:

- Η λειτουργία των pin 0 και 1 μπορεί να είναι και ως RX και TX της σειριακής, εφόσον η σειριακή θύρα ενεργοποιείται από το πρόγραμμα. Όταν το πρόγραμμα, λόγω χάρη, στέλνει 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 δηλώνει ότι αν ενεργοποιηθεί στο πρόγραμμα το σειριακό interface, χάνονται δύο ψηφιακές εισοδοι και έξοδοι.
- Τα pin 2 και 3 μπορούν να λειτουργήσουν και σαν εξωτερικά interrupt, δηλαδή interrupt 0 και 1 αντιστοίχως. Αναλυτικότερα, μέσω του προγράμματος μπορεί να γίνει η ρύθμιση, ώστε κατ' αποκλειστικότητα η λειτουργία να είναι ως ψηφιακές εισοδοι, στις οποίες όταν λαμβάνουν χώρα ορισμένες αλλαγές, να διακόπτεται η κανονική ροή του προγράμματος και να γίνεται η εκτέλεση μια συγκεκριμένης συνάρτησης. Σε εφαρμογές, στις οποίες απαιτείται συγχρονισμός μεγάλης ακρίβειας, τα εξωτερικά interrupt κρίνονται ως ιδιαίτερος χρήσιμα.
- Τέλος, η λειτουργία των pin 3, 5, 6, 9, 10 και 11 μπορεί να είναι και ως ψευδοαναλογικές έξοδοι με το σύστημα PWM (Pulse Width Modulation), το οποίο αποτελεί το ίδιο σύστημα που υπάρχει στις μητρικές των ηλεκτρονικών υπολογιστών, ώστε να γίνεται ο έλεγχος της

ταχύτητας των ανεμιστήρων. Με πιο απλά λόγια, για να μπορεί να γίνει η σύνδεση ενός LED σε ένα από αυτά τα pin και να υπάρχει πλήρης έλεγχος της φωτεινότητας του έχοντας ανάλυση 8bit, δηλαδή διακόσιες πενήντα έξι καταστάσεις από μηδέν σβηστό έως διακόσιες πενήντα πέντε πλήρως αναμμένο, χωρίς να υπάρχει η δυνατότητα να είναι απλά αναμμένο-σβηστό, το οποίο διαθέτουν οι υπόλοιπες ψηφιακές έξοδοι. Αξίζει να σημειωθεί ότι το PWM δεν αποτελεί στην ουσία αναλογικό σύστημα, καθώς όταν τεθεί στην έξοδό του η τιμή εκατόν είκοσι επτά, δεν συνεπάγεται ότι η έξοδος θα δώσει 2.5V αντίθετα με την κανονική τιμή των 5V, αλλά ότι θα δώσει ένα παλμό κατά τον οποίο θα πραγματοποιείται μια εναλλαγή μεγάλης συχνότητας και ίσων χρονικών διαστημάτων μεταξύ των τιμών 0 και 5V.

Στο κάτω μέρος της πλακέτας Arduino βρίσκεται μια σειρά από έξι pin έχοντας την σήμανση ANALOG IN, τα οποία έχουν αρίθμηση από το μηδέν έως το πέντε. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος χρησιμοποιώντας το ADC (Analog to Digital Converter - Μετατροπέας Αναλογικού Σήματος σε Ψηφιακό), το οποίο βρίσκεται στον μικροελεγκτή ενσωματωμένο. Η τροφοδοσία ενός από αυτά, για παράδειγμα, μπορεί να γίνει με μια τάση, η οποία έχει την δυνατότητα να κυμανθεί με ένα ποτενσιόμετρο από 0V ως μια τάση αναφοράς Vref, που η ρύθμισή της είναι στα 5V, αν, φυσικά, δεν μεσολαβήσουν αλλαγές. Μέσω του προγράμματος, λοιπόν, μπορεί να «διαβαστεί» η τιμή του pin ως ένας ακέραιος αριθμός ανάλυσης 10-bit, από μηδέν, όταν η τάση που υπάρχει στο pin είναι 0V, μέχρι χίλια είκοσι τρία, όταν η τάση που υπάρχει στο pin είναι 5V. Η ρύθμιση της τάση αναφοράς γίνεται ανάμεσα στα 2 και 5V τροφοδοτώντας το pin, που έχει την σήμανση AREF και βρίσκεται στην πλευρά απέναντι της πλακέτας, εξωτερικά με αυτή την τάση.

## 6.2.4 Τροφοδοσία

Στην πλακέτα Arduino υπάρχει η δυνατότητα τροφοδοσίας του ή από τον ηλεκτρονικό υπολογιστή διαμέσου της σύνδεσης USB ή από εξωτερική τροφοδοσία, η οποία μπορεί να παραχθεί μέσα από μία υποδοχή φης των 2.1mm, έχοντας θετικό πόλο στο κέντρο, που είναι τοποθετημένη στην κάτω αριστερή γωνία της πλακέτας Arduino. Αξίζει να σημειωθεί ότι στην πλακέτα Arduino η τάση λειτουργία της είναι 5V.

Με στόχο την αποφυγή τυχόν προβλημάτων, είναι απαραίτητο η εξωτερική τροφοδοσία να κυμαίνεται από 7 ως 12V και είναι δυνατόν να πηγάξει από ένα κοινό μετασχηματιστή του εμπορίου, όπως από μπαταρίες ή κάποια άλλη πηγή DC. Ακόμη, εντοπίζεται και μία συστοιχία από έξι pin έχοντας την σήμανση POWER κοντά στα pin αναλογικής εισόδου. Η λειτουργία, λοιπόν, του καθενός είναι η ακόλουθη:

- Το πρώτο έχοντας την ένδειξη RESET, αποσκοπεί στην επανεκκίνηση της πλακέτας Arduino, όταν, φυσικά, γειωθεί (σε οποιοδήποτε από τα τρία pin, που υπάρχουν στην πλακέτα Arduino έχοντας την ένδειξη GND).
- Το δεύτερο, που έχει την ένδειξη 3.3V, είναι σε θέση να τροφοδοτεί τα εξαρτήματα με τάση 3.3V. Η συγκεκριμένη τάση παράγεται από τον ελεγκτή Serial-over-USB και δεν προκύπτει από την εξωτερική τροφοδοσία και με αυτό, η μέγιστη δυνατή ένταση, η οποία μπορεί να παραχθεί είναι μόλις 50mA.
- Το τρίτο, με ένδειξη 5V, έχει την ικανότητα να τροφοδοτεί τα εξαρτήματα με τάση 5V. Η τάση αυτή, κατ' αναλογία του τρόπου τροφοδοσίας της πλακέτας Arduino προκύπτει ή με άμεσο τρόπο από την θύρα USB, η οποία σε κάθε περίπτωση λειτουργεί στα 5V ή με έμμεσο τρόπο μέσω εξωτερικής τροφοδοσίας, εφόσον έχει πραγματοποιηθεί το πέρασμα της από ένα ρυθμιστή τάσης «φέρνοντας» την στα 5V.
- Το τέταρτο και το πέμπτο pin, έχοντας την ένδειξη GND, αποτελούν γειώσεις.
- Το έκτο και τελευταίο pin, έχει την ένδειξη Vin και ο ρόλος που έχει είναι διπλός. Από την μία, έχει τη δυνατότητα να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας της πλακέτας Arduino, πάντα συνδυαστικά με το pin γείωσης δίπλα του, στην περίπτωση που δεν είναι βολική η χρήση της υποδοχής του φισ των 2.1mm. Από την άλλη, αν μέσω του φισ υπάρχει η σύνδεση εξωτερικής τροφοδοσίας, μπορεί αυτό το pin να χρησιμοποιηθεί για την τροφοδοσία εξαρτημάτων έχοντας την πλήρη τάση της εξωτερικής τροφοδοσίας 7V με 12V, πριν λάβει μέρος το πέρασμα από τον ρυθμιστή τάσης, όπως γίνεται και έχει αναφερθεί παραπάνω με το pin των 5V.

### **6.2.5 Ενσωματωμένα LED**

Στην πλακέτα Arduino εντοπίζουμε έναν διακόπτη microswitch και τέσσερα μικροσκοπικά LED. Από την μία πλευρά, η λειτουργία του διακόπτη, η οποία έχει την σήμανση RESET είναι σε θέση να επανεκκινεί την πλακέτα Arduino, ενώ από την άλλη πλευρά, αυτή του LED, που έχει τη σήμανση POWER είναι προφανέστατη.

Τα δύο LED, τα οποία έχουν τις σημάνσεις TX και RX, αποσκοπούν στην ένδειξη της λειτουργίας της σειριακής επικοινωνίας, δηλαδή όταν ανάβουν η πλακέτα Arduino πραγματοποιεί αποστολή ή λήψη αντίστοιχα δεδομένων διαμέσου της θύρας USB. Ο έλεγχος των LED γίνεται από τον ελεγκτή Serial-over-USB και όταν η σειριακή επικοινωνία συντελείται κατ' αποκλειστικότητα δια των ψηφιακών ακροδεκτών 0 και 1, σημειώνεται ότι τα LED δεν λειτουργούν.

Τέλος, επισημαίνεται και η ύπαρξη LED με τη σήμανση L. Η πλακέτα Arduino έχει ως βασική δομή λειτουργίας την ανάθεση του να ανάβει και να σβήνει ένα LED. Αυτό για να πραγματοποιηθεί με άμεσο τρόπο άμεσα, χωρίς να γίνει κάποια σύνδεση πάνω στην πλακέτα Arduino, οι κατασκευαστές του ενσωμάτωσαν ένα LED στη πλακέτα, που συνδέθηκε στο ψηφιακό ακροδέκτη 13. Με άλλα λόγια, αν δεν υπάρξει κάποια σύνδεση πάνω στο φυσικό ακροδέκτη 13 και του ανατεθεί η τιμή HIGH μέσα σε ένα πρόγραμμα, το ενσωματωμένο LED θα ανάβει.

### 6.3 Ολοκληρωμένο περιβάλλον ανάπτυξης



```
Arduino - 0011 Alpha
File Edit Sketch Tools Help
Blink
/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;           // LED connected to digital pin 13
void setup()              // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}
void loop()               // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}

Done compiling.

Binary sketch size: 1098 bytes (of a 14336 byte maximum)
22
```

Το ολοκληρωμένο περιβάλλον ανάπτυξης της πλακέτας Arduino (IDE) μπορεί να τρέξει σε πολλές πλατφόρμες, λόγω του ότι είναι γραμμένο σε Java. Σ' αυτό περιλαμβάνονται ένας επεξεργαστής κώδικα, (δηλαδή ένας επεξεργαστής κειμένου, ο οποίος είναι εφοδιασμένος με διάφορων ειδών εύχρηστα εργαλεία) και ένας μεταγλωττιστής, δίνοντας του την δυνατότητα το πρόγραμμα να φορτώνεται με περισσότερη ευκολία από τον ηλεκτρονικό υπολογιστή στην πλακέτα Arduino διαμέσου σειριακής θύρας.

Το ολοκληρωμένο περιβάλλον ανάπτυξης έχει ως βάση του την Processing (Γλώσσα Προγραμματισμού) και χαρακτηρίζεται από την σχεδίαση του, κατά την οποία οι νέοι χρήστες και ιδιαιτέρως αυτοί που δεν είναι και τόσο εξοικειωμένοι με την ανάπτυξη του λογισμικού μπορούν εύκολα να εισαχθούν στον προγραμματισμό. Αυτή, λοιπόν, η γλώσσα προγραμματισμού προέρχεται από την Wiring, δηλαδή μια γλώσσα που παρουσιάζει πολλές ομοιότητες με την C, κατά την οποία η λειτουργικότητα που διαθέτει είναι παρόμοια για μια πλακέτα, που έχει πιο περιορισμένη σχεδίαση, το περιβάλλον την οποίας, όμως, στηρίζεται, φυσικά, στην Processing.

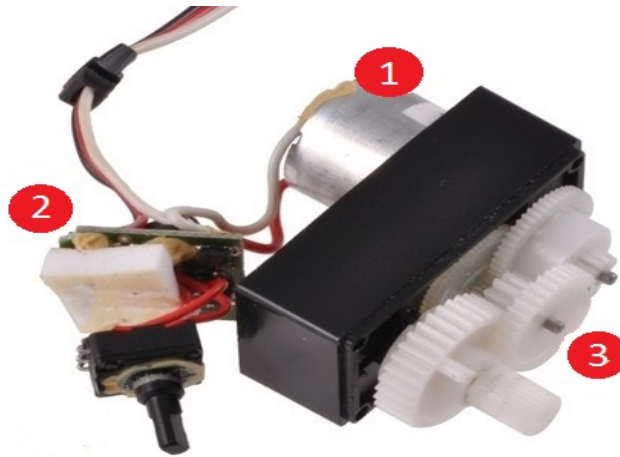
Ακόμη, το ολοκληρωμένο περιβάλλον ανάπτυξης, κατά την διαδικασία της μεταγλώττισης, είναι υπεύθυνο για την αποθήκευση του πηγαίου κώδικα του προγράμματος, που έχει συγγραφεί, σε ένα προσωρινό αρχείο και για την πρόσθεση σε αυτό το προσωρινό αρχείο όλων εκείνων που κρίνονται απαραίτητα ώστε να είναι δυνατή η μεταγλώττιση του. Τέλος, τα δεδομένα του τελικού αρχείου μεταφέρονται στην πλακέτα Arduino διαμέσου USB και για να επιτευχθεί όλη αυτή διαδικασία γίνεται χρήση του εργαλείου avrdude και του φορτωτή εκκίνησης της πλακέτας Arduino (από το ολοκληρωμένο περιβάλλον ανάπτυξης).

## 7 Κινητήρες servo

### 7.1.1 Ορισμός

Ο σερβομηχανισμός είναι μία συσκευή που αποτελείται από έναν ηλεκτροκινητήρα συνεχούς ρεύματος (1), ένα ηλεκτρονικό κύκλωμα που ελέγχει τη θέση του τελικού άξονα κίνησης (2) και ένα κιβώτιο υποβιβασμού της σχέσης μετάδοσης του κινητήρα (3).

Ο τελικός άξονας κίνησης δεν εκτελεί πλήρεις περιστροφές, αλλά περιστρέφεται μεταξύ δύο ακραίων θέσεων. Για τη λειτουργία του σερβομηχανισμού απαιτείται η παροχή της κατάλληλης ηλεκτρικής τάσης αλλά και ενός σήματος ελέγχου που καθορίζει τη θέση περιστροφής του τελικού άξονα. Τέλος, οι σερβομηχανισμοί χρησιμοποιούνται σε τηλεκατευθυνόμενα μοντέλα αλλά και σε πολλές ρομποτικές εφαρμογές.



7.1.1

### 7.1.2 Πλεονεκτήματα –Μειονεκτήματα

#### **Πλεονεκτήματα:**

- Χαμηλό κόστος.
- Μικρές διαστάσεις και εύχρηστο σχήμα: όλα τα τμήματά ενός σέρβο περιβάλλονται από ένα συμπαγές περίβλημα από το οποίο εξέρχει μόνο ο τελικός άξονας κίνησης.
- Παράγουν υψηλές τιμές ροπής.
- Δεν απαιτείται χρήση αισθητήρων και κυκλωμάτων ανάδρασης για τον προσδιορισμό της θέσης του άξονακίνησης.

#### **Μειονεκτήματα:**

- Αδυναμία εκτέλεσης πλήρους και συνεχούς περιστροφής.

## 7.2 TowerProMG996R - MG995

### TowerProMG995

Προδιαγραφές:

- Βάρος: 55 g
- Διαστάσεις: 40,7 x 19,7 x 42,9 χιλιοστά περίπου.
- Μέγιστη ροπή: 8,5 kgf · cm (4,8 V), 10 kgf · cm(6 V)
- Ταχύτητα λειτουργίας: 0.2 s / 60° (4,8 V), 0,16 s / 60° (6 V)
- Τάση λειτουργίας: 4.8 V - 7.2V
- Χρόνος αντίδρασης: 5 μs
- Σταθερά και αντικραδασμικά ρουλεμάν με σφαιρική σχεδίαση
- Εύρος θερμοκρασίας: 0 ° C - 55 ° C



7.2.

### TowerProMG996R

Προδιαγραφές:

- Βάρος: 55 g
- Διαστάσεις: 40,7 x 19,7 x 42,9 χιλιοστά περίπου.
- Μέγιστη ροπήλειτουργίας: 9,4 kgf · cm (4,8 V), 11 kgf · cm(6 V)
- Ταχύτητα λειτουργίας: 0.17 s / 60° (4,8 V), 0,14 s / 60° (6 V)
- Τάση λειτουργίας: 4.8 V - 7.2 V
- Ρεύμα Λειτουργίας 500 mA
- Μέγιστο ρεύμα λειτουργίας 2,5 A (6V)
- Χρόνος αντίδρασης: 5 μs
- Σταθερά και αντικραδασμικάρουλεμάν με σφαιρική σχεδίαση
- Εύρος θερμοκρασίας: 0 ° - 55 °C

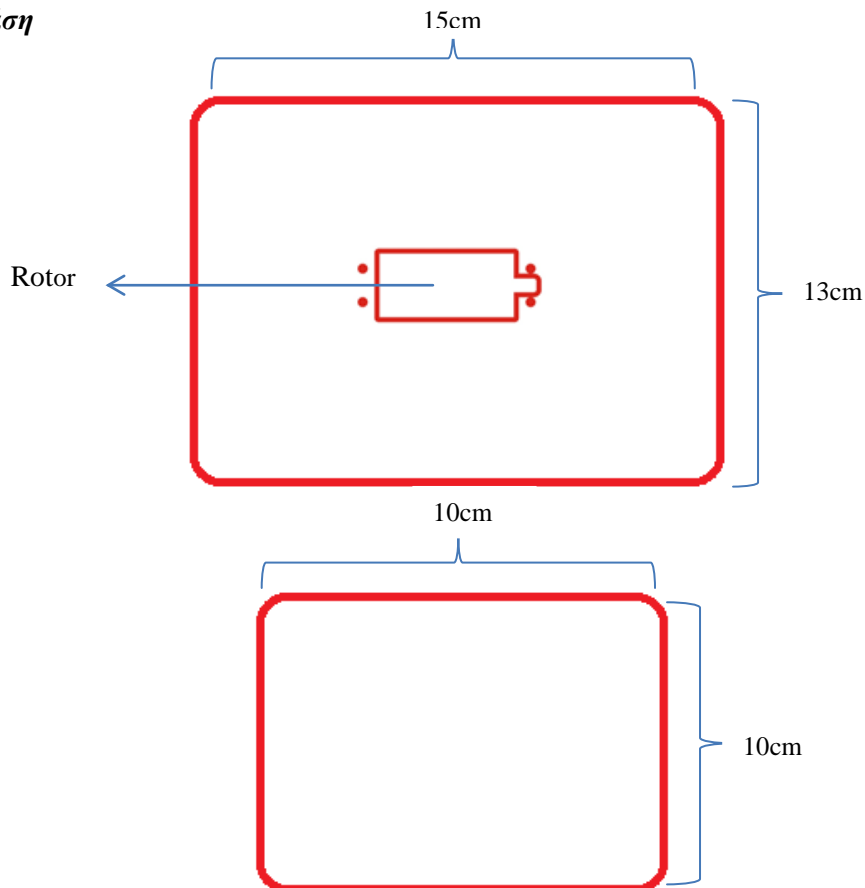
## 8 Σχεδιασμός –Μελέτη και βήματα κατασκευής του ρομποτικού βραχίονα

### 8.1 Εισαγωγή

Με σκοπό να υλοποιηθεί η κατασκευή του ρομποτικού βραχίονα έγινε διεξοδική μελέτη σε διάφορων ειδών πηγές, που περιλάμβαναν εικόνες και σχέδια από τη δημιουργία ρομποτικών βραχιόνων. Έχοντας ως βάση τις πληροφορίες που περισυλλέχθηκαν, οι οποίες, φυσικά, προσαρμόστηκαν στις ανάγκες του παρόντος εγχειρήματος, έγινε η συναρμολόγηση της κατασκευής, που θα παρουσιαστεί αναλυτικά παρακάτω.

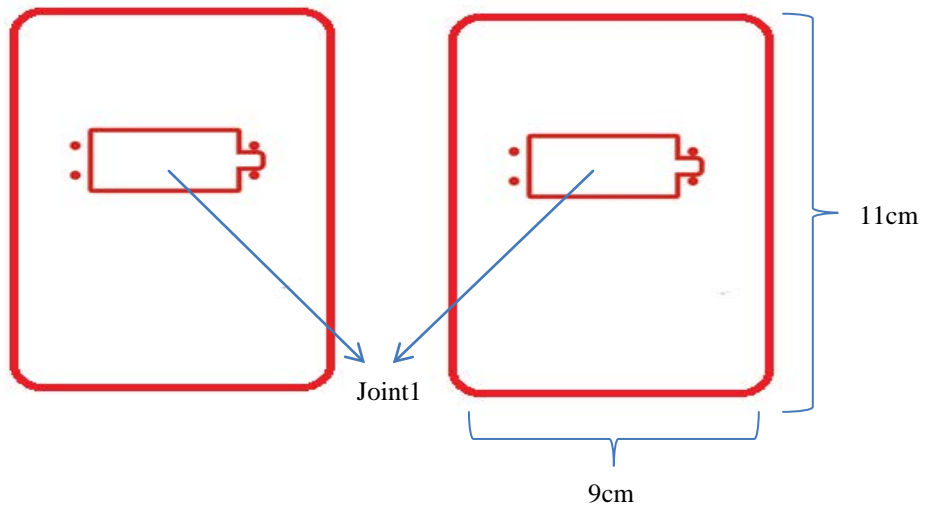
### 8.2 Σχέδια κατασκευής του βραχίονα

*Βάση*

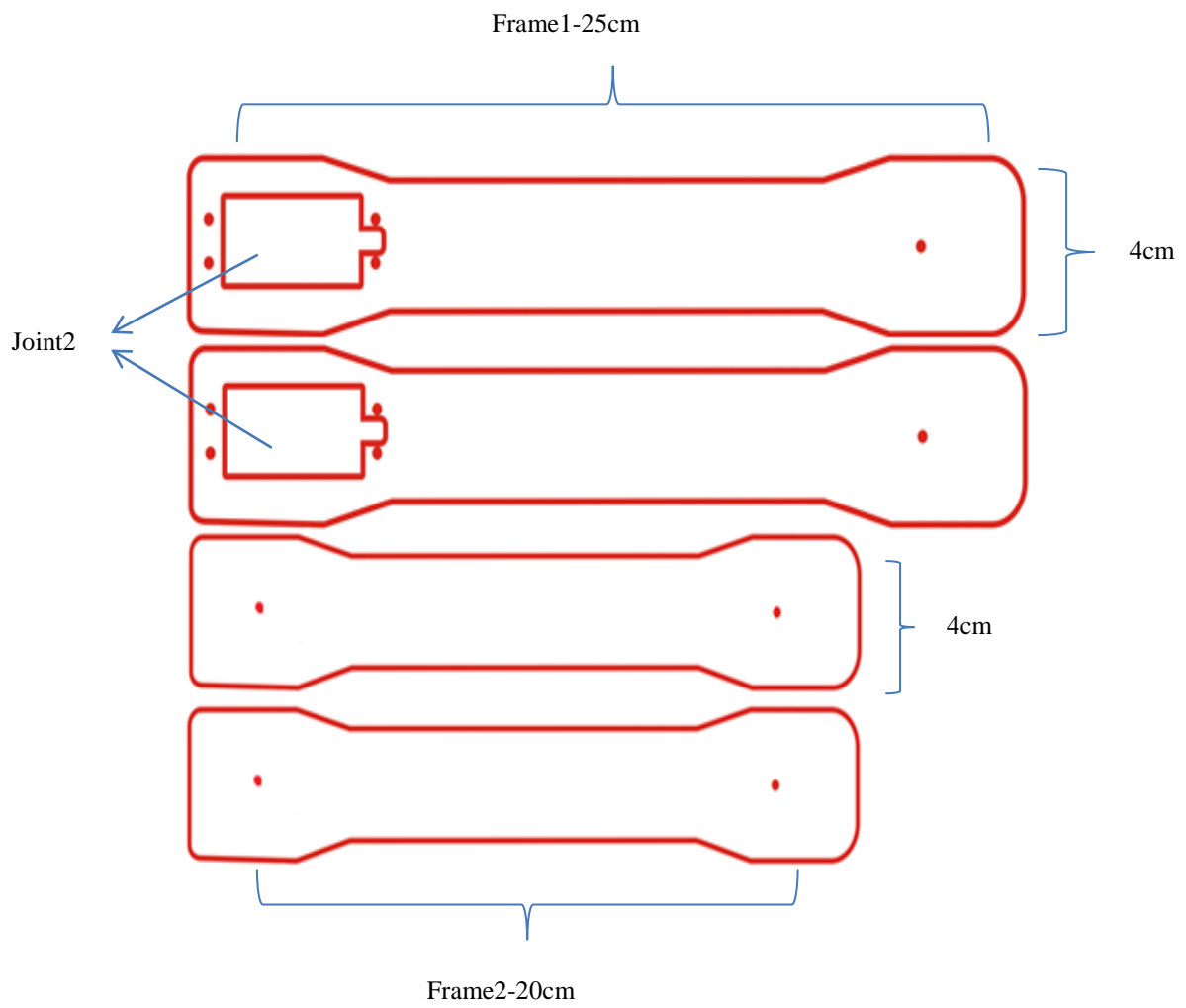




*Αρθρωση1*



*Αρθρωση2*



## Υποστηρίγματα Α, Β, Γ



## 8.3 Μελέτη και βήματα

### ➤ 1ο Βήμα

*Τοποθέτηση σερβοκινητήρα στο τμήμα rotor.*

Κατά την τοποθέτηση του σερβοκινητήρα υπήρξε δυσκολία στο πέρασμα των βιδών του. Χρησιμοποιώντας βιδολόγο δημιουργήθηκαν τρύπες μικρότερου μεγέθους από αυτές των βιδών και με την χρήση αναπτύρα θερμάνθηκαν οι βίδες και τοποθετήθηκαν στις ήδη υπάρχουσες τρύπες. Ο λόγος για τον οποίο οι τρύπες είναι μικρότερες και οι βίδες θερμάνθηκαν αποσκοπεί και στην αποφυγή πιθανού σπασίματος, αλλά και στην δημιουργία πάσων στο plexiglass.



8.3.(1)

### ➤ 2ο Βήμα

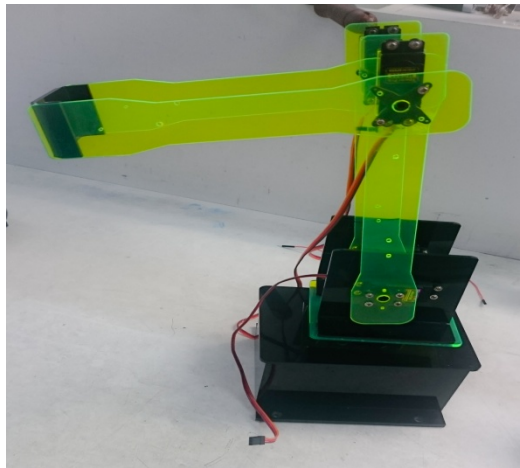
*Τοποθέτηση των σερβοκινητήρων στο τμήμα joint0.*

Λόγω του περάσματος των βιδών του σερβοκινητήρα έγινε χρήση βιδολόγου, ο οποίος άνοιξε τρύπες μικρότερου μεγέθους από αυτές των βιδών. Με τον αναπτύρα θερμάνθηκαν οι βίδες και τοποθετήθηκαν στις ήδη υπάρχουσες τρύπες και η παραπάνω διαδικασία πραγματοποιήθηκε αφενός για την αποφυγή πιθανού σπασίματος και αφετέρου για την δημιουργία πάσων στο plexiglass.

➤ **3ο Βήμα**

*Τοποθέτηση σερβοκινητήρα στο τμήμα joint1.*

Η τοποθέτηση του σερβοκινητήρα έλαβε μέρος με τη βοήθεια βιδολόγου και αναπτήρα. Αναλυτικότερα, με τον βιδολόγο δημιουργήθηκαν τρύπες μικρότερου μεγέθους από αυτές των βιδών και με τον αναπτήρα θερμάνθηκαν οι βίδες και τοποθετήθηκαν στις ήδη υπάρχουσες τρύπες. Αυτό είχε ως στόχο και την αποφυγή πιθανού σπασίματος και την δημιουργία πάσων στο plexiglass.



8.3.(2)

➤ **4ο Βήμα**

*Τοποθέτηση των A, B και Γ υποστηρίγματα στα πάνω τμήματα του βραχίονα frame1 και frame2 αντίστοιχα, με σκοπό την δημιουργία κατάλληλης σταθερής απόστασης μεταξύ τους.*

Σε αυτό το βήμα χρησιμοποιήθηκε ειδική κόλλα για plexiglass, καθώς και χάρακας για την εύρεση της μέσης απόστασης των frame1 και frame2, όπου τοποθετούνται τα A και B υποστηρίγματα αντίστοιχα.

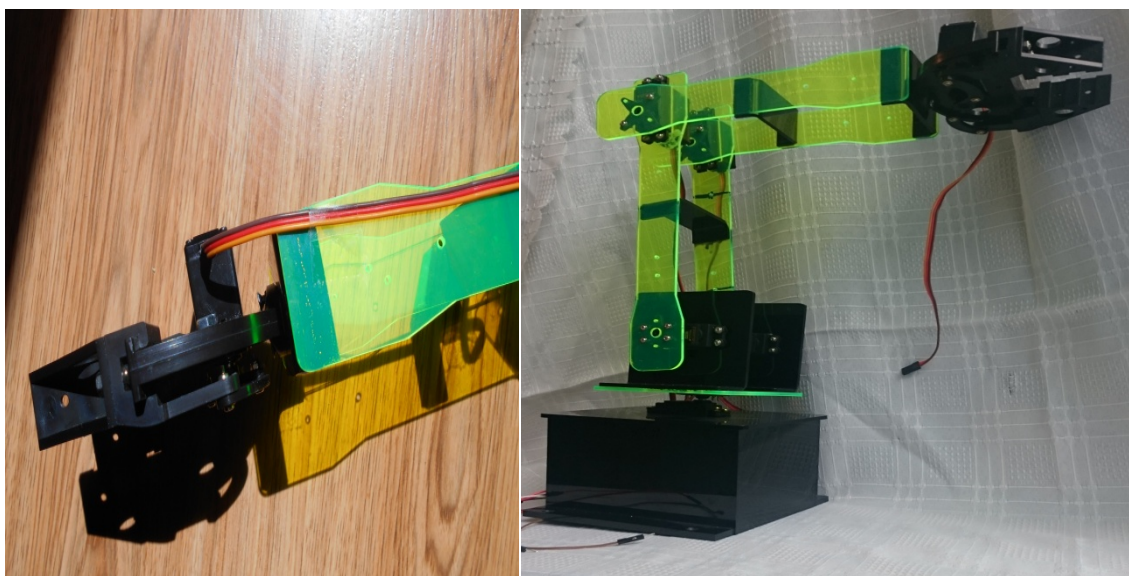


8.3.(3)

### ➤ 5ο Βήμα

*Εφαρμογή της αρπάγης πάνω στο Γ υποστήριγμα (όπου είχε τοποθετηθεί στο άκρο του frame2).*

Η τοποθέτηση της αρπάγης αποτέλεσε την ανάγκη να χρησιμοποιηθεί για ακόμη μία φορά ο βιδολόγος και ο αναπτήρας, για το πέρασμα των βιδών της. Τρύπες μικρότερου μεγέθους από αυτές των βιδών δημιουργήθηκαν από τον βιδολόγο και στη συνέχεια με τον αναπτήρα θερμάνθηκαν οι βίδες και τοποθετήθηκαν στις ήδη υπάρχουσες τρύπες. Η μέθοδος αυτή εφαρμόστηκε και για την αποφυγή πιθανού σπασίματος και για την δημιουργία πάσων στο plexiglass.



8.3.(4)

### ➤ 6ο Βήμα

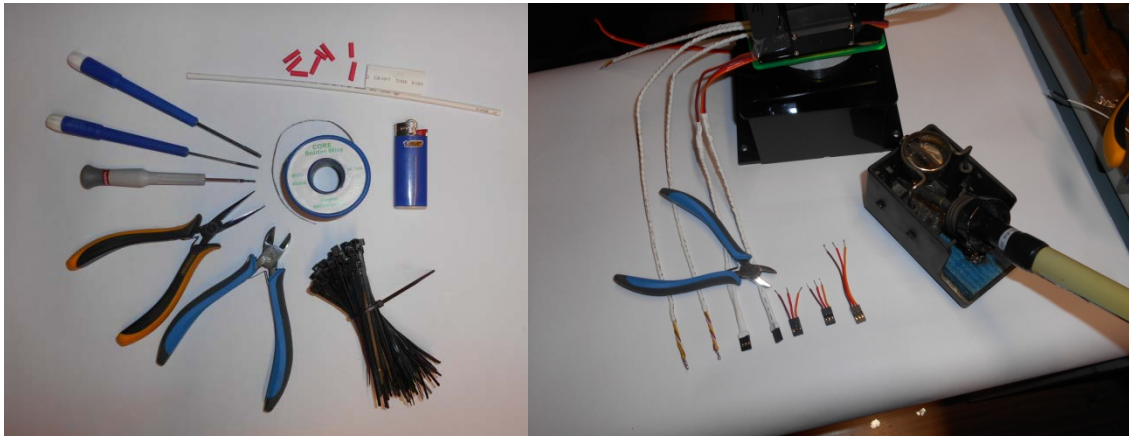
*Προέκταση των καλωδίων των σερβοκινητήρων.*

Στο βήμα αυτό παρατηρήθηκε ότι τα καλώδια των σερβοκινητήρων ήταν μικρού μήκους και η προέκτασή τους κρίθηκε απαραίτητη. Με ένα κοφτάκι πραγματοποιήθηκε η κοπή των καλωδίων, καθώς και η απογύμνωση τους. Στην συνέχεια με το καλάνι και το κολλητήρι έγινε η ένωση μεταξύ των παλιών και των νέων καλωδίων. Τέλος, με την χρήση μονωτικής ταινίας επιτεύχθηκε η μόνωση στην ένωση.

### ➤ 7ο Βήμα

*Τοποθέτηση θερμοσυστελλόμενων στα καλώδια.*

Θερμαίνοντας (μέσω ενός αναπτήρα) τα θερμοσυστελλόμενα εφαρμόζονται ακριβώς πάνω στα καλώδια των σερβοκινητήρων. Οι κύριοι λόγοι που χρησιμοποιήθηκαν τα θερμοσυστελλόμενα ήταν καλαισθητικοί, άλλα και καλύτερης μόνωσης των καλωδίων.



8.3.(5)

➤ **8ο Βήμα**

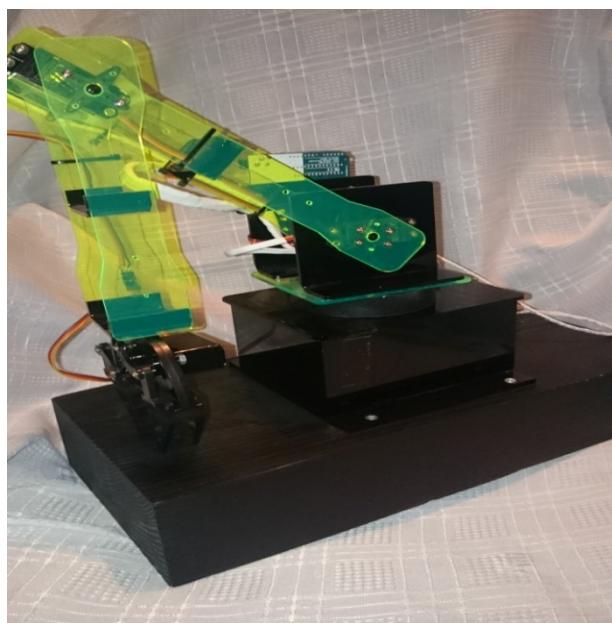
*Τρίψιμο με γυαλόχαρτο και χρωματισμός ξύλινης βάσης.*

Κατά την ολοκλήρωση του βραχίονα, με ένα κομμάτι γυαλόχαρτο smirdex finishing paper P320 αφαιρέθηκαν τα κομμάτια ξύλου από την ξύλινη βάση που προεξείχαν και με σπρέι χρωματίστηκε με ματ μαύρο χρώμα για λόγους καλαισθησίας.

➤ **9ο Βήμα**

*Τοποθέτηση του ρομποτικού βραχίονα πάνω σε ξύλινη βάση.*

Ο ρομποτικός βραχίονας, αρχικά, δεν ήταν σταθερός και έτσι η τοποθέτηση του σε μια καινούρια βάση ήταν απαραίτητη και γι' αυτό το βήμα, φυσικά, χρησιμοποιήθηκε ο βιδολόγος.



8.3.(6)

### ➤ 10ο Βήμα

Συνδεσμολογία του μικροελεγκτή *Arduino Uno* με τον βραχίονα.

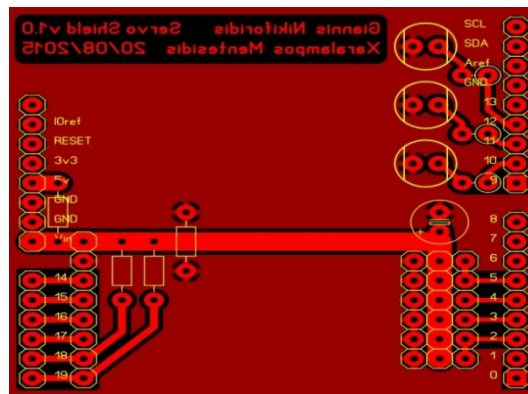
Τέλος έγινε η εφαρμογή της πλακέτας *Arduino* πάνω στο βραχίονα, στο σημείο της πρώτης άρθρωσης, ώστε να μην υπάρχει πρόβλημα με την πλέξη των καλωδίων.

## 8.4 Κατασκευή πλακέτας *shield*

### ➤ 1ο Βήμα

Σχεδιασμός των ηλεκτρονικών κυκλωμάτων της πλακέτας.

Ο σχεδιασμός έγινε μέσω ηλεκτρονικού υπολογιστή και το πρόγραμμα, που χρησιμοποιήθηκε, ονομάζεται *print layout*.

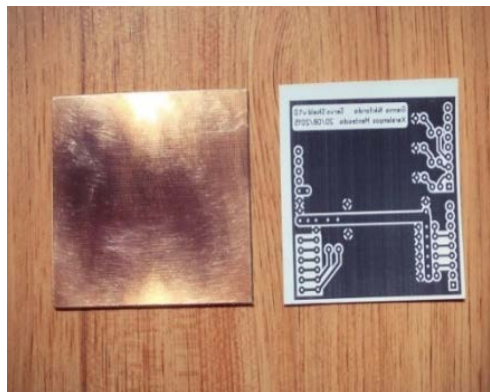


8.4.(1)

### ➤ 2ο βήμα

Εκτύπωση του ηλεκτρονικού σχεδίου.

Στο βήμα αυτό, η εκτύπωση έγινε σε φωτογραφικό χαρτί.



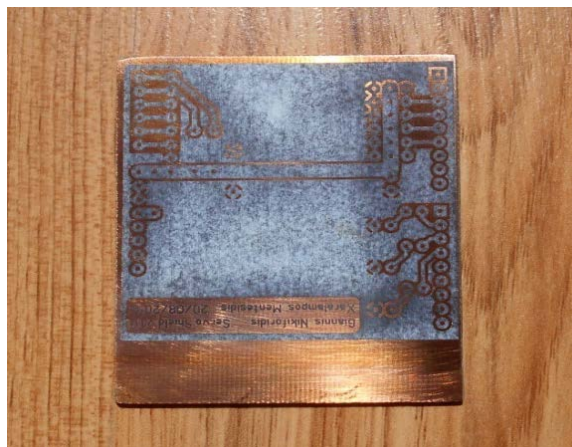
8.4.(2)



➤ **3ο βήμα**

*Μεταφορά ηλεκτρονικού σχεδίου πάνω σε χαλκό.*

Αρχικά, το σχέδιο τοποθετήθηκε πάνω στον χαλκό και με τη χρήση ηλεκτρικού σιδήρου θερμάνθηκε το φωτογραφικό χαρτί και προσκολλήθηκε στον χαλκό.



8.4.(3)

➤ **4ο βήμα**

*Αποχάλκωση με τριχλωριούχο σίδηρο.*

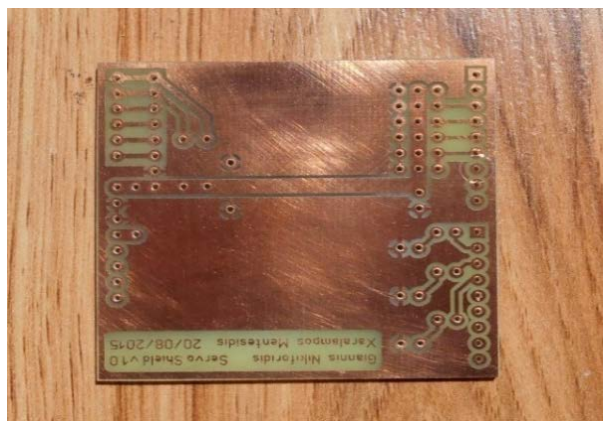
Ο χαλκός επιχειρείται να αφαιρεθεί ολοκληρωτικά από την πλακέτα, εκτός από το τμήμα που υπάρχει το τόνερ στην πλακέτα, το οποίο φέρει πάνω του το κύκλωμα. Ο χρόνος που απαιτήθηκε γι' αυτήν τη διαδικασία ήταν περίπου 25 λεπτά.



8.4.(4)

➤ **5ο βήμα**

Μετά από την αφαίρεση του τόνερ από την πλακέτα με την χρήση ασετόν.

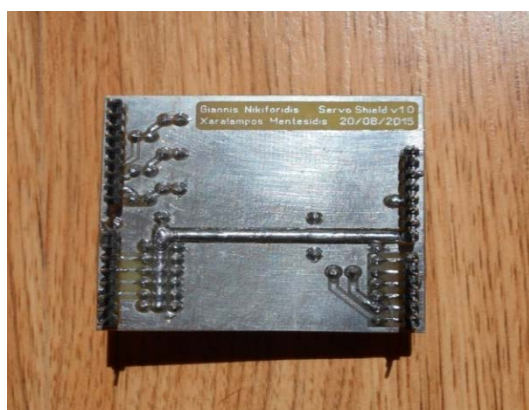


8.4.(5)

➤ **6ο βήμα**

Πλακέτα γανωμένη με καλάι.

Μετά την αποχάλκωση η πλακέτα γανώθηκε με καλάι για να αντέχουν οι γραμμές περισσότερο ρεύμα.



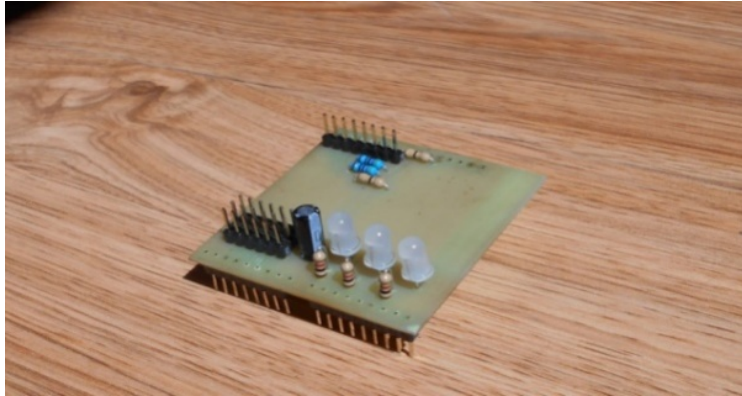
8.4.(6)

➤ **7ο βήμα**

Τελική διαμόρφωση πλακέτας.

Η πλακέτα αποτελείται από τα εξής: πυκνωτή με τάση λειτουργίας 16V και χωρητικότητα 47  $\mu\text{f}$ , πέντε αντιστάσεις 4,7 K $\Omega$ , δυο αντιστάσεις 0 $\Omega$ , τρία led, έξι πιν, στα οποία τοποθετούνται τα καλώδια των σερβοκινητήρων και οχτώ πιν, που όπως θα δούμε στην συνέχεια τοποθετείτε το τηλεχειριστήριο.

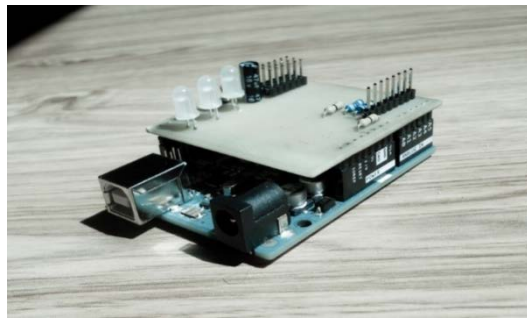




8.4.(7)

➤ **8ο Βήμα**

*Εφαρμογή της πλακέτας πάνω στην arduino.*



8.4.(8)

## 8.5 Κατασκευή τηλεχειριστηρίου

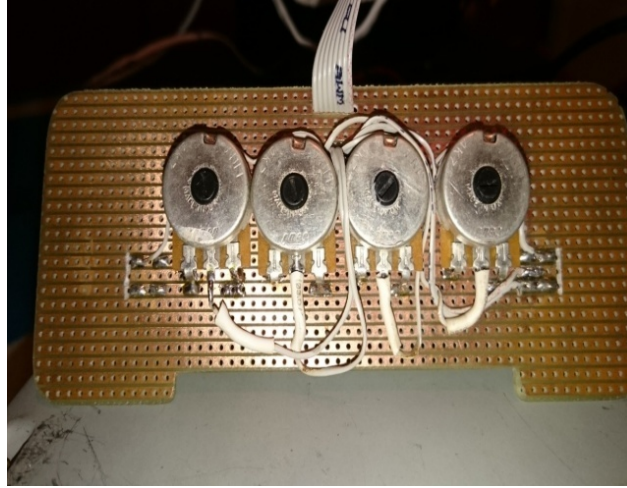
➤ **1ο Βήμα**

*Διαμόρφωση κατάλληλου σχήματος του τηλεχειριστηρίου.*

Ο κύριος σκελετός του τηλεχειριστηρίου είναι μια διατρητή πλακέτα και για την σχεδίαση του κρίθηκε απαραίτητη η χρήση χάρακα και μολυβιού.

➤ **2ο Βήμα**

*Εισαγωγή ποτενσιομέτρων και κουμπιών.*



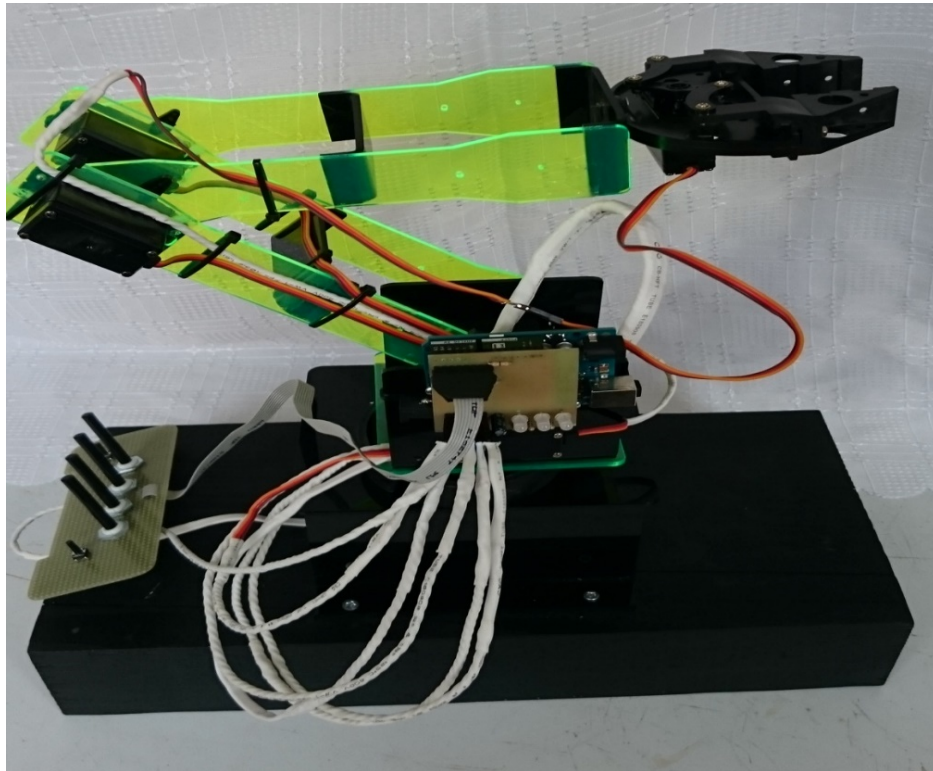
8.5.(1)

➤ **3ο Βήμα**

Σύνδεση με την πλακέτα *shield*.



8.5.(2)



*Ολοκληρωμένη κατασκευή 8.5.(3)*

## 9 Πρόγραμμα

```
#include <Servo.h> //Με την εντολή "#include" προσθέτουμε το αρχείο (την βιβλιοθήκη)
"Servo.h" στο πρόγραμμα μας
Servo SRVtrr; //Ορίζουμε τον σερβοκινητήρα "SRVtrr"
Servo SRVjnt0; //Ορίζουμε τον σερβοκινητήρα "SRVjnt0"
Servo SRVjnt1; //Ορίζουμε τον σερβοκινητήρα "SRVjnt1"
Servo SRVgrbr; //Ορίζουμε τον σερβοκινητήρα "SRVgrbr"
word rotor=137; //Ορίζουμε την μεταβλητή "rotor" (περιστροφέας)
word joint0=77; //Ορίζουμε την μεταβλητή "joint0" (άρθρωση0)
word joint1=121; //Ορίζουμε την μεταβλητή "joint1" (άρθρωση1)
word grabber=120; //Ορίζουμε την μεταβλητή "grabber" (αρπάγη)
word rotorRec; //Ορίζουμε την μεταβλητή "rotorRec"
word joint0Rec; //Ορίζουμε την μεταβλητή "joint0Rec"
word joint1Rec; //Ορίζουμε την μεταβλητή "joint1Rec"
word grabberRec; //Ορίζουμε την μεταβλητή "grabberRec"
word rotorRec2; //Ορίζουμε την μεταβλητή "rotorRec2"
word joint0Rec2; //Ορίζουμε την μεταβλητή "joint0Rec2"
word joint1Rec2; //Ορίζουμε την μεταβλητή "joint1Rec2"
word grabberRec2; //Ορίζουμε την μεταβλητή "grabberRec2"
int LED0= 10; //Ορίζουμε την μεταβλητή "LED0"
int LED1= 11; //Ορίζουμε την μεταβλητή "LED1"
int LED2= 12; //Ορίζουμε την μεταβλητή "LED2"
int Butt0= 18; //Ορίζουμε την μεταβλητή "Butt0"
int Butt1= 19; //Ορίζουμε την μεταβλητή "Butt1"
int ButtV0; //Ορίζουμε την μεταβλητή "ButtV0"
int ButtV1; //Ορίζουμε την μεταβλητή "ButtV1"
word Cnt0; //Ορίζουμε την μεταβλητή "Cnt0"
word Cnt1; //Ορίζουμε την μεταβλητή "Cnt1"
word Cnt2; //Ορίζουμε την μεταβλητή "Cnt2"
word Cnt3; //Ορίζουμε την μεταβλητή "Cnt3"
int fDemo0; //Ορίζουμε την μεταβλητή "fDemo0"
int fDemo1; //Ορίζουμε την μεταβλητή "fDemo1"
int fDemo2; //Ορίζουμε την μεταβλητή "fDemo2"
int fManual; //Ορίζουμε την μεταβλητή "fManual"
```

```

void setup() //Ορίζουμε την ρουτίνα Αρχικοποίησης "setup"
{
  SRVjnt0.attach(2); //Ορίζουμε το pin(2) ως ψηφιακή έξοδο προς τον σερβοκινητήρα joint0
  SRVrtr.attach(3); //Ορίζουμε το pin(3) ως ψηφιακή έξοδο προς τον σερβοκινητήρα rotor
  SRVgrbr.attach(4); //Ορίζουμε το pin(4) ως ψηφιακή έξοδο προς τον σερβοκινητήρα grabber
  SRVjnt1.attach(5); //Ορίζουμε το pin(5) ως ψηφιακή έξοδο προς τον σερβοκινητήρα joint1
  SetServo(); //Καλούμε την υπορουτίνα "SetServo"
  pinMode(Butt0, INPUT); //Ορίζουμε το Butt0= pin(18) ως ψηφιακή είσοδο
  pinMode(Butt1, INPUT); //Ορίζουμε το Butt1= pin(19) ως ψηφιακή είσοδο
  pinMode(LED0, OUTPUT); //Ορίζουμε το LED0= pin(10) ως ψηφιακή έξοδο
  pinMode(LED1, OUTPUT); //Ορίζουμε το LED1= pin(11) ως ψηφιακή έξοδο
  pinMode(LED2, OUTPUT); //Ορίζουμε το LED2= pin(12) ως ψηφιακή έξοδο
  rotorRec = analogRead(0); //Διαβάζουμε την τάση στα άκρα του ποτενσιόμετρου και
αποθηκεύουμε την τιμή του στην μεταβλητή "rotorRec"
  rotorRec = map(rotorRec, 0, 1023, 0, 180); //Χρησιμοποιώντας την εντολή "map"
αντιστοιχούμε την κλίμακα των αναλογικών μετρήσεων των 10bit στην κλίμακα του
σερβοκινητήρα 0-180
  joint0Rec = analogRead(1); //Διαβάζουμε την τάση στα άκρα του ποτενσιόμετρου και
αποθηκεύουμε την τιμή του στην μεταβλητή "joint0Rec"
  joint0Rec = map(joint0Rec, 0, 1023, 0, 180); //Χρησιμοποιώντας την εντολή "map"
αντιστοιχούμε την κλίμακα των αναλογικών μετρήσεων των 10bit στην κλίμακα του
σερβοκινητήρα 0-180
  joint1Rec = analogRead(2); //Διαβάζουμε την τάση στα άκρα του ποτενσιόμετρου και
αποθηκεύουμε την τιμή του στην μεταβλητή "joint1Rec"
  joint1Rec = map(joint1Rec, 0, 1023, 0, 180); //Χρησιμοποιώντας την εντολή "map"
αντιστοιχούμε την κλίμακα των αναλογικών μετρήσεων των 10bit στην κλίμακα του
σερβοκινητήρα 0-180
  grabberRec = analogRead(3); //Διαβάζουμε την τάση στα άκρα του ποτενσιόμετρου και
αποθηκεύουμε την τιμή του στην μεταβλητή "grabberRec"
  grabberRec = map(grabberRec, 0, 1023, 120, 45); //Χρησιμοποιώντας την εντολή "map"
αντιστοιχούμε την κλίμακα των αναλογικών μετρήσεων των 10bit στην κλίμακα του
σερβοκινητήρα 0-180
  LedBlink(); //Καλούμε την υπορουτίνα "LedBlink"
//Αρχικοποίηση Μεταβλητών
  rotor=137; joint0=77; joint1=121; grabber=120; //Ανάθεση τιμών στις μεταβλητές

```

```

digitalWrite(LED0,HIGH); digitalWrite(LED1,HIGH); digitalWrite(LED2,HIGH); //Ανάβουμε
LED0, LED1 και το LED2
Cnt0= 0; //Ανάθεση τιμής σε μεταβλητή
Cnt1= 0; //Ανάθεση τιμής σε μεταβλητή
Cnt2= 0; //Ανάθεση τιμής σε μεταβλητή
ButtV0= 1; //Ανάθεση τιμής σε μεταβλητή
ButtV1= 1; //Ανάθεση τιμής σε μεταβλητή
fDemo0= 1; //Ανάθεση τιμής σε μεταβλητή
fDemo1= 0; //Ανάθεση τιμής σε μεταβλητή
fDemo2= 0; //Ανάθεση τιμής σε μεταβλητή
fManual= 0; //Ανάθεση τιμής σε μεταβλητή
}
void loop(){//Ορίζουμε την ρουτίνα επανάληψης του κυρίως προγράμματος
//Παρακάτω χρησιμοποιούμε 4 επαναλήψεις while οι οποίες χρησιμοποιούνται για την
μανδάλωση του κάθε
//επιλεγόμενου μενού. Αρχικά επιλέγεται το Demo0 και στην πορεία ο χρήστης πλοηγείται στο
μενού.
while (fDemo0>0){ //Όσο η μεταβλητή fDemo0>0 τότε επανέλαβε
Demo0(); //Καλούμε την υπορουτίνα "Demo0"
}
while (fDemo1>0){ //Όσο η μεταβλητή fDemo1>0 τότε επανέλαβε
Demo1(); } //Καλούμε την υπορουτίνα "Demo1"
while (fDemo2>0){ //Όσο η μεταβλητή fDemo2>0 τότε επανέλαβε
Demo2(); //Καλούμε την υπορουτίνα "Demo2"
if (fManual==1) GoPotVal(); //Αν η μεταβλητή "fManual" πάρει την τιμή "1" τότε καλούμε
την υπορουτίνα "GoPotVal"
while (fManual>0){ //Όσο η μεταβλητή fManual>0 τότε επανέλαβε
Manual(); } //Καλούμε την υπορουτίνα "Manual"
}
// Υπορουτίνες
void LedBlink(){//Αναβοσβήνει τα LED 3εις φορές με διάρκεια 200ms
for (Cnt1=0; Cnt1<3; Cnt1++) {
digitalWrite(LED0,HIGH); digitalWrite(LED1,HIGH); digitalWrite(LED2,HIGH);
//Ανάβουμε τα LED0, LED1, LED2
delay(200); //Καθυστέρηση εκτέλεσης του προγράμματος 200ms
}
}

```



```

digitalWrite(LED0,LOW); digitalWrite(LED1,LOW); digitalWrite(LED2,LOW); //Σβήνουμε
τα LED0, LED1, LED2
delay(200); //Καθυστέρηση εκτέλεσης του προγράμματος 200ms
}
}
void Demo0(){//Είναι η στάση αρχικοποίησης την συσκευής
  rotor=137; joint0=77; joint1=121; grabber=120; //Ανάθεση τιμής σε μεταβλητή
  SetServo(); //Καλούμε την υπορουτίνα "SetServo"
  BreakDemo(); //Καλούμε την υπορουτίνα "BreakDemo"
  if (fManual==1) GoPotVal(); //Αν η μεταβλητή fManual πάρει την τιμή 1 τότε καλούμε την
υπορουτίνα "GoPotVal"
}
void Demo1(){//Η συσκευή εκτελεί μια πράξη αρπαγής ενός αντικειμένου μετακίνησης του και
επαναφοράς του στο ίδιο σημείο
  GoBot(); //Καλούμε την υπορουτίνα "GoBot"
  Grab(); //Καλούμε την υπορουτίνα "Grab"
  GoUp(); //Καλούμε την υπορουτίνα "GoUp"
  TurnRight(); //Καλούμε την υπορουτίνα " TurnRight "
  GoDown(); //Καλούμε την υπορουτίνα " GoDown "
  Release(); //Καλούμε την υπορουτίνα " Release "
  GoUp(); //Καλούμε την υπορουτίνα " GoUp "
  GoDown(); //Καλούμε την υπορουτίνα " GoDown "
  Grab(); //Καλούμε την υπορουτίνα " Grab "
  GoUp(); //Καλούμε την υπορουτίνα " GoUp "
  TurnLeft(); //Καλούμε την υπορουτίνα " TurnLeft "
  GoDown(); //Καλούμε την υπορουτίνα " GoDown "
  Release(); //Καλούμε την υπορουτίνα " Release "
  GoUp(); //Καλούμε την υπορουτίνα " GoUp "
  if (fManual==1) GoPotVal(); //Αν η μεταβλητή fManual πάρει την τιμή 1 τότε καλούμε την
υπορουτίνα "GoPotVal"
}
void Demo2(){//Η συσκευή εκτελεί μια παρουσίαση των ορίων των τιμών του κάθε
σερβοκινητήρα
  MinGrab(); //Καλούμε την υπορουτίνα "MinGrab"
  MaxGrab(); //Καλούμε την υπορουτίνα "MaxGrab"

```

```

MinRot(); //Καλούμε την υπορουτίνα "MinRot"
MaxRot(); //Καλούμε την υπορουτίνα "MaxRot"
MinJoint0(); //Καλούμε την υπορουτίνα "MinJoint0"
MaxJoint0(); //Καλούμε την υπορουτίνα "MaxJoint0"
GoTop(); //Καλούμε την υπορουτίνα "GoTop"
MinJoint1(); //Καλούμε την υπορουτίνα "MinJoint1"
MaxJoint1(); //Καλούμε την υπορουτίνα "MaxJoint1"
if (fManual==1) GoPotVal();//Αν η μεταβλητή fManual πάρει την τιμή 1 τότε καλούμε την
υπορουτίνα "GoPotVal"
}
void Manual(){ //Χειροκίνητη λειτουργία με χρήση ποτενσιόμετρων.
  rotor = analogRead(0); //Διαβάζουμε την τάση στα άκρα του ποτενσιόμετρου και
αποθηκεύουμε την τιμή του στην μεταβλητή "rotor"
  rotor = map(rotor, 0, 1023, 0, 180); //Χρησιμοποιώντας την εντολή "map" αντιστοιχούμε την
κλίμακα των αναλογικών μετρήσεων των 10bit στην κλίμακα του σερβοκινητήρα 0-180
  joint0 = analogRead(1); //Διαβάζουμε την τάση στα άκρα του ποτενσιόμετρου και
αποθηκεύουμε την τιμή του στην μεταβλητή "joint0"
  joint0 = map(joint0, 0, 1023, 0, 180); //Χρησιμοποιώντας την εντολή "map" αντιστοιχούμε την
κλίμακα των αναλογικών μετρήσεων των 10bit στην κλίμακα του σερβοκινητήρα 0-180
  joint1 = analogRead(2); //Διαβάζουμε την τάση στα άκρα του ποτενσιόμετρου και
αποθηκεύουμε την τιμή του στην μεταβλητή "joint1"
  joint1 = map(joint1, 0, 1023, 0, 180); //Χρησιμοποιώντας την εντολή "map" αντιστοιχούμε την
κλίμακα των αναλογικών μετρήσεων των 10bit στην κλίμακα του σερβοκινητήρα 0-180
  grabber = analogRead(3); //Διαβάζουμε την τάση στα άκρα του ποτενσιόμετρου και
αποθηκεύουμε την τιμή του στην μεταβλητή "grabber"
  grabber = map(grabber, 0, 1023, 120, 45); //Χρησιμοποιώντας την εντολή "map"
αντιστοιχούμε την κλίμακα των αναλογικών μετρήσεων των 10bit στην κλίμακα του
σερβοκινητήρα 0-180
  SetServo(); //Καλούμε την υπορουτίνα "SetServo"
  ButtRead(); //Καλούμε την υπορουτίνα "ButtRead"
  if (ButtV0==0 || ButtV1==0)//Αν η μεταβλητή ButtV0 ή η μεταβλητή ButtV1 πάρει την τιμή 0
τότε καλούμε την υπορουτίνα "ButtCase"
  ButtCase(); //Καλούμε την υπορουτίνα "ButtCase"
}
void ButtRead(){ //Διαβάζει την κατάσταση των κουμπιών "Butt0, Butt1" και την αποθηκεύει

```



σε μεταβλητές.

```
ButtV0= digitalRead(Butt0);//Διαβάζουμε την τιμή του Κουμπιού0  
ButtV1= digitalRead(Butt1);//Διαβάζουμε την τιμή του Κουμπιού1  
}
```

```
void BreakDemo(){ //Διαβάζει τα κουμπιά της συσκευής και αν έστω ένα πατηθεί ορίζει τα flag  
έτσι ώστε να μπει το πρόγραμμα στο μενού Manual
```

```
ButtRead(); //Καλούμε την υπορουτίνα "ButtRead"
```

```
if (ButtV0==0 || ButtV1==0){
```

```
    delay(100); //Καθυστέρηση εκτέλεσης του προγράμματος 100ms
```

```
    fDemo0= 0; //Ανάθεση τιμής σε μεταβλητή
```

```
    fDemo1= 0; //Ανάθεση τιμής σε μεταβλητή
```

```
    fDemo2= 0; //Ανάθεση τιμής σε μεταβλητή
```

```
    fManual=1; //Ανάθεση τιμής σε μεταβλητή
```

```
    ButtV0=1; //Ανάθεση τιμής σε μεταβλητή
```

```
    ButtV1=1; //Ανάθεση τιμής σε μεταβλητή
```

```
digitalWrite(LED0,HIGH); digitalWrite(LED1,LOW); digitalWrite(LED2,LOW); //Ανάβουμε  
το LED0 και σβήνουμε τα LED1, LED2
```

```
}
```

```
}
```

```
void ButtCase(){//Με βάση τον συνδυασμό των κουμπιών που πατήθηκαν ορίζει τα flags και  
κάνει αρχικοποίηση στις μεταβλητές για να εισέλθει ομαλά το πρόγραμμα στο αντίστοιχο  
μενού
```

```
ButtRead(); //Καλούμε την υπορουτίνα "ButtRead"
```

```
delay(100); //Καθυστέρηση εκτέλεσης του προγράμματος 100ms
```

```
ButtRead(); //Καλούμε την υπορουτίνα "ButtRead"
```

```
delay(100); //Καθυστέρηση εκτέλεσης του προγράμματος 100ms
```

```
if (ButtV0==0 && ButtV1==0){ //Routina Demo0
```

```
    fDemo0= 1; //Ανάθεση τιμής σε μεταβλητή
```

```
    fDemo1= 0; //Ανάθεση τιμής σε μεταβλητή
```

```
    fDemo2= 0; //Ανάθεση τιμής σε μεταβλητή
```

```
    fManual=0; //Ανάθεση τιμής σε μεταβλητή
```

```
    ButtV0=1; //Ανάθεση τιμής σε μεταβλητή
```

```
ButtV1=1; //Ανάθεση τιμής σε μεταβλητή
```

```
    digitalWrite(LED0,HIGH); digitalWrite(LED1,HIGH);
```

```
digitalWrite(LED2,HIGH); //Ανάβουμε τα LED0, LED1, LED2
```

```

rotorRec= rotor; //Ανάθεση τιμής σε μεταβλητή
  joint0Rec= joint0; //Ανάθεση τιμής σε μεταβλητή
  joint1Rec= joint1; //Ανάθεση τιμής σε μεταβλητή
  grabberRec= grabber; //Ανάθεση τιμής σε μεταβλητή
}
if (ButtV0==0){//Routina Demo1
  fDemo0= 0; //Ανάθεση τιμής σε μεταβλητή
  fDemo1= 1; //Ανάθεση τιμής σε μεταβλητή
  fDemo2= 0; //Ανάθεση τιμής σε μεταβλητή
  fManual=0; //Ανάθεση τιμής σε μεταβλητή
  ButtV0=1; //Ανάθεση τιμής σε μεταβλητή
  ButtV1=1; //Ανάθεση τιμής σε μεταβλητή
  digitalWrite(LED0,LOW); digitalWrite(LED1,HIGH);
digitalWrite(LED2,LOW); //Ανάβουμε το LED1 και σβήνουμε τα LED0, LED2
  rotorRec= rotor; //Ανάθεση τιμής σε μεταβλητή
  joint0Rec= joint0; //Ανάθεση τιμής σε μεταβλητή
  joint1Rec= joint1; //Ανάθεση τιμής σε μεταβλητή
  grabberRec= grabber; //Ανάθεση τιμής σε μεταβλητή
}
if (ButtV1==0){//Routina Demo2
  fDemo0= 0; //Ανάθεση τιμής σε μεταβλητή
  fDemo1= 0; //Ανάθεση τιμής σε μεταβλητή
  fDemo2= 1; //Ανάθεση τιμής σε μεταβλητή
  fManual=0; //Ανάθεση τιμής σε μεταβλητή
  ButtV0=1; //Ανάθεση τιμής σε μεταβλητή
  ButtV1=1; //Ανάθεση τιμής σε μεταβλητή
  digitalWrite(LED0,LOW); digitalWrite(LED1,LOW);
digitalWrite(LED2,HIGH); //Ανάβουμε το LED2 και σβήνουμε τα LED0, LED1
  rotorRec= rotor; //Ανάθεση τιμής σε μεταβλητή
  joint0Rec= joint0; //Ανάθεση τιμής σε μεταβλητή
  joint1Rec= joint1; //Ανάθεση τιμής σε μεταβλητή
  grabberRec= grabber; //Ανάθεση τιμής σε μεταβλητή
}
ButtV0=1; //Ανάθεση τιμής σε μεταβλητή
ButtV1=1; //Ανάθεση τιμής σε μεταβλητή

```

```

}
void SetServo(){ //Ορίζει την θέση του κάθε σερβοκινητήρα
    SRVrtr.write(rotor); //Ορίζουμε την θέση που πρέπει να πάει ο σερβοκινητήρας "rotor"
    SRVjnt0.write(joint0); //Ορίζουμε την θέση που πρέπει να πάει ο σερβοκινητήρας "joint0"
    SRVjnt1.write(joint1); //Ορίζουμε την θέση που πρέπει να πάει ο σερβοκινητήρας "joint1"
    SRVgrbr.write(grabber); //Ορίζουμε την θέση που πρέπει να πάει ο σερβοκινητήρας "grabber"
}
void GoDown(){ //Ο βραχίονας μετακινείται στην κάτω θέση
    rotorRec2=rotor; joint0Rec2=joint0; joint1Rec2=90; grabberRec2=grabber;
    GoSmooth(); //Καλούμε την υπορουτίνα "GoSmooth"
}
void GoUp(){ //Ο βραχίονας μετακινείται στην πάνω θέση
    rotorRec2=rotor; joint0Rec2=joint0; joint1Rec2=140; grabberRec2=grabber;
    GoSmooth(); //Καλούμε την υπορουτίνα "GoSmooth"
}
void Grab(){ //Οαρπαγέαςκλείνει
    rotorRec2=rotor; joint0Rec2=joint0; joint1Rec2=joint1; grabberRec2=45;
    GoSmooth(); //Καλούμε την υπορουτίνα "GoSmooth"
}
void Release(){ //Οαρπαγέαςανοίγει
    rotorRec2=rotor; joint0Rec2=joint0; joint1Rec2=joint1; grabberRec2=120;
    GoSmooth(); //Καλούμε την υπορουτίνα "GoSmooth"
}
void TurnRight(){ //Ο βραχίονας μετακινείται δεξιά
    rotorRec2=70; joint0Rec2=joint0; joint1Rec2=joint1; grabberRec2=grabber;
    GoSmooth(); //Καλούμε την υπορουτίνα "GoSmooth"
}
void TurnLeft(){ //Ο βραχίονας μετακινείται αριστερά
    rotorRec2=137; joint0Rec2=joint0; joint1Rec2=joint1; grabberRec2=grabber;
    GoSmooth(); //Καλούμε την υπορουτίνα "GoSmooth"
}
void GoBot(){ //Ο βραχίονας μετακινείται στην θέση κάτω2
    rotorRec2=137; joint0Rec2=156; joint1Rec2=90; grabberRec2=120;
    GoSmooth(); //Καλούμε την υπορουτίνα "GoSmooth"
}
}

```

```

void GoTop(){ //Ο βραχίονας μετακινείται στην θέση πάνω2
    rotorRec2=137; joint0Rec2=79; joint1Rec2=122; grabberRec2=120;
    GoSmooth(); //Καλούμε την υπορουτίνα "GoSmooth"
}

void MinRot(){ //Ο βραχίονας πραγματοποιεί περιστροφή στο κάτω όριο του σερβοκινητήρα
    rotorRec2=10; joint0Rec2=79; joint1Rec2=122; grabberRec2=120;
    GoSmooth(); //Καλούμε την υπορουτίνα "GoSmooth"
}

void MaxRot(){ //Ο βραχίονας πραγματοποιεί περιστροφή στο πάνω όριο του σερβοκινητήρα
    rotorRec2=170; joint0Rec2=79; joint1Rec2=122; grabberRec2=120;
    GoSmooth(); //Καλούμε την υπορουτίνα "GoSmooth"
}

void MinJoint0(){ //Ο βραχίονας πραγματοποιεί κίνηση στην άρθρωση joint0 στο κάτω όριο
    rotorRec2=137; joint0Rec2=14; joint1Rec2=122; grabberRec2=120;
    GoSmooth(); //Καλούμε την υπορουτίνα "GoSmooth"
}

void MaxJoint0(){ //Ο βραχίονας πραγματοποιεί κίνηση στην άρθρωση joint0 στο πάνω όριο
    rotorRec2=137; joint0Rec2=155; joint1Rec2=122; grabberRec2=120;
    GoSmooth(); //Καλούμε την υπορουτίνα "GoSmooth"
}

void MinJoint1(){ //Ο βραχίονας πραγματοποιεί κίνηση στην άρθρωση joint1 στο κάτω όριο
    rotorRec2=137; joint0Rec2=79; joint1Rec2=10; grabberRec2=120;
    GoSmooth(); //Καλούμε την υπορουτίνα "GoSmooth"
}

void MaxJoint1(){ //Ο βραχίονας πραγματοποιεί κίνηση στην άρθρωση joint1 στο πάνω όριο
    rotorRec2=137; joint0Rec2=79; joint1Rec2=175; grabberRec2=120;
    GoSmooth(); //Καλούμε την υπορουτίνα "GoSmooth"
}

void MinGrab(){ //Οαρπαγέαςκλείνει
    rotorRec2=137; joint0Rec2=79; joint1Rec2=122; grabberRec2=45;
    GoSmooth(); //Καλούμε την υπορουτίνα "GoSmooth"
}

void MaxGrab(){ //Οαρπαγέαςανοίγει
    rotorRec2=137; joint0Rec2=79; joint1Rec2=122; grabberRec2=120;
    GoSmooth(); //Καλούμε την υπορουτίνα "GoSmooth"
}

```

```

}
void GoSmooth(){ //Πραγματοποιεί ομαλή μετάβαση των σερβοκινητήρων από μία θέση σε μία
άλλη
  for (Cnt0=0; Cnt0<200; Cnt0++){ //Πουτίνα 200 επαναλήψεων με την χρήση του Counter0
    if (rotor<rotorRec2) rotor= rotor+1; //Αν rotor < rotorRec2 τότε αύξησε την μεταβλητή
"rotor" κατά ένα
    if (rotor>rotorRec2) rotor= rotor-1; //Αν rotor > rotorRec2 τότε μείωσε την μεταβλητή "rotor"
κατά ένα
  if (joint0<joint0Rec2) joint0= joint0+1; //Αν joint0 < joint0Rec2 τότε αύξησε την μεταβλητή
"joint0" κατά ένα
  if (joint0>joint0Rec2) joint0= joint0-1; //Αν joint0 > joint0Rec2 τότε μείωσε την μεταβλητή
"joint0" κατά ένα
  if (joint1<joint1Rec2) joint1= joint1+1; //Αν joint1 < joint1Rec2 τότε αύξησε την μεταβλητή
"joint1" κατά ένα
  if (joint1>joint1Rec2) joint1= joint1-1; //Αν joint1 > joint1Rec2 τότε μείωσε την μεταβλητή
"joint1" κατά ένα
  if (grabber<grabberRec2) grabber= grabber+1; //Αν grabber < grabberRec2 τότε αύξησε την
μεταβλητή "grabber" κατά ένα
  if (grabber>grabberRec2) grabber= grabber-1; //Αν grabber > grabberRec2 τότε μείωσε την
μεταβλητή "grabber" κατά ένα
  if (rotor==rotorRec2 && joint0==joint0Rec2 && joint1==joint1Rec2 &&
grabber==grabberRec2) Cnt0= 200; //Αν ισχύουν όλες οι συνθήκες μέσα στην παρένθεση τότε
η μεταβλητή Cnt0 παίρνει την τιμή 200
  SetServo(); //Καλούμε την υπορουτίνα "SetServo"
    BreakDemo(); //Καλούμε την υπορουτίνα "BreakDemo"
    delay(20); //Καθυστέρηση εκτέλεσης του προγράμματος 20ms
  }
}

void GoPotVal(){// η αρπάγη μεταβαίνει σταδιακά στην θέση των τιμών των ποτενσιομέτρων
//rotor=124; joint0=156; joint1=140; grabber=120;
for (Cnt0=0; Cnt0<200; Cnt0++){ //Πουτίνα 200 επαναλήψεων με την χρήση του Counter0
  if (rotor<rotorRec) rotor= rotor+1; //Αν rotor < rotorRec τότε αύξησε την μεταβλητή "rotor"
κατά ένα
  if (rotor>rotorRec) rotor= rotor-1; //Αν rotor > rotorRec τότε μείωσε την μεταβλητή "rotor"
κατά ένα

```

```

if (joint0<joint0Rec) joint0= joint0+1; //Αν joint0 < joint0Rec τότε αύξησε την μεταβλητή
"joint0" κατά ένα
if (joint0>joint0Rec) joint0= joint0-1; //Αν joint0 > joint0Rec τότε μείωσε την μεταβλητή
"joint0" κατά ένα
if (joint1<joint1Rec) joint1= joint1+1; //Αν joint1 < joint1Rec τότε αύξησε την μεταβλητή
"joint1" κατά ένα
if (joint1>joint1Rec) joint1= joint1-1; //Αν joint1 > joint1Rec τότε μείωσε την μεταβλητή
"joint1" κατά ένα
if (grabber<grabberRec) grabber= grabber+1; //Αν grabber < grabberRec τότε αύξησε την
μεταβλητή "grabber" κατά ένα
if (grabber>grabberRec) grabber= grabber-1; //Αν grabber > grabberRec τότε μείωσε την
μεταβλητή "grabber" κατά ένα
if (rotor==rotorRec && joint0==joint0Rec && joint1==joint1Rec && grabber==grabberRec)
Cnt0= 200; //Αν ισχύουν όλες οι συνθήκες μέσα στην παρένθεση τότε η μεταβλητή Cnt0
παίρνει την τιμή 200
SetServo(); //Καλούμε την υπορουτίνα "SetServo"
    BreakDemo(); //Καλούμε την υπορουτίνα "BreakDemo"
    delay(30); //Καθυστέρηση εκτέλεσης του προγράμματος 40ms
}
}

```

## 10 Ανάλυση του τρόπου λειτουργίας του βραχίονα

Συνδέοντας τον βραχίονα με το ρεύμα τοποθετείται αυτόματα στην αρχική θέση, όπου τον έχουμε ορίσει από το πρόγραμμα, δηλαδή όλες οι αρθρώσεις του είναι κατακόρυφα της βάσης. Σε αυτό το σημείο, με απλά λόγια, ο βραχίονας βρίσκεται σε θέση αναμονής.

Ακόμη, τα ποτενσιόμετρα είναι εκτός λειτουργίας και για την ενεργοποίησή τους και την είσοδο του βραχίονα σε manual χειρισμό είναι αρμόδια ένα εκ των δυο buttons, που βρίσκονται εκατέρωθεν των ποτενσιομέτρων. Εφόσον βρισκόμαστε σε manual control, οι αρθρώσεις δεν λαμβάνουν τυχαίες θέσεις, αλλά τις θέσεις αυτές όπου είναι στραμμένα τα αντίστοιχα ποτενσιόμετρα.

- Το αριστερό ποτενσιόμετρο είναι υπεύθυνο για τον χειρισμό της βάσης (rotor).
- Το επόμενο ελέγχει την λειτουργία της άρθρωσης 1 (joint1).
- Το τρίτο ελέγχει τον χειρισμό της άρθρωσης 2 (joint2).
- Το τελευταίο καθορίζει το άνοιγμα η το κλείσιμο της αρπάγης.

Ο βραχίονας μπορεί, επίσης, να εκτελέσει και τυποποιημένες κινήσεις, οι οποίες έχουν προκαθοριστεί κατά τον προγραμματισμό του arduino. Για την ενεργοποίησή τους και εφόσον ο βραχίονας βρίσκεται σε manual λειτουργία έχουν οριστεί τα δύο buttons.

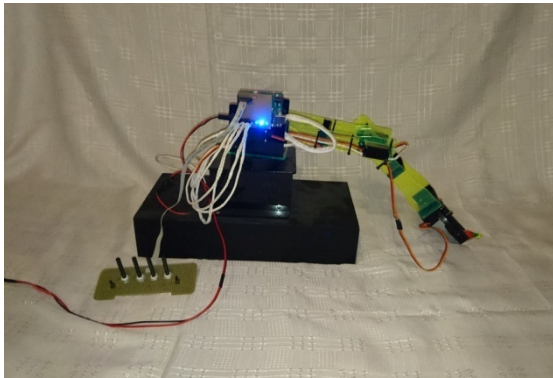
- Το αριστερό ενεργοποιεί την πρώτη κίνηση (demo1), η οποία είναι η αρπαγή και μεταφορά ενός αντικείμενου από μια θέση σε μια άλλη και αντιστρόφως (βλ. εικόνα 10 (2, 3, 4, 5)).
- Το δεξί ενεργοποιεί την δεύτερη τυποποιημένη κίνηση (demo2), που είναι το πλήρες φάσμα κίνησης του κάθε σερβοκινητήρα του βραχίονα. Οι κινήσεις γίνονται διαδοχικά ξεκινώντας από την βάση και καταλήγοντας στην αρπάγη.

Σημαντικό εδώ είναι να τονίσουμε ότι οι παραπάνω τυποποιημένες κινήσεις (demos) εκτελούνται συνεχόμενα και για την παύση τους χρειάζεται η μεταφορά σε manual λειτουργία (πάτημα ένα εκ των δυο buttons). Ο ρομποτικός βραχίονας δεν τίθεται αμέσως σε κατάσταση manual, αλλά αυτό συμβαίνει μετά την ολοκλήρωση της κίνησης. Τέλος, μπορούμε να θέσουμε τον βραχίονα στην θέση αναμονής με το ταυτόχρονο πάτημα των δυο buttons (demo0) (βλ. εικόνα 10 (1)).

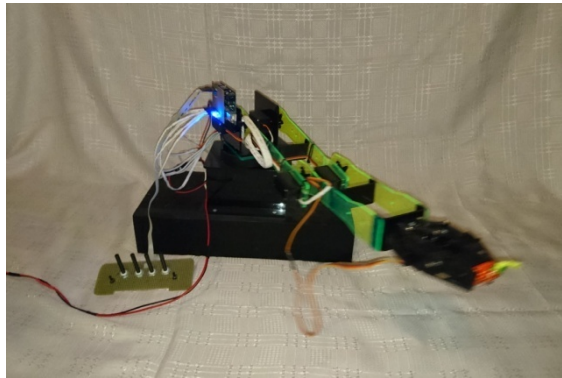




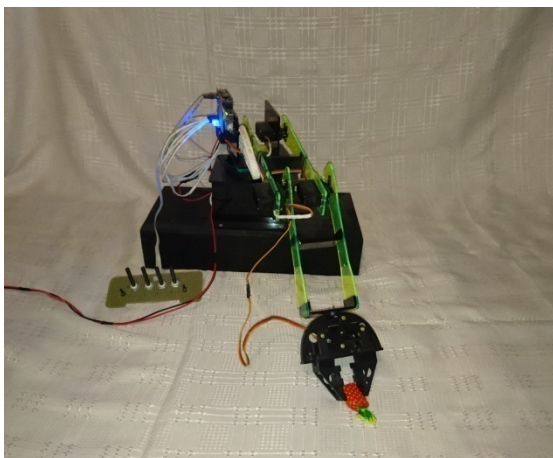
10 (1) Θέση αναμονής



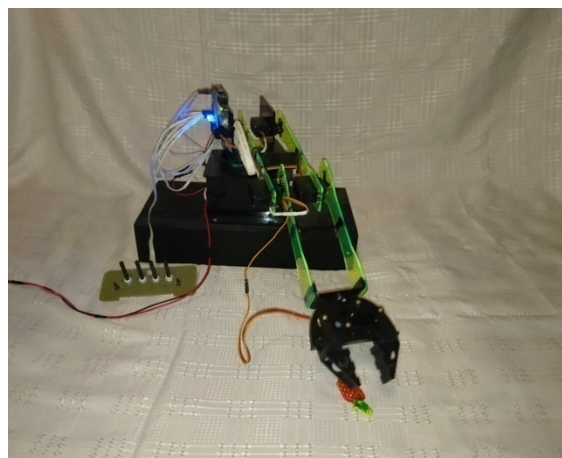
10 (2) Αρπαγή αντικειμένου



10 (3) Μεταφορά αντικειμένου



10 (4) Απελευθέρωση αντικειμένου



10 (5) Αποχώρηση αρπάγης



## 11 Πειράματα

Κατά τη διάρκεια της δημιουργίας της κατασκευής χρειάστηκε να πραγματοποιηθούν ορισμένα πειράματα, με σκοπό να διαπιστωθεί η ύπαρξη τυχόν προβλημάτων, έτσι ώστε να επιλυθούν με τον κατάλληλο εκάστοτε τρόπο. Τα προβλήματα που παρουσιάστηκαν μετά τα πειράματα, που θα αναφερθούν λεπτομερώς παρακάτω, και οι τρόποι που αντιμετωπίστηκαν οδήγησαν στην εξαγωγή ασφαλούς συμπεράσματος για την ορθότητα της κατασκευής. Πιο αναλυτικά, στο πρώτο πείραμα έγινε η σύνδεση του Arduino με τους σερβοκινητήρες, για να ελεγχθεί η σωστή λειτουργία τους, τα όρια τους και η θέση μηδέν. Από τη στιγμή που υπήρξε η επιβεβαίωση για την ακριβή λειτουργία όλων, προχώρησε η διαδικασία κατασκευής του ρομποτικού βραχίονα.

Στο δεύτερο πείραμα χρειάστηκε να γίνει μια δοκιμή στις κινήσεις του βραχίονα και εδώ αναδύθηκε ένα εμπόδιο. Λόγω του ότι έκανε τις κινήσεις αρκετά γρήγορα έσπασε το ένα μπράτσο, το οποίο, φυσικά, ξανακολλήθηκε, αλλά για να γίνει η πρόβλεψη ομαλοποίηση στις κινήσεις τοποθετήθηκε μια υπορουτίνα (Go Smooth), που βοήθησε καθοριστικά στην εξομάλυνση του προγράμματος. Σ' αυτό το σημείο κρίθηκε απαραίτητο πέρα από την εξομάλυνση του προγράμματος να γίνει μια προσθήκη για περισσότερη μηχανική υποστήριξη της κατασκευής. Αυτή η προσθήκη αφορά ένα επιπρόσθετο κομμάτι plexiglass, που εφαρμόστηκε ενδιάμεσα στα μπράτσα του βραχίονα.

Το τρίτο πείραμα που έγινε για να διαπιστωθεί αν οι προηγούμενες βελτιώσεις βοήθησαν στο να λειτουργεί η κατασκευή σωστά, έφερε κι αυτό με τη σειρά δυο ακόμη προβλήματα. Τα καλώδια από τη μια ήταν κοντά και χρειάστηκε να επιμηκυνθούν και από την άλλη παρατηρήθηκε η αστάθεια της κατασκευής, το ότι έχανε δηλαδή το κέντρο βάρους της. Σ' αυτή την περίπτωση αναγκαία ήταν η αύξηση της μηχανικής αντοχής με την προσθήκη ενός χάρτινου στεφανιού στον περιστροφέα.

Τέλος, το τέταρτο πείραμα, εφόσον η κατασκευή είχε ολοκληρωθεί, αφορούσε καθαρά την επαλήθευση των αρχικών υπολογισμών για τις κινήσεις του βραχίονα, τα όρια των σερβοκινητήρων και τη δύναμή του με βάση το βάρος των αντικειμένων που μπορούσε να σηκώσει και δεν αντιμετωπίστηκε κανένα πρόβλημα. Τα τέσσερα πειράματα βοήθησαν και στο να υπάρξει παραπάνω μελέτη για να αντιμετωπιστούν τα διάφορα προβλήματα, που προέκυψαν από αυτά, αλλά και πάνω από όλα στο να κατασκευαστεί ένας ρομποτικός βραχίονας τεσσάρων βαθμών ελευθερίας, ο οποίος δεν αντιμετωπίζει πλέον κανένα πρόβλημα και μπορεί να λειτουργεί σωστά και με ακρίβεια όλες τις κινήσεις, με βάση τις εντολές που του δίνονται.

## 12 Προβλήματα - Βελτιώσεις

### *Προβλήματα*

Κατά τη διαδικασία κατασκευής του ρομποτικού βραχίονα έγινε αρκετές φορές επανεξέταση του αρχικού σχεδίου με σκοπό το καλύτερο αποτέλεσμα, λόγω της πολυπλοκότητας της κατασκευής. Εντοπίστηκαν και σημειώθηκαν και αρκετά προβλήματα, αλλά από την άλλη πλευρά και ορισμένες βελτιώσεις, οι οποίες θα μπορούσαν να υλοποιηθούν σε μεταγενέστερες δοκιμές. Τα προβλήματα που παρατηρήθηκαν, αφορούν και την ορθή στάση και αντοχή του δημιουργήματος και τον τομέα της επίτευξης της κίνησης και θα παρατεθούν σε συνδυασμό με τις άμεσες λύσεις που εφαρμόστηκαν. Οι δυσκολίες κατά την δημιουργία της κατασκευής ώθησαν στην προσθήκη κι άλλων υλικών πέρα από όσα θα χρησιμοποιούταν κατά το αρχικό πλάνο.

Πιο αναλυτικά για να υπάρξει σταθερότητα της κατασκευής τοποθετήθηκε ένα κομμάτι ξύλο κάτω από την βάση, για να χρησιμεύει ως αντίβαρο. Χωρίς αυτό το αντίβαρο ο βραχίονας δεν μπορούσε να εκτελέσει την κίνηση σωστά, καθώς λόγω του βάρους του αντικειμένου που σήκωνε, έπεφτε και με αυτήν την τοποθέτηση στη βάση επιτεύχθηκε η εξισορρόπηση του. Κρίθηκε αναγκαία επίσης, η χρήση ενός χάρτινου στεφανιού στον περιστροφέα για περισσότερη μηχανική αντοχή. Η ταλάντευση των αρθρώσεων του βραχίονα αντιμετωπίστηκε με αυτόν τον τρόπο ενώνοντας με το χάρτινο στεφάνι τη βάση της κατασκευής με το λεπτό κομμάτι του υλικού (plexiglass), το οποίο με τη σειρά του αποτελεί σημείο στήριξης των αρθρώσεων. Η τελευταία προσθήκη στην κατασκευή ήταν αυτή κομματιών (plexiglass) ανάμεσα στα «μπράτσα» των αρθρώσεων, βοηθώντας στη μηχανική υποστήριξη, καθώς τα «μπράτσα» μπορούν να χαρακτηριστούν μ' αυτήν την προσθήκη πιο «δεμένα» και πάνω από όλα ανθεκτικά από την στιγμή που ο ρομποτικός βραχίονας μπαίνει σε λειτουργία.

Όσον αφορά τους σερβομηχανισμούς, ήταν φυσικό ότι και εκεί θα δημιουργούνταν ορισμένα ζητήματα. Η μεγάλη ταχύτητα που αναπτύσσουν σε σχέση με την μηχανική αντοχή του υλικού κατασκευής (plexiglass), οδήγησε στη δημιουργία μιας υπορουτίνας εξομάλυνσης. Οι απότομες αρχικές κινήσεις τους με την υπορουτίνα ομαλοποιήθηκαν, χωρίς να υπάρξει περαιτέρω θέμα με την αντοχή του υλικού. Ακόμη, κατά την διάρκεια των εκτεταμένων δοκιμών οι μικρότεροι σε δύναμη σερβομηχανισμοί καταστράφηκαν, διότι παρά τις προδιαγραφές του κατασκευαστή και τους υπολογισμούς που έγιναν, δεν επαρκούσαν για την παρούσα κατάσταση. Άλλη μια αλλαγή που έλαβε μέρος στους σερβομηχανισμούς ήταν η επιμήκυνση των καλωδίων τους για να συνδεθούν στην πλακέτα, με σκοπό ο ρομποτικός βραχίονας να εκτελεί όλες τις κινήσεις του σωστά χωρίς να αποσυνδέεται ή να προβάλλει αντίσταση κάποιο καλώδιο κατά την κίνηση του. Όμως δεν έφτανε μόνο η επιμήκυνση γιατί και

πάλι υπήρξε ένα πρόβλημα με την πλακέτα, στην οποία τελικά αλλάχθηκε μέρος, καθώς το ζήτημα εδώ ήταν πως με την επέκταση τα καλώδια μπλέκονταν, μέχρι να φτάσουν στην πλακέτα, και η κατάσταση χειροτέρευε όταν έμπαινε σε λειτουργία ο βραχίονας. Για τη σωστή, λοιπόν, λειτουργία και για να μην υπάρχει αυτό το μπέρδεμα στα καλώδια και εφόσον η πλακέτα πρέπει να τοποθετείται σε σταθερό μέρος, ήταν αναγκαία η τοποθέτηση της σε διαφορετικό μέρος από το αρχικό.

Επιπροσθέτως και στον μικροελεγκτή (arduino) προέκυψαν εμπόδια. Αρχικά, εξαιτίας του περιορισμένου ρεύματος εξόδου του arduino σε σχέση με το ρεύμα τροφοδοσίας, το οποίο απαιτείται για να τροφοδοτήσει και τον μικροελεγκτή και τους σερβομηχανισμούς, δρομολογήθηκε η ενίσχυση της κατασκευής με τη δημιουργία μιας πλακέτας οδήγησης (servoshield), η οποία τοποθετήθηκε πάνω στον μικροελεγκτή και είναι ικανή να κατανέμει σωστά το ρεύμα στον μικροελεγκτή (arduino) και στους σερβομηχανισμούς για την πλήρη λειτουργία του βραχίονα. Το ζήτημα, λοιπόν, είναι διπολικό από την μία όπως προαναφέρθηκε, είναι το τι θα γίνει για να κατανεμηθεί το ρεύμα σωστά και από την άλλη το πως θα πραγματοποιηθεί η τροφοδοσία της κατασκευής. Η λύση βρέθηκε στην προσαρμογή ενός τροφοδοτικού ηλεκτρονικού υπολογιστή, ώστε να συνδέεται στην πλακέτα οδήγησης (servoshield) και να «δίνει» την κατάλληλη τάση που χρειάζεται ο βραχίονας για να κινηθεί. Επιπλέον, λόγω του μεγάλου σε μέγεθος αρχικού προγράμματος ο μικροελεγκτής κολλούσε και δεν ανταποκρινόταν καταλλήλως. Παρατηρήθηκε, δηλαδή ότι αν γινόταν χρήση πάνω από το 70-72% της μνήμης του μικροελεγκτή τότε παρουσιάζονταν προβλήματα. Αυτό οδήγησε στο να κατασκευαστεί κώδικας μικρότερου μεγέθους, με αποτέλεσμα ο μικροελεγκτής να λειτουργεί σωστά.

Είναι φανερό ότι από όλα τα παραπάνω προβλήματα και τις λύσεις που βρέθηκαν η κατασκευή του ρομποτικού βραχίονα κρίνεται άκρως πολυσύνθετη. Πέρα όμως από την πολυπλοκότητα της και τις ώρες εργασίας για να λειτουργεί σωστά, τα απανωτά ζητήματα που προέκυψαν οδήγησαν στην προσθήκη εξαρτημάτων σε συνδυασμό με την επιλογή του υλικού της κατασκευής (plexiglass) αύξησαν αρκετά τον προϋπολογισμό. Τέλος, αν και θα παρατεθεί στις προτεινόμενες βελτιώσεις, το κόστος θα μπορούσε να μειωθεί αν γινόταν χρήση άλλων υλικών π.χ. ξύλο για την κατασκευή.

### ***Βελτιώσεις***

Στην διάρκεια της κατασκευής αντιμετωπίστηκαν διάφορων ειδών προβλήματα και με τις άμεσες δυνατές λύσεις για το εκάστοτε πρόβλημα, δρομολογήθηκαν συνεχείς αναδιαμορφώσεις του αρχικού πλάνου. Μέσα από αυτό, σημειώθηκε πολύωρη μελέτη, αλλά και έρευνα για να βγει εις πέρας το εγχείρημα της δημιουργίας του ρομποτικού βραχίονα

και ταυτόχρονα δόθηκε το έναυσμα για να εντοπισθούν κάποιες βελτιώσεις, οι οποίες θα μπορούσαν να λάβουν χώρα σε μεταγενέστερες προσπάθειες κατασκευής ρομποτικού βραχίονα.

Αρχικά, αναγκαία θα είναι η χρήση κάποιου κώδικα που περιέχει τεχνικές που προβλέπουν το Debount effe, δηλαδή απενεργοποιούν για ορισμένο χρόνο τα button0 και button1, καθώς μετά το πρώτο πάτημα επιφέρονται πολλές επαναλήψεις και με αυτές τις τεχνικές, λόγω της απενεργοποίησης, αυτό εμποδίζεται. Ακόμη, θα ήταν καλύτερη η χρήση σερβοκινητήρων πιο δυνατών για να μπορούν να σηκώνουν αντικείμενα μεγαλύτερου βάρους σε συνδυασμό, όμως, με μια καλύτερη καλωδίωση, για την αποφυγή, προβλημάτων που όπως προαναφέρθηκε προέκυψαν στο παρόν εγχείρημα. Η καλωδίωση, ταυτόχρονα, αφορά και το χειριστήριο που μεταφράζεται ως μία επιμήκυνση των καλωδίων, με στόχο ο βραχίονας να ελέγχεται και από μεγαλύτερη απόσταση.

Μία άλλη βελτίωση θα ήταν η τοποθέτηση περισσότερων σερβομηχανισμών πάνω στον βραχίονα, με αποτέλεσμα να λυγίζει και σε άλλα σημεία, πετυχαίνοντας έναν πιο ευέλικτο βραχίονα. Με απλά λόγια να υπάρχουν περισσότεροι βαθμοί ελευθερίας. Ωστόσο, αξίζει να σημειωθεί ότι λόγω του συνολικού κόστους θα ήταν καλύτερη η αλλαγή του υλικού κατασκευής, δηλαδή το plexiglass που εδώ χρησιμοποιήθηκε, να αντικατασταθεί για παράδειγμα με ξύλο ή αλουμίνιο κτλ..

Επίσης, άλλη μια προσθήκη που θα προσέλυε πολύ το ενδιαφέρον, σε ένα μελλοντικό εγχείρημα, θα ήταν αυτή των αισθητήρων και συνοπτικά αναφέρονται τριών ειδών αισθητήρες. Πρώτον, τερματικοί αισθητήρες που ομαλοποιούν τις κινήσεις των σερβομηχανισμών όταν φτάνουν τα μέγιστα και τα ελάχιστα όριά τους και σταματούν και δεν ξεφεύγουν εξ αυτών, με αποτέλεσμα να αποφευχθούν οι ζημιές στην κατασκευή. Δεύτερον, αισθητήρας πρόσκρουσης, έτσι ώστε ο βραχίονας να είναι σε θέση να αντιληφθεί την ύπαρξη ενός αντικειμένου και να μην υπάρξει σύγκρουση, προστατεύοντας και με αυτό τον τρόπο την ακεραιότητα του, αλλά και του αντικειμένου. Τρίτον, αισθητήρας αναγνώρισης αντικειμένου κατά τον οποίο η αρπάγη θα είναι ικανή να προσαρμοστεί μόνη της και να πιάσει το κάθε αντικείμενο.

Στις ήδη προαναφερθείσες βελτιώσεις μπορούμε να συμπεριλάβουμε και μερικές ακόμη. Την αλλαγή, δηλαδή της υπό ρουτίνας "Go Pot" με την υπό ρουτίνα "Go Smouth", για να μειωθεί κι άλλο το μέγεθος του κώδικα, ώστε να χρησιμοποιεί λιγότερη μνήμη από τον μικροελεγκτή, καθιστώντας το πρόγραμμα πιο «ελαφρύ».

Εύκολα, λοιπόν μπορεί ο καθένας να συμπεράνει ότι μετά την κατασκευή του ρομποτικού βραχίονα μέσα από τα προβλήματα που προέκυψαν, τις λύσεις που εφαρμόστηκαν αλλά και από τις βελτιώσεις που προτάθηκαν, έγινε μια πολυεπίπεδη μελέτη. Μέσα από την

οποία, με λίγα λόγια, δόθηκε η αφορμή για εμπλουτισμό των γνώσεων στον τομέα της ρομποτικής.

## 13 Επίλογος

Η διεκπεραίωση της παρούσας πτυχιακής εργασίας αναμφίβολα αποτελεί ένα μεγάλο κατόρθωμα, καθώς απαιτούσε πολύωρη εργασία και για την δημιουργία της κατασκευής, αλλά και για το θεωρητικό της κομμάτι. Στόχος ήταν η αρτιότητα της κατασκευής, ώστε να μπορεί να λειτουργεί χωρίς να υπάρχει κανένα πρόβλημα και φυσικά, η συγκέντρωση πληροφοριών για τη ρομποτική που σε συνδυασμό με το υλικό της συγγραφικής ομάδας για το πώς κατασκευάστηκε ο ρομποτικός βραχίονας να ταξινομηθεί με τέτοιο τρόπο, με σκοπό να επιτευχθεί πληρότητα και στο θεωρητικό μέρος. Αναμφίβολα, η ενασχόληση με τον τομέα της ρομποτικής μπορεί να χαρακτηριστεί ιδιαίτερος ενδιαφέρουσα, γιατί η επαφή με τα πεδία της μαγνήτισαν εύκολα την προσοχή.

Επιπροσθέτως, θα ήταν λάθος αν δεν αναφερόταν ότι η κατασκευή του ρομποτικού βραχίονα, αλλά και ο τρόπος λειτουργία του, λόγω της πολυπλοκότητας τους, δημιούργησαν διαφόρων ειδών προβλήματα. Αυτό, είχε ως αποτέλεσμα να αναθεωρηθεί αρκετές φορές το αρχικό πλάνο και να γίνουν εκτεταμένες προσπάθειες, για να αντιμετωπιστεί το κάθε πρόβλημα. Η επίλυση των εμποδίων, όπως αναφέρθηκε σε προηγούμενο κεφάλαιο, οδήγησε στην προσθήκη κομματιών, αλλά και σε γενικότερες αλλαγές. Το συμπέρασμα είναι πως αυτή η ασχολία πάνω με τον ρομποτικό βραχίονα και ως εκ τούτου με την ρομποτική βοήθησε σε σημαντικό βαθμό στην επιμόρφωση και στον εμπλουτισμό των γνώσεων, σε τέτοιο σημείο που, όπως και πάλι αναφέρθηκε σε προηγούμενο κεφάλαιο, προτάθηκαν και βελτιώσεις, οι οποίες θα μπορούσαν να εφαρμοστούν σε κάποιο μελλοντικό εγχείρημα.

Τέλος, μελετώντας σε καθημερινή σχεδόν βάση πάνω από τον τομέα της ρομποτικής, επαληθεύτηκε το ότι ένας από τους πιο ελκυστικούς τομείς της επιστήμης. Σίγουρα, είναι μια επιστήμη που έχει γρήγορη εξελικτική πορεία και το μόνο που πρέπει να ευελπιστούμε είναι η πορεία αυτή να έχει μόνο θετικό αντίκτυπο στην ανθρώπινη ζωή. Η ρομποτική, λοιπόν, είναι μια τεχνολογία με μέλλον και για το μέλλον.

## 14 Κροτολόγιο

Υλικά	Ποσότητα	Τιμή
Plexiglass	2 (50cm*50cm*3cm)	40€
Σερβομηχανισμοί	6	15€
Αρπάγη	1	7€
Πλακέτα arduino	1	25€
Πλακέτα shield	1	10€
Τηλεχειριστήριο	1	5€
Θερμοσυστελλόμενα	1(150cm)	1,50€
Τροφοδοτικό	1	20€
Ξύλινη βάση	1	2€
Έξοδα αποστολής	-	10€
Τελικό ποσό		135,5€

## 15 Βιβλιογραφία

### *Βιβλία*

- GrainJohn, Εισαγωγή στη ρομποτική, Μηχανική και Αυτόματος έλεγχος, Εκδόσεις Τζιόλα, 2008
- Δρ. Βολογιαννίδης Σταύρος, Συστήματα Αυτομάτου Ελέγχου, Θεωρία και Εφαρμογές, Διδακτικές σημειώσεις, Τμήμα Πληροφορικής & Επικοινωνιών , ΤΕΙ Σερρών, 2009
- Δουλγέρη Ζωή, ΡΟΜΠΟΤΙΚΗ, κινηματική δυναμική και έλεγχος αρθρωτών βραχιόνων, Εκδόσεις Κριτική, 2007
- Κανάραχος Ε. Ανδρέας-Καθηγητής Ε.Μ.Π, Μηχανισμοί και Ρομποτικά Συστήματα, Εκδόσεις Παπασωτηρίου, 1996
- Κουμουλής Ν. Φώτης, Βασίλης Γ. Μέρτζιος, Εισαγωγή στη ρομποτική, Εκδόσεις Παπασωτηρίου
- Ντίνας, Ν. & Σαϊτάκης. Ε. (2003). Μελέτη και κατασκευή ρομποτικού βραχίονα με ψηφιακά συστήματα. ΑΤΕΙ Κοζάνης- Σχολή Τεχνολογικών Εφαρμογών-Τμήμα Βιομηχανικού Σχεδιασμού

### *Ιστοσελίδες*

- <http://www.wikipedia.org/>
- <http://www.arduino-lab.weebly.com/>
- <http://www.grobot.gr/>
- <http://www.deltahacker.gr/>
- <http://www.arduino.cc/>
- <http://www.youtube.com/>
- <https://projectrobotics.wordpress.com/>
- [www.heroc.tuc.gr/Papers\\_Posters/pdf/8-1.pdf](http://www.heroc.tuc.gr/Papers_Posters/pdf/8-1.pdf)



