



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ

ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΚΑΤΑΣΚΕΥΗ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ Χ-Υ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΧΡΟΝΗΣ ΙΑΣΩΝ-ΙΩΑΝΝΗΣ

Επίβλεψη: Γεώργιος Καμπουράκης

Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2017

Χρόνης Ιάσων-Ιωάννης



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ

ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΚΑΤΑΣΚΕΥΗ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ Χ-Υ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Χρόνης Ιάσων-Ιωάννης

Επίβλεψη: Γεώργιος Καμπουράκης, Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 7 Νοεμβρίου 2017.

Χρόνης Ιάσων-Ιωάννης

.....

Καμπουράκης Γεώργιος
Καθηγητής Ε.Μ.Π.

.....

Κουκούτσης Ηλίας
Επ.Καθηγητής Ε.Μ.Π.

.....

Λούμος Βασίλειος
Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2017

.....

Χρόνης Ιάσων-Ιωάννης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

**Copyright© Χρόνης Ιάσων-Ιωάννης.
Copyright© Καμπουράκης Γεώργιος, 2017
Με επιφύλαξη παντός δικαιώματος. All rights reserved.**

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς το συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Σκοπός αυτής της διπλωματικής εργασίας ήταν η κατασκευή ενός ρομποτικού βραχίονα X-Y, ο οποίος θα έχει τη δυνατότητα να μεταφέρει έναν αναλυτή έντασης ήχου, καθώς και ένα μεγάφωνο εντός ενός ορθογωνικού πλαισίου.

Το πλαίσιο μέσα στο οποίο μπορεί να κινηθεί η κεφαλή του ρομπότ καλύπτει μια επιφάνεια 1.00m x 1.20m. Η βασική λειτουργία του ρομπότ είναι να σαρώνει την επιφάνεια αυτή με βήμα 5cm ή 10cm με σκοπό την καταγραφή αντανάκλασης ήχων πάνω σε δοκίμια. Το ρομπότ έχει επιπλέον τη δυνατότητα να κινείται και χειροκίνητα με τη χρήση είτε ενός πληκτρολογίου το οποίο δέχεται συντεταγμένες για την επιλογή μιας συγκεκριμένης θέσης, είτε με τη χρήση Joystick, το οποίο μπορεί να μετακινήσει την κεφαλή σε οποιαδήποτε κατεύθυνση εντός του πλαισίου.

Στο πρώτο κομμάτι της εργασίας, έχουμε τα εισαγωγικά στοιχεία για τη ρομποτική και τα καρτεσιανά ρομπότ, καθώς και μια στοιχειώδη κινηματική ανάλυση των βηματικών κινητήρων και της μετάδοσης κίνησης μέσω σπειρωμάτων.

Στη συνέχεια, αναλύονται κατά σειρά, η κατασκευή του μηχανολογικού και του ηλεκτρονικού μέρους της κατασκευής και ο προγραμματισμός του μικροεπεξεργαστή.

Τέλος, παρουσιάζονται τα συμπεράσματα και ιδέες για βελτίωση και περαιτέρω έρευνα πάνω στο αντικείμενο της παρούσης εργασίας

Λέξεις κλειδιά

Καρτεσιανό ρομπότ, ρομποτικός βραχίονας X-Y, βηματικός κινητήρας, σπειρώματα, Arduino

Abstract

The aim of this thesis was to construct a Cartesian robot, which can move a sound intensity analyzer, as well as a speaker inside a rectangular frame.

The frame, inside of whom the head of the robot can move, covers a surface of 1.00m x 1.20m. The robot's basic function is to scan this surface, making steps of 5cm or 10cm at a time in order to record sound reflections on a sample surface. The robot has, additionally, the ability to move manually, either using a keyboard to read coordinates of a specific position, or using a joystick that moves the head towards any direction inside the frame

The thesis' first part displays some introductory information about robotics and especially about Cartesian robots, as well as a basic kinematic analysis of movement transmission from stepper motors to lead screws.

Then, there is a display of the mechanical and electronic design of the robot and the programming of the microprocessor.

Finally, there is a short display of the conclusions we came in during this specific project, as well as ideas on how it can be improved and on further research that can be done on the project.

Key words

Cartesian robot, robotic arm X-Y, stepper motor, lead screw, Arduino

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω την οικογένειά μου, ιδιαιτέρως τους γονείς μου και την αδερφή μου, για τη στήριξη και συμπαράστασή της καθ' όλη τη διάρκεια των σπουδών μου, η οποία ολοκληρώνεται με την εκπόνηση της παρούσας διπλωματικής εργασίας.

Στη συνέχεια θα ήθελα να ευχαριστήσω τον κ. Καμπουράκη, τόσο για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα θέμα το οποίο συνδυάζει ένα μεγάλο εύρος των γνώσεων των οποίων θα πρέπει να έχει ένας ηλεκτρολόγος μηχανικός

Θα ήθελα να ευχαριστήσω τους φίλους μου, με τους οποίους πορευθήκαμε μαζί όλα αυτά τα χρόνια.

Τέλος θα ήθελα πέρα από τις ευχαριστίες μου να εκφράσω την ευγνωμοσύνη μου στον θείο μου κ. Βαθρακοκοίλη, ο οποίος ήταν δίπλα μου καθ' όλη τη διάρκεια της κατασκευής της εργασίας, ανιδιοτελώς, προμηθεύοντας με με τα απαραίτητα εργαλεία αλλά και με της τεχνικές του γνώσεις και δεξιότητες.

● ΠΕΡΙΕΧΟΜΕΝΑ

1	ΕΙΣΑΓΩΓΗ	13
1.1	ΓΕΝΙΚΑ ΓΙΑ ΤΗ ΡΟΜΠΟΤΙΚΗ	13
1.2	ΚΑΡΤΕΣΙΑΝΑ Η ΓΡΑΜΜΙΚΑ ΡΟΜΠΟΤ	14
2	ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΥΠΟ ΚΑΤΑΣΚΕΥΗΝ ΡΟΜΠΟΤ:	16
2.1	ΒΗΜΑΤΙΚΟΙ ΚΙΝΗΤΗΡΕΣ:	17
2.1.1	<i>Στατική:</i>	17
2.1.2	<i>Half-Step και Microstepping</i>	19
2.1.3	<i>Τριβή και νεκρή ζώνη</i>	20
2.1.4	<i>Δυναμική</i>	21
2.1.5	<i>Ροπή και ταχύτητα</i>	22
2.2	ΣΠΕΙΡΩΜΑΤΑ ΚΑΙ ΔΥΝΑΜΕΙΣ:.....	23
3	ΚΑΤΑΣΚΕΥΗ ΜΗΧΑΝΟΛΟΓΙΚΟΥ ΤΜΗΜΑΤΟΣ:	27
3.1	ΤΟ ΠΛΑΙΣΙΟ ΤΟΥ ΡΟΜΠΟΤ	27
3.2	ΤΟΠΟΘΕΤΗΣΗ ΣΠΕΙΡΩΜΑΤΩΝ ΚΑΙ ΡΟΥΛΕΜΑΝ ΓΙΑ ΤΗ ΜΕΤΑΦΟΡΑ ΤΟΥ ΑΞΟΝΑ	29
3.3	ΚΑΤΑΣΚΕΥΗ ΕΓΚΑΡΣΙΟΥ ΑΞΟΝΑ ΚΑΙ ΚΕΦΑΛΗΣ	32
3.4	ΛΙΣΤΑ ΥΛΙΚΩΝ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ	34
4	ΗΛΕΚΤΡΙΚΗ -ΗΛΕΚΤΡΟΝΙΚΗ ΥΠΟΔΟΜΗ ΚΑΙ ΣΧΕΔΙΑΣΗ	35
4.1	ΒΗΜΑΤΙΚΟΙ ΚΙΝΗΤΗΡΕΣ	36
4.2	DRIVERS ΒΗΜΑΤΙΚΩΝ ΚΙΝΗΤΗΡΩΝ	37
4.3	ΤΡΟΦΟΔΟΤΙΚΟ 12V/8A	39
4.3.1	<i>Ανάλυση κυκλώματος</i>	39
4.3.2	<i>Διαδικασία κατασκευής</i>	43
4.4	ΠΕΡΙΦΕΡΕΙΑΚΑ ΒΟΗΘΗΜΑΤΑ.....	50
4.4.1	<i>Οθόνη LCD 16x2</i>	50
4.4.2	<i>Πληκτρολόγιο Membrane 4x4 Matrix</i>	52
4.4.3	<i>Sparkfun Thumb Joystick</i>	52
4.5	ΟΡΓΑΝΩΣΗ ΤΟΥ ΚΟΥΤΙΟΥ	54
5	ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ARDUINO	56
5.1	ARDUINO.....	56
5.2	ΔΗΜΙΟΥΡΓΙΑ ΚΩΔΙΚΑ.....	57
5.3	Ο ΚΩΔΙΚΑΣ:.....	59
5.3.1	<i>Τμήμα δηλώσεων</i>	59
5.3.2	<i>Void setup()</i>	61
5.3.3	<i>Οι συναρτήσεις</i>	61
5.3.4	<i>Void loop()</i>	69
6	ΣΥΜΠΕΡΑΣΜΑΤΑ	71
	ΒΙΒΛΙΟΓΡΑΦΙΑ	72

1 ΕΙΣΑΓΩΓΗ

1.1 ΓΕΝΙΚΑ ΓΙΑ ΤΗ ΡΟΜΠΟΤΙΚΗ

Ρομποτική είναι ένας σύγχρονος σχετικά τεχνολογικός κλάδος της αυτοματοποίησης, που έχει ως αντικείμενο τη μελέτη, το σχεδιασμό και την λειτουργία των ρομπότ, καθώς και την έρευνα για την περαιτέρω ανάπτυξή τους.

Σύμφωνα με τον ορισμό του Ινστιτούτου ρομπότ των ΗΠΑ, “ρομπότ είναι μια επαναπρογραμματιζόμενη πολυλειτουργική χειριστική διάταξη, σχεδιασμένη για τη μετακίνηση υλικών, εξαρτημάτων, εργαλείων και εξειδικευμένων διατάξεων, μέσω μεταβλητών, προγραμματισμένων κινήσεων για την εκτέλεση μιας σειράς εργασιών”.

Κάθε ρομπότ αποτελείται από τρία ξεχωριστά συστήματα τα οποία συνεργάζονται για να εκτελέσουν μια προκαθορισμένη λειτουργία:

- το μηχανικό, το οποίο αναφέρεται αρχικά στο πλαίσιο, στη φόρμα ή στο σχήμα του ρομπότ τα οποία είναι τέτοια ώστε να του δίνουν τη δυνατότητα να εκτελέσει τη λειτουργία για την οποία κατασκευάστηκε. Το μηχανικό μέρος αποτελεί τη λύση του κατασκευαστή στην ολοκλήρωση του έργου και εξαρτάται από το περιβάλλον στο οποίο θα χρησιμοποιηθεί το ρομπότ. Με απλά λόγια, το μηχανολογικό σχέδιο προσαρμόζεται στη λειτουργία, την οποία ο κατασκευαστής θέλει να επιτύχει.
- το ηλεκτρονικό και ηλεκτρολογικό μέρος το οποίο τροφοδοτεί και ελέγχει το ρομπότ. Η τροφοδοσία γίνεται σε κάθε περίπτωση σε μορφή ηλεκτρικής ενέργειας, είτε άμεσα από το δίκτυο ηλεκτρικής ενέργειας, είτε με τη χρήση μπαταριών
- τον κώδικα, ο οποίος δίνει τη δυνατότητα στο ρομπότ να αποφασίζει πότε και πως θα κάνει κάτι. Το πρόγραμμα αποτελεί τον πυρήνα της όλης κατασκευής, αφού ανεξαρτήτως της ποιότητας της κατασκευής του μηχανικού και του ηλεκτρολογικού τμήματος, ένας κακώς σχεδιασμένος κώδικας μπορεί να προκαλέσει πολλά προβλήματα, όπως χαμηλές επιδόσεις ή αυξημένη κατανάλωση ενέργειας.

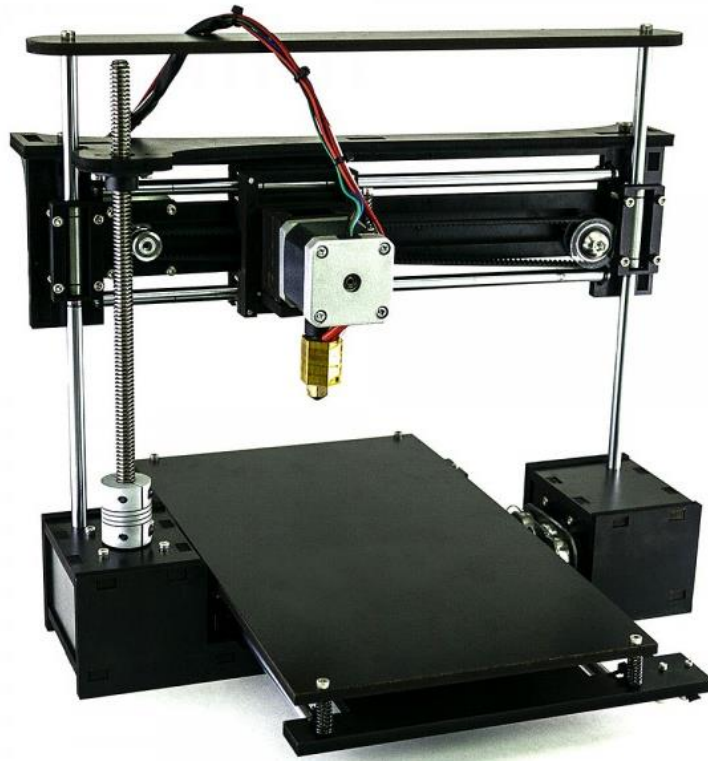
1.2 ΚΑΡΤΕΣΙΑΝΑ Η ΓΡΑΜΜΙΚΑ ΡΟΜΠΟΤ

Τα καρτεσιανά ρομπότ είναι ρομπότ στα οποία η κίνηση τους στο χώρο πραγματοποιείται γραμμικά σε έναν, δύο ή τρεις άξονες, σε γωνία ενενήντα μοιρών μεταξύ τους. Κινούνται δηλαδή σε ευθείες τροχιές και δεν περιστρέφονται. Για επιτευχθεί αυτή η ευθύγραμμη κίνηση χρησιμοποιούνται – μεταλλικοί συνήθως - άξονες, οι οποίοι λειτουργούν ως οδηγοί για να μεταφέρουν είτε φορτίο, είτε άλλους άξονες. Η διάταξη αυτή απλοποιεί ιδιαίτερα τον έλεγχο του ρομπότ και την επίλυση του βραχίονα.

Το κυρίως σώμα ενός τέτοιου ρομπότ αποτελείται από έναν έως τρεις άξονες, παρόλα αυτά η ίδια η βάση του ρομπότ μπορεί να μην είναι σταθερή, αλλά να κινείται και αυτή ευθύγραμμα μέσα σε ορισμένα όρια.

Τέτοια ρομπότ βρίσκουν εφαρμογές σε πολλούς τομείς της βιομηχανίας, όπως οι ζώνες μεταφοράς και οι μηχανές ανύψωσης και τοποθέτησης. Οι μηχανές CNC – Computer Numerical Control- είναι μηχανές που χρησιμοποιούνται για την κοπή και τη χάραξη διαφόρων επιφανειών και ανήκουν επίσης στην κατηγορία των καρτεσιανών ρομπότ. Τέλος, μια ευρεία και συνεχώς αναπτυσσόμενη εφαρμογή καρτεσιανών ρομπότ είναι οι XY-plotters και οι εκτυπωτές 3D.

Παρά τα πολλά πλεονεκτήματά τους τα καρτεσιανά ρομπότ σε αρκετές περιπτώσεις απορρίπτονται στη βιομηχανία. Αυτό συμβαίνει κυρίως γιατί στερούνται μηχανικής ευελιξίας – δεν μπορούν να αντιληφθούν αντικείμενα που δεν είναι ορατά από τη βάση τους- καθώς επίσης η ευθύγραμμη κίνηση που πραγματοποιούν τα αναγκάζει να κινούνται με μικρότερη ταχύτητα από ρομπότ με περιστρεφόμενη βάση.



Εκτυπωτής 3D

2 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΥΠΟ ΚΑΤΑΣΚΕΥΗΝ ΡΟΜΠΟΤ:

Το υπό κατασκευήν αντικείμενο αυτής της εργασίας είναι ένα καρτεσιανό ρομπότ δύο αξόνων -X,Y-, με βασικές διαστάσεις κίνησης 1.20m x1.00m. Η κεφαλή του ρομπότ σχεδιάστηκε για να μεταφέρει ένα μικρόφωνο και ένα μεγάφωνο σε απόσταση 15 εκατοστών από το επίπεδο κίνησης των αξόνων. Για την κίνηση του εγκάρσιου βραχίονα -X άξονας - χρησιμοποιούνται σαν οδηγία τα κάθετα σε αυτόν τμήματα του πλαισίου της κατασκευής -Y άξονας-, ενώ η κεφαλή κινείται πάνω στον εγκάρσιο βραχίονα.

Το ρομπότ έχει τη δυνατότητα να μεταφέρει φορτίο έως πέντε κιλά χωρίς να επηρεάζεται η απόδοσή του. Παρ' όλα αυτά κατά τη διάρκεια των πειραμάτων το συνολικό βάρος του φορτίου δε θα ξεπερνά το ένα κιλό.

Η κίνηση πραγματοποιείται σε βήματα 5cm ή 10cm σε κάθε άξονα κατόπιν επιλογής του χρήστη. Έτσι, για να ολοκληρωθεί η κίνηση του ρομπότ η κεφαλή θα πρέπει να σαρώσει όλη την επιφάνεια του πλαισίου του ρομπότ.

Η κίνηση γίνεται με χρήση τριών βηματικών κινητήρων, οδηγούμενοι από τρεις drivers TB6560 και η μετάδοση της γίνεται με σπειρώματα 16mm και παξιμάδια. Ο μικροεπεξεργαστής που χρησιμοποιείται είναι το Arduino Mega. Η τροφοδοσία τόσο των drivers όσο και του Arduino, γίνεται με τη χρήση τροφοδοτικού 12V/8A. Τα στοιχεία αυτά είναι οργανωμένα σε κουτί στο οποίο έχουν εφαρμοστεί επιπλέον οθόνη, πληκτρολόγιο και joystick για να διευκολύνεται η διενέργεια με το χρήστη.

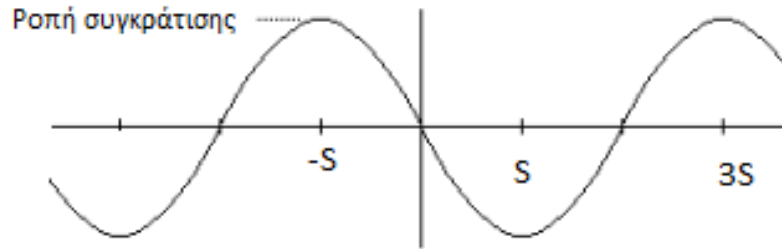
Κινηματική ανάλυση ρομπότ:

2.1 ΒΗΜΑΤΙΚΟΙ ΚΙΝΗΤΗΡΕΣ:

Οι βηματικοί κινητήρες είναι ηλεκτρικοί κινητήρες, οι οποίοι χρησιμοποιούν συνδυασμό ηλεκτρικών παλμών για την κίνησή τους. Ο κινητήρας αυτός αποτελείται από έναν δρομέα μαλακού σιδήρου με οδοντώσεις και έναν στάτη με τέσσερα ζεύγη ηλεκτρομαγνητών: A και A', B και B', C και C', D και D'. Για να κινηθεί ο δρομέας εφαρμόζεται διαδοχικά σε κάθε ένα από τα παραπάνω ζεύγη μια τάση ηλεκτρικού ρεύματος. Όταν το ηλεκτρικό ρεύμα μεταφέρεται από το ένα ζεύγος ηλεκτρομαγνητών στο διπλανό του ο ρότορας μετατοπίζεται κατά $1,8^\circ$ λόγω των μαγνητικών δυνάμεων που εφαρμόζονται σε αυτόν. Η γωνία αυτή ονομάζεται βήμα του κινητήρα.

2.1.1 Στατική:

Για ένα κινητήρα ο οποίος στρέφεται με βήμα S , το διάγραμμα ροπής ως προς τη γωνιακή θέση του ρότορα σε σχέση με κάποια αρχική θέση αναφοράς, σε γενικές γραμμές, προσεγγίζει το ημίτονο. Στην πράξη, η μορφή της καμπύλης εξαρτάται από την γεωμετρία των πόλων του στάτη και του δρομέα. Στην πιο κοινή περίπτωση, δηλαδή του κινητήρες μόνιμου μαγνήτη, δύο τυλιγμάτων η περίοδος θα είναι ίση με $4S$, όπως φαίνεται στο παρακάτω διάγραμμα:



Η μαθηματική έκφραση του διαγράμματος είναι:

$$T = -h \sin((\pi/2) / S) \theta$$

Όπου:

T: ροπή

h: ροπή συγκράτησης ή μέγιστη ροπή

S: γωνία βήματος

θ: γωνία άξονα

Στο σημείο αυτό πρέπει να αναφερθεί η διαφορά ανάμεσα στην ηλεκτρική και τη μηχανική γωνία άξονα. Όταν αναφερόμαστε στη μηχανική γωνία άξονα θ , 2π ακτίνια ισοδυναμούν με μία πλήρη περιστροφή του άξονα. Όταν αναφερόμαστε στην ηλεκτρική γωνία, μια πλήρης περιστροφή ισοδυναμεί με την περίοδο της γραφικής παράστασης ροπής-γωνίας.

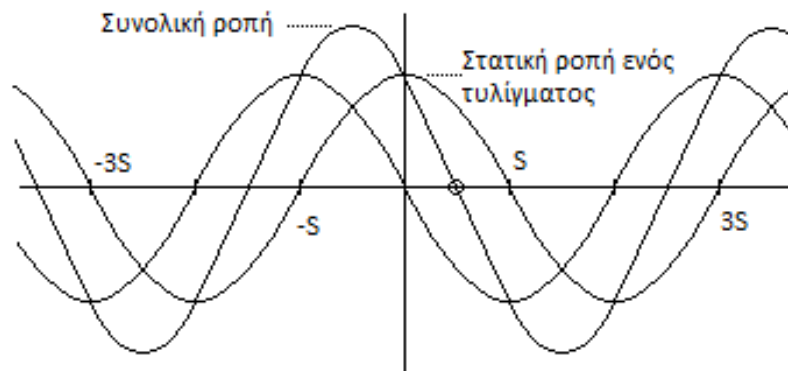
Υποθέτοντας ότι η καμπύλη ροπής-γωνίας προσεγγίζει το ημίτονο, όσο η ροπή παραμένει χαμηλότερη –κατά απόλυτη τιμή- από την μέγιστη ροπή του κινητήρα, ο ρότορας θα απέχει το πολύ κατά $\frac{1}{4}$ της περιόδου από το σημείο ισορροπίας (δηλαδή την κορυφή ή την κοιλάδα του ημιτόνου).

Ακόμα και σε περίπτωση που ο κινητήρας δεν τροφοδοτείται η ροπή δεν είναι πάντα μηδενική. Ιδιαίτερα στους κινητήρες δυο τυλιγμάτων, ο συνδυασμός της γεωμετρίας των όλων

με το μόνιμα μαγνητισμένο ρότορα μπορεί να δημιουργεί σημαντική παραμένουσα ροπή σε κατάσταση μηδενικής τροφοδοσίας.

2.1.2 Half-Step και Microstepping

Τροφοδοτώντας τώρα και τα δύο τυλίγματα του κινητήρα ταυτόχρονα, παράγεται μια καμπύλη ροπής-θέσης η οποία ισούται με το άθροισμα των καμπυλών για το κάθε τύλιγμα ξεχωριστά. Για ένα κινητήρα δυο τυλιγμάτων μόνιμου μαγνήτη, οι δυο καμπύλες θα έχουν διαφορά φάσης S , αν και τα δυο τυλίγματα διαρρέονται από το ίδιο ρεύμα. Οι κορυφές και η κοιλιά του αθροίσματος έχουν διαφορά φάσης $S/2$ από τις αρχικές καμπύλες, όπως φαίνεται στο παρακάτω διάγραμμα:



Αυτή είναι και η βάση του half-stepping. Η στατική ροπή της σύνθετης ροπής καμπύλης προκύπτει όταν τα δύο τυλίγματα έχουν αθροιστικά το μέγιστο ρεύμα. Η μέγιστη ροπή δίνεται από τον τύπο:

$$h_2 = 2^{0.5} h_1$$

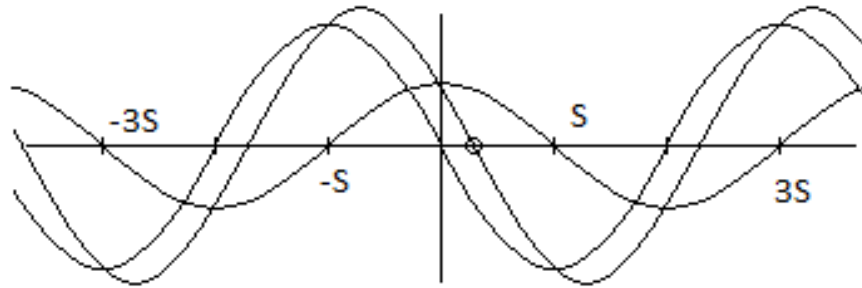
Όπου:

h_1 : μέγιστη ροπή ενός τυλιγματος

h_2 : σύνθετη μέγιστη ροπή δύο τυλιγμάτων

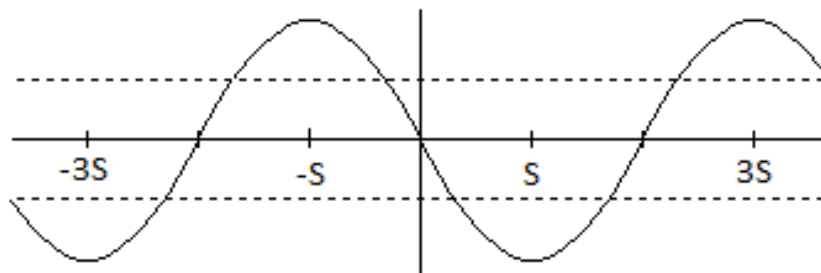
προϋποθέτοντας ότι κανένα μέρος του μαγνητικού κυκλώματος δεν φτάνει σε κορεσμό και ότι η καμπύλη για κάθε τύλιγμα είναι ιδανικό ημίτονο. Αν οποιοδήποτε από τα μαγνητικά μέρη του κινητήρα έρθει σε κορεσμό οι καμπύλες δε θα προστεθούν γραμμικά και κατά συνέπεια η θέση ισορροπίας δε θα βρίσκεται ακριβώς στη μέση των δυο αρχικών θέσεων ισορροπίας.

Η λειτουργία microstepping επιτρέπει στον κινητήρα να πραγματοποιεί ακόμα μικρότερα βήματα, επάγοντας διαφορετικά ρεύματα στα δύο τυλίγματα του κινητήρα, όπως φαίνεται στο διάγραμμα:

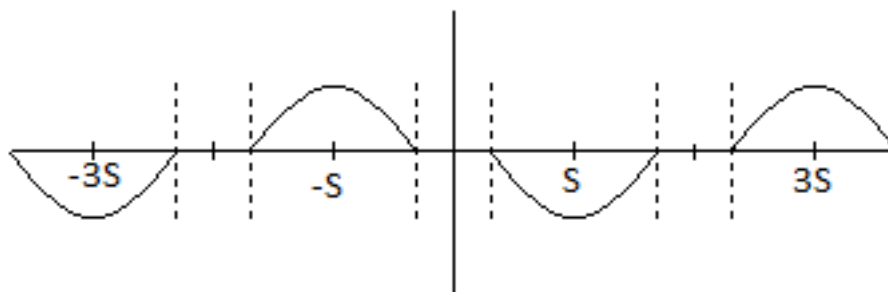


2.1.3 Τριβή και νεκρή ζώνη

Μέχρι τώρα στα διαγράμματα είχαν υποτεθεί ιδανικές καταστάσεις. Στην πραγματικότητα, για να στραφεί ο κινητήρας πρέπει να ξεπεράσει τη ροπή που δημιουργεί η τριβή. Όπως είναι γνωστό από τη μηχανική, υπάρχουν δύο κατηγορίες τριβής: η στατική τριβή και η τριβή ολίσθησης. Εδώ θα ασχοληθούμε με τη στατική τριβή. Υποθέτουμε ότι η ροπή της τριβής που πρέπει να υπερβεί ο κινητήρας είναι η μισή της μέγιστης ροπής του όπως φαίνεται στο διάγραμμα 4:



Η ροπή που απομένει για να κινήσει τον κινητήρα είναι το κομμάτι που βρίσκεται έξω από τις διακεκομμένες γραμμές. Έτσι η ροπή σαν συνάρτηση της γωνίας είναι η διαφορά των δύο καμπυλών:

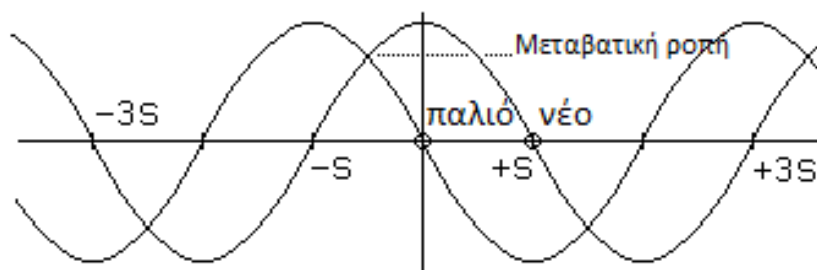


Οι συνέπειες της στατικής τριβής είναι δυο. Πρώτον, η συνολική διαθέσιμη ροπή του κινητήρα μειώνεται, και δεύτερον δημιουργείται μια νεκρή ζώνη σχεδόν σε κάθε σημείο ισορροπίας του κινητήρα. Αν ο ρότορας βρίσκεται μέσα σε νεκρή ζώνη, κατά την ισορροπία του σε κάποιο σημείο ισορροπίας, η ροπή της τριβής εξουδετερώνει την ροπή από τα τυλίγματα του κινητήρα και ο ρότορας δε θα μετακινηθεί.

Το κύριο πρόβλημα με τις νεκρές ζώνες είναι ότι περιορίζουν τη συνολική ακρίβεια. Για παράδειγμα, στην περίπτωση που η ροπή της στατικής τριβής είναι το $\frac{1}{2}$ της μέγιστης ροπής, ένας κινητήρας βήματος 90° θα έχει νεκρές ζώνες πλάτους 60° . Αυτό σημαίνει ότι θα έχουμε βήματα από 30° έως 150° , αναλόγως του σημείου στο οποίο θα σταματήσει ο ρότορας μέσα στη νεκρή ζώνη. Σε περίπτωση, μάλιστα, που το βήμα είναι μικρότερο από τη νεκρή ζώνη ο κινητήρας δεν θα κινηθεί καθόλου.

2.1.4 Δυναμική

Σε κάθε βήμα του κινητήρα έχουμε μετατόπιση του σημείου ισορροπίας κατά S ακτίνια. Έτσι, έχουμε μετακίνηση της καμπύλης κατά S όπως φαίνεται στο παρακάτω διάγραμμα



Το πρώτο πράγμα που παρατηρούμε, είναι ότι κατά τη μετάβαση από το ένα βήμα στο επόμενο η μέγιστη δυνατή ροπή είναι στο ελάχιστο όταν βρισκόμαστε στη μέση της απόστασης. Η ροπή αυτή ονομάζεται μεταβατική και είναι η μέγιστη ροπή που μπορεί να αποδώσει ο

κινητήρας καθώς κινείται. Το μέτρο της για τους κινητήρες δυο τυλιγμάτων, με μέγιστη ροπή h θα είναι $h/\sqrt{2}$.

2.1.5 Ροπή και ταχύτητα

Ως μέγιστη ταχύτητα του κινητήρα αναφέρεται η ταχύτητα κατά την οποία η ροπή του κινητήρα πέφτει στο μηδέν. Η μέγιστη ταχύτητα του κινητήρα είναι δύσκολο να υπολογιστεί, καθώς ο συντονισμός του συστήματος μπορεί να προκαλέσει πρόωρα ελάττωση της ροπής. Η ταχύτητα αποκοπής ορίζεται ως η ταχύτητα πάνω από την οποία η ροπή αρχίζει να ελαττώνεται. Όταν ο κινητήρας λειτουργεί κάτω από αυτή την ταχύτητα, οι αυξομειώσεις του ρεύματος που περνά από τα τυλίγματα του κινητήρα είναι ουσιαστικά ασήμαντα, ενώ καθώς περνάμε την ταχύτητα αποκοπής, η διάρκεια του βήματος του κινητήρα είναι συγκρίσιμη με το άθροισμα των χρόνων των αυξομειώσεων των ρευμάτων.

Βέβαια, για να γίνει λεπτομερής ανάλυση της σχέσης ταχύτητας-ροπής πρέπει να γνωρίζεις σε βάθος τους χρόνους με τους οποίους αυξάνεται και μειώνεται το ρεύμα στα τυλίγματα του κινητήρα. Οι χρόνοι αυτοί δεν είναι σταθεροί, αλλά εξαρτώνται από το σύστημα ελέγχου και για το λόγο αυτό στα data sheets σπάνια δίνονται καμπύλες ροπής ταχύτητας.

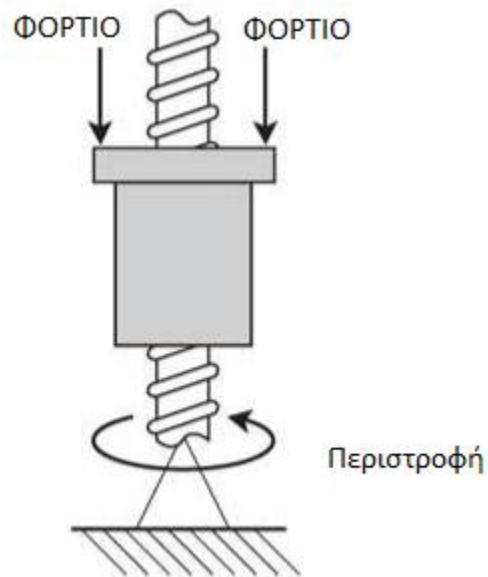
Το παρακάτω διάγραμμα δίνεται η καμπύλη ροπής-ταχύτητας αναφέρεται σε ένα απλό σύστημα ελέγχου



2.2 ΣΠΕΙΡΩΜΑΤΑ ΚΑΙ ΔΥΝΑΜΕΙΣ:

Όπως προαναφέρθηκε στην παρουσίαση της λειτουργίας του ρομπότ, επιλέχθηκε η χρήση αξόνων με σπειρώματα για την μετάδοση της κίνησης από τους βηματικούς κινητήρες στα επιμέρους φορτία στον κάθε άξονα τα οποία ακουμπούν στους άξονες με τη βοήθεια των κατάλληλων παξιμαδιών.

Τα σπειρώματα αυτά, μετατρέπουν την περιστροφική κίνηση του κινητήρα σε γραμμική κατά μήκος των αξόνων, με μια λειτουργία παρόμοια μιας «ελικοειδούς σφήνας» . Έτσι, κρατώντας τα παξιμάδια σταθερά ώστε να μην περιστρέφονται, οι κινητήρες περιστρέφουν τα σπειρώματα και σαν αποτέλεσμα τα παξιμάδια γλιστρούν πάνω στους άξονες



Η κινηματική ανάλυση του συστήματος είναι σχετικά απλή, καθώς το γινόμενο του βήματος της βίδας με τη γωνία περιστροφής (rad) της διαιρούμενο με 2π μας δίνει την μετατόπιση του παξιμαδιού:

$$x = l \frac{\theta}{2\pi} \quad (1)$$

Και παραγωγίζοντας την σχέση (1) προκύπτει η εξίσωση που μας δίνει τη γραμμική ταχύτητα του παξιμαδιού η οποία είναι το γινόμενο του βήματος του σπειρώματος με τη γωνιακή του ταχύτητα (rad/second) διαιρούμενο με 2π :

$$u = l \frac{\omega}{2\pi} \quad (2)$$

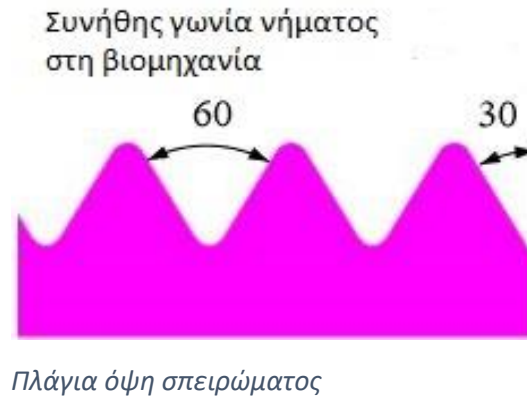
Από την σχέση (1) γίνεται εύκολα αντιληπτό ότι μία πλήρης περιστροφής της βίδας αντιστοιχεί σε μετατόπιση του παξιμαδιού ίση με το βήμα της αν αμελήσουμε το φαινόμενο backlash, το οποίο θα αναλυθεί αργότερα.

Σύμφωνα με την Αρχή Διατήρησης της Ενέργειας, συμπεραίνουμε ότι το έργο που παράγει ο βηματικός κινητήρας, το γινόμενο δηλαδή της απόδοσης του κινητήρα με τη ροπή ($N \cdot m$) και την γωνία περιστροφής, πρέπει να ισούται με το έργο που απαιτείται για την ευθύγραμμη κίνηση του παξιμαδιού, δηλαδή με το γινόμενο της παράλληλης στην κίνηση του δύναμης με τη μετατόπιση του παξιμαδιού.

Έτσι, αν η η απόδοση και T η ροπή του κινητήρα τότε για μια πλήρη περιστροφή έχουμε:

$$F = \frac{\eta \cdot 2\pi \cdot T}{l} \quad (3)$$

Η απόδοση η δεν είναι σταθερά, αλλά αλλάζει ανάλογα με το φορτίο που θα κληθεί να μετακινήσει ο κινητήρας. Όπως είναι λογικό, όσο το βάρος του φορτίου μειώνεται η απόδοση αυξάνεται καθώς η ασκούμενη δύναμη από το βάρος του φορτίου τείνει αναγκαστικά να στρέψει την έλικα προς τα κάτω. Αυτό σημαίνει ότι υπάρχει μια κρίσιμη κατάσταση του συντελεστή τριβής ανάμεσα στο σπείρωμα του παξιμαδιού και της βίδας, και της γεωμετρίας της βίδας στην οποία οριακά το ένα σώμα προκαλεί την περιστροφική κίνηση του άλλου.



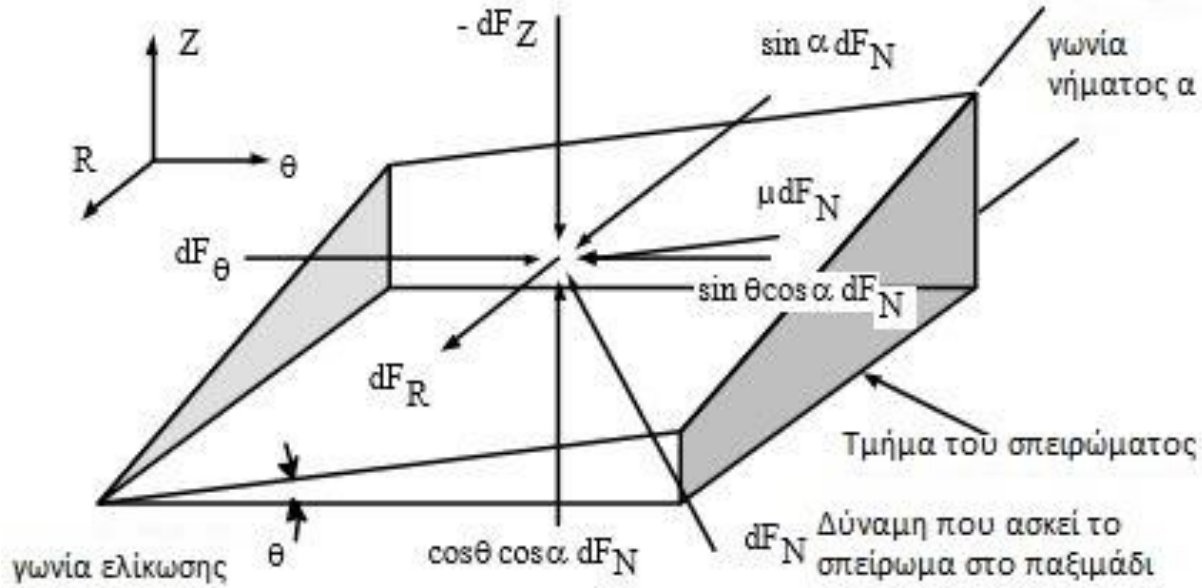
Στην κατάσταση αυτή, δεν μπορεί να υπάρξει κίνηση ανεξαρτήτως της δύναμης που ασκείται αν το βήμα της βίδας είναι μικρότερο από μία κρίσιμη τιμή

$$l \leq \frac{\pi \cdot \mu \cdot D_{pitch}}{\cos \alpha} \quad (4)$$

όπου α η γωνία νήματος που χρησιμοποιείται στα πλαίσια της ανάλυσης, μ ο συντελεστής τριβής και D_{pitch} η διάμετρος του σπειρώματος της βίδας.

Στην πραγματικότητα, η δύναμη που παράγει η ροπή του κινητήρα δεν πρέπει να ξεπεράσει μόνο την τριβή που αναπτύσσεται ανάμεσα στα σπείρωματα, αλλά και την τριβή που αναπτύσσεται ανάμεσα στα ρουλεμάν και τους οδηγούς που εξασφαλίζουν την ευθύγραμμη κίνηση. Η τριβή αυτή, βέβαια, περιορίζεται σε μεγάλο βαθμό με τη χρήση λιπαντικού ώστε να είναι πρακτικά αμελητέα.

Στο παρακάτω σχήμα φαίνονται αναλυτικά οι δυνάμεις που ασκούνται στο σπείρωμα κατά την περιστροφή:



Η απόδοση του κινητήρα κατά την άνοδο και την κάθοδο του φορτίου δίνονται από τους τύπους: η_{raise} η_{lower}

$$\eta_{\text{raise}} = \frac{(\cos\alpha - \mu\beta/\pi)}{(\cos\alpha - \pi\mu/\beta)} \quad (5)$$

$$\eta_{\text{lower}} = \frac{(\cos\alpha + \mu\beta/\pi)}{(\pi\mu/\beta - \cos\alpha)} \quad (6)$$

όπου:

$$\beta = l / D_{\text{pitch}} \quad (7)$$

3 ΚΑΤΑΣΚΕΥΗ ΜΗΧΑΝΟΛΟΓΙΚΟΥ ΤΜΗΜΑΤΟΣ:

Η κατασκευή του μηχανολογικού τμήματος έγινε σε τέσσερα διακριτά στάδια:

1. Το πλαίσιο και η βάση του καρτεσιανού ρομπότ
2. Τοποθέτηση σπειρωμάτων και ρουλεμάν για τη μεταφορά του άξονα
3. Κατασκευή του εγκάρσιου άξονα
4. Βάσεις για τη στήριξη των φορέων καλωδίων

Η ιδιαιτερότητα της συγκεκριμένης εφαρμογής είναι το μεγάλο, σε σύγκριση με άλλες παρόμοιες εφαρμογές, μέγεθος του ρομπότ. Έτσι, αν και το αυξανόμενο ενδιαφέρον για εφαρμογές όπως οι 3D εκτυπωτές θεωρητικά καθιστά εύκολη την εύρεση των απαραίτητων μηχανολογικών τμημάτων, στη συγκεκριμένη περίπτωση χρειάστηκε σε πολλά σημεία να ξεφύγουμε από τα ήδη υπάρχοντα προϊόντα στο εμπόριο, διότι δεν ήταν λειτουργικά, ή ήταν ιδιαίτερα κοστοβόρα.

3.1 ΤΟ ΠΛΑΙΣΙΟ ΤΟΥ ΡΟΜΠΟΤ

Για την κατασκευή του πλαισίου του ρομπότ χρησιμοποιήθηκαν τέσσερις στραντζαριστοί γαλβανισμένους σωλήνες τετραγωνικής διατομής. Η επιλογή, αυτή, έναντι των προφίλ αλουμινίου, που χρησιμοποιούνται συνηθέστερα έγινε για την εξοικονόμηση βάρους και υλικού.



Κάτοψη ρομπότ

Η στήριξη του ρομπότ σε οριζόντια θέση γίνεται με τη βοήθεια τεσσάρων μεταλλικών στηριγμάτων Φ50.



Για τη στήριξη του ρομπότ σε όρθια θέση κατασκευάστηκαν όμοιες βάσεις από στραντζαριστό μέταλλο. Αναλυτικά οι διαστάσεις των κομματιών που χρησιμοποιήθηκαν βρίσκονται στη λίστα των υλικών στο τέλος του κεφαλαίου.



3.2 ΤΟΠΟΘΕΤΗΣΗ ΣΠΕΙΡΩΜΑΤΩΝ ΚΑΙ ΡΟΥΛΕΜΑΝ ΓΙΑ ΤΗ ΜΕΤΑΦΟΡΑ ΤΟΥ ΑΞΟΝΑ

Τα σπειρώματα που χρησιμοποιήθηκαν είναι δύο ντίζες $\Phi 16$ μήκους 150 εκατοστών. Οι βάσεις πάνω στις οποίες στηρίζονται οι ντίζες είναι 2 Y-bearing plummers διατομής 12mm για κάθε σπείρωμα.



Y-bearing plummers

Στο ένα άκρο της ντίζας τοποθετήθηκε η βάση για το βηματικό κινητήρα καθώς και το κόπλερ μετάδοσης κίνησης



Βάση στήριξης βηματικού κινητήρα

Για τη σύνδεση του εγκάρσιου άξονα με τα σπειρώματα χρησιμοποιήθηκαν τετράγωνα παξιμάδια 16mm



Τέλος οι βάσεις που εφάπτονται στους οδηγούς έγιναν από πλάκες πολυαμιδίου

Στην παρακάτω εικόνα φαίνεται ο άξονας Y



Άξονας Y

Παράλληλα με τον οδηγό και το σπείρωμα τοποθετήθηκε μια μεταλλική βάση πάνω στην οποία θα στηριχθεί ο φορέας καλωδίων



Στην παρακάτω εικόνα φαίνεται η τελική μορφή του άξονα Y



Άξονας Y

3.3 ΚΑΤΑΣΚΕΥΗ ΕΓΚΑΡΣΙΟΥ ΑΞΟΝΑ ΚΑΙ ΚΕΦΑΛΗΣ

Ο εγκάρσιος άξονας είναι ο άξονας πάνω στον οποίο κινείται η κεφαλή. Το κομμάτι αυτό της κατασκευής περιλαμβάνει, όπως και το στάδιο 2, μία ντίζα μήκους 125 εκατοστών, ένα στραζαριστός τετραγωνικός σωλήνας μήκους επίσης 125 εκατοστών, καθώς επίσης και 2 Y-bearing plumpers για τη στήριξη του σπειρώματος. Και η κατασκευή του είναι παρόμοια με αυτή του βήματος 2

Η κεφαλή κατασκευάστηκε επίσης από πλάκα πολυαμιδίου, όπως οι βάσεις του άξονα X. Για τη μετάδοση κίνησης στην κορυφή της κεφαλής χρησιμοποιήθηκε ακόμα ένα τετράγωνο παξιμάδι 16mm.



Κεφαλή του ρομπότ

3.4 ΛΙΣΤΑ ΥΛΙΚΩΝ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ

3 στρατζαριστά 40mm x 40mm μήκους 150cm
2 στρατζαριστά 40mm x 40mm μήκους 125cm
6 τάπες 40mm x 40mm
2 ντίζες M16x2.0 μήκους 150cm
1 ντίζα M16x2.0 μήκους 130cm
6 Y-bearing plummers Φ12
14 βίδες M10x30
14 παξιμάδια M10
24 ροδέλες M10
3 βίδες M8x30
3 παξιμάδια M8
4 τεμάχια στρατζαριστό 30mm x 30 mm μήκους 20 cm
2 τεμάχια στρατζαριστό 40mm x 20 mm μήκους 25 cm
4 τάπες 30mm x 30mm
2 τάπες 40mm x 20mm
2 μεταλλικές γωνίες 70mm x 35mm
2 λάμες 40mm x 100mm
1 μεταλλική βάση 140mm x 5mm
1 μεταλλική βάση 120mm x 5mm
4 μεταλλικά στηρίγματα Φ50
2 μεταλλικές λάμες 4mm x 12mm
1 μεταλλική γωνία 5cm x 10cm
2 αρσενικές πεταλούδες M6x20
230cm πλαστικός φορέας καλωδίων

4 ΗΛΕΚΤΡΙΚΗ -ΗΛΕΚΤΡΟΝΙΚΗ ΥΠΟΔΟΜΗ ΚΑΙ ΣΧΕΔΙΑΣΗ

Η ηλεκτρική και ηλεκτρονική σχεδίαση έγινε σε δύο στάδια.

Το πρώτο και βασικό στάδιο περιλαμβάνει:

- 1 Arduino Mega 2560
- 3 βηματικούς κινητήρες 57BYGH420
- 3 stepper motor drivers TB6560
- 1 τροφοδοτικό 12V/8A

Κατά τη διάρκεια της κατασκευής κρίθηκε σκόπιμο να χρησιμοποιηθούν περιφερειακά βοηθήματα, τα οποία διευκολύνουν τόσο την πειραματική διαδικασία, αλλά και την αλληλεπίδραση του χρήστη με το ρομπότ. Στο δεύτερο στάδιο, λοιπόν, προστέθηκαν στην κατασκευή:

- 1 οθόνη High Contrast 16x2 Character LCD
- 1 πληκτρολόγιο membrane 4x4 Matrix
- 1 SparkFun Thumb Joystick

4.1 ΒΗΜΑΤΙΚΟΙ ΚΙΝΗΤΗΡΕΣ

Στις εφαρμογές καρτεσιανών ρομπότ χρησιμοποιούνται επί το πλείστο βηματικοί κινητήρες καθώς παρουσιάζουν πλεονεκτήματα που δεν έχει κανένα άλλο είδος ηλεκτρικού κινητήρα.



Βηματικός κινητήρας 57bygh420

Τα πλεονεκτήματα αυτά είναι:

- Σε αντίθεση με τους κινητήρες συνεχούς ρεύματος, δεν χρειάζεται φρένα για να μείνει ακίνητος ή για να επιβραδυνθεί.
- Στις μικρές ταχύτητες περιστροφής, αλλά και κατά την εκκίνησή του, παράγει μεγάλες τιμές ροπής.

- Δεν απαιτείται χρήση αισθητήρων και κυκλωμάτων ανάδρασης για τον προσδιορισμό της θέσης του άξονα κίνησης, αφού αυτή προσδιορίζεται εύκολα μετρώντας τα βήματα που έχει εκτελέσει ο κινητήρας

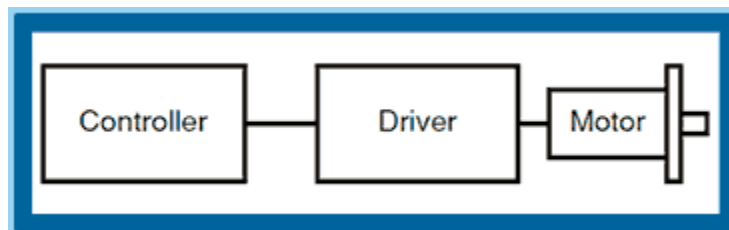
Παρόλα αυτά, υπάρχουν και κάποια μειονεκτήματα των βηματικών κινητήρων τα οποία πρέπει να ληφθούν υπόψιν:

- Θορυβώδης λειτουργία.
- Κατά τη μετακίνηση φορτίων μεγάλης μάζας υπάρχει ο κίνδυνος να μη σταματήσει ακαριαία ο κινητήρας, λόγω της αυξημένης αδράνειας.

Οι κινητήρες που χρησιμοποιήθηκαν είναι τύπου NEMA23 και χρησιμοποιούνται κατά κόρων σε κατασκευές CNC. Οι συγκεκριμένοι κινητήρες λειτουργούν με μέγιστο ρεύμα 2A και τάση 3.6V. Η μέγιστη ροπή τους είναι 9kg/cm και η γωνία του βήματος 1.8°. Αναλυτικά τα χαρακτηριστικά του, καθώς και το datasheet παρουσιάζονται στο παράρτημα.

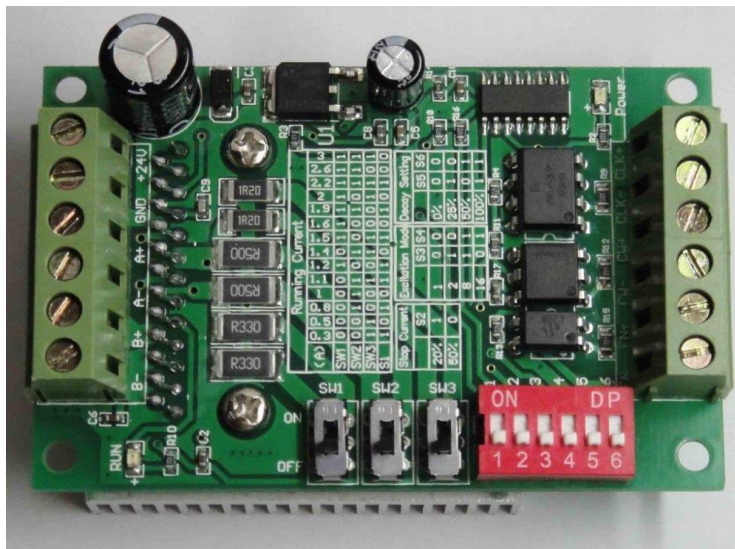
Χαρακτηρίζονται ως υβριδικοί βηματικοί κινητήρες υψηλής ροπής, δηλαδή μπορούν να λειτουργήσουν είτε ως μονοπολικοί είτε ως διπολικοί, και έχουν έξι καλώδια που ακολουθούν χρωματικό κώδικα. Όταν χρησιμοποιούνται ως μονοπολικοί κινητήρες γίνεται χρήση και των έξι καλωδίων, ενώ στην παρούσα εφαρμογή ως διπολικοί κινητήρες χρησιμοποιούνται μόνο τα τέσσερα εξ αυτών.

4.2 DRIVERS ΒΗΜΑΤΙΚΩΝ ΚΙΝΗΤΗΡΩΝ



Στην παραπάνω εικόνα φαίνεται ο ρόλος των drivers στον έλεγχο των κινητήρων. Οι drivers αποτελούν το δίαυλο επικοινωνίας των κινητήρων με τον μικροελεγκτή. Ο ρόλος των driver είναι να ελέγχουν το μέτρο και τη διεύθυνση του ρεύματος που διαρρέει τα τυλίγματα του κινητήρα. Ο driver παίρνει σαν είσοδο τους παλμούς από τον μικροελεγκτή και καθορίζει με ποιον τρόπο θα ενεργοποιηθούν τα τυλίγματα του κινητήρα με κατάλληλη αλληλουχία ώστε να παραχθεί κίνηση .

Ο driver που επιλέχθηκε για την κατασκευή είναι ο TB6560, ο οποίος φαίνεται στην εικόνα. Ο TB6560 δίνει πολλές επιλογές και μεγάλη ευελιξία στον έλεγχο του κινητήρα αφού προσφέρει δυνατότητα για full stepping, half stepping, αλλά και micro stepping μέχρι και στο 1/16 του πλήρους βήματος

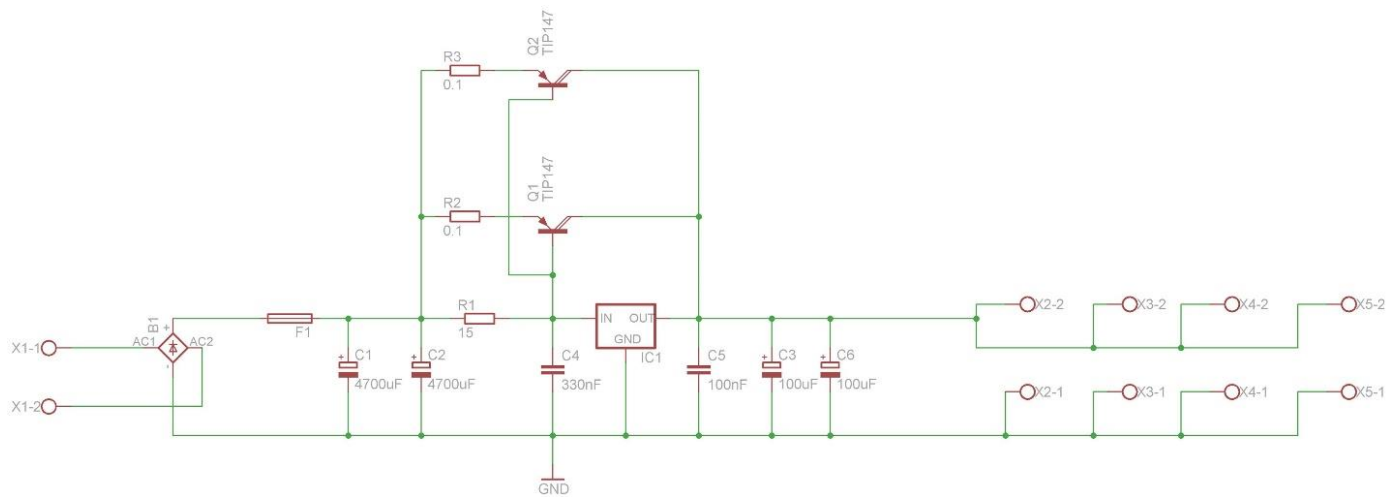


Stepper Motor Driver TB6560

4.3 ΤΡΟΦΟΔΟΤΙΚΟ 12V/8A

4.3.1 Ανάλυση κυκλώματος

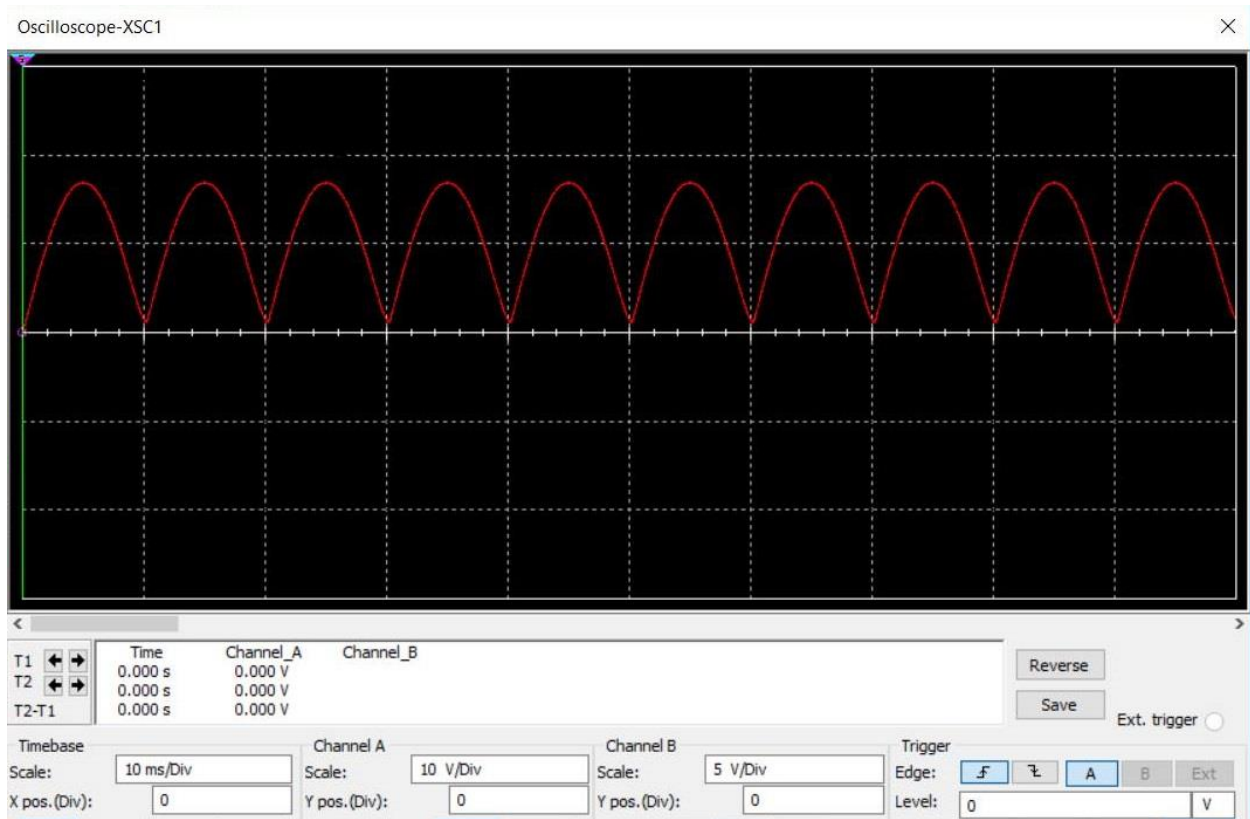
Για την τροφοδοσία των τριών TB6560, κατασκευάστηκε τροφοδοτικό 12V/8A, η κυκλωματική διάταξη του οποίου φαίνεται στην παρακάτω εικόνα:



Το κύκλωμα αποτελείται από πέντε στάδια:

1. Το στάδιο ανόρθωσης:

Το στάδιο ανόρθωσης αποτελείται από μία γέφυρα διόδων. Η είσοδός του είναι η AC τάση 12 Volt που λαμβάνεται από το δευτερεύον του μετασχηματιστή και στην έξοδό του λαμβάνου με το σήμα ανορθωμένο



Κυματομορφή εξόδου πρώτου σταδίου

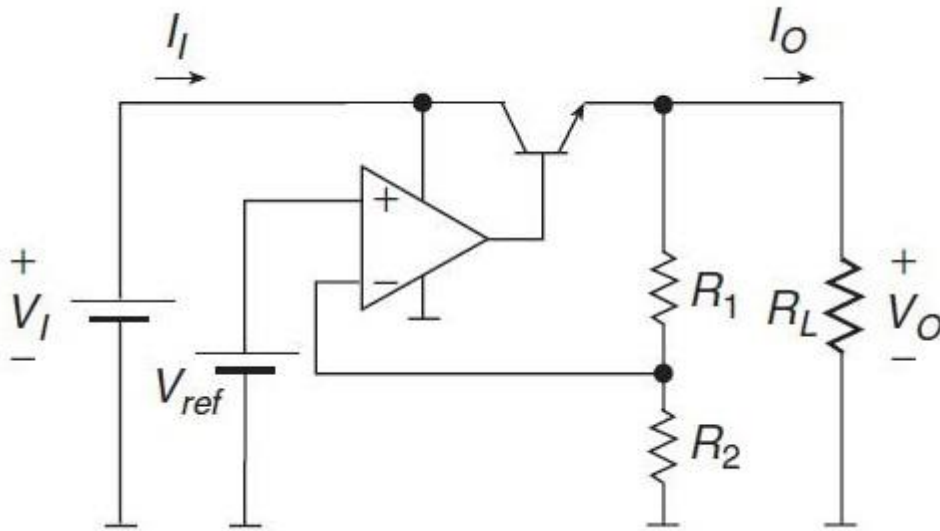
2. Φίλτρο εξομάλυνσης

Το φίλτρο εξομάλυνσης αποτελείται από δυο ηλεκτρολυτικούς πυκνωτές συνδεδεμένους παράλληλα. Το στάδιο αυτό περιορίζει τις μεγάλες διακυμάνσεις του σταδίου ανόρθωσης σε μεγάλο ποσοστό, εξαλείφοντας σχεδόν εξ' ολοκλήρου τις κυματώσεις.



3. Σταθεροποιητής τάσης

Ο σταθεροποιητής τάσης είναι ένα ολοκληρωμένο κύκλωμα του οποίου ο ρόλος είναι να διατηρεί σταθερή την τάση στο φορτίο, ανεξάρτητα από μεταβολές της τάσης του δικτύου ή του ρεύματος στο φορτίο. Παράλληλα στο ρυθμιστή έχουν συνδεθεί δυο κεραμικοί πυκνωτές, ένας στην είσοδο του σταδίου ο οποίος αποκόπτει το θόρυβο και ένας στην έξοδό του, ο οποίος είναι υπεύθυνος για την ευστάθεια του ρυθμιστή. Στην παρακάτω εικόνα φαίνεται το κύκλωμα του σταθεροποιητή τάσης



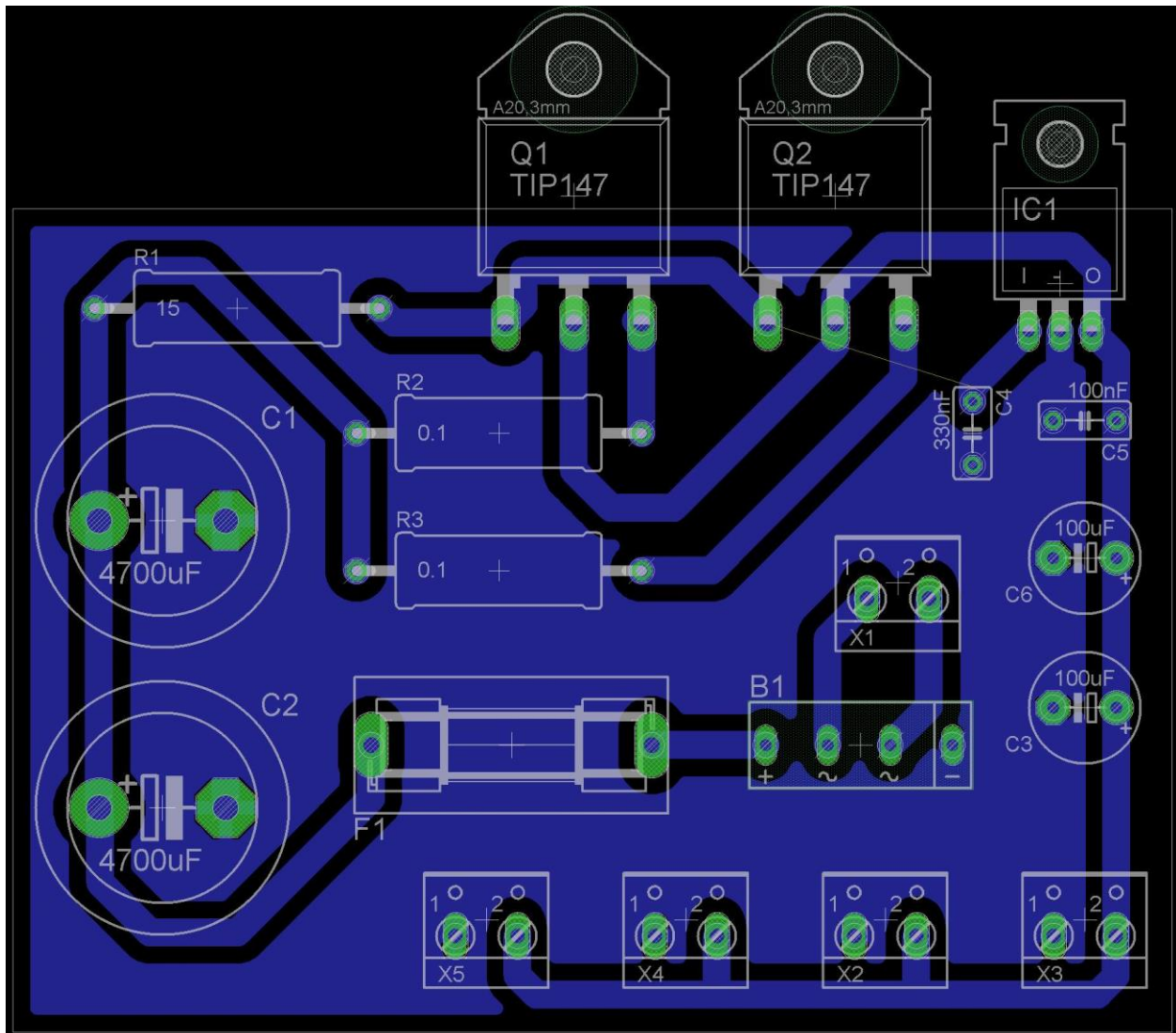
Βασικό κύκλωμα γραμμικού ρυθμιστή τάσης

4. Στάδιο ενίσχυσης ρεύματος

Εδώ χρησιμοποιούνται δυο τρανζίστορ pnp συνδεδεμένα παράλληλα με δύο αντιστάσεις στους εκπομπούς τους ώστε να ισομοιράζεται το ρεύμα και κατά συνέπεια η επιβάρυνση στα τρανζίστορ

5. Στάδιο εξόδου

Στο στάδιο αυτό έχουμε δύο ηλεκτρολυτικούς πυκνωτές συνδεδεμένους παράλληλα οι οποίοι χρησιμοποιούνται για σταθεροποίηση σε απότομες μεταβολές του ρεύματος.



Κάτω όψη πλακέτας τροφοδοτικού

4.3.2 Διαδικασία κατασκευής

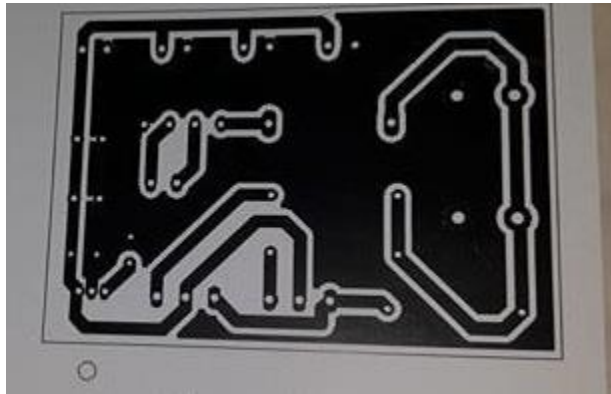
Τα υλικά που χρησιμοποιήθηκαν κατά τη διάρκεια της κατασκευής της πλακέτας είναι:

1. Διάλυμα τριχλωριούχου σιδήρου
2. Πλακέτα με χαλκό
3. Εκτυπωτής toner
4. Φωτογραφικό χαρτί (semiglossy)
5. Σχέδιο πλακέτας
6. Ηλεκτρικό σίδερο
7. Χαρτοταινία

Παρακάτω γίνεται αναλυτική περιγραφή των βημάτων που ακολουθήθηκαν με τη βοήθεια φωτογραφιών

Βήμα 1:

Εκτύπωση του κυκλώματος του τροφοδοτικού σε φωτογραφικό χαρτί με χρήση εκτυπωτή toner



Βήμα 2:

Μεταφορά του μελανιού από το φωτογραφικό χαρτί πάνω στην πλακέτα με τη βοήθεια ηλεκτρικού σίδερου



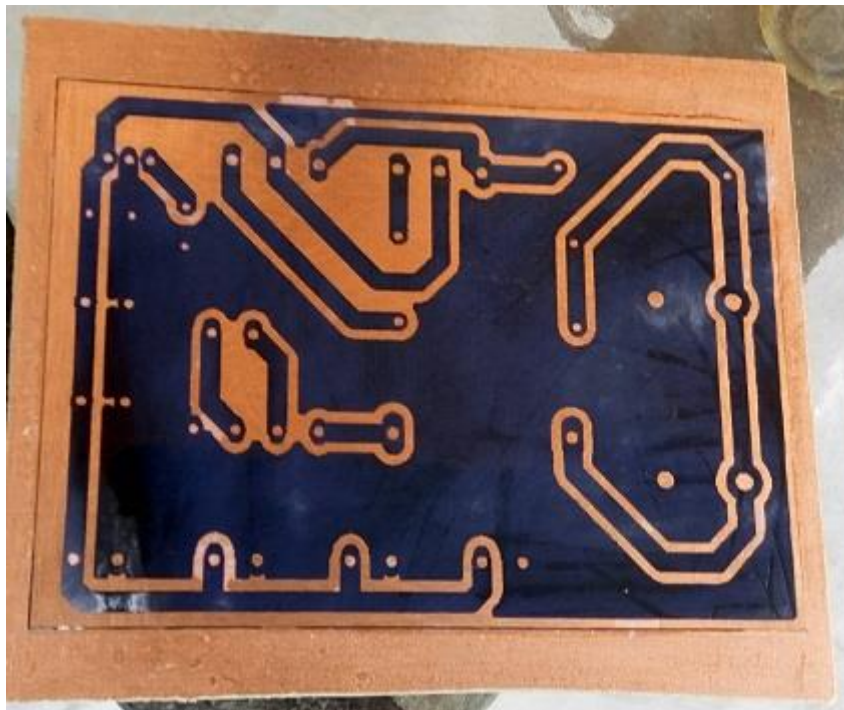
Βήμα 3

Βύθιση της πλακέτας σε χλιαρό νερό



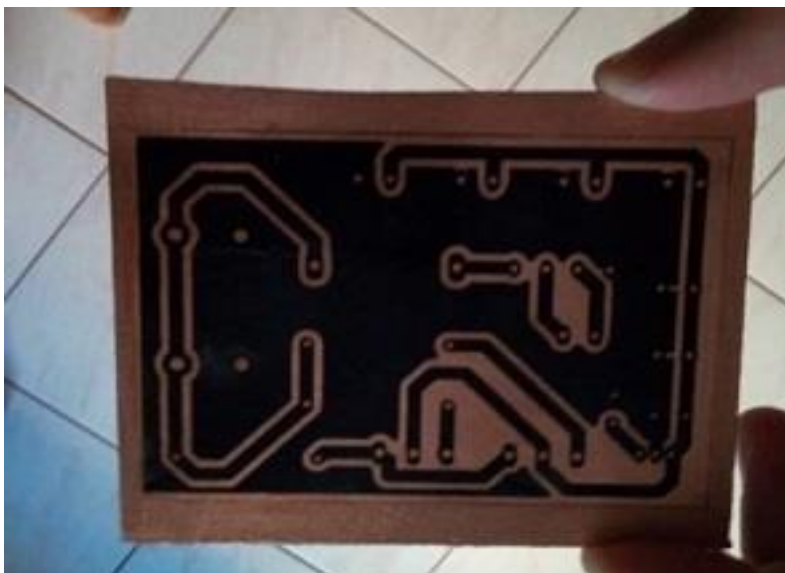
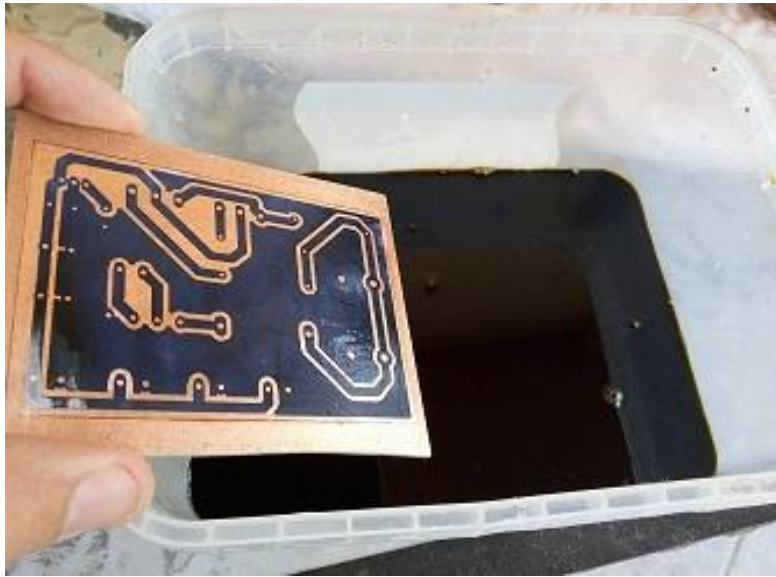
Βήμα 4:

Απομάκρυνση του φωτογραφικού χαρτιού



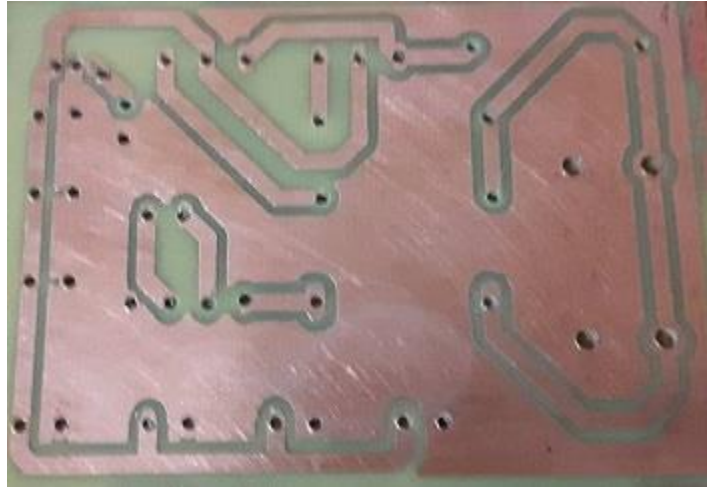
Βήμα 5:

Βύθιση της πλακέτας σε διάλυμα τριχλωριούχου σιδήρου, ώστε να γίνει η αποχάλκωση στα σημεία που δεν καλύπτονται από μελάνι



Βήμα 6:

Καθαρισμός πλακέτας από το μελάνι και τρύπημα πλακέτας



Τα στοιχεία που χρησιμοποιήθηκαν για την κατασκευή του τροφοδοτικού είναι:

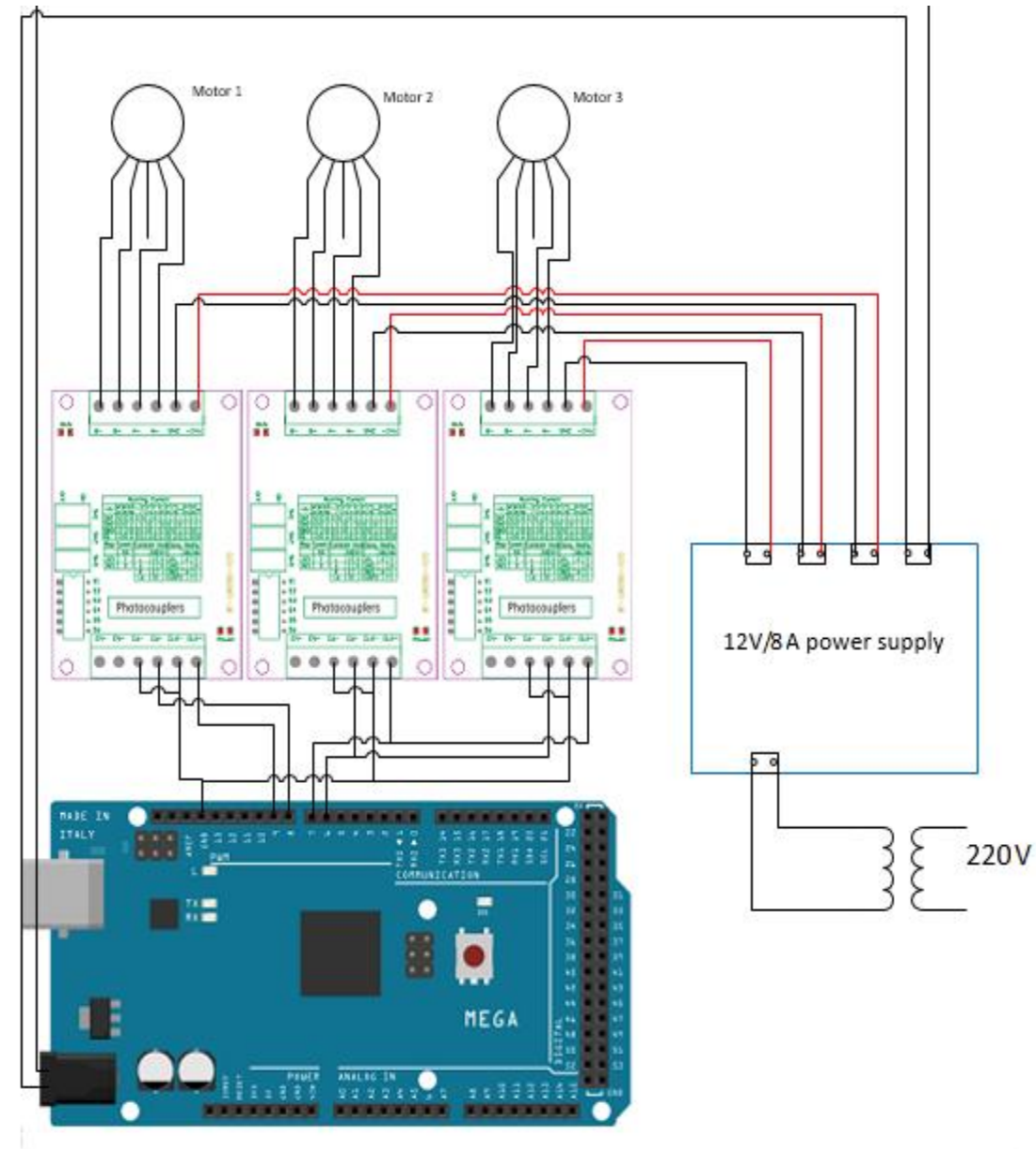
- 5 κλέμες πλακέτας 10mm 2p
- 2 ηλεκτρολυτικοί πυκνωτές 4700μF/50V
- 1 αντίσταση 15Ω/2 Watt
- 2 αντιστάσεις 0,1Ω/2 Watt
- 2 TIP147
- 1 κεραμικός πυκνωτής 100nF
- 1 κεραμικός πυκνωτής 330nF
- 2 ηλεκτρολυτικοί πυκνωτές 100nF/50V
- 1 γέφυρα ανόρθωσης 8A
- 1 ρυθμιστής τάσης
- 1 ψήκτρα αλουμινίου

Και έτσι έχουμε την τελική μορφή του τροφοδοτικού, όπως φαίνεται στην παρακάτω εικόνα:



Τελική μορφή τροφοδοτικού

Παρακάτω φαίνεται το ηλεκτρολογικό σχέδιο από την τροφοδοσία μέχρι τους τρεις κινητήρες



4.4 ΠΕΡΙΦΕΡΕΙΑΚΑ ΒΟΗΘΗΜΑΤΑ

4.4.1 Οθόνη LCD 16x2



LCD 16x2

Η οθόνη που χρησιμοποιήθηκε, είναι οθόνη δυο γραμμών και 16 στηλών, και χρησιμοποιεί τη διεπιφάνεια HD447080. Στο πίσω μέρος, έχει 16 pins από τα οποία όμως, για το σκοπό της εργασίας χρησιμοποιήσαμε τα 12. Ο ρόλος του κάθε pin αναλύεται παρακάτω.

PIN 1: Συνδέεται στη γείωση του Arduino

PIN 2: Συνδέεται στην τάση 5V του Arduino

PIN 3: Ελέγχει την αντίθεση της οθόνης μέσω μεταβλητής αντίστασης 10kΩ

PIN 4: Επιλέγει καταχωρητή εντολών όταν η τάση είναι χαμηλή, και καταχωρητή δεδομένων όταν είναι υψηλή.

PIN 5: Χρησιμοποιείται για ανάγνωση από την οθόνη. Στην περίπτωση μας γειώνεται

PIN 6: Ενεργοποιεί τα υπόλοιπα pins

PIN 7: Δεν χρησιμοποιείται

PIN 8: Δεν χρησιμοποιείται

PIN 9: Δεν χρησιμοποιείται

PIN 10: Δεν χρησιμοποιείται

PIN 11: pin δεδομένων 8-bit. Συνδέεται σε digital pin του Arduino

PIN 12: pin δεδομένων 8-bit. Συνδέεται σε digital pin του Arduino

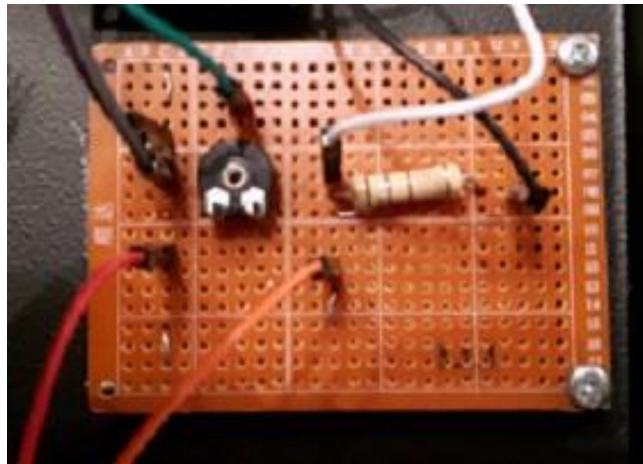
PIN 13: pin δεδομένων 8-bit. Συνδέεται σε digital pin του Arduino

PIN 14: pin δεδομένων 8-bit. Συνδέεται σε digital pin του Arduino

PIN 15: Ενεργοποιεί το backlight της οθόνης. Συνδέεται μέσω αντίστασης στα 5V

PIN 16: Ενεργοποιεί το backlight της οθόνης. Συνδέεται στη γείωση του Arduino

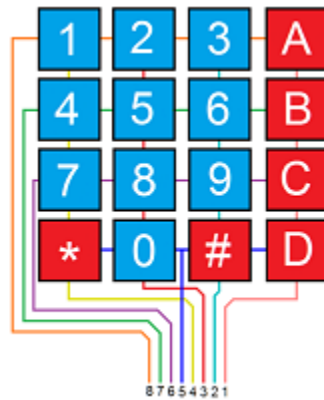
Δεδομένου ότι η οθόνη χρειάζεται τάση 5V και γείωση σε περισσότερα από ένα pins, χρησιμοποιήθηκε μια βοηθητική πλακέτα που βραχυκυκλώνει τα pins 2 και 15 και τα pins 1, 5 και 16. Στην πλακέτα υπάρχει, επίσης ένα trimmer 10kΩ του οποίου τα δύο άρα συνδέονται στην τάση 5V του Arduino και τη γείωση. Το wiper του trimmer συνδέεται στο pin 3 της οθόνης. Τέλος, παρατηρούμε μια αντίσταση 100Ω μεταξύ της τάσης και του pin 16 για την ενεργοποίηση του backlight της οθόνης.



Βοηθητική πλακέτα οθόνης

4.4.2 Πληκτρολόγιο Membrane 4x4 Matrix

Το πληκτρολόγιο που χρησιμοποιήθηκε είναι πληκτρολόγιο 16 πλήκτρων και χρησιμοποιεί ένα συνδυασμό τεσσάρων γραμμών και τεσσάρων στηλών για να παρέχει στο μικροελεγκτή δεκαέξι καταστάσεις, χρησιμοποιώντας συνολικά οκτώ από τις ψηφιακές εισόδους του. Κάτω από κάθε πλήκτρο υπάρχει ένας διακόπτης του οποίου το ένα άκρο συνδέεται σε μία γραμμή και το άλλο σε μία στήλη. Οι συνδέσεις αυτές φαίνονται στην παρακάτω εικόνα:



Για να προσδιορίσει ο μικροελεγκτής, ποιο πλήκτρο έχει πιεστεί εξετάσει την κατάσταση –ψηλά ή χαμηλά- κάθε στήλης, και στη συνέχεια την κατάσταση κάθε γραμμής. Έτσι, έχει τη δυνατότητα να προσδιορίσει ποιο πλήκτρο έχει πιεστεί.

4.4.3 Sparkfun Thumb Joystick



Το Joystick, ως περιφερειακό βοήθημα, επιτρέπει την κίνηση σε δύο διαστάσεις ανάλογα με τη φυσική θέση του Joystick διαβάζεται ως αναλογικό σήμα από το Arduino. Οι τιμές αυτές έχουν εύρος από 0 έως 1023 για κάθε άξονα. Το Arduino έχει τη δυνατότητα να αναγνωρίσει με μεγάλη ακρίβεια τη θέση στην οποία βρίσκεται το joystick.

Ορίζουμε, λοιπόν, τις ζώνες 0 έως 400 ως την αρνητική φορά των αξόνων, τις ζώνες 600 έως 1023 την θετική φορά και τις τιμές 400 έως 600 την ουδέτερη κατάσταση όπου δε πραγματοποιείται κάποια κίνηση.

Έτσι, μπορούμε να ορίσουμε ότι όταν το joystick βρίσκεται στην προκαθορισμένη θέση του το Arduino θα διαβάσει την τιμή (500,500) και δε θα πραγματοποιηθεί κάποια ενέργεια. Όταν πιέσουμε το joystick δεξιά τότε το Arduino θα διαβάσει την τιμή (1023,500) και η κεφαλή θα κινηθεί κατά την θετική διεύθυνση του άξονα X, και κατ' αντιστοιχία όταν το joystick πιεστεί προς το κάτω τότε θα διαβαστεί η τιμή (500, 0) και η κεφαλή θα κινηθεί κατά την αρνητική φορά του άξονα Y

4.5 ΟΡΓΑΝΩΣΗ ΤΟΥ ΚΟΥΤΙΟΥ

Για την ευκολότερη μετακίνηση, αλλά και την ασφάλεια των συνδέσεων, κατά την υλοποίηση του ηλεκτρονικού σχεδιασμού όλα στοιχεία τοποθετήθηκαν σε κουτί διαστάσεων 30cm x 22cm x 15cm. Στις παρακάτω φωτογραφίες φαίνεται το εσωτερικό του κουτιού, καθώς και η πάνω όψη και η πρόσοψη:



Εσωτερικό κουτιού

Στη φωτογραφία φαίνεται πως από το μετασχηματιστή η τάση οδηγείται στο τροφοδοτικό –πάνω αριστερά- στην έξοδο του οποίου έχουμε 4 κλέμες που τροφοδοτούν παράλληλα με τάση 12V DC τους 3 drivers και το Arduino Mega. Επίσης, με τα πολύχρωμα καλώδια έχουμε τη σύνδεση των drivers με το Arduino, αλλά και τα φics για τη σύνδεση του Arduino με τα περιφερειακά στο καπάκι του κουτιού - άνω δεξιά-.



Πρόσοψη – Διακόπτης ON-OFF

Στην πρόσοψη του κουτιού έχουμε το διακόπτη που συνδέει το ρομπότ με το δίκτυο. Η υποδοχή του διακόπτη είναι υποδοχή κοινού τροφοδοτικού υπολογιστή. Στη δεξιά πλευρά της πρόσοψης έχουμε την τρύπα από την οποία εξέρχονται τα καλώδια που συνδέουν τους drivers με τους κινητήρες.



Καπάκι Κουτιού

Στο καπάκι του κουτιού, τέλος, έχουν βιδωθεί το πληκτρολόγιο, το joystick και η οθόνη.

5 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ARDUINO

Ο μικροελεγκτής είναι ένας μικρός υπολογιστής, σε ένα μικρό ολοκληρωμένο κύκλωμα το οποίο περιλαμβάνει έναν επεξεργαστή, μνήμη, καθώς και περιφερειακά εισόδου/ εξόδου. Οι μικροελεγκτές είναι σχεδιασμένοι για να εκτελούν ενσωματωμένες εφαρμογές σε συσκευές αυτομάτου ελέγχου, όπως συστήματα ελέγχου κινητήρων αυτοκινήτων, τηλεχειριστήρια και ιατρικά εμφυτεύματα. Η εισαγωγή των μικροελεγκτών σε εφαρμογές όπως αυτές που προαναφέρθηκαν οδήγησαν στο να μειωθεί το κόστος, το μέγεθος καθώς και η κατανάλωση ενέργειας έναντι κατασκευών που χρησιμοποιούν μικροεπεξεργαστή, μνήμη και συσκευές εισόδου/εξόδου ξεχωριστά.



Μικροελεγκτής AT MEGA

Έτσι, σήμερα, υπάρχει τεράστια ποικιλία εφαρμογών μικροελεγκτών, η οποία οφείλεται στην μεγάλη απλότητα και ευελιξία που προσφέρουν. Ανάλογα με την εκάστοτε εφαρμογή το μέγεθος, η μνήμη, η τάση τροφοδοσίας καθώς και άλλα χαρακτηριστικά μπορεί να διαφέρουν.

Παρόλα αυτά η επιτυχία και η εξάπλωση των μικροελεγκτών οφείλεται σε κάποια στοιχεία που μοιράζονται όλοι από κοινού:

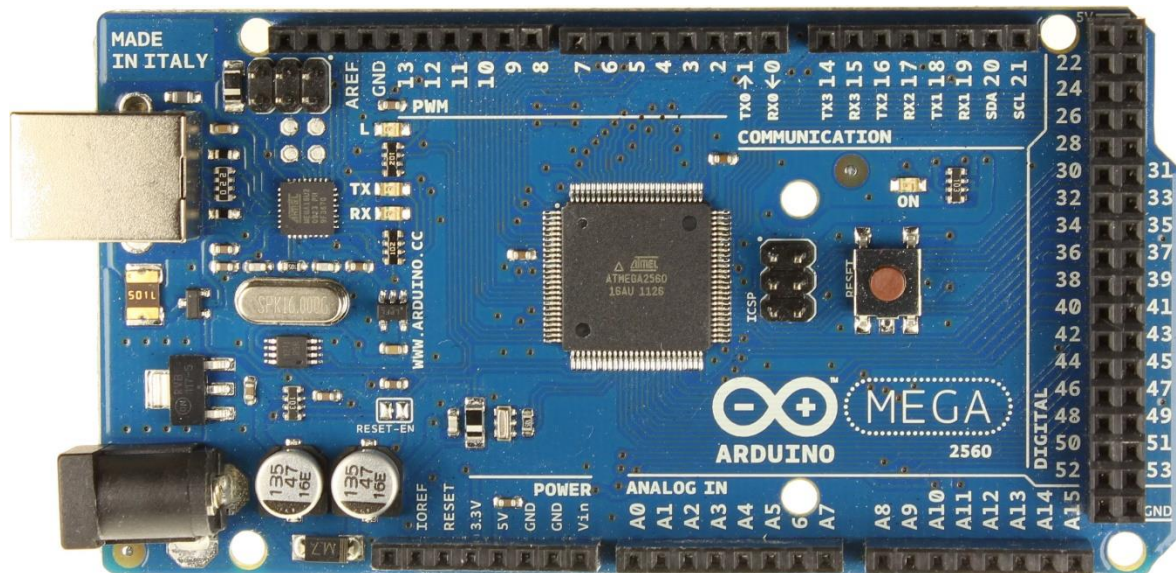
- Ισχυρά και προσεκτικά επιλεγμένα ενσωματωμένα ηλεκτρονικά στοιχεία, τα οποία μπορούν είτε ανεξάρτητα είτε μέσω συσκευών εισόδου-εξόδου να ελέγξουν διαδικασίες και συσκευές.
- Πολύ χαμηλές τιμές που επιτρέπουν την ενσωμάτωσή τους σε συσκευές στις οποίες μέχρι πρότινος δεν ήταν οικονομικά συμφέρον να ενσωματωθεί το οτιδήποτε.
- Χρειάζονται ελάχιστες γνώσεις προγραμματισμού. Ένας ηλεκτρονικός υπολογιστής και ένας μεταγλωττιστής, ο οποίος παρέχεται δωρεάν, είναι αρκετά για την ανάπτυξη οποιασδήποτε εφαρμογής, καθώς υπάρχει τεράστια ποικιλία βιβλιοθηκών, αλλά και έτοιμων προγραμμάτων για να φορτωθούν στην πλακέτα του μικροελεγκτή

5.1 ARDUINO

Το Arduino είναι ένας μικροελεγκτής ανοιχτού κώδικα που χρησιμοποιείται σε μεγάλο εύρος εφαρμογών. Οι πλακέτες Arduino έχουν τη δυνατότητα να διαβάζουν εισόδους- όπως το φως που αντιλαμβάνεται ένας αισθητήρας ή το πάτημα ενός διακόπτη- και να τα μετατρέπει σε εξόδους -να ανάψει μία λάμπα ή να ενεργοποιήσει έναν κινητήρα-. Ο προγραμματισμός του Arduino γίνεται με τη γλώσσα προγραμματισμού Wiring, η οποία είναι βασισμένη στη γλώσσα C++ εμπλουτισμένη με μεγάλο πλήθος βιβλιοθηκών. Παρ' όλα αυτά, ο προγραμματισμός της

πλατφόρμας μπορεί να γίνει και με τη χρήση άλλων γλωσσών, όπως η Python, η C# και η ArduBlock.

Στα πλαίσια αυτής της εργασίας χρησιμοποιήθηκε η πλατφόρμα Arduino Mega 2560. Το Arduino Mega 2560, είναι μια από τις ισχυρότερες πλατφόρμες Arduino και χρησιμοποιεί τον μικροελεγκτή AT Mega 2560. Διαθέτει, επίσης, 54 pins ψηφιακής εισόδου/εξόδου – από τα οποία τα 15 μπορούν να χρησιμοποιηθούν ως έξοδοι PWM, 16 pins αναλογικής εισόδου, ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz μια θύρα τροφοδοσίας καθώς και μια θύρα USB. Το Arduino Mega μπορεί με τον τρόπο αυτό να εκμεταλλευτεί πλήρως την ισχύ του μικροελεγκτή που χρησιμοποιεί



Arduino Mega 2560

Η γλώσσα, που χρησιμοποιήθηκε για την ανάπτυξη του κώδικα, είναι η γλώσσα wiring, δηλαδή η προκαθορισμένη γλώσσα προγραμματισμού μιας πλακέτας Arduino .

5.2 ΔΗΜΙΟΥΡΓΙΑ ΚΩΔΙΚΑ

Όπως αναφέρθηκε και στην εισαγωγή της εργασίας σκοπός αυτής της εφαρμογής ήταν η δημιουργία ενός ρομποτικού βραχίονα που θα πραγματοποιεί μια τετριμμένη διαδικασία. Κατά την διάρκεια της διαδικασίας αυτής, το ρομπότ σαρώνει μια επιφάνεια 1.20 x 1.00 m προχωρώντας με βήμα 5 ή 10 cm έπειτα από επιλογή του χρήστη. Κρίθηκε σκόπιμο, παρ' όλα

αυτά να συμπεριληφθεί στην εργασία και μια λειτουργία κινδύνου για την περίπτωση που χρειαστεί επαναφορά του συστήματος στην αρχική θέση σε περίπτωση επιπλοκής.

Έτσι συμπληρωματικά με τις βιβλιοθήκες των βηματικών κινητήρων, συμπεριλήφθηκαν και βιβλιοθήκες για περιφερειακά βοηθήματα. Πιο συγκεκριμένα χρησιμοποιήσαμε βιβλιοθήκες για:

- Οθόνη LCD 16x2
- Πληκτρολόγιο 16 πλήκτρων
- Joystick

Η βασική δομή των προγραμμάτων για Arduino αποτελείται από 3 διακριτά στάδια, Το τμήμα δηλώσεων, όπου δηλώνονται οι βιβλιοθήκες που χρησιμοποιούνται, όπως επίσης δηλώνονται και αρχικοποιούνται οι μεταβλητές . Το δεύτερο μέρος, ξεκινά με την εντολή void setup() και ο επεξεργαστής το τρέχει μόνο μια φορά κατά την εκτέλεση του προγράμματος. Τέλος το τρίτο μέρος ξεκινάει με την εντολή void loop() και εδώ συναντάμε τον κορμό του προγράμματος. Το τμήμα αυτό επαναλαμβάνεται επ' άοριστον από τον επεξεργαστή μέχρι να σταματήσει η τροφοδοσία του.

5.3 Ο ΚΩΔΙΚΑΣ:

5.3.1 Τμήμα δηλώσεων

Στο πρώτο τμήμα του κώδικα έχουμε όπως είπαμε την κλήση των βιβλιοθηκών, την δήλωση των μεταβλητών και την αντιστοίχιση της οθόνης και του πληκτρολογίου σε εισόδους του Arduino

```
#include <Keypad.h>
#include <LiquidCrystal.h>
#include <Stepper.h>
#define X_pin A0    // Pin A0 connected to joystick x axis
#define Y_pin A1    // Pin A1 connected to joystick y axis

bool diesi_pressed = false;

const int stepsPerRevolution = 800;
Stepper myStepper1(stepsPerRevolution, 50, 51);    //X axis stepper
Stepper myStepper2(stepsPerRevolution, 52, 53);    //Y axis parallel steppers

int direction;
int number1;
int number2;
const byte numRows= 4; //number of rows on the keypad
const byte numCols= 4; //number of columns on the keypad
int i = 0;
int j = 0;
int y = 0;
int x = 0;
int abs_x;
int abs_y;
int direction_x;
int direction_y;

char keypad[numRows][numCols]=
{
{'1', '2', '3', 'A'},
{'4', '5', '6', 'B'},
{'7', '8', '9', 'C'},
{'*', '0', '#', 'D'}
};

byte rowPins[numRows] = {37,35,33,31}; //Rows 0 to 3
byte colPins[numCols]= {29,27,25,23}; //Columns 0 to 3

Keypad myKeypad= Keypad(makeKeypad(keymap), rowPins, colPins, numRows,
numCols);
```

```
LiquidCrystal lcd(32, 30, 28, 26, 24, 22);
```

Στο τέλος του πρώτου τμήματος χρειάστηκε να κατασκευαστεί μια συνάρτηση που μετατρέπει της εισόδους του πληκτρολογίου από χαρακτήρες σε ακέραιους:

```
int GetNumber() // keypad
char input to integer conversion
{
    int num = 0;
    char key = NO_KEY;
    while (key == NO_KEY) {
        key = myKeypad.getKey();
    }
    while (key != 'A')
    {
        if (diesi_pressed) return -1;
        switch (key)
        {
            case NO_KEY:
                //number = 150;
                break;

            case '0': case '1': case '2': case '3': case '4':
            case '5': case '6': case '7': case '8': case '9':
                lcd.print(key);
                num = num * 10 + (key - '0');
                break;

            case '*':
                num = 0;
                lcd.clear();
                break;
        }

        do {
            key = myKeypad.getKey();
        } while (key == NO_KEY);
    }

    return num;
}
```

5.3.2 Void setup()

Στο δεύτερο τμήμα έχουμε την αρχικοποίηση της οθόνης, τα μηνύματα εισόδου και τη δήλωση της δίσησης ως ειδικό γεγονός, πριν οδηγηθούμε στο βασικό μενού. Η δίσηση θα χρησιμοποιείται για να επανεκκινηθεί τον κώδικα

```
void setup() {  
  
    lcd.begin(16, 2);  
    lcd.setCursor(0, 0);  
    lcd.print("JJ project");  
    delay(1500);  
    lcd.clear();  
  
    myKeypad.addEventListener(keypadEvent);  
}  
  
// Taking care of some special events.  
void keypadEvent(KeypadEvent key) {  
    switch (myKeypad.getState()) {  
  
        case PRESSED:  
            if (key == '#') {  
                diesi_pressed = true;  
            }  
            break;  
    }  
}
```

5.3.3 Οι συναρτήσεις

Από εκεί και πέρα πριν το τρίτο μέρος υπάρχουν οι συναρτήσεις που αφορούν τις τρεις πιθανές λειτουργίες του ρομπότ, οι οποίες καλούνται κατά την εκτέλεση του τρίτου μέρους.

Η πρώτη συνάρτηση αφορά τη βασική λειτουργία του ρομπότ, και δίνει στον χρήστη τη δυνατότητα να επιλέξει, με τη βοήθεια του πληκτρολογίου, το βήμα με το οποίο θα σαρωθεί η επιφάνεια

```
void MoveWithStep() {  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Select step ");  
    delay(1500);  
}
```

```

lcd.setCursor(0, 0);
lcd.print("          ");
lcd.print("Press: ");
lcd.setCursor(0, 1);
lcd.print("A for 20-step");
delay(1500);
lcd.setCursor(0, 1);
lcd.print("B for 10-step");
delay(1500);
lcd.clear();

char keypressed = myKeypad.getKey();

if (keypressed != NO_KEY) {

    if (keypressed == 'A')
//step length 5cm
    {
        lcd.print("20-step mode");
        delay(2000);
        lcd.clear();
        lcd.setCursor(0, 1);
        lcd.print("Processing...");
        delay(1500);
        lcd.print("Now at:");
        lcd.print("(");
        lcd.print(String(x));
        lcd.print(",");
        lcd.print(String(y));
        lcd.print(")");

        for (j = 0; j < 24; j++) {

            if (j % 2 == 0) {
                for (i = 0; i < 20; i++) {

                    myStepper1.setSpeed(100);
                    myStepper1.step(20000);
                    x = x + 5;
                    lcd.clear();
                    lcd.print("Now at:");
                    lcd.print("(");
                    lcd.print(String(x));
                    lcd.print(",");
                    lcd.print(String(y));
                    lcd.print(")");
                    delay(2000);

                }
                i = 0;
            }
            else {
                for (i = 0; i < 20; i++) {
                    myStepper1.setSpeed(100);
                    myStepper1.step(-20000);

```

Χρόνης Ιάσων-Ιωάννης

```
x = x - 5;
    lcd.clear();
    lcd.print("Now at:");
    lcd.print("(");
    lcd.print(String(x));
    lcd.print(",");
    lcd.print(String(y));
    lcd.print(")");
    delay(2000);
}
i = 0;
}

myStepper2.setSpeed(100);
myStepper2.step(20000);

y = y + 5;
lcd.clear();
lcd.print("Now at:");
lcd.print("(");
lcd.print(String(x));
lcd.print(",");
lcd.print(String(y));
lcd.print(")");
delay(2000);
}

for (j = 0; j < 12; j++) {
    myStepper2.setSpeed(100);
    myStepper2.step(-20000);
    myStepper2.step(-20000);
}

y = 0;
j = 0;

}

if (keypressed == 'B')
//step length 10 cm
{
    lcd.print("10-step mode");
    delay(2000);
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print("Processing...");
    delay(1500);
    lcd.print("Now at:");
    lcd.print("(");
    lcd.print(String(x));
    lcd.print(",");
    lcd.print(String(y));
    lcd.print(")");

    for (j = 0; j < 12; j++) {
        if (j % 2 == 0) {
            for (i = 0; i < 10; i++) {
```

```

        myStepper1.setSpeed(100);
        myStepper1.step(20000);
        myStepper1.step(20000);
x = x + 10;
        lcd.print("Now at:");
        lcd.print("(");
        lcd.print(String(x));
        lcd.print(",");
        lcd.print(String(y));
        lcd.print(")");
        delay(2000);
    }
    i = 0;
}
else {
    for (i = 0; i<10; i++) {
        myStepper1.setSpeed(100);
        myStepper1.step(-20000);
        myStepper1.step(-20000);
x = x - 10;
        lcd.print("Now at:");
        lcd.print("(");
        lcd.print(String(x));
        lcd.print(",");
        lcd.print(String(y));
        lcd.print(")");
        delay(2000);
    }
    i = 0;
}
}

myStepper2.setSpeed(100);
myStepper2.step(20000);
myStepper2.step(20000);
y = y + 10;
lcd.print("Now at:");
lcd.print("(");
lcd.print(String(x));
lcd.print(",");
lcd.print(String(y));
lcd.print(")");
delay(2000);

}

for (j = 0; j< 12; j++) {
    myStepper2.setSpeed(100);
    myStepper2.step(-20000);
    myStepper2.step(-20000);

}
x = 0;
y = 0;
j = 0;

```


Χρόνης Ιάσων-Ιωάννης

```
    }  
    delay(3000);  
}
```

Η δεύτερη συνάρτηση δέχεται ως είσοδο από το πληκτρολόγιο συντεταγμένες από (0,0) έως (100,120) και οδηγεί την κεφαλή στην δοθείσα θέση

```
void SpecificPosition() { // B  
    diesi_pressed = false;  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Define X:");  
    number1 = GetNumber();  
    if (diesi_pressed) return;  
  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Define Y:");  
  
    number2 = GetNumber();  
    if (diesi_pressed) return;  
  
    if ((number1 <= 100) && (number2 <= 120))  
    {  
        lcd.clear();  
        lcd.setCursor(0, 0);  
        lcd.print("You entered");  
        lcd.setCursor(0, 1);  
        lcd.print("(");  
        lcd.print(String(number1));  
        lcd.print(",");  
        lcd.print(String(number2));  
        lcd.print(")");  
  
        x = number1 * 4000 - x; //  
input translation  
        abs_x = abs(x);  
        direction_x = x / abs(x); // 1 or -1  
  
        myStepper1.setSpeed(100);  
  
        while (abs_x > 32000) {  
            myStepper1.step(direction_x * 32000);  
            abs_x = abs_x - 32000;  
        }  
  
        myStepper1.step(direction_x * abs_x);  
  
        y = number2 * 4000 - y;  
        abs_y = abs(y);  
        direction_y = y / abs(y);  
  
        myStepper2.setSpeed(100);  
  
        while (abs_y > 32000) {
```

```

    myStepper2.step(direction_y * 32000);
    abs_y = abs_y - 32000;
}

myStepper2.step(abs_y);

lcd.clear();

lcd.print("Point Reached");           // current position output
delay(1500);
lcd.clear();
lcd.print("Now at:");
lcd.print("(");
lcd.print(String(number1));
lcd.print(",");
lcd.print(String(number2));
lcd.print(")");

lcd.setCursor(0, 1);
lcd.print("Set new (X,Y)");
delay(1500);
lcd.clear();
}

if ((number1 > 100) || (number2 > 120)) // break condition. out of limits
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Invalid input");
    delay(2000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("try again");
    lcd.clear();
    lcd.print("Give (X,Y)");
    delay(1000);
    lcd.clear();
}
}

void PrintCoord(int x, int y) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(x);

    lcd.setCursor(0, 1);
    lcd.print(y);
    delay(200);
}
}

```

Τέλος, η τρίτη συνάρτηση αφορά τον έλεγχο του joystick, και επιτρέπει στο χρήστη να μεταφέρει την κεφαλή σε οποιοδήποτε σημείο του πλαισίου θέλει με ακρίβεια 2 χιλιοστών. Η λειτουργία αυτή μπορεί να χρησιμοποιηθεί και σε περίπτωση διακοπής λειτουργίας της κίνησης ώστε να επιστρέψει η κεφαλή στην αρχική της θέση.

```
void JoyStick() {
    diesi_pressed = false;
    do {
        myKeypad.getKey();
        if (analogRead(X_pin) >= 0 && analogRead(X_pin) <= 400) {
//Joystick mode--press in case of malfunction

            myStepper1.setSpeed(100);
            myStepper1.step(-800);

            if (analogRead(Y_pin) >= 0 && analogRead(Y_pin) <= 400) {
                lcd.clear();
                lcd.setCursor(0, 0);
                //PrintCoord(analogRead(X_pin), analogRead(Y_pin));

                myStepper2.setSpeed(100);
                myStepper2.step(-800);

            }

            if (analogRead(Y_pin) > 400 && analogRead(Y_pin) <= 600) {
                lcd.clear();
                lcd.setCursor(0, 0);
                //PrintCoord(analogRead(X_pin), analogRead(Y_pin));

            }

            if (analogRead(Y_pin) > 601 && analogRead(X_pin) <= 1024) {
                lcd.clear();
                lcd.setCursor(0, 0);
                //PrintCoord(analogRead(X_pin), analogRead(Y_pin));

                myStepper2.setSpeed(100);
                myStepper2.step(800);

            }

        }

        if (analogRead(X_pin) > 400 && analogRead(X_pin) <= 600) {

            if (analogRead(Y_pin) >= 0 && analogRead(Y_pin) <= 400) {
                lcd.clear();
                lcd.setCursor(0, 0);
                //PrintCoord(analogRead(X_pin), analogRead(Y_pin));

                myStepper2.setSpeed(100);
```

```

        myStepper2.step(-800);
    }

    if (analogRead(Y_pin) > 400 && analogRead(Y_pin) <= 600) {
        myKeypad.getKey();
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Move joystick");
        lcd.setCursor(0, 1);
        lcd.print("# to restart");
        delay(100);

        if (diesi_pressed) return;
    }

    if (analogRead(Y_pin) > 601 && analogRead(X_pin) <= 1024) {
        lcd.clear();
        lcd.setCursor(0, 0);
        //PrintCoord(analogRead(X_pin), analogRead(Y_pin));

        myStepper2.setSpeed(100);
        myStepper2.step(800);
    }

    if (analogRead(X_pin) > 601 && analogRead(X_pin) <= 1024) {

        if (analogRead(Y_pin) >= 0 && analogRead(Y_pin) <= 400) {
            lcd.clear();
            lcd.setCursor(0, 0);

            myStepper2.setSpeed(100);
            myStepper2.step(-800);
        }

        if (analogRead(Y_pin) > 400 && analogRead(Y_pin) <= 600) {
            lcd.clear();
            lcd.setCursor(0, 0);
        }

        if (analogRead(Y_pin) > 601 && analogRead(X_pin) <= 1024) {
            lcd.clear();
            lcd.setCursor(0, 0);
            myStepper2.setSpeed(100);
            myStepper2.step(800);
        }
    }
}

} while (diesi_pressed == false);
}

```

5.3.4 Void loop()

Στο τέλος έχουμε το τρίτο μέρος του προγράμματος δηλαδή το void loop(). Στο κομμάτι αυτό έχουμε το εισαγωγικά μηνύματα του μενού, το μενού και τέλος την κλήση των συναρτήσεων

```
void loop() {
    diesi_pressed = false;

    lcd.clear();
    //mode selection
    lcd.setCursor(0, 0);
    lcd.print("Mode Selection");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Step Mode:");
    delay(1500);
    lcd.setCursor(0, 1);
    lcd.print("                ");
    lcd.setCursor(0, 1);
    lcd.print("Press A");
    delay(1500);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Coord Mode: ");
    delay(1500);
    lcd.setCursor(0, 1);
    lcd.print("Press B");
    delay(1500);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Set Mode:");
    delay(1500);
    lcd.setCursor(0, 1);
    lcd.print("Press C");
    delay(1500);
    lcd.clear();

    lcd.setCursor(0,0);
    lcd.print("You selected: ");

    char keypressed = NO_KEY;
    while (keypressed == NO_KEY) {
        keypressed = myKeypad.getKey();
    }
    lcd.setCursor(0, 1);
    lcd.print(keypressed);

    if (keypressed == 'A' )
    //auto mode on
    {
```

```
        MoveWithStep();
    }
    else if (keypressed == 'B')
//auto mode on
    {
        SpecificPosition();
    }
    else if ( keypressed == 'C' )
    {
        JoyStick();
    }
    else
    {
        lcd.print(" Not Available");
        delay(2000);
    }
}
```

6 ΣΥΜΠΕΡΑΣΜΑΤΑ

Σκοπός αυτής της εργασίας ήταν η κατασκευή ενός ρομπότ για την καταγραφή μετρήσεων αντανάκλασής ήχου πάνω σε δοκίμια σε συνθήκες εργαστηρίου. Η εργασία αυτή αποτελεί μια πολύ καλή εισαγωγή στην κατασκευή των καρτεσιανών ρομπότ, και μπορεί να λειτουργήσει ως βάση για τη δημιουργία άλλων πιο σύνθετων εφαρμογών, συνδυάζοντας μάλιστα προγραμματιστικό, ηλεκτρονικό και κατασκευαστικό κομμάτι.

Η εργασία, την οποία εκτελεί το ρομπότ είναι απλή και δεν επιφέρει μεγάλη επιβάρυνση ούτε στους κινητήρες ή τους άξονες επειδή, πρώτον δεν απαιτείται απόλυτη ακρίβεια στις κινήσεις και δεύτερον η χρήση του θα είναι περιστασιακή και όχι καθημερινή. Για πιο απαιτητικές χρήσεις παρόμοιων εφαρμογών, θα χρησιμοποιηθούν ανάλογα, δυνατότερα τροφοδοτικά, και πιο αξιόπιστοι drivers για τους κινητήρες.

Με την προσθήκη των κατάλληλων εργαλείων στην κεφαλή, το ρομπότ μπορεί κάλλιστα να χαράξει, να κόψει, να γράψει αλλά και να ζωγραφίσει πάνω σε επιφάνειες, με τις κατάλληλες αλλαγές στον κώδικα.

Τέλος, με την προσθήκη ενός ακόμα άξονα Z στην κεφαλή του ρομπότ, μπορεί εύκολα να μετατραπεί σε ρομπότ τριών αξόνων για την πραγματοποίηση πιο σύνθετων λειτουργιών.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Εισαγωγή στη Ρομποτική - John J Craig
2. Ρομποτική - Σπύρος Γ. Τζαφέστας

ΑΝΑΦΟΡΕΣ ΣΤΟ ΔΙΑΔΙΚΤΥΟ

1. www.wikipedia.com (τελευταία πρόσβαση 5.11.2017)
2. <http://learn.mikroe.com/http://learn.mikroe.com/> (τελευταία πρόσβαση 28.10.2017)
3. http://www.robotlab.tuc.gr/ASSETS/PAPERS_PDF/ROBOTICS/LAB/4_STEPPER&SERVOS_LAB.pdf (τελευταία πρόσβαση 15.10.2017)
4. http://www.thomsonlinear.com/downloads/articles/Why_Lead_Screws_Best_Fit_Linear_Motion_Applications_tae.pdf (τελευταία πρόσβαση 5.11.2017)
5. <http://www.nmbtc.com/step-motors/engineering/basic-introduction-of-step-motors/> (τελευταία πρόσβαση 3.11.2017)
6. <http://homepage.cs.uiowa.edu/~jones/step/physics.html> (τελευταία πρόσβαση 5.11.2017)
7. <https://www.arduino.cc/> (τελευταία πρόσβαση 5.8.2017)
8. <http://www.instructables.com/> (τελευταία πρόσβαση 23.10.2017)