



**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΜΗΧΑΝΙΚΩΝ Τ.Ε**

CNC ΓΙΑ ΕΡΓΑΣΤΗΡΙΑΚΟΥΣ ΣΚΟΠΟΥΣ

ΕΠΙΒΛΕΠΩΝ:

Ε. Δοϊτσίδης

Επίκουρος Καθηγητής

Τουρκοκώστας Κυριάκος

Χανιά, 2015

ΕΥΧΑΡΙΣΤΙΕΣ

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω όλους όσους με βοήθησαν και με στήριξαν κατά τη διάρκεια των σπουδών μου στο ΤΕΙ ΚΡΗΤΗΣ και κυρίως τον επιβλέποντα καθηγητή μου Ε. Δοϊτσίδα, για την δυνατότητα που μου έδωσε να ασχοληθώ με αυτή τη διπλωματική εργασία, αλλά και την καθοδήγηση και τη βοήθεια που μου πρόσφερε καθ'όλη τη διάρκεια της. Επίσης ιδιαίτερες ευχαριστίες θα ήθελα να δώσω στον πατέρα μου Ηλία Τουρκοκώστα, στον Ιωάννη Γριζόπουλο και στον Ονούφριο Μαρκάκη για την συμπαράσταση και την πολύτιμη βοήθεια τους στην εκτέλεση της πτυχιακής αυτής εργασίας.

Τουρκοκώστας Κυριάκος

ΠΕΡΙΛΗΨΗ ΠΤΥΧΙΑΚΗΣ

Η παρούσα πτυχιακή εργασία πραγματεύεται την κατασκευή μιας μηχανής αριθμητικού ελέγχου μέσω H/Y, για το τρύπημα πλακετών για τις ανάγκες του εργαστηρίου. Η εργασία αυτή εκπονήθηκε στο εργαστήριο Ρομποτικής και Ευφυών Συστημάτων του Τμήματος Ηλεκτρονικών Μηχανικών του Τ.Ε.Ι Κρήτης.

Σκοπός της εργασίας είναι κατασκευή μιας μηχανής αριθμητικού ελέγχου με καθαρή επιφάνεια κατεργασίας 200X200 mm η οποία θα χρησιμοποιείται για το τρύπημα και τη χάραξη πλακετών μέσω H/Y.

Αρχικά αναλύθηκαν οι προδιαγραφές που πρέπει να έχει το μηχάνημα, όπως το υλικό κατασκευής, το μέγεθος, η ανάλυση ακρίβειας και το κόστος του. Στη συνέχεια σχεδιάστηκε σε πρόγραμμα τρισδιάστατου σχεδιασμού και κατασκευάστηκε το μηχανολογικό μέρος σε κατάλληλο εργαστήριο. Έπειτα σχεδιάστηκε το κύκλωμα οδήγησης του κινητήρα και η κεντρική πλακέτα που αναλαμβάνει την διασύνδεση με τον H/Y. Το τελευταίο στάδιο αφορούσε το προγραμματισμό του λογισμικού που χρησιμοποιήθηκε για την κατασκευή αυτή έτσι ώστε το μηχάνημα να εκτελεί τις εντολές του χρήστη.

Η εμπειρία που αποκτήθηκε κρίνεται ως σημαντική, καθώς οι μηχανές αριθμητικού ελέγχου καταλαμβάνουν μεγάλο μέρος της βιομηχανικής παραγωγής και σε συνδυασμό με τις γνώσεις που αποκτήθηκαν για την εργασία αυτή αποτελεί μία καλή βάση για το μέλλον.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1	1
ΕΙΣΑΓΩΓΗ	1
ΚΕΦΑΛΑΙΟ 2	2
ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ	2
2.1 ΓΕΝΙΚΑ ΓΙΑ ΤΑ ΡΟΜΠΟΤ	2
2.1.1 ΤΑΞΙΝΟΜΗΣΗ ΡΟΜΠΟΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ.....	2
2.2 ΡΟΜΠΟΤΙΚΑ ΣΥΣΤΗΜΑΤΑ	3
2.2.1 ΡΟΜΠΟΤΙΚΑ ΣΥΣΤΗΜΑΤΑ ΤΥΠΟΥ GANTRY	5
2.3 ΔΙΑΘΕΣΙΜΕΣ ΤΕΧΝΟΛΟΓΙΕΣ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	6
2.4 ΔΙΑΘΕΣΙΜΕΣ ΤΕΧΝΟΛΟΓΙΕΣ ΣΕ ΘΕΜΑΤΑ HARDWARE.....	8
ΚΕΦΑΛΑΙΟ 3	12
ΜΕΘΟΔΟΛΟΓΙΑ / ΣΧΕΔΙΑΣΗ	12
ΕΙΣΑΓΩΓΗ	12
3.1 ΣΧΕΔΙΑΣΜΟΣ	12
3.2 ΚΙΝΗΤΗΡΕΣ	12
3.3 ΣΤΑΔΙΑ ΥΛΟΠΟΙΗΣΗΣ ΜΗΧΑΝΟΛΟΓΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ	16
3.3.1 ΠΡΟΓΡΑΜΜΑ AUTOCAD	16
3.4 ΠΕΡΙΓΡΑΦΗ ΜΗΧΑΝΩΝ ΑΡΙΘΜΗΤΙΚΟΥ ΕΛΕΓΧΟΥ ΚΑΙ ΑΝΤΙΣΤΟΙΧΕΣ ΚΑΤΑΣΚΕΥΕΣ.....	16
3.5 ΥΛΙΚΑ ΚΑΙ ΔΟΜΙΚΑ ΜΕΡΗ ΚΑΤΑΣΚΕΥΗΣ.....	19
3.6 ΥΛΟΠΟΙΗΣΗ ΜΗΧΑΝΟΛΟΓΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	21
3.7 ΕΠΙΛΟΓΗ ΚΙΝΗΤΗΡΩΝ.....	25
ΚΕΦΑΛΑΙΟ 4	28
ΕΦΑΡΜΟΓΗ / ΥΛΟΠΟΙΗΣΗ	28
ΕΙΣΑΓΩΓΗ	28
4.1 ΟΔΗΓΟΣ ΚΙΝΗΣΗΣ ΚΙΝΗΤΗΡΑ A4988	28
4.2 ΗΛΕΚΤΡΟΝΙΚΟ ΚΥΚΛΩΜΑ ΚΙΝΗΣΗΣ ΒΗΜΑΤΙΚΟΥ ΚΙΝΗΤΗΡΑ.....	30
4.3 ARDUINO	31

4.4 ΣΥΝΔΕΣΜΟΛΟΓΙΑ ARDUINO	33
4.5 ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΔΙΕΠΑΦΗΣ	34
4.6 ΕΠΙΚΟΙΝΩΝΙΑ ΜΑΤLAB ΜΕ ARDUINO.....	36
4.7 ΑΡΧΙΚΑ ΠΕΙΡΑΜΑΤΑ	36
ΚΕΦΑΛΑΙΟ 5	37
5.1 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	37
ΒΙΒΛΙΟΓΡΑΦΙΑ	38
ΠΑΡΑΡΤΗΜΑ.....	40

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Στην παρούσα πτυχιακή εργασία περιγράφονται με λεπτομέρεια όλα τα στάδια ανάπτυξης μιας πρωτότυπης εργαλειομηχανής αριθμητικού ελέγχου(CNC) που κατασκευάστηκε για εκπαιδευτικούς σκοπούς.

Αναλύεται με λεπτομέρεια ο σχεδιασμός, η ανάπτυξη και η υλοποίηση ενός πλήρους λειτουργικού πρωτοτύπου με επιφάνεια κατεργασίας 200x200 πλήρως ελεγχόμενο από Η/Υ.

Συγκεκριμένα περιγράφεται ο σχεδιασμός του πρωτοτύπου με χρήση προγράμματος τρισδιάστατου σχεδιασμού, τα στάδια κατασκευής και τα πρώτα λειτουργικά πρωτότυπα. Στη συνέχεια, περιγράφεται η ανάπτυξη των ηλεκτρονικών μερών και τέλος του λογισμικού που αναπτύχθηκε και επιτρέπει τον πλήρη έλεγχο της συσκευής.

Το τελικό αποτέλεσμα ήταν πλήρως λειτουργικό και αποτέλεσαι ένα πρωτότυπου χαμηλού κόστους και υψηλών δυνατοτήτων.

ΚΕΦΑΛΑΙΟ 2

ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ

ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό αναφέρονται γενικές πληροφορίες σχετικά με τα ρομπότ καθώς και τις υπάρχουσες τεχνολογίες σχετικά με το ρομποτικό σύστημα της παρούσας πτυχιακής εργασίας.

2.1 ΓΕΝΙΚΑ ΓΙΑ ΤΑ ΡΟΜΠΟΤ

Η λέξη ρομπότ εμφανίστηκε για πρώτη φορά από τον Τσέχο σκηνοθέτη KarelCapekστο έργο του RUR που παίχθηκε για πρώτη φορά στο Παρίσι το 1922. Η εξέλιξη στην αυτοματοποίηση της παραγωγικής διαδικασίας στις βιομηχανίες αποτέλεσε την αιτία της μεγάλης αυτής στροφής του τεχνολογικού τομέα στη ρομποτική. Ανεξάρτητα όμως από τη βελτίωση της παραγωγικής διαδικασίας και τα οικονομικά οφέλη η ρομποτική τεχνολογία εφαρμόζεται σε όλους τους τομείς που συνθέτουν όχι μόνο την καθημερινότητα μας αλλά και την ύπαρξη μας όπως η εξερεύνηση του διαστήματος, των βυθών, της ιατρικής, τις αγροτικές εφαρμογές καθώς και την έρευνα και διάσωση . Αξίζει να σημειωθεί ότι η ρομποτική τεχνολογία στους τομείς αυτούς βρίσκεται ακόμα σε πρωταρχικό στάδιο, με το μέλλον φυσικά να είναι ιδιαίτερα αισιόδοξο και ελπιδοφόρο.[1]

Οι κύριες συνιστώσες επιστήμες που συνδράμουν και συνθέτουν τη ρομποτική έρευνα και τεχνολογία είναι οι ακόλουθες:

1. Μηχανική-Μηχανολογία
2. Μαθηματικά
3. Συστήματα αυτόματου ελέγχου
4. Τεχνητή νοημοσύνη
5. Τεχνολογία λογισμικού και προγραμματισμός
6. Ηλεκτρονική
7. Συστήματα και διοίκηση παραγωγής

2.1.1 ΤΑΞΙΝΟΜΗΣΗ ΡΟΜΠΟΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Η ταξινόμηση των βιομηχανικών ρομπότ αποτελεί τον οδηγό για τη σύγκριση ρομπότ παρόμοιων ιδιοτήτων και είναι απαραίτητη για την επιλογή του κατάλληλου ρομπότ για

μια συγκεκριμένη εφαρμογή. Τα κυριότερα κριτήρια ταξινόμησης των ρομποτικών συστημάτων είναι:

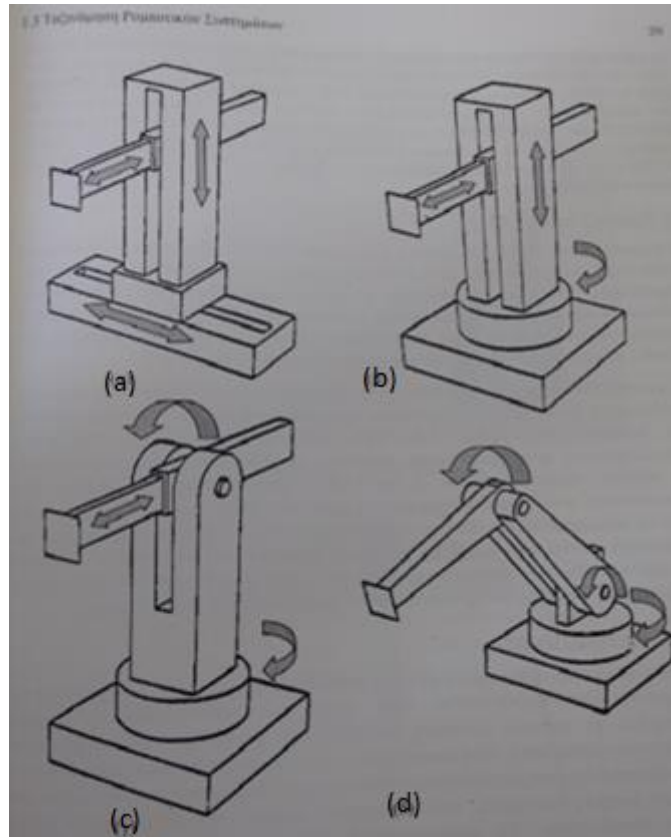
1. Η αρχή λειτουργίας (ρομπότ σταθερής στάσης, σερβοελεγχόμενα ρομπότ)
2. Μέθοδος ελέγχου κίνησης (ρομπότ σημείου-προς-σημείου, ρομπότ συνεχούς τροχιάς)
3. Γεωμετρικός σχηματισμός (καρτεσιανάρομπότ, κυλινδρικάρομπότ, σφαιρικά ρομπότ, αρθρωτά ρομπότ, ρομπότ τύπου SCARA,ρομπότ τύπου GANTRY)[1]

2.2 ΡΟΜΠΟΤΙΚΑ ΣΥΣΤΗΜΑΤΑ

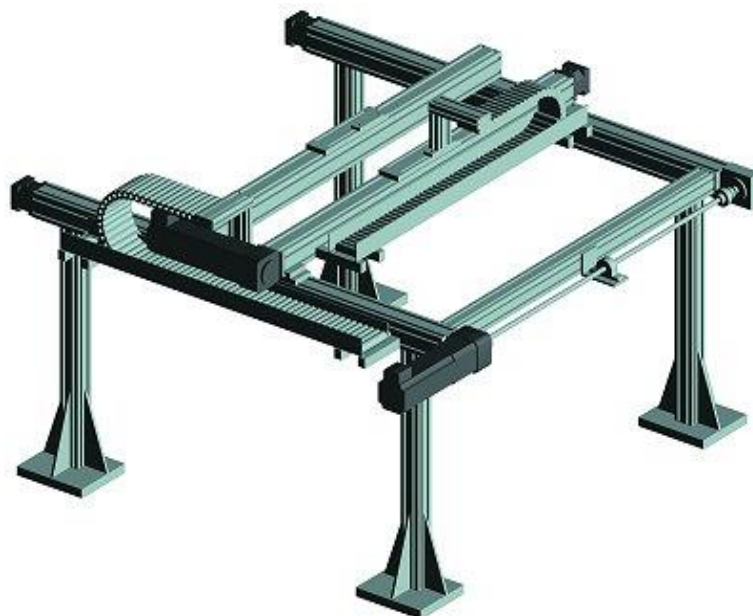
Στην προηγούμενη παράγραφο αναφέρθηκαν τα κριτήρια ταξινόμησης των ρομποτικών συστημάτων. Στο πρώτο κριτήριο εμφανίζονται τα ρομπότ σταθερής στάσης και προελεγχόμενα ρομπότ. Κύριο χαρακτηριστικό των ρομπότ σταθερής στάσης είναι ότι κάθε άξονας έχει συγκεκριμένο σταθερό μηχανικό όριο σε κάθε άκρο του και μπορεί να σταματάει μόνο σε αυτό. Αντίθετα τα προελεγχόμενα ρομπότ έχουν τη δυνατότητα να κινούνται μέσα από έναν άπειρο αριθμό σημείων κατά την εκτέλεση της λειτουργίας τους.

Το επόμενο κριτήριο ταξινόμησης αφορά τη μέθοδο ελέγχου κίνησης ρομποτικών συστημάτων. Στο κριτήριο αυτό συναντάμε δύο τύπους ρομπότ, τα ρομπότ σημείου-προς-σημείου και τα ρομπότ συνεχούς τροχιάς. Στον πρώτο τύπο κύριο χαρακτηριστικό είναι ότι το ρομπότ μπορεί να κινηθεί από μία θέση στην επόμενη χωρίς να προσδιοριστούν οι ενδιάμεσες τροχιές μεταξύ των σημείων ενώ αντίθετα στα ρομπότ συνεχούς τροχιάς το ρομπότ επαναλαμβάνει την κίνηση μέσα από σημεία σε μικρή απόσταση μεταξύ τους, τα οποία έχουν προγραμματισθεί σε μία σταθερή χρονική βάση κατά τη διάρκεια της λειτουργίας τους.

Ένα τρίτο σχήμα ταξινόμησης βασίζεται στο γεωμετρικό σχηματισμό του ρομποτικού συστήματος. Με τη χρήση της παρακάτω εικόνας αναλύεται το τρίτο κριτήριο ταξινόμησης.[1]



Εικόνα 2.1 Τύποι γεωμετρικών σχηματισμών ρομπότ:(α) Καρτεσιανό Ρομπότ (β) Κυλινδρικό Ρομπότ (γ) Σφαιρικό Ρομπότ (δ) Αρθρωτό ρομπότ



Εικόνα 2.2 Ρομπότ τύπουGantry

2.2.1 ΡΟΜΠΟΤΙΚΑ ΣΥΣΤΗΜΑΤΑ ΤΥΠΟΥ GANTRY

Στη παρούσα πτυχιακή εργασία θα μελετηθεί η λειτουργία ενός καρτεσιανού ρομπότ του οποίου οι τρεις άξονες έλεγχου είναι γραμμικοί (δηλαδή μετακινούνται σε ευθεία γραμμή και δεν περιστρέφονται) και είναι σε γωνία 90 μοιρών ο ένας με τον άλλο. Τα ρομπότ καρτεσιανών συντεταγμένων με τον οριζόντιο άξονα στηριγμένο και στα δύο άκρα του μερικές φορές ονομάζονται και **ρομπότ γκάντρι** (gantry robots). Η συγκεκριμένη αριθμητική διάταξη χρησιμοποιείται στις μηχανές αριθμητικού ελέγχου (computer numerical control machine ή CNC machine).

Η δομή μπορεί να είναι όμοια με τις εργαλειομηχανές (βάση, τραπέζι εργασίας, κ.ά.) αλλά τότε ο λόγος μεταξύ του χώρου εργασίας του ρομπότ και του χώρου που καταλαμβάνει είναι μικρότερος. Γενικά, τα χαρακτηριστικά (ηλεκτρονικό υλικό, πρόγραμμα ελέγχου, κλπ.) ενός καρτεσιανού ρομπότ είναι όμοια με εκείνα των εργαλειομηχανών υπολογιστικού αριθμητικού ελέγχου (CNC).

Έτσι η διακριτική ικανότητα και η επαναληψιμότητα ενός καρτεσιανού ρομπότ μπορεί να είναι πολύ καλή όπως και στις εργαλειομηχανές. Σε πολλά καρτεσιανά ρομπότ η βάση δεν είναι σταθερή αλλά μπορεί να κινείται μέσα σε ορισμένα όρια. Ο καρπός ενός καρτεσιανού ρομπότ μπορεί να ακολουθήσει μια ευθύγραμμη τροχιά, αν κάθε άξονας κινηθεί με σταθερή ταχύτητα. Στα άλλα είδη ρομπότ οι σχέσεις που δίνουν τις ταχύτητες των αξόνων για τη λήψη ευθύγραμμων τροχιών δεν είναι τόσο απλές. Στα ρομπότ αυτά πρέπει να γίνει μετασχηματισμούς των καρτεσιανών συντεταγμένων των αρθρώσεων του ρομπότ. Άλλο πλεονέκτημα των καρτεσιανών ρομπότ είναι η σταθερότητα της διακριτικής ικανότητας θέσης. Δηλαδή η ΒΜΔΙ (η διακριτική ικανότητα προγραμματισμού είναι η μικρότερη επιτρεπτή μεταβολή θέσης στα ρομποτικά προγράμματα, και είναι γνωστή σαν βασική μονάδα διακριτικής ικανότητας) είναι ορισμένη για κάθε άξονα και παραμένει σταθερή σε όλα τα σημεία του χώρου εργασίας του ρομπότ.

Αυτό δεν συμβαίνει στα μη καρτεσιανά ρομπότ. Παρά τα πλεονεκτήματα αυτά, τα καρτεσιανά ρομπότ δεν είναι προτιμητέα στη βιομηχανία. Τούτο συμβαίνει γιατί δεν έχουν μηχανική ευελιξία (δεν μπορούν λ.χ. να φθάσουν αντικείμενα που βρίσκονται στο πάτωμα ή δεν είναι ορατά από τη βάση τους). Επίσης η ταχύτητα λειτουργίας στο οριζόντιο επίπεδο είναι συνήθως μικρότερη από την αντίστοιχη ταχύτητα των ρομπότ που έχουν περιστρεφόμενη βάση. Τα Καρτεσιανά συστήματα σχεδιάζονται για να παρέχουν αξιόπιστη, λειτουργική, μεγάλης ακρίβειας και οικονομική λύση σε συγκολλήσεις πολύ μεγάλων κομματιών όπως σε συγκολλήσεις, containers, κάδων απορριμμάτων και μεταλλικών κτιρίων.[2]

2.3 ΔΙΑΘΕΣΙΜΕΣ ΤΕΧΝΟΛΟΓΙΕΣ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

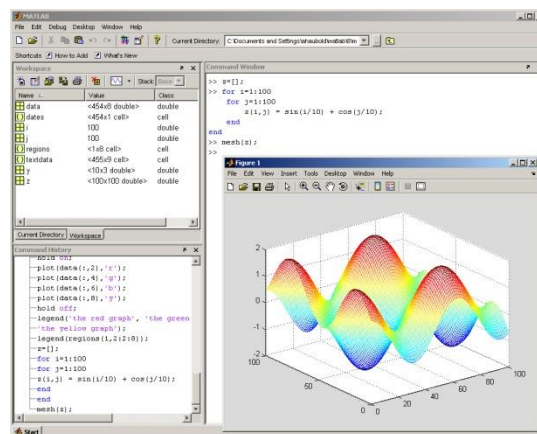
Η ραγδαία εξέλιξη της τεχνολογίας στο τομέα της πληροφορικής έχει επηρεάσει σε πολύ μεγάλο βαθμό τον τομέα της ρομποτικής. Λόγω του ισχυρότατου ανταγωνισμού αλλά και της τάσης ενσωμάτωσης των μικροελεγκτών σε κάθε ηλεκτρική και ηλεκτρονική συσκευή, η βιομηχανία μικροελεγκτών έχει καταλήξει στην παραγωγή ανταγωνιστικών μοντέλων μαζικής παραγωγής καθώς και μικροελεγκτών για πιο εξειδικευμένες εφαρμογές όπως η ρομποτική.

Η πιο διαδεδομένη γλώσσα προγραμματισμού των μικροελεγκτών είναι η C, η C++ και οι παραλλαγές τους. Σε τμήματα του λογισμικού όπου απαιτείται ταχύτητα η μικρό μέγεθος χρησιμοποιούμενης μνήμης, μπορεί να χρησιμοποιείται η Assembly. Όμως οι μεγαλύτερες απαιτήσεις σε λειτουργικότητα και η ευκολία προγραμματισμού της C έναντι της Assembly, σε συνδυασμό με την επάρκεια μνήμης των σύγχρονων μικροελεγκτών, έχουν γενικά εκτοπίσει την Assembly από τις περισσότερες εφαρμογές.

Σύμφωνα με τα παραπάνω υπάρχει μεγάλη ποικιλία στην επιλογή του λογισμικού για μια ρομποτική εφαρμογή όπως στην παρούσα πτυχιακή εργασία. Παρακάτω εμφανίζονται μερικές από τις επιλογές αυτές:

1. MATLAB

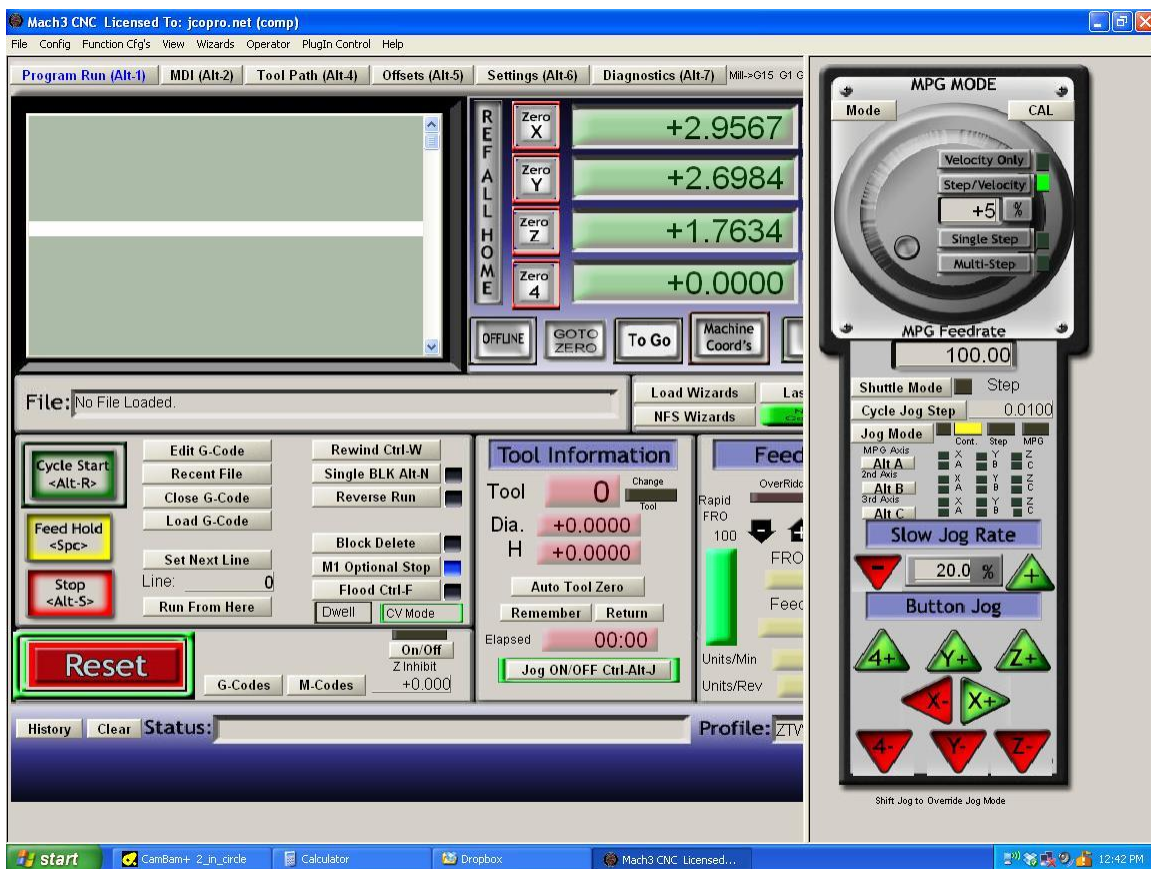
Χρησιμοποιείται κατά κύριο λόγο για την επίλυση μαθηματικών προβλημάτων, ωστόσο είναι πολύ "ισχυρό" και μπορεί να χρησιμοποιηθεί και για προγραμματισμό καθώς περιέχει εντολές από την C++ όπως την while, την switch και την if. Στον τομέα των γραφικών όσον αφορά τον μαθηματικό κλάδο μπορεί να υλοποιήσει συναρτήσεις πραγματικές, μιγαδικές, πεπλεγμένες συναρτήσεις δύο μεταβλητών και άλλες. Όσον αφορά τον στατιστικό κλάδο μπορεί να υλοποιήσει ιστογράμματα, τομεογράμματα, ραβδοδιαγράμματα, εμβαδογράμματα και άλλα.



Εικόνα 2.3 Γραφική απεικόνιση στο MATLAB

2. MACH3

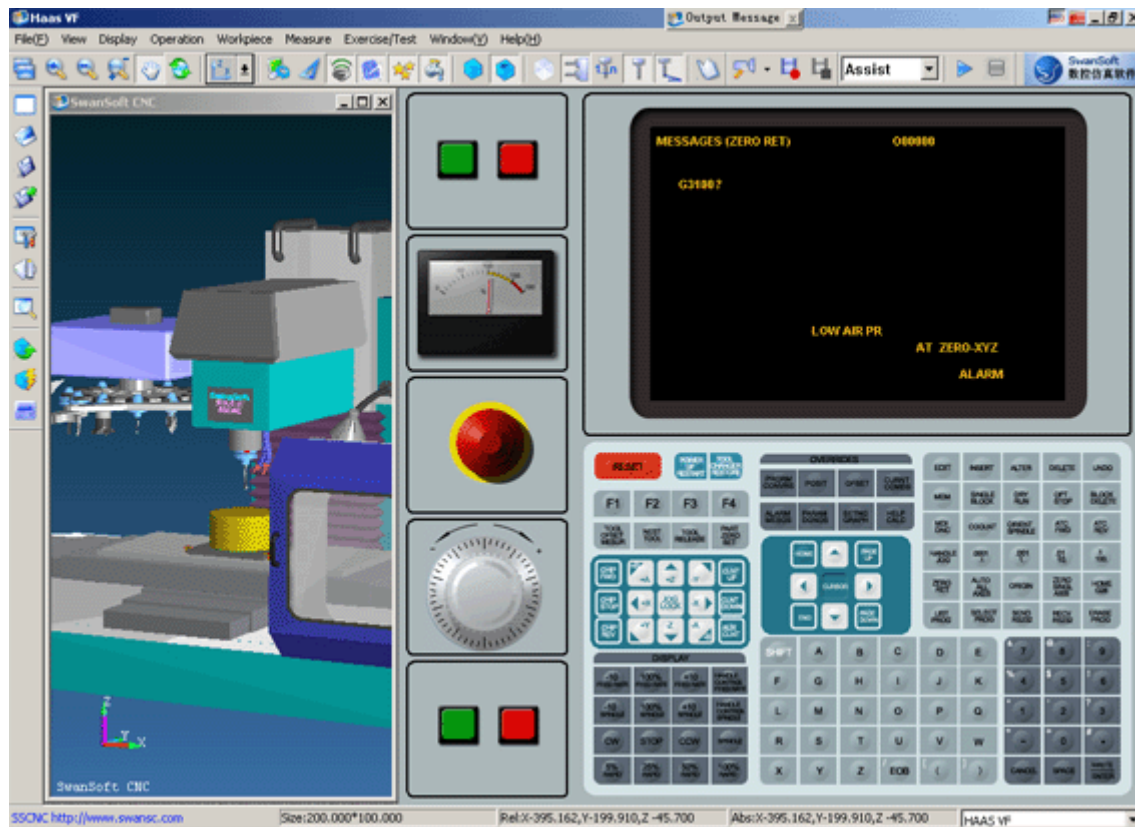
Mach3 μετατρέπει έναν τυπικό υπολογιστή σε έναν ελεγκτή CNC. Είναι πολύ πλούσια σε χαρακτηριστικά και προσφέρει μια μεγάλη αξία για εκείνους που χρειάζονται ένα πακέτο ελέγχου CNC. Mach3 λειτουργεί στα περισσότερα Windows PC να ελέγξει την κίνηση των κινητήρων (stepper & σέρβο) με την επεξεργασία G-κώδικα. Ενώ η οποία περιλαμβάνει πολλά προηγμένα χαρακτηριστικά, είναι διαθέσιμο το πιο έξυπνο λογισμικό ελέγχου CNC. Η Mach3 είναι προσαρμόσιμη και έχει χρησιμοποιηθεί για πολλές εφαρμογές με πολλούς τύπους υλικού.



Εικόνα 2.4 Κεντρικό μενού προγράμματος MACH3

3. CNCSIMULATOR

CNC Simulator είναι ένα Computer Numerical Control (CNC) προσομοιωτής. Μιμείται το μονοπάτι που το εργαλείο ενός CNC θα λάβει για ένα συγκεκριμένο πρόγραμμα. Η προσομοίωση είναι δύο διαστάσεων, με πρόβλεψη για την τελική προβολή σε 3D, χρησιμοποιώντας OpenGL.



Εικόνα 2.5 Κεντρικό μενού του προγράμματος CNC SIMULATOR

2.4 ΔΙΑΘΕΣΙΜΕΣ ΤΕΧΝΟΛΟΓΙΕΣ ΣΕ ΘΕΜΑΤΑ HARDWARE

Μία από τις κύριες συνιστώσες επιστήμες που συνδράμουν και συνθέτουν τη ρομποτική έρευνα και τεχνολογία είναι η ηλεκτρονική. Η ανάγκη για συνεχή εξέλιξη της τεχνολογίας έχει ως συνέπεια την ταχύτατη ανάπτυξη των ενσωματωμένων ηλεκτρονικών συστημάτων. Χάρη στην ταχύτατη αυτή εξέλιξη πολλές εταιρίες επέλεξαν να επενδύσουν σε αυτό το τομέα με αποτέλεσμα να εμφανίζονται συνεχώς καινοτόμες ιδέες που εξυπηρετούν στην ύπαρξη αποδοτικών ενσωματωμένων συστημάτων.

Σε αυτή τη πτυχιακή εργασία το ηλεκτρονικό κομμάτι αποτελείται από τον επεξεργαστή, τους κινητήρες και τους drivers των κινητήρων(A4988). Παρακάτω παρουσιάζονται λύσεις για το ηλεκτρονικό κομμάτι της εργασίας.

1. ARDUINO

Το **Arduino** είναι ένας single-board μικροελεκτήρας, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, SuperCollider. Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προ-συναρμολογημένες· το διάγραμμα και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους.[3]



Εικόνα 2.6 Πλακέτα ARDUINO

2. RASPBERRY PI

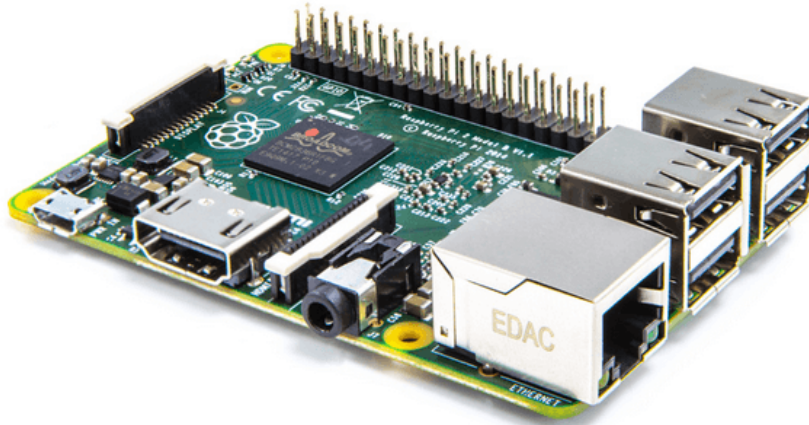
Το Raspberry Pi είναι ένας πλήρης υπολογιστής με μέγεθος πιστωτικής κάρτας.

Η τελευταία του έκδοση, Raspberry Pi 2, με τετραπύρρηνο επεξεργαστή τύπου ARM και 1GB RAM κοστίζει γύρω στα 45 ευρώ στην Ελλάδα, ενώ ξεχωριστά θα βρούμε το πλήρες kit με όσα χρειάζονται για να το αξιοποιήσουμε.

Παρά τον ελάχιστον όγκο του, το Raspberry Pi διαθέτει τετραπύρρηνο επεξεργαστή 900MHz, διπύρρηνη κάρτα γραφικών, 1GB RAM, τέσσερις θύρες

USB, έξοδο HDMI, τροφοδοτείται μέσω Micro USB, και 40 pins γενικής χρήσης για σύνδεση με άλλα ηλεκτρονικά και περιφερειακά.

Η σημαντικότερη καινοτομία της νέας έκδοσης του Raspberry Pi είναι πως θα μπορεί να τρέξει τα επερχόμενα Windows 10.



Εικόνα 2.7 Πλακέτα του RaspberryPi

3. A4988 Stepper Motor Driver Carrier

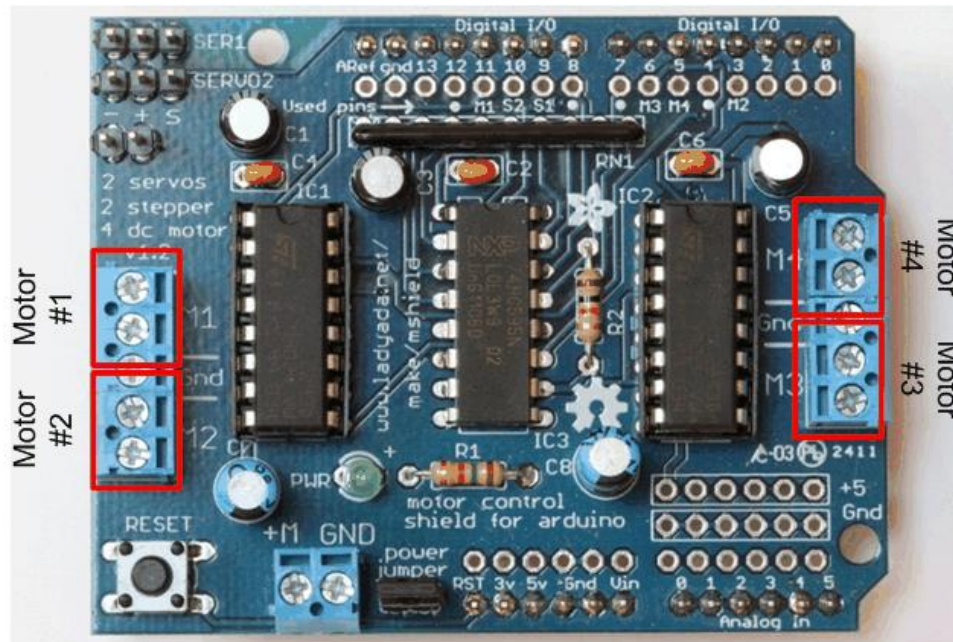
Ο A4988 είναι ένας οδηγός διπολικού βηματικού κινητήρα που διαθέτει ρυθμιζόμενο περιορισμό ρεύματος, προστασία υπερθέρμανσης και μέχρι 5 διαφορετικές αναλύσεις βήματος(1/16-βήμα).Λειτουργεί από 8-35Vκαι μπορεί να προσφέρει μέχρι και 1Aανά φάση χωρίς ψύκτρα.



Εικόνα2.8 A4988 Stepper Motor Driver

4. ADAFRUIT

Το Adafruit είναι πλακέτα που σχεδιάστηκε για να οδηγεί βηματικούς και DC, κινητήρες. Είναι κατάλληλο για ρομποτικές εφαρμογές καθώς έχει τη δυνατότητα να οδηγεί 4 DC κινητήρες και 2 βηματικούς. Αντί ενός οδηγού L293D τάρλινγκτον, έχουμε τώρα τον οδηγό MOSFET TB6612 με 1.2A ανά κανάλι και δυνατότητα μέχρι 3A. Επίσης, είναι πολύ χαμηλή πτώση τάσης του κινητήρα, ώστε να πάρει περισσότερη ροπή από τις μπαταρίες.



Εικόνα 2.9 Πλακέτα ADAFRUIT

ΚΕΦΑΛΑΙΟ 3

ΜΕΘΟΔΟΛΟΓΙΑ / ΣΧΕΔΙΑΣΗ

ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό θα αναπτυχθεί το θεωρητικό κομμάτι της σχεδίασης του CNCμηχανήματος. Θα αναλυθεί το μηχανολογικό κομμάτι της κατασκευής, δηλαδή τα υλικά που χρησιμοποιήθηκαν, τα προβλήματα που παρουσιάστηκαν και διάφορες εκτιμήσεις και προτάσεις πάνω στη μηχανολογική σχεδίαση. Επιπρόσθετα θα αναλυθεί το κομμάτι που αφορά τους κινητήρες που χρησιμοποιήθηκαν και εναλλακτικές προτάσεις πάνω σε αυτό.

3.1 ΣΧΕΔΙΑΣΜΟΣ

Προκειμένου να υλοποιηθεί ένα CNC μηχάνημα διερευνήθηκαν οι πιθανές λύσεις και ο τρόπος υλοποίησης. Στη συνέχεια θα περιγραφεί αναλυτικά η προτεινόμενη κατασκευή και τα δομικά της μέρη.

3.2 ΚΙΝΗΤΗΡΕΣ

Αρχικά έγινε διερεύνηση των πιθανών επιλογών όσον αφορά τους κινητήρες που θα χρησιμοποιούνταν στην κατασκευή. Οι εναλλακτικές λύσεις ήταν οι κινητήρες συνεχούς ρεύματος (dc motors), οι βηματικοί κινητήρες (stepper motors) και οι σερβοκινητήρες (servo motors).

1. Κινητήρες συνεχούς ρεύματος(dc motors)

Οι κινητήρες συνεχούς ρεύματος είναι αρκετά ελκυστική επιλογή, λόγω της δυνατότητας που παρέχουν για εύκολο έλεγχο ταχύτητας και ροπής. Έτσι, υπάρχουν εφαρμογές πολύ υψηλών απαιτήσεων στις οποίες οι κινητήρες συνεχούς ρεύματος είναι αναντικατάστατοι. Η λειτουργία του κλασσικού ηλεκτρικού κινητήρα συνεχούς ρεύματος βασίζεται στην αλληλεπίδραση δύο μαγνητικών πεδίων. Το πρώτο πεδίο δημιουργείται από μόνιμους μαγνήτες (διέγερση) που βρίσκονται συνήθως σταθερά προσαρμοσμένοι στον στάτη της μηχανής. Το δεύτερο πεδίο δημιουργείται στον δρομέα της μηχανής από το περιστρεφόμενο τύλιγμα τυμπάνου, που συνιστά έναν ηλεκτρομαγνήτη. Βασικό ρόλο στη λειτουργία της μηχανής συνεχούς ρεύματος παίζει ο συλλέκτης. Ο ρόλος του συλλέκτη είναι να αντιστρέφει τη φορά του ηλεκτρικού ρεύματος

τυμπάνου δύο φορές σε κάθε κύκλο, έτσι ώστε να αντιστρέφεται η πολικότητα του ηλεκτρομαγνήτη τυμπάνου την κατάλληλη χρονική στιγμή (όταν οι δύο αντίθετοι μαγνητικοί πόλοι βρίσκονται απέναντι) και τελικά να αλληλεπιδρούν εις το διηνεκές μαγνητικά τα δύο πεδία. Αυτό είναι αναγκαίο αφού το τύλιγμα τυμπάνου είναι περιστρεφόμενο και χωρίς το συλλέκτη η μηχανή θα σταματούσε άμεσα την πρώτη φορά που δύο αντίθετοι πόλοι θα βρίσκονταν απέναντι. [4]

Τα βασικά χαρακτηριστικά τους είναι:

1. Έχουν γωνία περιστροφής μεγαλύτερη των 360°
2. Μεγάλη ροπή
3. Δεν διαθέτουν ενσωματωμένα γρανάζια μείωσης στροφών
4. Δεν διαθέτουν έλεγχο θέσης του άξονα
5. Χρειάζονται εξωτερικό κυκλώματος οδήγησης
6. Μικρό κόστος[6]



Εικόνα 3.1 Κινητήρας συνεχούς ρεύματος

2. Βηματικοί κινητήρες (stepper motors)

Οι βηματικοί κινητήρες, σε γενικό πλαίσιο, σύμφωνα με τον Bishop (2002) είναι σύγχρονοι ηλεκτρικοί κινητήρες που μετατρέπουν την ηλεκτρική ενέργεια σε μηχανική. Ειδικότερα ως βηματικός κινητήρας ορίζεται μια ηλεκτρομηχανική

συσκευή η οποία μετατρέπει τους ηλεκτρικούς παλμούς σε ιδιαίτερες μηχανικές κινήσεις, δηλαδή σε πολλαπλά μηχανικά βήματα. Οι κινήσεις αυτές μπορεί να είναι είτε κυκλικές (περιστροφικός κινητήρας) είτε γραμμικές (γραμμικός κινητήρας)[5]

Τα βασικά χαρακτηριστικά τους είναι:

1. Έχουν γωνία περιστροφής μεγαλύτερη των 360°
2. Μικρή ροπή
3. Δεν διαθέτουν ενσωματωμένα γρανάζια μείωσης στροφών
4. Απαιτείται αρχικοποίηση για τον έλεγχο θέσης του άξονα
5. Χρειάζονται εξωτερικό κυκλώματος οδήγησης
6. Μικρό κόστος[6]



Εικόνα 3.2 Νηματικός κινητήρας

3. Σερβοκινητήρες

Οι σερβοκινητήρες χρησιμοποιούνται εδώ και πολλά χρόνια στα συστήματα αυτόματου ελέγχου κλειστού βρόγχου και κυρίως σε εφαρμογές που απαιτείται έλεγχος ταχύτητας, θέσης και ροπής του άξονα του κινητήρα. Χρησιμοποιούνται σε ρομποτικούς βραχίονες, αυτόματες εργαλειομηχανές, τηλεκατευθυνόμενα μοντέλα και σε αυτόματα συστήματα πλοήγησης πλοίων και αεροπλάνων. Με την εξέλιξη των σερβοκινητήρων η τεχνολογία πέρασε από τον κλασσικό έλεγχο ταχύτητας (speed control) στο συνολικό έλεγχο κίνησης (motion control). Αυτή η εξέλιξη έπαιξε σημαντικό ρόλο στον τομέα της ρομποτικής αλλά και σε άλλες τεχνολογίες.

Ένας σερβοκινητήρας πρέπει να αναπτύσσει υψηλή ροπή ώστε να αλλάζει γρήγορα η ταχύτητα του, να περιστρέφεται σταθερά σε μεγάλη περιοχή ταχυτήτων και να έχει υψηλή ταχύτητα απόκρισης στις εντολές του συστήματος

ελέγχου. Οι σερβοκινητήρες δεν μπορούν να επιτελέσουν την τόσο σημαντική λειτουργία τους λειτουργώντας μεμονωμένα, αλλά αποτελούν το βασικό στοιχείο ενός συστήματος αυτοματισμού που επιτελεί έλεγχο κίνησης. Στο συνολικό σερβόσύστημα εντάσσονται εκτός από τον σερβοκινητήρα, η μονάδα ελέγχου που στην πλειοψηφία των περιπτώσεων είναι ένα PLC και ο σερβοενισχυτής (servodriver) που αποτελεί το συνδεδετικό κρίκο του συνολικού συστήματος. Οι σερβοκινητήρες κατασκευαστικά μοιάζουν πολύ με τους κοινούς κινητήρες αλλά δεν είναι ίδιοι. Διαφέρουν στο ότι ενσωματώνουν διατάξεις μέτρησης και σύστημα ανάδρασης το οποίο χρησιμοποιείται σε συνδυασμό με έναν σερβομηχανισμό οδήγησης με σκοπό να ελεγχθεί είτε η ροπή, είτε η ταχύτητα του, είτε η θέση του.[10]

Βασικά χαρακτηριστικά σερβοκινητήρα :

1. Έχουν γωνία περιστροφής περίπου 180°
2. Μεγάλη ροπή
3. Διαθέτουν ενσωματωμένα γρανάζια μείωσης στροφών
4. Διαθέτουν έλεγχο θέσης του άξονα
5. Δεν χρειάζονται εξωτερικό κυκλώματος οδήγησης
6. Μεγάλο κόστος[6]



Εικόνα 3.3 Σερβοκινητήρας

Οι κινητήρες που επιλέχθηκαν να τοποθετηθούν στο CNC είναι τρεις βηματικοί και ένας κινητήρας συνεχούς ρεύματος. Σημαντικό ρόλο στην επιλογή των βηματικών κινητήρων αντί των σερβοκινητήρων έπαιξε το χαμηλό κόστος τους. Επίσης η

περιστροφή του άξονα των βηματικών κινητήρων σε διακριτά σταθερά βήματα με αποτέλεσμα τον καλύτερο έλεγχο θέσης του άξονα αποτελεί βασικός λόγος της επιλογής τους.

3.3 ΣΤΑΔΙΑ ΥΛΟΠΟΙΗΣΗΣ ΜΗΧΑΝΟΛΟΓΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

Η μελέτη και ο σχεδιασμός του μηχανολογικού συστήματος αποτελούν ένα από τα πιο βασικά σημεία αυτής της κατασκευής.

Τα στάδια υλοποίησης του μηχανολογικού μέρους της πτυχιακής αυτής εργασίας ήταν: α) ο σχεδιασμός της εργασίας στο σχεδιαστικό πρόγραμμα AutoCAD, β) η αγορά των υλικών, γ) η επεξεργασία των υλικών σε μηχανουργείο και δ) συναρμολόγηση του συστήματος.

3.3.1 ΠΡΟΓΡΑΜΜΑ AUTOCAD

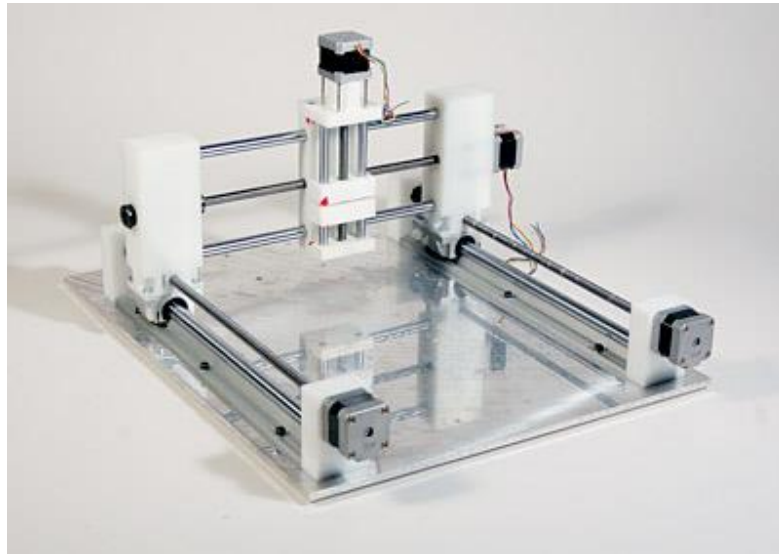
Το AutoCAD είναι το πιο διαδεδομένο πρόγραμμα CAD σε παγκόσμιο επίπεδο που απευθύνεται σε αρχιτέκτονες, πολιτικούς μηχανικούς, διακοσμητές, σχεδιαστές, κ.λπ. και γενικότερα για όσους σχεδιάζουν τεχνικό σχέδιο.

Διαθέτει εργαλεία και δυνατότητες για σχεδιασμό σε δύο ή τρεις διαστάσεις και ακόμα πιο εύκολο νεορεαλισμό αλλά πάντα συμβατό με παλαιότερες εκδόσεις βοηθάει ακόμη περισσότερο τον χρήστη στην διεκπεραίωση της δουλειάς του. Με το AutoCAD η αξία των παραγόμενων σχεδίων μεγιστοποιείται. Η επεξεργασία, η διαμόρφωση, η κοινοποίηση των σχεδίων γίνεται πλέον σε πραγματικό χρόνο εύκολα, γρήγορα και πάντα με τη γνωστή ακρίβεια του AutoCAD. Μέσα από το Internet ή κάποιο τοπικό δίκτυο ολόκληρη η ομάδα μελέτης μοιράζεται τις εργασίες, επιταχύνοντας έτσι την ολοκλήρωση του έργου.[7]

3.4 ΠΕΡΙΓΡΑΦΗ ΑΡΙΘΜΗΤΙΚΟΥ ΕΛΕΓΧΟΥ ΚΑΙ ΑΝΤΙΣΤΟΙΧΕΣ ΚΑΤΑΣΚΕΥΕΣ

Οι εργαλειομηχανές είναι μηχανήματα που χρησιμοποιούνται για την κατεργασία υλικών. Η διαδικασία της κατεργασίας περιλαμβάνει πολλές λειτουργίες όπως το κόψιμο, το τρύπημα, την αφαίρεση υλικού κ.α. Η διαφορά μιας εργαλειομηχανής και μιας μηχανής αριθμητικού ελέγχου είναι ότι οι δεύτερες είναι δυνατόν να εκτελούν περισσότερες από μία λειτουργίες μέσω ηλεκτρονικού υπολογιστή. Επιπρόσθετα οι μηχανές αριθμητικού ελέγχου προσφέρουν πιο παραγωγική διαδικασία καθώς διαθέτουν μονάδες ελέγχου έτσι ώστε να ελαττώσουν όσο πιο πολύ γίνεται την παρέμβαση κάποιου χειριστή. Έτσι λοιπόν ο έλεγχος επιτρέπει περίπλοκες και ακριβείς κινήσεις με αποτέλεσμα οι μηχανές αριθμητικού ελέγχου σε συνδυασμό με προηγμένα προγράμματα να μπορούν να κάνουν κατεργασίες που παλιότερα θα ήταν ασύμφορες ή αδύνατες με χειροκίνητες εργαλειομηχανές.

Η αρχική επιλογή του σχεδιασμού του CNC επιλέχθηκε βάση παρόμοιων κατασκευών στον βιομηχανικό τομέα. Στη συνέχεια χρειάστηκε να διαφοροποιηθεί καθώς σημαντικό ρόλο έπαιξαν τα υλικά που επιλέχθηκαν για τη δημιουργία της κατασκευής. Το πρόγραμμα AutoCAD αποδείχθηκε πολύ εύχρηστο και λειτουργικό με αποτέλεσμα το κομμάτι της σχεδίασης να αποτελέσει ένα πολύ χρήσιμο οδηγό στη διαδικασία της συναρμολόγησης. Στα σχήματα 3.4, 3.5 και 3.6 παρουσιάζονται ενδεικτικά κάποιες μηχανές CNC.



Εικόνα 3.4 Τυπικό μηχάνημα τριών αξόνων για την κατασκευή πλακετών



Εικόνα 3.5 Τυπικό CNCRouter



Εικόνα 3.6 Τυπικό CNCRouter μεγάλου μεγέθους

3.5 ΥΛΙΚΑ ΚΑΙ ΔΟΜΙΚΑ ΜΕΡΗ ΚΑΤΑΣΚΕΥΗΣ

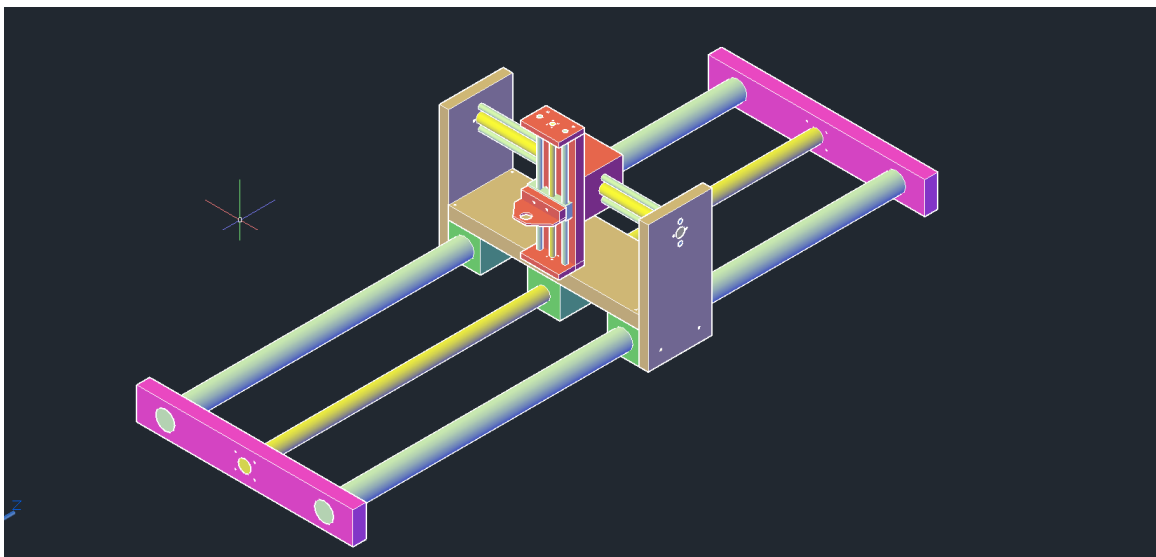
Οι κύριοι παράγοντες που έπαιξαν σημαντικότερο ρόλο στην επιλογή των υλικών της κατασκευής είναι το βάρος του μηχανήματος καθώς πρόκειται για ένα μηχάνημα για εργαστηριακούς σκοπούς και το κόστος.

Το υλικό που επιλέχθηκε για το βασικό κορμό της κατασκευής είναι το αλουμίνιο, καθώς πρόκειται για ένα υλικό ελαφρύ και εύκολο στην κατεργασία. Όσο αφορά το κόστος του, δεν είναι ιδιαίτερα φτηνό αλλά σε συνδυασμό με το βάρος του και την ευκολία στην κατεργασία αποτελεί, την ιδανική επιλογή.

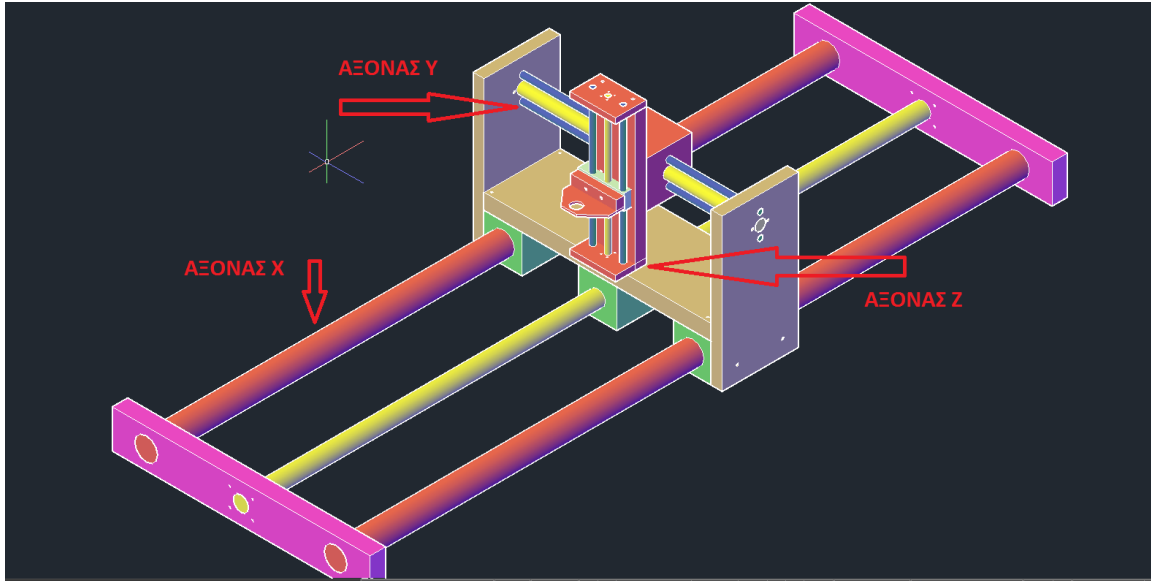
Η κατασκευή πρέπει να μπορεί να παρέχει σταθερή και απόλυτα ελεγχόμενη κίνηση του διατρητικού εργαλείου σε σχέση με την πλακέτα. Για να επιτευχθεί αυτό, διαθέτει 3 κινητά μέρη που κινούνται το κάθε ένα κατά έναν από τους 3 άξονες X, Y και Z.

Οι έξι άξονες μεγέθους Φ14 είναι από χάλυβα ολικής βαφής και χρωμίωσης είναι ρεκτιφιαρισμένοι και χρησιμοποιούνται για ευθύγραμμη κίνηση. Επίσης για τη κίνηση χρησιμοποιήθηκαν 3 κομμάτια κοχλίας μεγέθους Φ12 από χάλυβα και βήμα 3mm καθώς και 3 φωσφορούχα παξιμάδια κατάλληλα για την κίνηση του περικοχλιού. Επιπρόσθετα χρησιμοποιήθηκαν 4 βάσεις για τη στήριξη του συστήματος καθώς και ένα πολυεστερικό τραπέζι εργασίας πάχος 5mm χρώματος κόκκινο.

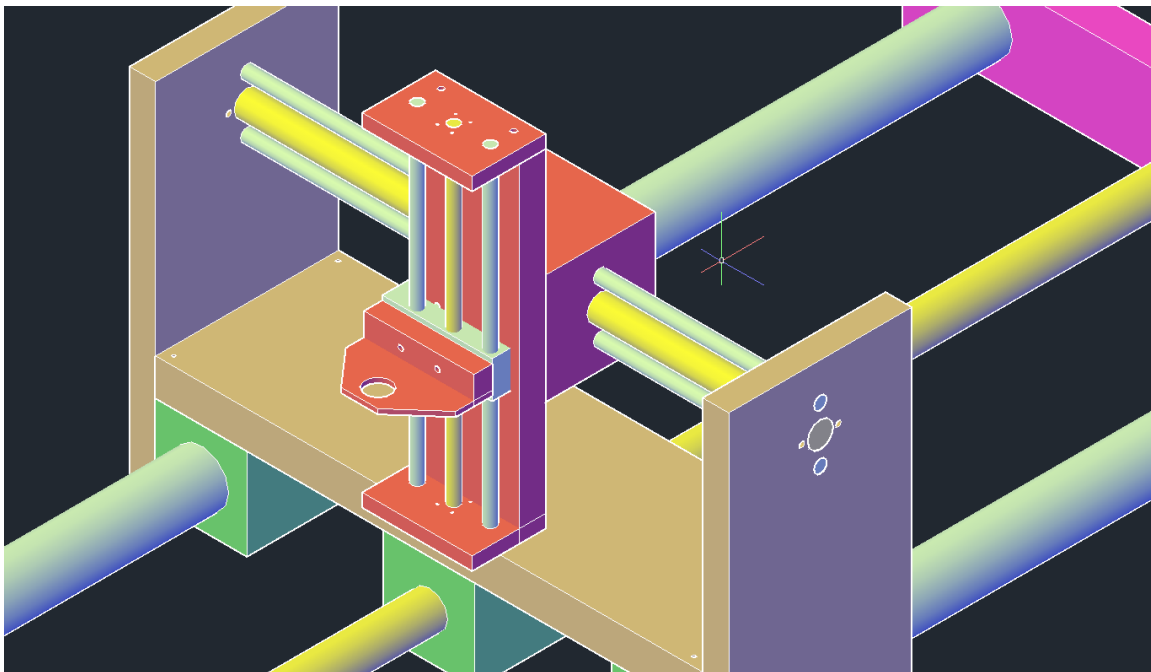
Στη συνέχεια ακολουθεί φωτογραφικό υλικό του σχεδίου στο πρόγραμμα autocad.



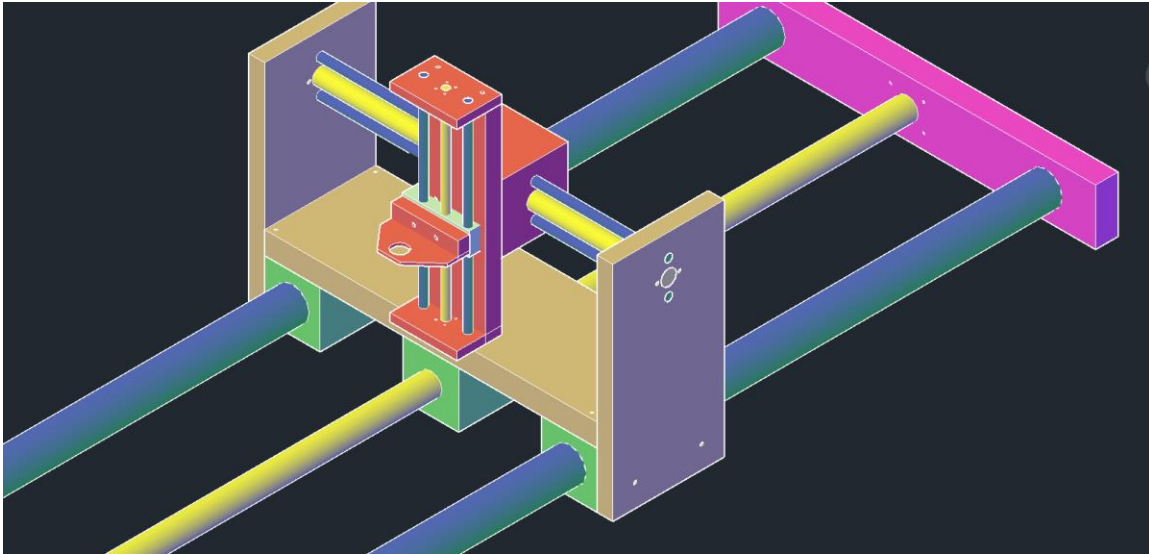
Εικόνα 3.7 3D αναπαράσταση μηχανολογικού σχεδίου στο Autocad



Εικόνα 3.8 Ορισμός αξόνων σε 3D αναπαράσταση



Εικόνα 3.9 Με κίτρινο χρώμα ο κοχλίας μεγέθους $\Phi 12$ από χάλυβα



Εικόνα 3.10 Άξονες μεγέθους Φ14 είναι από χάλυβα ολικής βαφής και χρωμίωσης

3.6 ΥΛΟΠΟΙΗΣΗ ΜΗΧΑΝΟΛΟΓΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

Η πρώτη φάση υλοποίησης αφορά το μηχανολογικό μέρος της κατασκευής. Αρχικά έγινε η αγορά των κομματιών αλουμινίου που στελεχώνουν το κύριο μέρος της κατασκευής. Στη συνέχεια υπέστησαν επεξεργασία ώστε να γίνει σταδιακά η μοντελοποίηση του μηχανήματος.

Η επεξεργασία έγινε σε τόρνο, φρέζα, κορδέλα και τρυπάνι έτσι ώστε να μπορέσουν να τοποθετηθούν τα ρουλεμάν, οι ράβδοι οδήγησης, οι κοχλίες και φυσικά οι αντάπτορες που θα στηριχτούν οι κινητήρες. Η πρώτη ενέργεια ήταν η κοπή των ράβδων οδήγησης σε κορδέλα και η επεξεργασία των βάσεων της μηχανής έτσι ώστε να συναρμολογηθούν μεταξύ τους και να συντελέσουν το σκελετό της κατασκευής. Έπειτα έγινε επεξεργασία σε κάθε κομμάτι ξεχωριστά και συναρμολογήθηκαν όλα μαζί. Στη συνέχεια ακολουθεί φωτογραφικό υλικό της κατεργασίας του αλουμινίου.



Εικόνα 3.11 Βάσεις αλουμινίου



Εικόνα 3.11 Επεξεργασία της βάσης για ρουλεμάν σε τόρνο



Εικόνα 3.12 Ρουλεμαν



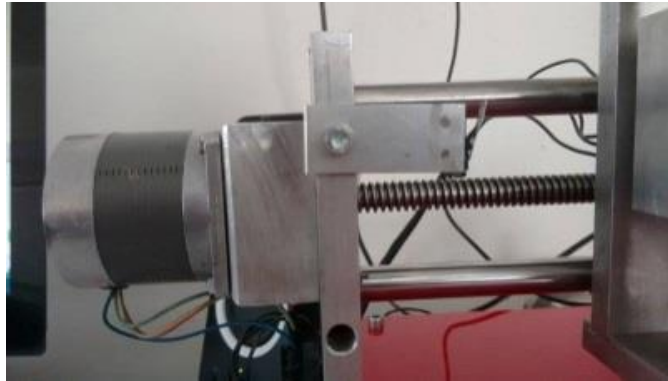
Εικόνα 3.12 Συναρμολόγηση βάσεων και αξόνων



Εικόνα 3.12 Διαδικασία συναρμολόγησης

Ιδιαίτερα δύσκολη ήταν η επεξεργασία των βάσεων με ρουλεμάν καθώς έπρεπε να γίνει τοποθέτηση ρουλεμάν και στα δύο άκρα της βάσης έτσι ώστε να επιτευχθεί βελτιστοποίηση της κίνησης του κινητού μέρους της κατασκευής. Επίσης μεγάλο βαθμό δυσκολίας συναντήθηκε στην δημιουργία του μετατροπέα για τους κινητήρες καθώς έγινε με ένα είδους μηχανισμό που λέγεται κόμπλερ. Το κόμπλερ είναι ένας μηχανισμός που χρησιμεύει για την εμπλοκή και απεμπλοκή της μετάδοσης της κίνησης μεταξύ των διάφορων μερών. Ο άξονας των κινητήρων συνδέεται με τους κοχλίες έτσι ώστε να δίνεται κίνηση στο μηχάνημα.

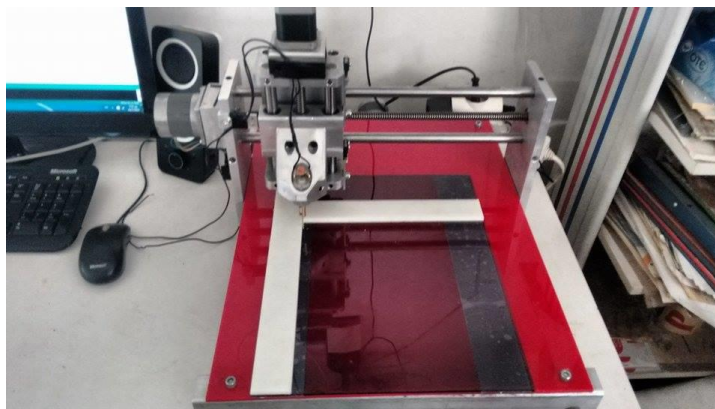
Μετά την τοποθέτηση των κινητήρων το τελευταίο κομμάτι της μηχανολογικής εργασίας ήταν η τοποθέτηση του τραπεζιού εργασίας και η τοποθέτηση των τερματικών διακοπών έτσι ώστε να αρχικοποιούμε το σημείο εκκίνησης της μηχανής. Ακολουθεί φωτογραφικό υλικό.



Εικόνα 3.13 Τοποθέτηση τερματικού διακόπτη στον άξονα Y



Εικόνα 3.14 Τερματικός διακόπτης



Εικόνα 3.15 Ολοκλήρωση της κατασκευής

3.7 ΕΠΙΛΟΓΗ ΚΙΝΗΤΗΡΩΝ

Η τελική επιλογή των κινητήρων, περιλαμβάνει τρεις βηματικούς κινητήρες και ένακινητήρα συνεχούς ρεύματος. Οι τρεις βηματικοί κινητήρες αποτελούνται από ρότορα 2 πηνίων(bipolar) οπότε η διαφορά που έχουν, δεν είναι στον τρόπο λειτουργίας τους αλλά στο βήμα.

Στον πίνακα 1 παρουσιάζονται τα βασικά χαρακτηριστικά των βηματικών κινητήρων που χρησιμοποιήθηκαν.

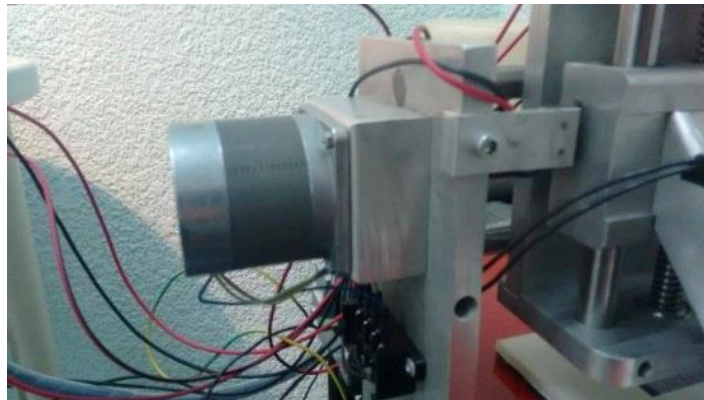
ΚΙΝΗΤΗΡΑΣ	ΜΟΙΡΕΣ/ΒΗΜΑ	ΡΕΥΜΑ/ΦΑΣΗ
X ΑΞΟΝΑΣ	1.6 ⁰ /step	1.4A/Φάση
Y ΑΞΟΝΑΣ	1.8 ⁰ /step	1.7A/Φάση
Z ΑΞΟΝΑΣ	0.9 ⁰ /step	1.7A/Φάση

Πίνακας 1 Χαρακτηριστικά βηματικών κινητήρων που χρησιμοποιήθηκαν

Ο τέταρτος κινητήρας της κατασκευής είναι ένας κινητήρας συνεχούς ρεύματος. Χρησιμοποιείτε στο τελευταίο άκρο του μηχανήματος το οποίο χρησιμοποιείται για το αυτόματο τρύπημα της πλακέτας. Πρόκειται για ένα μοτέρ 12V και 6600 στροφών, τοποθετημένο σε ειδικό αντάπτορα για τη στήριξη του. Στον άξονα του έχει τοποθετηθεί τσοκ στο οποίο μπαίνει ένα τρυπάνι κοβαλτίου 2 mm.



Εικόνα 3.16 Κινητήρας στον άξονα Z



Εικόνα 3.17 Κινητήρας στον άξονα Y



Εικόνα 3.18 Κινητήρας στον άξονα X



Εικόνα 3.19 Κινητήρας συνεχούς ρεύματος

ΚΕΦΑΛΑΙΟ 4

ΗΛΕΚΤΡΟΝΙΚΑ

ΕΙΣΑΓΩΓΗ

Στο προηγούμενο κεφάλαιο αναλύθηκε το μηχανολογικό κομμάτι του CNCμηχανήματος και έγινε παρουσίαση των κινητήρων που χρησιμοποιήθηκαν. Στο κεφάλαιο αυτό θα αναλυθεί λεπτομερώς το ηλεκτρονικό κομμάτι της κατασκευής αλλά και η τεχνολογία λογισμικού και προγραμματισμού που χρησιμοποιήθηκε. Πιο συγκεκριμένα θα αναλυθεί ο οδηγός των βηματικών κινητήρων που χρησιμοποιήθηκε, ο επεξεργαστής και ο προγραμματισμός του.

4.1 ΟΔΗΓΟΣ ΚΙΝΗΣΗΣ ΚΙΝΗΤΗΡΑ (A4988)

Για την καθοδήγηση των βηματικών κινητήρων χρησιμοποιήθηκαν οδηγοί A4988 της pololu. Πρόκειται για ένα ολοκληρωμένο σύστημα οδήγησης βηματικού κινητήρα το οποίο διαθέτει ενσωματωμένο μεταφραστή και είναι πολύ εύκολο στο χειρισμό.

Ο συγκεκριμένος οδηγός αντέχει μέχρι 2Α και είναι σχεδιασμένος για να λειτουργεί σε διπολικούς κινητήρες και χρησιμοποιεί παροχή ρεύματος 8-35V καθώς επίσης και ρυθμιστή ως προς την ένταση ρεύματος που θα παρέχει. Ο μεταφραστής είναι το κλειδί για την εύκολη χρήση του A4988 καθώς δίνοντας απλά ένα παλμό στην είσοδο του βήματος ο κινητήρας κάνει το πρώτο μικρό βήμα.

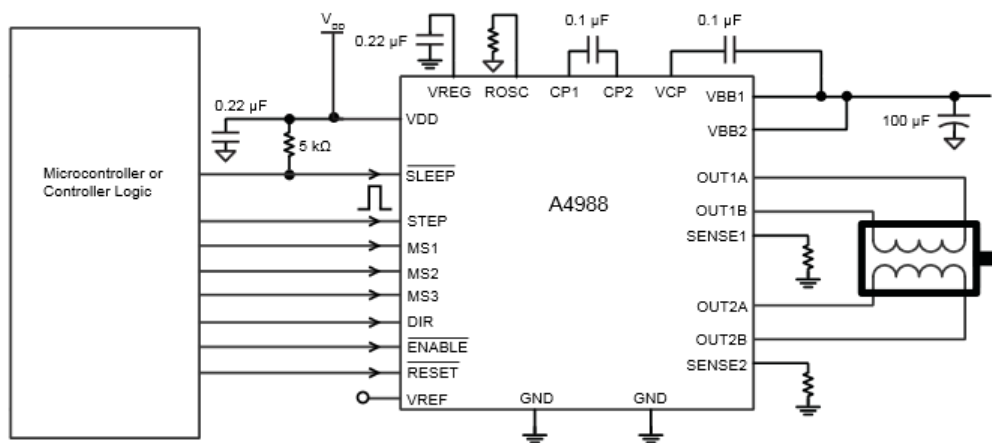
Κατά τη διάρκεια της κίνησης, ο έλεγχος ψαλιδισμού επιλέγει αυτόματα μία κατάσταση νεκρού χρόνου, η οποία είναι αργή ή μικτή. Στη μικτή λειτουργία η συσκευή αρχικοποιείται σε μία γρήγορη κατάσταση νεκρού χρόνου. Η μικτή κατάσταση έχει ως αποτέλεσμα να μειώνει τον ακουστικό θόρυβο του κινητήρα, να έχει αυξημένη ακρίβεια στο βήμα και μειωμένη κατανάλωση ισχύος. Επίσης αξίζει να αναφερθεί ότι ο οδηγός διαθέτει ένα εσωτερικό κύκλωμα ελέγχου διόρθωσης, για τη βελτίωση της κατανάλωσης ισχύος παρέχοντας τετράγωνο μεταβαλλόμενο συνεχές ρεύμα (PWM). Το εσωτερικό αυτό κύκλωμα προστασίας περιλαμβάνει, θερμική διακοπή με υστέρηση και διακοπή ρεύματος.

Επιπρόσθετα ο οδηγός αυτός διαθέτει και μία κατάσταση sleepmode, κατά την οποία ελαχιστοποιείται η κατανάλωση ενέργειας όταν ο κινητήρας δεν είναι σε χρήση. Η λειτουργία αυτή απενεργοποιεί μεγάλο μέρος του εσωτερικού κυκλώματος, συμπεριλαμβανομένων των FET εξόδου και τον σταθεροποιητή ρεύματος. Ένα λογικό 0 στο pin της λειτουργίας αυτής βάζει τον οδηγό σε κατάσταση sleepmode. Αντίθετα με ένα

λογικό 1 ο οδηγός ξεκινάει να δουλεύει κανονικά. Η καθυστέρηση στην αλλαγή της κατάστασης από το sleepmode σε κανονική διαρκεί 1ms πριν την εντολή του βήματος.

Ο οδηγός A4988 είναι μία ιδανική διάταξη για εφαρμογές στις οποίες δεν υπάρχει διαθέσιμος επεξεργαστής. [9]

Typical Application Diagram



Εικόνα 4.1 Λογικό διάγραμμα του οδηγού (A4988)

Στον πίνακα 2 παρουσιάζονται τα χαρακτηριστικά του οδηγού (A4988):

Χαρακτηριστικό	Συμβολισμός	Τιμή	Μονάδα μέτρησης
Load supply voltage	V_{BB}	35	V
Output current	I_{OUT}	2	A
Logic input voltage	V_{IN}	-0.3 με 5.5	V
Logic supply voltage	V_{DD}	-0.3 με 5.5	V
VBBx to OUTx		35	V
Sense voltage	V_{SENSE}	0.5	V
Reference voltage	V_{REF}	5.5	V
Operating Ambient Temperature	T_A	-20 με 85	°C
Maximim Junction	$T_J(max)$	150	°C
Storage Temperature	T_{stg}	-55 με 150	°C

Πίνακας 2 Χαρακτηριστικά βηματικών κινητήρων που χρησιμοποιήθηκαν

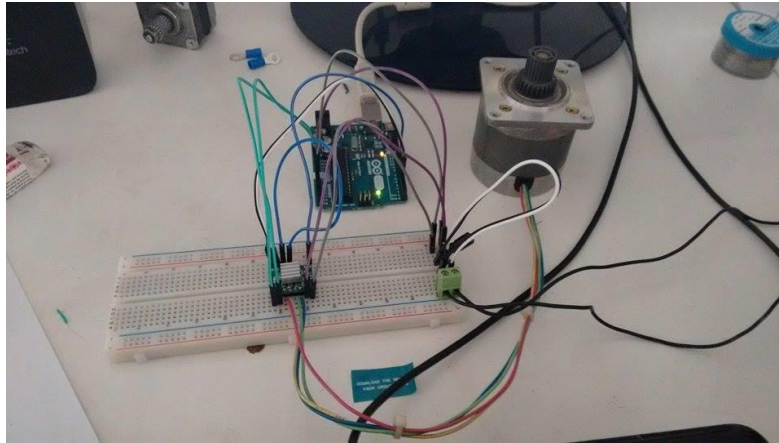
4.2 ΗΛΕΚΤΡΟΝΙΚΟ ΚΥΚΛΩΜΑ ΚΙΝΗΣΗΣ ΒΗΜΑΤΙΚΟΥΚΙΝΗΤΗΡΑ (A4988)

Για να είναι δυνατόν να ελεγχθούν και να οδηγηθούν οι κινητήρες χρησιμοποιήθηκε ένα κύκλωμα οδήγησης. Η ύπαρξη τριών βηματικών κινητήρων απαιτούσε 3 ξεχωριστούς οδηγούς, έναν για κάθε κινητήρα.

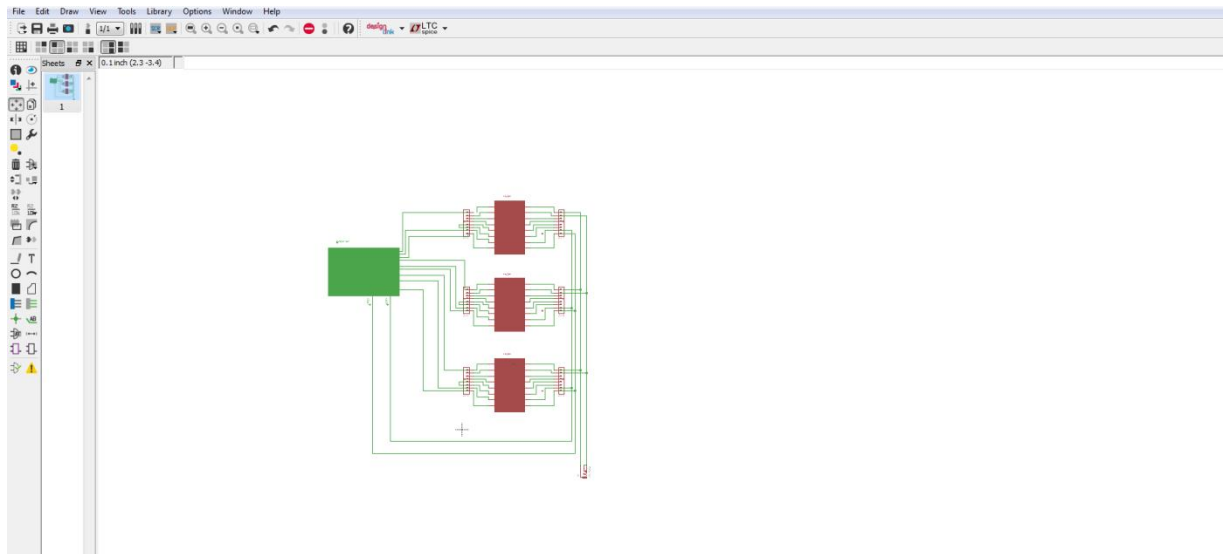
Αρχικά χρησιμοποιήθηκε ένας οδηγός A4988 πειραματικά σε breadboard, για να κατανοηθεί πλήρως η λειτουργία του και αν είναι κατάλληλος για αυτή την εργασία. Το πειραματικό κύκλωμα περιλάμβανε ένα βηματικό κινητήρα, έναν οδηγό A4988, ένα τροφοδοτικό για πειραματικό και τον μικροελεγχτή. Η λειτουργία του οδηγού κρίθηκε αρκετά ικανοποιητική και για αυτό το λόγο κατασκευάστηκε μία πλακέτα με τρεις οδηγούς επάνω. Στην ουσία πάνω στην πλακέτα είναι τρία ίδια αλλά διαφορετικά κυκλώματα. Το μικρό κόστος αλλά και τα χαρακτηριστικά του οδηγού αυτού έκαναν την επιλογή αυτή, ιδανική.

Ο κάθε οδηγός αποτελείται από 16 pin με αποτέλεσμα να τοποθετηθούν κλέμες 8 εισόδων/εξόδων από την κάθε μεριά. Στη πλακέτα επίσης τοποθετήθηκε ένα βύσμα τροφοδοσίας, για τα 12V που χρειάζονται οι οδηγοί καθώς επίσης και τρεις διπλές κλέμες για τη διανομή της τροφοδοσίας αυτής. Τα pins 1,2,9,10 και 16 συνδέθηκαν στο ARDUINO. Τα pins 11 και 12 βραχυκυκλώθηκαν μεταξύ τους και τα 3,4,5,6 συνδέθηκαν

στους κινητήρες. Τέλος τα pins 7 και 8 είναι συνδεδεμένα στην τροφοδοσία 12V για τους κινητήρες. Η σχεδίαση της πλακέτας έγινε στο πρόγραμμα EAGLE.



Εικόνα 4.2 Πειραματικό κομμάτι ελέγχου οδηγού κινητήρα



Εικόνα4.3 Σχεδίαση κυκλώματος στο πρόγραμμα EAGLE

4.3 ARDUINO

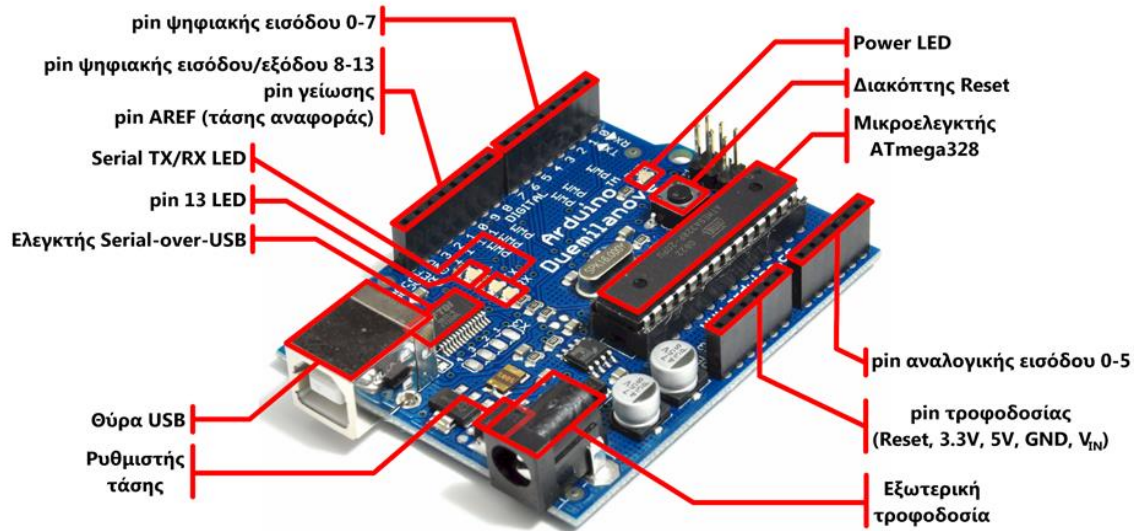
Όπως αναφέρθηκε και προηγουμένως το Arduino είναι μια «ανοικτού κώδικα» πλατφόρμα «πρωτοτυποποίησης» ηλεκτρονικών βασισμένη σε ευέλικτο και εύκολο στη χρήση hardware και software που προορίζεται για οποιονδήποτε έχει λίγη προγραμματιστική εμπειρία, στοιχειώδεις γνώσεις ηλεκτρονικών και ενδιαφέρεται να δημιουργήσει διαδραστικά αντικείμενα ή περιβάλλοντα. Στην ουσία, πρόκειται για ένα

ηλεκτρονικό κύκλωμα που βασίζεται στον μικροελεγκτή ATmega της Atmel και του οποίου όλα τα σχέδια, καθώς και το software που χρειάζεται για την λειτουργία του, διανέμονται ελεύθερα και δωρεάν ώστε να μπορεί να κατασκευαστεί από τον καθένα.

Αφού κατασκευαστεί, μπορεί να συμπεριφερθεί σαν ένας μικροσκοπικός υπολογιστής, αφού ο χρήστης μπορεί να συνδέσει επάνω του πολλαπλές μονάδες εισόδου/εξόδου και να προγραμματίσει τον μικροελεγκτή να δέχεται δεδομένα από τις μονάδες εισόδου, να τα επεξεργάζεται και να στέλνει κατάλληλες εντολές στις μονάδες εξόδου.

Η ρομποτική είναι μια από τις πολλές εφαρμογές στις οποίες το Arduino διαπρέπει. Το Arduino βασίζεται στον ATmega328, έναν 8-bit RISC μικροελεγκτή, τον οποίο χρονίζει στα 16MHz. Ο ATmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων:

1. 2Kb μνήμης SRAM που είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματά σας για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. κατά το runtime. Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή αν γίνει reset.
2. 1Kb μνήμης EEPROM η οποία μπορεί να χρησιμοποιηθεί για «ωμή» εγγραφή/ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματά σας κατά το runtime. Σε αντίθεση με την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset οπότε είναι το ανάλογο του σκληρού δίσκου.
3. 32Kb μνήμης Flash, από τα οποία τα 2Kb χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware αυτό που στην ορολογία του Arduino ονομάζεται bootloader είναι αναγκαίο για την εγκατάσταση των δικών σας προγραμμάτων στον μικροελεγκτή μέσω της θύρας USB, χωρίς δηλαδή να χρειάζεται εξωτερικός hardware programmer. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή σας. Η μνήμη Flash, όπως και η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset. Επίσης, ενώ η μνήμη Flash υπό κανονικές συνθήκες δεν προορίζεται για χρήση runtime μέσα από τα προγράμματά σας, λόγω της μικρής συνολικής μνήμης που είναι διαθέσιμη σε αυτά (2Kb SRAM + 1Kb EEPROM), έχει σχεδιαστεί μια βιβλιοθήκη που επιτρέπει την χρήση όσου χώρου περισσεύει (30Kb μείον το μέγεθος του προγράμματός σας σε μεταγλωττισμένη μορφή).



Εικόνα 4.4 Ανάλυση πλακέτας ARDUINO

Όσο αφορά για τη γλώσσα προγραμματισμού η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring, μια παραλλαγή C/C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega, και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++. Για compiler χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc.

Λόγω της καταγωγής της από την C, στην γλώσσα του Arduino μπορείτε να χρησιμοποιήσετε ουσιαστικά τις ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπων δεδομένων και τους ίδιους τελεστές όπως και στην C. Πέρα από αυτές όμως, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για την διαχείριση του ειδικού hardware του Arduino. [8]

4.4 ΣΥΝΔΕΣΜΟΛΟΓΙΑ ARDUINO

Όπως αναφέρθηκε και παραπάνω το ARDUINO είναι η καρδιά στο ηλεκτρονικό κομμάτι της κατασκευής. Χρησιμοποιήθηκαν όλες οι ψηφιακές εισοδοι/έξοδοι, μία αναλογική και δυο pin τροφοδοσίας.

Στον πίνακα 3 αναλύεται η συνδεσμολογία που ακολουθήθηκε:

Pin	Σήμανση	Συνδεσμολογία
2	Digital	Σύνδεση στο τερματικό 1

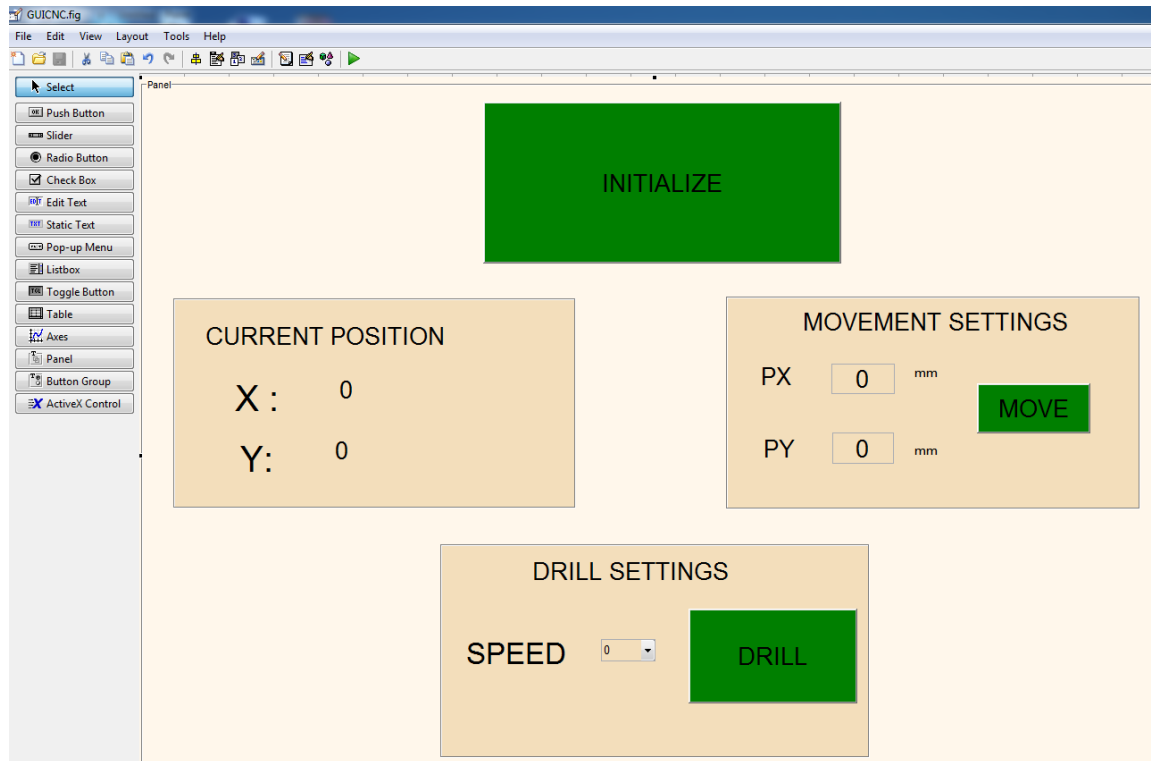
3	Digital	Σύνδεση στο τερματικό 2
4	Digital	Σύνδεση στον driver 3 στο pin direction
5	Digital	Σύνδεση στον driver 3 στο pin step
6	Digital	Σύνδεση στον driver 2 στο pin direction
7	Digital	Σύνδεση στον driver 2 στο pin step
8	Digital	Σύνδεση στον driver 1 στο pin direction
9	Digital	Σύνδεση στον driver 1 στο pin step
10	Digital	Σύνδεση στον driver 1 στο pin enable
11	Digital	Σύνδεση στον driver 2 στο pin enable
12	Digital	Σύνδεση στον driver 3 στο pin enable
13	Digital	Σύνδεση στο τερματικό 3
A0	Analog	Σύνδεση στο τερματικό 4
5V	Power	Σύνδεση στους drivers στο pin VDD
GND	Power	Σύνδεση στους drivers στο pin GND

Πίνακας 3 Σύνδεσμολογία ARDUINO με τον οδηγό (A4988) και τερματικούς διακόπτες

4.5 ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΔΙΕΠΑΦΗΣ

Όπως αναφέρθηκε και παραπάνω το MATLAB είναι ένα πρόγραμμα υπολογιστών για ανθρώπους που χρησιμοποιούν αριθμητικούς υπολογισμούς, ειδικά στη γραμμική άλγεβρα (πίνακες). Ξεκίνησε ως ένα πρόγραμμα "Εργαστηρίου Πινάκων" ("MATrixLABoratory") που είχε σκοπό να παρέχει αλληλεπιδρώσα προσπέλαση στις βιβλιοθήκες Linpack και Eispack. Από τότε έχει αναπτυχθεί αρκετά, για να γίνει ένα ισχυρότατο εργαλείο στην οπτικοποίηση, στον προγραμματισμό, στην έρευνα, στην επιστήμη των μηχανικών, και στις επικοινωνίες.

Για την ανάπτυξη γραφικού περιβάλλοντος με σκοπό την διευκόλυνση του χρήστη στη λειτουργία του CNC αναπτύχθηκε ένα γραφικό περιβάλλον διεπαφής στο πρόγραμμα MATLAB.



Εικόνα 4.5 Γραφικό περιβάλλον διεπαφής

Σε αυτό το περιβάλλον διεπαφής, ο χρήστης εισάγει τις τιμές(Px,Py) των σημείων που επιθυμεί και πατώντας το κουμπί “MOVE” και πραγματοποιείται η κίνηση.Αφού οριστούν τα (Px,Py) αποθηκεύονται στο “CURRENTPOSITION” και εν συνεχεία εισάγονται τα καινούρια (Px,Py) με σκοπό τον καθορισμό του επόμενου σημείου. Στη συνέχεια αφού καθοριστεί το σημείο που θα τρυπηθεί ο χρήστης επιλέγει τις παραμέτρους του τρυπήματος.

Ο χρήστης διαθέτει την επιλογή της ταχύτητας του άξονα z’ ανάλογα το υλικό που θέλει να επεξεργαστεί. Στη συνέχεια αφού επιλέξει την ταχύτητα που επιθυμεί και πατώντας το κουμπί “DRILL” και πραγματοποιείται η ενέργεια του τρυπήματος.

Η τελευταία επιλογή του χρήστη είναι το κουμπί “INITIALIZE”.Με αυτή την επιλογή ο χρήστης έχει τη δυνατότητα να επαναφέρει τους άξονες στην αρχή τους.

4.6 ΕΠΙΚΟΙΝΩΝΙΑ MATLAB ΜΕ ARDUINO

Η δημιουργία του γραφικού περιβάλλοντος διεπαφήςήταν αποτέλεσμα της επικοινωνίας του Arduinoκαι του matlabμέσω σειριακής.Στην ουσία οι εντολές προγραμματισμού που χρησιμοποιούνται στο matlabχρησιμοποιούνται και στο Arduino.

Αρχικά έγινε μια κωδικοποίηση των συναρτήσεων του Arduino έτσι ώστε να είναι δυνατόν να χρησιμοποιούνται στο περιβάλλον του matlab καθώς δεν ήταν δυνατόν να εκτελούνται όλες οι εντολές στο περιβάλλον του matlab. Στη συνέχεια αφού καθορίστηκε το γραφικό περιβάλλον ακολούθησε ο προγραμματισμός του γραφικού περιβάλλοντος.

Στον πίνακα 5 αναλύεται το γραφικό περιβάλλον και ο προγραμματισμός του.

MATLAB	ARDUINO
INITIALIZE	Συνάρτηση STARTUPZERO
MOVE	Συνάρτηση MOVECNC
DRILL	Συνάρτηση DRILL

Πίνακας 5 Ανάλυση γραφικού περιβάλλοντος διεπαφής

4.7 ΑΡΧΙΚΑ ΠΕΙΡΑΜΑΤΑ

Κατά τη διάρκεια της εργασίας αυτής έγιναν πολλά πειράματα για να φτάσει το CNC στην τελική του μορφή.

Αρχικά, το πρώτο πείραμα είχε να κάνει με τα μηχανολογικά μέρη της κατασκευής, όπως τα ρουλεμάν, οι ντίζες και οι κοχλίες. Επίσης πολύ μεγάλη σημασία είχε η επιλογή των κινητήρων. Πιο συγκεκριμένα, δοκιμάστηκε το κάθε μοτέρ ξεχωριστά στον οδηγό A4988 και στο arduino έτσι ώστε να εξακριβωθεί αν είναι κατάλληλο για το CNC. Στη συνέχεια έπρεπε να γίνει η επιλογή του οδηγού κίνησης των κινητήρων. Ο λόγος που προτιμήθηκε ο A4988 ήταν για λόγους χαμηλού κόστους.

Το τελευταίο πείραμα είχε να κάνει με τον προγραμματισμό της κατασκευής καθώς ήταν συγκεκριμένες οι εντολές που λειτουργούν και στο Arduino και στο Matlab με αποτέλεσμα να γίνουν πολλές αλλαγές.

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ/ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

5.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Ολοκληρώνοντας την παρούσα εργασία και έχοντας μελετήσει διεξοδικά τα ζητήματα που πραγματεύτηκε, μπορούμε να εξάγουμε κάποια πολύ χρήσιμα συμπεράσματα.

Αρχικά ότι το CNC για εργαστηριακούς σκοπούς που κατασκευάστηκε, είναι ένα αξιόπιστο μηχάνημα το οποίο μπορεί να εκτελεί με ακρίβεια τρύπες σε πλακέτες αλλά και σε όποιο άλλο υλικό επιθυμεί ο χρήστης. Επίσης το εύχρηστο γραφικό περιβάλλον διεπαφής κάνει αυτή τη κατασκευή ένα πολύτιμο εργαλείο για χρήστες που δεν έχουν σχέση με τον τομέα της τεχνολογίας.

Επιπρόσθετα μεγάλο πλεονέκτημα είναι το μεγάλο εύρος εργασιών που εκτελεί το μηχάνημα αυτό, καθώς με μία μικρή αλλαγή του τελευταίου άκρου μπορεί εκτός από το τρύπημα να χαράζει διάφορες επιφάνειες. Επίσης το βάρος του αποτελεί ένα επιπλέον πλεονέκτημα, καθώς μπορεί να μετακινηθεί πολύ εύκολα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] ΡΟΜΠΟΤΙΚΗ 3^η ΕΚΔΟΣΗ (Δ.Μ.ΕΜΙΡΗΣ-Δ.Ε.ΚΟΥΛΟΥΡΙΩΤΗΣ)

[2] http://3ogelptolrobot.weebly.com/uploads/9/6/1/0/9610973/tipoi_robot.pdf

[3] <https://el.wikipedia.org/wiki/Arduino>

[4] ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ-ΑΠΟΣΤΟΛΟΣ Κ. ΣΕΡΓΙΑΔΗΣ-ΔΙΕΡΕΥΝΗΣΗ ΛΕΙΤΟΥΡΓΙΑΣ ΚΑΙ ΚΑΤΑΣΚΕΥΗ ΟΔΗΓΗΣΗΣ ΒΗΜΑΤΙΚΟΥ ΚΙΝΗΤΗΡΑ ΜΕΣΩ ΠΑΡΑΛΛΗΣ ΘΥΡΑΣ-ΑΘΗΝΑ, ΜΑΙΟΣ 2009

[5] ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ-ΧΑΡΑΛΑΜΠΟΣ Α. ΓΡΗΓΟΡΟΠΟΥΛΟΣ-ΓΡΑΜΜΙΚΟΣ ΒΗΜΑΤΙΚΟΣ ΚΙΝΗΤΗΡΑΣ ΜΕ ΣΕΡΒΟΕΛΕΓΚΤΗ-ΑΘΗΝΑ, ΜΑΡΤΙΟΣ 2013

[6] ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ- ΣΑΚΑΡΟΣ ΘΩΜΑΣ- ΚΑΤΑΣΚΕΥΗ ΚΑΙ ΕΛΕΓΧΟΣ ΚΑΤΑΚΟΡΥΦΟΥ ΑΡΘΡΩΤΟΥ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ ΤΕΣΣΑΡΩΝ ΒΑΘΜΩΝ ΕΛΕΥΘΕΡΙΑΣ- Χανιά, Δεκέμβριος 2009

[7] <http://drawwithautocad.blogspot.gr/2009/05/autocad.html>

[8] <http://deltahacker.gr/arduino-intro/>

[9] <http://www.farnell.com/datasheets/1923364.pdf>

[10] ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ-ΑΚΗΣ Α. ΤΖΙΑΜΑΛΗΣ-ΣΕΡΒΟΚΙΝΗΤΗΡΕΣ ΚΑΙ ΣΕΡΒΟΚΙΝΗΤΗΡΙΑ ΣΥΣΤΗΜΑΤΑ-ΑΘΗΝΑ, ΙΑΝΟΥΑΡΙΟΣ 20

ΠΑΡΑΡΤΗΜΑ

Ακολουθεί ο κώδικας στο MATLAB για το γραφικό περιβάλλον διεπαφής και ο κώδικας στο ARDUINOMATLAB :

```
function varargout = GVICNC(varargin)
% GVICNC MATLAB code for GVICNC.fig
%   GVICNC, by itself, creates a new GVICNC or
raises the existing
%   singleton*.
%
%   H = GVICNC returns the handle to a new GVICNC or
the handle to
%   the existing singleton*.
%
%   GVICNC('CALLBACK',hObject,eventData,handles,...)
calls the local
%   function named CALLBACK in GVICNC.M with the
given input arguments.
%
%   GVICNC('Property','Value',...) creates a new
GVICNC or raises the
%   existing singleton*. Starting from the left,
property value pairs are
%   applied to the GUI before GVICNC_OpeningFcn gets
called. An
%   unrecognized property name or invalid value
makes property application
%   stop. All inputs are passed to
GVICNC_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose
"GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
GVICNC

% Last Modified by GUIDE v2.5 24-Sep-2015 23:46:40

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
```

```

gui_State = struct('gui_Name',      mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn', @GUICNC_OpeningFcn, ...
'gui_OutputFcn',  @GUICNC_OutputFcn, ...
'gui_LayoutFcn',  [] , ...
'gui_Callback',   []);
if nargin && ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GUICNC is made visible.
function GUICNC_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject      handle to figure
% eventdata    reserved - to be defined in a future
version of MATLAB
% handles      structure with handles and user data (see
GUIDATA)
% varargin     command line arguments to GUICNC (see
VARARGIN)

%fclose(arduino); % end communication with arduino
% initiate arduino communication
% Choose default command line output for GUICNC
delete(instrfind({'Port'}, {'COM3'}))
global arduino;
global curr_x;
global curr_y;
curr_x = 0;
curr_y = 0;
arduino = serial('COM3'); % create serial communication
object on port COM4
arduino.BaudRate = 38400;
% arduino.StopBits = 1;

```

```
% arduino.Terminator = 'CR/LF';
fopen(arduino);
disp('Connecting with arduino...');
pause(10);
handles.output = hObject;
guidata(hObject, handles);

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUICNC wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the
command line.
function varargout = GUICNC_OutputFcn(hObject,
eventdata, handles)
% varargout cell array for returning output args (see
VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future
version of MATLAB
% handles structure with handles and user data (see
GUIDATA)

% Get default command line output from handles
structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata,
handles)

% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future
version of MATLAB
% handles structure with handles and user data (see
GUIDATA)
global arduino;
fprintf(arduino, '%d\n', 99)
set(handles.text3, 'String', 0)
```

```
set(handles.text1,'String', 0)
% fclose(arduino); % end communication with arduino

function edit7_Callback(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future
version of MATLAB
% handles      structure with handles and user data (see
GUIDATA)
% Hints: get(hObject,'String') returns contents of
edit7 as text
%           str2double(get(hObject,'String')) returns
contents of edit7 as a double

% --- Executes during object creation, after setting
all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future
version of MATLAB
% handles      empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future
version of MATLAB
% handles      structure with handles and user data (see
GUIDATA)
```



```
% Hints: get(hObject,'String') returns contents of
edit6 as text
%         str2double(get(hObject,'String')) returns
contents of edit6 as a double
% global arduino;
% if str2num(get(handles.edit7 , 'String')) >= 0 &
str2num(get(handles.edit6 , 'String')) <= 50
%     fprintf(arduino , '%d\n',06)
%     pause(0.1);
%     fprintf(arduino , '%d\n',
str2num(get(handles.edit6 , 'String')))
%
% end

% --- Executes during object creation, after setting
all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata,
handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
global curr_x ;
global curr_y ;
global arduino;
```

```

%disp('mpika kai edw');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% edwstelnw to new_y
if str2num(get(handles.edit7 , 'String')) >= 0 &
str2num(get(handles.edit7 , 'String')) <= 500
%disp('mpika kai edw2');
%   fprintf(arduino , '%d\r\n %d\r\n', [06
str2num(get(handles.edit7 , 'String')) ])
fprintf(arduino , '%d', 06)
pause(1);
fprintf(arduino , '%d', str2num(get(handles.edit7
, 'String')))
%   fprintf(arduino , '%d\n',
str2double(get(handles.edit7 , 'String')))
%disp('mpika kai edw3');
%   sprintf(' ESTEILA TO   %d',
str2double(get(handles.edit7 , 'String')))
%   sprintf('%d', str2num(get(handles.edit7
, 'String')))
%fread(arduino)
end

pause(2);
if str2num(get(handles.edit6 , 'String')) >= 0 &
str2num(get(handles.edit6 , 'String')) <= 500
%disp('mpika kai edw2');
fprintf(arduino , '%d', 05)
pause(1);
fprintf(arduino , '%d', str2num(get(handles.edit6
, 'String')))
%disp('mpika kai edw3');
%sprintf('%d', str2double(get(handles.edit6
, 'String')))
%sprintf('%d', str2num(get(handles.edit6 , 'String')))

end

pause(2);
fprintf(arduino , '%d', 00)

curr_x= str2num(get(handles.edit6 , 'String')) ;
curr_y= str2num(get(handles.edit7 , 'String')) ;
set(handles.text3, 'String', curr_x)
set(handles.text1, 'String', curr_y)
% fread(arduino);

```

```
% --- Executes during object creation, after setting
all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting
all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of
edit2 as text
%         str2double(get(hObject,'String')) returns
contents of edit2 as a double
global arduino;
if str2num(get(handles.edit2 , 'String')) >= 50 &
str2num(get(handles.edit2 , 'String')) <= 100
fprintf(arduino , '%d\n',02)
pause(0.1);
fprintf(arduino , '%d\n', str2num(get(handles.edit2
, 'String'))))

end

% --- Executes during object creation, after setting
all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit3 as text
%         str2double(get(hObject,'String')) returns
contents of edit3 as a double
```

```

global arduino;
if str2num(get(handles.edit3 , 'String')) >= 50 &
str2num(get(handles.edit3 , 'String')) <= 100
fprintf(arduino , '%d\n', 03)
pause(0.1);
fprintf(arduino , '%d\n', str2num(get(handles.edit3
, 'String')))

end

% --- Executes during object creation, after setting
all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata,
handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
global arduino;
fprintf(arduino , '%d\n', 01)

% --- Executes on key press with focus on edit4 and
none of its controls.

```

```

function edit4_KeyPressFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  structure with the following fields (see
UICONTROL)
%   Key: name of the key that was pressed, in lower
case
%   Character: character interpretation of the key(s)
that was pressed
%   Modifier: name(s) of the modifier key(s) (i.e.,
control, shift) pressed
% handles    structure with handles and user data (see
GUIDATA)
global arduino;
%disp('mpika kai edw');
if str2num(get(handles.edit4 , 'String')) >= 0 &
str2num(get(handles.edit4 , 'String')) <= 50
%disp('mpika kai edw2');
fprintf(arduino , '%d\n', 04)
pause(1);
fprintf(arduino , '%d\n', str2double(get(handles.edit4
, 'String')))
%disp('mpika kai edw3');
sprintf('%d', str2double(get(handles.edit4 , 'String')))

end

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata,
handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: contents = cellstr(get(hObject, 'String'))
returns popupmenu1 contents as cell array
%   contents{get(hObject, 'Value')} returns
selected item from popupmenu1
global arduino;
% disp('mpika kai edw');
contents = get(handles.popupmenu1, 'String');
poplvalue = contents{get(handles.popupmenu1, 'Value')};

```

```
if str2num(pop1value) >= 0 & str2num(pop1value) <= 50
%disp('mpika kai edw2');
fprintf(arduino , '%d', 04)
pause(1);
fprintf(arduino , '%d', str2num(pop1value))
%disp('mpika kai edw3');
fread(arduino)
fread(arduino)

%sprintf('%d', str2num(pop1value))
end

% --- Executes during object creation, after setting
all properties.
function popupmenu1_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: popupmenu controls usually have a white
background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata,
handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: contents = cellstr(get(hObject, 'String'))
returns popupmenu2 contents as cell array
```

```

%         contents{get(hObject,'Value')} returns
selected item from popupmenu2
globalarduino;
% disp('mpika kai edw');
contents = get(handles.popupmenu2, 'String');
pop2value = contents{get(handles.popupmenu2, 'Value')};
if str2num(pop2value) >= 0 & str2num(pop2value) <= 100
%disp('mpika kai edw2');
fprintf(arduino , '%d\n', 02)
pause(1);
fprintf(arduino , '%d\n', str2num(pop2value))
%     disp('mpika kai edw3');
%fread(arduino)

%sprintf('%d', str2num(pop1value))
end

% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata,
handles)
% hObject     handle to popupmenu3 (see GCBO)
% eventdata   reserved - to be defined in a future
version of MATLAB
% handles     structure with handles and user data (see
GUIDATA)

% Hints: contents = cellstr(get(hObject,'String'))
returns popupmenu3 contents as cell array
%         contents{get(hObject,'Value')} returns
selected item from popupmenu3
globalarduino;
% disp('mpika kai edw');
contents = get(handles.popupmenu3, 'String');
pop3value = contents{get(handles.popupmenu3, 'Value')};
if str2num(pop3value) >= 0 & str2num(pop3value) <= 100
%     disp('mpika kai edw2');
fprintf(arduino , '%d\n', 03)
pause(1);
fprintf(arduino , '%d\n', str2num(pop3value))
%     disp('mpika kai edw3');
%     fread(arduino)

```



```
%      sprintf('%d', str2num(pop1value))
end

% --- Executes during object creation, after setting
all properties.
function popupmenu3_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to popupmenu3 (see GCBO)
% eventdata    reserved - to be defined in a future
version of MATLAB
% handles      empty - handles not created until after
all CreateFcns called

% Hint: popupmenu controls usually have a white
background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
set(hObject, 'BackgroundColor', 'white');
end

% --- Executes during object creation, after setting
all properties.
function popupmenu2_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to popupmenu2 (see GCBO)
% eventdata    reserved - to be defined in a future
version of MATLAB
% handles      empty - handles not created until after
all CreateFcns called

% Hint: popupmenu controls usually have a white
background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
set(hObject, 'BackgroundColor', 'white');
End
```

ARDUINO:

```
int pin_term1 = 2; // for x axis
```

```

int pin_term2 = 3;    // for y axis
int pin_term3 = 13;   // for z axis
int pin_term4 = A0;   // for z axis
intdelay_x =0;
intdelay_y =0;
intdelay_z =10;
intnew_x ;
intnew_y ;
intcurr_x = 0;
intcurr_y = 0;
//intcurr_z = 10;

void setup() {
  /* initialize serial
  */
  Serial.begin(38400);
  //rithmosmetaforasdedomenwn++++++++++++
  /* Ports definition
  */
  pinMode(8,OUTPUT); //dir
  pinMode(9,OUTPUT); //step
  pinMode(6,OUTPUT); //dir2
  pinMode(7,OUTPUT); //step2
  pinMode(4,OUTPUT); //dir3
  pinMode(5,OUTPUT); //step3
  pinMode(10,OUTPUT); //enable
  pinMode(11,OUTPUT); //enable2
  pinMode(12,OUTPUT); //enable3
  digitalWrite(10,HIGH); //set enable1 low
  digitalWrite(11,HIGH); //set enable2 low
  digitalWrite(12,HIGH); //set enable3 low
  pinMode(pin_term1, INPUT);
  pinMode(pin_term2, INPUT);
  pinMode(pin_term3, INPUT); //termatikokatw
  pinMode(pin_term4, INPUT);
  digitalWrite(pin_term1, HIGH);
  digitalWrite(pin_term2, HIGH);
  digitalWrite(pin_term3, HIGH);
  analogWrite(pin_term4, 255);
  //startupzero();
}

void startupzero() { // Arxikopoiw to mhxanimasto (0,0,0)
  int total_steps = 0;
  int total_steps2 = 0;
  Serial.println(" ΜΠΙΚΑ ΣΤΗΝΝ ΑΡΧΙΚΟΠΟΙΗΣΗ ");

```

```
while(digitalRead(pin_term1)) {  
  
  go(1,1,1,0);  
  
    total_steps2 = total_steps2 + 1;  
  
  }  
  
  Serial.println("Eginearxikopoihshston x aksona");  
  Serial.println(total_steps2);  
  delay(200);  
  while(digitalRead(pin_term2)) {  
  go(1,1,2,0);  
  total_steps = total_steps + 1;  
  }  
  Serial.println("Eginearxikopoihshston y aksona");  
  Serial.println(total_steps);  
  delay(200);  
  while(analogRead(pin_term4) == 1023) {  
  go(0,10,3,0);  
  }  
  Serial.println("Eginearxikopoihshston z aksona");  
  curr_x = 0;  
  curr_y = 0;  
  
  }  
  
void loop() {  
  
  intval = 0;          /* generic value read from serial */  
  
  boolean flag,flag2,flag3 = true;  
  int step, direction, axis;  
  
  /* The following instruction constantly checks if  
  anything  
  is available on the serial port. Nothing gets executed in  
  the loop if nothing is available to be read, but as soon  
  as anything becomes available, then the part coded after  
  the if statement (that is the real stuff) gets executed */  
  
  if (Serial.available() > 0) {  
  val = Serial.parseInt();  
  Serial.flush();
```

```
if (val == 97) {
  delay(1000);
  go(0,692,1,0);
}

if (val == 98) {
  delay(1000);
  moveCNC( 0, 0, 10, 10);
}
Serial.println(val);
startupzero();

}
if (val == 00){

moveCNC(curr_x, curr_y,new_x, new_y);
  //Serial.println("ok");
curr_x = new_x;
curr_y = new_y;
}
if (val==01){

drill();
}
if (val==02){
while(!Serial.available());
while(Serial.available() >0 ) {

delay_x = Serial.parseInt();

  //Serial.println("delay_x");
}

}
if (val==03){
while(!Serial.available());
while(Serial.available() >0 ) {

delay_y = Serial.parseInt();
  //Serial.println("delay_y");
}
}
if (val == 04){

while(!Serial.available());

while(Serial.available() > 0 ) {
```

```
delay_z = Serial.parseInt();

    }
}
if (val==05){

while(!Serial.available());
while(Serial.available() >0 ) {

new_x = Serial.parseInt();
Serial.flush();
    //Serial.println(new_x);
    }

    }
if (val==06){
while(!Serial.available());
while(Serial.available() >0 ) {

new_y = Serial.parseInt();
Serial.flush();

    }
    }
}

int go(int direction, int step, int axiss, int delay_speed)
{
int pin_step, pin_axis, pin_enable;

Serial.print(direction);
Serial.println(axiss);

if (axiss == 1){    // for x axis
pin_axis = 8;
pin_step = 9;
pin_enable = 10;
    //Serial.println(" THA KOUNHSW TON X AKSONA");
}
if (axiss == 2){ // for y axis
pin_axis = 6;
pin_step = 7;
pin_enable = 11;
    }
}
```

```
if (axiss == 3){ // for z axis
pin_axis = 4;
pin_step = 5;
pin_enable = 12;

}
digitalWrite(pin_enable, LOW);

if (direction == 0){
digitalWrite(pin_axis, LOW) ;
}else if (direction == 1){
digitalWrite(pin_axis, HIGH) ;
}

digitalWrite(pin_enable, LOW);

for (int x = 0; x < step; x++)
{
digitalWrite(pin_step, HIGH);
delayMicroseconds(900);
digitalWrite(pin_step, LOW);
delayMicroseconds(900);

if (delay_speed == 0) {

}else{
delay(delay_speed);
}
}
digitalWrite(pin_enable, HIGH);

}

void drill(){
while(digitalRead(pin_term3)){
go(1,10,3,delay_z);
}
delay(1000);

while(analogRead(pin_term4) == 1023){
go(0,10,3,0);
}
}

voidmoveCNC(intcurr_x, intcurr_y,intnew_x, intnew_y)
{
```

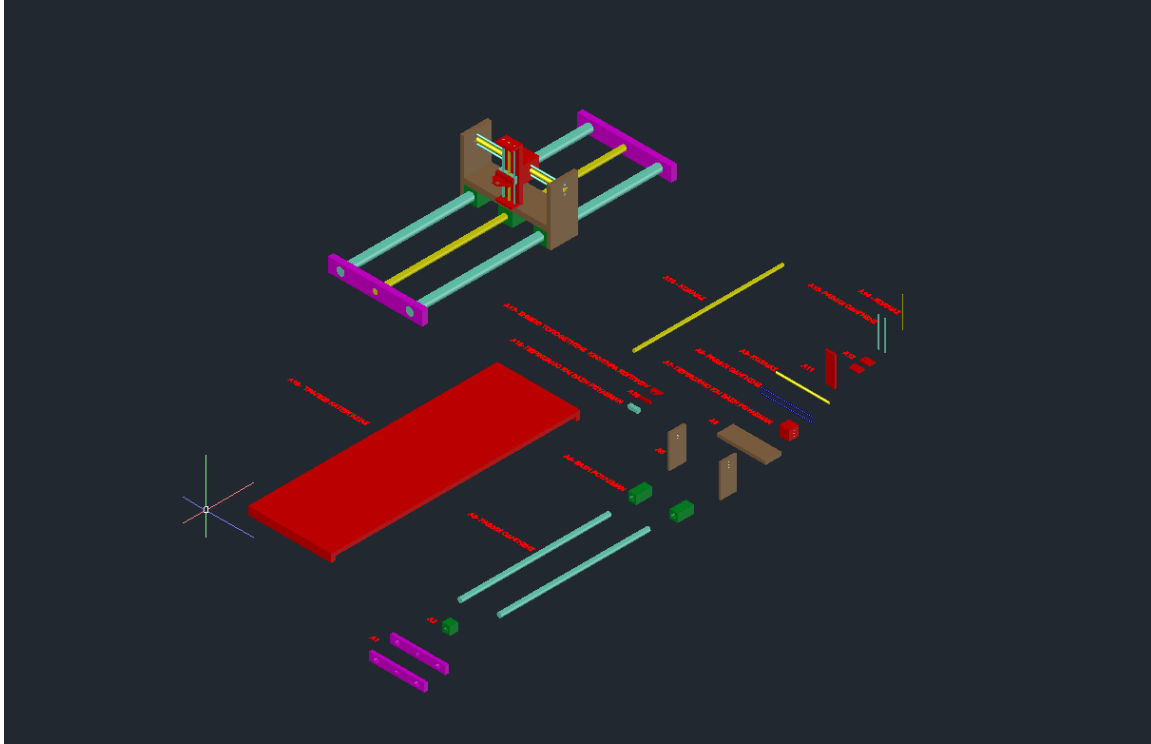
```
int step_1 = abs(curr_x - new_x) * 64.8;//63.7;
int step_2 = abs(curr_y - new_y) * 68; ////72.55;");
Serial.print(curr_x);
Serial.print(" , ");
Serial.println(curr_y);
");
Serial.print(new_x);
Serial.print(" , ");
Serial.println(new_y);
Serial.print(" , " );Serial.println(step_2);
Serial.print(delay_x);
Serial.print(" , ;Serial.println(delay_y);

if (curr_x<new_x)
{
for (int x = 0; x< step_1; x++)
{
go(0,1,1,delay_x);
}
}
else if (curr_x>new_x)
{
for (int x = 0; x< step_1; x++)
{
go(1,1,1,delay_x);
}
}

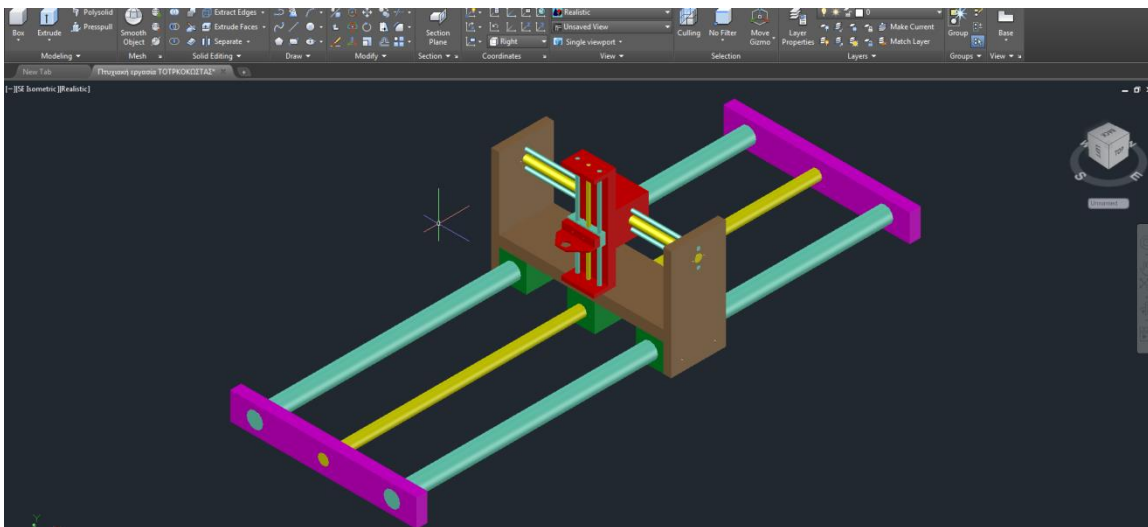
if (curr_y<new_y)
{
for (int x = 0; x< step_2; x++)
{
go(0,1,2,delay_y);
}
}

else if (curr_y>new_y)
{
for (int x = 0; x< step_2; x++)
{
go(1,1,2,delay_y);
}
}
}
```

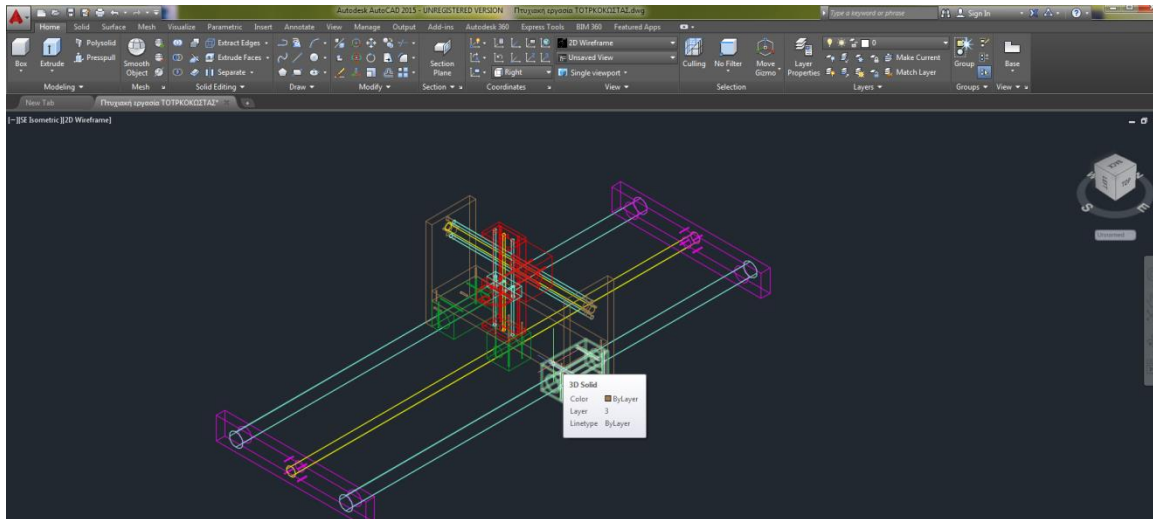
Ακολουθεί φωτογραφικό υλικό από μηχανολογικό σχέδιο στο πρόγραμμα Autocad και της συναρμολόγησης της κατασκευής :



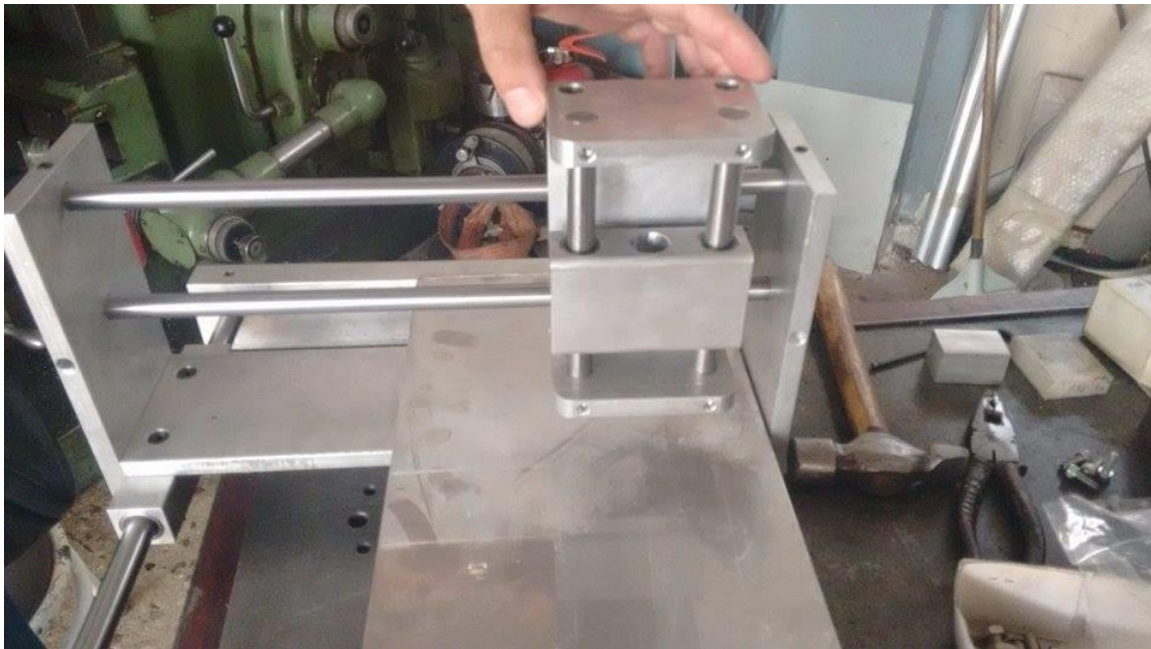
Εικόνα 1 Μηχανολογικό σχέδιο της κατασκευής και όλων των εξαρτημάτων που χρησιμοποιήθηκαν σε 3Dδιάσταση



Εικόνα 2 Μηχανολογικό σχέδιο κατασκευής σε 3Dδιάσταση



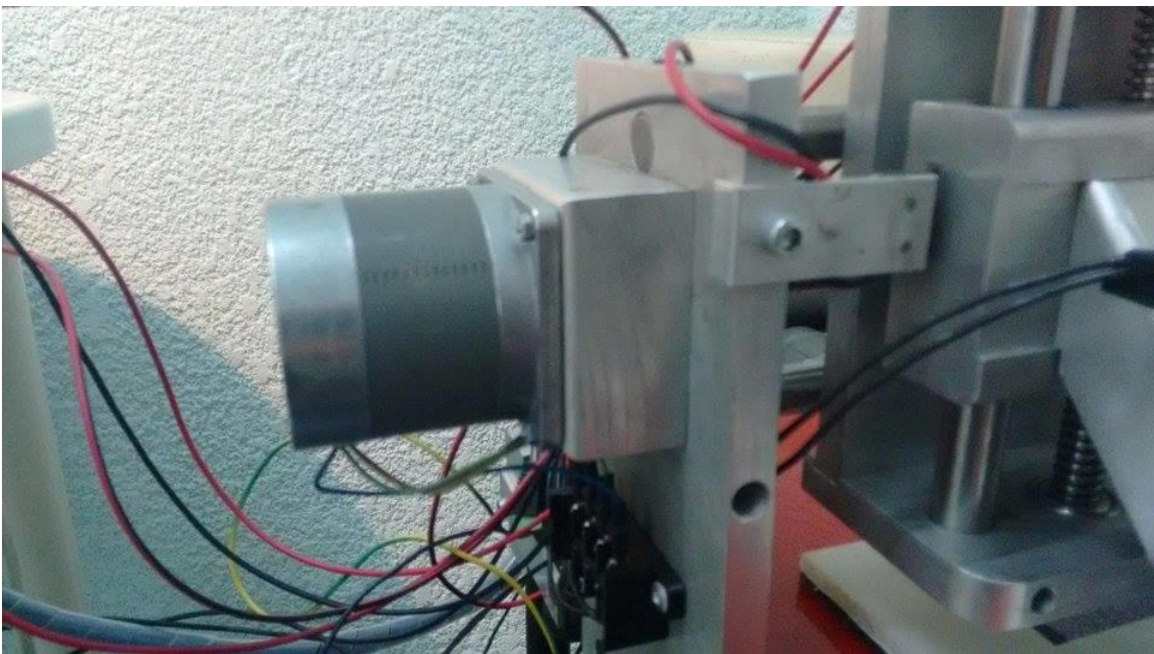
Εικόνα 3 Μηχανολογικό σχέδιο κατασκευής σε 2Dδιάσταση



Εικόνα 3 Στάδιο συναρμολόγησης



Εικόνα 4 Τοποθέτηση αξόνων στον Υ άξονα



Εικόνα 5 Τοποθέτηση τερματικού στον Υ άξονα



Εικόνα 6 Συναρμολόγηση X και Y άξονα