

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
«Σχεδιασμός και Υλοποίηση CNC »



Των φοιτητών:
Δελκούση Γεωργίου
Αριθμός Μητρώου 517027

Κωνσταντινούδη Ανδρέα
Αριθμός Μητρώου 517070

Επιβλέπων:
Ονοματεπώνυμο Δημήτριος
Παπακώστας

Βαθμίδα Καθηγητής

Ημερομηνία: 10/6/24

Τίτλος Π.Ε.: Σχεδιασμός και Υλοποίηση CNC

Κωδικός Π.Ε. 24175

Ονοματεπώνυμο φοιτητή/τών: Δελκούσης Γεώργιος και Κωνσταντινίδης Ανδρέας

Ονοματεπώνυμο εισηγητή Δημήτριος Παπακώστας

Ημερομηνία ανάληψης Π.Ε. 28-3-24

Ημερομηνία περάτωσης Π.Ε. 6-9-24

Βεβαιώνουμε πως είμαστε οι συγγραφείς αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχαμε για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχουμε καταγράψει τις όποιες πηγές από τις οποίες κάναμε χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνουμε ότι αυτή η εργασία προετοιμάστηκε από εμάς προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία των φοιτητών Δελκούση Γεώργιου και Κωνσταντινίδη Ανδρέα που την εκπόνησαν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, οι συγγραφείς/δημιουργοί εκχωρούν στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας των συγγραφέων/δημιουργών, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιοδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση των συγγραφέων/δημιουργών.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων των συγγραφέων, εκ μέρους του Τμήματος.

*«Αφιερώνουμε την παρούσα
Πτυχιακή Εργασία
στην οικογένειά μας και στους φίλους μας.»*

Πρόλογος

Η κατασκευή ενός CNC για πλακέτες προσφέρει μια εξαιρετική ευκαιρία για την αξιοποίηση και την περαιτέρω ανάπτυξη των γνώσεων και δεξιοτήτων μας στην κατασκευή ηλεκτρονικών κυκλωμάτων. Η δημιουργία και ανάπτυξη ενός τέτοιου μηχανήματος, το οποίο είναι ικανό να παράγει πανομοιότυπες πλακέτες κυκλωμάτων με υψηλή ακρίβεια, αποτελεί έναν ιδιαίτερα χρήσιμο εργαλείο για τη συνεχή πρόοδο και εξέλιξη στον τομέα των ηλεκτρονικών κατασκευών. Αυτό το έργο δεν προσφέρει μόνο πρακτική εμπειρία στην κατασκευή και σχεδίαση PCB, αλλά και τη δυνατότητα να εξετάσουμε σε βάθος τη διαδικασία παραγωγής, επιτρέποντάς μας να κατανοήσουμε καλύτερα τις παραμέτρους που επηρεάζουν την ποιότητα και την ακρίβεια των ηλεκτρονικών κυκλωμάτων. Έτσι, με μια πιο οικονομική λύση, αποκτούμε πρόσβαση σε ένα CNC μηχάνημα που θα μας βοηθήσει στις σπουδές μας και θα δοκιμάσει τις γνώσεις που αποκτήσαμε κατά τη διάρκεια των σπουδών μας στα ηλεκτρικά κυκλώματα. Αυτό είναι ιδιαίτερα σημαντικό για όσους επιθυμούν να ακολουθήσουν καριέρα στην ηλεκτρονική μηχανική ή τη ρομποτική, καθώς προσφέρει εμπειρία που δεν μπορεί να αποκτηθεί μόνο με θεωρητική εκπαίδευση. Αξίζει να σημειωθεί ότι τα έτοιμα CNC μηχανήματα συχνά περιλαμβάνουν πολλές λειτουργίες και δυνατότητες που ενδέχεται να είναι σύνθετες και να μην εφάπτονται στους σκοπούς της παρούσας εργασίας. Επιπλέον, η συγκεκριμένη κατασκευή προσφέρει τη δυνατότητα συνεχούς βελτίωσης, επιτρέποντάς μας να την προσαρμόζουμε και να την εξελίσσουμε ανάλογα με τις ανάγκες μας.

Περίληψη

Τα ρομποτικά συστήματα και οι τεχνολογίες αυτοματισμού έχουν επιφέρει επανάσταση σε πολλούς βιομηχανικούς τομείς, συμπεριλαμβανομένων της κατασκευής και της μηχανικής κατεργασίας. Ειδικά στον τομέα της μηχανικής κατεργασίας, οι μηχανές κοπής, όπως οι τόρνοι και οι φρέζες, διαδραματίζουν κρίσιμο ρόλο σε εργασίες υψηλής ακρίβειας, όπως η περιστροφή κυλινδρικών τεμαχίων για την κατασκευή συμμετρικών σχημάτων και η χάραξη λείων επιφανειών. Ωστόσο, οι παραδοσιακές μηχανές τόρνου χαρακτηρίζονται συχνά από υψηλό κόστος, σημαντικές απαιτήσεις χώρου και πολύπλοκες διαδικασίες συντήρησης. Αυτοί οι παράγοντες έχουν οδηγήσει στην αναζήτηση εναλλακτικών λύσεων που είναι πιο οικονομικά αποδοτικές, ευκολότερα προσβάσιμες και προσαρμόσιμες για εφαρμογές κατεργασίας μικρής κλίμακας. Τα τελευταία χρόνια, οι εξελίξεις στις παραγωγικές διαδικασίες και στην τεχνολογία μικροελεγκτών έχουν δημιουργήσει νέες προοπτικές για την ανάπτυξη προσιτών και ευέλικτων μηχανών, οι οποίες ανταποκρίνονται στις σύγχρονες ανάγκες της βιομηχανίας.

Η παρούσα πτυχιακή εργασία επικεντρώνεται στον σχεδιασμό, την κατασκευή και την αξιολόγηση ενός μικρού συστήματος CNC (Computer Numerical Control). Τα CNC συστήματα είναι υπολογιστικά ελεγχόμενες μηχανές που χρησιμοποιούνται ευρέως στη βιομηχανία για την κατεργασία υλικών, όπως μέταλλα, ξύλο και πλαστικό. Το συγκεκριμένο CNC σύστημα έχει σχεδιαστεί ειδικά για την κατεργασία πλακετών PCB (Printed Circuit Board).

Η έρευνα ξεκίνησε με μια ενδελεχή ανασκόπηση των υφιστάμενων CNC μηχανών, με στόχο την επιλογή των καταλληλότερων υλικών για την ανάπτυξη του συστήματος. Η κατασκευή και συναρμολόγηση του συστήματος πραγματοποιήθηκαν με ιδιαίτερη προσοχή, διασφαλίζοντας την ορθή ευθυγράμμιση και την ακρίβεια των κινήσεων του CNC. Οι δοκιμές που ακολούθησαν έδειξαν ότι το σύστημα είναι ικανό να κατεργάζεται πλακέτες PCB με υψηλή αποτελεσματικότητα και ακρίβεια.

Συνοψίζοντας, η εργασία αυτή οδήγησε στην ανάπτυξη ενός μικρού CNC συστήματος, το οποίο, με χαμηλό κόστος, προσφέρει τη δυνατότητα περαιτέρω κατασκευών στο μέλλον, τόσο από τους ίδιους τους ερευνητές όσο και από τους συμφοιτητές τους.

Design and Implementation of CNC

Abstract

Robotic systems and automation technologies have revolutionized many industrial sectors, including manufacturing and machining. Specifically, in the field of machining, cutting machines such as lathes and mills play a critical role in high-precision tasks, such as rotating cylindrical parts to create symmetrical shapes and engraving smooth surfaces. However, traditional lathe machines are often characterized by high cost, significant space requirements, and complex maintenance procedures. These factors have led to the search for alternative solutions that are more cost-effective, easily accessible, and adaptable for small-scale machining applications. In recent years, advancements in manufacturing processes and microcontroller technology have created new opportunities for the development of affordable and flexible machines that meet modern industry needs.

This thesis focuses on the design, construction, and evaluation of a small CNC (Computer Numerical Control) system. CNC systems are computer-controlled machines widely used in the industry for machining materials such as metals, wood, and plastic. This CNC system has been specifically designed for machining PCB (Printed Circuit Board) boards.

The research began with a thorough review of existing CNC machines, with the goal of selecting the most suitable materials for the system's development. The construction and assembly of the system were carried out with great care, ensuring proper alignment and accuracy of the CNC's movements. Subsequent tests showed that the system is capable of machining PCB boards with high efficiency and precision.

In summary, this project led to the development of a small CNC system, which, at a low cost, offers the potential for further constructions in the future, both by the researchers themselves and by their fellow students.

Ευχαριστίες

Η παρούσα πτυχιακή εργασία εκπονήθηκε από τους Γεώργιο Δελκούση και Ανδρέα Κωνσταντινιάδη, προπτυχιακούς φοιτητές του Τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου Ελλάδος, κατά το ακαδημαϊκό έτος 2023-2024, υπό την επίβλεψη του καθηγητή κ. Βάσσιου Βασιλείου.

Θα θέλαμε να εκφράσουμε την ειλικρινή μας ευγνωμοσύνη και εκτίμηση προς τον καθηγητή μας για την ευκαιρία που μας προσέφερε, με την ανάθεση του θέματος της πτυχιακής εργασίας, για τις πολύτιμες γνώσεις που μοιράστηκε μαζί μας και για την ανεκτίμητη βοήθεια που μας προσέφερε καθ' όλη τη διάρκεια της εκπόνησης της εργασίας μας. Η αφοσίωση, το ενδιαφέρον και ο χρόνος που διέθεσε για να μας καθοδηγήσει ήταν καθοριστικά για την επιτυχή ολοκλήρωση του έργου μας. Επίσης, εκφράζουμε τις ευχαριστίες μας στους καθηγητές του τμήματος, των οποίων η διδασκαλία και το περιεχόμενο των μαθημάτων τους μας παρείχαν τις απαραίτητες γνώσεις και έμπνευση, συμβάλλοντας με τον δικό τους τρόπο στην επιλογή και ολοκλήρωση της παρούσας εργασίας.

Τέλος, θα θέλαμε να ευχαριστήσουμε τις οικογένειές μας για την συνεχή προσφορά και υποστήριξη κατά την διάρκεια των φοιτητικών μας χρόνων, χρόνων που υπήρξαν δυσκολίες, απογοητεύσεις αλλά ταυτόχρονα και επιτυχίες, χαρές και χαμόγελα.

Δελκούσης Γεώργιος,
Κωνσταντινιάδης Ανδρέας,
Θεσσαλονίκη, Αύγουστος 2024

ΠΕΡΙΕΧΟΜΕΝΑ

Πρόλογος.....	5
Περίληψη.....	6
Design and Implementation of CNC	7
Abstract	7
Ευχαριστίες.....	8
Κατάλογος Σχημάτων.....	11
Συντομογραφίες.....	12
Κεφάλαιο 1ο: Εισαγωγή στις Μηχανές CNC.....	13
1.1 Εισαγωγή.....	13
1.2 Ορισμός του CNC	13
1.3 Ιστορική αναδρομή	14
1.4 Σύστημα CNC	15
1.4.1 CNC Φρέζα.....	15
1.4.2 Τι μπορεί να παράγει μία CNC φρέζα	16
1.4.3 Διαφορά μεταξύ CNC φρέζας και ενός CNC router.....	17
Επίλογος.....	18
Κεφάλαιο 2ο: Μικροελεγκτές – Θεωρητικό υπόβαθρο	19
2.1 Τι είναι ο μικροελεγκτής.....	19
2.1.1 Arduino.....	19
2.1.2 Από τι αποτελείται το Arduino PCB	20
2.1.3 Arduino Shields	22
2.1.4 Γιατί χρειαζόμαστε Shields.....	22
2.1.5 Τύποι Arduino Shields.....	22
2.1.6 Κώδικας του Arduino	24
2.2 Ο κώδικας G (G-Code)	26
2.2.1 Που χρησιμοποιείται	26
2.2.2 Γιατί χρησιμοποιείται	26
2.2.3 Πώς λειτουργεί	27
2.2.5 Διάγραμμα Ροής	28
2.3 Κινητήρες.....	29
2.3.1 Βηματικοί κινητήρες	29
2.3.2 DC Ηλεκτροκινητήρας	30
2.4 Έλεγχος Στροφών.....	31
2.5 Σημάδια αναφοράς.....	32
2.6 Τερματικοί διακόπτες.....	32
2.7 Σύστημα ψύξης και λίπανσης.	33
2.8 Τράπεζα επεξεργασίας τεμαχίων.....	33
2.9 Διακόπτης έκτακτης ανάγκης και ασφάλεια	34
Επίλογος.....	34
Κεφάλαιο 3ο: Κατασκευή CNC Φρέζας – Ανάπτυξη υλικού.....	35
3.1 Εισαγωγή.....	35

3.2 Προμήθεια υλικών.....	35
Πίνακας Κόστους Κατασκευής.....	35
3.3 Δημιουργία υλικών.....	37
3.3.1 Δημιουργία υλικών σε σχεδιαστικό πρόγραμμα.....	37
3.3.2 Σχεδιαστικό περιβάλλον στον 3D Printer.....	38
3.3.3 Λογισμικό χρήσης CNC (UGS).....	40
3.4 Υλικά κατασκευής.....	40
3.4.1 Arduino UNO Microcontroller.....	40
3.4.2 Arduino CNC Shield.....	41
3.4.1 Βηματικός κινητήρας.....	42
3.4.2 Κινητήρας κατεργασίας.....	43
3.4.3. Τερματικοί διακόπτες.....	43
3.4.4 Ρουλεμάν κλειστού τύπου και γραμμικά.....	44
3.4.5. Άξονες κίνησης.....	44
3.4.6. Προφίλ αλουμινίου.....	45
3.4.7 Συζευκτήρες αξόνων.....	45
3.5 Συναρμολόγηση.....	46
Επίλογος.....	47
Κεφάλαιο 4ο: Συμπεράσματα.....	48
4.1 Εισαγωγή.....	48
4.1.1 Πλεονεκτήματα.....	48
4.1.2. Μειονεκτήματα.....	48
4.2. Αποτελέσματα.....	49
4.3 Αστοχίες.....	49
4.4 Βελτιώσεις της κατασκευής.....	50
Επίλογος.....	51
Κεφάλαιο 5ο: Βιβλιογραφία.....	52
Παράρτημα.....	55

Κατάλογος Σχημάτων

Εικόνα 1: CNC Φρέζα.....	16
Εικόνα 2: CNC Router	17
Εικόνα 3: Το Arduino Uno PCB.....	20
Εικόνα 4: Διάγραμμα ροής κώδικα Arduino.	24
Εικόνα 5: Διάγραμμα ροής λειτουργίας G-Code.....	28
Εικόνα 6: Κύκλωμα οδήγησης μονοπολικού (αριστερά) και διπολικού (δεξιά)[fuselab].....	30
Εικόνα 7: Σημάδι αναφοράς(marker).....	32
Εικόνα 8: Σχεδίαση βάσεων στήριξης στο tinkercad	37
Εικόνα 9: Σχεδίαση βάσεων ατέρμωνων αξόνων στο tinkercad.....	37
Εικόνα 10: Σχεδίαση τραπεζών αξόνων στο tinkercad	38
Εικόνα 11: Σχεδίαση αντιολισθητικής βάσης της κατασκευής στο tinkercad.....	38
Εικόνα 12: Εκτυπωτής 3D.....	39
Εικόνα 13: Σχεδιαστικό λογισμικό 3D εκτυπωτή	39
Εικόνα 14: Περιβάλλον ελέγχου - λειτουργίας του CNC.....	40
Εικόνα 15: CNC shield.....	42
Εικόνα 16:Βηματικός κινητήρας.....	42
Εικόνα 17: DC κινητήρας.....	43
Εικόνα 18: Τερματικός διακόπτης (Endstop)	43
Εικόνα 19: γραμμικά ρουλεμάν.....	44
Εικόνα 20: κλειστού τύπου ρουλεμάν	44
Εικόνα 21: άξονας ακριβείας.....	44
Εικόνα 22: ατέρμονος άξονας	45
Εικόνα 23: Προφίλ αλουμίνιου.	45
Εικόνα 24: Σταθερός συζευκτήρας άξονα	45
Εικόνα 25: Ευέλικτος συζευκτήρας άξονα	46
Εικόνα 26: Συναρμολογημένο πρωτότυπο	47

Συντομογραφίες

CNC	Computer Numerical Control
NC	Numerical Control
PLC	Programmable Logic Controller
CIM	Computer Integrated Manufacturing
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
PCB	Printed Circuit Board
IDE	Integrated Development Environment
USB	Universal Serial Bus
LED	Light Emitting Diode
GPS	Global Positioning System
PWM	Pulse Width Modulation
LCD	Liquid Crystal Display
DC	Direct Current,
AC	Alternating Current
3D	Three Dimensions
OSB	Oriented Strand Board
XYZ	Καρτεσιανές συντεταγμένες στο επίπεδο
STL	Standard Template Library
UGS	Universal G-code Sender
RPM	Revolutions Per Minute

Κεφάλαιο 1ο: Εισαγωγή στις Μηχανές CNC

1.1 Εισαγωγή

Οι ραγδαίες εξελίξεις στη βιομηχανία και την τεχνολογία της μηχανικής τα τελευταία χρόνια έχουν μεταμορφώσει τον τρόπο με τον οποίο πραγματοποιείται η παραγωγή, επηρεάζοντας όχι μόνο την οικονομική ανάπτυξη, αλλά και τον τρόπο με τον οποίο οι σύγχρονες κοινωνίες αντιμετωπίζουν την καινοτομία και την τεχνολογική πρόοδο. Η ανάγκη για διαρκώς αυξανόμενη παραγωγικότητα, σε συνδυασμό με τις απαιτήσεις για υψηλότερη ποιότητα και ταχύτητα στις διαδικασίες, έχει καταστήσει αναγκαία την εισαγωγή προηγμένων τεχνολογιών και καινοτόμων μεθόδων ελέγχου στις ηλεκτρονικές μηχανές.

Αυτό το πλαίσιο ταχείας τεχνολογικής εξέλιξης καθιστά απαραίτητη την ανάπτυξη ολοένα και πιο εξελιγμένων και εξειδικευμένων συστημάτων μηχανικής, με ιδιαίτερη έμφαση στη βελτίωση της ακρίβειας, της αξιοπιστίας και της απόδοσης των μηχανημάτων. Στο επίκεντρο αυτών των εξελίξεων βρίσκονται οι εργαλειομηχανές CNC και η τεχνολογία NC, οι οποίες αποτελούν τον ακρογωνιαίο λίθο της σύγχρονης βιομηχανικής παραγωγής.

Το παρόν κεφάλαιο προσφέρει μια εις βάθος ανάλυση των θεμελιωδών αρχών λειτουργίας των εργαλειομηχανών CNC, ενώ παράλληλα παρέχει μια συνολική επισκόπηση της τεχνολογίας NC. Εξετάζει την ταξινόμηση και την αρχιτεκτονική των συστημάτων NC, καθώς και τις τεχνικές προγραμματισμού τους για διαφορετικές εφαρμογές και συστήματα συντεταγμένων. Αυτή η γνώση είναι απαραίτητη για την κατανόηση της λειτουργίας και της συνεισφοράς αυτών των τεχνολογιών στη σύγχρονη βιομηχανική παραγωγή, καθώς και για τη διαμόρφωση των μελλοντικών τάσεων στην τεχνολογία μηχανικής.

Στα κεφάλαια που ακολουθούν θα εξετάσουμε τι είναι τα CNC, το θεωρητικό υπόβαθρο σχετικά με την ιστορία, το λογισμικό και τον ηλεκτρονικό τομέα τους. Ακόμα, θα δούμε μια πλήρη αναφορά για την κατασκευή της εργασίας μας αλλά και των υλικών που χρησιμοποιήσαμε, είτε αυτά τα αγοράσαμε είτε τα σχεδιάσαμε. Τέλος, θα αναφερθούμε στα συμπεράσματα στα οποία καταλήξαμε και τις μελλοντικές βελτιώσεις που προτείνουμε.

1.2 Ορισμός του CNC

Πρόκειται για ένα πλήρως ή μερικώς αυτοματοποιημένο μηχάνημα, στο οποίο οι παραγγελίες αποστέλλονται σε μορφή κώδικα και αποθηκεύονται σε υλικό μέσο (όπως δισκέτα, USB κ.λπ.). Η εργαλειομηχανή αυτή είναι εξοπλισμένη με σύστημα αριθμητικού ελέγχου (CNC - Computer Numerical Control), το οποίο έχει τη δυνατότητα να υπολογίζει τις συντεταγμένες των σημείων που ορίζουν μια τροχιά (παρεμβολή). Χάρη στην τεχνολογία CNC, κατέστη δυνατή η αυτόματη παραγωγή εξαρτημάτων με τα υψηλότερα πρότυπα ακρίβειας και πολυπλοκότητας. Η διαδικασία αυτή εξασφαλίζει αποτελεσματική λειτουργία και ακριβή, συνεπή αναπαραγωγή αποτελεσμάτων, καθιστώντας την αναπόσπαστο μέρος της σύγχρονης μηχανικής κατεργασίας. Ο πρωταρχικός ρόλος μιας μηχανής CNC (Computer Numerical Control) είναι να δημιουργεί κινήσεις με ακρίβεια. Η μηχανή λαμβάνει δεδομένα σχετικά με τη θέση, την ταχύτητα και την επιτάχυνση και τα επεξεργάζεται για να παράγει ψηφιακές οδηγίες εξόδου. Χάρη στην υψηλή υπολογιστική της ισχύ και την ευελιξία στη χρήση, η μηχανή CNC υπερέρχει σε σχέση με ένα προγραμματιζόμενο PLC, προσφέροντας πιο προηγμένες δυνατότητες και μεγαλύτερη ακρίβεια στον έλεγχο κινήσεων[1].

Ας ρίξουμε τώρα μια ματιά στην ιστορία της τεχνολογίας CNC, τι επηρέασε τη δημιουργία της και σε ποιους τομείς χρησιμοποιείται σήμερα.

1.3 Ιστορική αναδρομή

Το 1949-1950, ο John Parsons ξεκίνησε την ανάπτυξη της τεχνολογίας NC (Numerical Control) στο Ινστιτούτο Τεχνολογίας της Μασαχουσέτης (MIT) στο Cambridge, των ΗΠΑ. Η έννοια του αριθμητικού ελέγχου (NC) βασίζεται στον έλεγχο των εργαλειομηχανών μέσω μιας συσκευής κωδικοποίησης δεδομένων, χωρίς τη χρήση λογισμικού, αποτελώντας μια καθαρά υλική προσέγγιση. Η εξέλιξη αυτή προκλήθηκε από μια παραγγελία της Πολεμικής Αεροπορίας των ΗΠΑ, η οποία απαιτούσε τα κρίσιμα μέρη μεγάλων αεροσκαφών να μην κατασκευάζονται πλέον ως συγκολλημένοι σύνδεσμοι, αλλά από συμπαγές μέταλλο. Καθώς η δημιουργία των απαιτούμενων προτύπων και μοντέλων με τις συμβατικές τεχνολογίες ήταν ιδιαίτερα χρονοβόρα και κοστοβόρα, αποφασίστηκε η ανάπτυξη ενός συστήματος ελέγχου. Αυτό έγινε εφικτό καθώς τα περιγράμματα των μεγάλων αυτών τεμαχίων μπορούσαν να περιγραφούν με μαθηματικές συναρτήσεις. Ο Parsons κατάφερε έτσι να δημιουργήσει το πρώτο μηχάνημα ελεγχόμενο από NC, το οποίο ονομάστηκε «Cincinnati Hydrotel»[2]

Ο πρώτος έλεγχος NC με μνήμη προγράμματος έγινε διαθέσιμος στις αρχές της δεκαετίας του 1970, η μετάβαση στην τεχνολογία CNC ολοκληρώθηκε μόλις το 1978, με την ανάπτυξη της πρώτης μηχανής CNC. Μέχρι τα μέσα της δεκαετίας του 1990, τα προγράμματα CNC γράφονταν χειροκίνητα, μια διαδικασία που απαιτούσε εξαιρετική συγκέντρωση από τον προγραμματιστή, καθώς ακόμη και τα παραμικρά λάθη μπορούσαν να προκαλέσουν ζημιά, μεγάλου κόστους, στα μηχανήματα. Στα τέλη της δεκαετίας του 1990 και στις αρχές της νέας χιλιετίας, η δημιουργία προγραμμάτων CNC άρχισε να γίνεται απευθείας μέσω συστημάτων CAD/CAM, μια πρακτική που παραμένει σταθερή και αξιοποιείται μέχρι και σήμερα. Αυτή τη στιγμή, βρισκόμαστε σε φάση ανάπτυξης ενός νέου συστήματος, το λεγόμενο CIM (Computer Integrated Manufacturing), με το οποίο ο προγραμματισμός στο μέλλον θα πραγματοποιείται πλήρως αυτόνομα, εξαλείφοντας την ανάγκη για ανθρώπινη παρέμβαση [1].

Η ανάπτυξη της τεχνολογίας CNC έθεσε νέες βάσεις στον κλάδο των κατασκευών, καθώς μετασημάτισε ριζικά τις διαδικασίες παραγωγής στη βιομηχανία μηχανουργικής κατεργασίας. Συγκεκριμένα, επέτρεψε την πλήρη αυτοματοποίηση της διαδικασίας κατασκευής, από τα αρχικά στάδια έως την τελική επεξεργασία του υλικού, καλύπτοντας κάθε φάση, από τη σύσφιξη του τεμαχίου εργασίας και το φρεζάρισμα, μέχρι την αλλαγή εργαλείων και τον ποιοτικό έλεγχο. Χάρη στην ταχύτερη και εξαιρετικά ακριβή κίνηση των αξόνων και των εργαλείων, κατέστη δυνατός ο εξορθολογισμός τόσο της σειριακής όσο και της μεμονωμένης παραγωγής, με αποτέλεσμα τη βελτίωση της αποτελεσματικότητας και της οικονομικής αποδοτικότητας. Σήμερα, η τεχνολογία CNC χρησιμοποιείται κυρίως στην επεξεργασία μετάλλων, αλλά εφαρμόζεται ευρέως και σε άλλες διαδικασίες κατεργασίας, όπως στην επεξεργασία πλαστικών ή ξύλινων υλικών [3].

Η μετάβαση από το δημοφιλές PLC (Programmable Logic Controller) σε έναν ελεγκτή που βασίζεται σε υπολογιστή αποτέλεσε μια σημαντική εξέλιξη στη βιομηχανία, εισάγοντας μια σειρά από πλεονεκτήματα που έχουν βελτιώσει σημαντικά τις διαδικασίες παραγωγής και ελέγχου. Ενώ τα PLC παραμένουν δημοφιλή για τον έλεγχο βιομηχανικών αυτοματισμών λόγω της αξιοπιστίας και της ανθεκτικότητάς τους, οι ελεγκτές που βασίζονται σε υπολογιστή έχουν αναδειχθεί ως η επόμενη γενιά στον βιομηχανικό έλεγχο [4]. Σύμφωνα με τον Ι. Μπερέτα, τα PLC αποτέλεσαν γρήγορα αναπόσπαστο κομμάτι της βιομηχανικής παραγωγής και αξιοποιήθηκαν σε πλήθος εφαρμογών, λόγω των πολλών πλεονεκτημάτων και του μικρού ποσοστού κινδύνου που προσφέρουν. Συγκεκριμένα, τα PLC χρησιμοποιούν μικροεπεξεργαστές με την μορφή ολοκληρωμένου κυκλώματος, που με τον κατάλληλο προγραμματισμό, προσφέρουν στους χειριστές μια ποικιλία επιλογών συμπεριφοράς και λειτουργιών. Με τον τρόπο αυτό ο προγραμματισμός μπορεί να γίνει με βάση τις απαιτήσεις

του προβλήματος, ιδιότητα στην οποία οφείλει και την ονομασία “Programmable” [5]. Στο σημείο αυτό, αξίζει να αναφέρουμε ότι η ευκολία που διαθέτουν τα PLC στον προγραμματισμό τους, τα καθιστά ιδανική επιλογή για την αυτοματοποίηση των CNC σε βασικά επίπεδα.

1.4 Σύστημα CNC

Το σύστημα ελέγχου του κέντρου κατεργασίας CNC έχει την δυνατότητα να διαχειρίζεται την εργαλειομηχανή, επιτρέποντας τον αυτόματο τερματισμό, την αλλαγή κατεύθυνσης και την ταχύτητα σύμφωνα με τις οδηγίες του ψηφιακού προγράμματος. Μπορεί να ρυθμίζει αυτόματα την ταχύτητα τροφοδοσίας, την κατεύθυνση και τη διαδρομή της επεξεργασίας [6]. Επιπλέον, το σύστημα έχει την ικανότητα να επιλέγει το κατάλληλο εργαλείο κοπής και να προσαρμόζει την ποσότητα κοπής αλλά και να ρυθμίζει τη διαδρομή λειτουργίας βάσει του μεγέθους του εργαλείου, εκτελώντας παράλληλα όλες τις απαραίτητες βοηθητικές ενέργειες που απαιτούνται κατά τη διάρκεια της επεξεργασίας.

Στο σημείο αυτό θα ήταν σημαντικό να αναφερθούμε στα μέρη από τα οποία αποτελείται ένας τόννος CNC. Ειδικότερα, αποτελείται από την κεφαλή, τον πυργίσκο, το σύστημα μετάδοσης τροφοδοσίας, το υδραυλικό σύστημα, τα συστήματα ψύξης και λίπανσης, καθώς και άλλα κομμάτια τα οποία θα αναλύσουμε στην συνέχεια. Αξίζει να σημειώσουμε ότι το σύστημα τροφοδοσίας τόννου CNC και το σύστημα οριζόντιας τροφοδοσίας τόννου διαφέρουν σημαντικά στη δομή τους, όπως απεικονίζεται σε ένα τυπικό διάγραμμα σύνθεσης της μηχανικής δομής ενός τόννου CNC. Η κίνηση του οριζόντιου άξονα στον τόννο μεταδίδεται στο στήριγμα εργαλείου μέσω του πλαισίου αλλαγής ταχυτήτων, του κιβωτίου τροφοδοσίας και του συρόμενου κιβωτίου, επιτρέποντας τη διαμήκη και εγκάρσια κίνηση τροφοδοσίας. Στον τόννο CNC, η τροφοδοσία ελέγχεται από έναν σερβοκινητήρα, ο οποίος μέσω της σφαιρικής βίδας κινεί την πλάκα ολίσθησης και τον πυργίσκο, επιτρέποντας την κίνηση στην κατεύθυνση του άξονα Z (διαμήκης) και την εγκάρσια κίνηση. Επιπλέον, η λεπτή κατεργασία στον τόννο CNC περιλαμβάνει τη δυνατότητα δημιουργίας διαφόρων τύπων σπειρωμάτων. Η συντονισμένη κίνηση μεταξύ της περιστροφής του άξονα και της κίνησης του στηρίγματος εργαλείου ελέγχεται πλήρως από το σύστημα CNC.

1.4.1 CNC Φρέζα

Όπως αναφέραμε προηγουμένως, το CNC έχει ενσωματωθεί σε μια ποικιλία νέων τεχνολογιών και μηχανημάτων, λόγω της ποικιλίας των πλεονεκτημάτων που προσφέρει. Ο πιο κοινός τύπος μηχανήματος που χρησιμοποιείται σε αυτόν τον τομέα είναι γνωστός ως «Φρέζα CNC».

Η CNC φρέζα αποτελεί έναν εξειδικευμένο τύπο μηχανής CNC, σχεδιασμένο για τη διαδικασία φρεζαρίσματος, η οποία είναι παρόμοια με τη διάτρηση ή την κοπή. Στο φρεζάρισμα, χρησιμοποιείται ένα κυλινδρικό εργαλείο κοπής, το οποίο μπορεί να περιστρέφεται σε διάφορες κατευθύνσεις. Σε αντίθεση με την παραδοσιακή διάτρηση, μια φρέζα έχει τη δυνατότητα να κινείται κατά μήκος πολλών αξόνων. Ο κόφτης της CNC φρέζας μπορεί να δημιουργεί μια ευρεία γκάμα σχημάτων, υποδοχών, οπών και άλλων απαιτούμενων αποτυπωμάτων. Επιπλέον, με μια μηχανή φρεζαρίσματος CNC, το τεμάχιο εργασίας μπορεί να μετακινηθεί σε συγκεκριμένες κατευθύνσεις πάνω από το εργαλείο φρεζαρίσματος,

παρέχοντας μεγαλύτερη ευελιξία. Αντίθετα, μια μηχανή διάτρησης επιτρέπει μόνο μονοαξονική κίνηση, περιορίζοντας έτσι τη συνολική παραγωγική της ικανότητα..

Οι φρέζες CNC ταξινομούνται συχνά ανάλογα με τον αριθμό των αξόνων στους οποίους μπορούν να εργαστούν. Κάθε άξονας επισημαίνεται με ένα συγκεκριμένο γράμμα. Για παράδειγμα, οι άξονες X και Y αντιπροσωπεύουν την οριζόντια κίνηση του τεμαχίου εργασίας της φρέζας, ο άξονας Z υποδηλώνει την κάθετη κίνηση, ενώ ο άξονας W αντιπροσωπεύει τη διαγώνια κίνηση στο κατακόρυφο επίπεδο.

Οι περισσότερες μηχανές φρεζαρίσματος CNC προσφέρουν δυνατότητες κίνησης σε 3 έως 5 άξονες. Για την εύρυθμη λειτουργία τους προτείνεται ο προγραμματισμός των σύγχρονων φρεζών με τεχνολογία CAM (Computer-Aided Manufacturing). Σε αντίθεση, με την χρήση παραδοσιακών χειροκίνητων εργαλείων, τα οποία δεν μπορούν να παράγουν εξειδικευμένα σχήματα, οι μηχανές CNC, είναι η ιδανική λύση για τις κατεργασίες αυτές, καθώς παρέχουν αυτή τη δυνατότητα. Επιπλέον, οι περισσότεροι CNC δρομολογητές είναι εξοπλισμένοι με ειδικές συσκευές ψύξης, οι οποίες διοχετεύουν υγρό στο εργαλείο κοπής κατά τη διάρκεια της παραγωγικής διαδικασίας, βελτιώνοντας την απόδοση και την ακρίβεια της κατεργασίας.



Εικόνα 1: CNC Φρέζα.[7].

1.4.2 Τι μπορεί να παράγει μία CNC φρέζα

Οι φρέζες CNC (Computer Numerical Control) έχουν αναδειχθεί ως εξαιρετικά αποδοτικές λύσεις για τη δημιουργία πρωτοτύπων και την παραγωγή σύνθετων εξαρτημάτων, ειδικά σε μικρές παρτίδες. Η ακρίβεια, η ταχύτητα και η ποιότητα των προϊόντων που παράγονται με τη χρήση CNC μηχανών είναι σημαντικά πλεονεκτήματα σε σχέση με τις παραδοσιακές χειροκίνητες μεθόδους φρεζαρίσματος. Οι CNC φρέζες μπορούν να επεξεργαστούν μια ευρεία γκάμα υλικών με μεγάλη ακρίβεια, από μέταλλα και πλαστικά μέχρι ξύλο και πιο σύνθετα υλικά. Η

αυτοματοποίηση της διαδικασίας σημαίνει επίσης ότι οι επαναλαμβανόμενες εργασίες γίνονται με σταθερή ποιότητα, ελαχιστοποιώντας τα σφάλματα και τη σπατάλη υλικών. Παρότι υπάρχουν ακόμα μηχανές χειροκίνητου φρεζαρίσματος, η τεχνολογία CNC έχει καταστήσει αυτές τις μηχανές λιγότερο ανταγωνιστικές, ειδικά με την πτώση του κόστους κατεργασίας CNC. Η επιλογή του σωστού CNC δρομολογητή μπορεί να βελτιώσει σημαντικά τις διαδικασίες παραγωγής, μειώνοντας το χρόνο κατασκευής και αυξάνοντας την ακρίβεια και την ποιότητα του τελικού προϊόντος.

1.4.3 Διαφορά μεταξύ CNC φρέζας και ενός CNC router

Η διαφορά μεταξύ ενός CNC router και μιας φρέζας CNC έγκειται στη λειτουργικότητα και στην περιοχή εργασίας. Ένα CNC router είναι συνήθως σχεδιασμένο για την επεξεργασία μαλακών υλικών όπως πλαστικό, ξύλο και ελαφρά μέταλλα, όπως το αλουμίνιο. Μία CNC φρέζα, από την άλλη πλευρά, είναι κυρίως κατάλληλη για την κοπή όλων των τύπων υλικών όπως σκληρά πλαστικά, ξύλο και μέταλλα όπως τιτάνιο και ισχυρά κράματα μετάλλων. Μία CNC φρέζα είναι συνήθως πολύ ισχυρότερη από ένα CNC router επειδή χρησιμοποιείται για την κοπή βαρέων μετάλλων, ιδανική για βιομηχανικούς σκοπούς.



Εικόνα 2: CNC Router[8].

Όσον αφορά την ακρίβεια, οι CNC φρέζες υπερτερούν των CNC routers, καθώς είναι σχεδιασμένες να επιτυγχάνουν εξαιρετικά υψηλή ακρίβεια, ακόμη και στην επεξεργασία μετάλλων. Οι μηχανές φρεζαρίσματος CNC μπορούν να επιτύχουν ακρίβεια στο επίπεδο του χιλιοστού, γεγονός που τις καθιστά ιδανικές για βιομηχανίες όπου το περιθώριο λάθους είναι ελάχιστο. Αυτός είναι ο λόγος για τον οποίο χρησιμοποιούνται ευρέως σε εφαρμογές που απαιτούν απόλυτη ακρίβεια και ποιότητα,

διασφαλίζοντας την κατασκευή εξαρτημάτων υψηλών προδιαγραφών.

Από την άλλη πλευρά ένα CNC router συνήθως συνδέεται με την επεξεργασία ξύλου, εφόσον παρέχει υψηλότερες ταχύτητες σε σύγκριση με το φρεζάρισμα CNC. Η ταχύτητα ενός δρομολογητή CNC κυμαίνεται μεταξύ 13.000 και 18.000, ενώ μιας φρέζας μεταξύ 800 και 1000 στροφών ανά λεπτό.

Επίλογος

Στο κεφάλαιο αυτό αναλύσαμε εκτενώς το τι είναι ένα CNC και την ιστορία της τεχνολογίας αυτής. Επιπλέον, διαχωρίσαμε τις έννοιες CNC φρέζας και router καθώς και το τι είναι δυνατό να παράξει μια τέτοια μηχανή. Η παραπάνω ενότητα είναι απαραίτητη για την πλήρη κατανόηση του σκοπού των CNC ώστε να εμβαθύνουμε στις επόμενες ενότητες σε άλλα εξίσου σημαντικά ζητήματα σχετικά με το θεωρητικό, το λογισμικό, το ηλεκτρονικό και κατασκευαστικό κομμάτι της εργασίας.

Κεφάλαιο 2^ο: Μικροελεγκτές – Θεωρητικό υπόβαθρο

2.1 Τι είναι ο μικροελεγκτής

Στο σημείο αυτό είναι σημαντικό να μιλήσουμε για την λειτουργία και την δομή των μικροελεγκτών προκειμένου να κατανοήσουμε καλύτερα την πλακέτα Arduino, σημαντικό μέρος της παρούσας κατασκευής. Αρχικά, με τον όρο «μικροελεγκτής» αναφερόμαστε σε ένα ολοκληρωμένο κύκλωμα που συμπεριλαμβάνει τον επεξεργαστή (CPU), τις μνήμες (RAM, ROM) και διάφορες άλλες περιφερειακές μονάδες εισόδου και εξόδου, γεγονός που του προσδίδει την ονομασία «μικροϋπολογιστής» [9]. Οι μικροελεγκτές βρίσκουν εφαρμογή σε μια ευρεία γκάμα προϊόντων και έχουν ενσωματωθεί σε πολλές πτυχές της καθημερινής μας ζωής. Ενδεικτικά παραδείγματα όπου συναντάμε μικροελεγκτές είναι τα αυτοκίνητα, τα κινητά τηλέφωνα, οι αριθμομηχανές, καθώς και άλλες ηλεκτρονικές συσκευές. Η βασική τους λειτουργία έγκειται στο ότι χωρίς την ανάγκη μεγάλων υπολογιστικών συστημάτων, προσφέρουν μια οικονομική και αποτελεσματική λύση για την κατασκευή συστημάτων με δυνατότητα επεξεργασίας δεδομένων. Δύο από τους πιο γνωστούς μικροελεγκτές είναι το Arduino και το Raspberry Pi [10].

2.1.1 Arduino

Το Arduino είναι μια πλατφόρμα ανοιχτού κώδικα που αποτελείται τόσο από μια πλακέτα τυπωμένου κυκλώματος (Printed Circuit Board – PCB) όσο και από μια εφαρμογή λογισμικού γνωστή ως ολοκληρωμένο περιβάλλον ανάπτυξης (IDE). Το λογισμικό εκτελείται στον υπολογιστή και χρησιμοποιείται για την εγγραφή και τη μεταφόρτωση κώδικα στο PCB χρησιμοποιώντας ένα καλώδιο USB [11].

Το υλικό και το λογισμικό Arduino αποτελούν την ιδανική λύση για άτομα που ενδιαφέρονται να σχεδιάζουν και να κατασκευάζουν διαδραστικά έργα. Ειδικότερα, το Arduino απευθύνεται σε σχεδιαστές, καλλιτέχνες και ερασιτέχνες που επιθυμούν να πειραματιστούν με την τεχνολογία και να δημιουργήσουν πρωτότυπες εφαρμογές, όπως συσκευές που αλληλεπιδρούν με το περιβάλλον ή το χρήστη. Η ευκολία στη χρήση του και η μεγάλη κοινότητα υποστήριξης καθιστούν το Arduino ένα εξαιρετικό εργαλείο για την ανάπτυξη καινοτόμων έργων, εφόσον οι πλακέτες μπορούν να αλληλεπιδράσουν με κάμερες, διακόπτες, LED, κινητήρες, κουμπιά, ηχεία και πολλά άλλα.

Το Arduino PCB διαθέτει πολλές εφαρμογές όπως η ρομποτική και τα συστήματα ήχου. Χρησιμοποιείται για το σχεδιασμό πρωτοτύπων και την ανάπτυξη ελέγχου που βασίζεται σε κώδικα. Ο κώδικας Arduino είναι μια έκδοση της C++, μια εύχρηστη γλώσσα προγραμματισμού, καθιστώντας το Arduino PCB πιο προσιτό και πιο ελκυστικό για αρχάριους. Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη αυτόνομων διαδραστικών αντικειμένων ή μπορεί να συνδεθεί με έναν υπολογιστή, ένα δίκτυο ή ακόμα και το διαδίκτυο για να ανακτήσει και να στείλει δεδομένα από και προς το Arduino και στη συνέχεια να ενεργήσει σε αυτά τα δεδομένα [12].

Με άλλα λόγια, μπορεί να στείλει ένα σύνολο δεδομένων που λαμβάνονται από ορισμένους αισθητήρες σε έναν ιστότοπο, ο οποίος μπορεί στη συνέχεια να εμφανιστεί με τη μορφή γραφήματος. Όπως αναφέραμε παραπάνω, το Arduino μπορεί να συνδεθεί

με LED, οθόνες dot matrix, κουμπιά, διακόπτες, κινητήρες, αισθητήρες θερμοκρασίας, αισθητήρες πίεσης, αισθητήρες απόστασης, δέκτες GPS, μονάδες Ethernet ή σχεδόν οτιδήποτε εξάγει δεδομένα ή μπορεί να ελεγχθεί. Μια ματιά στο διαδίκτυο θα φέρει στην επιφάνεια πληθώρα έργων όπου ένα Arduino έχει χρησιμοποιηθεί για την ανάγνωση δεδομένων ή τον έλεγχο μιας εκπληκτικής ποικιλίας συσκευών.

2.1.2 Από τι αποτελείται το Arduino PCB

Υπάρχουν πολλοί τύποι πλακετών Arduino που προορίζονται για διαφορετικούς σκοπούς. Παρά το γεγονός ότι υπάρχουν ορισμένα διαφορετικά στοιχεία σε κάθε πλακέτα, αυτά τα χαρακτηριστικά είναι συνήθως συμβατά με όλους τους τύπους πλακέτας Arduino:



Εικόνα 3: Το Arduino Uno PCB. [12]

USB/Κυλινδρικό βύσμα τροφοδοσίας

Κάθε πλακέτα Arduino χρειάζεται πρόσβαση στην τροφοδοσία. Αυτό μπορεί να επιτευχθεί μέσω ενός καλωδίου USB που συνδέεται από τον υπολογιστή σας στην πλακέτα ή μέσω τροφοδοσίας από τον τοίχο. Στο παραπάνω σχήμα 1, η θύρα USB επισημαίνεται ως (1) αλλά και η υποδοχή βύσματος (για τροφοδοσία από τον τοίχο) επισημαίνεται ως (2). Μια πλακέτα Arduino μπορεί να λαμβάνει ισχύ μικρότερη από 20 βολτ. Οποιαδήποτε τροφοδοσία υψηλότερη από αυτή μπορεί να καταστρέψει την πλακέτα Arduino. Η συνιστώμενη ποσότητα ισχύος είναι μεταξύ 6 και 12 βολτ. Για τη φόρτωση κώδικα στην πλακέτα Arduino, η μέθοδος που χρησιμοποιείται είναι μέσω καλωδίου USB.

Pins

Τα pins επιτρέπουν τη σύνδεση των καλωδίων στην πλακέτα για τη δημιουργία κυκλωμάτων, που συνήθως χρησιμοποιούνται σε συνδυασμό με μια πλακέτα και σύρμα. Τα pins έχουν μαύρες πλαστικές «κεφαλίδες» για να μπορούν τα καλώδια να εισχωρήσουν απευθείας στην πλακέτα. Μια πλακέτα Arduino έχει μια σειρά από pins, που βρίσκονται γύρω από τις άκρες, τις οποίες μπορείτε να δείτε στο Σχήμα 1. Αυτά

είναι: [12]

GND (Γείωση) – Ένα Arduino έχει πολλαπλούς ακροδέκτες GND , οι οποίοι χρησιμοποιούνται για τη γείωση του κυκλώματος. Ένα κύκλωμα γειώνεται όταν σε ένα από τα αγωγία καλώδια που είναι συνδεδεμένα στο κύκλωμα, δίνεται σκόπιμα μια απευθείας διαδρομή προς τη γη. [13]

5V & 3,3V – Αυτές οι ακίδες είναι απλές. Το ένα τροφοδοτεί 3,3 V και το άλλο 5V. Τα περισσότερα εξαρτήματα σε ένα Arduino μπορούν να «τρέξουν» οποιαδήποτε ποσότητα ενέργειας. [13]

Αναλογικό – Μπορούμε να δούμε αυτές τις ακίδες στην Εικόνα 1, κάτω από την ενότητα «αναλογική είσοδος». Αυτές οι ακίδες μπορούν να διαβάσουν το σήμα από έναν αναλογικό αισθητήρα και να το μετατρέψουν σε αναγνώσιμη ψηφιακή εικόνα. [13]

Ψηφιακή – Στην αντίθετη πλευρά του πίνακα μπορούμε να δούμε τις ψηφιακές ακίδες (0-13). Αυτά μπορούν να χρησιμοποιηθούν για ψηφιακή είσοδο και έξοδο. [13]

PWM – Ορισμένες από τις ψηφιακές ακίδες έχουν ένα σύμβολο «~» δίπλα τους στο σχήμα 1 παραπάνω. Αυτές οι ακίδες μπορούν να χρησιμοποιηθούν για τακτική ψηφιακή χρήση, αλλά έχουν επίσης τη δύναμη να χρησιμοποιηθούν για διαμόρφωση πλάτους παλμού (PWM). Το PWM είναι μια μέθοδος για την επίτευξη αναλογικών αποτελεσμάτων με ψηφιακά μέσα. [13]

AREF – Αυτό αναφέρεται στην Αναλογική Αναφορά. Πολλές φορές αυτό δεν χρησιμοποιείται, αλλά μπορεί να οριστεί μια εξωτερική τάση αναφοράς ως τη μέγιστη ισχύ για τους ακροδέκτες αναλογικής εισόδου. [13]

Κουμπί Reset – Πατώντας το κουμπί reset συνδέουμε προσωρινά την ακίδα επαναφοράς στη γείωση και επανεκκινεί οποιονδήποτε κωδικό έχει φορτωθεί στο Arduino. [13]

Ένδειξη ισχύος LED – Δίπλα στο σημείο που λέει «ON» στην Εικόνα 1 υπάρχει μια λυχνία LED. Αυτό θα πρέπει να ανάβει όταν το Arduino είναι συνδεδεμένο σε μια πηγή ρεύματος και υποδεικνύει ότι η πλακέτα έχει ρεύμα. [13]

TX RX LED – TX (μετάδοση) και RX (λήψη) υποδεικνύουν τις ακίδες που επιτρέπουν τη σειριακή επικοινωνία. Αυτά τα σημάδια φαίνονται στο Σχήμα 1 δύο φορές, δίπλα στις ψηφιακές ακίδες 0 και 1 και στη συνέχεια επίσης από τις ενδείξεις LED TX και RX . Αυτές οι ενδείξεις LED δείχνουν πότε το Arduino εκπέμπει και λαμβάνει δεδομένα. [13]

Κεντρικό IC – Σύντομο για ολοκληρωμένο κύκλωμα και φαίνεται στο Σχήμα 1 . Αυτός είναι ουσιαστικά ο εγκέφαλος του Arduino PCB. Τα IC μπορεί να διαφέρουν από πλακέτα σε πλακέτα, αλλά είναι σημαντικό να γνωρίζει κανείς ποιο από αυτά χρησιμοποιείται, καθώς θα είναι μια βασική γνώση για τη φόρτωση ενός νέου προγράμματος στην πλακέτα με το λογισμικό Arduino. Ο τύπος του IC συνήθως υποδεικνύεται στην επιφάνειά του. [13]

Ρυθμιστής τάσης – Ο ρυθμιστής τάσης είναι σημαντικός για τη διατήρηση της σωστής ποσότητας ισχύος που κυκλοφορεί μέσω της πλακέτας. Ουσιαστικά απομακρύνει κάθε υπερβολική ισχύ για να μην βλάψει το Arduino. Ωστόσο, αυτό είναι εντός λογικής και δεν σημαίνει ότι μπορούμε να το συνδέσουμε σε οποιαδήποτε πηγή ρεύματος. Το μέγιστο θα πρέπει να είναι 20 βολτ. [13]

2.1.3 Arduino Shields

Όπως έχουμε δει, το Arduino είναι μια δημοφιλής ηλεκτρονική πλατφόρμα ανοιχτού κώδικα που έχει φέρει επανάσταση στον κόσμο των αυτοσχέδιων κατασκευών. Χάρη στους ισχυρούς μικροελεγκτές και το ευέλικτο περιβάλλον προγραμματισμού, το Arduino δίνει τη δυνατότητα σε ενθουσιώδεις δημιουργούς και επαγγελματίες να υλοποιήσουν τις δημιουργικές τους ιδέες. Ένα από τα βασικά χαρακτηριστικά που καθιστούν το Arduino ακόμα πιο ισχυρό είναι η διαθεσιμότητα των Shield. Οι Shield Arduino είναι πλακέτες επέκτασης που μπορούν να συνδεθούν στην πλακέτα Arduino, προσθέτοντας νέες λειτουργίες και δυνατότητες στις διάφορες εφαρμογές, επιτρέποντας στους χρήστες να προσαρμόζουν το σύστημα στις συγκεκριμένες ανάγκες του έργου τους.

Τα Arduino Shields είναι πλακέτες επέκτασης [13] που μπορούν εύκολα να συνδεθούν σε μια πλακέτα Arduino, παρέχοντας πρόσθετες λειτουργίες και δυνατότητες. Τα Shields έχουν σχεδιαστεί για να εφαρμόζουν τέλεια στην πλακέτα Arduino, καθιστώντας την μια απρόσκοπτη διαδικασία ενσωμάτωσης. Χρησιμοποιώντας τα Arduino Shields, μπορεί κανείς γρήγορα και αποτελεσματικά να προσθέτει χαρακτηριστικά όπως συνδεσιμότητα Wi-Fi, έλεγχο κινητήρα, οθόνες LCD και πολλά άλλα στις διάφορες κατασκευές.

2.1.4 Γιατί χρειαζόμαστε Shields

Τα Arduino Shields προσφέρουν πολλές δυνατότητες που τα καθιστούν βασικό συστατικό σε κάθε έργο Arduino. Παρακάτω παραθέτουμε ονομαστικά τα πλεονεκτήματα:

- **Εκτεταμένη λειτουργικότητα:** Τα shields μας επιτρέπουν να προσθέτουμε νέες δυνατότητες στις δημιουργίες μας, χωρίς την ανάγκη περίπλοκης καλωδίωσης ή πρόσθετων εξαρτημάτων. Παρέχουν έναν εύκολο και βολικό τρόπο για την βελτίωση της λειτουργικότητας της πλακέτας Arduino.
- **Εύκολη ενσωμάτωση :** Τα Arduino shields έχουν σχεδιαστεί ειδικά για να εφαρμόζουν σε πλακέτες Arduino, εξασφαλίζοντας τέλεια εφαρμογή και εύκολη ενσωμάτωση. Αυτό εξαλείφει την ανάγκη για πολύπλοκη καλωδίωση ή συγκόλληση, καθιστώντας το προσβάσιμο για αρχάριους και έμπειρους χρήστες.
- **Ευελιξία :** Με μια μεγάλη γκάμα διαθέσιμων shields, με ευρεία δυνατότητα επιλογής αυτών που ταιριάζουν καλύτερα στις απαιτήσεις του έργου. Είτε χρειαζόμαστε συνδεσιμότητα Wi-Fi, έλεγχο κινητήρα ή ενσωμάτωση αισθητήρα, υπάρχει ένα shield διαθέσιμο για να ικανοποιήσει κάθε ανάγκη.
- **Εξοικονόμηση χρόνου :** Η χρήση "ασπίδων" Arduino εξοικονομεί χρόνο και προσπάθεια για τον σχεδιασμό και την κατασκευή πολύπλοκων κυκλωμάτων. Αντί να ξεκινήσουμε από την αρχή, μπορούμε απλά να συνδέσουμε το shield και να αρχίσουμε να εργαζόμαστε αμέσως για την κατασκευή μας.

2.1.5 Τύποι Arduino Shields

Υπάρχουν διάφοροι τύποι "ασπίδων" Arduino που διατίθενται στην αγορά, καθένας

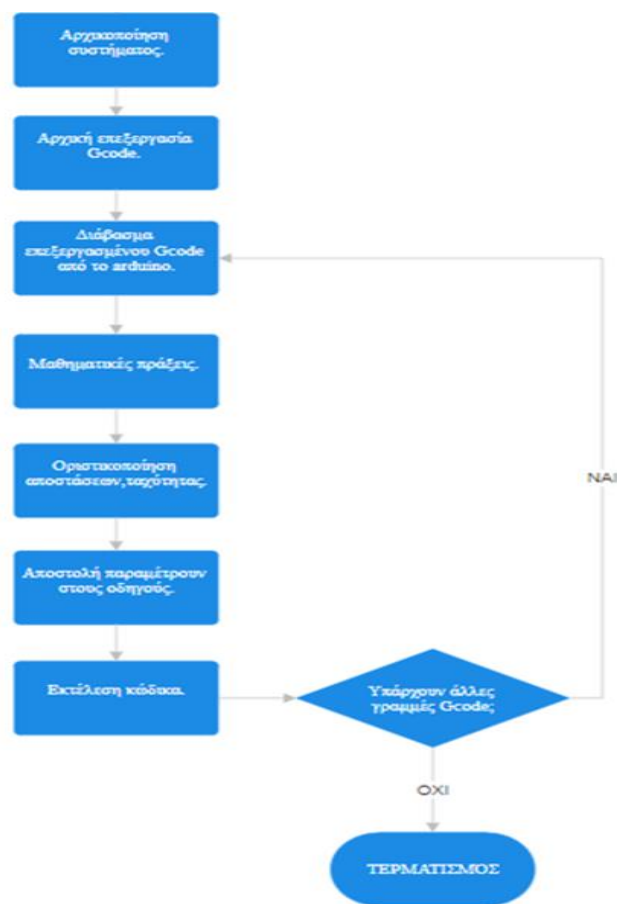
από τους οποίους καλύπτει συγκεκριμένες απαιτήσεις έργου. Ας εξερευνήσουμε μερικές από τις πιο δημοφιλείς "ασπίδες" Arduino και τις λειτουργίες τους.

- **Ethernet Shield:** Συνδέει την πλακέτα Arduino στο διαδίκτυο. Παρέχει τη δυνατότητα επικοινωνίας με διακομιστές Ιστού, αποστολής και λήψης δεδομένων και πρόσβασης σε διαδικτυακούς πόρους.
- **Proto Shield:** Έχει σχεδιαστεί για τη δημιουργία προσαρμοσμένων κυκλωμάτων και τη δημιουργία πρωτοτύπων. Παρέχει έναν κενό καμβά για τη συγκόλληση ηλεκτρονικών εξαρτημάτων απευθείας πάνω στο Shield, επιτρέποντας την σχεδίαση και την δοκιμή κυκλωμάτων. Το Shield Proto είναι ιδανικό για προχωρημένους χρήστες ή όσους θέλουν να δημιουργήσουν μοναδικά έργα με συγκεκριμένες απαιτήσεις κυκλώματος.
- **Shield ρελέ:** Χρησιμοποιείται για τον έλεγχο συσκευών που απαιτούν υψηλό ρεύμα και τάση, όπως κινητήρες, φώτα και συσκευές. Παρέχει πολλαπλά ρελέ που μπορούν να ελεγχθούν από την πλακέτα Arduino, επιτρέποντάς μας να αλλάζουμε φορτία υψηλής ισχύος με ευκολία. Το Shield ρελέ χρησιμοποιείται συνήθως σε έργα οικιακού αυτοματισμού, ρομποτικής και βιομηχανικών συστημάτων ελέγχου.
- **Motor Shield:** Έχει σχεδιαστεί ειδικά για τον έλεγχο των κινητήρων με την πλακέτα Arduino. Επιτρέπει τον έλεγχο την κατεύθυνση και την ταχύτητα των κινητήρων DC ή την οδήγηση βηματικών κινητήρων. Το Motor Shield είναι ιδανικό για έργα που περιλαμβάνουν ρομποτική, μηχανοκίνητα οχήματα ή οποιαδήποτε εφαρμογή που απαιτεί ακριβή έλεγχο κινητήρα.
- **LCD Shield:** Το LCD Shield έρχεται με ενσωματωμένη οθόνη LCD και πληκτρολόγιο, διευκολύνοντας έτσι τη δημιουργία έργων με διεπαφές χρήστη. Παρέχει έναν βολικό τρόπο εμφάνισης πληροφοριών, μενού και προτροπών στον χρήστη. Το Shield LCD χρησιμοποιείται συνήθως σε έργα όπως ψηφιακά θερμόμετρα, έξυπνοι ελεγκτές σπιτιού και καταγραφείς δεδομένων.
- **Bluetooth Shield:** Επιτρέπει την ασύρματη επικοινωνία μεταξύ της πλακέτας Arduino και άλλων συσκευών με δυνατότητα Bluetooth, όπως smartphone, tablet και υπολογιστές. Μας επιτρέπει να δημιουργούμε έργα που μπορούν να αλληλεπιδρούν με εφαρμογές για κινητά ή να ανταλλάσσουν δεδομένα ασύρματα. Το Bluetooth Shield χρησιμοποιείται συνήθως σε εφαρμογές οικιακού αυτοματισμού, τεχνολογίας φορητών συσκευών και τηλεχειριστηρίου.
- **Xbee Shield:** Επιτρέπει την ασύρματη επικοινωνία με την πλακέτα Arduino. Χρησιμοποιεί την τεχνολογία Zigbee, επιτρέποντάς μας να επικοινωνούμε ασύρματα με άλλες συσκευές ή να δημιουργούμε δίκτυα πλέγματος. Το Shield Xbee είναι ιδανική για έργα που απαιτούν τηλεχειρισμό, ασύρματα δίκτυα αισθητήρων ή επικοινωνία μεταξύ πολλών πλακών Arduino.
- **Touchpad Shield:** Προσφέρει κουμπιά και επιθέματα ευαίσθητα στην αφή που μπορούν να ενσωματωθούν στα έργα μας στο Arduino. Μας επιτρέπει να δημιουργούμε διαδραστικές διεπαφές που ανταποκρίνονται στην αφή, καθιστώντας το ιδανικό για έργα που απαιτούν είσοδο χρήστη ή χειριστήρια αφής. Το Shield επιφάνειας αφής χρησιμοποιείται συνήθως σε κονσόλες παιχνιδιών, διαδραστικές οθόνες και διεπαφές ανθρώπου-μηχανής.

- CNC Shield:** Αυτή η πλακέτα επέκτασης, μπορεί να χρησιμοποιηθεί για μηχανές χάραξης, εκτυπωτές 3D. Σχεδιάστηκε για να καλύψει τη ζήτηση για μια λύση ελεγκτή χαμηλού κόστους για μηχανές CNC DIY. Σχεδιάστηκε για να είναι 100% συμβατό με το Grbl, τον διεργασμένο G-Code ανοιχτού κώδικα, και να ταιριάζει στο δημοφιλές Arduino Uno [14]. Το CNC Shield μπορεί να χρησιμοποιηθεί για τον έλεγχο πολλών διαφορετικών τύπων μηχανών CNC, συμπεριλαμβανομένων μηχανών φρεζαρίσματος CNC, μηχανών χάραξης / κοπής λέιζερ, μηχανών σχεδίασης, εκτυπωτών 3D ή οποιουδήποτε έργου που χρειάζεται έλεγχο ακριβείας βηματικών κινητήρων.

2.1.6 Κώδικας του Arduino

Στην παρακάτω υποενότητα αυτή θα παραθέσουμε σε ένα διάγραμμα ροής τον κώδικα που τρέχει το Arduino της βιβλιοθήκης GRBL, για την ομαλή λειτουργία του G-code και του CNC συστήματος.



Εικόνα 4: Διάγραμμα ροής κώδικα Arduino.

Παρακάτω θα αναλύσουμε τον κώδικα του διαγράμματος ροής.

- Αρχικοποίηση συστήματος :** Αρχικά ρυθμίζουμε την ταχύτητα μετάδοσης σειριακού (baud rate) και διακοπών (interrupts), και φορτώνουμε τις ρυθμίσεις GRBL από την μνήμη EEPROM, ενεργοποιούμε τις διακοπές και κάνουμε εκκαθάριση των μεταβλητών του υποσυστήματος.

2. **Αρχική επεξεργασία gcode** : Αφαιρούνται τα κενά και τα σχόλια του Gcode και γράφονται όλες οι εντολές με κεφαλαία γράμματα.
3. **Ανάγνωση επεξεργασμένου Gcode** : Αφού πρώτα γίνει η αρχική επεξεργασία το Arduino διαβάζει το Gcode από την σειριακή είσοδο.
4. **Μαθηματικές πράξεις** : Διαβάζοντας των κώδικα το Arduino προχωράει σε μαθηματικές πράξεις που παράγονται από τις εντολές του Gcode.
5. **Οριστικοποίηση μαθηματικών πράξεων** : Το Arduino οριστικοποιεί τις μαθηματικές πράξεις που αφορούν την ταχύτητα των κινητήρων, τις συντεταγμένες και την αλληλουχία κινήσεων.
6. **Αποστολή παραμέτρων στους οδηγούς κινητήρων** : Έπειτα, αποστέλλονται στους οδηγούς τα παραπάνω δεδομένα με την μορφή παλμών ώστε να λειτουργήσουν σωστά οι οδηγοί
7. **Εκτέλεση κώδικα και λειτουργία CNC** : Σε αυτό το σημείο το μηχάνημα μας λειτουργεί και εκτελεί την εντολή Gcode που έχει έρθει από το λογισμικό χρήσης του μηχανήματος μας.
8. **Υπάρχουν άλλες γραμμές Gcode ;** : Το σύστημα μας εδώ καλείται να ελέγξει αν υπάρχει νέα γραμμή εντολής Gcode. Στην περίπτωση αυτή υπάρχουν δύο αποφάσεις.
 - I. Αν υπάρχει νέα γραμμή κώδικα, τότε το Arduino επιστρέφει στην τρίτη εντολή και κάνει από την αρχή την διαδικασία για όσο υπάρχουν νέες εντολές.
 - II. Αν δεν υπάρχει νέα εντολή, τότε ο κώδικας μας τερματίζει και το σύστημα μας απενεργοποιείται ή εισέρχεται σε κατάσταση αναμονής.

Ο κώδικας που χρησιμοποιήθηκε για το παραπάνω διάγραμμα παρατίθεται στο παράρτημα.

2.2 Ο κώδικας G (G-Code)

Το G-code (γεωμετρικός κώδικας) είναι η πιο διαδεδομένη γλώσσα προγραμματισμού για CNC (Computer Numerical Control) μηχανές, οι οποίες χρησιμοποιούνται για την αυτοματοποιημένη επεξεργασία υλικών. Αν και δεν είναι γλώσσα προγραμματισμού με την αυστηρή έννοια του όρου, αποτελεί ένα σύνολο εντολών υψηλού επιπέδου που καθοδηγούν τις μηχανές σχετικά με τις κινήσεις και τις λειτουργίες τους. Αυτές οι εντολές ελέγχουν τους κινητήρες των αξόνων, ρυθμίζοντας την ταχύτητά τους και καθορίζουν την θέση που θα κινηθούν.

Παράλληλα, ο G-code είναι επίσης η κυρίαρχη γλώσσα για μηχανές 3D εκτύπωσης και CNC συστήματα. Σε αυτές τις εφαρμογές, ο G-code καθοδηγεί την κίνηση των εξαρτημάτων με ακρίβεια, προκειμένου να δημιουργήσει τρισδιάστατα αντικείμενα με βάση τις καθορισμένες οδηγίες που του έδωσε ο προγραμματιστής για κίνηση, θέσεις, και τις ειδικές απαιτήσεις της κάθε μηχανής.

Παρακάτω παρουσιάζουμε και εξηγούμε μερικά παραδείγματα εντολών G-code, ώστε να τα εξηγήσουμε καλύτερα:

- G21: Ορίζει τη βάση μέτρησης σε μετρικό, mm [15].
- G0 X10: Το εργαλείο κινείται 10 mm κατά μήκος του άξονα X από την προηγούμενη θέση με τη μέγιστη ταχύτητα διέλευσης [15].
- G1 X10 F100: Το εργαλείο κινείται 10 mm κατά μήκος του άξονα X με καθορισμένη ταχύτητα 150 mm/s [15].

2.2.1 Που χρησιμοποιείται

Ο κώδικας G (G-Code) χρησιμοποιείται σε διάφορες εφαρμογές που αφορούν την αυτοματοποιημένη κατασκευή και τον έλεγχο μηχανών. Τον G-code τον συναντάμε κυρίως σε μηχανές CNC, διότι με τις εντολές είναι εύκολο να καταφέρει κανείς την καθοδήγηση των αξόνων τους. Επιπροσθέτως χρησιμοποιείται ευρέως σε 3D εκτυπωτές, για τον έλεγχο της εκτύπωσης αντικειμένων, στρώση προς στρώση, καθορίζοντας την κίνηση του εκτυπωτή και την εκτύπωση του υλικού. Ο G-code είναι επίσης χρήσιμος σε μηχανές κοπής με λέιζερ, πλασματικές κοπές, και άλλες αυτοματοποιημένες μηχανές. Τα μηχανήματα αυτά χρησιμοποιούν G-Code για να καθορίσουν τις διαδρομές κοπής με ακρίβεια σε διάφορα υλικά, όπως μέταλλα, πλαστικά και ξύλο. Τέλος συχνά τον συναντάει κανείς στην ρομποτική βιομηχανία, σε ρομποτικούς βραχίονες που χρησιμοποιούνται για τη συναρμολόγηση, συγκόλληση, ή βαφή προϊόντων και εξαρτημάτων.

2.2.2 Γιατί χρησιμοποιείται

Ο κώδικας G προορίζεται να λέει στις μηχανές τι να κάνουν ή πώς να κινηθούν. Κωδικοποιεί μια τρισδιάστατη διαδικασία συλλογισμού, καθοδηγεί τους κόπτες, τις κεφαλές εκτύπωσης κ.λπ. μέσα από μια διαδρομή που τα τοποθετεί για την προβλεπόμενη εργασία και εκτελεί την εντολή. Στην πορεία, ο G-code καθορίζει τις ταχύτητες του άξονα και τις ροές ψυκτικού υγρού και ελέγχει τυχόν απαραίτητες αλλαγές στο εργαλείο. Η γλώσσα επιτρέπει στους προγραμματιστές CNC (ή σε άλλους μηχανισμούς που βασίζονται σε υπολογιστή) να γράφουν ακολουθίες cut-and-move που δημιουργούν λεπτομερείς διαδικασίες που μπορούν να

ακολουθήσουν οι μηχανές τους χωρίς επίβλεψη.

Ο κύριος λόγος για τη χρήση του είναι η ανάγκη για ακρίβεια και επαναληψιμότητα στην παραγωγή των αγαθών, καθώς επιτρέπει στις μηχανές να εκτελούν πολύπλοκες κινήσεις με απόλυτη συνέπεια . Επιπλέον, η ευελιξία του G-Code επιτρέπει στους προγραμματιστές να εκτελέσουν διαφορετικές εργασίες σε μια μηχανή, έτσι επιτυγχάνεται η μείωση του χρόνου και του κόστους παραγωγής. Εξασφαλίζει επίσης την αυτοματοποίηση και ελαχιστοποιεί τα ανθρώπινα λάθη, διασφαλίζοντας υψηλή ποιότητα και αποδοτικότητα στη βιομηχανική παραγωγή. Για όλους αυτούς τους λόγους επιλέγεται η χρήση του συγκεκριμένου κώδικα .

2.2.3 Πώς λειτουργεί

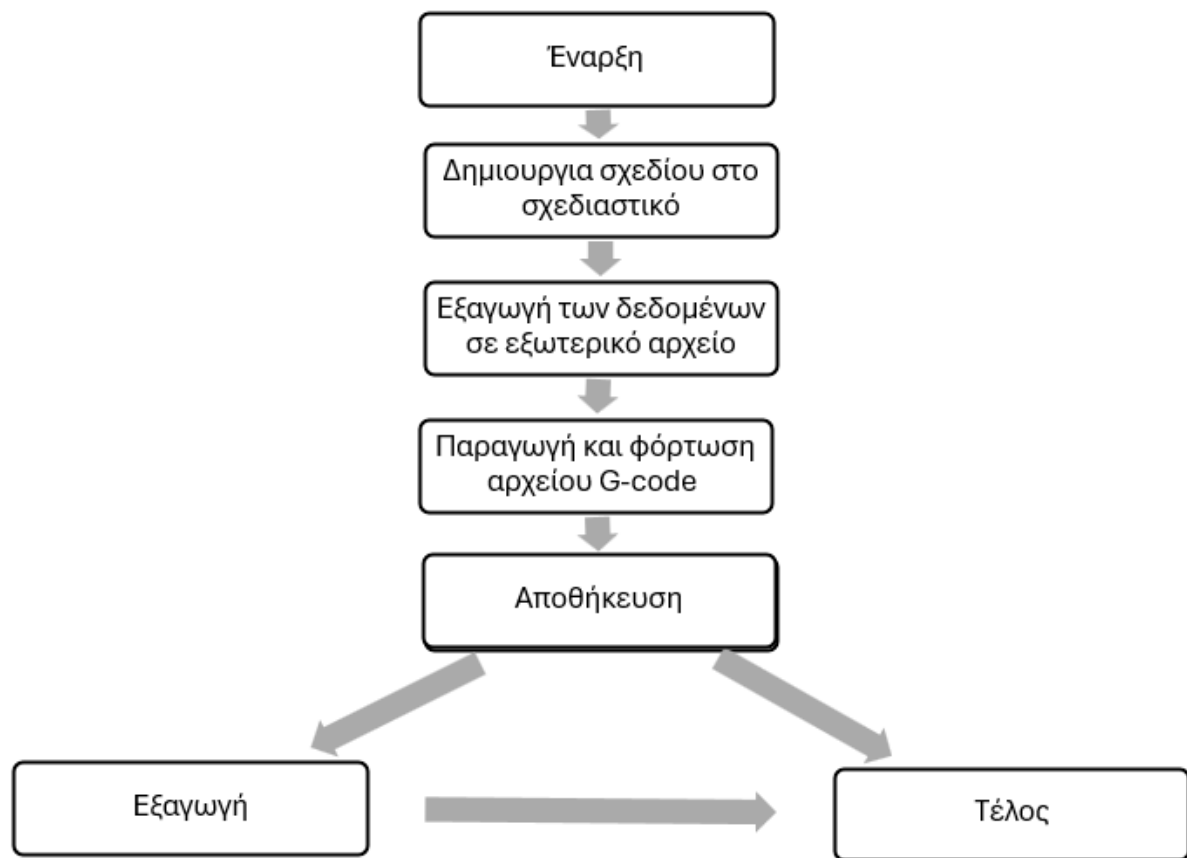
Ο G-code προγραμματίζεται γραμμή προς γραμμή, ώστε να μπορεί να διαβαστεί από το μηχανήμα μας (CNC ή οτιδήποτε άλλο). Πρακτικά δημιουργείται ένα ορθογώνιο σύστημα συντεταγμένων όπου μέσα σε αυτό περιορίζονται οι κινήσεις και οι εντολές που προγραμματίζει ο χειριστής . Οι εντολές μετατρέπονται σε σήματα/ παλμούς, ώστε να μπορούν να αναγνωριστούν από τους κινητήρες που υπάρχουν στο σύστημα. Οι παλμοί αυτοί είναι ηλεκτρικά σήματα που δημιουργούνται από διακόπτες, ON (5V ή 3,3V ανάλογα με τον κινητήρα) ή OFF (0V δηλαδή GND). Για παράδειγμα, έστω ότι έχουμε ένα βηματικό κινητήρα με 200 βήματα ανά πλήρη περιστροφή . Αν θέλουμε ο κινητήρας να περιστραφεί κατά 90 μοίρες και γνωρίζουμε ότι κάθε παλμός μετακινεί τον κινητήρα κατά 1,8 μοίρες, τότε θα χρειαστούμε 50 παλμούς για να επιτευχθεί αυτή η κίνηση. Αυτοί οι 50 παλμοί θα είναι μια ακολουθία από 50 ON και 50 OFF καταστάσεις που στέλνονται στον κινητήρα για να τον κάνουν να κινηθεί κατά τα 90 μοίρες [15].

Οι βασικές κατηγορίες είναι οι εξής :

- **Κινηματικές Εντολές:**
 - G00 (Ταχεία μετακίνηση) [16]:** Μετακίνηση του εργαλείου στη δεδομένη θέση όσο το δυνατόν γρηγορότερα χωρίς να λαμβάνεται υπόψη η κοπή.
 - G01 (Γραμμική κοπή) [16]:** Γραμμική μετακίνηση του εργαλείου με συγκεκριμένη ταχύτητα κοπής.
 - G02 (Κυκλική κοπή δεξιόστροφα) [16]:** Κυκλική κίνηση δεξιόστροφα γύρω από ένα συγκεκριμένο σημείο.
 - G03 (Κυκλική κοπή αριστερόστροφα):** Κυκλική κίνηση αριστερόστροφα.
- **Εντολές Διεύθυνσης [16]:**
 - G17:** Επιλογή του επιπέδου XY.
 - G18:** Επιλογή του επιπέδου ZX.
 - G19:** Επιλογή του επιπέδου YZ.
- **Εντολές Ρυθμίσεων [16]:**
 - G20:** Χρήση ιντσών ως μονάδα μέτρησης.
 - G21:** Χρήση χιλιοστών ως μονάδα μέτρησης.
 - G28:** Επιστροφή του εργαλείου στο σημείο αναφοράς
- **Εντολές Διάστασης[16] :**
 - G90:** Απόλυτες συντεταγμένες .
 - G91:** Σχετικές συντεταγμένες .
- **Εντολές Σπινάλ[16] :**
 - G32:** Ειδική κωδικοποίηση για σπειρώματα.
 - G76:** Αυτοματοποιημένη δημιουργία σπειρώματος σε έναν άξονα.

- **Εντολές Παύσης ή Αναμονής[16]:**
 - G04:** Αναμονή για συγκεκριμένο χρονικό διάστημα.
- **Εντολές Εργαλείων[16] :**
 - M06:** Αλλαγή εργαλείου.
 - M03:** Εκκίνηση άξονα με δεξιόστροφη κίνηση.
 - M05:** Σταμάτημα άξονα.
- **Εντολές Τερματισμού[16]:**
 - M02:** Τερματισμός προγράμματος.
 - M30:** Τερματισμός προγράμματος και επιστροφή στην αρχή του.

2.2.5 Διάγραμμα Ροής



Εικόνα 5: Διάγραμμα ροής λειτουργίας G-Code.

Το διάγραμμα ροής για τον κώδικα G (G-code) είναι ένας οπτικός τρόπος για να αναπαρασταθεί η διαδικασία που ακολουθεί το CNC μηχάνημα κατά την εκτέλεση του κώδικα.

1. **Έναρξη:** Αρχίζει η διαδικασία. Σε αυτό το στάδιο, εξετάζονται οι προδιαγραφές και οι απαιτήσεις του έργου.
2. **Δημιουργία σχεδίου στο σχεδιαστικό:** Δημιουργείται το σχέδιο σε κάποιο σχεδιαστικό πρόγραμμα του τύπου CAD/CAM, για παράδειγμα Autodesk, OpenSCAD [17] ή το TinkerCad, που επιλέξαμε να χρησιμοποιήσουμε κι εμείς.

3. **Εξαγωγή των δεδομένων σε εξωτερικό αρχείο:** Το σχέδιο εξάγεται σε κάποιο εξωτερικό αρχείο συνήθως με την μορφή STEP ή STL και εισάγεται σε CAM λογισμικό.
4. **Παραγωγή και φόρτωση αρχείου G-code:** Από το STEP/STL αρχείο δημιουργείται ο G-code μέσω των απαραίτητων βιβλιοθηκών στον μικροελεγκτή που μεταφράζει τα σημεία του αρχείου STL σε κίνηση των κινητήρων, συντεταγμένες στον χώρο προς κάθε κατεύθυνση[18]. Έτσι μετατρέπονται οι κινήσεις σε σήματα παλμών για τους κινητήρες των αξόνων. Οι παλμοί αυτοί είναι ηλεκτρικά σήματα που δημιουργούνται από διακόπτες, ON (5V ή 3,3V, ανάλογα με τον κινητήρα) ή OFF (0V δηλαδή GND), η συχνότητα που γίνεται αυτή η διαδικασία, καθορίζει με την ταχύτητα που θα κινηθεί
5. **Αποθήκευση:** Το αρχείο G-code αποθηκεύεται στον προεπιλεγμένο φάκελο.

Στη συνέχεια, μετά την "Αποθήκευση" υπάρχει διακλάδωση σε δύο επιλογές:

- a) **Εξαγωγή:** Εξάγεται το αρχείο και λειτουργεί το CNC μέχρι την περάτωση της διεργασίας.
- b) **Τέλος:** Η διαδικασία ολοκληρώνεται, χωρίς εξαγωγή και ο χρήστης μπορεί να θέσει σε λειτουργία το σύστημα (εφόσον έχει πλέον έτοιμο το αρχείο) όταν εκείνος θελήσει.

2.3 Κινητήρες

2.3.1 Βηματικοί κινητήρες

Οι κινητήρες βηματικής λειτουργίας χρησιμοποιούνται ευρέως σε εφαρμογές όπως εκτυπωτές 3D, μηχανές CNC και ρομποτική λόγω της ακρίβειας και της αξιοπιστίας τους. Παρακάτω παρουσιάζεται μια επισκόπηση του τρόπου λειτουργίας τους και των λόγων που οι βηματικοί κινητήρες είναι κατάλληλοι για αυτές τις εργασίες.

Ένας βηματικός κινητήρας είναι μια ηλεκτρομηχανική συσκευή που μετατρέπει ηλεκτρικούς παλμούς σε συγκεκριμένες μηχανικές κινήσεις. Σε αντίθεση με τους τυπικούς κινητήρες DC που περιστρέφονται συνεχώς όταν τροφοδοτούνται, οι βηματικοί κινητήρες περιστρέφονται σε διακριτά βήματα, επιτρέποντας ακριβή έλεγχο της γωνιακής θέσης, της ταχύτητας και της επιτάχυνσής τους.

Κύρια χαρακτηριστικά των NEMA 17 είναι τα παρακάτω:

Οι κινητήρες NEMA 17 συνήθως διαθέτουν 200 βήματα ανά περιστροφή, γεγονός που σημαίνει ότι περιστρέφονται κατά 1,8 μοίρες ανά βήμα, προσφέροντας υψηλή ακρίβεια και έλεγχο.

Αυτοί οι κινητήρες παρέχουν συνήθως ροπή ίση με 0.45 N/m [19], που είναι επαρκής για εργασίες που απαιτούν μέτρια δύναμη. Αυτή η ροπή τους καθιστά κατάλληλους για πολλές εφαρμογές, όπως σε εκτυπωτές 3D, CNC μηχανές και ρομποτικά συστήματα.

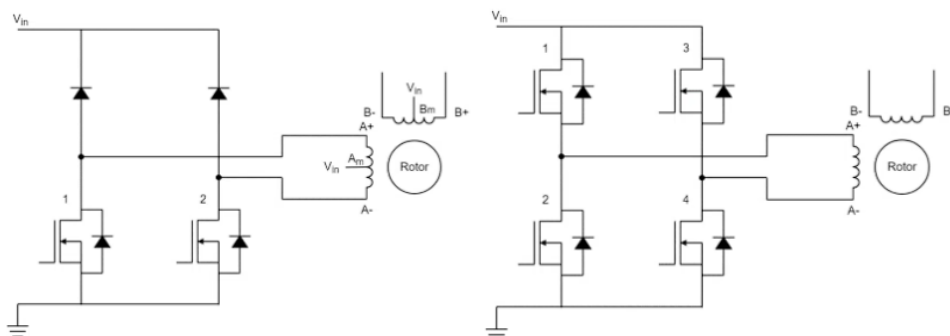
Η δομή των κινητήρων NEMA 17 περιλαμβάνει τον ρότορα, που είναι το περιστρεφόμενο μέρος του κινητήρα και συνήθως κατασκευάζεται από μόνιμο μαγνήτη, και τον στάτορα, το σταθερό μέρος που περιέχει τα πηνία. Η συνδυασμένη λειτουργία του ρότορα και του στάτορα επιτρέπει στον κινητήρα να παράγει κίνηση με ακρίβεια και σταθερότητα.

Επιπρόσθετα όσον αφορά την λειτουργία τους, όταν ρέει ρεύμα μέσω των περιελίξεων του στάτορα, παράγεται ένα μαγνητικό πεδίο με το οποίο ευθυγραμμίζεται ο ρότορας (ένας μαγνήτης ή μαγνητικό υλικό), έπειτα ο οδηγός του κινητήρα ενεργοποιεί τα πηνία με συγκεκριμένη σειρά, προκαλώντας τον ρότορα να κινηθεί σε μικρά, ακριβή βήματα. Κάθε παλμός που αποστέλλεται στον κινητήρα αντιστοιχεί σε ένα βήμα του ρότορα.

Άξιο αναφοράς είναι οι τύποι περιελίξεων που διαθέτουν αυτοί οι κινητήρες, καθώς χωρίζονται σε διπολικούς, μονοπολικούς και υβριδικούς. Στους διπολικούς βηματικούς κινητήρες, κάθε πηνίο έχει μόνο δύο διαθέσιμα καλώδια, γεγονός που απαιτεί τη χρήση γέφυρας τύπου H (H-

bridge) για τον έλεγχο της κατεύθυνσης του ρεύματος [20].

Όταν τα MOSFET 1 και 4 ενεργοποιούνται, το ρεύμα ρέει από το A+ στο A-, ενώ εάν τα MOSFET 2 και 3 είναι σε λειτουργία, το ρεύμα ρέει από το A- στο A+. Αυτή η διάταξη χρειάζεται ένα πιο περίπλοκο κύκλωμα οδήγησης, αλλά επιτρέπει την αλλαγή της κατεύθυνσης του μαγνητικού πεδίου, προσφέροντας μεγαλύτερη ροπή σε σύγκριση με τους μονοπολικούς κινητήρες. Οι διπολικοί κινητήρες είναι συχνά προτιμητέοι σε εφαρμογές όπου η αυξημένη απόδοση και η μεγαλύτερη ροπή είναι κρίσιμες, ενώ, λιγότερο συχνά χρησιμοποιούν μονοπολικές περιελίξεις που έχουν κεντρική λήψη, επιτρέποντας στο ρεύμα να ρέει προς μία κατεύθυνση για κάθε μισό του πηνίου. Αυτοί είναι πιο απλοί στην οδήγηση αλλά λιγότερο αποδοτικοί [21].



Εικόνα 6: 6 Κύκλωμα οδήγησης μονοπολικού (αριστερά) και διπολικού (δεξιά)[21].

Η μικροβηματική κίνηση αναφέρεται στην τεχνική ρύθμισης του ρεύματος που ρέει στις περιελίξεις του κινητήρα με έναν πιο ομαλό και συνεχόμενο τρόπο. Αυτό επιτρέπει στον κινητήρα να εκτελεί βήματα μικρότερα από το τυπικό του μέγεθος βήματος, βελτιώνοντας έτσι την ανάλυση της κίνησης και μειώνοντας τις δονήσεις. Η μέθοδος αυτή επιτυγχάνεται μέσω της προσεκτικής διαχείρισης της τάσης που εφαρμόζεται στα πηνία του στάτορα, δημιουργώντας ενδιάμεσες θέσεις ανάμεσα στα τυπικά βήματα. Ο έλεγχος των κινητήρων NEMA 17 βασίζεται σε μια συγκεκριμένη ακολουθία ενεργοποίησης των πηνίων, η οποία καθορίζει τόσο την κατεύθυνση όσο και την ταχύτητα περιστροφής του κινητήρα. Ο οδηγός του κινητήρα που χρησιμοποιούμε είναι υπεύθυνος για τη σωστή εφαρμογή αυτής της ακολουθίας, κάνοντας ομαλή και ακριβή την λειτουργία. Η κατεύθυνση της περιστροφής μπορεί να αλλάξει αντιστρέφοντας την ακολουθία των παλμών, ενώ η ταχύτητα του κινητήρα ρυθμίζεται μέσω της συχνότητας των παλμών που αποστέλλονται στον οδηγό.

2.3.2 DC Ηλεκτροκινητήρας

Ο κινητήρας που χρησιμοποιείται στο τρυπάνι είναι ένας DC κινητήρας που δεν διαθέτει βούρτσες για την αλλαγή ροής του ρεύματος (Brushless DC Motors)[22], εν αντιθέσει με τους παραδοσιακούς που έχουν (Brushed DC Motors). Ο λόγος που επιλέξαμε αυτόν τον κινητήρα είναι τα βασικά του χαρακτηριστικά. Αρχικά, θα μελετήσουμε τη μηχανική διάταξη του κινητήρα. Ένας BLDC στον ρότορά του περιέχει μόνιμους μαγνήτες, ενώ ο στάτορας τις περιελίξεις. Ακόμα, πρέπει να γνωρίζουμε ότι ο ρότορας αποκτά κίνηση από το περιστρεφόμενο μαγνητικό πεδίο που δημιουργείται στον στάτορα από τις αλλαγές της ροής του ρεύματος. Επιπλέον, όπως ήδη αναφέραμε, αυτοί οι ηλεκτροκινητήρες δεν έχουν βούρτσες στον συλλέκτη, που σημαίνει ότι δεν υπάρχει φθορά μεταξύ βουρτσών και συλλέκτη, γεγονός που αυξάνει την ανθεκτικότητα και τη διάρκεια ζωής, καθώς επίσης αυξάνεται η αποδοτικότητα και

μειώνεται αισθητά ο θόρυβος. Τέλος, οι BLDC κινητήρες προτιμώνται για την ανθεκτικότητα, την υψηλή απόδοση και τον ακριβή έλεγχο ταχύτητας, έχοντας ένα ευρύ σύνολο εφαρμογών, όπως τα ηλεκτρικά οχήματα, τα ηλεκτρικά εργαλεία και τις οικιακές συσκευές κτλ. Τα παραπάνω είναι οι λόγοι που στην εργασία μας επιλέξαμε να χρησιμοποιήσουμε αυτόν τον τύπο κινητήρα [23].

2.4 Έλεγχος Στροφών

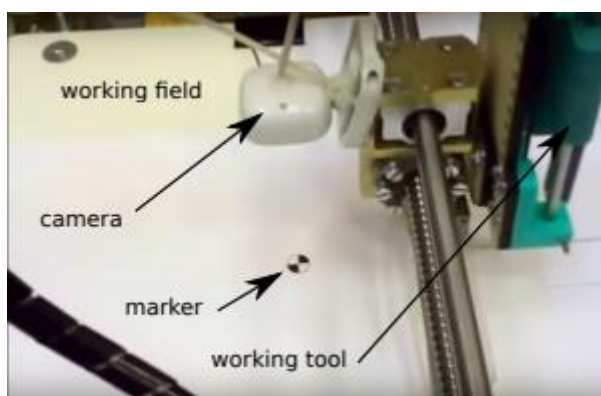
Τα περισσότερα CNC και τα μηχανήματα που περιέχουν στην κατασκευή τους κινητήρες, κάνουν χρήση κυκλωμάτων ελέγχου στροφών, διότι είναι απαραίτητα κυκλώματα που αφορούν την βέλτιστη ποιότητα, την αποτελεσματικότητα και την ζωτικότητα του εξοπλισμού και των εργαλείων κατεργασίας. Γενικότερα, στους λόγους που καθιστούν τον έλεγχο στροφών απαραίτητο εργαλείο περιλαμβάνονται η ακρίβεια στην κατεργασία, η προστασία εργαλείων και μηχανής, η βελτιστοποίηση παραγωγικότητας και η ευελιξία επεξεργασίας διάφορων υλικών. Πιο αναλυτικά, η ιδανική ταχύτητα περιστροφής του εργαλείου κοπής, γνωστού και ως ατράκτου, είναι απαραίτητη για την ορθή επεξεργασία των υλικών εξαιτίας των ιδιοτήτων τους που διαφέρουν. Επιπλέον, με αυτόν τον τρόπο προστατεύονται τα εργαλεία κοπής και η μηχανή από υπερβολική φθορά. Όλα τα παραπάνω, έχουν ως αποτέλεσμα την αβίαστη χρήση της μηχανής αυξάνοντας την παραγωγικότητα, μειώνοντας τον χρόνο κατεργασίας και δημιουργώντας ένα ασφαλές περιβάλλον τόσο για τους χρήστες όσο και για τις μηχανές. Ειδικότερα, τα συστήματα αυτά και πιο συγκεκριμένα ερασιτεχνικά όπως τα δικό μας κάνουν πολύ συχνά χρήση PWM(Pulse Width Modulation) κυκλωμάτων. Τα κυκλώματα αυτά, επί της ουσίας είναι μικρά κυκλώματα ανόρθωσης για τον έλεγχο της ισχύος που παρέχεται σε ηλεκτρονικές συσκευές. Η βασική λειτουργία των PWM είναι η μεταβολή της αναλογίας του ενεργού σήματος σε σχέση με το ολικό σήμα. Με άλλα λόγια, όσο περισσότερο βρίσκεται σε ενεργή κατάσταση το κύκλωμά μας, τόσο περισσότερο θα βρίσκεται σε λειτουργία η μηχανή μας, αυτό είναι συνυφασμένο και με την ταχύτητα των κινητήρων. Ο χρόνος κατά τον οποίον θα τροφοδοτείται ο κινητήρας μας θα αυξάνει την μέση τάση του κυκλώματος άρα και την ταχύτητα του και όσο περισσότερο βρίσκεται σε κατάσταση αποκοπής το κύκλωμα μας, τόσο θα μειώνεται ο κύκλος λειτουργίας και η μέση τάση τροφοδοσίας, συνεπώς και η ταχύτητα του κινητήρα μας[24]. Έτσι, το PWM είναι μια ιδιαίτερα ευέλικτη και αποτελεσματική τεχνική ελέγχου, η οποία χρησιμοποιείται σε ένα ευρύ φάσμα τεχνολογικών και βιομηχανικών εφαρμογών.

Επιπλέον, ο έλεγχος στροφών μπορεί να γίνει μέσω υπέρυθρων, μια μέθοδο ευρέως χρησιμοποιημένη, η οποία αφορά στην εκπομπή υπέρυθρου φωτός και η συλλογή του σε έναν αισθητήρα. Η επιφάνεια, η οποία δέχεται τις ακτίνες φωτός, πρέπει να έχει τουλάχιστον δυο φάσεις, μια όπου ανακλά τις ακτίνες πίσω και μια που τις απορροφά ώστε να μπορούμε να διακρίνουμε υψηλή και χαμηλή κατάσταση στον αισθητήρα φωτός μας. Συνήθως κατεργαζόμαστε μια οπή στο κέλυφος του συμπιεστή για παρόμοιους λόγους με τους οποίους αναφέραμε παραπάνω και τοποθετούμε τον εκπομπό και τον δέκτη κοντά στο περικόχλιο ατράκτου ή στο κυκλικό σημείο του συμπιεστή όπου ακουμπάει το περικόχλιο. Τέλος, με ένα πινέλο βάφουμε ένα μέρος της επιφάνειας, συνήθως με μαύρη μπογιά, ώστε να πετύχουμε τις δύο φάσεις που αναφέραμε, μια ανάκλασης και μια απορρόφησης. Κύρια πλεονεκτήματα μέτρησης, με την μέθοδο αυτήν, είναι πως εμφανίζει πολύ μεγαλύτερη ταχύτητα μέτρησης συγκριτικά με τον αισθητήρα eddy, καθώς και μηδενική επιρροή από μαγνητικά και ηλεκτρικά πεδία. Τέλος, το κόστος μονάδας είναι σημαντικά μικρότερο από μια μονάδα μαγνητικών πεδίων. Το μειονέκτημα που εμφανίζει το σύστημα αυτό είναι η επιρροή του από διαρροή

φωτός, ακαθαρσιών και ποιότητα επιφάνειας, τόσο στην ανάκλαση φωτός όσο και στις ανοχές απόστασης από τον αισθητήρα. Αν η επιφάνεια μέτρησης, δηλαδή, είναι έκκεντρη με τον κύριο άξονα περιστροφής και συνεπώς δεν κρατάει σταθερή απόσταση με την κάθε περιστροφή, το σήμα μπορεί να αποκτήσει σημαντικό θόρυβο για να είναι χρήσιμο [25].

2.5 Σημάδια αναφοράς

Τα σημάδια αναφοράς (fiducial marks), όπως φαίνονται στην επόμενη εικόνα, είναι ένα σύνολο σημείων κοντά στο τεμάχιο εργασίας τα οποία επιστρέφουν διάφορες πληροφορίες στο σύστημα. Η πιο βασική από αυτές είναι η οριοθέτηση του χώρου εργασίας και η λειτουργία μηδενισμού παραμέτρων (homíng) που επιστρέφει το εργαλείο στην βασική του θέση πριν την έναρξη κατεργασίας. Τα σημάδια αυτά επιστρέφουν πληροφορίες για το υλικό, τα σημεία εντολών και τη σειρά των εργασιών. Τέτοια σημάδια παρατηρούνται πολύ συχνά στις πλακέτες PCB τις οποίες χρησιμοποιούμε καθημερινά. Η αρχή λειτουργίας τους είναι η χρήση ενός προβολέα που χρησιμοποιείται για την απεικόνιση γραμμών περιγράμματος [26], ο οποίος αναγνωρίζει σημεία όπως η τοποθέτηση διαφόρων στοιχείων στην πλακέτα, τα όρια της πλακέτας κτλ. Συνήθως, τα παρατηρούμε στις γωνίες των πλακετών μας, καθώς είναι σημεία χωρίς μεταλλικές επιστρώσεις που να επηρεάζουν το κύκλωμα της πλακέτας.



Εικόνα 7: 7 Σημάδι αναφοράς(marker)[26]

2.6 Τερματικοί διακόπτες

Οι τερματικοί διακόπτες (endstop) είναι αναπόσπαστα εξαρτήματα στις μηχανές CNC (Computer Numerical Control), που εξυπηρετούν διάφορες λειτουργίες βελτιώνοντας την απόδοση, την ασφάλεια και την ακρίβεια του συστήματος.

Οι διακόπτες αυτοί χρησιμοποιούνται για να ορίσουν ένα γνωστό σημείο αναφοράς για τη μηχανή CNC γνωστό και ως “home”. Όταν η μηχανή ενεργοποιείται, δεν γνωρίζει την ακριβή θέση του εργαλείου ή του αντικειμένου εργασίας και για αυτό πρέπει να οριοθετηθεί η έναρξή της. Με την ενεργοποίηση των διακοπών αυτών λαμβάνει σήμα η μηχανή και γνωρίζει τα όρια του χώρου εργασίας και κίνησης. Έτσι, συμβάλλουν στην αυτοματοποίηση των μηχανών CNC, επιτρέποντάς τους να βρουν τη θέση έναρξης τους χωρίς χειροκίνητη παρέμβαση. Αυτή η αυτοματοποίηση μειώνει τον χρόνο ρύθμισης και αυξάνει την αποδοτικότητα της ροής εργασιών και την καθιστά κρίσιμη για την επαναληψιμότητα, επιτρέποντας στη μηχανή να παράγει πολλά αντίγραφα με μεγάλη ποιότητα.

Οι διακόπτες endstop διαδραματίζουν επίσης σημαντικό ρόλο στην αποτροπή της υπέρβασης των φυσικών ορίων της μηχανής. Εάν η μηχανή προσπαθήσει να κινηθεί πέρα από την επιτρεπτή της εμβέλεια, ο διακόπτης θα ενεργοποιηθεί, σταματώντας την περαιτέρω κίνηση.

Αυτός ο μηχανισμός ασφαλείας βοηθά στην προστασία τόσο του συστήματος όσο και του αντικειμένου εργασίας από ζημιές. Επιπλέον, σε ορισμένα συστήματα, αυτοί προσφέρουν άμεση διακοπή της κατεργασίας για αποφυγή καίριων σφαλμάτων.

Με την μεγάλη χρήση στο πέρασμα του χρόνου, παράγοντες όπως η μηχανική φθορά, η θερμική διαστολή μπορούν να οδηγήσουν σε σφάλματα θέσης στις μηχανές CNC. Οι διακόπτες endstop βοηθούν στην μείωση αυτών των προβλημάτων επιτρέποντας στη μηχανή να επαναφέρει περιοδικά τη θέση της, διορθώνοντας τυχόν σφάλματα. Τέλος, ο σημαντικότερος λόγος ύπαρξης τους σε αυτές τις μηχανές, είναι η ασφάλεια που προσφέρουν. Ορισμένοι διακόπτες είναι ηλεκτρονικά συνδεδεμένοι με τον κατάλληλο τρόπο, ώστε να διακόπτουν ολόκληρη την λειτουργία του CNC για την αποφυγή έκθεσης του εργατικού δυναμικού σε κίνδυνο και την καταστροφή μηχανικών μερών όπως η άτρακτος, τα μοτέρ, οι αισθητήρες και η τράπεζα εργασίας.

2.7 Σύστημα ψύξης και λίπανσης.

Τα συστήματα ψύξης και λίπανσης διαδραματίζουν κρίσιμο ρόλο στα CNC μηχανήματα. Κατά την κατεργασία, αναπτύσσονται εξαιρετικά υψηλές θερμοκρασίες λόγω της τριβής μεταξύ του εργαλείου και του υλικού κοπής. Εάν η θερμότητα αυτή δεν απομακρυνθεί αποτελεσματικά, υπάρχει σοβαρός κίνδυνος καταστροφής τόσο του εργαλείου όσο και του επεξεργαζόμενου τεμαχίου. Για τον λόγο αυτό, το σύστημα ψύξης εφαρμόζει ειδικό υγρό στο σημείο κοπής, το οποίο απορροφά και απομακρύνει τη θερμότητα, διατηρώντας την περιοχή σε ασφαλή θερμοκρασία [27].

Παράλληλα, το σύστημα λίπανσης, που συχνά λειτουργεί σε συνδυασμό με το σύστημα ψύξης, λιπαίνει την περιοχή κατεργασίας. Η λίπανση αυτή μειώνει τις δυνάμεις τριβής, διασφαλίζοντας μεγαλύτερη ακρίβεια στην κοπή και παρατείνοντας τη διάρκεια ζωής του εργαλείου κοπής. Επιπλέον, το λιπαντικό μέσο έχει την ιδιότητα να απομακρύνει τα ρινίσματα και τα υπολείμματα από την περιοχή κοπής, εξασφαλίζοντας έτσι μια καθαρή και αποτελεσματική διαδικασία κατεργασίας.

2.8 Τράπεζα επεξεργασίας τεμαχίων.

Η τράπεζα επεξεργασίας αποτελεί το βασικό επίπεδο στο οποίο πραγματοποιείται η κατεργασία του τεμαχίου σε ένα CNC μηχανήμα. Τα κύρια σημεία που πρέπει να επισημανθούν είναι τα εξής. Πρώτον, το υλικό κατασκευής της τράπεζας πρέπει να είναι εξαιρετικά άκαμπτο και να διαθέτει μεγάλη αντοχή, ώστε να αντέχει τις δυνάμεις που αναπτύσσονται κατά τη διάρκεια της κατεργασίας. Εξίσου σημαντική είναι η επιπεδότητα της τράπεζας, καθώς οποιαδήποτε ανωμαλία στην επιφάνειά της μπορεί να οδηγήσει σε σφάλματα και ανακρίβειες στο τελικό αποτέλεσμα του τεμαχίου προς επεξεργασία. Επιπλέον, η τράπεζα πρέπει να έχει υψηλή αντικραδασμική ικανότητα, ώστε να απορροφά και να μειώνει τις δονήσεις που παράγονται κατά την κατεργασία, αποτρέποντας έτσι τυχόν αρνητικές επιπτώσεις στην ποιότητα του τελικού προϊόντος. Τέλος, η τράπεζα επεξεργασίας εξυπηρετεί κρίσιμες λειτουργίες, όπως η σταθερή στήριξη του υλικού και η ακριβής ευθυγράμμιση του εργαλείου κοπής. Εκτός από την παροχή μιας σταθερής επιφάνειας για την κατεργασία, διασφαλίζει ότι το υλικό παραμένει ακίνητο και σωστά τοποθετημένο, ενώ η επιπεδότητα της τράπεζας διευκολύνει την ακριβή ευθυγράμμιση του εργαλείου, εξασφαλίζοντας ένα ακριβές και ποιοτικό τελικό αποτέλεσμα.

2.9 Διακόπτης έκτακτης ανάγκης και ασφάλεια

Τα συστήματα CNC έχουν μεγάλο βαθμό επικινδυνότητας, διότι στο σύνολο τους αποτελούνται από πληθώρα μηχανικών μερών τα οποία κινούνται σε υψηλές ταχύτητες και με μεγάλες δυνάμεις. Είναι απαραίτητο λοιπόν να εξοπλίζονται με διάφορες δικλείδες ασφαλείας και διακόπτες τερματισμού έκτακτης ανάγκης. Πιο αναλυτικά, τα CNC στο πλαίσιο τους εφαρμόζονται προστατευτικά καλύμματα και θύρες που εμποδίζουν την πρόσβαση στα μηχανικά μέρη για την αποφυγή τραυματισμών του προσωπικού λειτουργίας κατά την διάρκεια εργασίας. Παράλληλα, με την χρήση αισθητήρων ασφαλείας για την αποφυγή σφαλμάτων, υπερθέρμανσης και υπερφόρτωσης πολλά μηχανήματα διαθέτουν μεθόδους κλειδώματος για να μην επιτραπεί η ακούσια και μη εξειδικευμένη χρήση τους. Έτσι είναι απαραίτητος ο προστατευτικός εξοπλισμός των χειριστών που απαρτίζεται από γάντια, προστατευτικά γυαλιά και ωτοασπίδες. Τέλος, επειδή τα παραπάνω μέσα δύναται να αποτύχουν είναι απαραίτητη η προσθήκη ενός διακόπτη τερματισμού λειτουργίας έκτακτης ανάγκης. Οι διακόπτες αυτοί, συνήθως βρίσκονται εντός της εμβέλειας του χειριστή και είναι εύκολο να αναγνωρισθούν καθώς διαφέρουν από όλα τα υπόλοιπα εξαρτήματα της μηχανής.

Επίλογος

Στην δεύτερη ενότητα της εργασίας μας αναπτύξαμε λεπτομερώς το θεωρητικό υπόβαθρο του λογισμικού και των ηλεκτρονικών εξαρτημάτων που χρησιμοποιούνται συνήθως στις μηχανές CNC. Είδαμε πολύ αναλυτικά τον κώδικα G και ακόμα περισσότερο τις αρχές λειτουργίας των ηλεκτρονικών στοιχείων.

Κεφάλαιο 3^ο: Κατασκευή CNC Φρέζας – Ανάπτυξη υλικού

3.1 Εισαγωγή

Τα CNC αποτελούνται ράβδους, ρουλεμάν και ηλεκτρικά εξαρτήματα. Προκειμένου να συναρμολογηθούν και να συνδεθούν στην προβλεπόμενη θέση τους, απαιτούνται διάφορα υλικά. Φυσικά, είναι επίσης πιθανό να επεξεργαστούν τα υπάρχοντα εξαρτήματα, π.χ. να τρυπήσουμε μια τρύπα σε ένα στήριγμα για να συνδέσουμε μια ράβδο. Αλλά αυτό απαιτεί ειδικά εργαλεία, χρόνο και εμπειρία δεξιοτεχνίας, καθώς η αλλαγή ή το λάθος είναι οριστικό. Τα προφίλ αλουμινίου επιτρέπουν την τοποθέτηση μιας βίδας σε οποιοδήποτε σημείο. Τα ειδικά κατασκευασμένα πλαστικά προϊόντα, κατασκευασμένα από έναν 3D εκτυπωτή, είναι πολύ προσιτά σε τιμή και μπορούν γρήγορα να ανακατασκευαστούν και να αντικατασταθούν σε περίπτωση σφαλμάτων. Εκτός από ένα κατσαβίδι και ένα ασύρματο κατσαβίδι συν τρυπάνια, δεν χρειάζονται άλλα εργαλεία για τη σύνδεση των εξαρτημάτων μεταξύ τους.

3.2 Προμήθεια υλικών

Πίνακας Κόστους Κατασκευής

A/A	ΑΝΤΙΚΕΙΜΕΝΟ	ΑΡΙΘΜΟΣ	ΑΞΙΑ
1	CNC shield v3 V4 engraving machine 3D Printer+ A4988 / driver expansion board UNO R3 with USB cable	1	35,00 €
2	Machifit 775 Motor DC 12-24V Electric Drill with Drill Chuck for Polishing Drilling Cutting	1	35,00 €
3	Drillpro 10pcs M5 T Sliding Nut Zinc Plated Carbon Steel T Sliding Nut for 2020 Aluminum Profile	2	5 €
4	17 Stepper Motor 42 motor Nema 17 motor 1.5A 38mm 3D Printer Motor and CNC XYZ	3	60,00 €
5	Grub Screw set M3x3	1	0,25 €
6	Machifit 775 Motor Bracket U Shape Fixed Mounting Base	1	9,00 €
7	5pcs 3M Strong Pad Mounting Tape Double Sided Adhesive Acrylic Foam Tape	1	4,00 €
8	10pcs M5x10 mm M5 screw yuan cup Half round pan head black	3	9,00 €
9	End stop Micro Limit Switch for I3 Delta Kossel Makerbot Printer RAMPS 1.4	3	3,00 €
10	10pcs Dupont Wire Color Jumper Cable 2.54mm 1P-1P Female-Female for Arduino	2	1,50 €
11	Aluminum Solid Coupler 5mm - 8mm	3	4.35 €
12	Ball Bearing - 688ZZ (8mm Bore, 16mm OD)	3	1,45 €

ΚΕΦΑΛΑΙΟ 3ο

13	Bolt M4 - L10mm DIN7985	10	0,4€
14	Lead Screw T10 1000mm (Lead 8mm)	1	12,1€
15	Linear Ball Bearing - 8mm diameter Large - LML8UU	6	15,5€
16	Linear Rail Shaft Guide/Support - 8mm Diameter - SK8 - 2pcs	2	4,21€
17	Motor Mount Plate for NEMA 17 - 4mm	1	2,91€
18	Nut for Lead Screw T10 Lead 8mm (Brass	3	9,55€
19	Precision Shaft - D8mm x L500mm	4	16,15€
20	Aluminum Flexible Coupler 5mm - 10mm	2	7,74€
21	Stepper Motor Mount - Nema 17	1	1,53€
22	V-Slot 2020 500mm - Natural Anodized	1	4€
23	Μπουλόνι STANDERS εξάγωνο 4xM20mm	2	9,78 €
24	Μπουλόνι STANDERS εξάγωνο 4xM16mm	2	9,18 €
25	Σετ 9τμ κλειδιά αλλεν 1.5-10mm	1	2,09 €
26	Παξιμάδι STANDERS από ατσάλι 10mm	1	1,39 €
27	Παξιμάδι STANDERS από ατσάλι 8mm	3	3,87 €
28	Μπουλόνι STANDERS εξάγωνο 4xM30mm	2	3,18 €
29	Μπουλόνι STANDERS εξάγωνο 4xM40mm	2	3,58 €
30	Τροφοδοτικό Παγκου Topshak DC 0-30V/10A	1	45€
31	Ευλόστοκος Mercola Οξυα	1	1,5 €
32	Μεταλλικό σπρέι γκρι	1	12 €
33	Σμυριδόχαρτο	1	3 €
34	OSB wood	1	7 €
35	Σφηκτήρας ραύδου	1	12,5€
36	Σετ σφιγκτήρες DEXTER 4 τεμαχίων	1	5,59€
37	Πιστόλι θερμόκολλας RAPID Hobby 12mm	1	10,90€
38	PLA πλαστικό εκτύπωσης	1	24,99€

39	Σετ 25τμχ τρυπάνια-μύτες DEXTER	1	10,99
40	Σύνολο		405,47 €

3.3 Δημιουργία υλικών

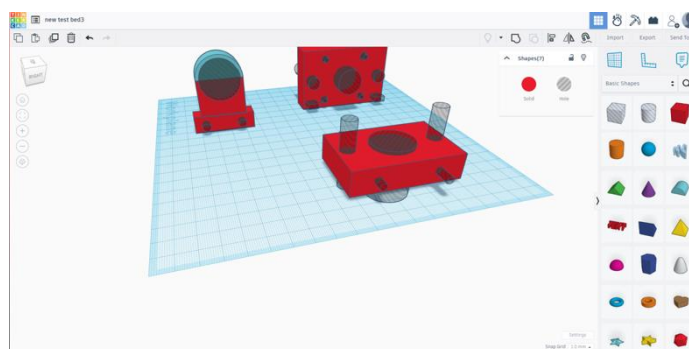
3.3.1 Δημιουργία υλικών σε σχεδιαστικό πρόγραμμα

Αρχικά παραγγείλαμε τα εξαρτήματα που θα χρειαζόμασταν. Ξεκινήσαμε κόβοντας τα σίδερα (1*ατέρμονοι άξονες και 2*άξονες ακριβείας και 3*προφίλ αλουμίνιου), αφού κόψαμε τα σίδερα ξεκινήσαμε να σχεδιάζουμε σε σχεδιαστικό πρόγραμμα (tinkercad) τα κομμάτια τα οποία θα θέλαμε να εκτυπώσουμε με τον 3D printer (Εικόνα 2).



Εικόνα 88: Σχεδίαση βάσεων στήριξης στο tinkercad

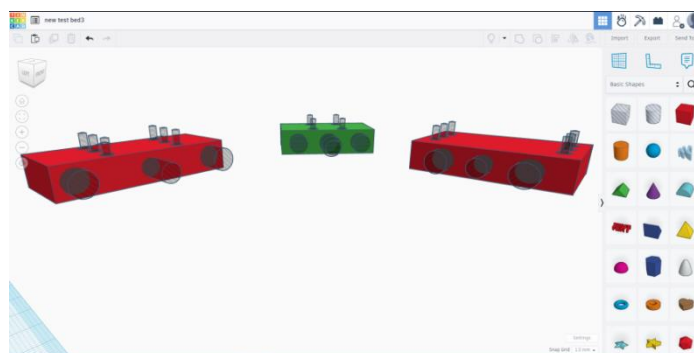
Αφού τυπώσαμε τα πλαστικά ξεκινήσαμε να στήνουμε το μηχάνημα. 2 μεγάλοι κύβοι πλαστικοί που χρησιμοποιούνται ως βάσεις φέρουν στο εσωτερικό τους τις βάσεις του μηχανήματος που είναι τα προφίλ αλουμίνιου και αυτά όλα μαζί είναι βιδωμένα στο ξύλο το οποίο είναι η βάση του μηχανήματος.



Εικόνα 99: Σχεδίαση βάσεων ατέρμονων αξόνων στο tinkercad

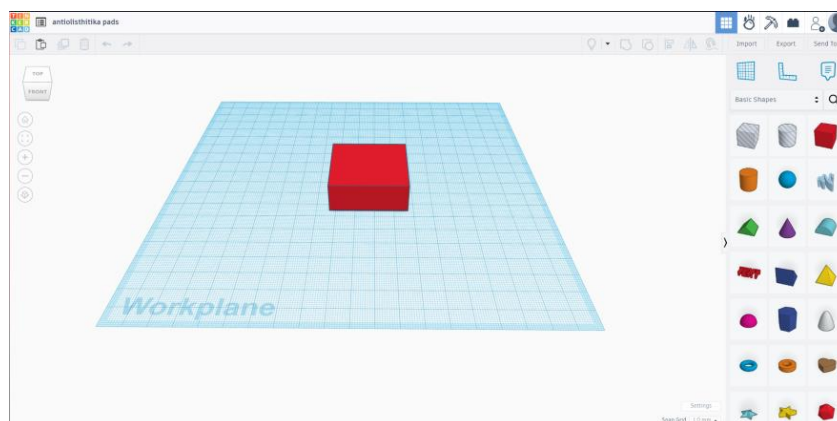
Στην παραπάνω εικόνα παρουσιάζουμε την σχεδίαση των βάσεων, των ατέρμονων αξόνων και των οδηγών κίνησης, οι οποίες, είναι πολύ σημαντικές για την κατασκευή μας, διότι από το κέντρο τους διέρχονται οι ατέρμονοι άξονες που είναι οι βασικοί άξονες κίνησης του CNC συστήματος. Αυτό συμβαίνει, επειδή στο ένα τους άκρο φέρουν τους βηματικούς κινητήρες (Nema 17), ενώ στο άλλο, έχουμε τοποθετήσει τα ρουλεμάν αποφόρτισης των αξόνων. Ακόμα, στις βάσεις σχεδιάσαμε οπές όπου εισέρχονται και ακινητοποιούνται οι οδηγοί κίνησης των

τραπεζών (Εικόνα 4).



Εικόνα 1010: Σχεδίαση τραπεζών αξόνων στο tinkercad

Στην συνέχεια, σχεδιάσαμε τις τράπεζες του κάθε άξονα. Κάθε τράπεζα έχει την δική της λειτουργία. Η πρώτη από τα αριστερά, είναι αυτή του άξονα X, που μετακινεί το εργαλείο κοπής πάνω στον άξονα αυτόν. Η επόμενη στην σειρά, ενώνεται μαζί με την πρώτη και είναι αυτή που φέρει το εργαλείο κοπής και το κινεί στον άξονα Z, ο οποίος μας επιτρέπει την διάτρηση και την χάραξη του υλικού μας. Τέλος η τρίτη βάση είναι αυτή που βρίσκεται στον ατέρμονο άξονα Y, όπου βιδώνεται στην ξύλινη βάση της κατασκευής μας και έχουμε στερεώσεις πάνω στην βάση αυτήν, την επιφάνεια εργασίας μας.



Εικόνα 1111: Σχεδίαση αντιολισθητικής βάσης της κατασκευής στο tinkercad

Τέλος σχεδιάσαμε 4 κομμάτια της παραπάνω φωτογραφίας, τα οποία βιδώσαμε και κολλήσαμε από την κάτω μεριά της κατασκευής, προκειμένου αυτή να είναι σταθερή και παράλληλη με τον πάγκο εργασίας μας. Όλα τα προαναφερθέντα κομμάτια σχεδιάστηκαν και αποθηκεύτηκαν ως αρχεία STL για να περαστούν στο πρόγραμμα του εκτυπωτή και να διαλέξουμε τις παραμέτρους εκτύπωσής τους που επιθυμούμε.

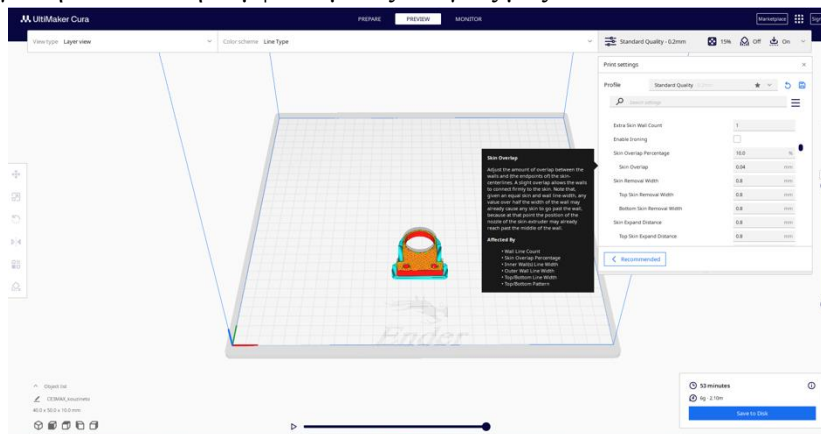
3.3.2 Σχεδιαστικό περιβάλλον στον 3D Printer

Για να καταφέρουμε να εκτυπώσουμε τα σχέδια που δημιουργήσαμε στο tinkercad, εκτός από τον 3D εκτυπωτή (εικόνα 8) χρειάστηκε να εργαστούμε και στο σχεδιαστικό λογισμικό του, διότι από το περιβάλλον εργασίας του προγράμματος μπορούμε να ρυθμίσουμε διάφορες παραμέτρους εκτύπωσης (εικόνα 9).



Εικόνα 1212: Εκτυπωτής 3D

Ο εκτυπωτής αυτός, δουλεύει παρόμοια με την πτυχιακή μας εργασία, έχει τρεις άξονες κίνησης (X,Y,Z), έναν βηματικό κινητήρα για κάθε άξονα και την τράπεζα εκτύπωσης. Ακόμα, διαθέτει ένα καρούλι με το υλικό εκτύπωσης και το ακροφύσιο που θερμαίνεται λιώνοντας το υλικό και παράγει την εκτύπωση σύμφωνα με τις ανάγκες μας.

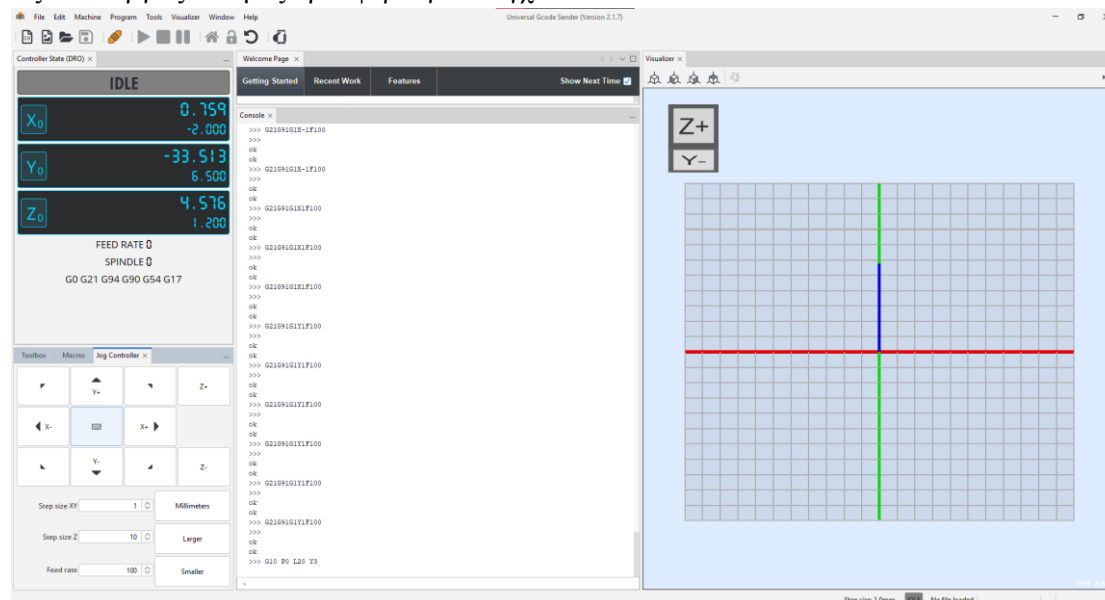


Εικόνα 13:13 Σχεδιαστικό λογισμικό 3D εκτυπωτή

Στην παραπάνω εικόνα παρουσιάζουμε το πρόγραμμα στο οποίο εργαστήκαμε προκειμένου να εκτυπώσουμε τα κομμάτια προς συναρμολόγηση για την κατασκευή μας. Στην εικόνα φαίνεται η προετοιμασία της εκτύπωσης, καθώς και οι κινήσεις που θα κάνει ο εκτυπωτής κατά την διάρκεια της λειτουργίας του. Επίσης πάνω δεξιά διακρίνεται το πλαίσιο επιλογής παραμέτρων, όπως για παράδειγμα: η χυχύτητα εκτύπωσης, η πυκνότητα της εκτύπωσης, η θερμοκρασία του ακροφύσιου και της τράπεζας και άλλες επιλογές σύμφωνα με τις δυνατότητες του εκτυπωτή και τις ανάγκες μας.

3.3.3 Λογισμικό χρήσης CNC (UGS)

Για να είμαστε σε θέση να ελέγξουμε τις κινήσεις του CNC μας και να το χειριστούμε, χρησιμοποιήσαμε το UGS (Universal G-code Sender). Αυτό είναι μια αυτόνομη εφαρμογή Java που περιλαμβάνει όλες τις εξωτερικές εξαρτήσεις για να είναι συμβατό με όλους τους ηλεκτρονικούς υπολογιστές και τα λειτουργικά που έχουν (windows, macOS κτλ.). Στην εικόνα που ακολουθεί μπορούμε να δούμε το περιβάλλον του προγράμματος, καθώς και κάποιες από τις λειτουργίες που μας προσφέρει για έλεγχο του CNC.



Εικόνα 1414: Περιβάλλον ελέγχου - λειτουργίας του CNC

Μπορούμε να διακρίνουμε ότι στο αριστερό κάτω μέρος έχουμε έναν πίνακα ελέγχου που χειροκίνητα μπορούμε να διαλέξουμε το μήκος διαδρομής, την ταχύτητα, την κλίμακα και τους άξονες του οποίους θέλουμε να χειριστούμε. Παράλληλα, στο κέντρο της οθόνης μας, η εφαρμογή μας προσφέρει μια κονσόλα στην οποία μπορούμε να δώσουμε οι ίδιοι τις εντολές που προαναφέραμε, στην μορφή G-κώδικα και ό,τι αφορούν αυτές. Επιπρόσθετα, στο δεξιό κομμάτι παρακολουθούμε στην τρισδιάστατη απεικόνιση την ακριβή τοποθεσία και κίνηση του συστήματός μας. Τέλος, να αναφέρουμε ότι, στην γραμμή εργαλείων του προγράμματος μας διακρίνεται αν έχει γίνει επιτυχής σύνδεση του arduino μας καθώς και άλλες σημαντικές λειτουργίες, όπως η εξατομικευμένη ρύθμιση της μηχανής μας στις ανάγκες μας.

3.4 Υλικά κατασκευής

3.4.1 Arduino UNO Microcontroller

Όπως ήδη αναφέρθηκε το Arduino Uno είναι μια βάση πλακέτας μικροελεγκτή στο ATmega328. Διαθέτει 14 ψηφιακές ακίδες εισόδου/εξόδου και 6 αναλογικές ακίδες εισόδου. Από τις 14 ακίδες, 6 ακίδες εισόδου/εξόδου μπορούν να χρησιμοποιηθούν ως έξοδοι PWM. Το Arduino Uno είναι ένας προηγμένος μικροελεγκτής που χρησιμοποιείται ως κιτ εκμάθησης για την κατασκευή συσκευών. Είναι εύκολο να ανεβάσουμε ένα πρόγραμμα μέσα στον μικροελεγκτή σε σύγκριση με τον παραδοσιακό μικροελεγκτή. Με το Arduino, ο χρήστης μπορεί να γράψει απευθείας την κωδικοποίηση μέσα σε αυτό μόνο με ένα φορητό υπολογιστή που έκανε εγκατάσταση λογισμικού Arduino και ένα καλώδιο USB. Το Arduino μπορεί επίσης να ξαναγραφτεί τη στιγμή ανεβάζοντας απευθείας μια νέα κωδικοποίηση στον μικροελεγκτή.

3.4.2 Arduino CNC Shield

Για την οδήγηση του μηχανήματος προμηθευτήκαμε Arduino CNC Shield (εικόνα 9) το οποίο είναι σχεδιασμένο, ώστε να συνδέεται εύκολα στην πλακέτα Arduino, παρέχοντας πρόσθετες λειτουργίες και δυνατότητες. Όπως ήδη αναφέρθηκε, αυτή η πλακέτα επέκτασης είναι κατάλληλη για μηχανές χάραξης. Το CNC Shield διαθέτει 4 διαφορετικές εξόδους που η κάθε μία από αυτές προσφέρεται για έναν βηματικό κινητήρα (X, Y, Z, A). Επιλέξαμε να χρησιμοποιήσουμε τους οδηγούς A4988 που λειτουργούν με τροφοδοσία 12-24V DC [28].

Το CNC shield χρησιμοποιεί για λογισμικό ελέγχου το GRBL. Το GRBL είναι ένα λογισμικό ανοιχτού κώδικα που εφαρμόζεται σε μικροελεγκτές, όπως το Arduino Uno που επιλέξαμε. Το GRBL μετατρέπει τις εντολές G-code σε σήματα ελέγχου, ώστε να διαβάζονται από τους βηματικούς κινητήρες. Για να πραγματοποιηθεί αυτή η διαδικασία απαιτούνται τέσσερα στάδια:

Το 1ο στάδιο στο GRBL είναι η ανάγνωση και η ανάλυση των εντολών G-code. Ο G-code, περιγράφει κινήσεις που πρέπει να κάνει το εργαλείο μηχανής, όπως μετακίνηση κατά μήκος των αξόνων με μία συγκεκριμένη ταχύτητα, όπως φαίνεται στο παρακάτω παράδειγμα.

G1 Z20 F200 ; Γραμμική κίνηση προς τα πάνω στον άξονα Z κατά 20 μονάδες με ταχύτητα 200

- **G1:** Γραμμική κίνηση με καθορισμένη ταχύτητα (που καθορίζεται από την παράμετρο F).
- **Z20:** Μετακινεί το εργαλείο/κεφαλή κατά 20 μονάδες προς τα πάνω στον άξονα Z. Στην συνέχεια στο 2ο στάδιο σχεδιάζει την διαδρομή που πρέπει να καλύψει το εργαλείο με τον πιο "ομαλό" τρόπο που γίνεται. Δηλαδή οι αλλαγές στην κατεύθυνση και στην ταχύτητα γίνονται σταδιακά και όχι απότομα, ώστε να μην επηρεάζεται η ακρίβεια και να έχουμε το βέλτιστο αποτέλεσμα.

Έπειτα στο 3ο στάδιο οι εντολές μετατρέπονται σε ηλεκτρονικά σήματα/παλμούς. Το GRBL υπολογίζει τον αριθμό των παλμών που χρειαζόμαστε, ώστε να φτάσει στην θέση που επιθυμούμε, καθώς και την συχνότητά τους, ώστε να ρυθμιστεί η ανάλογη ταχύτητα. Όπως φαίνεται στο παρακάτω παράδειγμα:

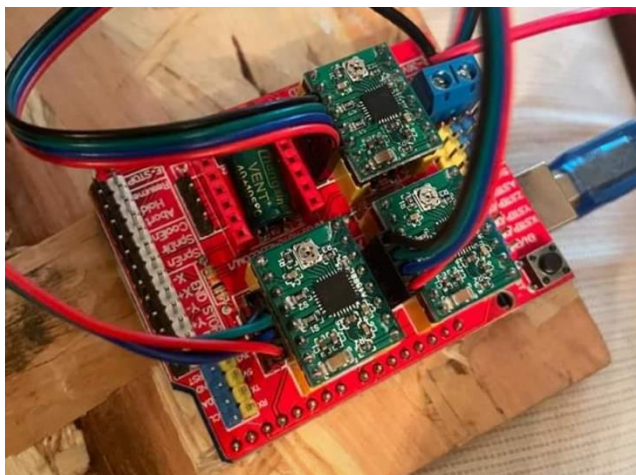
G1 X50 Y25 F1500 ; Κινήσου στη θέση X=50, Y=25 με ταχύτητα 1500 mm/min

G1 X70 Y50 F2000 ; Κινήσου στη θέση X=70, Y=50 με ταχύτητα 2000 mm/min

G1 X50 Y25 F1500:

- **G1:** Αυτή η εντολή είναι για την εκτέλεση μιας γραμμικής κίνησης.
- **X50 Y25:** Το εργαλείο πρέπει να κινηθεί στη θέση X=50 και Y=25. Το GRBL υπολογίζει την απόσταση και την κατεύθυνση που πρέπει να κινηθεί το εργαλείο.
- **F1500:** Καθορίζει την ταχύτητα κίνησης (feed rate) στα 1500 mm/min. Το GRBL μετατρέπει αυτή την ταχύτητα σε συχνότητα παλμών για τους stepper motors.

Το τελευταίο στάδιο, 4ο είναι αυτό της εκτέλεσης όπου με βάση όλα τα παραπάνω εκτελεί τις εντολές που του δώσαμε.



Εικόνα 1515: CNC shield

3.4.1 Βηματικός κινητήρας

Οι βηματικοί κινητήρες χρησιμοποιούνται για την υψηλή ικανότητα εργασίας και το επίπεδο ακρίβειάς τους, όπως αυτό που κατασκευάζουμε για το έργο μηχανών CNC. Σε αντίθεση με τους κινητήρες DC, οι βηματικοί κινητήρες είναι σύγχρονοι ηλεκτροκινητήρες χωρίς ψήκτρες που μπορούν να λειτουργήσουν και προς τις δύο κατευθύνσεις. Έχουν δυνατότητα συγκράτησης ροπής σε μηδενικές στροφές και ακριβή ψηφιακό έλεγχο χωρίς σύστημα ανάδρασης ανάλογα με το συνολικό μέγεθος του έργου και της εφαρμογής. Οι βηματικοί κινητήρες είναι συσκευές σταθερής ισχύος. Καθώς αυξάνεται η ταχύτητα, η ροπή μειώνεται, οπότε θα πρέπει να προσπαθήσουμε να βρούμε μια χρυσή τομή για την ανάγκη οδήγησης της μηχανής CNC. Οι βηματικοί κινητήρες διατίθενται σε διαφορετικούς τύπους και πολλαπλή περιέλιξη πηνίου. Οι μονοπολικό βηματικοί κινητήρες είναι τέτοιοι τύποι κινητήρων που είναι εύκολοι στην οδήγηση και έχουν συγκριτικά χαμηλή ροπή και ταχύτητα, από την άλλη πλευρά οι διπολικό κινητήρες είναι δύσκολο να οδηγηθούν αλλά έχουν υψηλή ροπή και υψηλή ταχύτητα. Ψάχνουμε για έναν κινητήρα με δυνατότητες υψηλής ταχύτητας και υψηλής ακρίβειας και θα μπορούσε να μεταφέρει ένα αντικείμενο λογικού μεγέθους, έτσι το διπολικό θα ήταν μια βολική επιλογή.



Εικόνα 1616:Βηματικός κινητήρας

Οι κινητήρες αυτοί, μας δίνουν κίνηση σε κάθε άξονα, συνεπώς καταλαβαίνουμε πως είναι από τα βασικότερα υλικά της κατασκευής γιατί χωρίς αυτούς δεν θα μπορούσαμε ούτε να τρυπήσουμε, ούτε να λειάνουμε, ούτε να έχουμε κίνηση στους καρτεσιανούς άξονες.

3.4.2 Κινητήρας κατεργασίας

Ο κινητήρας της ατράκτου είναι ένας BLDC [22] της σειράς 775 ,ο οποίος είναι ενσωματωμένος στον καρτεσιανό άξονα Z όπως φαίνεται στην επόμενη εικόνα. Αυτός ο κινητήρας τροφοδοτείται σύμφωνα με τις ονομαστικές τιμές λειτουργίας του [29] και είναι υπεύθυνος για την διάτρηση του υλικού κατεργασίας.



Εικόνα 1717: DC κινητήρας

3.4.3. Τερματικοί διακόπτες

Οι τερματικοί διακόπτες έχουν ένα απλό κύκλωμα λειτουργίας. Αυτό αποτελείται από τρεις ακροδέκτες, τους common, normally closed και normally open. Στον ακροδέκτη normally closed συνδέουμε την γείωση και στον common συνδέουμε το σήμα από το cnc shield. Σε κάθε άξονα του συστήματος έχουμε προσθέσει από έναν τέτοιο διακόπτη. Έτσι, κάθε φορά που θα πιεστεί από τις βάσεις ο διακόπτης, θα αποστέλλεται σήμα στο cnc shield και το Arduino που θα διακόπτει την λειτουργία του εκάστοτε μοτέρ για τον κάθε άξονα. Με αυτόν τον τρόπο μπορούμε να οριοθετήσουμε το CNC και να προσθέσουμε την λειτουργία “home” με την οποία θέτουμε εμείς την αρχή των αξόνων και την αρχή του CNC για την σωστή ευθυγράμμιση και κατεργασία της πλακέτας PCB [30].



Εικόνα 1818: Τερματικός διακόπτης [30]

3.4.4 Ρουλεμάν κλειστού τύπου και γραμμικά

Για να μπορέσουμε να κάνουμε εφικτή και ομαλή την κίνηση στους άξονες χρειαστήκαμε δυο τύπους ρουλεμάν. Τα συνηθισμένα κλειστού τύπου τα οποία κρατάνε σταθερούς τους ατέρμονους άξονες και τα γραμμικά ρουλεμάν [31] τα οποία επιτρέπουν την παράλληλη ολίσθηση της ράβδου. Παρακάτω τα παρουσιάζουμε σε εικόνες.



Εικόνα 19: γραμμικά ρουλεμάν



Εικόνα 2019: κλειστού τύπου ρουλεμάν

3.4.5. Άξονες κίνησης

Χωρίς τους άξονες κίνησης δεν θα ήταν εφικτή η κατασκευή του αυτοσχέδιου CNC. Αυτοί, είναι ουσιαστικά οι καρτεσιανοί άξονες του επιπέδου. Παρακάτω παραθέτουμε εικόνες με τους ατέρμονους άξονες [32] και τους άξονες ακριβείας[33].



Εικόνα 2120: άξονας ακριβείας



Εικόνα 2221: ατέρμονος άξονας

3.4.6. Προφίλ αλουμινίου

Τα προφίλ αλουμινίου αποτελούν του βασικούς πυλώνες στήριξης ολόκληρου του συστήματος. Αυτά, έχουν αυλακώσεις ώστε να μπορούμε με βίδες να ακινητοποιήσουμε υλικά της κατασκευής [34].



Εικόνα 2322: Προφίλ αλουμίνιου.

3.4.7 Συζευκτικές αξόνων

Οι συζευκτικές αξόνων είναι τα μεταλλικά κομμάτια που ενώνουν τους βηματικούς κινητήρες με τους ατέρμονους άξονες. Στην κατασκευή χρησιμοποιήσαμε δυο τύπους τέτοιων εργαλείων, τους σταθερούς και τους ευέλικτους. Οι δεύτεροι, επιτρέπουν μια μικρή ταλάντωση του άξονα για να μετριάσει την ταλάντωση ολόκληρου του άξονα που δεν την επιθυμούμε.



Εικόνα 2423: Σταθερός συζευκτήρας άξονα



Εικόνα 2524: Ευέλικτος συζευκτήρας άξονας

3.5 Συναρμολόγηση

Στη συνέχεια, μετά την προμήθεια των υλικών και την εκτύπωση εξαρτημάτων σε 3D printer, ακολούθησε η συναρμολόγηση πάνω στα προφίλ αλουμινίου με βίδες, κατάλληλες βάσεις και κουζινέτα (πλαστικά και αλουμινίου) στερεώσαμε τους βηματικούς κινητήρες (Stepper Motors) τους οποίους συνδέσαμε με τους ατέρμονους άξονες (ραβδωτά σίδερα) μέσω συζευκτών αξόνων (shaft couplers).

Για την ομαλή λειτουργία των ατέρμονων αξόνων και για να μπορούν να πηγαioέρχονται οι βάσεις σε κάθε άξονα X,Y,Z χρησιμοποιούμε τους άξονες ακριβείας που περνάνε μέσα από τις βάσεις (στις οποίες έχουμε βάλει γραμμικά ρουλεμάν) ώστε να λειτουργήσουν ως οδηγοί για τις βάσεις και να γλιστρήσουν εκεί πάνω οι βάσεις.

Στο απέναντι άκρο από κάθε βηματικό κινητήρα έχουμε βάλει ένα 3D printed κουζινέτο με ένα ρουλεμάν στην μέση ώστε να απορροφά τους κραδασμούς και να κρατάει τον ατέρμονο άξονα σε ευθεία. Έπειτα με ένα κομμάτι ξύλο (OSB) έχουμε κολλήσει τις βάσεις των αξόνων X και Z (4*) μεταξύ τους, πάνω στον άξονα Z έχουμε στερεώσει το Moter (Machifit 775) που χρησιμοποιούμε ως τρυπάνι.

Αφού τελειώσαμε με τους άξονες X και Z ξεκινήσαμε να κατασκευάζουμε τον άξονα Y. Ο άξονας Y είναι κολλημένος στην ξύλινη βάση της κατασκευής. Δώσαμε μεγάλη προσοχή στην καθετότητα του άξονα Y με τον άξονα X, κάναμε τις μετρήσεις για να είναι οι δυο άξονες κάθετοι μεταξύ τους, ώστε κατά την διάρκεια λειτουργίας να δουλεύει το σύστημα ορθά και να μπορεί να χαράζει σε ευθεία χωρίς να αποκλίνει δεξιά ή αριστερά. Πάνω στην βάση του άξονα Y τοποθετήθηκε ένα ξύλινο κομμάτι το οποίο λέγεται τράπεζα και είναι το κομμάτι όπου θα γίνεται η επεξεργασία, χάραξη και διάτρηση των πλακετών. Κοινώς, είναι ο χώρος εργασίας - λειτουργίας της κατασκευής μας.

Όλες οι βάσεις κινούνται με τον ίδιο τρόπο. Έχουν τρεις τρύπες. Στις δυο ακριανές περνάνε τα γραμμικά ρουλεμάν, που από μέσα τους περνάνε οι άξονες ακριβείας και στην τρύπα στην μέση είναι πακτωμένο ένα παξιμάδι με ραβδώσεις ίδιες με του ατέρμονου άξονα έτσι ώστε να βιδώνει και να ξεβιδώνει η βάση χωρίς όμως να αλλάζει η γωνιά της και να κινείται ευθύγραμμα μπρος και πίσω, πάνω στον ατέρμονο άξονα. Αφού τελειώσαμε με τους άξονες περάσαμε στην ενεργοποίηση των κινητήρων όπου για να γίνει αυτό έπρεπε να ρυθμίσουμε τους οδηγούς (stepper drivers) των βηματικών κινητήρων (stepper motors) γυρνώντας το ποτενσιόμετρο της τάσης αναφοράς. Αφού ρυθμίστηκαν σωστά οι τάσεις περάσαμε την βιβλιοθήκη του G-CODE στο Arduino Uno. Το Arduino Uno με το κατάλληλο εξάρτημα (CNC SHIELD) το οποίο έχει πάνω τους οδηγούς των μοτέρ. Μέσω του κώδικα-G ελέγχει την κίνηση και τις στροφές, συνεπώς ελέγχει την κίνηση στους άξονες της κατασκευής. Ολόκληρη η τροφοδοσία του έργου γίνεται από ένα τροφοδοτικό πάγκου το οποίο τροφοδοτεί τα βηματικά μοτέρ και το μοτέρ του τρυπανιού. Το μοτέρ του τρυπανιού από το datasheet του το τροφοδοτούμε με συγκεκριμένη τάση και ρεύμα ώστε να κρατάμε σταθερές στροφές στο τρυπάνι για να καταφέρουμε να κάνουμε την δουλειά μας. Για να μπορέσουμε να

λειτουργήσουμε το σύστημα μας το ελέγχουμε μέσω ενός προγράμματος στον υπολογιστή (UNIVERSAL G-CODE SENDER εικόνα 10) το οποίο συνδέεται με το Arduino Uno το οποίο έχει ήδη την βιβλιοθήκη του g code. Οι εντολές που δίνουμε μέσω του λογισμικού (Universal G-code Sender) μεταφέρονται από τον υπολογιστή στο Arduino που μέσω του CNC shield μεταφράζονται σε εντολές κίνησης, σειράς και ταχύτητας στα μοτέρ.

Στο σημείο αυτό οφείλουμε να σημειώσουμε πως επιλέγουμε να ενώσουμε τους άξονες XZ μεταξύ τους και όχι τους XY που κάνουν οι περισσότεροι, αυτό το κάνουμε γιατί με την διάταξη XZ η κατασκευή και ο έλεγχος είναι ευκολότερη καθώς και με τα μικρά μοτέρ που έχουμε αντιστεκόμαστε μόνο σε δυνάμεις κοπής που είναι πιο μικρές από τις δυνάμεις που θα δημιουργούνταν αν είχαμε ενώσει τους άξονες XY που θα θέλαμε πιο δυνατούς κινητήρες ώστε να μπορούν να επιταχύνουν και να κρατήσουν σταθερή ταχύτητα κοπής, διότι όσο μεγαλύτερο το τεμάχιο κατεργασίας στην διάταξη XY τόσο μεγαλύτερες απαιτήσεις έχουμε από τους κινητήρες.



Εικόνα 2625: Συναρμολογημένο πρωτότυπο

Επίλογος

Συνοψίζοντας, η κατασκευή μας αποτελείται από διάφορα εξαρτήματα, κάποια από τα οποία τα αγοράσαμε και άλλα τα κατασκευάσαμε σύμφωνα με τις δυνατότητες και τις ανάγκες μας. Παραθέσαμε, λοιπόν, μια εκτενή ανάλυση για τον ρόλο του κάθε εξαρτήματος, για το πως το κατασκευάσαμε και τέλος πως καταφέραμε να τα συνθέσουμε όλα μαζί, ώστε να δημιουργήσουμε το πρωτότυπο CNC.

Κεφάλαιο 4ο: Συμπεράσματα

4.1 Εισαγωγή

Παραπάνω είδαμε αναλυτικά τα μέρη από τα οποία αποτελείται η κατασκευή μας, καθώς και τον τρόπο με τον οποίο έγινε η συναρμολόγηση. Ολοκληρώνοντας την κατασκευή μας και βλέποντας την λειτουργία της κρίνουμε απαραίτητο στο σημείο αυτό να μιλήσουμε για τα πλεονεκτήματα και τα μειονεκτήματα ενός τέτοιου μηχανήματος. Ακόμα, θα αναφερθούμε στις παρατηρήσεις που κάναμε στην λειτουργία του CNC αλλά και τις πιθανές βελτιώσεις της εργασίας.

4.1.1 Πλεονεκτήματα

Οι μηχανές CNC (Computer Numerical Control) προσφέρουν πολλά πλεονεκτήματα στη βιομηχανία, καθιστώντας τις απαραίτητες για πολλές εφαρμογές. Αρχικά, διακρίνονται για την υψηλή ακρίβεια και επαναληψιμότητα που παρέχουν[35]. Με την ρύθμιση και τον προγραμματισμό μιας μηχανής CNC, μπορεί να παράγει το ίδιο κομμάτι με ακρίβεια, κάτι που είναι δύσκολο να πραγματοποιηθεί με χειροκίνητες διαδικασίες. Αυτό διασφαλίζει την ποιότητα κάθε παραγόμενου κομματιού. Επιπλέον, οι μηχανές CNC αυξάνουν την παραγωγικότητα, καθώς είναι ιδανικές για μαζική παραγωγή εξαρτημάτων, διότι έχουν την δυνατότητα να λειτουργούν αυτόματα για μεγάλα χρονικά διαστήματα. Παράλληλα, η προγραμματισμένη λειτουργία τους μειώνει τον παράγοντα ανθρώπινου λάθους καθιστώντας τις ασφαλέστερες στον χώρο εργασίας. Ένα ακόμα σημαντικό προτέρημα είναι η ικανότητα στην κατεργασία διαφόρων υλικών όπως ξύλο, μέταλλο και πλαστικό, δίνοντας τους την δυνατότητα επεξεργασίας πολύπλοκων και πολλαπλών σχεδίων. Τέλος, οι μηχανές CNC μειώνουν το εργατικό κόστος παραγωγής τεμαχίων γιατί με την αυτοματοποίηση τους ελαχιστοποιούν το εργατικό δυναμικό κάτι που τις καθιστά μακροπρόθεσμα οικονομικές.

4.1.2. Μειονεκτήματα

Παρά τα σημαντικά πλεονεκτήματα των μηχανών CNC στα οποία αναφερθήκαμε παραπάνω, παρουσιάζονται και μειονεκτήματα που θα αναλύσουμε εκτενέστερα σε αυτήν την ενότητα. Ξεκινώντας, ένα σύστημα CNC είναι ιδιαίτερα δαπανηρό επειδή μαζί με την αγορά του ίδιου πρέπει να συνυπολογίσουμε ως έξοδα και την αγορά εργαλείων, λογισμικού καθώς και την εγκατάσταση του συστήματος στον χώρο. Συμπληρωματικά, ένα από τα μειονεκτήματα ιδιαίτερης σημασίας είναι η συντήρηση της μηχανής. Η συντήρηση είναι μια πολυεπίπεδη διαδικασία, καθώς περιλαμβάνει την αντικατάσταση των μηχανικών μερών, την αλλαγή των κοπτικών μέσων και την επιδιόρθωση του λογισμικού. Οι προαναφερθείσες βλάβες απαιτούν την παρέμβαση εξειδικευμένων τεχνικών. Είναι κατανοητό ότι, εκτός από το σημαντικό κόστος που συνεπάγεται, η διαδικασία αυτή θα έχει επίσης σημαντική επίπτωση στον χρόνο παραγωγής. Οι μηχανές CNC, ως τεχνολογικά προηγμένα συστήματα, είναι άρρηκτα συνδεδεμένες με την τεχνολογία. Αυτό σημαίνει ότι οποιαδήποτε βλάβη ή τεχνικό πρόβλημα μπορεί να διακόψει την παραγωγική διαδικασία για μεγάλο χρονικό διάστημα, προκαλώντας σημαντικές καθυστερήσεις και οικονομικές επιπτώσεις. Επιπλέον, η αυτοματοποίηση που προσφέρουν οι CNC μηχανές αντικαθιστά τα παραδοσιακά μέσα κατεργασίας, οδηγώντας σε

απώλεια θέσεων εργασίας σε τομείς που βασίζονται σε πιο παραδοσιακές τεχνικές. Η εργασία σε τέτοια συστήματα απαιτεί υψηλή εξειδίκευση, εκτεταμένη γνώση και εμπειρία, κάτι που δεν είναι πάντα εύκολο να επιτευχθεί ή να καλυφθεί επαρκώς.

4.2. Αποτελέσματα

Με το πέρας της κατασκευής της εργασίας μας και θέτοντας τη σε λειτουργία κάναμε κάποιες μετρήσεις, οι οποίες αφορούν την τάση, το ρεύμα και την ισχύ που καταναλώνει η κατασκευή μας σε διάφορες περιπτώσεις. Με αυτόν τον τρόπο και λαμβάνοντας υπόψη τα δεδομένα από τα datasheet του τρυπανιού είμαστε σε θέση να ελέγξουμε την ταχύτητα της ατράκτου μας, με σκοπό την σωστή κατεργασία της PCB πλακέτας.

Στον παρακάτω πίνακα παραθέτουμε τις προαναφερθείσες μετρήσεις:

ΠΕΡΙΠΤΩΣΕΙΣ	ΤΑΣΗ(V)	ΕΝΤΑΣΗ(A)	ΙΣΧΥΣ(W)
Τρυπάνι χωρίς αρίδα	24V	0.22A	5.04W
Τρυπάνι με αρίδα	24V	0.74A	17.76W
Τρυπάνι χωρίς αρίδα με βηματικούς κινητήρες	24V	0.23A	5.52W
Τρυπάνι με αρίδα με βηματικούς κινητήρες	24V	0.73A	17.5W
Τρυπάνι κατά την διάτρηση και χρήση βηματικών κινητήρων	24V	1.48A	35.52W

4.3 Αστοχίες

Κατά τη διάρκεια της κατασκευής του πρωτοτύπου, συναντήσαμε μια σειρά από προκλήσεις και αστοχίες που κληθήκαμε να αντιμετωπίσουμε στην αρχική κατασκευή. Πρώτον, κατά το τρόχισμα των αξόνων ακριβείας και των ατέρμονων αξόνων, λόγω της θερμικής διαστολής από τις υψηλές θερμοκρασίες, παρατηρήθηκαν παραμορφώσεις στα άκρα των ράβδων. Η αντιμετώπιση αυτού του προβλήματος περιλάμβανε την επανεπεξεργασία των ατέρμονων αξόνων με λίμα και την επαναδημιουργία αυλακώσεων, ενώ για τους άξονες ακριβείας εφαρμόστηκε τρόχισμα των γωνιών για ένα λείο αποτέλεσμα.

Στη συνέχεια, κατά τη συναρμολόγηση της κατασκευής, παρατηρήσαμε αποκλίσεις στις διαστάσεις των τυπωμένων υλικών, γεγονός που μας ανάγκασε να επανασχεδιάσουμε και να εκτυπώσουμε εκ νέου τα πλαστικά μέρη. Μετά τις κατάλληλες τροποποιήσεις, πετύχαμε αποδεκτά αποτελέσματα με μικρότερες και ανεκτές αποκλίσεις. Η συναρμολόγηση, μετά τον επανασχεδιασμό, πραγματοποιήθηκε πιο ομαλά. Ωστόσο, κατά τις πρώτες δοκιμαστικές λειτουργίες, παρατηρήθηκε ανεπίτρεπτη ταλάντωση των ατέρμονων αξόνων που στηρίζουν τις τράπεζες στήριξης και κατεργασίας. Για την αντιμετώπιση αυτού του ζητήματος, αντικαταστήσαμε τους αρχικούς συμπλέκτες με νέους ελαστικούς, οι οποίοι επέτρεψαν την ταλάντωση του ατέρμονου άξονα στη βάση του, χωρίς να επηρεάζει έντονα την κίνηση.

Επιπλέον, στην τράπεζα κατεργασίας παρατηρήθηκε ανομοιογένεια και έλλειψη ζυγοστάθμισης λόγω των τρυπών και των βιδών που χρησιμοποιήθηκαν για τη στερέωση της με την πλαστική βάση του άξονα. Ως λύση, αντικαταστήσαμε τα εξαγωνικά μπουλόνια με βίδες στρογγυλής κεφαλής και τις σφίξαμε με κλειδί τύπου Άλεν. Στη συνέχεια, περάσαμε την τράπεζα με ξυλόστοκο για να καλυφθούν οι τρύπες και οι κεφαλές των βιδών, εξασφαλίζοντας ένα λείο και παράλληλο με τη βάση επίπεδο, κάθετο προς το τρυπάνι, για ασφαλή και ακριβή κατεργασία.

Η εκτύπωση των υλικών στον 3D εκτυπωτή δεν ακολούθησε τον αρχικό σχεδιασμό, λόγω έλλειψης εμπειρίας. Μετά από εκτενή έρευνα και συμβουλές από εμπειρότερους συμφοιτητές, καταφέραμε να μάθουμε την ορθή λειτουργία τόσο του εκτυπωτή, όσο και του λογισμικού σχεδίασης του. Υστέρα από πολλές αποτυχημένες προσπάθειες και αλλεπάλληλους σχεδιασμούς των υλικών εκ νέου, καταφέραμε να επιτύχουμε τα σωστά αποτελέσματα σύμφωνα με τις ανάγκες μας

Η σταθερότητα και η στιβαρότητα της κατασκευής αποτέλεσαν σημαντική πρόκληση, καθώς χωρίς αυτά, τα αποτελέσματα της λειτουργίας της μηχανής θα παρουσίαζαν μεγάλες αποκλίσεις. Το πρόβλημα επιλύθηκε με το σχεδιασμό πιο σταθερών τετράγωνων βάσεων, με περισσότερες θέσεις για κοχλίες και ροδέλες, και με την ασφάλιση των προφίλ αλουμινίου στις βάσεις χρησιμοποιώντας πιστόλι θερμοκόλλησης. Επίσης, η κοπή των ξύλων με ηλεκτρικό πριόνι παρήγαγε ανομοιόμορφα κομμάτια λόγω θρυμματισμού των σημείων κοπής, το οποίο επιλύθηκε με τη χρήση χαρτοταινίας για την αποφυγή δημιουργίας τραχιών και αδύναμων επιφανειών.

Τέλος, η οργάνωση και η σύνθεση ολόκληρης της εργασίας αποτέλεσαν το πιο βασικό πρόβλημα, καθώς τα συνεχόμενα λάθη οδήγησαν σε επανασχεδιασμό και αναπροσαρμογή εργαλείων και υλικών. Μαζί με την καταστροφή υλικών και εργαλείων κατά τη διάρκεια των δοκιμών, αναγκαστήκαμε να καθυστερήσουμε το χρονοδιάγραμμα και να αυξήσουμε το κόστος κατασκευής. Παρά τα εμπόδια, η πολύωρη και επίμονη προσπάθεια που καταβάλαμε μας έδωσε την απαραίτητη εμπειρία για να ολοκληρώσουμε την κατασκευή και να προτείνουμε λύσεις και ιδέες βελτίωσης για την ομαλότερη λειτουργία στο μέλλον. Αξίζει να σημειωθεί ότι η εργασία αυτή ήταν ιδιαίτερα κοστοβόρα, γεγονός που μας απέτρεψε από την προμήθεια καταλληλότερων υλικών και εργαλείων, τα οποία θα συνέβαλαν στην ευκολότερη εκτέλεση της διαδικασίας.

4.4 Βελτιώσεις της κατασκευής

Οι αναπόφευκτες αστοχίες που προέκυψαν κατά τη διάρκεια της κατασκευής αποτελούν έναυσμα για τη βελτίωσή της. Αρχικά, θα εξετάσουμε τις βελτιώσεις που θα μπορούσαν να πραγματοποιηθούν εάν είχαμε στη διάθεσή μας έναν μεγαλύτερο οικονομικό προϋπολογισμό για την εργασία, και στη συνέχεια θα αναφερθούμε στις γενικές βελτιώσεις των μηχανών CNC. Ξεκινώντας με την κατασκευή μας, θα μπορούσαμε να είχαμε αναζητήσει σε μηχανουργεία και σιδηρουργεία τα αντίστοιχα πλαστικά εξαρτήματα που τυπώσαμε, τα οποία θα μπορούσαν να κατασκευαστούν από ένα κράμα μετάλλων, με σκοπό να μην υπάρχουν αποκλίσεις στις διαστάσεις και η ανθεκτικότητά τους να είναι ιδανική για το εγχείρημά μας. Επιπλέον, το τρόχισμα των ατέρμωνων ράβδων και αξόνων ακριβείας θα μπορούσε να πραγματοποιηθεί από επαγγελματία σιδηρουργό με την απαραίτητη εμπειρία, προκειμένου να αποφευχθούν τα προαναφερθέντα λάθη. Επίσης, η ξύλινη βάση του αυτοσχεδίου CNC θα μπορούσε να κατασκευαστεί από ένα είδος μετάλλου κατάλληλο για την απορρόφηση των κραδασμών και την καλύτερη σταθεροποίηση της κατασκευής. Αυτές οι βελτιώσεις θα είχαν σημαντική επίδραση στη στιβαρότητα, η οποία αποτελεί κλειδί για την ορθή κατεργασία των υλικών.

Επιπρόσθετα, τα κλειστού τύπου ρουλεμάν που χρησιμοποιήθηκαν για τη στήριξη των αξόνων θα μπορούσαν να αντικατασταθούν με αυτορρυθμιζόμενα. Αυτά, έχουν τη δυνατότητα να προσαρμόζουν τη θέση τους, αντισταθμίζοντας την κακή ευθυγράμμιση των αξόνων, γεγονός που θα βελτιώνει σημαντικά την ακρίβεια και την αποτελεσματικότητα της κατασκευής. Αξίζει να αναφερθούμε και στη δύναμη των κινητήρων που χρησιμοποιήσαμε. Πρωτίστως, οι βηματικοί κινητήρες NEMA 17 θα μπορούσαν να αντικατασταθούν με μεγαλύτερους και ισχυρότερους κινητήρες που λειτουργούν σε υψηλότερα επίπεδα τάσης και ρεύματος, καθώς και να έχουν περισσότερα βήματα ανά περιστροφή. Αυτό θα μας έδινε ένα πιο ακριβές αποτέλεσμα και μεγαλύτερη ταχύτητα κίνησης των αξόνων. Ακριβώς όπως με τους βηματικούς κινητήρες, έτσι και ο κινητήρας του τρυπανιού να αντικατασταθεί με έναν κινητήρα DC με υψηλότερες ονομαστικές τιμές λειτουργίας, ώστε να είναι σε θέση να επεξεργαστεί γρηγορότερα και με περισσότερη ακρίβεια διάφορα υλικά κατεργασίας.

Επίσης, στον ηλεκτρονικό τομέα, η αλλαγή η οποία θα ήταν σοφό να πραγματοποιηθεί είναι η αντικατάσταση του Arduino Uno με έναν μικροελεγκτή με μεγαλύτερη υπολογιστική ισχύ. Αντίστοιχα, στη θέση των οδηγών των βηματικών κινητήρων θα μπορούσαμε να είχαμε χρησιμοποιήσει οδηγούς με μεγαλύτερη αντοχή στο ρεύμα. ώστε να είμαστε σε θέση να οδηγήσουμε ισχυρότερους βηματικούς κινητήρες και να κατασκευάσουμε ένα πιο αθόρυβο σύστημα γιατί οι A4988 που επιλέξαμε είναι θορυβώδεις.

Τέλος, εξαιρετικά σημαντικό είναι να εστιάσουμε την προσοχή μας στα θέματα ασφαλείας στην λειτουργία του CNC. Λόγω της λειτουργίας των μηχανών CNC με υψηλές ταχύτητες και ισχυρές δυνάμεις, για την ελαχιστοποίηση του κινδύνου και την αποφυγή τραυματισμών του εργατικού δυναμικού, είναι απαραίτητο τα κινούμενα μέρη, όπως οι μάντες χρονισμού και οι άξονες, να καλύπτονται επαρκώς, ενώ η περιοχή εργασίας πρέπει να οριοθετείται με την εγκατάσταση προστατευτικών καλυμμάτων στις πλευρές. Επιπλέον, για την πρόληψη ατυχημάτων και σφαλμάτων κατά τη διαδικασία κατεργασίας, τα ρινίσματα που παράγονται κατά την επεξεργασία του υλικού πρέπει να απομακρύνονται μέσω αισθητήρων και κατάλληλων συστημάτων ηλεκτρικής απορρόφησης.

Τέλος, τα μηχανήματα CNC, σε γενική ανάλυση, διαθέτουν μια πληθώρα διαφορετικών λειτουργιών, οι οποίες δεν μπορούν να επιτευχθούν από τους ίδιους μηχανισμούς. Ως εκ τούτου, τα CNC πρέπει να είναι εξοπλισμένα με έναν δίσκο εργαλείων που επιτρέπει την αυτοματοποιημένη αντικατάσταση και σύνδεση των εργαλείων, διασφαλίζοντας έτσι την ομαλή και αποδοτική λειτουργία τους.

Επίλογος

Καθώς ολοκληρώνουμε την εξερεύνηση των CNC συστημάτων και των εφαρμογών τους στον τομέα της βιομηχανίας, αποκαλύπτεται η αξιοσημείωτη συμβολή αυτής της τεχνολογίας στη βελτίωση της παραγωγής τεμαχίων. Η ακρίβεια, η ταχύτητα και η ευελιξία των μηχανών αυτών προσφέρουν ένα εξαιρετικό εργαλείο για την δημιουργία περίπλοκων σχεδίων που δημιουργούνται ως ψηφιακά μοντέλα σε προγράμματα CAD/CAM. Η αυτοματοποίηση των μηχανών CNC υπόσχεται μεγάλη ακρίβεια και επαναληψιμότητα στην παραγωγή τεμαχίων με μικρό κόστος και μεγάλη ταχύτητα, θέτοντας την εκβιομηχάνιση σε νέες βάσεις. Η επιτυχής υλοποίηση του συγκεκριμένου έργου αναδεικνύει τη σημασία της προσεκτικής σχεδίασης και της ακριβούς εκτέλεσης σε κάθε στάδιο κατασκευής και λειτουργίας, καθιστώντας το CNC μηχανήμα αναπόσπαστο εργαλείο στη σύγχρονη βιομηχανία.

Κεφάλαιο 5ο: Βιβλιογραφία

- [1] A. Overby, *CNC machining handbook : building, programming, and implementation*. New York: McGraw-Hill, 2011.
- [2] Π. Τσάπας, *Μελέτη και Κατασκευή Συστήματος Κίνησης, Μετρήσεων και Ελέγχου σε Τρισδιάστατο Καρτεσιανό Ρομποτικό Μηχανισμό*. Εθνικό Μετσόβιο Πανεπιστήμιο, 2019.
- [3] Α. Ορφανίδης and Π. Μπότσαρης, *Εφαρμογές συστημάτων cnc/cam/cae*. Δημοκρίτειο Πανεπιστήμιο Ξάνθης, 2012.
- [4] Δ. Κανδρής και Α. Βελώνη, *Βιομηχανική Πληροφορική και Αυτοματισμός*. Θεσσαλονίκη, Ελλάδα, Εκδόσεις Τζιόλα, 2023
- [5] Ι. Μπερέτας, *Αυτοματισμός με την χρήση PLC, Ολοκληρωμένες πρακτικές εφαρμογές*, 1st ed. ΤΖΙΟΛΑ, 2020.
- [6] S. Krar, A. Gill και Δ. Κανδρής, *Μηχανές Αριθμητικού Ελέγχου CNC*. Θεσσαλονίκη, Ελλάδα, Εκδόσεις Τζιόλα, 2014.
- [7] “CNC Slant Lathe.” Hitecindustry.com, 2018, www.hitecindustry.com/page81.html?product_id=77. Accessed 2 Sept. 2024.
- [8] “CNC Router Table Machine with 4th Rotary Axis - Cnc Wood Router - HITEC CNC.” Hiteccnc.com, 2024, www.hiteccnc.com/cnc-wood-router/show-16-150-1.html. Accessed 2 Sept. 2024.
- [9] M. Bates, *PIC microcontrollers : an introduction to microelectronics*. Oxford: Newnes, 2011.
- [10] Μπουλταδάκης Στ., Πατουλίδης Γ., Ασημόπουλος Ν., ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΙΚΡΟΕΛΕΓΚΤΩΝ ΓΙΑ ΜΗΧΑΝΙΚΟΥΣ . Εκδόσεις Τζιόλα, 2017.
- [11] Thompson, M, *Arduino: The Essential Step by Step Guide to Begin Your Own Projects (DIY Programming Projects, STEM)*, 2018.
- [12] Arduino, “Arduino® UNO R3,” Aug. 2024. Accessed: Sep. 01, 2024. [Online]. Available: <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>
- [13] Arduino. “Arduino .” *Arduino.cc*, Arduino, 2018, www.arduino.cc/. [Πρόσβαση: 29-Αυγ-2024]
- [14] unitymanufacture. “What Is a CNC Shield? - Unity Manufacture.” *Unity Manufacture*, 12 Apr.2023, unitymanufacture.com/what-is-a-cnc-shield/#:~:text=A%20CNC%20Shield%20works%20by%20receiving%20signals%20from. [Πρόσβαση: 29-Αυγ-2024]

- [15] Dejan, “G-code Explained | List of Most Important G-code Commands,” *HowToMechatronics*, May 06, 2020. <https://howtomechatronics.com/tutorials/g-code-explained-list-of-most-important-g-code-commands/> (accessed Aug. 31, 2024).
- [16] D. McClements, “G-Code: Definition, Function, and Different Types,” *www.xometry.com*, Oct. 24, 2022. <https://www.xometry.com/resources/machining/what-is-g-code/> (accessed Sep. 03, 2024).
- [17] to, Contributors. “Parametric CAD Software.” *Wikibooks.org*, Wikimedia Foundation, Inc., 8 Nov. 2009, en.wikibooks.org/wiki/OpenSCAD_User_Manual. Accessed 29 Aug. 2024. [Πρόσβαση: 29-Αυγ-2024]
- [18] N. J. Matjelo, Thapelo Moeti, and Moeti Sekhonyana, “Implementation of CNC Machine Prototype Targeted for Lesotho Industrial Manufacturing Applications,” *International Research Journal of Engineering and Technology (IRJET)*, 2021. https://www.academia.edu/49344699/Implementation_of_CNC_Machine_Prototype_Targeted_for_Lesotho_Industrial_Manufacturing_Applications (accessed Jun. 11, 2024).
- [19] alldatasheet.com, “17HS4401 PDF,” *Alldatasheet.com*, 2024. <https://www.alldatasheet.com/datasheet-pdf/view/1131976/MOTIONKING/17HS4401.html> (accessed Aug. 26, 2024).
- [20] Tantos, Andras . “H-Bridges – the Basics | Modular Circuits.” *Modularcircuits.com*, 17 Apr. 2012, www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/. [Πρόσβαση: 29-Αυγ-2024]
- [21] fuselab. “Τι είναι Βηματικός Κινητήρας; (Stepper Motor) | FuseLab | 3D Printers & CNCs.” *FuseLab*, 18 Aug. 2023, fuselab.gr/ti-einai-vimatikos-kinitiras-stepper-motor/. Accessed 2 Sept. 2024.
- [22] Electrical4U. “Brushless DC Motors (BLDC): What Are They & How Do They Work?” *Electrical4U*, 24 Dec. 2015, www.electrical4u.com/brushless-dc-motors/. [Πρόσβαση: 29-Αυγ-2024]
- [23] Δ. Κανδρής, *Ηλεκτρικές Μηχανές (5η Έκδοση)*. Θεσσαλονίκη, Ελλάδα: Εκδόσεις Τζιόλα, 2023.
- [24] Brown, Nicholas. “Pulse Width Modulation (PWM) Basics: How PWM Works.” *Kompulsa*, www.kompulsa.com/introduction-pwm-pulse-width-modulation-works/. [Πρόσβαση: 29-Αυγ-2024]
- [25] C. W. De, *Control Sensors and Actuators*. Prentice Hall, 1989.
- [26] Afanasev, Maxim Ya *et al.*, “Machine vision for auto positioning in the modular industrial equipment: A case study,” 2020, pp. 223–229. doi: <https://doi.org/10.1109/ICPS48405.2020.9274689>
- [27] I. Dincer, *Συστήματα Ψύξης και Εφαρμογές*, 3η έκδ. Τζιόλας, 2024.

- [28] alldatasheet.com, “A4988 PDF,” *Alldatasheet.com*, 2024.
<https://www.alldatasheet.com/datasheet-pdf/view/338780/ALLEGRO/A4988.html> (accessed May 25, 2024).
- [29] HandsonTechnology, “Handson Technology Data Specs 775 Ball Bearing DC Motor.” Accessed: Aug. 03, 2024. [Online]. Available:
https://www.handsontec.com/dataspecs/motor_fan/775-Motor.pdf
- [30] “How to Use Limit Switches with Arduino and Grbl: Types, Wiring, and Software Setup | Circuitist.com.” *Circuitist.com*, 2022, www.circuitist.com/limit-switches-types-and-wiring/. Accessed 29 Aug. 2024. [Πρόσβαση: 29-Αυγ-2024]
- [31] *GRobotronics*, “Γραμμικό Ρουλεμάν Μακρύ - 8mm - LML8UU,” *grobotronics.com*, 2019. <https://grobotronics.com/linear-ball-bearing-8mm-diameter-large-lml8uu.html> (accessed Aug. 26, 2024).
- [32] Thomson, “Lead Screws.” Accessed: Aug. 26, 2024. [Online]. Available:
https://www.thomsonlinear.com/downloads/screws/Neff_LeadScrews_bruk.pdf
- [33] LintechMotion, “LINTECH ® 42 Specifications Subject to Change without Notice Ordering Guide,” May 2020. Accessed: Aug. 03, 2024. [Online]. Available:
https://www.lintechmotion.com/pdf/LinTech_Round_Rail_Shafting_2020-05.pdf
- [34] Aluxprofile.com. “Aluminium Slot Profile 2020 V-Slot.” *Aluxprofile.com*, 2020, www.aluxprofile.com/aluminium-slot-profile-2020-v-slot/a3663. [Πρόσβαση 2 Sept. 2024].
- [35] Team Xometry, “CNC Turning Service,” *Xometry.com*, 2015.
<https://www.xometry.com/capabilities/cnc-machining-service/cnc-turning-service/> (accessed Aug. 26, 2024).

Παράρτημα

Gcode.cpp

```
#include "gcode.h"
#include <string.h>
#include "nuts_bolts.h"
#include <math.h>
#include "settings.h"
#include "motion_control.h"
#include "spindle_control.h"
#include "coolant_control.h"
#include "errno.h"
#include "protocol.h"
#include "report.h"

// Declare gc extern struct
parser_state_t gc;

#define FAIL(status) gc.status_code = status;

static int next_statement(char *letter, float *float_ptr, char *line, uint8_t *char_counter);

static void select_plane(uint8_t axis_0, uint8_t axis_1, uint8_t axis_2)
{
    gc.plane_axis_0 = axis_0;
    gc.plane_axis_1 = axis_1;
    gc.plane_axis_2 = axis_2;
}

void gc_init()
{
    memset(&gc, 0, sizeof(gc));
    gc.feed_rate = settings.default_feed_rate; // Should be zero at initialization.
    // gc.seek_rate = settings.default_seek_rate;
    select_plane(X_AXIS, Y_AXIS, Z_AXIS);
    gc.absolute_mode = true;

    // Load default G54 coordinate system.
    if (!(settings_read_coord_data(gc.coord_select, gc.coord_system))) {
        report_status_message(STATUS_SETTING_READ_FAIL);
    }
}
```

```

// Sets g-code parser position in mm. Input in steps. Called by the system abort and hard
// limit pull-off routines.
void gc_set_current_position(int32_t x, int32_t y, int32_t z)
{
    gc.position[X_AXIS] = x/settings.steps_per_mm[X_AXIS];
    gc.position[Y_AXIS] = y/settings.steps_per_mm[Y_AXIS];
    gc.position[Z_AXIS] = z/settings.steps_per_mm[Z_AXIS];
}

static float to_millimeters(float value)
{
    return(gc.inches_mode ? (value * MM_PER_INCH) : value);
}

// Executes one line of 0-terminated G-Code. The line is assumed to contain only uppercase
// characters and signed floating point values (no whitespace). Comments and block delete
// characters have been removed. All units and positions are converted and exported to grbl's
// internal functions in terms of (mm, mm/min) and absolute machine coordinates, respectively.
uint8_t gc_execute_line(char *line)
{
    // If in alarm state, don't process. Immediately return with error.
    // NOTE: Might not be right place for this, but also prevents $N storing during alarm.
    if (sys.state == STATE_ALARM) { return(STATUS_ALARM_LOCK); }

    uint8_t char_counter = 0;
    char letter;
    float value;
    int int_value;

    uint16_t modal_group_words = 0; // Bitflag variable to track and check modal group words
    in block
    uint8_t axis_words = 0; // Bitflag to track which XYZ(ABC) parameters exist in block

    float inverse_feed_rate = -1; // negative inverse_feed_rate means no inverse_feed_rate
    specified
    uint8_t absolute_override = false; // true(1) = absolute motion for this block only {G53}
    uint8_t non_modal_action = NON_MODAL_NONE; // Tracks the actions of modal group 0
    (non-modal)

    float target[3], offset[3];
    clear_vector(target); // XYZ(ABC) axes parameters.
    clear_vector(offset); // IJK Arc offsets are incremental. Value of zero indicates no change.

    gc.status_code = STATUS_OK;

    /* Pass 1: Commands and set all modes. Check for modal group violations.
    NOTE: Modal group numbers are defined in Table 4 of NIST RS274-NGC v3, pg.20 */

```

```

uint8_t group_number = MODAL_GROUP_NONE;
while(next_statement(&letter, &value, line, &char_counter)) {
    int_value = trunc(value);
    switch(letter) {
        case 'G':
            // Set modal group values
            switch(int_value) {
                case 4: case 10: case 28: case 30: case 53: case 92: group_number =
MODAL_GROUP_0; break;
                case 0: case 1: case 2: case 3: case 80: group_number = MODAL_GROUP_1; break;
                case 17: case 18: case 19: group_number = MODAL_GROUP_2; break;
                case 90: case 91: group_number = MODAL_GROUP_3; break;
                case 93: case 94: group_number = MODAL_GROUP_5; break;
                case 20: case 21: group_number = MODAL_GROUP_6; break;
                case 54: case 55: case 56: case 57: case 58: case 59: group_number =
MODAL_GROUP_12; break;
            }
            // Set 'G' commands
            switch(int_value) {
                case 0: gc.motion_mode = MOTION_MODE_SEEK; break;
                case 1: gc.motion_mode = MOTION_MODE_LINEAR; break;
                case 2: gc.motion_mode = MOTION_MODE_CW_ARC; break;
                case 3: gc.motion_mode = MOTION_MODE_CCW_ARC; break;
                case 4: non_modal_action = NON_MODAL_DWELL; break;
                case 10: non_modal_action = NON_MODAL_SET_COORDINATE_DATA; break;
                case 17: select_plane(X_AXIS, Y_AXIS, Z_AXIS); break;
                case 18: select_plane(X_AXIS, Z_AXIS, Y_AXIS); break;
                case 19: select_plane(Y_AXIS, Z_AXIS, X_AXIS); break;
                case 20: gc.inches_mode = true; break;
                case 21: gc.inches_mode = false; break;
                case 28: case 30:
                    int_value = trunc(10*value); // Multiply by 10 to pick up Gxx.1
                    switch(int_value) {
                        case 280: non_modal_action = NON_MODAL_GO_HOME_0; break;
                        case 281: non_modal_action = NON_MODAL_SET_HOME_0; break;
                        case 300: non_modal_action = NON_MODAL_GO_HOME_1; break;
                        case 301: non_modal_action = NON_MODAL_SET_HOME_1; break;
                        default: FAIL(STATUS_UNSUPPORTED_STATEMENT);
                    }
                    break;
                case 53: absolute_override = true; break;
                case 54: case 55: case 56: case 57: case 58: case 59:
                    gc.coord_select = int_value-54;
                    break;
                case 80: gc.motion_mode = MOTION_MODE_CANCEL; break;
                case 90: gc.absolute_mode = true; break;
                case 91: gc.absolute_mode = false; break;
                case 92:

```

```

    int_value = trunc(10*value); // Multiply by 10 to pick up G92.1
    switch(int_value) {
        case 920: non_modal_action = NON_MODAL_SET_COORDINATE_OFFSET;
break;
        case 921: non_modal_action = NON_MODAL_RESET_COORDINATE_OFFSET;
break;
        default: FAIL(STATUS_UNSUPPORTED_STATEMENT);
    }
    break;
    case 93: gc.inverse_feed_rate_mode = true; break;
    case 94: gc.inverse_feed_rate_mode = false; break;
    default: FAIL(STATUS_UNSUPPORTED_STATEMENT);
    }
    break;
case 'M':
    // Set modal group values
    switch(int_value) {
        case 0: case 1: case 2: case 30: group_number = MODAL_GROUP_4; break;
        case 3: case 4: case 5: group_number = MODAL_GROUP_7; break;
    }
    // Set 'M' commands
    switch(int_value) {
        case 0: gc.program_flow = PROGRAM_FLOW_PAUSED; break; // Program pause
        case 1: break; // Optional stop not supported. Ignore.
        case 2: case 30: gc.program_flow = PROGRAM_FLOW_COMPLETED; break; //
Program end and reset
        case 3: gc.spindle_direction = 1; break;
        case 4: gc.spindle_direction = -1; break;
        case 5: gc.spindle_direction = 0; break;
#ifdef ENABLE_M7
        case 7: gc.coolant_mode = COOLANT_MIST_ENABLE; break;
#endif
        case 8: gc.coolant_mode = COOLANT_FLOOD_ENABLE; break;
        case 9: gc.coolant_mode = COOLANT_DISABLE; break;
        default: FAIL(STATUS_UNSUPPORTED_STATEMENT);
    }
    break;
    }
    // Check for modal group multiple command violations in the current block
    if (group_number) {
        if ( bit_istrue(modal_group_words,bit(group_number)) ) {
            FAIL(STATUS_MODAL_GROUP_VIOLATION);
        } else {
            bit_true(modal_group_words,bit(group_number));
        }
        group_number = MODAL_GROUP_NONE; // Reset for next command.
    }
}
}

```

```

// If there were any errors parsing this line, we will return right away with the bad news
if (gc.status_code) { return(gc.status_code); }

/* Pass 2: Parameters. All units converted according to current block commands. Position
   parameters are converted and flagged to indicate a change. These can have multiple
   connotations
   for different commands. Each will be converted to their proper value upon execution. */
float p = 0, r = 0;
uint8_t l = 0;
char_counter = 0;
while(next_statement(&letter, &value, line, &char_counter)) {
  switch(letter) {
    case 'G': case 'M': case 'N': break; // Ignore command statements and line numbers
    case 'F':
      if (value <= 0) { FAIL(STATUS_INVALID_STATEMENT); } // Must be greater than zero
      if (gc.inverse_feed_rate_mode) {
        inverse_feed_rate = to_millimeters(value); // seconds per motion for this motion only
      } else {
        gc.feed_rate = to_millimeters(value); // millimeters per minute
      }
      break;
    case 'I': case 'J': case 'K': offset[letter-'I'] = to_millimeters(value); break;
    case 'L': l = trunc(value); break;
    case 'P': p = value; break;
    case 'R': r = to_millimeters(value); break;
    case 'S':
      if (value < 0) { FAIL(STATUS_INVALID_STATEMENT); } // Cannot be negative
      // TBD: Spindle speed not supported due to PWM issues, but may come back once
resolved.
      // gc.spindle_speed = value;
      break;
    case 'T':
      if (value < 0) { FAIL(STATUS_INVALID_STATEMENT); } // Cannot be negative
      gc.tool = trunc(value);
      break;
    case 'X': target[X_AXIS] = to_millimeters(value); bit_true(axis_words,bit(X_AXIS));
break;
    case 'Y': target[Y_AXIS] = to_millimeters(value); bit_true(axis_words,bit(Y_AXIS));
break;
    case 'Z': target[Z_AXIS] = to_millimeters(value); bit_true(axis_words,bit(Z_AXIS));
break;
    default: FAIL(STATUS_UNSUPPORTED_STATEMENT);
  }
}

// If there were any errors parsing this line, we will return right away with the bad news
if (gc.status_code) { return(gc.status_code); }

```

/* Execute Commands: Perform by order of execution defined in NIST RS274-NGC.v3, Table 8, pg.41.

NOTE: Independent non-motion/settings parameters are set out of this order for code efficiency

and simplicity purposes, but this should not affect proper g-code execution. */

// ([F]: Set feed and seek rates.)

// TODO: Seek rates can change depending on the direction and maximum speeds of each axes. When

// max axis speed is installed, the calculation can be performed here, or maybe in the planner.

if (sys.state != STATE_CHECK_MODE) {

 // ([M6]: Tool change should be executed here.)

 // [M3,M4,M5]: Update spindle state
 spindle_run(gc.spindle_direction);

 // [*M7,M8,M9]: Update coolant state
 coolant_run(gc.coolant_mode);

}

// [G54,G55,...,G59]: Coordinate system selection

if (bit_istrue(modal_group_words,bit(MODAL_GROUP_12))) { // Check if called in block
 float coord_data[N_AXIS];

 if (!(settings_read_coord_data(gc.coord_select,coord_data))) {

return(STATUS_SETTING_READ_FAIL); }

 memcpy(gc.coord_system,coord_data,sizeof(coord_data));

}

// [G4,G10,G28,G30,G92,G92.1]: Perform dwell, set coordinate system data, homing, or set axis offsets.

// NOTE: These commands are in the same modal group, hence are mutually exclusive. G53 is in this

// modal group and do not effect these actions.

uint8_t home_select;

switch (non_modal_action) {

 case NON_MODAL_DWELL:

 if (p < 0) { // Time cannot be negative.

 FAIL(STATUS_INVALID_STATEMENT);

 } else {

 // Ignore dwell in check gcode modes

 if (sys.state != STATE_CHECK_MODE) { mc_dwell(p); }

 }

 break;

 case NON_MODAL_SET_COORDINATE_DATA:

 int_value = trunc(p); // Convert p value to int.

```

    if((l != 2 && l != 20) || (int_value < 1 || int_value > N_COORDINATE_SYSTEM)) { // L2
and L20. P1=G54, P2=G55, ...
    FAIL(STATUS_UNSUPPORTED_STATEMENT);
    } else if (!axis_words && l==2) { // No axis words.
    FAIL(STATUS_INVALID_STATEMENT);
    } else {
    int_value--; // Adjust P index to EEPROM coordinate data indexing.
    if (l == 20) {
        settings_write_coord_data(int_value,gc.position);
        // Update system coordinate system if currently active.
        if (gc.coord_select == int_value) {
memcpy(gc.coord_system,gc.position,sizeof(gc.position)); }
        } else {
        float coord_data[N_AXIS];
        if (!settings_read_coord_data(int_value,coord_data)) {
return(STATUS_SETTING_READ_FAIL); }
        // Update axes defined only in block. Always in machine coordinates. Can change non-
active system.
        uint8_t i;
        for (i=0; i<N_AXIS; i++) { // Axes indices are consistent, so loop may be used.
            if ( bit_istrue(axis_words,bit(i)) ) { coord_data[i] = target[i]; }
        }
        settings_write_coord_data(int_value,coord_data);
        // Update system coordinate system if currently active.
        if (gc.coord_select == int_value) {
memcpy(gc.coord_system,coord_data,sizeof(coord_data)); }
        }
    }
    axis_words = 0; // Axis words used. Lock out from motion modes by clearing flags.
    break;
case NON_MODAL_GO_HOME_0: case NON_MODAL_GO_HOME_1:
    // Move to intermediate position before going home. Obeys current coordinate system and
offsets
    // and absolute and incremental modes.
    if (axis_words) {
        // Apply absolute mode coordinate offsets or incremental mode offsets.
        uint8_t i;
        for (i=0; i<N_AXIS; i++) { // Axes indices are consistent, so loop may be used.
            if ( bit_istrue(axis_words,bit(i)) ) {
                if (gc.absolute_mode) {
                    target[i] += gc.coord_system[i] + gc.coord_offset[i];
                } else {
                    target[i] += gc.position[i];
                }
            }
        }
        } else {
            target[i] = gc.position[i];
        }
    }
}

```

```

    mc_line(target[X_AXIS], target[Y_AXIS], target[Z_AXIS], settings.default_seek_rate,
false);
}
// Retrieve G28/30 go-home position data (in machine coordinates) from EEPROM
float coord_data[N_AXIS];
home_select = SETTING_INDEX_G28;
if (non_modal_action == NON_MODAL_GO_HOME_1) { home_select =
SETTING_INDEX_G30; }
if (!settings_read_coord_data(home_select,coord_data)) {
return(STATUS_SETTING_READ_FAIL); }
mc_line(coord_data[X_AXIS], coord_data[Y_AXIS], coord_data[Z_AXIS],
settings.default_seek_rate, false);
memcpy(gc.position, coord_data, sizeof(coord_data)); // gc.position[] = coord_data[];
axis_words = 0; // Axis words used. Lock out from motion modes by clearing flags.
break;
case NON_MODAL_SET_HOME_0: case NON_MODAL_SET_HOME_1:
home_select = SETTING_INDEX_G28;
if (non_modal_action == NON_MODAL_SET_HOME_1) { home_select =
SETTING_INDEX_G30; }
settings_write_coord_data(home_select,gc.position);
break;
case NON_MODAL_SET_COORDINATE_OFFSET:
if (!axis_words) { // No axis words
FAIL(STATUS_INVALID_STATEMENT);
} else {
// Update axes defined only in block. Offsets current system to defined value. Does not
update when
// active coordinate system is selected, but is still active unless G92.1 disables it.
uint8_t i;
for (i=0; i<=2; i++) { // Axes indices are consistent, so loop may be used.
if (bit_istrue(axis_words,bit(i)) ) {
gc.coord_offset[i] = gc.position[i]-gc.coord_system[i]-target[i];
}
}
}
axis_words = 0; // Axis words used. Lock out from motion modes by clearing flags.
break;
case NON_MODAL_RESET_COORDINATE_OFFSET:
clear_vector(gc.coord_offset); // Disable G92 offsets by zeroing offset vector.
break;
}

// [G0,G1,G2,G3,G80]: Perform motion modes.
// NOTE: Commands G10,G28,G30,G92 lock out and prevent axis words from use in motion
modes.
// Enter motion modes only if there are axis words or a motion mode command word in the
block.
if ( bit_istrue(modal_group_words,bit(MODAL_GROUP_1)) || axis_words ) {

```

```

// G1,G2,G3 require F word in inverse time mode.
if ( gc.inverse_feed_rate_mode ) {
    if (inverse_feed_rate < 0 && gc.motion_mode != MOTION_MODE_CANCEL) {
        FAIL(STATUS_INVALID_STATEMENT);
    }
}
// Absolute override G53 only valid with G0 and G1 active.
if ( absolute_override && !(gc.motion_mode == MOTION_MODE_SEEK ||
gc.motion_mode == MOTION_MODE_LINEAR)) {
    FAIL(STATUS_INVALID_STATEMENT);
}
// Report any errors.
if (gc.status_code) { return(gc.status_code); }

// Convert all target position data to machine coordinates for executing motion. Apply
// absolute mode coordinate offsets or incremental mode offsets.
// NOTE: Tool offsets may be appended to these conversions when/if this feature is added.
uint8_t i;
for (i=0; i<=2; i++) { // Axes indices are consistent, so loop may be used to save flash space.
    if ( bit_istrue(axis_words,bit(i)) ) {
        if (!absolute_override) { // Do not update target in absolute override mode
            if (gc.absolute_mode) {
                target[i] += gc.coord_system[i] + gc.coord_offset[i]; // Absolute mode
            } else {
                target[i] += gc.position[i]; // Incremental mode
            }
        }
    } else {
        target[i] = gc.position[i]; // No axis word in block. Keep same axis position.
    }
}

switch (gc.motion_mode) {
    case MOTION_MODE_CANCEL:
        if (axis_words) { FAIL(STATUS_INVALID_STATEMENT); } // No axis words allowed
while active.
        break;
    case MOTION_MODE_SEEK:
        if (!axis_words) { FAIL(STATUS_INVALID_STATEMENT); }
        else { mc_line(target[X_AXIS], target[Y_AXIS], target[Z_AXIS],
settings.default_seek_rate, false); }
        break;
    case MOTION_MODE_LINEAR:
        // TODO: Inverse time requires F-word with each statement. Need to do a check. Also
need
        // to check for initial F-word upon startup. Maybe just set to zero upon initialization
        // and after an inverse time move and then check for non-zero feed rate each time. This

```

```

// should be efficient and effective.
if (!axis_words) { FAIL(STATUS_INVALID_STATEMENT);}
else { mc_line(target[X_AXIS], target[Y_AXIS], target[Z_AXIS],
  (gc.inverse_feed_rate_mode) ? inverse_feed_rate : gc.feed_rate,
gc.inverse_feed_rate_mode); }
break;
case MOTION_MODE_CW_ARC: case MOTION_MODE_CCW_ARC:
// Check if at least one of the axes of the selected plane has been specified. If in center
// format arc mode, also check for at least one of the IJK axes of the selected plane was
sent.
if ( !( bit_false(axis_words,bit(gc.plane_axis_2)) ) ||
  ( !r && !offset[gc.plane_axis_0] && !offset[gc.plane_axis_1] ) ) {
  FAIL(STATUS_INVALID_STATEMENT);
} else {
  if (r != 0) { // Arc Radius Mode
    /*

```

We need to calculate the center of the circle that has the designated radius and passes through both the current position and the target position. This method calculates the following

set of equations where [x,y] is the vector from current to target position, d == magnitude of

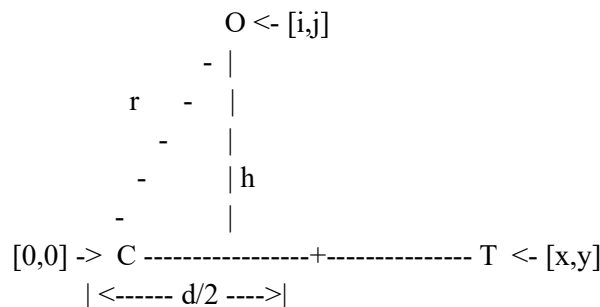
that vector, h == hypotenuse of the triangle formed by the radius of the circle, the distance to

the center of the travel vector. A vector perpendicular to the travel vector [-y,x] is scaled to the

length of h [-y/d*h, x/d*h] and added to the center of the travel vector [x/2,y/2] to form the new point

[i,j] at [x/2-y/d*h, y/2+x/d*h] which will be the center of our arc.

$$\begin{aligned}
 d^2 &= x^2 + y^2 \\
 h^2 &= r^2 - (d/2)^2 \\
 i &= x/2 - y/d*h \\
 j &= y/2 + x/d*h
 \end{aligned}$$



- C - Current position
- T - Target position
- O - center of circle that pass through both C and T
- d - distance from C to T
- r - designated radius

h - distance from center of CT to O

Expanding the equations:

$$\begin{aligned}d &\rightarrow \sqrt{x^2 + y^2} \\h &\rightarrow \sqrt{4 * r^2 - x^2 - y^2}/2 \\i &\rightarrow (x - (y * \sqrt{4 * r^2 - x^2 - y^2})) / \sqrt{x^2 + y^2}) / 2 \\j &\rightarrow (y + (x * \sqrt{4 * r^2 - x^2 - y^2})) / \sqrt{x^2 + y^2}) / 2\end{aligned}$$

Which can be written:

$$\begin{aligned}i &\rightarrow (x - (y * \sqrt{4 * r^2 - x^2 - y^2})/\sqrt{x^2 + y^2})/2 \\j &\rightarrow (y + (x * \sqrt{4 * r^2 - x^2 - y^2})/\sqrt{x^2 + y^2})/2\end{aligned}$$

Which we for size and speed reasons optimize to:

$$\begin{aligned}h_x2_div_d &= \sqrt{4 * r^2 - x^2 - y^2}/\sqrt{x^2 + y^2} \\i &= (x - (y * h_x2_div_d))/2 \\j &= (y + (x * h_x2_div_d))/2\end{aligned}$$

*/

```
// Calculate the change in position along each selected axis
float x = target[gc.plane_axis_0]-gc.position[gc.plane_axis_0];
float y = target[gc.plane_axis_1]-gc.position[gc.plane_axis_1];

clear_vector(offset);
// First, use h_x2_div_d to compute 4*h^2 to check if it is negative or r is smaller
// than d. If so, the sqrt of a negative number is complex and error out.
float h_x2_div_d = 4 * r*r - x*x - y*y;
if (h_x2_div_d < 0) { FAIL(STATUS_ARC_RADIUS_ERROR);
return(gc.status_code); }
// Finish computing h_x2_div_d.
h_x2_div_d = -sqrt(h_x2_div_d)/hypot(x,y); // == -(h * 2 / d)
// Invert the sign of h_x2_div_d if the circle is counter clockwise (see sketch below)
if (gc.motion_mode == MOTION_MODE_CCW_ARC) { h_x2_div_d = -h_x2_div_d;
}
```

/* The counter clockwise circle lies to the left of the target direction. When offset is positive,

the left hand circle will be generated - when it is negative the right hand circle is generated.

T <-- Target position

^

Clockwise circles with this center

Clockwise circles with this center will

have

will have > 180 deg of angular travel | < 180 deg of angular travel, which is a good thing!

\ | /

center of arc when h_x2_div_d is positive -> x <----- | -----> x <- center of arc when h_x2_div_d is negative

|
|

C <-- Current position */

```
// Negative R is g-code-alese for "I want a circle with more than 180 degrees of travel" (go figure!),
```

```
// even though it is advised against ever generating such circles in a single line of g-code. By
```

```
// inverting the sign of h_x2_div_d the center of the circles is placed on the opposite side of the line of
```

```
// travel and thus we get the unadvisably long arcs as prescribed.
```

```
if (r < 0) {
```

```
    h_x2_div_d = -h_x2_div_d;
```

```
    r = -r; // Finished with r. Set to positive for mc_arc
```

```
}
```

```
// Complete the operation by calculating the actual center of the arc
```

```
offset[gc.plane_axis_0] = 0.5*(x-(y*h_x2_div_d));
```

```
offset[gc.plane_axis_1] = 0.5*(y+(x*h_x2_div_d));
```

```
} else { // Arc Center Format Offset Mode
```

```
    r = hypot(offset[gc.plane_axis_0], offset[gc.plane_axis_1]); // Compute arc radius for mc_arc
```

```
}
```

```
// Set clockwise/counter-clockwise sign for mc_arc computations
```

```
uint8_t isclockwise = false;
```

```
if (gc.motion_mode == MOTION_MODE_CW_ARC) { isclockwise = true; }
```

```
// Trace the arc
```

```
mc_arc(gc.position, target, offset, gc.plane_axis_0, gc.plane_axis_1, gc.plane_axis_2,  
       (gc.inverse_feed_rate_mode) ? inverse_feed_rate : gc.feed_rate,  
gc.inverse_feed_rate_mode,
```

```
    r, isclockwise);
```

```
}
```

```
break;
```

```
}
```

```
// Report any errors.
```

```
if (gc.status_code) { return(gc.status_code); }
```

```

// As far as the parser is concerned, the position is now == target. In reality the
// motion control system might still be processing the action and the real tool position
// in any intermediate location.
memcpy(gc.position, target, sizeof(target)); // gc.position[] = target[];
}

// M0,M1,M2,M30: Perform non-running program flow actions. During a program pause, the
buffer may
// refill and can only be resumed by the cycle start run-time command.
if (gc.program_flow) {
    plan_synchronize(); // Finish all remaining buffered motions. Program paused when
complete.
    sys.auto_start = false; // Disable auto cycle start. Forces pause until cycle start issued.

    // If complete, reset to reload defaults (G92.2,G54,G17,G90,G94,M48,G40,M5,M9).
Otherwise,
    // re-enable program flow after pause complete, where cycle start will resume the program.
    if (gc.program_flow == PROGRAM_FLOW_COMPLETED) { mc_reset(); }
    else { gc.program_flow = PROGRAM_FLOW_RUNNING; }
}

return(gc.status_code);
}

// Parses the next statement and leaves the counter on the first character following
// the statement. Returns 1 if there was a statements, 0 if end of string was reached
// or there was an error (check state.status_code).
static int next_statement(char *letter, float *float_ptr, char *line, uint8_t *char_counter)
{
    if (line[*char_counter] == 0) {
        return(0); // No more statements
    }

    *letter = line[*char_counter];
    if ((*letter < 'A') || (*letter > 'Z')) {
        FAIL(STATUS_EXPECTED_COMMAND_LETTER);
        return(0);
    }
    (*char_counter)++;
    if (!read_float(line, char_counter, float_ptr)) {
        FAIL(STATUS_BAD_NUMBER_FORMAT);
        return(0);
    };
    return(1);
}
Grblmain.cpp
#include <avr/interrupt.h>

```

```

#include <avr/pgmspace.h>
#include "config.h"
#include "planner.h"
#include "nuts_bolts.h"
#include "stepper.h"
#include "spindle_control.h"
#include "coolant_control.h"
#include "motion_control.h"
#include "gcode.h"
#include "protocol.h"
#include "limits.h"
#include "report.h"
#include "settings.h"
#include "serial.h"

// Declare system global variable structure
system_t sys;

int startGrbl(void)
{
    // Initialize system
    serial_init(); // Setup serial baud rate and interrupts
    settings_init(); // Load grbl settings from EEPROM
    st_init(); // Setup stepper pins and interrupt timers
    sei(); // Enable interrupts

    memset(&sys, 0, sizeof(sys)); // Clear all system variables
    sys.abort = true; // Set abort to complete initialization
    sys.state = STATE_INIT; // Set alarm state to indicate unknown initial position

    for(;;) {

        // Execute system reset upon a system abort, where the main program will return to
        // this loop.
        // Once here, it is safe to re-initialize the system. At startup, the system will
        // automatically
        // reset to finish the initialization process.
        if (sys.abort) {
            // Reset system.
            serial_reset_read_buffer(); // Clear serial read buffer
            plan_init(); // Clear block buffer and planner variables
            gc_init(); // Set g-code parser to default state
            protocol_init(); // Clear incoming line data and execute startup lines
            spindle_init();
        }
    }
}

```

```

coolant_init();
limits_init();
st_reset(); // Clear stepper subsystem variables.

// Sync cleared gcode and planner positions to current system position, which is
only
// cleared upon startup, not a reset/abort.
sys_sync_current_position();

// Reset system variables.
sys.abort = false;
sys.execute = 0;
if (bit_istrue(settings.flags,BITFLAG_AUTO_START)) { sys.auto_start = true; }

// Check for power-up and set system alarm if homing is enabled to force homing
cycle
// by setting Grbl's alarm state. Alarm locks out all g-code commands, including the
// startup scripts, but allows access to settings and internal commands. Only a
homing
// cycle '$H' or kill alarm locks '$X' will disable the alarm.
// NOTE: The startup script will run after successful completion of the homing
cycle, but
// not after disabling the alarm locks. Prevents motion startup blocks from crashing
into
// things uncontrollably. Very bad.
#ifdef HOMING_INIT_LOCK
    if (sys.state == STATE_INIT &&
bit_istrue(settings.flags,BITFLAG_HOMING_ENABLE)) { sys.state =
STATE_ALARM; }
#endif

// Check for and report alarm state after a reset, error, or an initial power up.
if (sys.state == STATE_ALARM) {
    report_feedback_message(MESSAGE_ALARM_LOCK);
} else {
    // All systems go. Set system to ready and execute startup script.
    sys.state = STATE_IDLE;
    protocol_execute_startup();
}
}

protocol_execute_runtime();
protocol_process(); // ... process the serial protocol

```

```

    }
    return 0; /* never reached */
}
Protocol.cpp

```

```

#include <avr/io.h>
#include <avr/interrupt.h>
#include "protocol.h"
#include "gcode.h"
#include "serial.h"
#include "print.h"
#include "settings.h"
#include "config.h"
#include "nuts_bolts.h"
#include "stepper.h"
#include "report.h"
#include "motion_control.h"

```

```

static char line[LINE_BUFFER_SIZE]; // Line to be executed. Zero-terminated.
static uint8_t char_counter; // Last character counter in line variable.
static uint8_t iscomment; // Comment/block delete flag for processor to ignore
comment characters.

```

```

void protocol_init()
{
    char_counter = 0; // Reset line input
    iscomment = false;
    report_init_message(); // Welcome message

```

```

    PINOUT_DDR &= ~(PINOUT_MASK); // Set as input pins
    PINOUT_PORT |= PINOUT_MASK; // Enable internal pull-up resistors. Normal
high operation.
    PINOUT_PCMSK |= PINOUT_MASK; // Enable specific pins of the Pin Change
Interrupt
    PCICR |= (1 << PINOUT_INT); // Enable Pin Change Interrupt
}

```

// Executes user startup script, if stored.

```

void protocol_execute_startup()
{
    uint8_t n;
    for (n=0; n < N_STARTUP_LINE; n++) {
        if (!(settings_read_startup_line(n, line))) {

```

```

    report_status_message(STATUS_SETTING_READ_FAIL);
} else {
    if (line[0] != 0) {
        printString(line); // Echo startup line to indicate execution.
        report_status_message(gc_execute_line(line));
    }
}
}
}

// Pin change interrupt for pin-out commands, i.e. cycle start, feed hold, and reset. Sets
// only the runtime command execute variable to have the main program execute these
when
// its ready. This works exactly like the character-based runtime commands when picked
off
// directly from the incoming serial data stream.
ISR(PINOUT_INT_vect)
{
    // Enter only if any pinout pin is actively low.
    if ((PINOUT_PIN & PINOUT_MASK) ^ PINOUT_MASK) {
        if (bit_isfalse(PINOUT_PIN, bit(PIN_RESET))) {
            mc_reset();
        } else if (bit_isfalse(PINOUT_PIN, bit(PIN_FEED_HOLD))) {
            sys.execute |= EXEC_FEED_HOLD;
        } else if (bit_isfalse(PINOUT_PIN, bit(PIN_CYCLE_START))) {
            sys.execute |= EXEC_CYCLE_START;
        }
    }
}

// Executes run-time commands, when required. This is called from various check
points in the main
// program, primarily where there may be a while loop waiting for a buffer to clear space
or any
// point where the execution time from the last check point may be more than a fraction
of a second.
// This is a way to execute runtime commands asynchronously (aka multitasking) with
grbl's g-code
// parsing and planning functions. This function also serves as an interface for the
interrupts to
// set the system runtime flags, where only the main program handles them, removing
the need to
// define more computationally-expensive volatile variables. This also provides a
controlled way to

```

```

// execute certain tasks without having two or more instances of the same task, such as
the planner
// recalculating the buffer upon a feedhold or override.
// NOTE: The sys.execute variable flags are set by any process, step or serial interrupts,
pinouts,
// limit switches, or the main program.
void protocol_execute_runtime()
{
  if (sys.execute) { // Enter only if any bit flag is true
    uint8_t rt_exec = sys.execute; // Avoid calling volatile multiple times

    // System alarm. Everything has shutdown by something that has gone severely
wrong. Report
    // the source of the error to the user. If critical, Grbl disables by entering an infinite
    // loop until system reset/abort.
    if (rt_exec & (EXEC_ALARM | EXEC_CRIT_EVENT)) {
      sys.state = STATE_ALARM; // Set system alarm state

      // Critical event. Only hard limit qualifies. Update this as new critical events surface.
      if (rt_exec & EXEC_CRIT_EVENT) {
        report_alarm_message(ALARM_HARD_LIMIT);
        report_feedback_message(MESSAGE_CRITICAL_EVENT);
        bit_false(sys.execute, EXEC_RESET); // Disable any existing reset
        do {
          // Nothing. Block EVERYTHING until user issues reset or power cycles. Hard
limits
          // typically occur while unattended or not paying attention. Gives the user time
          // to do what is needed before resetting, like killing the incoming stream.
        } while (bit_isspace(sys.execute, EXEC_RESET));

        // Standard alarm event. Only abort during motion qualifies.
      } else {
        // Runtime abort command issued during a cycle, feed hold, or homing cycle.
Message the
        // user that position may have been lost and set alarm state to enable the alarm
lockout
        // to indicate the possible severity of the problem.
        report_alarm_message(ALARM_ABORT_CYCLE);
      }
      bit_false(sys.execute, (EXEC_ALARM | EXEC_CRIT_EVENT));
    }

    // Execute system abort.
    if (rt_exec & EXEC_RESET) {

```

```

sys.abort = true; // Only place this is set true.
return; // Nothing else to do but exit.
}

// Execute and serial print status
if (rt_exec & EXEC_STATUS_REPORT) {
    report_realtime_status();
    bit_false(sys.execute,EXEC_STATUS_REPORT);
}

// Initiate stepper feed hold
if (rt_exec & EXEC_FEED_HOLD) {
    st_feed_hold(); // Initiate feed hold.
    bit_false(sys.execute,EXEC_FEED_HOLD);
}

// Reinitializes the stepper module running state and, if a feed hold, re-plans the
buffer.
// NOTE: EXEC_CYCLE_STOP is set by the stepper subsystem when a cycle or
feed hold completes.
if (rt_exec & EXEC_CYCLE_STOP) {
    st_cycle_reinitialize();
    bit_false(sys.execute,EXEC_CYCLE_STOP);
}

if (rt_exec & EXEC_CYCLE_START) {
    st_cycle_start(); // Issue cycle start command to stepper subsystem
    if (bit_istrue(settings.flags,BITFLAG_AUTO_START)) {
        sys.auto_start = true; // Re-enable auto start after feed hold.
    }
    bit_false(sys.execute,EXEC_CYCLE_START);
}
}

// Overrides flag byte (sys.override) and execution should be installed here, since they
// are runtime and require a direct and controlled interface to the main stepper program.
}

// Directs and executes one line of formatted input from protocol_process. While mostly
// incoming streaming g-code blocks, this also executes Grbl internal commands, such
as
// settings, initiating the homing cycle, and toggling switch states. This differs from
// the runtime command module by being susceptible to when Grbl is ready to execute

```

the

```
// next line during a cycle, so for switches like block delete, the switch only effects
// the lines that are processed afterward, not necessarily real-time during a cycle,
// since there are motions already stored in the buffer. However, this 'lag' should not
// be an issue, since these commands are not typically used during a cycle.
uint8_t protocol_execute_line(char *line)
{
    // Grbl internal command and parameter lines are of the form '$4=374.3' or '$' for help
    if(line[0] == '$') {

        uint8_t char_counter = 1;
        uint8_t helper_var = 0; // Helper variable
        float parameter, value;
        switch( line[char_counter] ) {
            case 0 : report_grbl_help(); break;
            case '$' : // Prints Grbl settings
                if ( line[++char_counter] != 0 ) {
return(STATUS_UNSUPPORTED_STATEMENT); }
                else { report_grbl_settings(); }
                break;
            case '#' : // Print gcode parameters
                if ( line[++char_counter] != 0 ) {
return(STATUS_UNSUPPORTED_STATEMENT); }
                else { report_gcode_parameters(); }
                break;
            case 'G' : // Prints gcode parser state
                if ( line[++char_counter] != 0 ) {
return(STATUS_UNSUPPORTED_STATEMENT); }
                else { report_gcode_modes(); }
                break;
            case 'C' : // Set check g-code mode
                if ( line[++char_counter] != 0 ) {
return(STATUS_UNSUPPORTED_STATEMENT); }
                // Perform reset when toggling off. Check g-code mode should only work if Grbl
                // is idle and ready, regardless of alarm locks. This is mainly to keep things
                // simple and consistent.
                if ( sys.state == STATE_CHECK_MODE ) {
                    mc_reset();
                    report_feedback_message(MESSAGE_DISABLED);
                } else {
                    if (sys.state) { return(STATUS_IDLE_ERROR); }
                    sys.state = STATE_CHECK_MODE;
                    report_feedback_message(MESSAGE_ENABLED);
                }
            }
        }
    }
}
```

```

    break;
case 'X' : // Disable alarm lock
    if ( ( line[++char_counter] != 0 ) ) {
return(STATUS_UNSUPPORTED_STATEMENT); }
    if (sys.state == STATE_ALARM) {
        report_feedback_message(MESSAGE_ALARM_UNLOCK);
        sys.state = STATE_IDLE;
        // Don't run startup script. Prevents stored moves in startup from causing
accidents.
    }
    break;
case 'H' : // Perform homing cycle
    if (bit_istrue(settings.flags,BITFLAG_HOMING_ENABLE)) {
        // Only perform homing if Grbl is idle or lost.
        if ( sys.state==STATE_IDLE || sys.state==STATE_ALARM ) {
            mc_go_home();
            if (!sys.abort) { protocol_execute_startup(); } // Execute startup scripts after
successful homing.
        } else { return(STATUS_IDLE_ERROR); }
        } else { return(STATUS_SETTING_DISABLED); }
    break;
// case 'J' : break; // Jogging methods
// TODO: Here jogging can be placed for execution as a separate subprogram. It
does not need to be
// susceptible to other runtime commands except for e-stop. The jogging function is
intended to
// be a basic toggle on/off with controlled acceleration and deceleration to prevent
skipped
// steps. The user would supply the desired feedrate, axis to move, and direction.
Toggle on would
// start motion and toggle off would initiate a deceleration to stop. One could
'feather' the
// motion by repeatedly toggling to slow the motion to the desired location. Location
data would
// need to be updated real-time and supplied to the user through status queries.
// More controlled exact motions can be taken care of by inputting G0 or G1
commands, which are
// handled by the planner. It would be possible for the jog subprogram to insert
blocks into the
// block buffer without having the planner plan them. It would need to manage
de/ac-celerations
// on its own carefully. This approach could be effective and possibly size/memory
efficient.
case 'N' : // Startup lines.

```

```

if ( line[++char_counter] == 0 ) { // Print startup lines
  for (helper_var=0; helper_var < N_STARTUP_LINE; helper_var++) {
    if (!(settings_read_startup_line(helper_var, line))) {
      report_status_message(STATUS_SETTING_READ_FAIL);
    } else {
      report_startup_line(helper_var,line);
    }
  }
  break;
} else { // Store startup line
  helper_var = true; // Set helper_var to flag storing method.
  // No break. Continues into default: to read remaining command characters.
}
default : // Storing setting methods
  if(!read_float(line,          &char_counter,          &parameter))      {
return(STATUS_BAD_NUMBER_FORMAT); }
  if(line[char_counter++] !=          '=')          {
return(STATUS_UNSUPPORTED_STATEMENT); }
  if (helper_var) { // Store startup line
    // Prepare sending gcode block to gcode parser by shifting all characters
    helper_var = char_counter; // Set helper variable as counter to start of gcode block
    do {
      line[char_counter-helper_var] = line[char_counter];
    } while (line[char_counter++] != 0);
    // Execute gcode block to ensure block is valid.
    helper_var = gc_execute_line(line); // Set helper_var to returned status code.
    if (helper_var) { return(helper_var); }
    else {
      helper_var = trunc(parameter); // Set helper_var to int value of parameter
      settings_store_startup_line(helper_var,line);
    }
  } else { // Store global setting.
    if(!read_float(line,          &char_counter,          &value))      {
return(STATUS_BAD_NUMBER_FORMAT); }
    if(line[char_counter] != 0) { return(STATUS_UNSUPPORTED_STATEMENT);
}
    return(settings_store_global_setting(parameter, value));
  }
}
return(STATUS_OK); // If '$' command makes it to here, then everything's ok.

} else {
  return(gc_execute_line(line)); // Everything else is gcode
}

```

```

}

// Process and report status one line of incoming serial data. Performs an initial filtering
// by removing spaces and comments and capitalizing all letters.
void protocol_process()
{
    uint8_t c;
    while((c = serial_read()) != SERIAL_NO_DATA) {
        if ((c == '\n') || (c == '\r')) { // End of line reached

            // Runtime command check point before executing line. Prevent any further line
            // executions.
            // NOTE: If there is no line, this function should quickly return to the main program
            // when
            // the buffer empties of non-executable data.
            protocol_execute_runtime();
            if (sys.abort) { return; } // Bail to main program upon system abort

            if (char_counter > 0) { // Line is complete. Then execute!
                line[char_counter] = 0; // Terminate string
                report_status_message(protocol_execute_line(line));
            } else {
                // Empty or comment line. Skip block.
                report_status_message(STATUS_OK); // Send status message for syncing
                // purposes.
            }
            char_counter = 0; // Reset line buffer index
            iscomment = false; // Reset comment flag

        } else {
            if (iscomment) {
                // Throw away all comment characters
                if (c == ')') {
                    // End of comment. Resume line.
                    iscomment = false;
                }
            } else {
                if (c <= ' ') {
                    // Throw away whitespace and control characters
                } else if (c == '/') {
                    // Block delete not supported. Ignore character.
                } else if (c == '(') {
                    // Enable comments flag and ignore all characters until ')' or EOL.
                }
            }
        }
    }
}

```

```
    iscomment = true;
} else if (char_counter >= LINE_BUFFER_SIZE-1) {
    // Throw away any characters beyond the end of the line buffer
} else if (c >= 'a' && c <= 'z') { // Uppcase lowercase
    line[char_counter++] = c-'a'+'A';
} else {
    line[char_counter++] = c;
}
}
}
}
}
```