



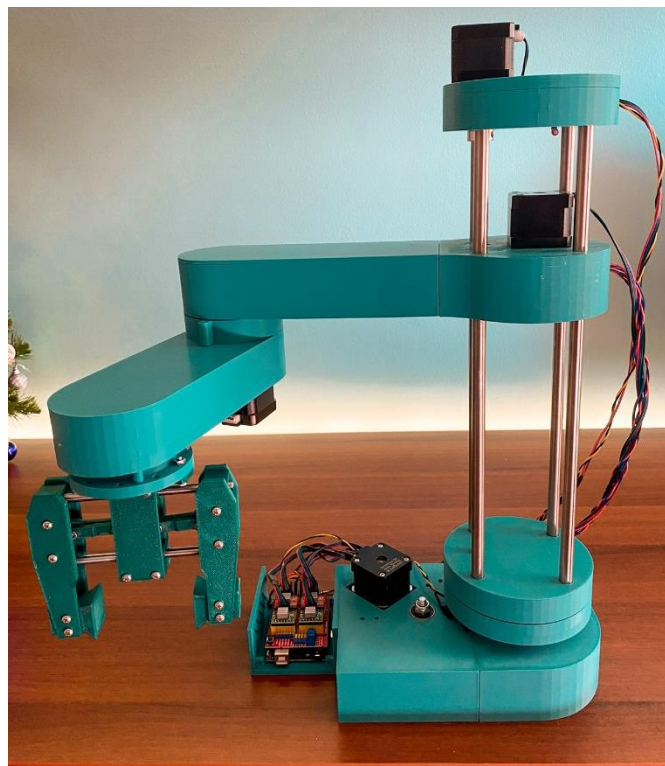
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ
ΠΑΡΑΓΩΓΗΣ

ΘΕΜΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

"Ανάπτυξη Ρομποτικού Βραχίονα Scara για Εκπαιδευτική Χρήση"



ΟΝΟΜΑ ΦΟΙΤΗΤΗ:

ΑΝΤΩΝΑΚΗΣ ΙΩΑΝΝΗΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

ΑΒΡΑΑΜ ΧΑΤΖΟΠΟΥΛΟΣ

ΑΙΓΑΛΕΩ, ΟΚΤΩΒΡΙΟΣ 2022

Η παρούσα διπλωματική εργασία εγκρίθηκε ομόφωνα από την τριμελή εξεταστική επιτροπή, η οποία ορίστηκε από την Γ.Σ. του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής του Πανεπιστημίου Δυτικής Αττικής, σύμφωνα με το νόμο και τον εγκεκριμένο Οδηγό Σπουδών του τμήματος.

Επιβλέπων: Χατζόπουλος Αβραάμ
Λέκτορας

Επιτροπή Αξιολόγησης:

.....
Χατζόπουλος Αβραάμ
Λέκτορας Εφαρμογών

.....
Σκλαβούνου Ελένη Ορσαλία
Λέκτορας Εφαρμογών

.....
Δρόσος Χρήστος
ΕΔΙΠ

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Αντωνάκης Ιωάννης του Παναγιώτη, με αριθμό μητρώου 45289 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης & Παραγωγής, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της πτυχιακής/διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών



Αντωνάκης Ιωάννης

ΠΕΡΙΛΗΨΗ

Ο Ρομποτικός Βραχίονας SCARA αποτελεί ένα τύπο βιομηχανικού ρομπότ που χάρη στη διάταξη της άρθρωσης παράλληλου άξονα ο βραχίονας είναι συμβατός στην κατεύθυνση X-Y αλλά άκαμπτος στην κατεύθυνση Z, προσδίδοντάς του ιδιαίτερο πλεονέκτημα σε λειτουργίες συναρμολόγησης. Επιπλέον η διάταξη των δύο αξόνων του του επιτρέπει να εκτείνεται σε περιορισμένες περιοχές, να ανασύρεται ή να αναδιπλώνεται στη μέση, καθιστώντας το κατάλληλο για φόρτωση/εκφόρτωση και μεταφορά εξαρτημάτων. Σκοπός της παρούσης διπλωματικής εργασίας είναι ο σχεδιασμός και η ανάπτυξη ενός ρομπότ SCARA καθώς και του συνοδευτικού λογισμικού του, για εκπαιδευτική χρήση και την αξιοποίηση του σε εργαστηριακές ασκήσεις του μαθήματος της ρομποτικής.

Επιμέρους στόχοι της διπλωματικής εργασίας είναι η προσομοίωση του ρομποτικού βραχίονα καθώς και ο σχεδιασμός και η υλοποίηση των κινηματικών εξισώσεών του σε λογισμικό MATLAB με την χρήση του Simulink, ώστε να επιτευχθεί ο έλεγχος του ρομποτικού βραχίονα σε εικονικό περιβάλλον αλλά και ο έλεγχος του πραγματικού συστήματος που εκτελείται σε πραγματικό χρόνο. Επάνω σε αυτό μπορούν να ερευνηθούν και τρόποι διασύνδεσης και επικοινωνίας του Arduino με το MATLAB και το Simulink. Επιπροσθέτως μπορούμε να μελετήσουμε, να σχεδιάσουμε και να κατασκευάσουμε τον επιμέρους κώδικα σε γλώσσα Processing με τον οποίο θα πραγματοποιείται η προσομοίωση της κινηματικής του Ρομπότ σε εικονικό κόσμο.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

SCARA, Ρομπότ, Arduino

ABSTRACT

The SCARA Robotic Arm is a type of industrial robot that, thanks to its parallel axis joint arrangement, the arm is compliant in the X-Y direction but rigid in the Z direction, giving it a particular advantage in assembly operations. In addition, its two-axle layout allows it to extend into confined areas, retract or fold in half, making it suitable for loading/unloading and transporting components. The purpose of this thesis is the design and development of a SCARA robot as well as its accompanying software, for educational use and its utilization in laboratory exercises of the robotics course.

Additional objectives of the thesis are the simulation of the robotic arm as well as the design and implementation of its kinematic equations in MATLAB software using Simulink, in order to complete the control of the robotic arm in a virtual environment as well as the control of the real system that is performed in real time. On top of that, ways to interface and communicate Arduino with MATLAB and Simulink can be researched. In addition we can study, design and build the individual code in Editing language that will create the simulation of the Robot's kinematics in a virtual world.

KEYWORDS

Scara, Robot, Arduino

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ	3
ΠΕΡΙΛΗΨΗ.....	4
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ.....	4
ABSTRACT	5
KEYWORDS	5
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	6
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ	8
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	8
1. ΕΙΣΑΓΩΓΗ.....	9
2. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ	12
2.1 Ρομπότ	12
2.2 Είδη – Κατηγορίες Ρομπότ.....	12
2.3 Είδη – Κατηγορίες Ρομποτικών Βραχιόνων.....	13
2.3 Ρομποτικός Βραχίονας SCARA	14
2.4 Κινηματική Ορθή και Αντίστροφη.....	14
Ορθή Κινηματική	14
Αντίστροφη Κινηματική.....	15
2.5 Προσομοίωση.....	15
2.6 Matlab.....	15
2.7 Processing.....	15
3. ΚΑΘΟΡΙΣΜΟΣ ΠΡΟΔΙΑΓΡΑΦΩΝ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ.....	16
4. ΣΧΕΔΙΑΣΜΟΣ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ	17
Μηχανολογικό σχέδιο	21
6. ΤΡΙΣΔΙΑΣΤΑΤΗ ΕΚΤΥΠΩΣΗ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ.....	22
7. ΗΛΕΚΤΡΟΝΙΚΟ ΚΥΚΛΩΜΑ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ	23
Μικροελεγκτής (Arduino Uno R3).....	23
CNC Shield V3.....	24
Stepper Drivers (A4988)	25
Limit Switch	25
Ηλεκτρονικό κύκλωμα	26
8. ΣΥΝΑΡΜΟΛΟΓΗΣΗ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ.....	27

8.1 Λειτουργικά Μέρη και Εξαρτήματα Κατασκευής	27
Ρουλεμάν	27
Ball Bearing.....	27
Thrust Bearing	28
Leadscrew.....	28
GT2 Ιμάντες Χρονισμού & Γρανάζια	28
8.2 Συναρμολόγηση Κατασκευής.....	28
8.3 Λίστα υλικών.....	42
9. ΑΝΑΛΥΣΗ ΚΑΙ ΣΥΓΓΡΑΦΗ ΛΟΓΙΣΜΙΚΟΥ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ	49
Διάγραμμα Ροής Κώδικα Arduino	49
Διάγραμμα Ροής Κώδικα Processing	56
Κώδικας Προγράμματος Arduino	64
Κώδικας Προγράμματος Processing	68
10. ΠΡΟΣΟΜΟΙΩΣΗ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ	79
10.1 Προσομοίωση του ρομπότ μέσω του λογισμικού MATLAB.....	79
10.2 Προσομοίωση του ρομπότ μέσω της γλώσσας προγραμματισμού Processing.	80
11. ΣΥΜΠΕΡΑΣΜΑΤΑ.....	81
13. ΒΙΒΛΙΟΓΡΑΦΙΑ.....	82

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1 : 3D Printed Εξαρτήματα του Ρομποτικού Βραχίονα.....	43
Πίνακας 2 : Ηλεκτρονικά Εξαρτήματα Ρομποτικού Βραχίονα.....	44
Πίνακας 3 : Μηχανικά Εξαρτήματα Ρομποτικού Βραχίονα.....	48

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Σχεδίαση Βάσης Ρομποτικού Βραχίονα.	18
Εικόνα 2: Σχεδίαση Άξονα Z Ρομποτικού Βραχίονα.	18
Εικόνα 3: Σχεδίαση 1 ^{ου} Συνδέσμου Ρομποτικού Βραχίονα.....	19
Εικόνα 4: Σχεδίαση 2 ^{ου} Συνδέσμου Ρομποτικού Βραχίονα.....	20
Εικόνα 5: Σχεδίαση Άκρου Εργασίας Ρομποτικού Βραχίονα.	21
Εικόνα 6: Μηχανολογικό Σχέδιο Ρομποτικού Βραχίονα.	21
Εικόνα 7: Διαδικασία Τρισδιάστατης Εκτύπωσης.	23
Εικόνα 8: Arduino Uno R3 Board.	24
Εικόνα 9 : CNC Shield V3.	24
Εικόνα 10 : Stepper Driver (A4988).	25
Εικόνα 11 : Limit Switch (Τερματικός Διακόπτης).	26
Εικόνα 12 : Ηλεκτρονικό Κύκλωμα Ελέγχου Ρομποτικού Βραχίονα.....	26
Εικόνα 13 : Τοποθέτηση Εποξικής Κόλλας & Βιδών Στήριξης.....	29
Εικόνα 14 : Σύσφιξη Βάσης Ρομποτικού Βραχίονα.....	29
Εικόνα 15 : Τοποθέτηση Ρουλεμάν Στην Βάση.....	30
Εικόνα 16: Τοποθέτηση Ωστικών Ρουλεμάν.	30
Εικόνα 17: Βίδωμα & Στήριξη 1 ^{ης} Άρθρωσης.	31
Εικόνα 18: Συναρμολόγηση Γραναζιού Μείωσης.....	31
Εικόνα 19: Ολοκληρωμένη Βάση Ρομποτικού Βραχίονα.....	32
Εικόνα 20: Συναρμολόγηση Βάσης Άξονα Z.....	33
Εικόνα 21: Ολοκληρωμένη Βάση 1ης Άρθρωσης.....	33
Εικόνα 22: Τοποθέτηση 1ου Συνδέσμου στην Βάση του.....	34
Εικόνα 23: Συναρμολόγηση 1ου Συνδέσμου	34
Εικόνα 24: Εγκατάσταση 2 ^{ης} Άρθρωσης Στον 2 ^ο Σύνδεσμο.	35
Εικόνα 25: Στήριξη 2 ^{ης} Άρθρωσης & Ένωση 1 ^{ου} -2 ^{ου} Συνδέσμου.	36
Εικόνα 26: Τοποθέτηση Ρουλεμάν Στον 2 ^ο Σύνδεσμο.....	36
Εικόνα 27: Εγκατάσταση 3 ^{ης} Άρθρωσης.....	37
Εικόνα 28: Κάτω Μέρος 2 ^{ου} Συνδέσμου	37
Εικόνα 29: Μηχανισμός Άκρου Εργασίας & Κάτω Μέρος 2 ^{ου} Συνδέσμου.....	38
Εικόνα 30: Στήριξη Μηχανισμού Άκρου Εργασίας & Κάτω Μέρος 2ου Συνδέσμου.	38
Εικόνα 31: Ηλεκτρολογική Σύνδεση & Στήριξη Καλωδίων στον 1 ^ο Σύνδεσμο.....	39
Εικόνα 32: Τοποθέτηση Συνδέσμων Βραχίονα Στον Άξονα Z.	40
Εικόνα 33: Επάνω Μέρος Άξονα Z.....	40
Εικόνα 34: Διαδικασία Σύνδεσης Καλωδίων.....	41

1. ΕΙΣΑΓΩΓΗ

Ο ρομποτικός βραχίονας SCARA αποτελεί ένα τύπο βιομηχανικού ρομπότ που χάρη στη διάταξη της άρθρωσης παράλληλου άξονα ο βραχίονας είναι συμβατός στην κατεύθυνση X-Y αλλά άκαμπτος στην κατεύθυνση Z, προσδίδοντας του ιδιαίτερο πλεονέκτημα σε λειτουργίες συναρμολόγησης. Επιπλέον η διάταξη των δύο αξόνων του του επιτρέπει να εκτείνεται σε περιορισμένες περιοχές, να ανασύρεται ή να αναδιπλώνεται στη μέση, καθιστώντας το κατάλληλο για φόρτωση/εκφόρτωση και μεταφορά εξαρτημάτων. Σκοπός της παρούσης διπλωματικής εργασίας είναι ο σχεδιασμός και η ανάπτυξη ενός ρομπότ SCARA καθώς και του συνοδευτικού λογισμικού του, για εκπαιδευτική χρήση και την αξιοποίηση του σε εργαστηριακές ασκήσεις του μαθήματος της ρομποτικής.

Επιμέρους ερευνητικοί στόχοι της εργασίας είναι η διερεύνηση δυνατότητας προσομοίωσης του προτεινόμενου ρομπότ σε εικονικό χώρο αλλά και ο έλεγχος της κινηματικής του στον πραγματικό κόσμο μέσω του λογισμικού MATLAB με την χρήση του Simulink. Εν συνεχεία μπορεί να μελετηθεί και η υλοποίηση της προσομοίωσης του ρομπότ στην γλώσσα προγραμματισμού Processing προκειμένου να αξιοποιηθεί σε εργαστηριακές ασκήσεις προσομοίωσης και μελέτης κινηματικής ρομποτικών βραχιόνων τύπου Scara.

Για τη διεκπεραίωση της παρούσης διπλωματικής εργασίας ακολουθήθηκαν τα παρακάτω βήματα **μεθοδολογίας**:

Το πρώτο βήμα είναι η βιβλιογραφική έρευνα και η έρευνα αγοράς με σκοπό τον καθορισμό των κατασκευαστικών προδιαγραφών του ρομπότ. Για τον λόγο αυτόν υλοποιήθηκαν αναζητήσεις στο διαδίκτυο, σε ερευνητικά περιοδικά και βάσεις επιστημονικών πηγών με σκοπό να επιτευχθεί η κατανόηση της λειτουργίας και η εφαρμογές ενός ρομποτικού βραχίονα βιομηχανικής χρήσης. Επίσης έγιναν αναζητήσεις σε εταιρείες παραγωγής βιομηχανικών ρομποτικών βραχιόνων (π.χ. Fanuc, ABB, Kuka κλπ.), και συγκεκριμένα τύπου Scara, ώστε να καταγραφούν ειδικότερα και να κατανοηθούν εκτενέστερα ο τρόπος λειτουργίας ενός τέτοιου βραχίονα, οι κατασκευαστικές προδιαγραφές για την ορθή λειτουργία τόσο μηχανολογικά όσο και ηλεκτρολογικά ανάλογα το μέγεθος, το βάρος του καθώς το περιβάλλον εντός του οποίου θα εργάζεται.

Στο δεύτερο βήμα της διεκπεραίωσης της κατασκευής του ρομπότ ακολούθησε η επιλογή βηματικών κινητήρων βάση των μηχανολογικών προδιαγραφών που καθορίστηκαν κατά τον μηχανολογικό σχεδιασμό του προηγούμενο βήματος. Για την επιλογή τους ελήφθησαν υπόψη αρχικά το μέγεθος και το βάρος της κάθε άρθρωσης, και στην συνέχεια η επιθυμητή ταχύτητα. Υπολογίστηκε η απαιτούμενη δύναμη στους συνδέσμους για την ομαλή κίνηση των αρθρώσεων και βάση αυτών των υπολογισμών επιλέχθηκαν οι κατάλληλοι, σε δύναμη, ροπή και μέγεθος, βηματικοί κινητήρες. Στην συνέχεια ακολούθησε έρευνα στις ηλεκτρολογικές τους προδιαγραφές ώστε να συλλεχθούν τα απαραίτητα δεδομένα για την ορθή λειτουργία και απόδοση τους. Βάση αυτών σχεδιάστηκε το κατάλληλο ηλεκτρονικό κύκλωμα επιλέγοντας τους σωστούς drivers (A4866) ώστε να υπάρχει επαρκή παροχή ρεύματος οδήγησης, η κατάλληλη πλακέτα σύνδεσης (CNC Shield V3) των drivers στον επιθυμητό μικροελεγκτή για την ακριβή οδήγηση τους αλλά και την σωστή τροφοδοσία τους.

Επιπροσθέτως επιλέχθηκαν και τα υπόλοιπα απαραίτητα μικροηλεκτρονικά για την λειτουργία του βραχίονα. Σε αυτά ανήκουν οι τερματικοί διακόπτες για τον καθορισμό της αρχικής θέσης της κάθε άρθρωσης ώστε να μπορεί να ορίζεται το αρχικό σημείο (0,0,0) του άκρου εργασίας του βραχίονα. Τέλος χρειάστηκε ένας σερβοκινητήρας για τον έλεγχο κίνησης του άκρου εργασίας καθώς και τα καλώδια ώστε να πραγματοποιηθεί η μεταξύ τους σύνδεση.

Το τρίτο βήμα της δημιουργίας του ρομποτικού βραχίονα ήταν ο τρισδιάστατος σχεδιασμός του πρότυπου ρομποτικού βραχίονα βάση των κατασκευαστικών προδιαγραφών που καθορίστηκαν στο πρώτο βήμα αλλά και του πρότυπου σχεδίου του ηλεκτρονικού κυκλώματος που καθορίστηκε στο βήμα δύο της υλοποίησης του ρομποτικού βραχίονα. Για να επιτευχθεί αυτό έγινε χρήση λογισμικού περιβάλλοντος SolidWorks στο οποίο σχεδιάστηκε βηματικά το κάθε τμήμα του με εκτενή μελέτη ώστε να επιτευχθεί κατά το μέγιστο δυνατό, η πιο αποτελεσματική λειτουργία, η καλύτερη μετάδοση κίνησης από τους κινητήρες στις αρθρώσεις, η ευκολότερη συναρμολόγηση και καλωδίωση του στην συνέχεια, αλλά και συντήρηση του όταν και αν αυτή χρειαστεί μελλοντικά. Ο σχεδιασμός ξεκίνησε από την βάση, στην συνέχεια σχεδιάστηκε ο άξονας Z, οι δύο αρθρώσεις, οι σύνδεσμοι μεταξύ όλων αυτών, οι μηχανισμοί μετάδοσης κίνησης από τους κινητήρες και τέλος το άκρο εργασίας του ρομποτικού βραχίονα.

Στο τέταρτο βήμα ακολούθησε η δημιουργία των εξαρτημάτων του ρομποτικού βραχίονα τα οποία σχεδιάστηκαν στο προηγούμενο βήμα της παρούσης διπλωματικής εργασίας. Για την ευκολότερη, ταχύτερη αλλά και οικονομικότερη δυνατή υλοποίηση του ρομποτικού βραχίονα αποφασίστηκε ότι η το καταλληλότερη διαδικασία παραγωγής θα ήταν το 3D Printing, χρησιμοποιώντας πλαστικό ως υλικό κατασκευής και συγκεκριμένα το PLA (πολυγαλακτικό οξύ). Αφού εκτυπώθηκαν όλα τα σχεδιασμένα εξαρτήματα του πρότυπου σχεδίου στην συνέχεια πραγματοποιήθηκε ένας έλεγχος ποιότητας. Κατά την διαδικασία αυτή ελέγχθηκε εάν οι διαστάσεις των εξαρτημάτων και οι ανοχές τους ήταν εντός των επιθυμητών πλαισίων ώστε να μην δημιουργηθεί πρόβλημα κατά το βήμα της συναρμολόγησης συνεπώς και της ομαλής λειτουργίας των κινητών μερών του βραχίονα. Σε περιπτώσεις που έρχονταν μικροδιορθώσεις, αυτές έγιναν με εργαλεία χειρός, όπως η λίμα, το γυαλόχαρτο αλλά και τρυπάνι.

Το τέταρτο βήμα της διεκπεραίωσης της παρούσης διπλωματικής εργασίας αποτελούνταν από την μηχανολογική συναρμολόγηση των τμημάτων του ρομποτικού βραχίονα. Αρχικά έγινε η συναρμολόγηση των πλαστικών μερών με τα μηχανικά εξαρτήματα στήριξης όπως ήταν οι βίδες και τα παξιμάδια, οι άξονες οδήγησης και στήριξης, οι μάντες μετάδοσης κίνησης και τα εξαρτήματα των κινητών μερών και αρθρώσεων όπως ήταν τα ρουλεμάν. Κατά την συνέχεια του βήματος αυτού προσαρμόστηκαν στον βραχίονα τα μέρη του ηλεκτρονικού κυκλώματος της κατασκευής, τα οποία ήταν οι βηματικοί κινητήρες που προαναφέρθηκαν, οι τερματικοί διακόπτες, οι οδηγίες των κινητήρων και η πλακέτα ελέγχου τους, ο μικροελεγκτής του κυκλώματος καθώς και τα απαραίτητα για την σύνδεσμοι τους, καλώδια.

Στο επόμενο και πέμπτο βήμα της υλοποίησης του βραχίονα, ακολούθησε ο σχεδιασμός, η συγγραφή και οι πληθώρα δοκιμών του κώδικα λειτουργίας αλλά και του κώδικα δημιουργίας ενός γραφικού περιβάλλοντος ελέγχου του ρομποτικού βραχίονα. Κατά την διάρκεια του βήματος αυτού αρχικά σχεδιάστηκε ο κώδικας ελέγχου του βραχίονα, βάση του τρόπου λειτουργίας των ρομπότ βιομηχανικού τύπου που μελετήθηκε στο βήμα ένα. Για να γίνει αυτό χρησιμοποιήθηκε η μέθοδος του λογικού διαγράμματος ροής (Flow Chart) το οποίο αποτέλεσε την βάση για να γραφεί ο κώδικας ελέγχου αναλυτικά στην γλώσσα προγραμματισμού του μικροελεγκτή (Arduino Uno). Επίσης χρειάστηκε να υλοποιηθεί και κώδικας δημιουργίας γραφικού περιβάλλοντος ελέγχου, όπου μετά από εκτενή έρευνα στο διαδίκτυο, έγινε επιλογή της γλώσσας Processing. Αυτή αποτελεί μια γλώσσα προγραμματισμού ανοιχτού κώδικα η οποία είναι βασισμένη στην Java, επάνω στην οποία προϋπήρχαν γνώσεις, οπότε ήταν ταχύτερη και πιο βατή η εκμάθησή της, συνεπώς και ευκολότερη η χρήση της σε σχέση με το να γίνει εκ νέου η μελέτη και η εκμάθηση μιας άλλης γλώσσας.

Τέλος αφού ολοκληρώθηκε και το πέμπτο βήμα, και έγιναν όλες οι απαραίτητες δοκιμές του κώδικα καθώς και μικροδιορθώσεις στην συναρμολόγηση των μηχανολογικών μερών του βραχίονα που προέκυψαν κατά την διάρκεια των δοκιμών, ο ρομποτικός βραχίονας είχε πλέον ολοκληρωθεί βάση των μηχανολογικών και ηλεκτρονικών προδιαγραφών που σχεδιάστηκαν κατά τα πρώτα βήματα της υλοποίησης του, και ήταν έτοιμος προς χρήση.

Μετά από αυτό τέθηκαν κάποιοι επιμέρους ερευνητικοί στόχοι για μελλοντικές αναβαθμίσεις και εξέλιξη του ρομποτικού βραχίονα. Ο πρώτος από αυτούς ήταν η εικονική προσομοίωση της κινηματικής του ρομπότ σε περιβάλλον MATLAB με την χρήση του Simulink, δηλαδή ο χρήστης να μπορεί αρχικά κατά την εκτέλεση των κινήσεων του ρομποτικού βραχίονα στον πραγματικό κόσμο να μπορεί να βλέπει την αντίστοιχη κίνηση του στο εικονικό περιβάλλον. Για να υλοποιηθεί αυτό χρειάζεται να δημιουργηθούν οι αντίστοιχες εξισώσεις ορθής και ανάστροφης κινηματικής του ρομποτικού βραχίονα σε κώδικα MATLAB. Για την σύνταξη του κώδικα αυτού μπορούν να χρησιμοποιηθούν οι εξισώσεις κινηματικής που δημιουργήθηκαν για την εκπόνηση της παρούσης διπλωματικής εργασίας, με τις απαραίτητες αλλαγές ώστε να μπορεί να εκτελεστεί μέσω MATLAB. Στην συνέχεια θα πρέπει να υλοποιηθεί η κατάλληλη συνδεσμολογία μεταξύ MATLAB και Arduino, και η μεταξύ τους επικοινωνία μέσω Simulink. Ένας επιπλέον ερευνητικός στόχος θα ήταν η ίδια διαδικασία να υλοποιηθεί και στην γλώσσα της Processing. Να δημιουργηθεί δηλαδή πέραν από το περιβάλλον ελέγχου ένα γραφικό περιβάλλον προσομοίωσης κίνησης του ρομποτικού βραχίονα ώστε ο χρήστης εκτελώντας κινήσεις εντός του περιβάλλοντος ελέγχου να μπορεί είτε εντός του ίδιου παραθύρου είτε σε διαφορετική να βλέπει την εικονική απεικόνιση της εντολής αυτής.

2. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

2.1 Ρομπότ

Ρομπότ ορίζεται κάθε μηχανική συσκευή οποιασδήποτε μορφής η οποία μπορεί είτε να αντικαθιστά εντελώς είτε να υποκαθιστά τον άνθρωπο σε διαφόρων ειδών εργασίες. Ένα ρομπότ μπορεί να λειτουργεί είτε ελεγχόμενο άμεσα από τον άνθρωπο είτε αυτόνομα και αυτόματα υπό τον έλεγχο ενός προγραμματισμένου υπολογιστή.

Τα ρομπότ χρησιμοποιούνται για να επιτελούν εργασίες υψηλού κινδύνου ή δυσκολίας για τον άνθρωπο. Μπορούν επίσης να παράγουν έργο σε χαμηλότερο χρόνο και κόστος από ότι ο άνθρωπος πράγμα το οποίο τα καθιστά ιδανικά για αυτοματοποιημένες παραγωγές αλλά και άλλες εργασίες. Ανάλογα (Βικιπαίδεια, 2021).

2.2 Είδη – Κατηγορίες Ρομπότ

AMR (Autonomous Mobile Robots): Η αλλιώς αυτόνομα φορητά ρομπότ τα οποία έχουν την δυνατότητα να κινούνται εντός του περιβάλλοντος εργασίας τους αλλά και να λαμβάνουν αποφάσεις κατά την διάρκεια της λειτουργίας του σχεδόν σε πραγματικό χρόνο. Για την λήψη αυτών των αποφάσεων διαθέτουν κάμερες και αισθητήρια μέσω των οποίων λαμβάνουν πληροφορίες για τον χώρο εργασίας τους. Επιπροσθέτως διαθέτουν ενσωματωμένο επεξεργαστή ο οποίος βοηθά αρχικά στην ανάλυση τις εικόνες και των ερεθισμάτων που λαμβάνει το ρομπότ από το περιβάλλον εργασίας του και εν συνεχεία στην λήψη τεκμηριωμένων αποφάσεων όπως η μετακίνηση στον χώρο ενώ αποφεύγουν κάποιο φυσικό εμπόδιο, επιλογή του σωστού αντικειμένου. Αποτελούν λύσεις για εφαρμογές οι οποίες δεν απαιτούν υψηλή ανθρώπινη συνεισφορά για να επιτελεστούν (Intel, 2022).

AGV (Automated Guided Vehicle): Αυτά τα ρομπότ σε αντίθεση με τα αυτόνομα ρομπότ, λειτουργούν με προκαθορισμένες διαδρομές και στο πλείστον των φορών κρίνεται απαραίτητη η επίβλεψη από τον χειριστή. Αυτή η κατηγορία ρομπότ είναι κατάλληλη για ελεγχόμενα περιβάλλοντα όπως είναι οι αποθήκες, εργοστάσια, εταιρίες μεταφορών και χρησιμοποιούνται για την μετακίνηση αντικειμένων εντός του χώρου (Intel, 2022).

Humanoids: Είναι τα λεγόμενα ανθρωπόμορφα ρομπότ, τα οποία είναι σχεδιασμένα για να εξυπηρετούν λειτουργικούς κατά κύριο λόγο σκοπούς, όπως είναι η εκτέλεση ανθρωποκεντρικών εργασιών. Αυτά τα ρομπότ, όπως και τα AMR, χρησιμοποιούν αισθητήρια διαφόρων ειδών, ανάλογα με την λειτουργία την οποία είναι κατασκευασμένα να υλοποιούν. Από τα αισθητήρια λαμβάνουν ερεθίσματα από το περιβάλλον εργασίας τα οποία επεξεργάζονται κάνοντας χρήση τεχνητής νοημοσύνης και τέλος λαμβάνουν αποφάσεις για την καλύτερη δυνατή υλοποίηση της εργασίας τους.

Αυτά μπορούν να αντικαταστήσουν το ανθρώπινο δυναμικό σε ένα τεράστιο εύρος εφαρμογών όπως εργασίες επικίνδυνες ή επίπονες για το ανθρώπινο σώμα (π.χ. Συντηρήσεις ή επιδιορθώσεις βλαβών σε τοξικό περιβάλλον, αντιμετώπιση καταστροφών), διεξαγωγές ερευνητικών πειραμάτων π.χ. Διάστημα, ιατρικές εφαρμογές π.χ. ανάπτυξη προσθετικών άκρων ακόμα και διασκέδαση π.χ. θεματικά πάρκα (Intel, 2022) (Humanoid Robot, 2022) (Humanoid Robots, 2022).

Cobots: Τα Cobots είναι σχεδιασμένα να συνεργάζονται με τους ανθρώπους. Σε αντίθεση με τους υπόλοιπους τύπους ρομπότ τα οποία εκτελούν τις εργασίες τους είτε αυτόνομα, χωρίς δηλαδή την παρουσία ανθρώπινου δυναμικού, είτε σε αυστηρά απομονωμένους χώρους εργασίας. Αυτά τα ρομπότ έχουν την δυνατότητα να μοιράζονται τον χώρο εργασίας τους με τους εργαζόμενους για να τους βοηθούν στην υλοποίηση των εργασιών τους. Μερικά παραδείγματα εφαρμογών που μπορεί να χρησιμοποιηθεί αυτή η κατηγορία ρομπότ είναι η μεταφορά υλικών εντός κτηρίων, αυτοματοποιημένες μη εργονομικές διαδικασίες όπως η μεταφορά βαριών αντικειμένων, εργασίες τροφοδοσίας ή συναρμολόγηση-συντήρηση μηχανών (Intel, 2022) (Cobot, 2022).

Hybrids: Αυτού του είδους τα ρομπότ δημιουργούνται κάνοντας συνδυασμούς των προηγούμενων τύπων ρομπότ ώστε να δημιουργηθούν υβριδικά μοντέλα τα οποία είναι ικανά για πιο σύνθετες και εξεζητημένες εργασίες. Ένα χαρακτηριστικό παράδειγμα αυτής της κατηγορίας ρομπότ είναι ο συνδυασμός ενός AMR με έναν ρομποτικό βραχίονα με το οποίο μπορεί να επιτευχθεί ο χειρισμός δεμάτων ή αντικειμένων σε περιβάλλοντα όπως ένα εργοστάσιο ή μια αποθήκη (Intel, 2022).

2.3 Είδη – Κατηγορίες Ρομποτικών Βραχιόνων

Articulated : Αυτό το είδος ρομπότ αποτελείται από περιστροφικούς συνδέσμους των οποίων οι δομή μπορεί να είναι απλή και να αποτελείται από 2 έως 10 αρθρώσεις ή και περισσότερες σε πιο σύνθετες δομές. Η σύνδεση του συστήματος του βραχίονα με την βάση του γίνεται με μια περιστρεφόμενη άρθρωση, ενώ η σύνδεση μεταξύ των συνδέσμων του βραχίονα γίνεται με περιστρεφόμενες αρθρώσεις. Κάθε άρθρωση ονομάζεται άξονας και παρέχει έναν επιπλέον βαθμό ελευθερίας ή εύρος κίνησης. Ο μέσος αριθμός αξόνων ενός ρομπότ βιομηχανικού συστήματος αυτοματισμού είναι τέσσερις ή έξι (Robots, 2020).

Cartesian : Τα ρομπότ αυτής της κατηγορίας ονομάζονται και ευθύγραμμα. Αυτά αποτελούνται από τρεις αρθρώσεις γραμμικής κίνησης οι οποίες χρησιμοποιούν το σύστημα καρτεσιανών συντεταγμένων (X,Y,Z). Μπορεί επίσης να διαθέτουν προσαρμοσμένο καρπό ο οποίος να επιτρέπει την περιστροφική του κίνηση (Robots, 2020).

Cylindrical : Το είδος αυτών των ρομπότ αποτελείται από έναν, τουλάχιστον, περιστροφικό σύνδεσμο στην βάση και έναν, τουλάχιστον, πρισματικό σύνδεσμο για την σύνδεση μεταξύ των συνδέσμων. Η περιστροφική άρθρωση πραγματοποιεί κατά μήκος του άξονά της μία περιστροφική κίνηση, ενώ η πρισματική άρθρωση κινείται γραμμικά. Αυτά τα ρομπότ συνήθως τοποθετούνται σε κυλινδρικά περιβάλλοντα εργασίας (Robots, 2020).

Polar : Η αλλιώς σφαιρικά ρομπότ, η διαμόρφωση των οποίων πραγματοποιείται με την σύνδεση της βάσης και του βραχίονα, από μία περιστρεφόμενη άρθρωση, και τον συνδυασμό μιας γραμμικής άρθρωσης και δύο περιστροφικών αρθρώσεων. Κατ' αυτόν τον τρόπο οι άξονες σχηματίζουν ένα σύστημα πολικών συντεταγμένων (Robots, 2020).

SCARA : Το βασικό χαρακτηριστικό αυτού του είδους ρομπότ είναι ότι είναι άκαμπτα ως προς την κατεύθυνση του άξονα Z και ότι η διάταξη των βραχιόνων πραγματοποιείται με 2 ή παραπάνω συνδέσμους παρόμοιους με αυτούς των ανθρώπινων βραχιόνων. Κινείται συνήθως σε κυκλικό περιβάλλον και χρησιμοποιείται για εφαρμογές ρομποτικής συναρμολόγησης (Robots, 2020) (Wikipedia, SCARA, 2021).

Delta : Τα ρομπότ αυτού του τύπου μοιάζουν λειτουργικά, με αράχνη. Η κατασκευή τους αποτελείται από μια βάση επάνω στην οποία συνδέονται με αρθρώσεις 3 παραλληλόγραμμοι βραχίονες που κινούνται με συντονισμένο τρόπο. Οι παραλληλόγραμμοι βραχίονες κινούν το άκρο εργασίας (του ρομπότ). Βασικό πλεονέκτημα αυτή της κατηγορίας ρομπότ είναι η ταχύτητα και η ακρίβεια κίνησης. Οι βασικές τους εφαρμογές είναι σε βιομηχανίες φαρμάκων, ηλεκτρονικών, αλλά και τροφίμων (Robots, 2020).

2.3 Ρομποτικός Βραχίονας SCARA

Το ακρωνύμιο SCARA σημαίνει βραχίονας ρομπότ επιλεκτικής συμμόρφωσης ή αρθρωτός βραχίονας ρομπότ επιλεκτικής συμμόρφωσης. Το Scara είναι ένα ρομπότ βιομηχανικού τύπου. Όπως αναφέρεται παραπάνω βασικό χαρακτηριστικό αυτού του είδους ρομπότ είναι ότι είναι άκαμπτα ως προς την κατεύθυνση του άξονα Z και ότι η διάταξη των βραχιόνων πραγματοποιείται με 2 ή παραπάνω συνδέσμους παρόμοιους με αυτούς των ανθρώπινων βραχιόνων. Δηλαδή ένα ρομπότ Scara τεσσάρων αξόνων αποτελείται αρχικά από έναν σύνδεσμο εσωτερικού τύπου ο οποίος συνδέει την βάση του ρομπότ με τον πρώτο βραχίονα και περιστρέφεται γύρω από τον άξονα Z, στην συνέχεια ο πρώτος βραχίονας συνδέεται μέσω ενός εξωτερικού συνδέσμου με τον δεύτερο βραχίονα που περιστρέφεται γύρω από μια άρθρωση Z, ο οποίος με τη σειρά του συνδέεται με έναν άξονα περιστροφικής κίνησης ο οποίος έχει την δυνατότητα να κινείται και πάνω-κάτω. Αυτό λειτουργεί συνήθως σε κυκλικό περιβάλλον και χρησιμοποιείται για εφαρμογές ρομποτικής συναρμολόγησης (Wikipedia, SCARA, 2021) (FlexiBowl, 2020).

2.4 Κινηματική Ορθή και Αντίστροφη

Ορθή Κινηματική

Η ορθή κινηματική αναφέρεται στη χρήση των κινηματικών εξισώσεων ενός ρομπότ για τον υπολογισμό της θέσης του τελικού τελεστή από καθορισμένες τιμές για τις παραμέτρους της άρθρωσης.

Η αλλιώς, δεδομένης μια κινηματικής αλυσίδας που αποτελείται από συνδέσμους και αρθρώσεις με πολλαπλούς βαθμούς ελευθερίας, ορθή κινηματική ορίζεται ως η εύρεση της θέσης και του προσανατολισμού του τελικού τελεστή στον επιχειρησιακό χώρο εργασίας όταν είναι γνωστές όλες οι παράμετροι της άρθρωσης (Rosroboticslearning, 2021).

Αντίστροφη Κινηματική

Στην ρομποτική, η αντίστροφη κινηματική είναι διαδικασία μαθηματικών υπολογισμών των μεταβλητών παραμέτρων άρθρωσης που απαιτούνται για να μετακινηθεί το άκρο μιας κινηματικής αλυσίδας, όπως ένα ρομπότ, σε μια δεδομένη θέση και προσανατολισμό σε σχέση με το αρχικό σημείο 0. Χρησιμοποιείται για το λεγόμενο motion planning το οποίο είναι ο προσδιορισμός της κίνησης ενός ρομπότ έτσι ώστε το άκρο εργασίας του να μετακινηθεί από την αρχική θέση (Σημείο 0) στην επιθυμητή (Wikipedia, Inverse Kinematics, 2022).

2.5 Προσομοίωση

Προσομοίωση ρομποτικής είναι ένα εικονικό περιβάλλον όπου ο χειριστής μπορεί να σχεδιάσει, να ελέγχει αλλά και να αλληλοεπιδρά με ρομποτικές μηχανές μέσω της χρήσης προγραμματισμού σε περιβάλλον υπολογιστή (Learning, 2021).

2.6 Matlab

Το MATLAB είναι μια γλώσσα προγραμματισμού πολλαπλών χρήσεων και ένα περιβάλλον κατάλληλο για αριθμητικούς υπολογισμούς που αναπτύχθηκε από τη MathWorks. Το MATLAB δίνει την δυνατότητα χειρισμού πινάκων, σχεδίαση συναρτήσεων και δεδομένων, υλοποίηση αλγορίθμων, δημιουργία διεπαφών χρήστη και διασύνδεση με προγράμματα γραμμένα σε άλλες γλώσσες (WikiPedia, 2019).

2.7 Processing

Η Processing αποτελεί μια γλώσσα προγραμματισμού ανοικτού κώδικα και παράλληλα ένα προγραμματιστικό περιβάλλον το οποίο απευθύνεται σε χρήστες που θέλουν να προγραμματίσουν εικόνες, animation και ήχο. Ξεκίνησε ως εκπαιδευτικό εργαλείο και εξελίχθηκε σε ένα επαγγελματικό προγραμματιστικό περιβάλλον (Wikipedia, Proscessing, 2021).

3. ΚΑΘΟΡΙΣΜΟΣ ΠΡΟΔΙΑΓΡΑΦΩΝ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ

Σε αυτό το κεφάλαιο της διεκπεραίωσης της παρούσης διπλωματικής εργασίας, θα εξηγηθεί και θα αναλυθεί εκτενέστερα η διαδικασία η οποία ακολουθήθηκε για να επιτευχθεί ο καθορισμός των προδιαγραφών κατασκευής του ρομποτικού βραχίονα. Αυτή χωρίστηκε σε δύο βασικές εργασίες. Η πρώτη ήταν η έρευνα αγοράς, η αναζήτηση στο διαδίκτυο καθώς και σε έντυπα επιστημονικού περιεχομένου και η καταγραφή των δεδομένων αυτών. Στο δεύτερο μέρος έγινε η μελέτη αυτών και από αυτήν προέκυψε ο καθορισμός των τελικών υλικών, μηχανολογικών και ηλεκτρολογικών προδιαγραφών του ρομπότ.

Αρχικά πραγματοποιήθηκε εκτεταμένη έρευνα στο διαδίκτυο, σε βάσεις επιστημονικών πηγών, ερευνητικών ιστοσελίδων, καθώς και εταιριών κατασκευής βιομηχανικών ρομποτικών βραχιόνων όπως επίσης και σε επιστημονικού/ερευνητικού περιεχομένου περιοδικά αλλά και βιβλία εκπαιδευτικής ρομποτικής. Από αυτά συγκεντρώθηκαν, σε πρώτο στάδιο, πληροφορίες γενικού περιεχομένου με σκοπό να κατανοηθεί η κατασκευή και η λειτουργία ενός ρομπότ, και στην συνέχεια έγινε εμβάθυνση στην κατηγορία των βιομηχανικών ρομποτικών βραχιόνων. Σε αυτήν την κατηγορία αρχικά μελετήθηκε ο τρόπος λειτουργίας ενός βραχίονα σε συνάρτηση με το περιβάλλον και τις συνθήκες εργασίας του. Επιπροσθέτως πραγματοποιήθηκαν εκτενέστερες αναζητήσεις επάνω στους ρομποτικούς βραχίονες τύπου Scara, ψάχνοντας στο διαδίκτυο σε ιστοσελίδες εταιριών παραγωγής βιομηχανικών ρομπότ όπως είναι η Fanuc, η ABB, η Kuka και άλλες. Από αυτές συλλέχθηκαν τα απαραίτητα δεδομένα ώστε να πραγματοποιηθεί ο καθορισμός των τελικών προδιαγραφών του ρομποτικού βραχίονα αλλά και της λειτουργίας της οποίας θα είναι ικανός να επιτελέσει.

Πραγματοποιήθηκε η ανάλογη έρευνα στην αγορά, κατά κύριο λόγο του Ελλαδικού χώρου αλλά και του εξωτερικού, ώστε να υπάρξει επίγνωση της διαθεσιμότητας των υλικών σε συνάρτηση με την ταχύτητα παράδοσης τους, και του κόστους που θα είχαν. Χρησιμοποιώντας τα δεδομένα που συλλέχθηκαν από τις προηγούμενες αναζητήσεις, αυτά τις παραπάνω έρευνας και λαμβάνοντας υπ' όψη, τις κατασκευαστικές δυνατότητες αλλά και του οικονομικού κεφαλαίου που υπήρχε, πάρθηκε η απόφαση : τα βασικότερα μέρη του ρομποτικού βραχίονα να κατασκευαστούν από πλαστικό. Αυτό υλοποιήθηκε κάνοντας χρήση ενός 3D Printer το οποίο εξυπηρέτησε αρχικά στο κόστος της κατασκευής, το οποίο ήταν χαμηλότερο σε σχέση με την επιλογή κάποιου μετάλλου ως πρώτο υλικό όπως επίσης και πιο εύκολο να επεξεργαστεί με εργαλεία χειρός, πράγμα το οποίο χρειάστηκε για την αφαίρεση ατελειών, τις τρισδιάστατης εκτύπωσης, κατά την διάρκεια της συναρμολόγησης. Επιπλέον το πλαστικό είναι ένα ελαφρύ υλικό το οποίο καθιστά την λειτουργία του ρομποτικού βραχίονα ευκολότερη και λιγότερο απαιτητική, από θέμα ακρίβειας ανοχών, δυνάμεων των βηματικών κινητήρων για την κίνηση των αρθρώσεων, αλλά εξυπηρετεί και στην ευκολότερη μεταφορά του μετά την συναρμολόγηση.

Στην συνέχεια μετά την επιλογή του βασικού υλικού, καθορίστηκαν οι διαστάσεις των τμημάτων του ρομποτικού βραχίονα, ώστε να μπορεί να γίνει ο υπολογισμός του μεγέθους τους και του βάρους τους κατά προσέγγιση.

Αυτό βοήθησε στην αποτελεσματικότερη επιλογή βηματικών κινητήρων ώστε να έχουν την δύναμη και την ροπή στρέψης που θα χρειαζόνταν για να κινηθεί ο βραχίονας. Επιπλέον βάση των βηματικών κινητήρων και των χαρακτηριστικών τους έγινε και η επιλογή του ανάλογου μηχανολογικού εξοπλισμού για να επιτευχθεί η μετάδοση κίνησης στις αρθρώσεις του ρομποτικού βραχίονα χωρίς απώλειες ή αστοχίες.

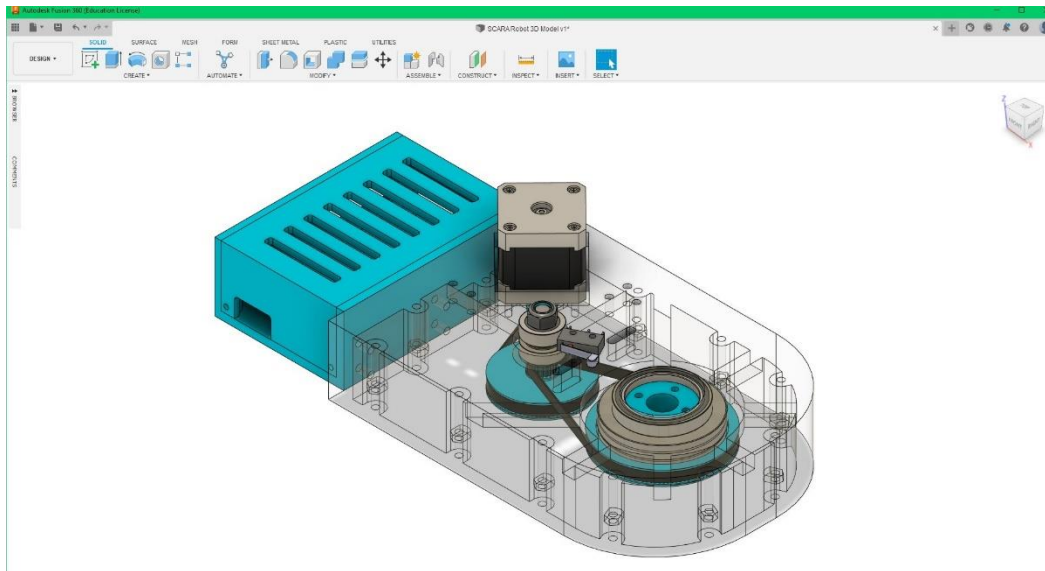
Βάση του τύπου των κινητήρων που επιλέχθηκε όμως έγινε και ο καθορισμός του κατάλληλου ηλεκτρολογικού κυκλώματος καθώς και των εξαρτημάτων που χρειάστηκαν για τον έλεγχο, την οδήγηση τους και την τροφοδοσία τους, αλλά και για την λειτουργία του ρομποτικού βραχίονα.

Τέλος, βάση αυτών των υλικών, μηχανολογικών και ηλεκτρολογικών προδιαγραφών οι οποίες καθορίστηκαν από την διαδικασία που περιεγράφηκε στο συγκεκριμένο κεφάλαιο, υλοποιήθηκαν τα επόμενα βήματα της διπλωματικής εργασίας.

4. ΣΧΕΔΙΑΣΜΟΣ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ

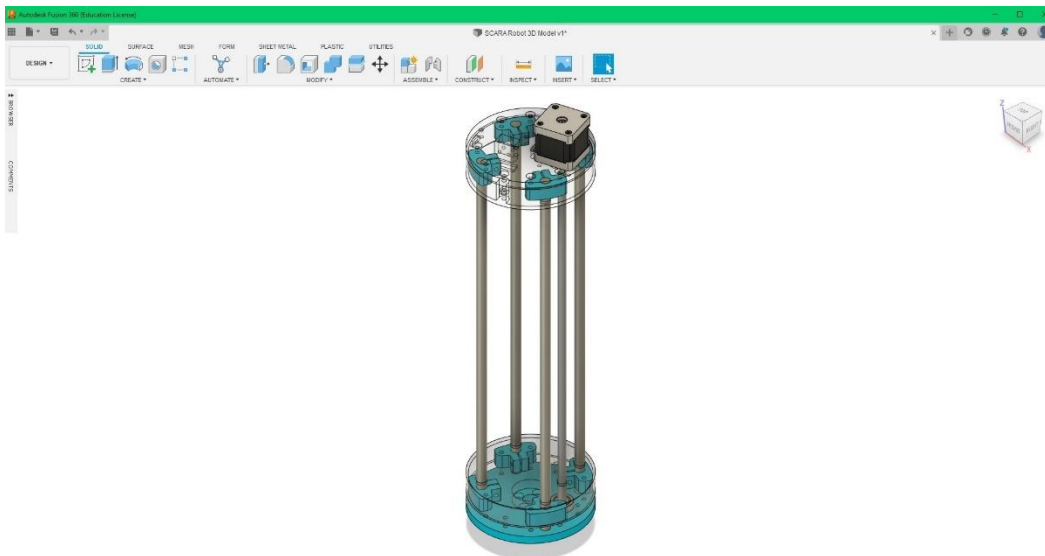
Στην συγκεκριμένη ενότητα θα γίνει περιγραφή της διαδικασίας του τρισδιάστατου σχεδιασμού του ρομποτικού βραχίονα καθώς επίσης θα διευκρινιστούν εις βάθος τα βήματα τα οποία ακολουθήθηκαν στο σχεδιαστικό λογισμικό που χρησιμοποιήθηκε για την τελική υλοποίηση του σχεδίου βραχίονα. Λαμβάνοντας υπ' όψη τα δεδομένα των προδιαγραφών που καθορίστηκαν στο προηγούμενο βήμα, ο σχεδιασμός του ρομποτικού βραχίονα χωρίστηκε σε πέντε βασικά μέρη. Αυτά είναι ήταν η βάση, ο άξονας Z, ο πρώτος σύνδεσμος μεταξύ άξονα Z και της 2^{ης} άρθρωσης, ο δεύτερος σύνδεσμος μεταξύ της 2^{ης} και της 3^{ης} άρθρωσης, το άκρο εργασίας και ο χώρος στο οποίο θα τοποθετούνταν, ο μικροελεγκτής αλλά και ηλεκτρονικά εξαρτήματα οδήγησης και τροφοδοσίας των βηματικών κινητήρων.

Αρχικά η σχεδίαση ξεκίνησε με την βάση του ρομποτικού βραχίονα. Για να γίνει αυτό λήφθηκε υπόψιν αρχικά το βάρος του άξονα Z και των συνδέσμων που θα στηρίζονταν επάνω του, ο χώρος ο οποίος θα χρειαζόνταν για τον βηματικό κινητήρα, και την μετάδοση της κίνησης από αυτόν στην άρθρωση. Επιπλέον, για την σωστή μετάδοση υπολογίστηκε η δύναμη η οποία χρειάζεται ο κινητήρας για να περιστρέψει τον άξονα αυτόν και κατά συνέπεια όλον τον βραχίονα, και σύμφωνα με αυτόν τον υπολογισμό υλοποιήθηκε μια μείωση βήματος 20:1. Αυτή έγινε σε δύο στάδια, χρησιμοποιήθηκαν δύο ιμάντες, ενδιάμεσα από τους οποίους υπήρχε ένα διπλό γρανάζι. Στο πρώτο στάδιο γίνεται η μετάδοση της κίνησης από τον κινητήρα στο ενδιάμεσο γρανάζι-μειωτήρα και στο δεύτερο στάδιο γίνεται η μετάδοση της κίνησης από αυτό στο γρανάζι της άρθρωσης. Για την στήριξη του ενδιάμεσου γραναζιού σχεδιάστηκε ένα σύστημα στο οποίο χρησιμοποιήθηκε μια βίδα η οποία, περιστρέφεται εντός δύο ομόκεντρων κυκλικών ρουλεμάν με σκοπό την ελαχιστοποίηση των τριβών, ενώ για το γρανάζι της άρθρωσης το οποίο είναι άμεσα συνδεδεμένο με αυτήν και περιστρέφεται ομοίως εντός δύο ομόκεντρων ρουλεμάν, αλλά για την σωστή στήριξη του σχεδιάστηκε με τρόπο τέτοιο ώστε να στηρίζεται επάνω σε ένα ωστικό ρουλεμάν. Τέλος, έγινε ο σχεδιασμός της βάσης του βηματικού μοτέρ καθώς και οι τρύπες για την στήριξη: του τερματικού διακόπτη της άρθρωσης, για την στήριξη του μοτέρ, για την πιθανή χρήση τεντοτήρα στους ιμάντες αλλά και για την πάκτωση της βάσης.



Εικόνα 1: Σχεδίαση Βάσης Ρομποτικού Βραχίονα.

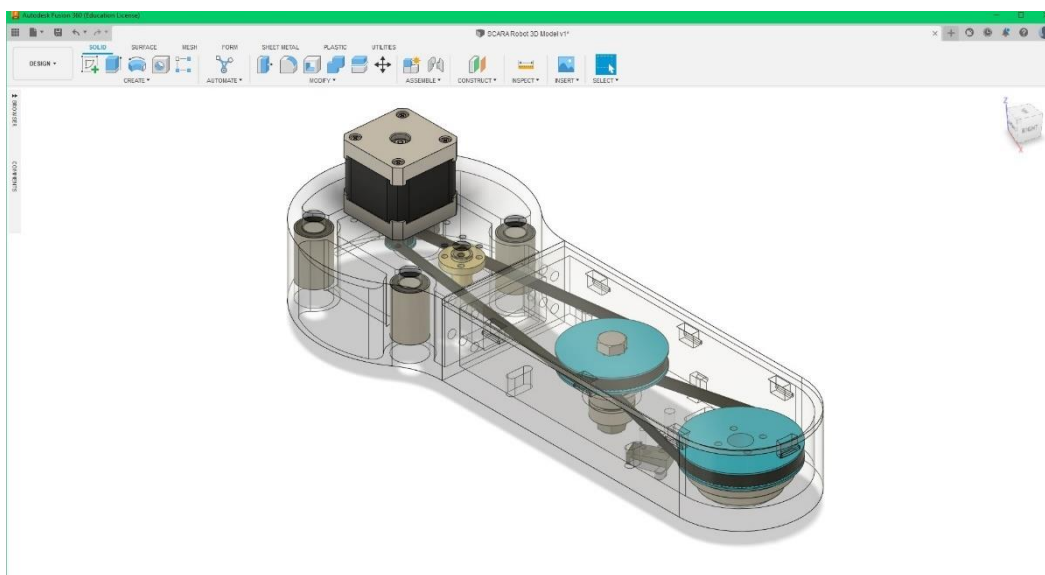
Στο επόμενο βήμα ακολούθησε η σχεδίαση της πρώτης άρθρωσης και του άξονα Z ο οποίος θα στηριζόταν επάνω της. Η βάση της άρθρωσης αυτής είναι ένας δίσκος, της οποίας το κάτω άκρο είναι άμεσα συνδεδεμένο με το γρανάτζι της, και στηρίζεται σε ένα δεύτερο ωστικό ρουλεμάν. Η κίνηση επί του άξονα Z, αποφασίστηκε να γίνει με μεταλλικές ράβδους. Για να μπορεί να γίνει η σύνδεση τους με την άρθρωση σχεδιάστηκε μια κυκλική πλάκα όμοιας διαμέτρου με του δίσκου, επάνω στην οποία υπάρχουν οι κατάλληλες τρύπες για τις ράβδους. Επιπλέον, σχεδιάστηκαν οι σφικτήρες στήριξης των αξόνων, ώστε να γίνει στιβαρότερο το δέσιμο τους, καθώς επίσης και η θέση για την τοποθέτηση ένα κυκλικού ρουλεμάν για την στήριξη της ντίζας κίνησης του μπράτσου του βραχίονα επί του άξονα Z. Τέλος, προστέθηκε ένα κάλυμμα για προστατευτικούς και αισθητικούς λόγους επάνω στην άρθρωση.



Εικόνα 2: Σχεδίαση Άξονα Z Ρομποτικού Βραχίονα.

Αφού ολοκληρώθηκε η σχεδίαση της 1^{ης} άρθρωσης του ρομποτικού βραχίονα και το κάτω μέρος του άξονα Z στην συνέχεια ακολούθησε η βάση του 1^{ου} συνδέσμου, υπεύθυνου για την ένωση του άξονα Z με την 2^η άρθρωση, καθώς και το επάνω μέρος, η κορυφή δηλαδή, του ρομποτικού βραχίονα. Η βάση αυτή διαθέτει 4 θύρες για την τοποθέτηση γραμμικών κυλινδρικών ρουλεμάν τα οποία κινούνται επί των μεταλλικών ράβδων του άξονα Z που προαναφέρθηκαν. Επίσης, σχεδιάστηκε ώστε να ενώνεται με βίδες με τον 1^ο σύνδεσμο και η μετάδοση κίνησης από τον βηματικό κινητήρα προς την 2^η άρθρωση να είναι εφικτή με ιμάντα. Τέλος, τοποθετήθηκε το παξιμάδι οδήγησης της βάσης αλλά και ο βηματικός κινητήρας της δεύτερης άρθρωσης. Έπειτα ακολούθησε το επάνω μέρος του άξονα Z. Για αυτό όπως και για το κάτω άκρο, σχεδιάστηκε ένας δίσκος στον οποίο στηρίχθηκαν, οι μεταλλικοί ράβδοι, οι σφικτήρες όμοιοι με τους αυτούς του κάτω άκρου, ο βηματικός κινητήρας κίνησης της βάσης του πρώτου συνδέσμου επί του άξονα Z, καθώς και ο τερματικός διακόπτης.

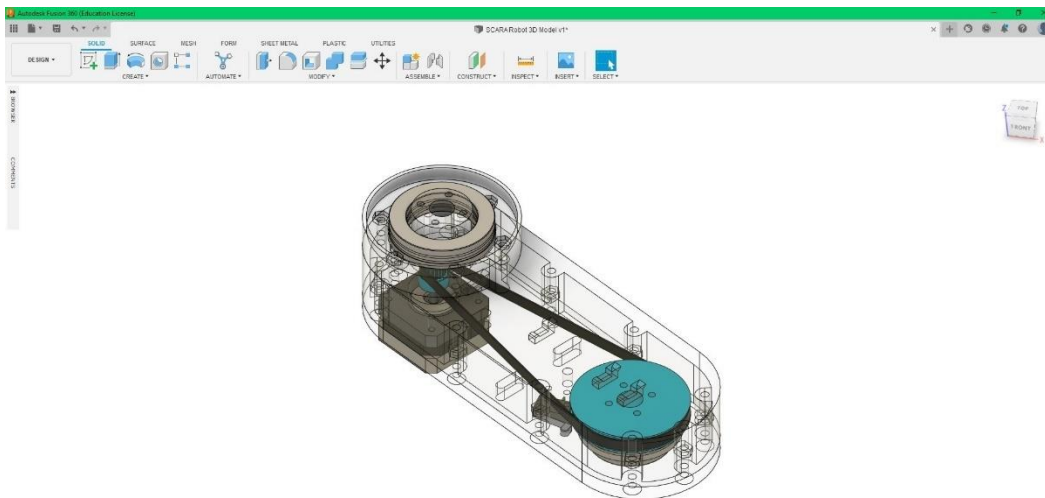
Την συνέχεια της διαδικασίας της τρισδιάστατης σχεδίασης, ακολούθησε το σχέδιο του πρώτου συνδέσμου και της 2^{ης} άρθρωσης του ρομποτικού βραχίονα. Εδώ σχεδιάστηκε ένα μακρόστενο παραλληλόγραμμο του οποίου το ένα άκρο έγινε ημικυκλικό για αισθητικούς λόγους. Η πρώτη πλευρά σχεδιάστηκε με τρόπο τέτοιο ώστε η ένωση μεταξύ του συνδέσμου και της βάσης του να γίνεται με την βοήθεια βιδών αλλά και να επιτρέπεται η μετάδοση κίνησης από το μοτέρ στην άρθρωση με την χρήση ιμάντα όπως και την βάση του ρομποτικού βραχίονα. Σε αυτήν την περίπτωση η δύναμη η οποία απαιτούσε ο βηματικός κινητήρας για την περιστροφή της άρθρωσης ήταν μικρότερη από αυτή που υπολογίστηκε για στην βάση του βραχίονα, οπότε υλοποιήθηκε μια μικρότερη μείωση βήματος 16:1. Αυτή σχεδιάστηκε με τον ίδιο τρόπο που ακολουθήθηκε για την 1^η άρθρωση, χρησιμοποιήθηκαν δηλαδή δύο ιμάντες και ένα ενδιάμεσο διπλό γρανάζι. Έπειτα το ημικυκλικό άκρο σχεδιάστηκε ώστε στο εσωτερικό μέρος του συνδέσμου να βρίσκεται το γρανάζι μετάδοσης κίνησης στην 2^η άρθρωση. Αυτό είναι άμεσα συνδεδεμένο με την άρθρωση, η στήριξη και η περιστροφή της οποίας υλοποιήθηκε ακολουθώντας το μοτίβο της 1^{ης}. Τέλος, προστέθηκαν κάποια γαντζάκια στο εσωτερικό του συνδέσμου για την τοποθέτηση των καλωδίων του ηλεκτρονικού κυκλώματος καθώς και ο τερματικός διακόπτης της 2^{ης} άρθρωσης.



Εικόνα 3: Σχεδίαση 1^{ου} Συνδέσμου Ρομποτικού Βραχίονα.

Μετά την ολοκλήρωση του πρώτου συνδέσμου του ρομποτικού βραχίονα ακολούθησε η σχεδίαση της 2^{ης} άρθρωσης και του 1^{ου} μέρους του 2^{ου} συνδέσμου. Όπως αναφέρθηκε στο προηγούμενο βήμα η στήριξη και η περιστροφή της άρθρωσης υλοποιήθηκε όπως αυτή της 1^{ης} άρθρωσης. Για να επιτευχθεί αυτό σχεδιάστηκε ένας κυκλικός δίσκος στο εσωτερικό του οποίου, δημιουργήθηκε ένα έδρανο ώστε να στηριχτεί-πατήσει το ωστικό ρουλεμάν καθώς και οι κατάλληλες τρύπες όπου θα έμπαιναν οι βίδες με τις οποίες θα βιδώνονταν η άρθρωση με το γρανάζι μετάδοσης κίνησης του 1^{ου} συνδέσμου αλλά και οι βίδες στήριξης της 2^{ης} άρθρωσης με τον 2^ο σύνδεσμο.

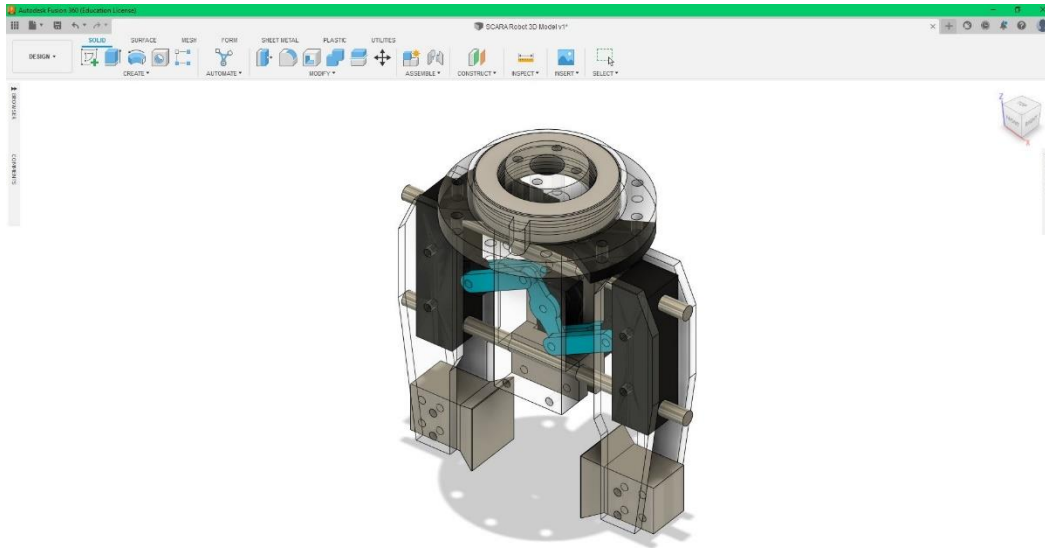
Το επόμενο βήμα της σχεδίασης ήταν ο 2^{ος} σύνδεσμος, ο οποίος χωρίστηκε σε δύο μέρη για την διευκόλυνση της συναρμολόγησης αλλά και της εκτύπωσης του, και ακριβώς όπως ο 1^{ος} σχεδιάστηκε αρχικά ως ένα ορθογώνιο παραλληλόγραμμο με την διαφορά ότι στην περίπτωση του 2^{ου} έγιναν και τα δύο του άκρα ημικυκλικά. Το πρώτο άκρο σχεδιάστηκε ώστε από την εξωτερική του μεριά να φιλοξενεί τον βηματικό κινητήρα της 3^{ης} άρθρωσης του ρομποτικού βραχίονα. Στο ενδιάμεσο από τα δύο άκρα του συνδέσμου, θα γίνονταν η μετάδοση κίνησης από τον βηματικό κινητήρα στην 4^η άρθρωση, αλλά αυτήν την φορά άμεσα δηλαδή χωρίς κάποιον μειωτήρα. Επίσης, όπως και στον προηγούμενο σύνδεσμο έτσι και σε αυτόν, δημιουργήθηκαν υποδοχές για σύστημα τεντοτήρα, όπως και για τερματικό διακόπτη άρθρωσης. Το άλλο άκρο σύμφωνα με τις προκαθορισμένες προδιαγραφές θα στηρίζει την 4^η και τελευταία άρθρωση του ρομποτικού βραχίονα. Η αρχιτεκτονική η οποία ακολουθήθηκε ήταν και σε αυτήν την περίπτωση ίδια με αυτή των προηγούμενων αρθρώσεων. Οπότε αφού ολοκληρώθηκε η σχεδίαση του 1^{ου} τμήματος του συνδέσμου, ακολούθησε το 2^ο το οποίο είχε το ίδιο σχήμα με το προηγούμενο ώστε να το καλύπτει, καθώς διαθέτει και τις κατάλληλες τρύπες ώστε το 1^ο να βιδωθεί επάνω του. Επίσης το 2^ο μέρος, στηρίζεται με τις κατάλληλες βίδες επάνω στην 2^η άρθρωση.



Εικόνα 4: Σχεδίαση 2^{ου} Συνδέσμου Ρομποτικού Βραχίονα.

Στο τελευταίο βήμα της σχεδίασης υλοποιήθηκε το άκρο εργασίας του ρομποτικού βραχίονα καθώς και ο μηχανισμός λειτουργίας του (αρπάγη). Το άκρο σχεδιάστηκε έτσι ώστε να φέρει δύο δαγκάνες, την μια απέναντι της άλλης οι οποίες ανοιγοκλείνουν μεταξύ τους με την βοήθεια ενός σέρβο. Αυτό βρίσκεται στο κέντρο του μηχανισμού της αρπάγης, η περιστροφή του οποίου κινεί τα δύο άκρα επάνω σε δύο παράλληλους μεταλλικούς άξονες. Στα άκρα αυτά στηρίζονται οι

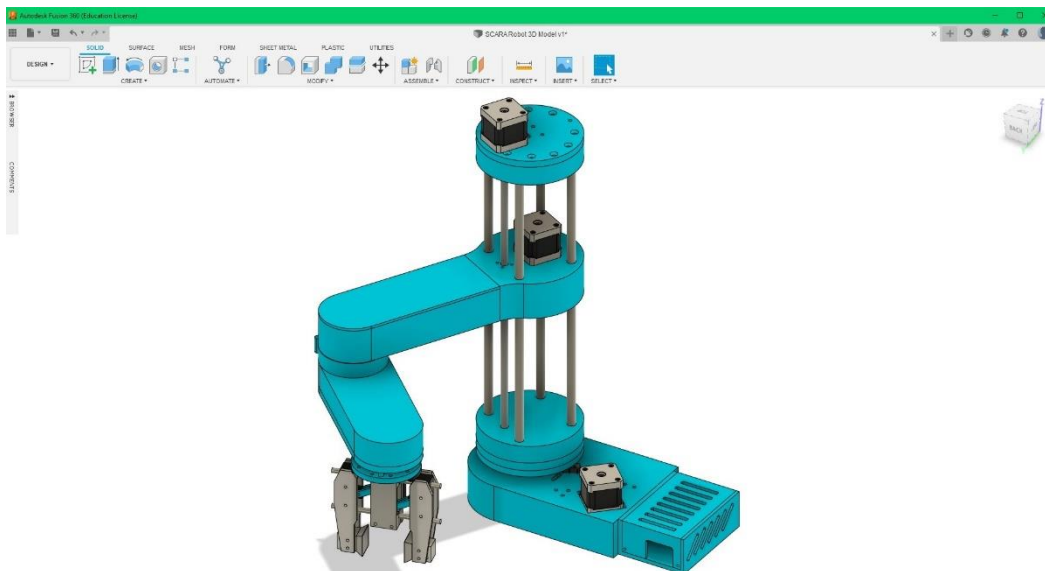
δαγκάνες του άκρου εργασίας, που προαναφέρθηκαν. Τέλος, το άκρο εργασίας συνδέεται άμεσα με την 4^η άρθρωση του ρομποτικού βραχίονα.



Εικόνα 5: Σχεδίαση Άκρου Εργασίας Ρομποτικού Βραχίονα.

Μηχανολογικό σχέδιο

Στο **μηχανολογικό σχέδιο** που ακολουθεί παρουσιάζονται όλα τα τμήματα του ρομποτικού βραχίονα μαζί ώστε να παρουσιαστεί το τελικό πρότυπο με όλα εξαρτήματα της κατασκευής.



Εικόνα 6: Μηχανολογικό Σχέδιο Ρομποτικού Βραχίονα.

6. ΤΡΙΣΔΙΑΣΤΑΤΗ ΕΚΤΥΠΩΣΗ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ

Στο κεφάλαιο αυτό θα επεξηγηθεί η διαδικασία που ακολουθήθηκε κατά την διάρκεια της τρισδιάστατης εκτύπωσης του ρομποτικού βραχίονα. Θα γίνει επίσης αναφορά, στο υλικό το οποίο χρησιμοποιήθηκε για την εκτύπωση των εξαρτημάτων του βραχίονα, στο είδος, και τις βασικές ρυθμίσεις αλλά και ιδιότητες του 3D Printer που χρειάστηκαν για να υλοποιηθεί η διαδικασία αυτή, καθώς επίσης και το software με το οποίο προγραμματίστηκε.

Για την διαδικασία της τρισδιάστατης εκτύπωσης του ρομποτικού βραχίονα χρησιμοποιήθηκε ένας Cartesian 3D Printer. Οι εκτυπωτές αυτοί χρησιμοποιούν το, γνωστό από τα μαθηματικά, καρτεσιανό σύστημα συντεταγμένων ώστε να μετακινούν την κεφαλή εκτύπωσης στην σωστή θέση στον χώρο εργασίας και να εναποθέτουν το λιωμένο πλαστικό νήμα στην επιθυμητή θέση. Ο εκτυπωτής που χρησιμοποιήθηκε για την συγκεκριμένη εφαρμογή, μετακινεί το θερμαινόμενο “τραπέζι” κατά τον άξονα Y ενώ η κεφαλή εκτύπωσης μετακινείται κατά τον άξονα X και Z. (The 4 Types of FDM 3D Printer Explained, 2022)

Το υλικό το οποίο χρησιμοποιήθηκε για την κατασκευή του ρομποτικού βραχίονα ήταν το PLA. Το πολυγαλακτικό οξύ, το οποίο είναι γνωστό και ως πολυλακτίδιο (PLA) είναι ένας θερμοπλαστικός αλειφατικός πολυεστέρας του οποίου η παραγωγή προέρχεται από ανανεώσιμες πηγές ενέργειας. Επίσης έχει χαμηλό σημείο τήξης, υψηλή αντοχή, χαμηλή θερμική διαστολή, ιδιότητες οι οποίες το καθιστούν ιδανικό υλικό για τους 3D printers και για αυτό είναι και το πιο ευρέως χρησιμοποιούμενο πλαστικό υλικό στις τρισδιάστατες εκτυπώσεις. (Polylactic acid, 2022)

Στην συνέχεια για να ξεκινήσει η διαδικασία της εκτύπωσης απαιτείται ένα λογισμικό (Slicer) το οποίο λαμβάνει ένα σύνολο πληροφοριών, οι οποίες προέρχονται από το αρχείο του σχεδίου που εξάγει το πρόγραμμα σχεδίασης, και τις μετατρέπει σε μια μορφή η οποία μπορεί να γίνει κατανοητή από τον εκτυπωτή. Σε αυτά συμπεριλαμβάνονται πληροφορίες όπως είναι, οι προδιαγραφές του υλικού, οι κατάλληλες θερμοκρασίες για την εκτύπωση του, το ύψος του κάθε στρώματος (layer height), κτλ. (Slicer in 3D Printing: Definition, Features, and How it Works, 2022)

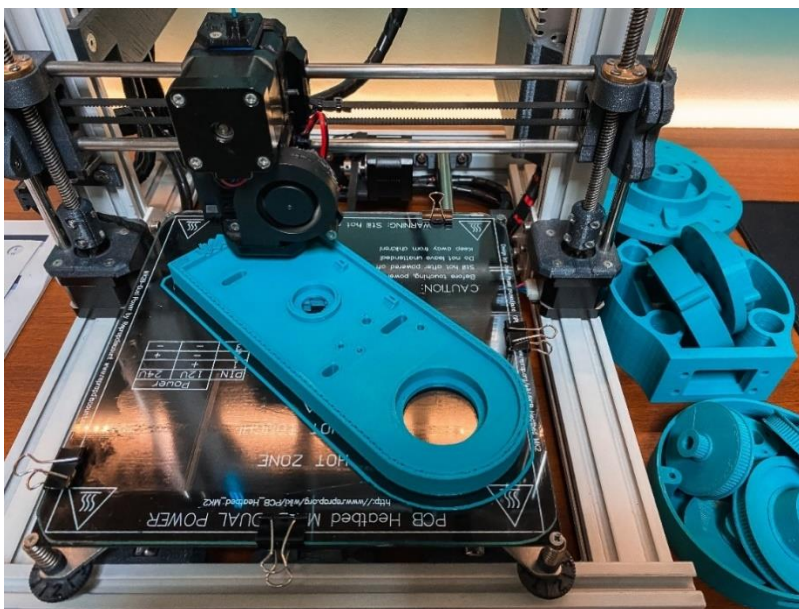
Στην συγκεκριμένη περίπτωση ως slicer χρησιμοποιήθηκε το Simplify3D. Σε αυτόν επιλέχθηκαν οι βασικές ρυθμίσεις της εκτύπωσης οι οποίες αναφέρθηκαν παραπάνω. Ενδεικτικά μερικές από αυτές ήταν :

Layer height : 0.20mm

Θερμοκρασία Εκτύπωσης : 210°C

Θερμοκρασία Επιφάνειας Εκτύπωσης : 50°C

Ταχύτητα Εκτύπωσης : 60 mm/s



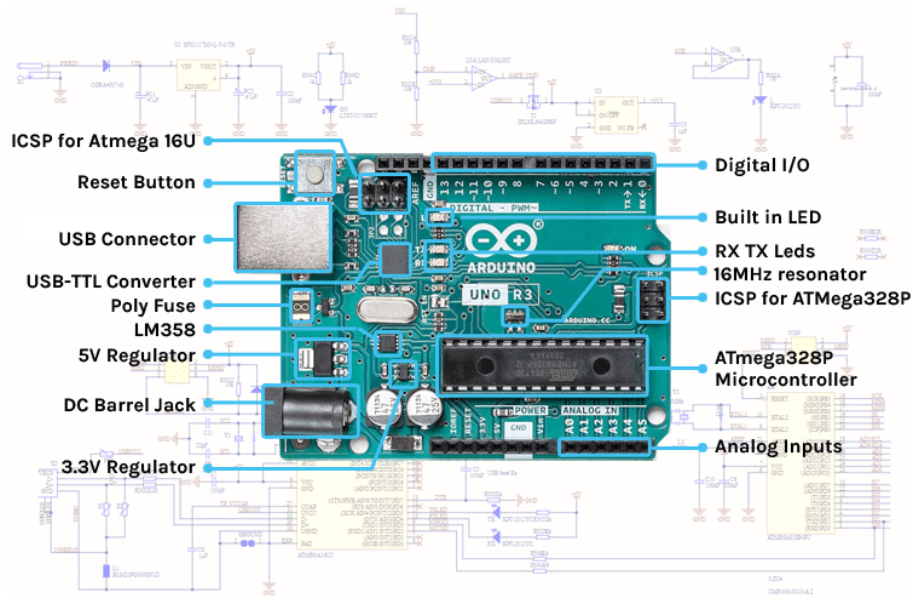
Εικόνα 7: Διαδικασία Τρισδιάστατης Εκτύπωσης.

7. ΗΛΕΚΤΡΟΝΙΚΟ ΚΥΚΛΩΜΑ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ

Στο επόμενο και 7^ο κεφάλαιο της παρούσης διπλωματικής εργασίας παρουσιάζεται το ηλεκτρονικό κύκλωμα το οποίο χρησιμοποιείται για την τροφοδοσία, την κίνηση και τον έλεγχο του ρομποτικού βραχίονα. Επιπροσθέτως, παρουσιάζονται μεμονωμένα τα επιμέρους ηλεκτρονικά εξαρτήματα και γίνεται αναφορά στα χαρακτηριστικά τους καθώς και επεξήγηση της λειτουργίας τους, ώστε να γίνει κατανοητή η χρήση τους στον ρομποτικό βραχίονα. Τέλος, φαίνεται ο σχεδιασμός του κυκλώματος ελέγχου της κατασκευής που υλοποιήθηκε με την βοήθεια του λογισμικού Fritzing καθώς και του Paint (Windows).

Μικροελεγκτής (Arduino Uno R3)

Το Arduino Uno είναι ένας μικροελεγκτής, ανοιχτού κώδικα, η πλακέτα του οποίου φέρει έναν (8-bit) μικροεπεξεργαστή, τον ATmega328P. Η πλακέτα αυτή διαθέτει ένα σύνολο ψηφιακών και αναλογικών εισόδων-εξόδων, οι οποίες μπορούν να δεχθούν ή να δώσουν σήματα εισόδου ή εξόδου αντίστοιχα. Εκτενέστερα διαθέτει 14 ψηφιακές θύρες εισόδου/εξόδου, εκ των οποίων οι 6 διαθέτουν PWM (Power Width Modulation), καθώς επίσης και 6 αναλογικές θύρες εισόδου/εξόδου οι οποίες προγραμματίζονται μέσω του Arduino IDE. Η τροφοδοσία γίνεται είτε μέσω καλωδίου USB είτε από εξωτερικό τροφοδοτικό με τάση 9-20V. (Arduino Uno, 2022)

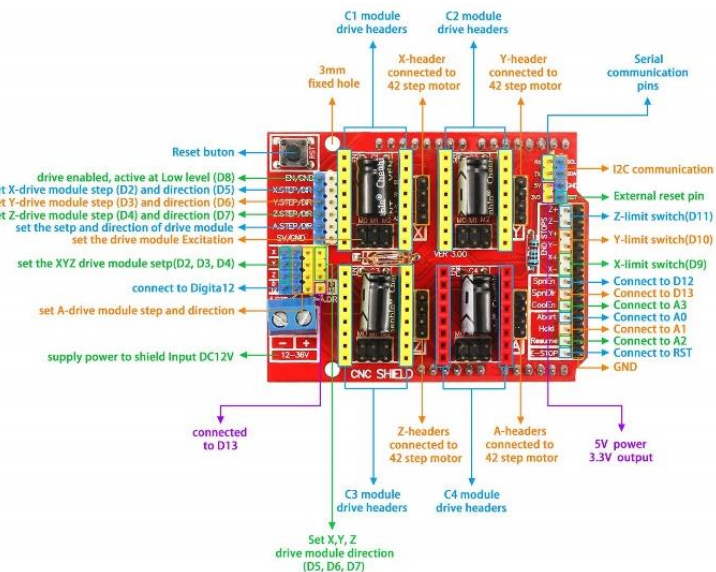


Εικόνα 8: Arduino Uno R3 Board.

(Everything you need to know about the Arduino Hardware, 2022)

CNC Shield V3

Το CNC Shield αποτελεί μια πλακέτα επέκτασης για το Arduino Uno η οποία είναι ικανή να ελέγχει έως και τέσσερις βηματικούς κινητήρες ταυτόχρονα με την βοήθεια οδηγών βηματικών κινητήρων (Stepper Drivers).



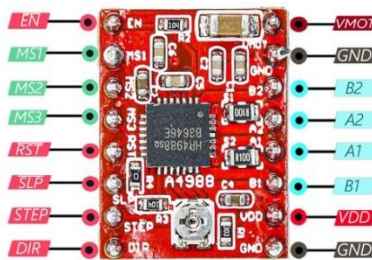
Εικόνα 9 : CNC Shield V3.

(CNC V3.0 PWM pins, 2022; Interfacing A4988 Stepper Motor Driver with Arduino, 2021)

Είναι κατάλληλα σχεδιασμένη έτσι ώστε να δέχεται τέσσερις τέτοιους οδηγούς, και επιτρέπει την εύκολη διαμόρφωση των παραμέτρων των μοτέρ με σκοπό το έλεγχο κίνησής τους. Αυτό την καθιστά κατάλληλη για διάφορες μηχανές οι οποίες λειτουργούν σε άξονες όπως είναι: η φρέζα CNC, η χάραξη/κοπή με Laser, μηχανές σχεδιασμού και 3D Printers. Επιπροσθέτως διαθέτει και άκρα στα οποία μπορούν να συνδεθούν τερματικοί διακόπτες για την οριοθέτηση της κίνησης (CNC Shield, 2019).

Stepper Drivers (A4988)

Ο μικροελεγκτής (Arduino Uno) ο οποίος χρησιμοποιείται στην συγκεκριμένη εφαρμογή δεν είναι ικανός να παρέχει αρκετό ρεύμα ώστε να επιτευχθεί η ορθή λειτουργία των βηματικών κινητήρων. Για τον λόγο αυτό χρησιμοποιούνται τα Stepper Driver. Αυτά αποτελούνται από μια πλακέτα και ένα τσιπ το οποίο επιτρέπει την απομόνωση της παροχής ισχύος στους κινητήρες από αυτήν στο Arduino. Σε συνδυασμό με μικροελεγκτή, έχουν την δυνατότητα να μπορούν να πετύχουν υψηλές ταχύτητες περιστροφής του κινητήρα ελέγχοντας ξεχωριστά το κάθε του πηνίο. Τα Stepper Driver που χρησιμοποιήθηκαν για τον ρομποτικό βραχίονα είναι τα A4988, είναι ικανά να παρέχουν έως και 2A συνεχούς ρεύματος στα πηνία των κινητήρων, το οποίο σε συνδυασμό με το χαμηλό τους κόστος και το μεγάλο εύρος εφαρμογών που έχουν, τα καθιστά ιδανικά (Stepper Motor Driver, 2018).



Εικόνα 10 : Stepper Driver (A4988).

(Interfacing A4988 Stepper Motor Driver with Arduino, 2021)

Limit Switch

Το Limit Switch είναι μια ηλεκτρομηχανική συσκευή, η οποία λειτουργεί σε συνδυασμό είτε με την κίνηση ενός μηχανικού μέρους είτε την παρουσία κάποιου αντικειμένου. Αυτή η συσκευή δεν είναι άλλο παρά ένας διακόπτης ο οποίος στο πλείστον των εφαρμογών χρησιμοποιείται για τον καθορισμό του ορίου της διαδρομής του μηχανικού μέρους, από όπου και παίρνει το όνομα του (Limit Switch). Στο ρομποτικό βραχίονα χρησιμοποιείται ένας τέτοιος σε κάθε άρθρωση ώστε να ορίζεται το σημείο 0 του. (Limit Switches Explained, 2020)

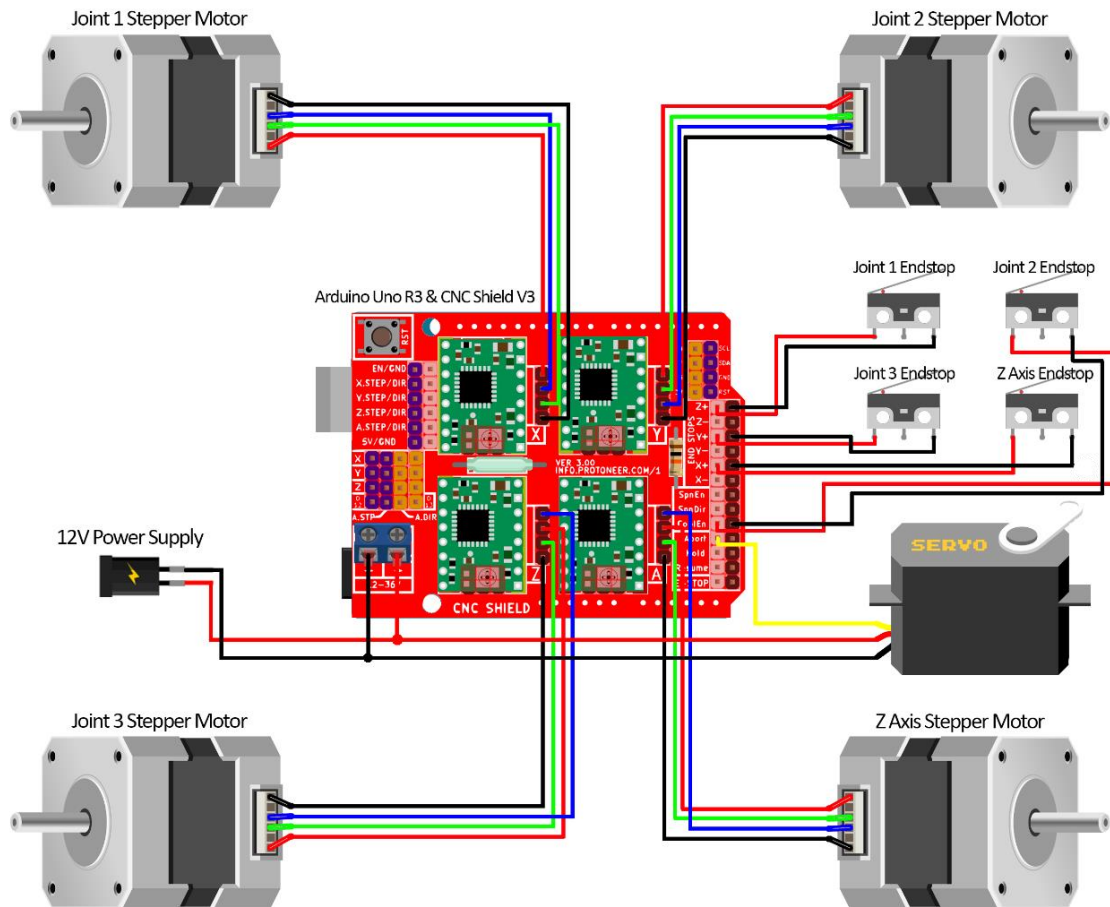


Εικόνα 11 : Limit Switch (Τερματικός Διακόπτης).

(What is a Micro Switch : Working & Its Applications, 22)

Ηλεκτρονικό κύκλωμα

Σε αυτήν την ενότητα και τελευταία του 7^{ου} κεφαλαίου της παρούσης διπλωματικής εργασίας παρουσιάζεται ολόκληρο το κύκλωμα ελέγχου του ρομποτικού βραχίονα, περιλαμβάνονται δηλαδή όλα τα εξαρτήματα που περιεγράφηκαν παραπάνω, στο τελικό σχέδιο που υλοποιήθηκε μέσω του λογισμικού Fritzing.



Εικόνα 12 : Ηλεκτρονικό Κύκλωμα Ελέγχου Ρομποτικού Βραχίονα

8. ΣΥΝΑΡΜΟΛΟΓΗΣΗ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ

Σε αυτό το κεφάλαιο γίνεται η αναλυτική περιγραφή των μηχανικών μερών και εξαρτημάτων που χρησιμοποιήθηκαν στην κατασκευή του ρομποτικού βραχίονα καθώς και μια εκτεταμένη επεξήγηση της διαδικασίας της συναρμολόγησης αρχικά των μηχανικών μερών του ρομποτικού βραχίονα με τα 3D Printed κομμάτια του αλλά και οι τροποποιήσεις οι οποίες χρειάστηκε να γίνουν με σκοπό να επιτευχθεί η σωστή εφαρμογή και λειτουργία τους. Επιπλέον, περιγράφονται τα βήματα που ακολουθήθηκαν κατά την υλοποίηση της συνδεσμολογίας των ηλεκτρονικών εξαρτημάτων της κατασκευής. Τέλος, ακολουθούν οι τρεις αναλυτικές λίστες με τα 3D Printed κομμάτια, ηλεκτρονικά εξαρτήματα, αλλά και τα μηχανικά εξαρτήματα κατά αυτή την σειρά.

8.1 Λειτουργικά Μέρη και Εξαρτήματα Κατασκευής

Σε αυτή την παράγραφο της 8^{ης} ενότητας παρουσιάζονται τα επιμέρους μηχανικά και λειτουργικά μέρη αλλά και τα μηχανικά εξαρτήματα από τα οποία αποτελείται η κατασκευή.

Ρουλεμάν

Το ρουλεμάν είναι ένα μηχανικό στοιχείο του οποίου ο σκοπός είναι να περιορίζει την τριβή μεταξύ κινούμενων μερών. Είναι σχεδιασμένα έτσι ώστε, είτε να διευκολύνει την γραμμική κίνηση ενός κινούμενου μέρους είτε μια ελεύθερη περιστροφή γύρω από μία ράβδο είτε ακόμα και να σταματήσει κάποια κίνηση ασκώντας έλεγχο στις δυνάμεις οι οποίες ασκούνται μεταξύ των κινούμενων μερών. Το μεγαλύτερο ποσοστό των ρουλεμάν της αγοράς χρησιμοποιούνται με σκοπό την εξάλειψη της τριβής και κατά συνέπεια την διευκόλυνση της κίνησης. Τα ρουλεμάν ανάλογα με την σχεδίαση τους και την λειτουργία τους στην εφαρμογή για την οποία προορίζονται ταξινομούνται σε κατηγορίες. (Bearing (mechanical), 2022)

Στην παρούσα εφαρμογή οι τύποι οι οποίοι χρησιμοποιούνται είναι τα Ball Bearings (Ένσφαιρα Ρουλεμάν) και τα Thrust Bearings (Ωστικά Ρουλεμάν) τα οποία θα αναλυθούν παρακάτω.

Ball Bearing

Το Ball Bearing είναι ένας τύπος κυκλικού ρουλεμάν το οποίο με την χρήση μικρών μπαλών διατηρεί τον διαχωρισμό μεταξύ δύο, συνήθως μεταλλικών, στεφανιών. Τα δύο αυτά στεφάνια έχουν το ένα μικρότερη διάμετρο από το άλλο και καθώς περιστρέφονται, οι μπάλες με την σειρά τους έχουν την ιδιότητα να κυλιούνται κατά την περιστροφή. Η κύλιση αυτή έχει πολύ χαμηλότερο συντελεστή τριβής συγκριτικά με την περίπτωση που οι δύο επίπεδες επιφάνειες να γλιστρούσαν μεταξύ τους. Σκοπός χρήσης του συγκεκριμένου τύπου ρουλεμάν είναι η απαλοιφή των περιστροφικών τριβών που μπορεί να υπάρχουν μεταξύ ακτινικών ή αξονικών φορτίων. (Ball bearing, 2022)

Thrust Bearing

Το Thrust Bearing (ωστικό ρουλεμάν) αποτελεί έναν τύπο περιστρεφόμενου ρουλεμάν. Είναι σχεδιασμένα με τρόπο τέτοιο, ώστε να υποστηρίζουν την περιστροφή εξαρτημάτων υπό αξονικό φορτίο. Το είδος thrust bearing που χρησιμοποιούνται στην συγκεκριμένη εφαρμογή είναι τα thrust ball bearing τα οποία αποτελούνται από δύο δαχτυλίδια, παράλληλα μεταξύ τους και όπως στα ball bearing, φέρουν μπάλες ενδιάμεσα που συγκρατούνται όλες επάνω σε ένα τρίτο δαχτυλίδι. (Thrust Bearing, 2022)

Leadscrew

Leadscrew είναι ο τραπεζοειδής κοχλίας, ο οποίος χρησιμοποιείται σε εφαρμογές που είναι θεμιτή η μετατροπή μιας στροφικής κίνησης σε γραμμική. Χρησιμοποιούνται συνήθως σε εφαρμογές χαμηλής ισχύος για την ρύθμιση θέσης. (Leadscrew, 2022)

Στην περίπτωση του ρομποτικού βραχίονα επιλέχθηκε για τον έλεγχο θέσης του άξονα Z.

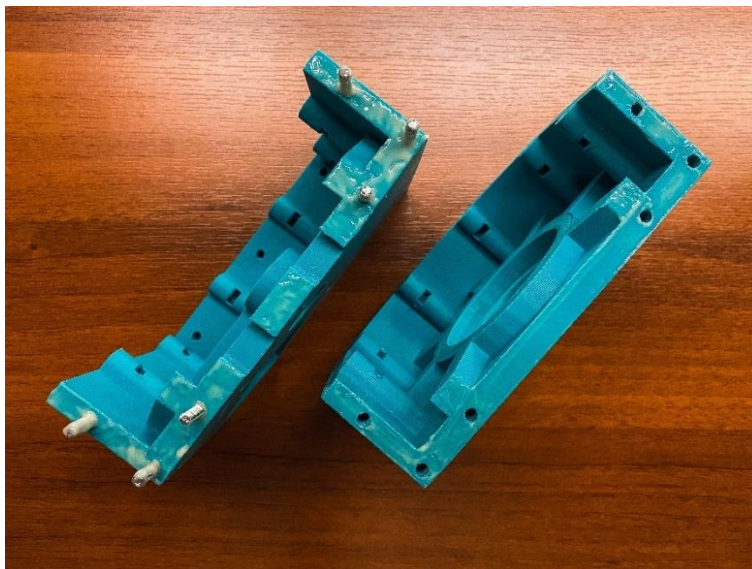
GT2 Ιμάντες Χρονισμού & Γρανάζια

Οι οδοντωτοί ιμάντες χρονισμού είναι ιμάντες μετάδοσης γραμμικής κίνησης. Φέρουν ένα στρογγυλεμένο προφίλ “δοντιού” με βήμα (Απόσταση μεταξύ δοντιών) 2mm, το οποίο είναι σχεδιασμένο ώστε το δόντι του ιμάντα να εφαρμόζει με ακρίβεια στην αυλάκωση του γραναζιού, οπότε όταν αλλάζει η κατεύθυνση κίνησης του γραναζιού, δεν υπάρχει χώρος να κινηθεί το δόντι του ιμάντα εντός της αυλάκωσης. Τα γρανάζια που χρησιμοποιούνται για την μετάδοση κίνησης φέρουν τρύπα η οποία έχει εσωτερική διάμετρο όμοια με αυτή του άξονα μετάδοσης κίνησης, καθώς και βίδα σύσφιξης του γραναζιού στον άξονα. (GT2 Timing Belt, 2015)

8.2 Συναρμολόγηση Κατασκευής

Σε αυτήν την παράγραφο, την 2^η της ενότητας 8 της διπλωματικής εργασίας γίνεται η αναλυτική περιγραφή της συναρμολόγησης του ρομποτικού βραχίονα βήμα προς βήμα.

Αρχικά η συναρμολόγηση ξεκίνησε με την βάση του ρομποτικού βραχίονα. Αυτή λόγω, του μεγάλου μεγέθους της συγκριτικά με την επιφάνεια του 3D Printer, που χρησιμοποιήθηκε για την 3D εκτύπωση των πλαστικών μερών, χρειάστηκε να εκτυπωθεί σε δύο τμήματα τα οποία στην συνέχεια ενώθηκαν με εποξική κόλλα δύο συστατικών καθώς και με σκουλικόβιδες στο εσωτερικό οι οποίες χρησίμευσαν στο να επιτευχθεί μια πιο στιβαρή κόλληση και επιπλέον να κολληθούν ευθύγραμμα τα τμήματα μεταξύ τους.



Εικόνα 13 : Τοποθέτηση Εποξικής Κόλλας & Βιδών Στήριξης.

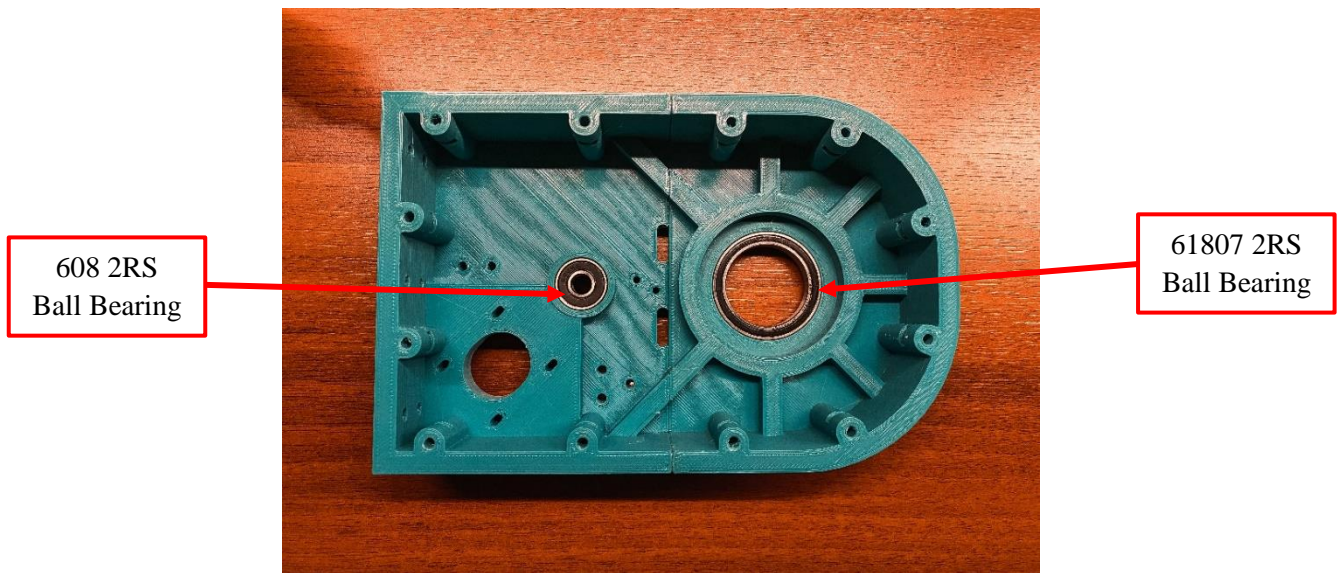
Έπειτα ασκήθηκε η κατάλληλη πίεση με σφικκτήρα ώστε να κολληθεί σωστά. Αφού πέρασε το χρονικό διάστημα, το οποίο αναγράφεται στις προδιαγραφές της κόλλας, που απαιτείται ώστε να στεγνώσει πλήρως (περίπου 24 ώρες) ξεκίνησε η συναρμολόγηση της βάσης.



Εικόνα 14 : Σύσφιξη Βάσης Ρομποτικού Βραχίονα.

Ξεκινώντας την συναρμολόγηση της βάσης χρειάστηκε να τριφτούν τα έδρανα των ρουλεμάν ώστε να μπορεί να επιτευχθεί η τέλεια εφαρμογή τους. Έπειτα τοποθετήθηκαν πρώτα τα ρουλεμάν τύπου 608 2RS για την στήριξη και περιστροφή της βίδας στην οποία θα τοποθετηθεί το γρανάζι για την μετάδοση κίνησης από τον βηματικό κινητήρα στο γρανάζι της άρθρωσης.

Μετά τοποθετήθηκε το ρουλεμάν τύπου 61807 2RS στο έδρανο της άρθρωσης, το οποίο είναι υπεύθυνο για την ομαλή περιστροφή της επάνω στην βάση.



Εικόνα 15 : Τοποθέτηση Ρουλεμάν Στην Βάση.

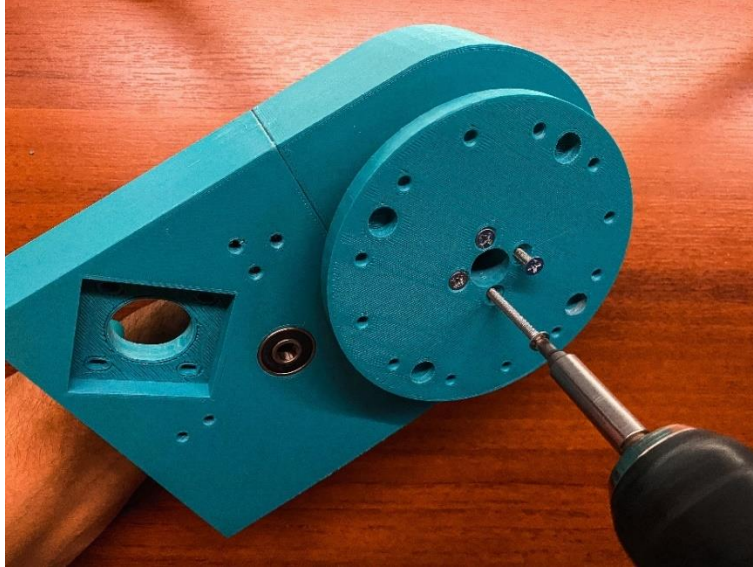
Μετά την τοποθέτηση των ρουλεμάν στην βάση ακολούθησε η συναρμολόγηση και τοποθέτηση του μηχανισμού περιστροφής της 1^{ης} άρθρωσης. Αρχικά τοποθετήθηκε γράσο στα ωστικά ρουλεμάν, με εσωτερική διάμετρο 40mm και εξωτερική 60mm, για την λίπανση, ομαλή λειτουργία τους και την προστασία τους από πιθανή διάβρωση. Έπειτα τοποθετήθηκαν τα ωστικά ρουλεμάν, το ένα στο γρανάζι μετάδοσης κίνησης και το δεύτερο στο έδρανο της βάσης της 1^{ης} άρθρωσης.



Εικόνα 16: Τοποθέτηση Ωστικών Ρουλεμάν.

Το γρανάζι μίπηκε στο εσωτερικό και κάτω μέρος της βάσης κατά τρόπο τέτοιο ώστε να εφάπτεται με το εσωτερικό του ρουλεμάν έως το μισό του ύψους του.

Στην συνέχεια τοποθετήθηκε η βάση της άρθρωσης στο επάνω μέρος της βάσης με τρόπο τέτοιο ώστε οι τρύπες να είναι συνευθειακές με τις αντίστοιχες του γραναζιού και παράλληλα να εφάπτονται τέλεια μεταξύ τους ώστε να μην δημιουργηθεί κενό κατά το πέρασμα των φρεζάτων βιδών M4 x 55mm οι οποίες ασφαλίστηκαν με παξιμάδια ασφαλείας στο άκρο τους, ώστε να επιτευχθεί η στιβαρή σύνδεση τους.



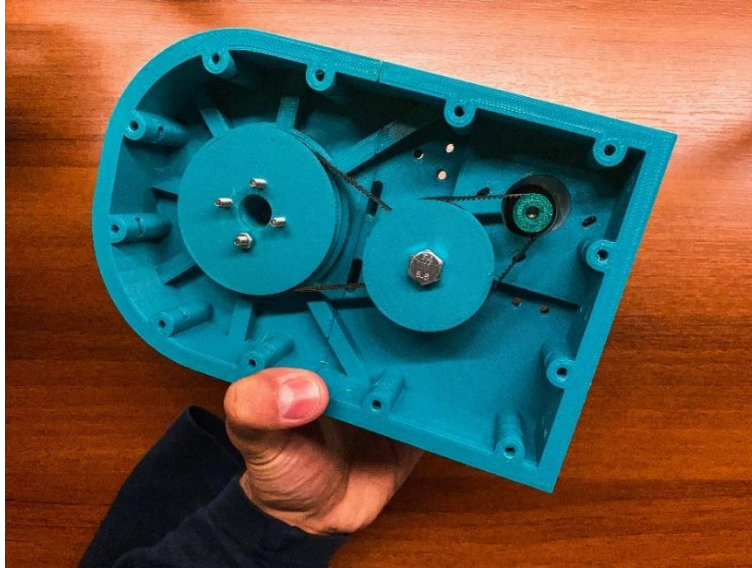
Εικόνα 17: Βίδωμα & Στήριξη 1^{ης} Άρθρωσης.

Ολοκληρώνοντας την στήριξη της βάσης, υλοποιήθηκε η στήριξη του γραναζιού-μείωσης. Αυτό στηρίχτηκε με μία βίδα M8 x 45mm και μια ροδέλα εσωτερικής διαμέτρου 8mm, ώστε το γρανάτζι να μην περιστρέφεται επάνω στο πλευρό του ρουλεμάν, και ασφαλίστηκε όπως και στην περίπτωση της άρθρωσης με παξιμάδι ασφαλείας.



Εικόνα 18: Συναρμολόγηση Γραναζιού Μείωσης.

Πριν συσφιχθεί το γρανάζι όμως έπρεπε να τοποθετηθούν οι κλειστοί ιμάντες μετάδοσης κίνησης, ώστε να είναι σωστά τεντωμένοι για να μην υπάρχουν απώλειες βημάτων στην κίνηση. Για να γίνει αυτό πρώτα μπήκε το 3^ο γρανάζι του εσωτερικού της βάσης, αυτό το οποίο είναι υπεύθυνο για την μετάδοση κίνησης από τον βηματικό κινητήρα, στα υπόλοιπα γρανάζια. Αυτό κολλήθηκε επάνω στον άξονα του κινητήρα. Στην συνέχεια τοποθετήθηκαν οι ιμάντες, και έγινε η επιλογή της θέσης του βηματικού κινητήρα ώστε να είναι ιδανική η τάνυση του ιμάντα. Αυτός με την σειρά του βιδώθηκε με 4 βίδες M3 x 12mm και ροδέλες M3 ώστε να είναι μεγαλύτερη η επιφάνεια πίεσης του κεφαλιού της βίδας επάνω στην βάση.



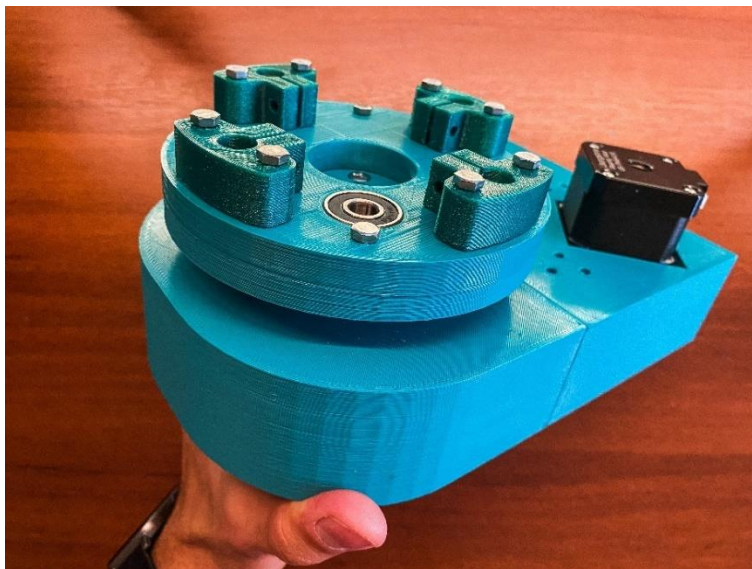
Εικόνα 19: Ολοκληρωμένη Βάση Ρομποτικού Βραχίονα.

Στο επόμενο βήμα της συναρμολόγησης του ρομποτικού βραχίονα, έγινε η συναρμολόγηση του άξονα Z του ρομποτικού βραχίονα. Το βήμα αυτό χωρίστηκε σε δύο διαδικασίες.

Η πρώτη ήταν η διαδικασία της συναρμολόγησης του κάτω μέρους του άξονα και η δεύτερη, αυτή του πάνω μέρους. Αυτό, έγινε έτσι ώστε αφού ολοκληρωθεί το κάτω μέρος του άξονα Z, να υπάρχει ελευθερία χώρου και κίνησης κατά την συναρμολόγηση της 2^{ης} και της 3^{ης} άρθρωσης. Μετά την συναρμολόγηση αυτών ολοκληρώθηκε ο άξονας Z με το δέσιμο του πάνω μέρους του.

Αρχικά τοποθετήθηκε και πακτώθηκε με εξάγωνες βίδες M4 x 20mm και παξιμάδια επάνω στην 1^η άρθρωση ο κυκλικός δίσκος του άξονα, πάνω στον οποίο στηρίχθηκαν κατά το ίδιο τρόπο με πριν και βίδες M4 x 35mm αυτήν την φορά, οι σφιγκτήρες για τις μεταλλικές ράβδους επάνω στις οποίες θα κινείται η 2^η άρθρωση. Η συναρμολόγηση του χαμηλότερου τμήματος του άξονα Z ολοκληρώθηκε με την τοποθέτηση του ρουλεμάν σταθεροποίησης του τραπεζοειδή κοχλία.

Στην συνέχεια μετρήθηκαν και κόπηκαν οι ράβδοι στο επιθυμητό μήκος και λειάνθηκαν τα άκρα τους ώστε να επιτευχθεί η ομαλότερη τοποθέτησή τους εντός των σφιγκτήρων. Έπειτα μπήκαν οι ράβδοι στις εσοχές των σφιγκτήρων και ακολούθησε η συναρμολόγηση της βάσης του 1^{ου} συνδέσμου.



Εικόνα 20: Συναρμολόγηση Βάσης Άξονα Z.

Αυτό το βήμα ξεκίνησε με την βάση της άρθρωσης, σε αυτήν τρίφτηκαν και την λειάνθηκαν με γυαλόχαρτο τα έδρανα των γραμμικών ρουλεμάν, ώστε να μπαίνουν με μικρή πίεση και να μην φεύγουν από την θέση τους κατά την διάρκεια της κίνησης του άξονα Z. Έπειτα έγινε η στήριξη του βηματικού κινητήρα, στον οποίο πρώτα είχε τοποθετηθεί το γρανάζι μετάδοσης κίνησης, επάνω στην βάση στην οποία βιδώθηκε με βίδες M3 x 12mm, και ροδέλες M3. Τέλος, βιδώθηκε το παξιμάδι του τραπεζοειδή κοχλία με βίδες M3 x 16mm και παξιμάδια M3.



Εικόνα 21: Ολοκληρωμένη Βάση 1ης Άρθρωσης.

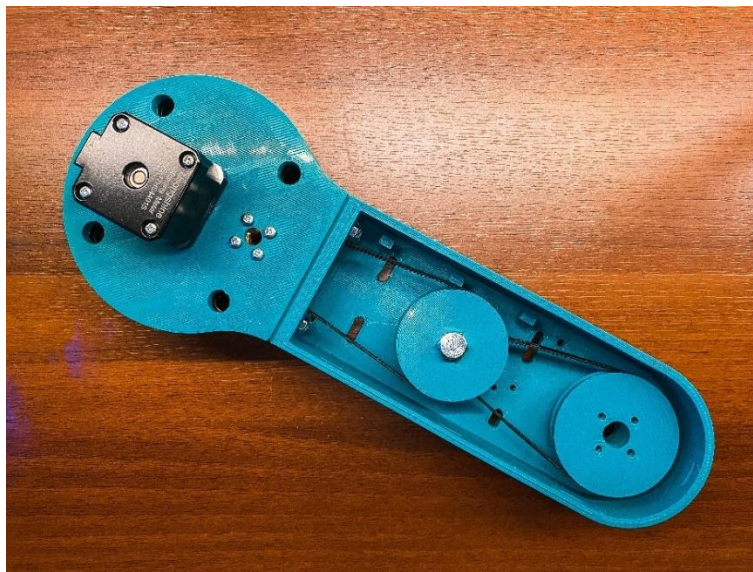
Μετά την τοποθέτηση των εξαρτημάτων στην βάση της 1^{ου} συνδέσμου του ρομποτικού βραχίονα, ακολούθησε η συναρμολόγηση του 1^{ου} συνδέσμου. Εδώ όπως και στα προηγούμενα βήματα, ακολούθησε η διαδικασία καθαρισμού και λείανσης ατελειών που έμειναν από την τρισδιάστατη εκτύπωση και έπειτα τοποθετήθηκαν τα ρουλεμάν. Αναλυτικά χρησιμοποιήθηκαν δύο ρουλεμάν

τύπου 608 RS για την βίδα στήριξης του ενδιάμεσου γραναζιού μείωσης, και ένα ρουλεμάν 61806 2RS στην θέση περιστροφή της 2^{ης} άρθρωσης. Τέλος, ο 1^{ος} σύνδεσμος του ρομποτικού βραχίονα ενώθηκε με την βάση του με εξάγωνες βίδες M5 x 25mm και τα αντίστοιχα παξιμάδια M5.



Εικόνα 22: Τοποθέτηση 1ου Συνδέσμου στην Βάση του

Αφού ολοκληρώθηκε η ένωση, ακολούθησε η συναρμολόγηση των κινούμενων μερών του 1^{ου} συνδέσμου και αυτά τα οποία είναι υπεύθυνα για την μετάδοση της κίνησης στην 2^η άρθρωση. Πρώτα τοποθετήθηκαν οι δύο κλειστοί μάντες GT2 στις θέσεις στις οποίες προορίζονταν. Έπειτα, έγινε η στήριξη του γραναζιού μείωσης, κατά τον ίδιο τρόπο που υλοποιήθηκε και στην βάση του ρομποτικού βραχίονα και τοποθετήθηκε ο πρώτος μάντας.



Εικόνα 23: Συναρμολόγηση 1ου Συνδέσμου

Μετά την στήριξη του 1^{ου} γραναζιού ακολούθησε το γρανάζι της 2^{ης} άρθρωσης, στο οποίο αφού περάστηκε ο 2^{ος} μάντας σε συνδυασμό με το 1^ο γρανάζι (μείωσης), εφαρμόστηκε το ωστικό ρουλεμάν με εσωτερική διάμετρο 35mm και εξωτερική διάμετρο 52mm, ώστε να πατήσει στο εσωτερικό του συνδέσμου και να μπορεί να πραγματοποιηθεί η ομαλή περιστροφή του γραναζιού. Για την βέλτιστη λειτουργία και αντοχή του στην διάβρωση λιπάνθηκε και αυτό με γράσο.

Τέλος, αφού τοποθετήθηκε το γρανάζι της 2^{ης} άρθρωσης παρατηρήθηκε ότι οι μάντες δεν έφεραν την επιθυμητή τάνυση. Στην προκειμένη περίπτωση χρειάστηκε να αξιοποιηθούν οι εσοχές που έφερε ο σύνδεσμος για την εγκατάσταση τεντοτήρων. Αυτοί δημιουργήθηκαν με φρεζάτες βίδες M5 x 40mm, ροδέλες και παξιμάδια M5 καθώς και μια τροχαλία χωρίς δόντια (Idler Pulley).

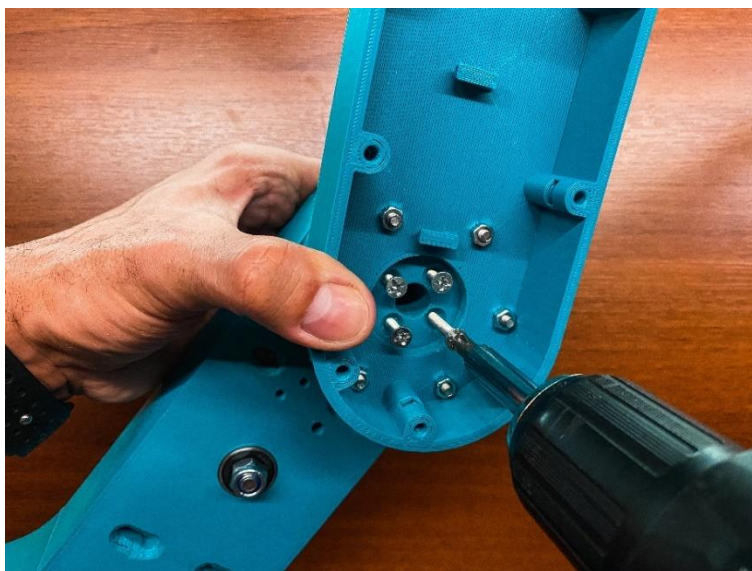
Στο επόμενο βήμα της συναρμολόγησης του ρομποτικού βραχίονα, ενώθηκε η 2^η άρθρωση, αρχικά με το επάνω μέρος του 2^{ου} συνδέσμου και στην συνέχεια με το γρανάζι που τοποθετήθηκε προηγουμένως στο εσωτερικό του 1^{ου} συνδέσμου. Αρχικά, αφού λιπάνθηκε, τοποθετήθηκε στο εσωτερικό έδρανο της άρθρωσης το ωστικό ρουλεμάν, το οποίο έφερε όμοια χαρακτηριστικά με το προηγούμενο. Έπειτα, μπήκαν οι εξάγωνες βίδες M5 x 25mm, πέρασαν στο εσωτερικό του 2^{ου} συνδέσμου από τις κατάλληλες τρύπες και σφίχτηκαν εφαρμόζοντας παξιμάδια M5 στα άκρα του.



Εικόνα 24: Εγκατάσταση 2^{ης} Άρθρωσης Στον 2^ο Σύνδεσμο.

Εν συνεχεία του βήματος αυτού, μπήκαν στις εσοχές τις άρθρωσης οι φρεζάτες βίδες M4 x 50mm για την σύσφιξη της με το αντίστοιχο γρανάζι. Οπότε αφού η άρθρωση τοποθετήθηκε στην εσοχή του 1^{ου} συνδέσμου με τρόπο τέτοιο ώστε οι τρύπες του γραναζιού να συμπίπτουν με αυτές της 2^{ης} άρθρωσης βιδώθηκαν και ασφαλίστηκαν με τα αντίστοιχα παξιμάδια M4.

Μετά από το επάνω μέρος του 2^{ου} συνδέσμου, υλοποιήθηκε η τοποθέτηση των μηχανολογικών εξαρτημάτων στο κάτω μέρος του, με τον ίδιο τρόπο και βήματα που ακολουθήθηκαν στην βάση, και στον 1^ο σύνδεσμο.



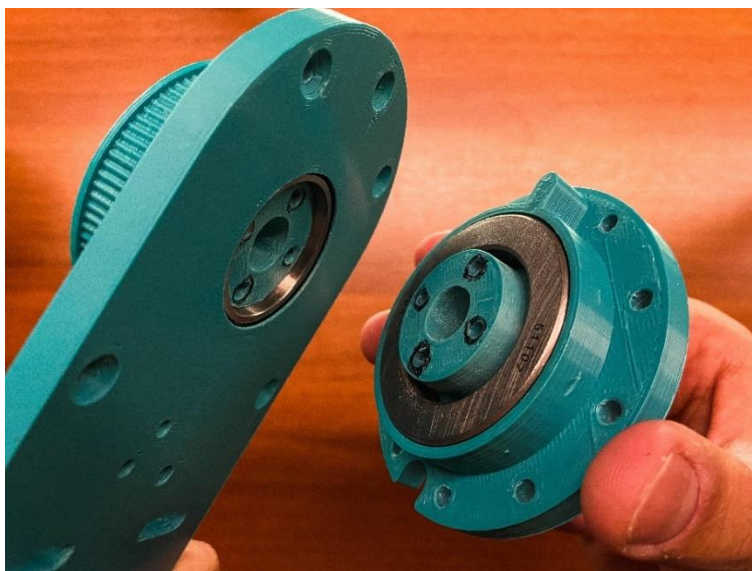
Εικόνα 25: Στήριξη 2^{ης} Άρθρωσης & Ένωση 1^{ου}-2^{ου} Συνδέσμου.

Σε αντίθεση όμως με τις προηγούμενες περιπτώσεις ο 2^{ος} σύνδεσμος δεν χρειάστηκε ρουλεμάν τύπου 608 2RS μιας και δεν υπάρχει γρανάζι μείωσης και η μετάδοση κίνησης γίνεται ευθέως από το μοτέρ στο γρανάζι της 4^{ης} άρθρωσης μόνο με τον κλειστό ιμάντα. Εδώ, πρώτα μπήκε στο έδρανο της 3^{ης} άρθρωσης το ρουλεμάν τύπου 61806 2RS και έπειτα στηρίχτηκε ο βηματικός κινητήρας με βίδες M3 x 12mm και ροδέλες M3.



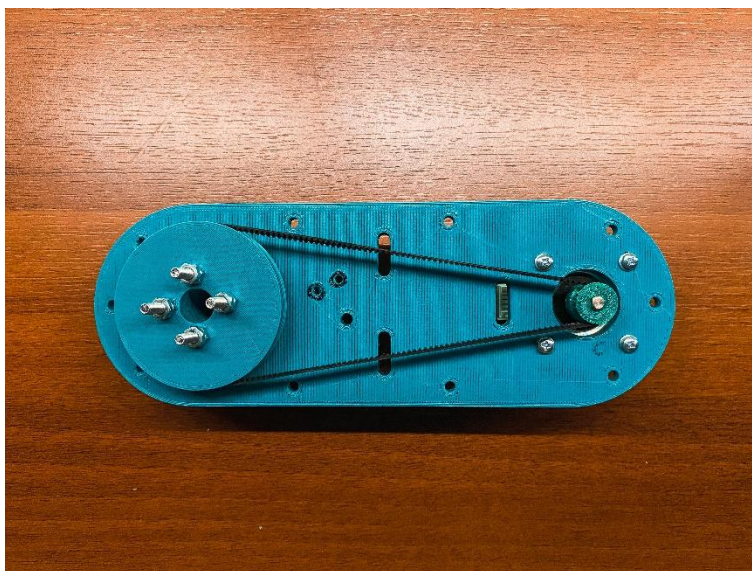
Εικόνα 26: Τοποθέτηση Ρουλεμάν Στον 2^ο Σύνδεσμο

Αφού βιδώθηκε στον σύνδεσμο ο βηματικός κινητήρας, στον οποίο προηγουμένως είχε στηριχτεί το γρανάζι μετάδοσης κίνησης, έγινε η προετοιμασία για την εγκατάσταση του γραναζιού της 4^{ης} άρθρωσης. Εδώ, χρησιμοποιήθηκε άλλο ένα ωστικό ρουλεμάν όμοιο με αυτό της προηγούμενης άρθρωσης και στην συνέχεια έγινε η τοποθέτηση του γραναζιού στο εσωτερικό του κάτω μέρους του συνδέσμου.



Εικόνα 27: Εγκατάσταση 3^{ης} Άρθρωσης

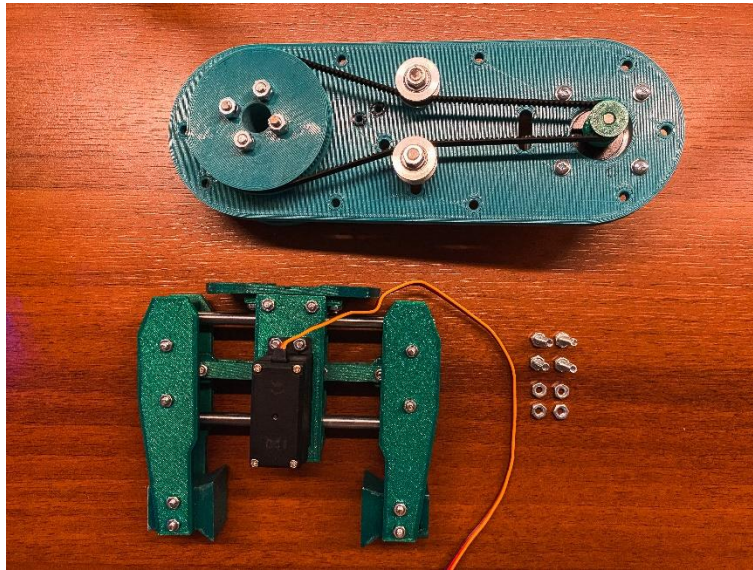
Στην αντίστοιχη εξωτερική θέση τοποθετήθηκε, ακολουθώντας όμοια βήματα, η 4^η άρθρωση. Αυτή, αφού περάστηκε γύρω της ο ιμάντας μετάδοσης κίνησης ενώθηκε με το γρανάτζι χρησιμοποιώντας βίδες M4 x 50mm και τα αντίστοιχα παξιμάδια M4. Τέλος όπως και στην περίπτωση του 2^{ου} συνδέσμου κρίθηκε απαραίτητη η εφαρμογή τεντοτήρων διότι ο ιμάντας δεν είχε την κατάλληλη τάνυση.



Εικόνα 28: Κάτω Μέρος 2^{ου} Συνδέσμου

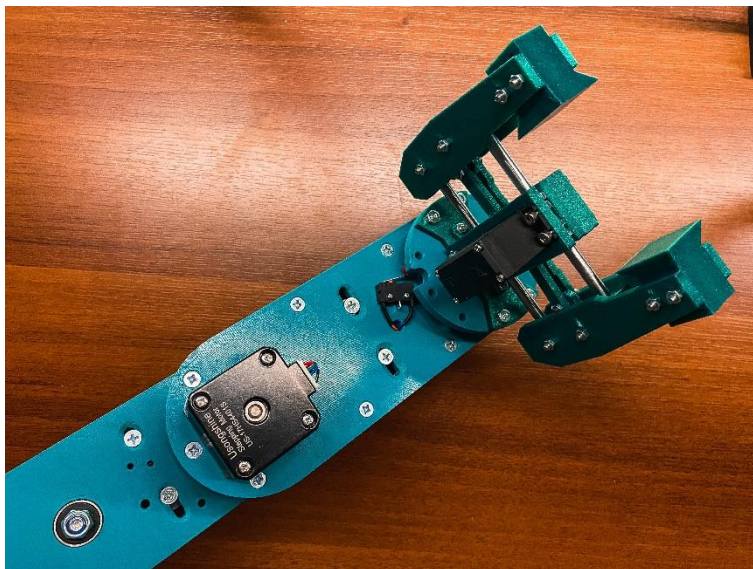
Στο επόμενο βήμα της συναρμολόγησης, υλοποιήθηκε η κατασκευή του τελευταίου τμήματος του ρομποτικού βραχίονα, δηλαδή ο μηχανισμός του άκρου εργασίας του. Αυτό το στάδιο ξεκίνησε με την στήριξη του κινητήρα servo στην βάση του με τέσσερις βίδες M3 x 10mm και την συναρμολόγηση των δύο άκρων όπου στην κάθε μεριά χρειάστηκαν τέσσερις βίδες M3 x 25mm και τα αντίστοιχα παξιμάδια M3. Έπειτα μπήκαν οι άξονες επάνω στους οποίους κινούνται τα δύο

άκρα. Στην συνέχεια τοποθετήθηκαν τα άκρα επάνω στους άξονες και βιδώθηκαν με τα άκρα του σέρβο. Τέλος για την προστασία του μηχανισμού κίνησης του σέρβο, τοποθετήθηκε ένα καπάκι το οποίο βιδώθηκε επάνω στην βάση του.



Εικόνα 29: Μηχανισμός Άκρου Εργασίας & Κάτω Μέρος 2^{ου} Συνδέσμου.

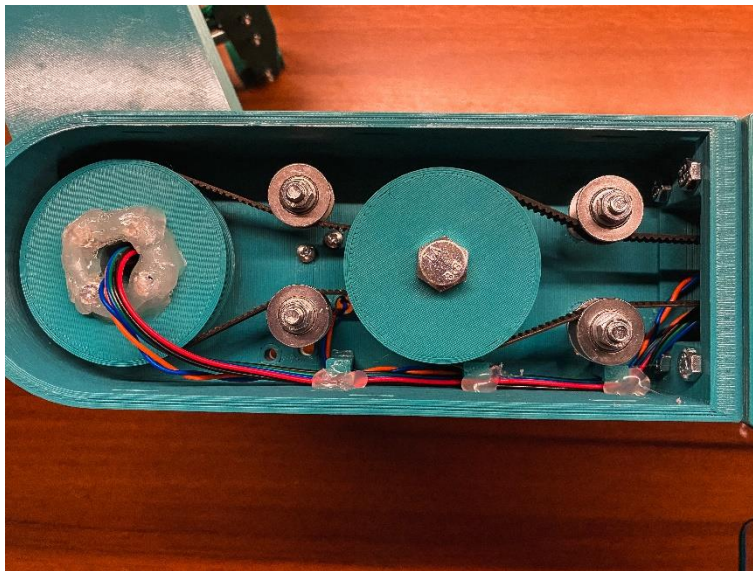
Μετά την συναρμολόγηση του μηχανισμού του άκρου εργασίας του ρομποτικού βραχίονα, υλοποιήθηκε η ένωση του μηχανισμού με το 2^ο σύνδεσμο. Εδώ χρησιμοποιήθηκε μια πλάκα, η οποία αρχικά βιδώθηκε στο επάνω μέρος του μηχανισμού με τέσσερις βίδες M3 x 8mm και τα αντίστοιχα παξιμάδια και στην συνέχεια στηρίχτηκε στο κάτω μέρος της 3^{ης} άρθρωσης με τέσσερις βίδες M4 x 16mm, τα οποία ασφαλίστηκαν με παξιμάδια ασφαλείας .



Εικόνα 30: Στήριξη Μηχανισμού Άκρου Εργασίας & Κάτω Μέρος 2ου Συνδέσμου.

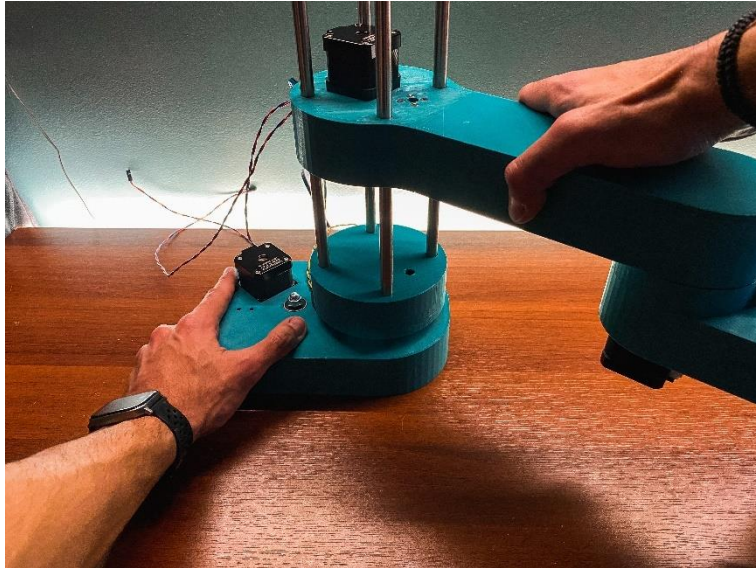
Την συνέχεια της συναρμολόγησης ακολούθησε η συνδεσμολογία του ηλεκτρολογικού κυκλώματος. Για να επιτευχθεί αυτό κολλήθηκαν καλώδια με ακροδέκτες στα limit switch έτσι ώστε να μπορούν να συνδεθούν με την πλακέτα ελέγχου του ρομποτικού βραχίονα. Έπειτα, στηρίχτηκαν το κάθε ένα στην ανάλογη άρθρωση ξεκινώντας από την 3^η πηγαίνοντας προς την βάση. Στην 3^η άρθρωση αφού βιδώθηκε το limit switch, συνδέθηκε και το καλώδιο του βηματικού κινητήρα, τα οποία περάστηκαν στο εσωτερικό της και στηρίχτηκαν με θερμή κόλλα, ώστε να αποφευχθεί τυχόν τριβή ή ζημιά στα καλώδια κατά την κίνηση των μηχανικών μερών. Το κάτω μέρος του συνδέσμου, αφού περάστηκαν τα καλώδια του στο εσωτερικό της 2^{ης} άρθρωσης, ενώθηκε με το επάνω μέρος με φρεζάτες βίδες M4 x 25mm.

Αφού περάστηκαν τα καλώδια στο εσωτερικό της 2^{ης} άρθρωσης και κατέληξαν στο εσωτερικό του 1^{ου} συνδέσμου, παρατηρήθηκε ότι κατά την κίνηση του 2^{ου} συνδέσμου τα καλώδια μπλέκονταν στα παξιμάδια ασφαλείας που βρίσκονταν περιμετρικά του κέντρου της 2^{ης} άρθρωσης. Για να αποφευχθεί αυτό το φαινόμενο και η ενδεχόμενη καταστροφή καλωδίων, εφαρμόστηκε θερμή κόλλα γύρω και πάνω στα παξιμάδια. Έπειτα, τοποθετήθηκε στον σύνδεσμο το limit switch της 2^{ης} άρθρωσης. Εδώ, όπως και στην προηγούμενη περίπτωση είχαν τοποθετηθεί τα καλώδια, τα οποία περάστηκαν και αυτά στο εσωτερικό του συνδέσμου και μαζί με τα καλώδια από τον προηγούμενο σύνδεσμο, στηρίχτηκαν με θερμή κόλλα σε όλο το μήκος 1^{ου} συνδέσμου μέχρι την βάση του.



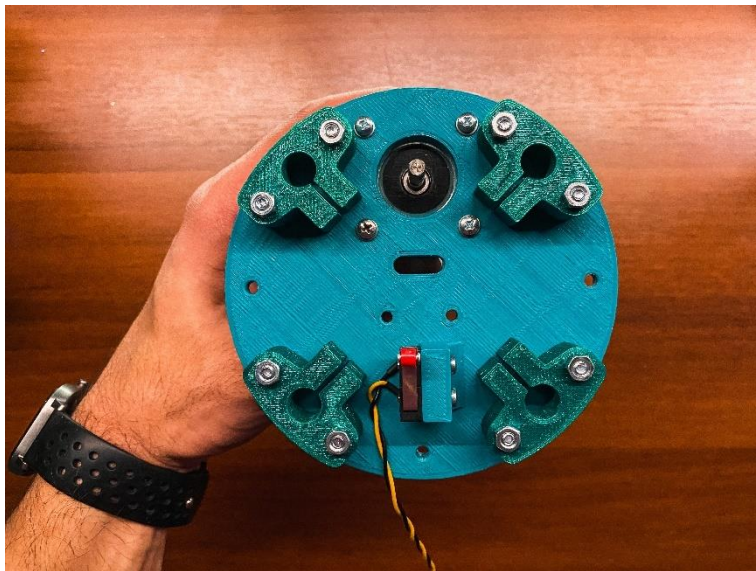
Εικόνα 31: Ηλεκτρολογική Σύνδεση & Στήριξη Καλωδίων στον 1^ο Σύνδεσμο.

Τέλος, αφού ολοκληρώθηκε τόσο η μηχανολογική όσο και η ηλεκτρολογική συναρμολόγηση του 2^{ου} και του 1^{ου} συνδέσμου του ρομποτικού βραχίονα, τοποθετήθηκε το καπάκι προστασίας του 1^{ου} συνδέσμου. Έπειτα αφού συνδέθηκε το καλώδιο του βηματικού κινητήρα της 2^{ης} άρθρωσης, περάστηκε στο εσωτερικό της βάσης του 1^{ου} βραχίονα και μαζί με τα υπόλοιπα οδηγήθηκε στην έξοδο την οποία έχει σχεδιαστεί στο πίσω μέρος της. Στο σημείο αυτό το πλήρως συναρμολογημένο τμήμα του ρομποτικού βραχίονα τοποθετήθηκε επάνω στις γραμμικές ράβδους του άξονα Z. Αφού ελέγχθηκε η ομαλή κίνηση της βάσης επάνω στις ράβδους, εξασφαλίστηκε δηλαδή η παραλληλία τους, βιδώθηκαν οι βίδες στους σφικτήρες ώστε να σταθεροποιηθούν οι ράβδοι στην θέση τους και τοποθετήθηκε το προστατευτικό καπάκι.



Εικόνα 32: Τοποθέτηση Συνδέσμων Βραχίονα Στον Άξονα Z.

Στο επόμενο βήμα της συναρμολόγησης του ρομποτικού βραχίονα υλοποιήθηκε η 2^η διαδικασία από το βήμα της συναρμολόγησης του Άξονα Z, αυτή δηλαδή του επάνω μέρους. Όπως και στο κάτω μέρος έτσι και στο πάνω τοποθετήθηκαν οι σφιγκτήρες, με την διαφορά ότι σε αυτήν την περίπτωση η στήριξη τους έγινε με φρεζάτες βίδες M4 x 25mm ώστε να μην εξέλθουν από την εξωτερική μεριά του πάνω μέρους. Στην συνέχεια τοποθετήθηκε ο βηματικός κινητήρας και βιδώθηκε στο επάνω μέρος με βίδες M3 x 12mm και ροδέλες M3.

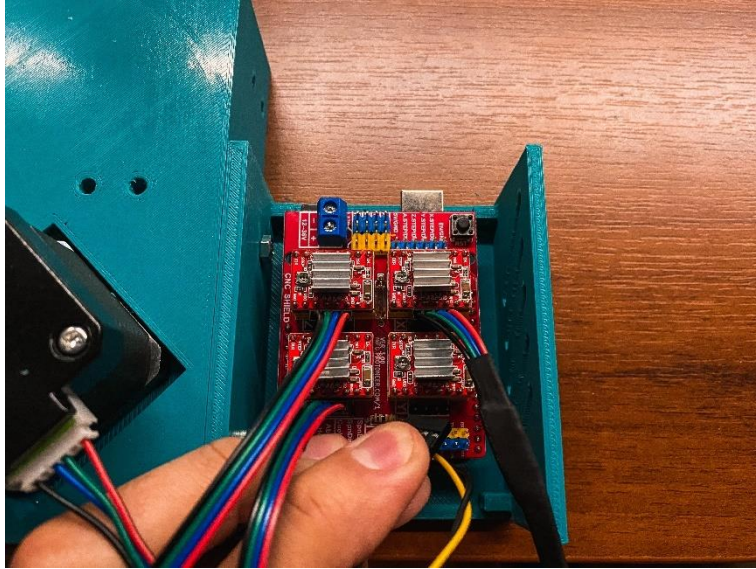


Εικόνα 33:Επάνω Μέρος Άξονα Z

Έπειτα στηρίχθηκε το limit switch του άξονα Z και υλοποιήθηκε η ηλεκτρολογική συναρμολόγηση των καλωδίων του, και των καλωδίων του βηματικού κινητήρα. Στο άκρο του άξονα του κινητήρα τοποθετήθηκε ένας μετατροπέας από 5mm σε 8mm ώστε να συνδεθεί το άκρο του μωτέρ με τον τραπεζοειδή κοχλία. Τέλος, περάστηκε ο τραπεζοειδής κοχλίας στον παξιμάδι της βάσης του

πρώτου συνδέσμου, έως ότου φτάσει το ρουλεμάν στο κάτω άκρο του άξονα Z, και ακολούθησε το πάνω μέρος.

Το τελευταίο βήμα κατά το οποίο ολοκληρώθηκε η διαδικασία της συναρμολόγησης του ρομποτικού βραχίονα, ήταν η εγκατάσταση της πλακέτας ελέγχου στην θήκη της και η σωστή σύνδεση των καλωδίων επάνω της. Αφού στηρίχτηκε το Arduino και η πλακέτα ελέγχου μπήκε επάνω του ξεκίνησε η σύνδεση των καλωδίων στα οποία προηγουμένως κολλήθηκαν σημάσεις ώστε να γίνει σωστά και εύκολα η σύνδεση τους στην πλακέτα. Πρώτα κολλήθηκαν τα καλώδια των βηματικών κινητήρων και στην συνέχεια τα καλώδια των limit switch.



Εικόνα 34: Διαδικασία Σύνδεσης Καλωδίων

8.3 Λίστα υλικών

Στη λίστα αυτήν αναγράφονται αναλυτικά όλα τα υλικά που αξιοποιήθηκαν για την κατασκευή. Αρχικά στον πίνακα 1 αναγράφονται όλα τα, απαραίτητα για τον ρομποτικό βραχίονα, εξαρτήματα τα οποία κατασκευάστηκαν με την βοήθεια 3D εκτυπωτή. Αναλυτικά αναφέρεται περιγραφή για κάθε ένα από αυτά, αριθμός τεμαχίων, βάρος σε γραμμάρια, ο χρόνος που χρειάστηκε να εκτυπωθεί το κάθε τεμάχιο και τέλος το κόστος του υλικού που χρησιμοποιήθηκε.

Αντικείμενο	Περιγραφή	Τεμάχια	Βάρος	Χρόνος Εκτύπωσης	Κόστος Υλικού
Base (Part 1)	1 ^ο Τμήμα της Βάσης του ρομποτικού Βραχίονα	1	142γρ.	4ω. 52λ.	4.07 €
Base (Part 2)	2 ^ο Τμήμα της Βάσης του ρομποτικού Βραχίονα	1	164γρ.	5ω. 39λ.	4.70 €
Joint 1 Coupler	Άρθρωση 1 ^η Ένωση Βάσης με τον πρώτο άξονα του ρομποτικού Βραχίονα	1	75γρ.	2ω. 34λ.	2.15 €
Z-Axis Bottom Plate	Βάση του Άξονα Z	1	54γρ.	1ω. 50λ.	1.54 €
Z-Axis Bottom Plate Cover	Καπάκι Βάσης του Άξονα Z	1	48γρ.	1ω. 35λ.	1.92 €
Z-Axis Mount Platform	Ένωση Βραχίονα με τον Άξονα Z	1	148.γρ	4ω. 58λ.	4.24 €
Z-Axis Top Plate	Κορυφή του Άξονα Z	1	52γρ.	1ω. 49λ.	1.50 €
Z-Axis Top Plate Cover	Καπάκι Κορυφής του Άξονα Z	1	50γρ.	1ω. 48λ.	1.42 €
Arm 1	1 ^ο Σύνδεσμος Βραχίονα	1	142γρ.	4ω. 38λ.	4.07 €
Arm 1 Cover	Καπάκι 1 ^{ου} Συνδέσμου Βραχίονα	1	47γρ.	1ω. 35λ.	1.34 €
Joint 2 Coupler	Άρθρωση 2 ^η Ένωση 1 ^{ου} Συνδέσμου του Βραχίονα με τον 2 ^ο	1	36γρ.	1ω. 45λ.	1 €
Arm 2	2 ^{ος} Σύνδεσμος Βραχίονα	1	126γρ.	4ω. 23λ.	3.62 €
Arm 2 Cover	Καπάκι 2 ^{ου} Συνδέσμου Βραχίονα	1	67γρ.	2ω. 16λ.	1.92 €
Joint 3 Coupler	Άρθρωση 3 ^η	1	30γρ.	1ω. 35λ.	0.86 €
Joint 3 to Gripper Connector	Ένωση 2 ^{ου} Συνδέσμου του Βραχίονα με το Άκρο	1	6.5γρ	0ω. 21λ.	0.20 €
Gripper Servo Holder	Βάση Στήριξης Servo στο άκρο	1	13.5γρ.	0ω. 47λ.	0.38 €
Gripper Cover	Καπάκι της Βάσης Στήριξης Servo	1	10γρ.	0ω. 33λ.	0.28 €
Gripper Link	1 ^ο Μέρος του Συνδέσμου	2	2γρ.	0ω. 4λ.	0.05 €

1	Μεταξύ Σέρβο και Μηχανισμό Άκρου				
Gripper Link 2	2 ^ο Μέρος του Συνδέσμου Μεταξύ Σέρβο και Μηχανισμό Άκρου	2	2γρ.	0ω. 4λ.	0.05 €
Gripper Mechanism Slider	Εξάρτημα Μηχανισμού Γλίστρας του Άκρου	2	10γρ.	0ω. 35λ.	0.28 €
Gripper Left Hand	Στήριξη Άκρου στον Μηχανισμό της Γλίστρας	2	7γρ.	0ω. 21λ.	0.21 €
Gripper Right Hand	Στήριξη Άκρου στον Μηχανισμό της Γλίστρας	2	7γρ.	0ω. 21λ.	0.21 €
Gripper End	Άκρο Εργασίας	2	8.2γρ.	0ω. 25λ.	0.23 €
Arduino Uno Case (Part 1)	Κάτω μέρος Κουτιού κάλυψης Arduino και CNC Board	1	50γρ.	2ω. 38λ.	1.42 €
Arduino Uno Case (Part 2)	Άνω μέρος Κουτιού κάλυψης Arduino και CNC Board	1	30γρ.	1ω. 45λ.	0.86 €
Smooth Rod Clamp	Σφιγκτήρας Ράβδων Άξονα Z	8	7γρ.	0ω. 22λ.	0.21 €
GT2 Pulley - 22 - 80 teeth	Διπλό Γρανάζι 22-80 Δοντιών	1	12γρ.	0ω. 41λ.	0.34 €
GT2 Pulley - 23 - 80 teeth	Διπλό Γρανάζι 23-80 Δοντιών	1	12γρ.	0ω. 42λ.	0.34 €
GT2 Pulley - 90 teeth - J1	Γρανάζι 90 Δοντιών 1 ^{ης} Άρθρωσης	1	18γρ.	1ω. 0λ.	0.51 €
GT2 Pulley - 92 teeth - J2	Γρανάζι 92 Δοντιών 2 ^{ης} Άρθρωσης	1	18γρ.	1ω. 5λ.	0.51 €
GT2 Pulley - 110 teeth - J3	Γρανάζι 110 Δοντιών 3 ^{ης} Άρθρωσης	1	27γρ.	1ω. 24λ.	0.80 €
GT2 Pulley - Parametric	Γρανάζι Βηματικών Κινητήρων	3	2γρ.	0ω. 8λ.	0.05 €

Πίνακας 1 : 3D Printed Εξαρτήματα του Ρομποτικού Βραχίονα

Εν συνεχεία στον Πίνακα 2 αναγράφονται όλα τα, απαραίτητα για τον ρομποτικό βραχίονα, ηλεκτρονικά εξαρτήματα. Αναλυτικά αναφέρεται περιγραφή για το κάθε ένα από αυτά, αριθμός τεμαχίων, το κόστος ανά 1 τεμάχιο, καθώς και πληροφορίες προμηθευτή με το αντίστοιχο link.

Αντικείμενο	Περιγραφή	Τεμάχια	Κόστος Τεμαχίου	Προμηθευτής
Arduino UNO	Πλακέτα Arduino	1	11.90 €	Grobotronics https://grobotronics.com/funduino-uno-rev3-arduino-uno-compatible.html
CNC Shield		1	6.10 €	Cableworks

V3				https://www.cableworks.gr/ilektronika/arduino-and-microcontrollers/motors/drivers/v3-engraver-3d-printer-cnc-shield-and-expansion-board-a4988-driver-for-arduino/
Stepper Motor Drivers	Drivers Βηματικών Κινητήρων	4	3.92 €	Cableworks https://www.cableworks.gr/ilektronika/arduino-and-microcontrollers/motors/stepper/3d-printer-a4988-reprap-stepper-motor-driver-module/
Stepper Motor	Βηματικοί Κινητήρες	4	7.6 €	Alixpress 5pcs/lot 3D Printer 4 lead Nema17 Stepper Motor 42 motor Nema17 motor 42BYGH 1.5A (17HS4401S) motor for Ender 3 TT S Stepper Motor - AliExpress
Limit-Switch KW12-3	Τερματικός Διακόπτης Άρθρωσης	4	0.45 €	Cableworks https://www.cableworks.gr/ilektronika/arduino-and-microcontrollers/components-and-ic/limit-switch-kw12-3-micro-roller-lever-arm-normally-open-close/
Servo MG996R	Σέρβο	1	10.40 €	https://grobotronics.com/servo-standard-11kg.cm-metal-gears-waveshare-mg996r.html
Female Crimp For Dupont Terminal	Θηλυκό Pin	11	0.04 €	Grobotronics https://grobotronics.com/crimp-for-dupont-female.html
Dupont Terminal 2-Pin	Τερματικό 2 Pin	4	0.05 €	Grobotronics https://grobotronics.com/terminal-female-2-pins-2.54mm-dupont.html
Dupont Terminal 1-Pin	Τερματικό 1 Pin	3	0.05 €	Grobotronics https://grobotronics.com/terminal-female-1-pins-2.54mm-dupont.html
Cable 0.35mm 1m	Καλώδιο	6	0.20 €	Grobotronics https://grobotronics.com/wire-stranded-0.35mm2-grey-black.html

Πίνακας 2 : Ηλεκτρονικά Εξαρτήματα Ρομποτικού Βραχίονα

Τέλος στον πίνακα 3 αναγράφονται όλα τα, απαραίτητα για τον ρομποτικό βραχίονα, μηχανικά εξαρτήματα. Αναλυτικά αναφέρεται περιγραφή για το κάθε ένα από αυτά, αριθμός τεμαχίων, το κόστος ανά 1 τεμάχιο, καθώς και πληροφορίες προμηθευτή με το αντίστοιχο link

Αντικείμενο	Περιγραφή	Τεμάχια	Κόστος Τεμαχίου	Προμηθευτής
Thrust Ball Bearing 40x60x13	Ωστικό Ρουλεμάν	2	3.25 €	
Thrust Ball Bearing 35x52x12	Ωστικό Ρουλεμάν	4	2.45 €	
Radial Ball Bearing 61807 2RS	Ρουλεμάν	1	3.27 €	
Radial Ball Bearing 61806 2RS	Ρουλεμάν	2	5.00 €	
Radial Ball Bearing 608 2RS	Ρουλεμάν	5	0.30 €	
Linear Ball Bearing LM10UU	Γραμμικό Ρουλεμάν	5	1.60 €	https://grobotronics.com/linear-ball-bearing-10mm-diameter-lm10uu.html
M8X50mm Hex.	Βίδα εξάγωνη διαμέτρου 8mm και μήκους 50mm	2	0.25 €	
M5x40mm Φρεζάτη	Βίδα φρεζάτη διαμέτρου 5mm και μήκους 40mm	2	0.035	
M5x35mm Φρεζάτη	Βίδα φρεζάτη διαμέτρου 5mm και μήκους 35mm	2	0.035 €	
M5x30mm Φρεζάτη	Βίδα φρεζάτη διαμέτρου 5mm και μήκους 30mm	2	0.030 €	
M5x20mm Φρεζάτη	Βίδα φρεζάτη διαμέτρου 5mm και μήκους	4	0.025 €	

	20mm			
M4x60mm Φρεζάτη	Βίδα φρεζάτη διαμέτρου 4mm και μήκους 60mm	4	0.060 €	
M4x55mm Φρεζάτη	Βίδα φρεζάτη διαμέτρου 4mm και μήκους 55mm	4	0.055 €	
M4x50mm Φρεζάτη	Βίδα φρεζάτη διαμέτρου 4mm και μήκους 50mm	4	0.050 €	
M4x35mm Hex.	Βίδα εξάγωνη διαμέτρου 4mm και μήκους 35mm	8	0.025 €	
M4x25mm Φρεζάτη	Βίδα φρεζάτη διαμέτρου 4mm και μήκους 25mm	10	0.020 €	
M4x25mm Hex.	Βίδα εξάγωνη διαμέτρου 4mm και μήκους 25mm	4	0.020 €	
M4x20mm Hex.	Βίδα εξάγωνη διαμέτρου 4mm και μήκους 20mm	22	0.015 €	
M4x16mm Φρεζάτη	Βίδα φρεζάτη διαμέτρου 4mm και μήκους 16mm	4	0.010 €	
M3x30mm	Βίδα πομπέ	8	0.020 €	

Πομπέ	διαμέτρου 3mm και μήκους 30mm			
M3x20mm Πομπέ	Βίδα πομπέ διαμέτρου 3mm και μήκους 20mm	4	0.015 €	
M3x12mm Πομπέ	Βίδα πομπέ διαμέτρου 3mm και μήκους 12mm	16	0.010 €	
M3x12mm Φρεζάτη	Βίδα φρεζάτη διαμέτρου 3mm και μήκους 12mm	4	0.010 €	
M3x10mm Πομπέ	Βίδα πομπέ διαμέτρου 3mm και μήκους 10mm	8	0.010 €	
Ροδέλες M8	Ροδέλα 8mm	2	0.01 €	
Ροδέλες M5	Ροδέλα 5mm	20	0.01 €	
Ροδέλες M3	Ροδέλα 3mm	16	0.01 €	
Παξιμάδια M8	Παξιμάδι 8mm	2	0.020 €	
Παξιμάδια M5	Παξιμάδι 5mm	22	0.015 €	
Παξιμάδια M4	Παξιμάδι 4mm	60	0.010 €	
Παξιμάδια M3	Παξιμάδι 3mm	26	0.010 €	
Κλειστός Ιμάντας 400mm	Κλειστός ιμάντας διαμέτρου 400mm	2	2.20 €	https://grobotronics.com/timing-belt-gt2-400mm.html
Κλειστός Ιμάντας 300mm	Κλειστός ιμάντας διαμέτρου 300mm	2	1.90 €	https://fuselab.gr/product/gt2-closed-loop-timing-belt-rubber-300mm/
Κλειστός Ιμάντας 200mm	Κλειστός ιμάντας διαμέτρου	1	1.40 €	https://grobotronics.com/timing-belt-gt2-200mm.html

	200mm			
Rod Shaft 10mm 1m	Ράβδος διαμέτρου 10mm και μήκους 1m	2	7.50 €	https://gamvrinos.com/el/
Rod Shaft 6mm 1m	Ράβδος διαμέτρου 6mm και μήκους 1m	1		https://gamvrinos.com/el/
Lead Screw 8mm- 500mm.	Σπειρωτή ντίζα διαμέτρου 8mm και μήκους 500mm	1	9.90 €	https://grobotronics.com/lead-screw-t8-200mm-clone.html
Lead Screw Nut.	Περικόγλιο σπειρωτής ντίζα διαμέτρου 8mm.	1	1.60 €	https://grobotronics.com/nut-for-lead-screw-t8-brass.html

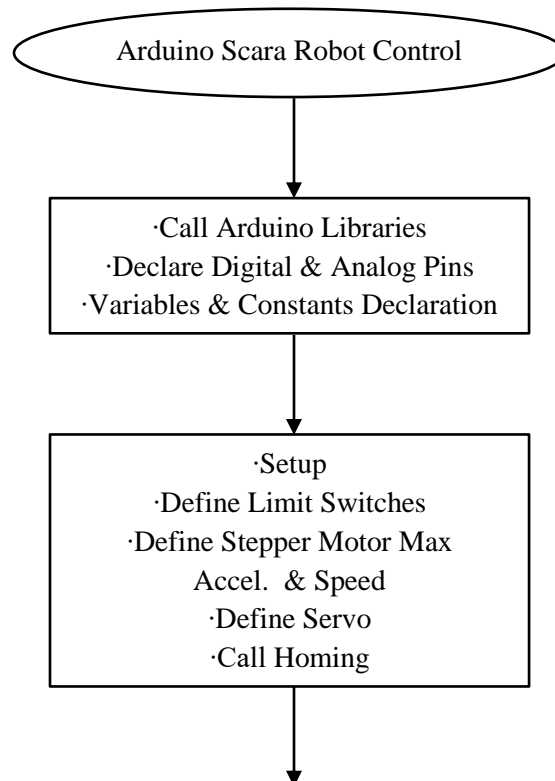
Πίνακας 3 : Μηχανικά Εξαρτήματα Ρομποτικού Βραχίονα

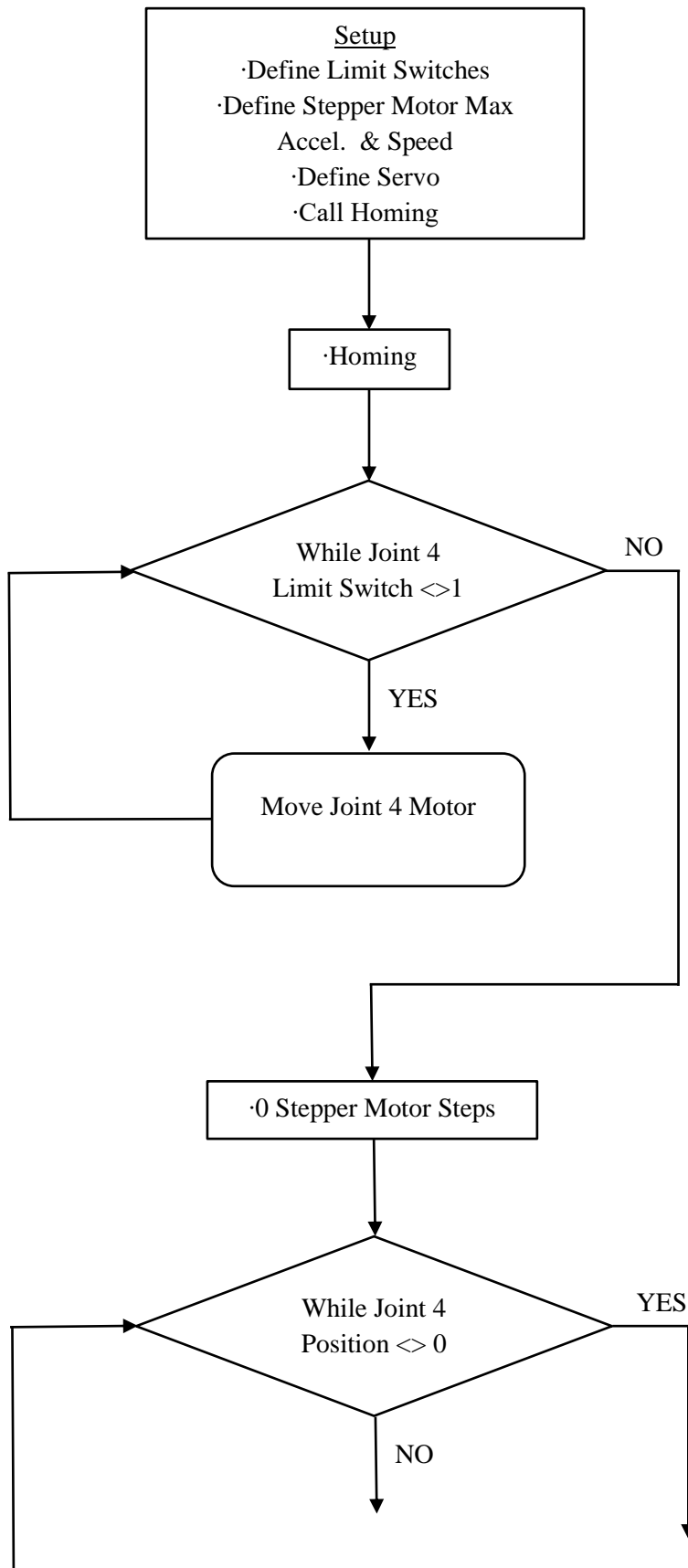
9. ΑΝΑΛΥΣΗ ΚΑΙ ΣΥΓΓΡΑΦΗ ΛΟΓΙΣΜΙΚΟΥ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ

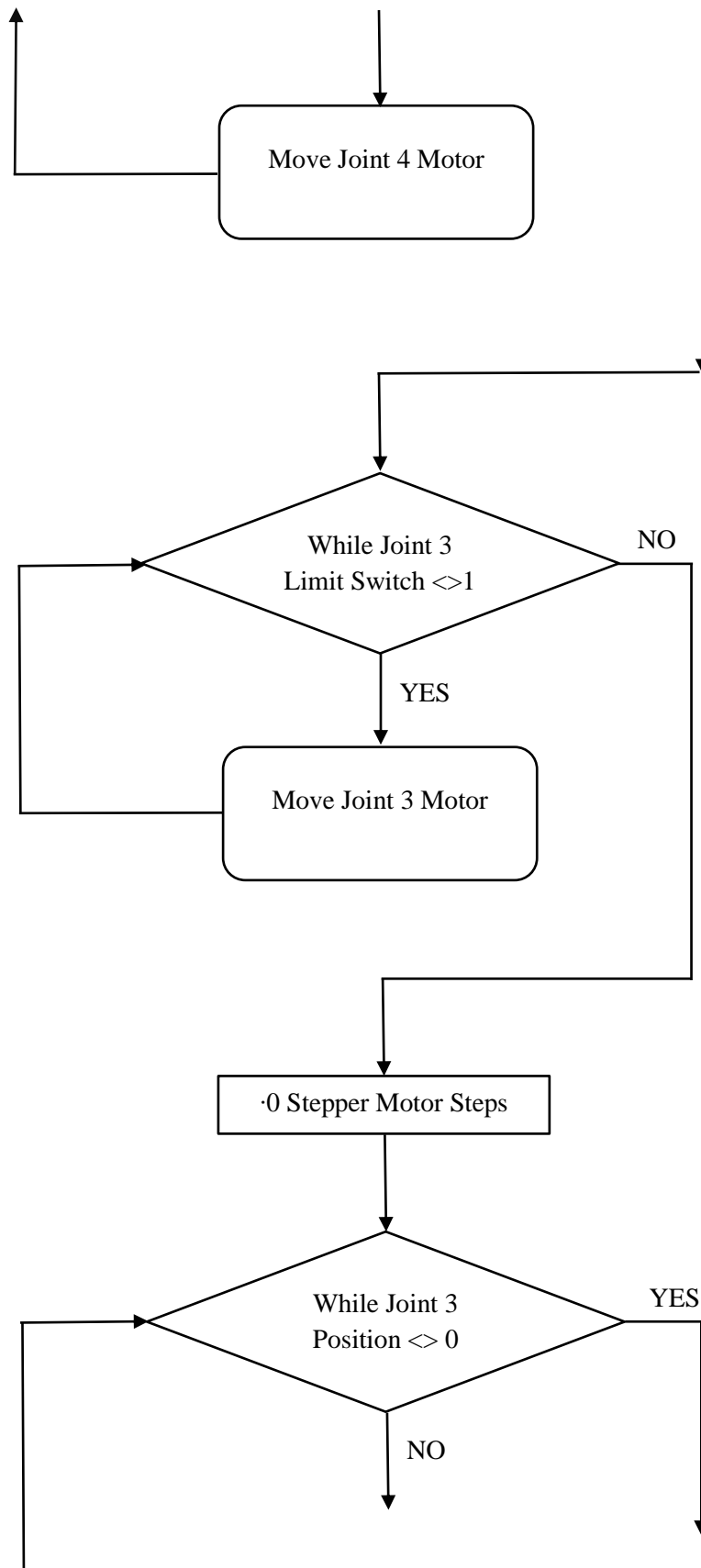
Στην ενότητα αυτήν και 9^η της παρούσης διπλωματικής εργασίας, γίνεται αναλυτική επεξήγηση και ο σχολιασμός του τρόπου με τον οποίο σχεδιάστηκε και συγγράφηκε ο κώδικας ελέγχου του ρομποτικού βραχίονα καθώς και ο κώδικας με τον οποίο υλοποιήθηκε το γραφικό περιβάλλον ελέγχου του ρομποτικού βραχίονα. Πιο αναλυτικά παρουσιάζεται το διάγραμμα ροής βάση του οποίου υλοποιήθηκε το λογισμικό, δηλαδή η λογική η οποία ακολουθεί ο κώδικας ελέγχου του ρομπότ και στην συνέχεια αυτούσιος ο κώδικας ελέγχου αλλά και ο κώδικας του γραφικού περιβάλλοντος με εκτεταμένο σχολιασμό.

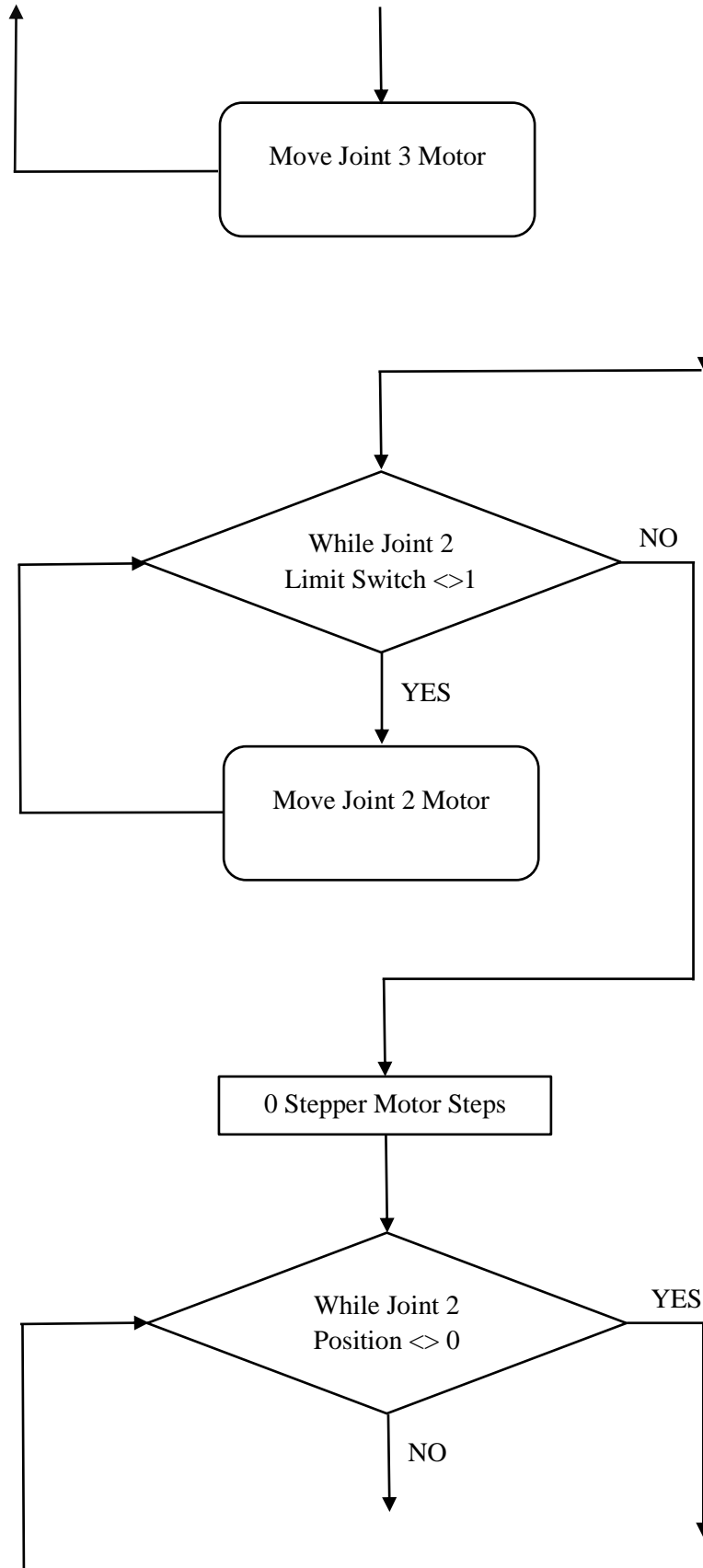
Διάγραμμα Ροής Κώδικα Arduino

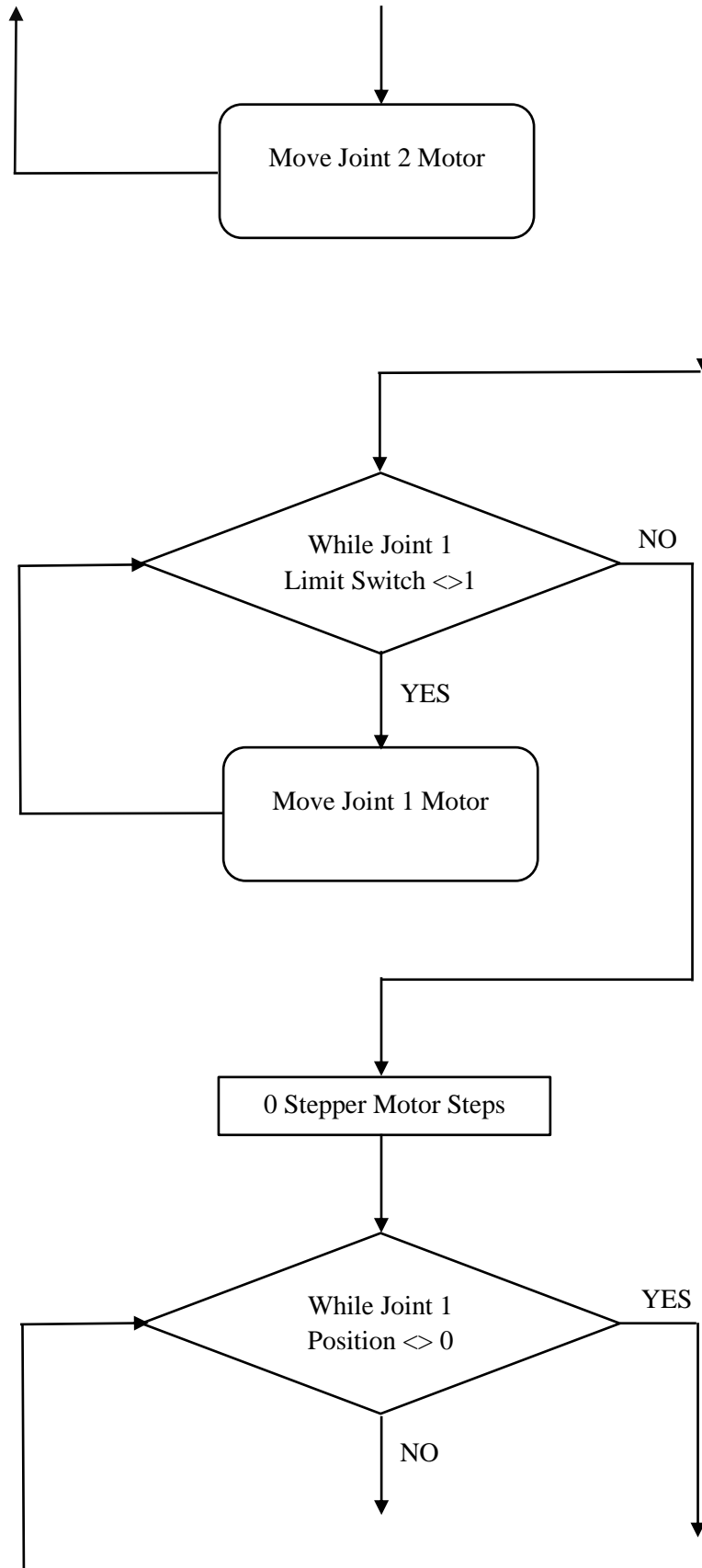
Το **διάγραμμα ροής** ή **flow chart** του προγράμματος που φέρει ο μικροελεγκτής της διπλωματικής εργασίας αποτυπώνει τη βασική δομή του "τρόπου σκέψης" του. Περιλαμβάνει όλες τις λειτουργίες του ρομποτικού βραχίονα, χωρίς ωστόσο να εισέρχεται σε εξειδικευμένες λεπτομέρειες. Αυτές αποτυπώνονται πλήρως από τον κώδικα του προγράμματος. Για το σχεδιασμό του διαγράμματος ροής χρησιμοποιήθηκε το πρόγραμμα Word.

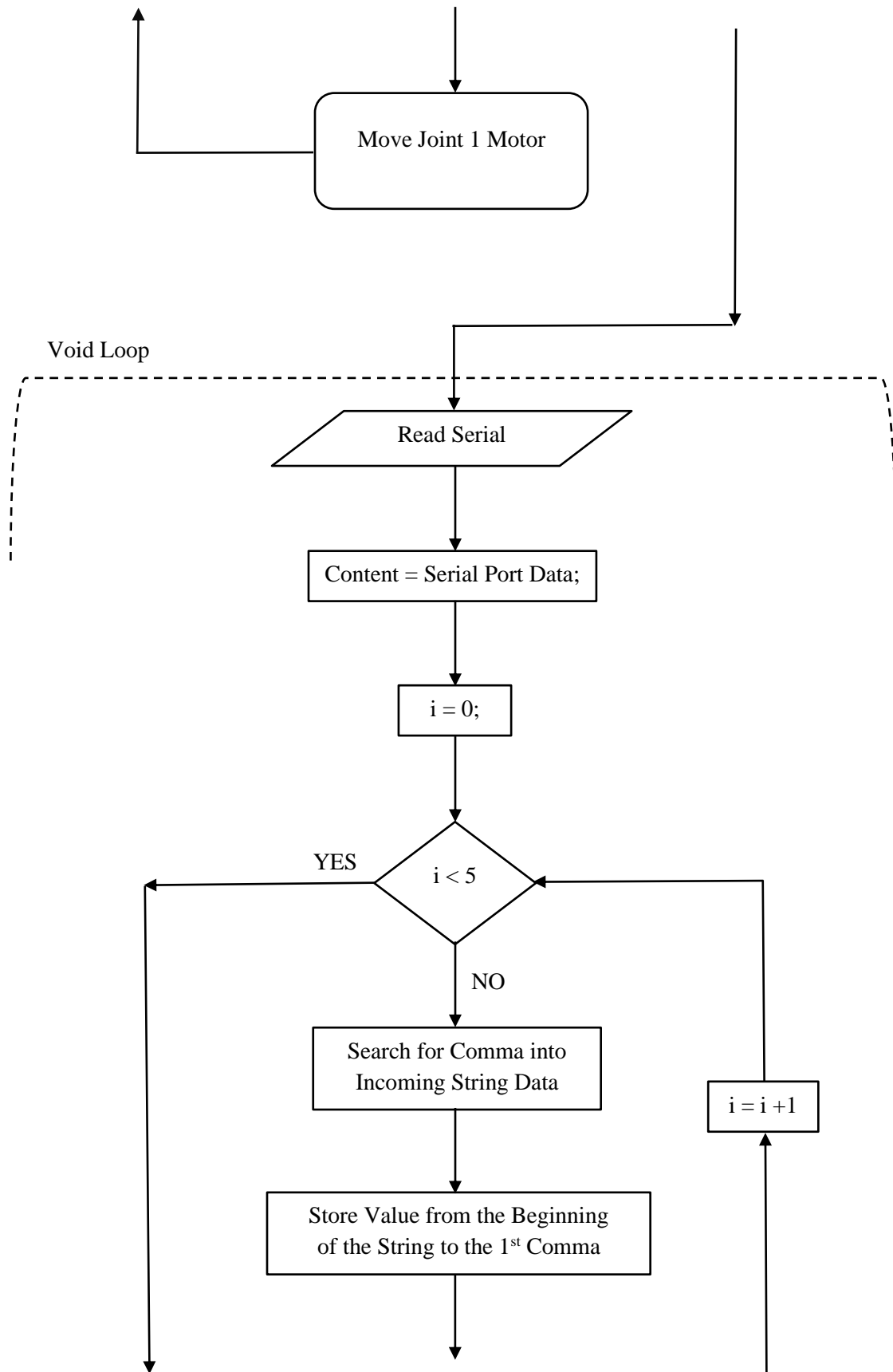


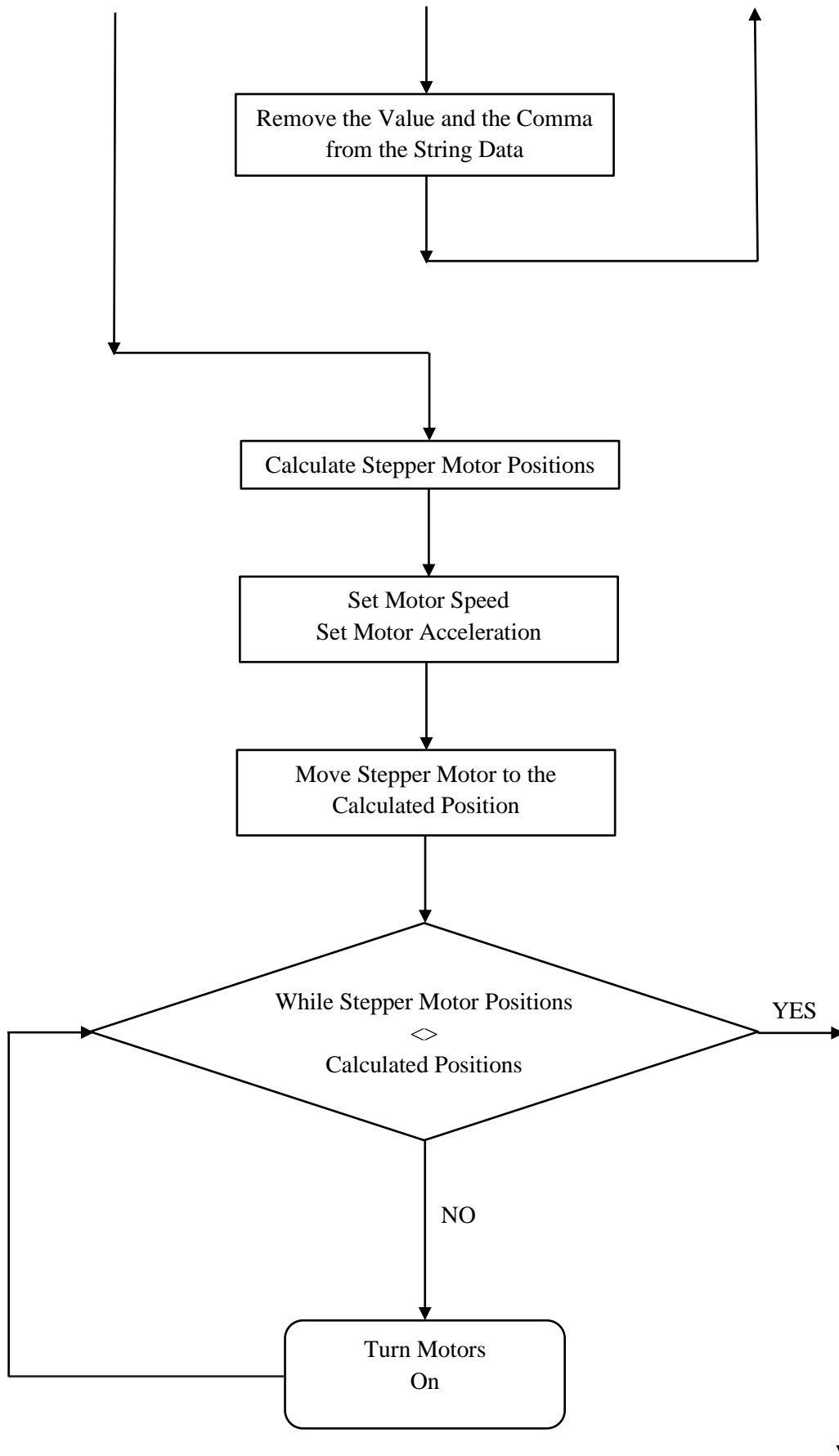


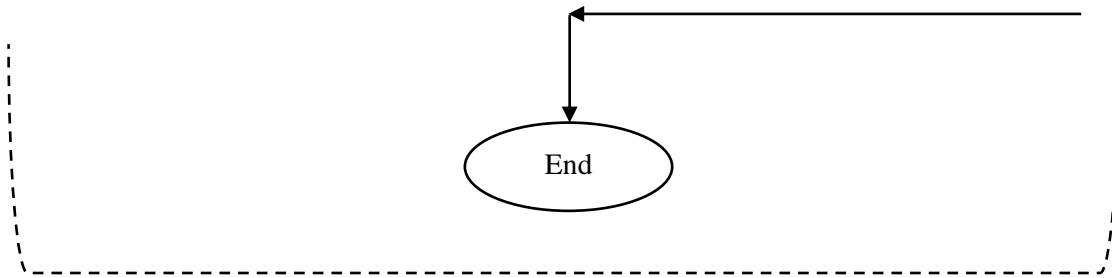




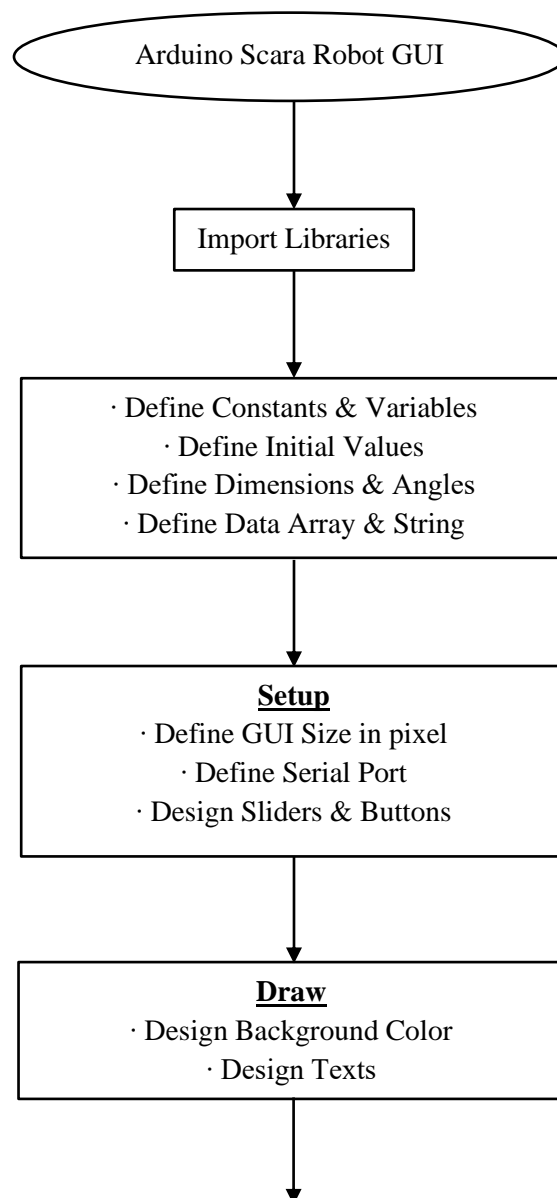


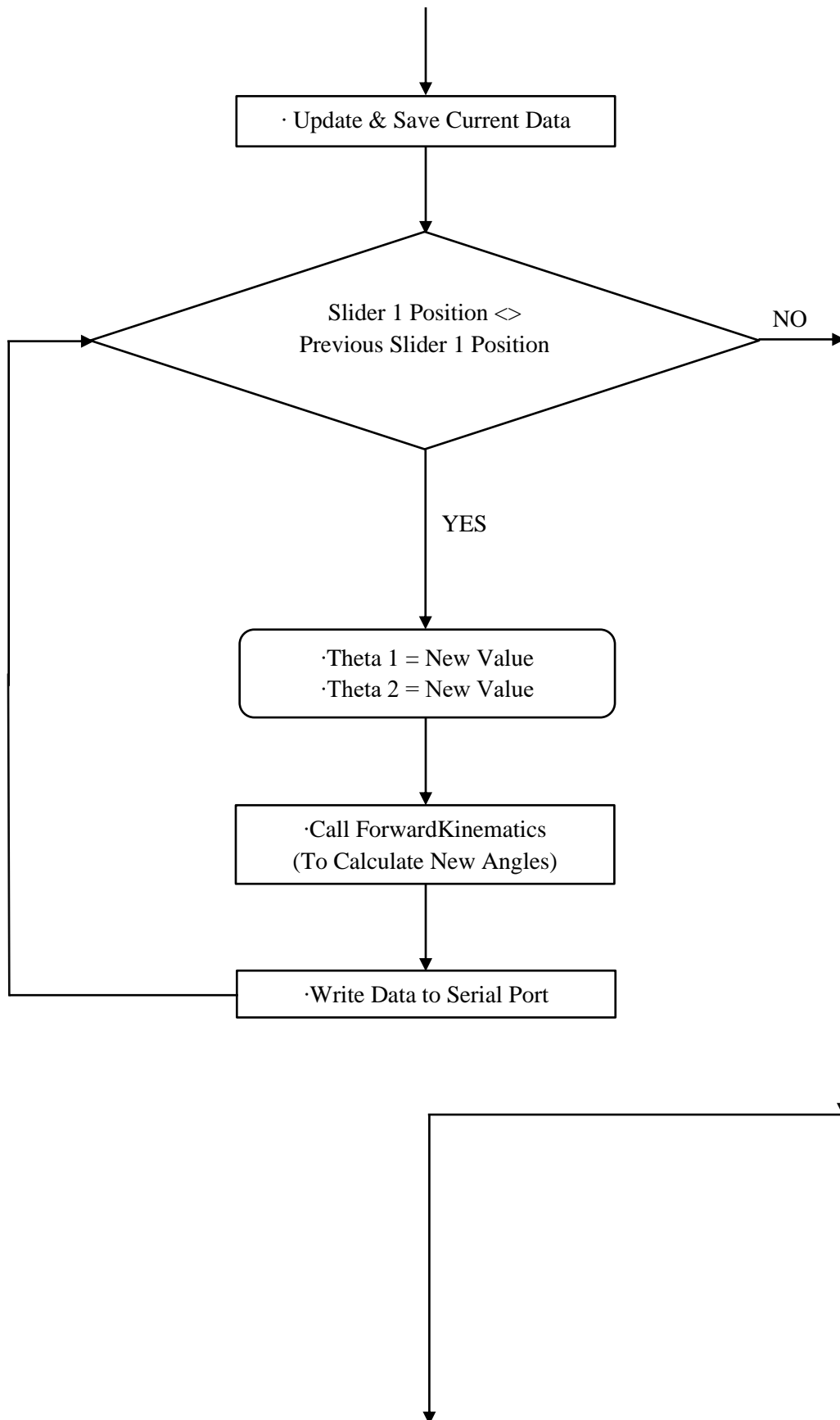


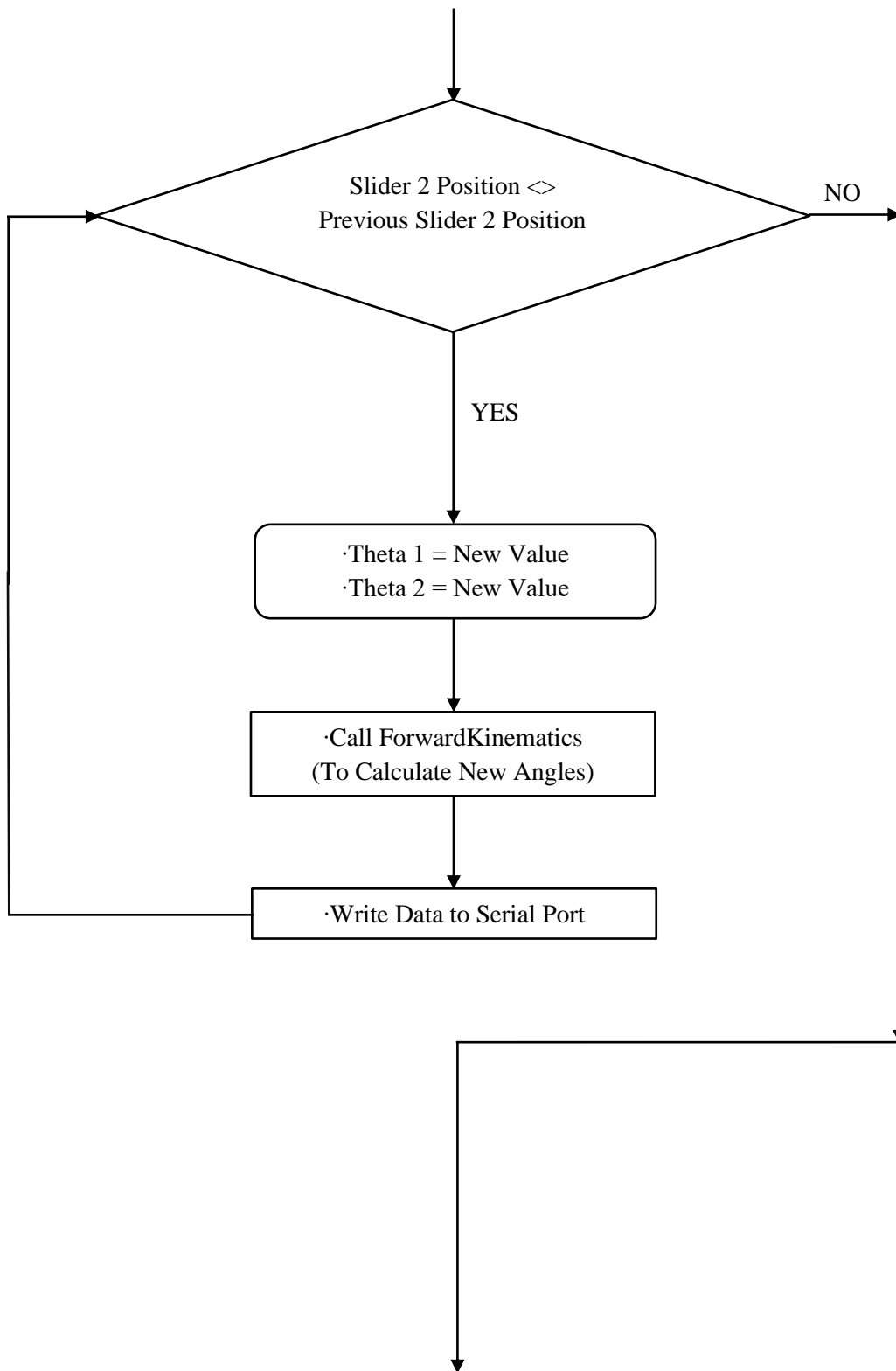


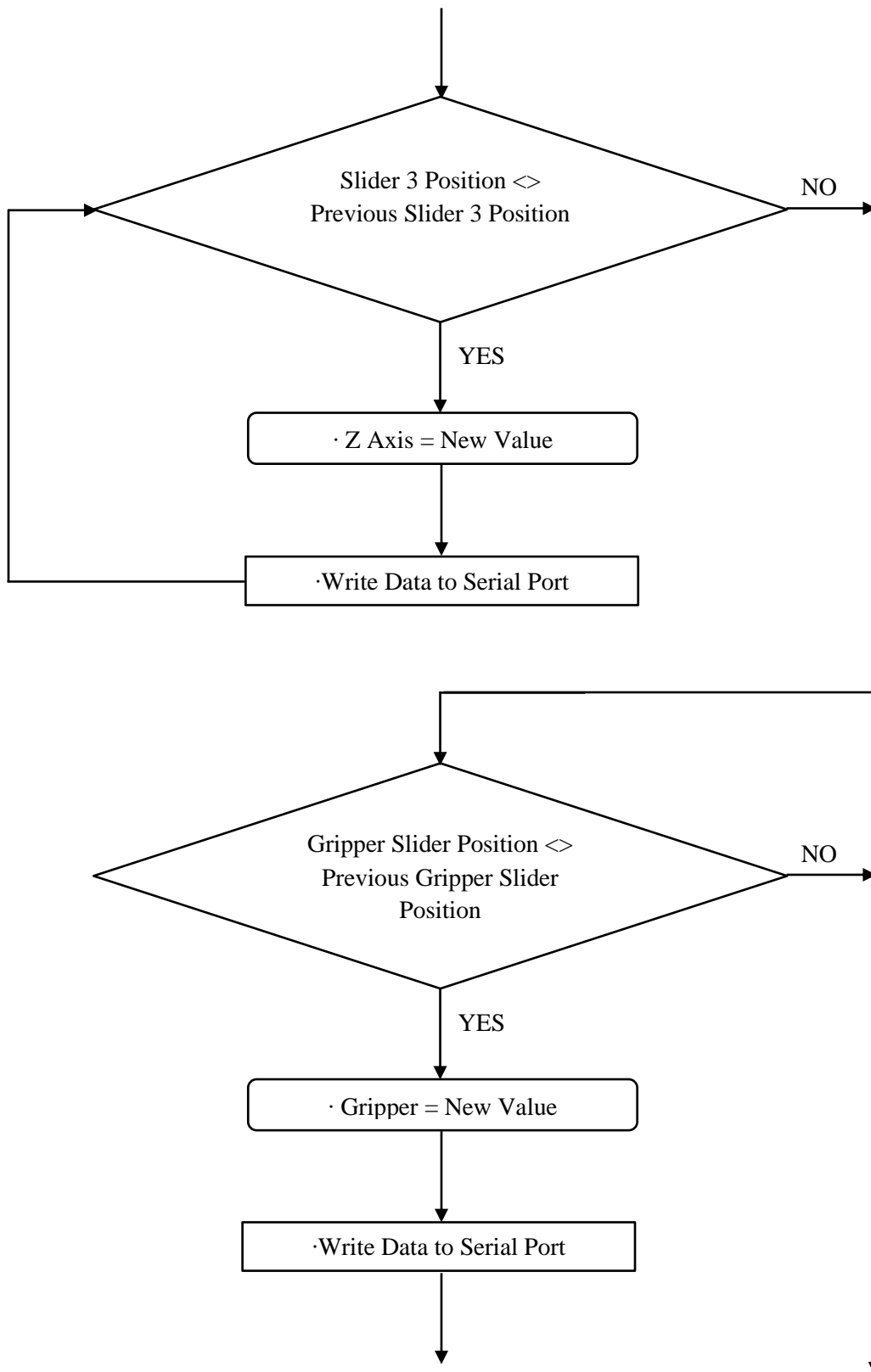


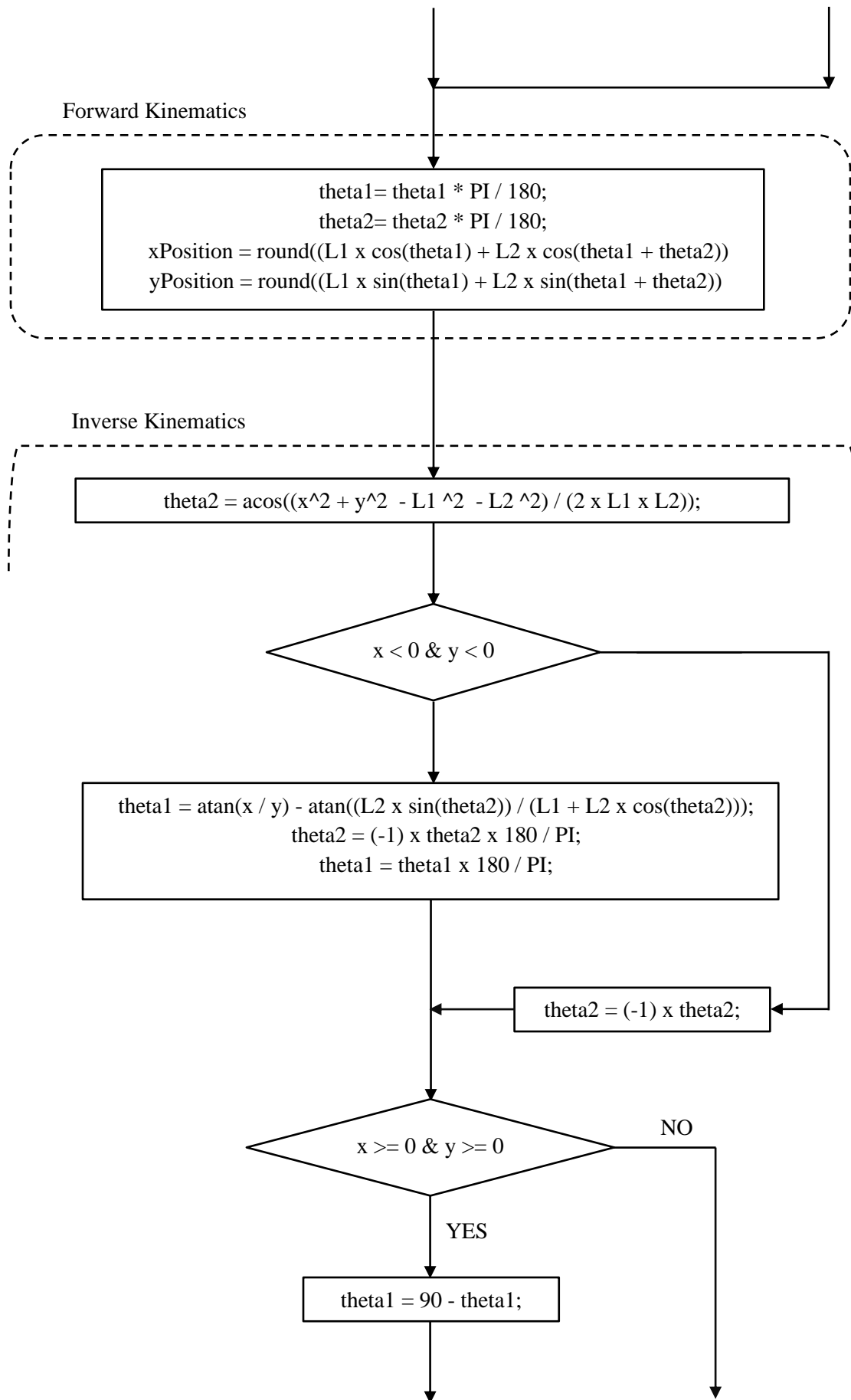
Διάγραμμα Ροής Κώδικα Processing

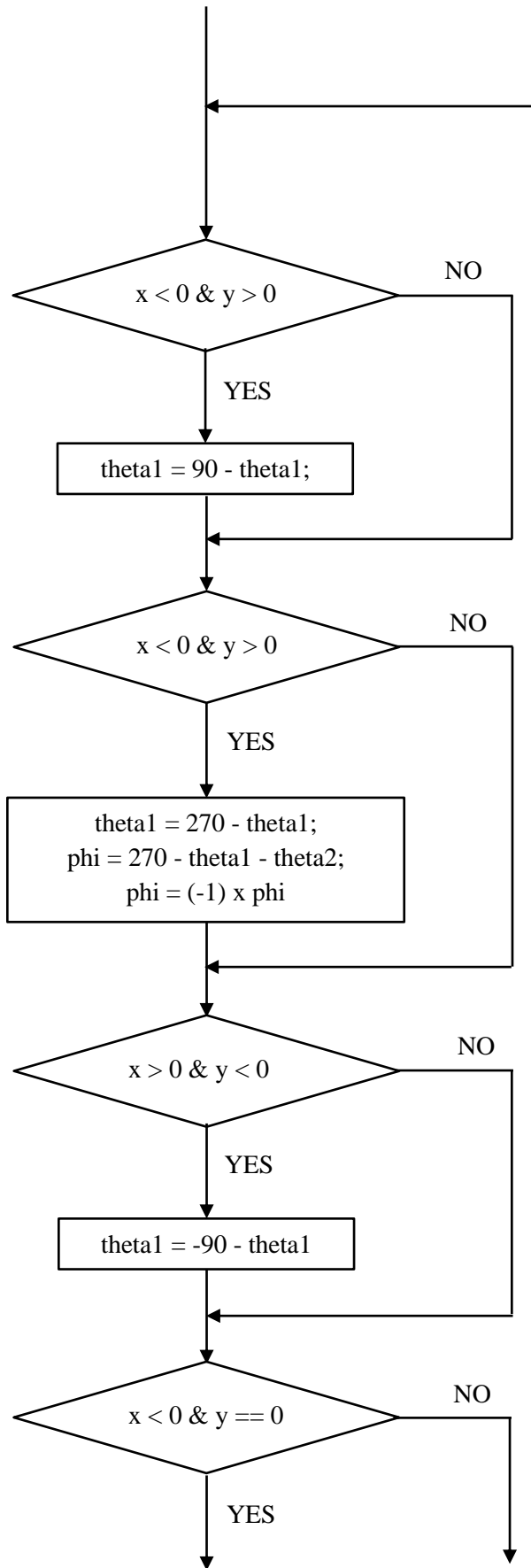


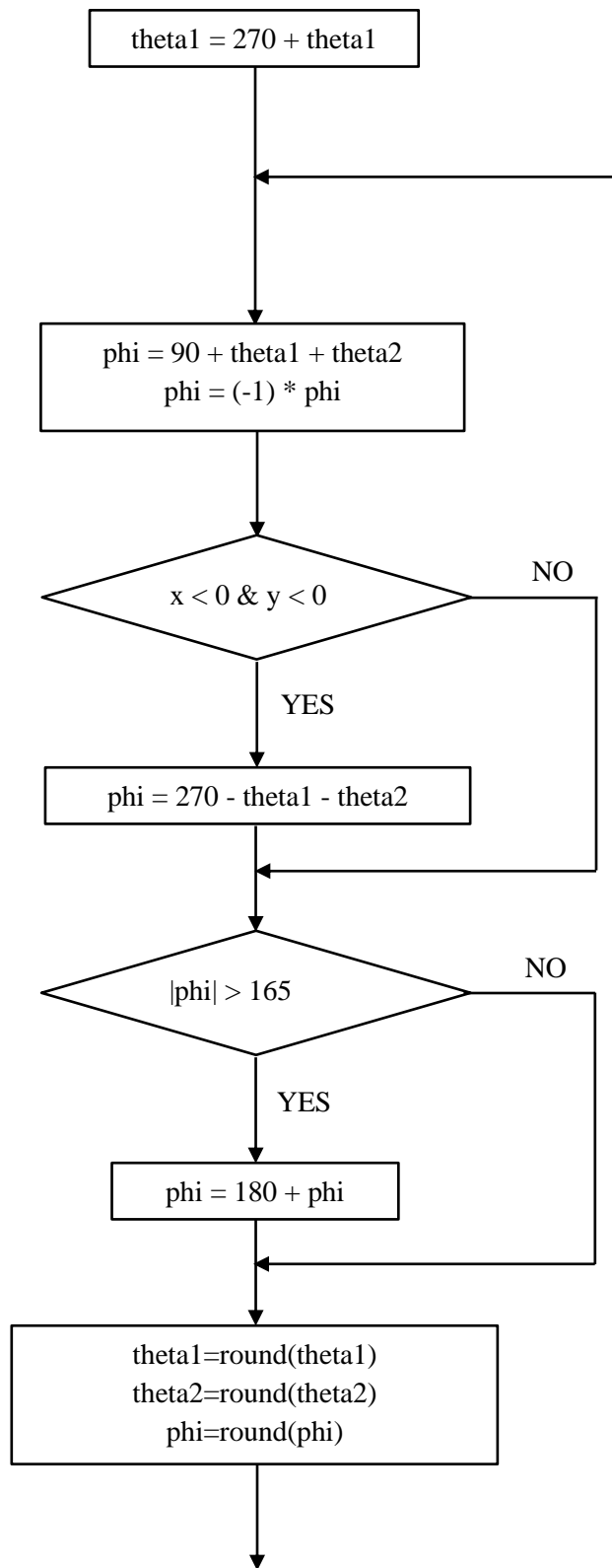


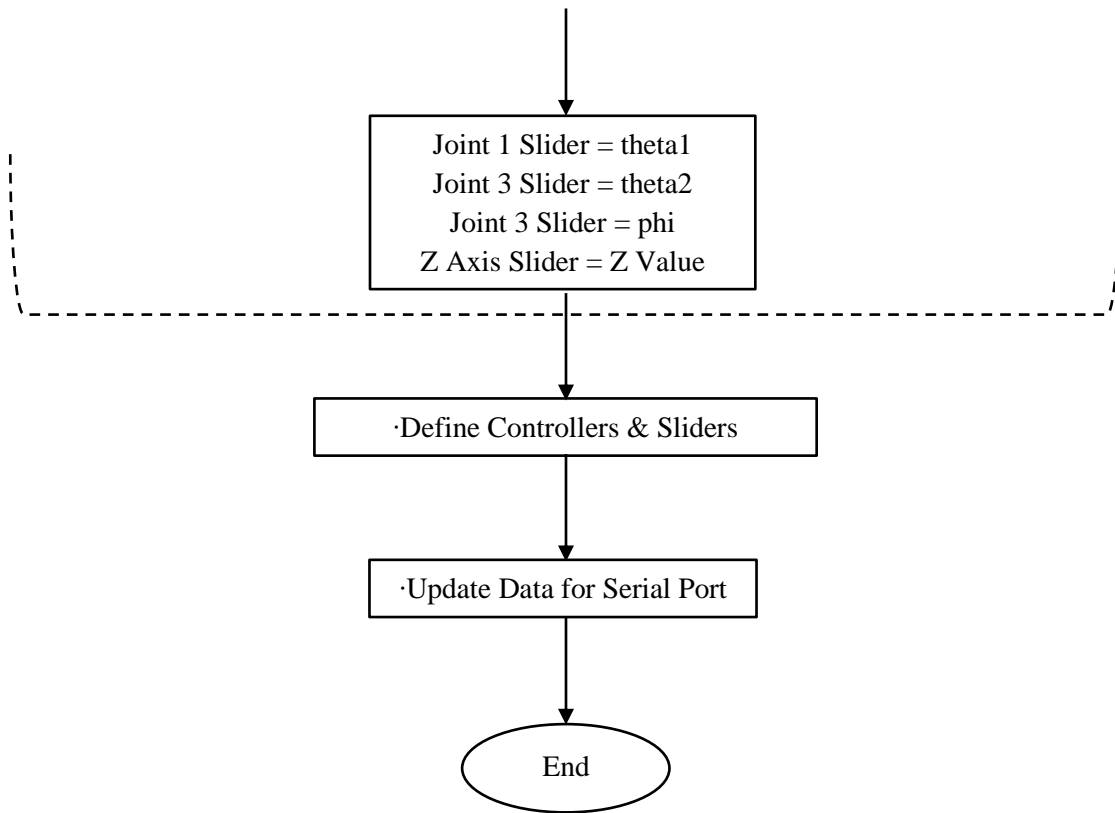












Κώδικας Προγράμματος Arduino

Παρακάτω ακολουθεί ο κώδικας του μικροελεγκτή Arduino Uno που χρησιμοποιήθηκε στην συγκεκριμένη εφαρμογή, ώστε να υλοποιηθεί ο έλεγχος του ρομποτικού βραχίονα. Συνοπτικά ο κώδικας που γράφτηκε, λαμβάνει τα δεδομένα τα οποία στέλνονται μέσω της σειριακής θύρας επικοινωνίας από το Interface της Processing. Στην συνέχεια αποθηκεύονται και μετατρέπονται σε κατανοητές για το ρομπότ εντολές ώστε να υλοποιηθούν οι αντίστοιχες επιθυμητές κινήσεις.

```
/****** Scara Robot Controller
* Purpose
*   The microcontroller (Arduino) controls the stepper angles by reading string
*   data that are being sent to its serial port from the processing GUI.
* Hardware
*   CNC Shield v3 is connected to Arduino Uno v3.
*   A4988 Steppers Drivers are connected to the CNC Shield stepper ports
*   End Stop switches are connected on the CNC Shield Pins. Signal outputs of
*   the end switch are connected at X+, Y+, Z+ and CoolEn pin of the CNC Shield
*   and the ground is connected at the GND Pins of the shield.
*   Servo supply pins are connected directly to the power supply and the
*   control pin is connected at the CNC Shield Pin
* Software
*   Uses Arduino Standard Library Calls analogRead(),digitalRead(),
*   Serial.readString, Serial.available().
*   Uses Arduino Library Calls AccelStepper.h, Servo.h, math.h
* Reference
*   v2.0, I. Antonakis, Dec. 2022.
******/

//Call Arduino Libraries.
#include <AccelStepper.h>
#include <Servo.h>
#include <math.h>

// Define Limitswitches Digital & Analog Pins.
#define limitSwitch1 11 //Define the 1st Limit Switch.
#define limitSwitch2 10 //Define the 2nd Limit Switch.
#define limitSwitch3 9 //Define the 3rd Limit Switch.
#define limitSwitch4 A3 //Define the 4th Limit Switch.

//Define Stepper Motor Pins
AccelStepper stepper1(1, 2, 5); //Define 1st Stepper Motor Drive Pins (Driver,
STEP, DIR).
AccelStepper stepper2(1, 3, 6); //Define 2nd Stepper Motor Drive Pins (Driver,
STEP, DIR).
AccelStepper stepper3(1, 4, 7); //Define 3rd Stepper Motor Drive Pins (Driver,
STEP, DIR).
AccelStepper stepper4(1, 12, 13); //Define 4th Stepper Motor Drive Pins (Driver,
STEP, DIR).

Servo gripperServo; //Define Servo.

//Define Scara Robot Arm Dimensions.
```

```

double x = 10.0;
double y = 10.0;
double L1 = 228; // L1 = 228mm
double L2 = 136.5; // L2 = 136.5mm
double theta1, theta2, phi, z; //Define Stepper Motor Angles.

//Define Stepper Motors Position Variables.
int stepper1Position, stepper2Position, stepper3Position, stepper4Position;

//Define Constants For Converting Axis Angles to Stepper Motor Steps.
const float theta1AngleToSteps = 44.444444;
const float theta2AngleToSteps = 35.555555;
const float phiAngleToSteps = 10;
const float zDistanceToSteps = 100;

//Define String For Reading Serial Port's Data.
String content = "";

//Define Array For Storing Data Received From Processing GUI .
int data[10];

void setup() {
  Serial.begin(115200); //Define Serial Port's Baud Rate.

  //Define Limit Switch Use
  pinMode(limitSwitch1, INPUT_PULLUP);
  pinMode(limitSwitch2, INPUT_PULLUP);
  pinMode(limitSwitch3, INPUT_PULLUP);
  pinMode(limitSwitch4, INPUT_PULLUP);

  //Define Stepper Motors Max Speed & max Acceleration.
  stepper1.setMaxSpeed(4000); //Define 1st Stepper Motor Max Speed.
  stepper1.setAcceleration(2000); //Define 1st Stepper Motor Max Acceleration.
  stepper2.setMaxSpeed(4000); //Define 2nd Stepper Motor Max Speed.
  stepper2.setAcceleration(2000); //Define 2nd Stepper Motor Max Acceleration.
  stepper3.setMaxSpeed(4000); //Define 3rd Stepper Motor Max Speed.
  stepper3.setAcceleration(2000); //Define 3rd Stepper Motor Max Acceleration.
  stepper4.setMaxSpeed(4000); //Define 4th Stepper Motor Max Speed.
  stepper4.setAcceleration(2000); //Define 4th Stepper Motor Max Acceleration.

  //Define Servo Pins And Data.
  gripperServo.attach(A0, 600, 2500); //Define Servo Pins (Pin, Minimum Pulse
  Width in ms, Max Pulse Width in ms).
  data[4] = 180; //Set Set Initial Position - Open
  Gripper.
  gripperServo.write(data[4]); //Send Servo Initial Position Data.
  delay(1000); //Delay 1 Sec For Data To be Sent.
  data[3] = 100; //Set Z Axis Initial Height.

  //Call Homing Routine For Setting Home Axes Positions.
  homing();
}

void loop() {

  if (Serial.available()) {
    content = Serial.readString(); //Read Serial Port's Incoming Data From
    Processing GUI.
    Serial.readString();
    //Split String into Integer Data and Store into the Array.
    for (int i = 0; i < 5; i++) {
      int index = content.indexOf(","); //Search for the First Comma "," into
      String.

```

```

        data[i] = atol(content.substring(0, index).c_str()); //Save Value from
String's Begging to the First Comma ",".
        content = content.substring(index + 1); //Remove the previous Value and the
Comma "," from String.
    }
}

//Calculate Stepper Motor Positions (Steps) According to the Array Data.
stepper1Position = data[0] * theta1AngleToSteps; //Calculate 1st Stepper Motor
Position
stepper2Position = data[1] * theta2AngleToSteps; //Calculate 2nd Stepper Motor
Position
stepper3Position = data[2] * phiAngleToSteps; //Calculate 3rd Stepper Motor
Position
stepper4Position = data[3] * zDistanceToSteps; //Calculate 4th Stepper Motor
Position

//Define Stepper Motors Starting Speed
stepper1.setSpeed(2000); //Define 1st Stepper Motor Starting Speed
stepper2.setSpeed(2000); //Define 2nd Stepper Motor Starting Speed
stepper3.setSpeed(2000); //Define 3rd Stepper Motor Starting Speed
stepper4.setSpeed(2000); //Define 4th Stepper Motor Starting Speed

//Define Stepper Motors Starting Acceleration
stepper1.setAcceleration(2000); //Define 1st Stepper Motor Starting
Acceleration
stepper2.setAcceleration(2000); //Define 2nd Stepper Motor Starting
Acceleration
stepper3.setAcceleration(2000); //Define 3rd Stepper Motor Starting
Acceleration
stepper4.setAcceleration(2000); //Define 4th Stepper Motor Starting
Acceleration

//Move Stepper Motors to the Positions Calculated Previously
stepper1.moveTo(stepper1Position);
stepper2.moveTo(stepper2Position);
stepper3.moveTo(stepper3Position);
stepper4.moveTo(stepper4Position);

while (stepper1.currentPosition() != stepper1Position ||
stepper2.currentPosition() != stepper2Position || stepper3.currentPosition() !=
stepper3Position || stepper4.currentPosition() != stepper4Position) {
    stepper1.run();
    stepper2.run();
    stepper3.run();
    stepper4.run();

//Activate Stepper Motors While their Current Position is Different than the
Desired.

}
delay(100);
gripperServo.write(100);
delay(300);
}

void serialFlush() {
    while (Serial.available() > 0) { //While there are characters in the serial
buffer, because Serial.available is bigger than 0
        Serial.read(); // Read one character
    }
}
}

```

```

//Define Homing Routine
void homing() {
  //4th Stepper Motor Homing Procedure
  while (digitalRead(limitSwitch4) != 1) { //Run While Limitswitch 4 is not
pressed
    stepper4.setSpeed(1500);
    stepper4.runSpeed();
    stepper4.setCurrentPosition(17000); // When the Limit Switch is Pressed Set
4th Stepper Motors Position to 0 Steps
  }
  delay(20);
  stepper4.moveTo(10000);
  while (stepper4.currentPosition() != 10000) {
    stepper4.run();
  }

  //3rd Stepper Motor Homing Procedure
  while (digitalRead(limitSwitch3) != 1) { //Run While Limitswitch 3 is not
pressed
    stepper3.setSpeed(-1100);
    stepper3.runSpeed();
    stepper3.setCurrentPosition(-1662); // When the Limit Switch is Pressed Set
3rd Stepper Motors Position to 0 Steps
  }
  delay(20);

  stepper3.moveTo(0);
  while (stepper3.currentPosition() != 0) {
    stepper3.run();
  }

  //2nd Stepper Motor Homing Prosedure
  while (digitalRead(limitSwitch2) != 1) { //Run While Limitswitch 2 is not
pressed
    stepper2.setSpeed(-1300);
    stepper2.runSpeed();
    stepper2.setCurrentPosition(-5420); // When the Limit Switch is Pressed Set
2nd Stepper Motors Position to 0 Steps
  }
  delay(20);

  stepper2.moveTo(0);
  while (stepper2.currentPosition() != 0) {
    stepper2.run();
  }

  //1st Stepper Motor Homing Prosedure
  while (digitalRead(limitSwitch1) != 1) { //Run While Limitswitch 1 is not
pressed
    stepper1.setSpeed(-1200);
    stepper1.runSpeed();
    stepper1.setCurrentPosition(-3955); // When the Limit Switch is Pressed Set
1st Stepper Motors Position to 0 Steps
  }
  delay(20);
  stepper1.moveTo(0);
  while (stepper1.currentPosition() != 0) {
    stepper1.run();
  }
}

```

Κώδικας Προγράμματος Processing

Σε αυτή την παράγραφο του κεφαλαίου της ανάλυσης του λογισμικού ελέγχου του ρομποτικού βραχίονα ακολουθεί η παρουσίαση του κώδικα λειτουργίας στο λογισμικό της γλώσσας Processing. Συγκεκριμένα στην προκειμένη εφαρμογή υλοποιήθηκε ένα πρόγραμμα το οποίο δημιουργεί ένα γραφικό περιβάλλον με δυναμικά πεδία, τα οποία ελέγχουν τις κινήσεις των αξόνων του ρομποτικού βραχίονα. Αυτό επιτυγχάνεται αποστέλλοντας τα δεδομένα μέσω σειριακής επικοινωνίας στον μικροελεγκτή Arduino.

```
/*
   Arduino based SCARA Robot GUI
*/
import processing.serial.*;
import controlP5.*;
import static processing.core.PApplet.*;

Serial myPort; //Define Serial Port
ControlP5 cp5; //ControlP5 Object

//Define Initial Slider Values

int j1Slider = 0;
int j2Slider = 0;
int j3Slider = 0;
int zSlider = 100;
int j1JogValue = 0;
int j2JogValue = 0;
int j3JogValue = 0;
int zJogValue = 0;
int speedSlider = 500;
int accelerationSlider = 500;
int gripperValue = 180;
int gripperAdd=180;
int positionsCounter = 0;

int saveStatus = 0;
int runStatus = 0;

int slider1Previous = 0;
int slider2Previous = 0;
int slider3Previous = 0;
int sliderzPrevious = 100;
int speedSliderPrevious = 500;
int accelerationSliderPrevious = 500;
int gripperValuePrevious = 100;

//Define Inverse Kinematics Variable
boolean activeIK = false;

//Define Dimensions & Joint Angles
int xP=365;
int yP=0;
int zP=100;
float L1 = 228; // L1 = 228mm
float L2 = 136.5; // L2 = 136.5mm
float theta1, theta2, phi, z;

//Define String for Data Transfer
String[] positions = new String[100];
```



```

String data;

void setup() {

    size(960, 800); //Define GUI Size
    myPort = new Serial(this, "COM3", 115200); //Define Serial COMMunication port

    cp5 = new ControlP5(this);

//Define GUI Font & Control Font
PFont pfont = createFont("Arial", 25, true);
ControlFont font = new ControlFont(pfont, 22);
ControlFont font2 = new ControlFont(pfont, 25);

//Joint 1 Controls
cp5.addSlider("j1Slider") //Create 1st Joint Slider
    .setPosition(110, 190) //Define 1st Joint Slider Position in
GUI
    .setSize(270, 30) //Define 1st Joint Slider Size
    .setRange(-90, 266) //Define 1st Joint Slider Value Range
    .setColorLabel(#3269c2) //Define 1st Joint Slider Color
    .setFont(font) //Define 1st Joint Slider Font
    .setCaptionLabel("") //Define 1st Joint Slider Label
    ;
cp5.addButton("j1JogMinus") //Create 1st Joint - Jog Movement
    .setPosition(110, 238) //Define 1st Joint's - Jog Button
Position in GUI
    .setSize(90, 40) //Define 1st Joint's - Jog Button Size
    .setFont(font) //Define 1st Joint's - Jog Button Font
    .setCaptionLabel("JOG-") //Define 1st Joint's - Jog Button Label
    ;
cp5.addButton("j1JogPlus") //Create 1st Joint + Jog Movement
    .setPosition(290, 238) //Define 1st Joint's + Jog Button
Position in GUI
    .setSize(90, 40) //Define 1st Joint's + Jog Button Size
    .setFont(font) //Define 1st Joint's + Jog Button Font
    .setCaptionLabel("JOG+") //Define 1st Joint's + Jog Button Label
    ;
cp5.addNumberbox("j1JogValue") //Create 1st Joint Jog Value Controller
    .setPosition(220, 243) //Define 1st Joint Jog Value Controller
- Position in Gui
    .setSize(50, 30) //Define 1st Joint Jog Value Controller
- Size
    .setRange(0, 20) //Define 1st Joint Jog Value Controller
- Value Range
    .setFont(font) //Define 1st Joint Jog Value Controller
- Font
    .setMultiplier(0.1) //Define 1st Joint Jog Value Controller
- Multiplier
    .setValue(1) //Define 1st Joint Jog Value Controller
- Initial Value
    .setDirection(Controller.HORIZONTAL) //Define 1st Joint Jog Value Controller
- Controller Direction
    .setCaptionLabel("") //Define 1st Joint Jog Value Input Label
    ;

//Joint 2 Controls
cp5.addSlider("j2Slider") //Create 2nd Joint Slider
    .setPosition(110, 315) //Define 2nd Joint Slider Position in
GUI
    .setSize(270, 30) //Define 2nd Joint Slider Size
    .setRange(-150, 150) //Define 2nd Joint Slider Value Range
    .setColorLabel(#3269c2) //Define 2nd Joint Slider Color

```

```

        .setFont(font) //Define 2nd Joint Slider Font
        .setCaptionLabel("") //Define 2nd Joint Slider Label
    ;
    cp5.addButton("j2JogMinus") //Create 2nd Joint - Jog Movement
        .setPosition(110, 363) //Define 2nd Joint's - Jog Button
Position in GUI
        .setSize(90, 40) //Define 2nd Joint's - Jog Button Size
        .setFont(font) //Define 2nd Joint's - Jog Button Font
        .setCaptionLabel("JOG-") //Define 2nd Joint's - Jog Button Label
    ;
    cp5.addButton("j2JogPlus") //Create 2nd Joint + Jog Movement
        .setPosition(290, 363) //Define 2nd Joint's + Jog Button
Position in GUI
        .setSize(90, 40) //Define 2nd Joint's + Jog Button Size
        .setFont(font) //Define 2nd Joint's + Jog Button Font
        .setCaptionLabel("JOG+") //Define 2nd Joint's + Jog Button Label
    ;
    cp5.addNumberbox("j2JogValue") //Create 2nd Joint Jog Value Controller
        .setPosition(220, 368) //Define 2nd Joint Jog Value Controller
- Position in GUI
        .setSize(50, 30) //Define 2nd Joint Jog Value Controller
- Size
        .setRange(0, 20) //Define 2nd Joint Jog Value Controller
- Value Range
        .setFont(font) //Define 2nd Joint Jog Value Controller
- Font
        .setMultiplier(0.1) //Define 2nd Joint Jog Value Controller
- Multiplier
        .setValue(1) //Define 2nd Joint Jog Value Controller
- Initial Value
        .setDirection(Controller.HORIZONTAL) //Define 2nd Joint Jog Value Controller
- Controller Direction
        .setCaptionLabel("") //Define 2nd Joint Jog Value Input Label
    ;

//Joint 3 Controls
cp5.addSlider("j3Slider") //Create 3rd Joint Slider
        .setPosition(110, 440) //Define 3rd Joint Slider Position in
GUI
        .setSize(270, 30) //Define 3rd Joint Slider Size
        .setRange(-162, 162) //Define 3rd Joint Slider Value Range
        .setColorLabel(#3269c2) //Define 3rd Joint Slider Color
        .setFont(font) //Define 3rd Joint Slider Font
        .setCaptionLabel("") //Define 3rd Joint Slider Label
    ;
    cp5.addButton("j3JogMinus") //Create 3rd Joint - Jog Movement
        .setPosition(110, 493) //Define 3rd Joint's - Jog Button
Position in GUI
        .setSize(90, 40) //Define 3rd Joint's - Jog Button Size
        .setFont(font) //Define 3rd Joint's - Jog Button Font
        .setCaptionLabel("JOG-") //Define 3rd Joint's - Jog Button Label
    ;
    cp5.addButton("j3JogPlus") //Create 3rd Joint + Jog Movement
        .setPosition(290, 493) //Define 3rd Joint's + Jog Button
Position in GUI
        .setSize(90, 40) //Define 3rd Joint's + Jog Button Size
        .setFont(font) //Define 3rd Joint's + Jog Button Font
        .setCaptionLabel("JOG+") //Define 3rd Joint's + Jog Button Label
    ;
    cp5.addNumberbox("j3JogValue") //Create 3rd Joint Jog Value Controller
        .setPosition(220, 493) //Define 3rd Joint Jog Value Controller
- Position in GUI

```

```

        .setSize(50, 30) //Define 3rd Joint Jog Value Controller
- Size
        .setRange(0, 20) //Define 3rd Joint Jog Value Controller
- Value Range
        .setFont(font) //Define 3rd Joint Jog Value Controller
- Font
        .setMultiplier(0.1) //Define 3rd Joint Jog Value Controller
- Multiplier
        .setValue(1) //Define 3rd Joint Jog Value Controller
- Initial Value
        .setDirection(Controller.HORIZONTAL) //Define 3rd Joint Jog Value Controller
- Controller Direction
        .setCaptionLabel("") //Define 3rd Joint Jog Value Input Label
        ;

//Z controls
cp5.addSlider("zSlider") //Create Z Axis Joint Slider
        .setPosition(110, 565) //Define Z Axis Slider Position in GUI
        .setSize(270, 30) //Define Z Axis Slider Size
        .setRange(0, 150) //Define Z Axis Slider Value Range
        .setColorLabel(#3269c2) //Define Z Axis Slider Color
        .setFont(font) //Define Z Axis Slider Font
        .setCaptionLabel("") //Define Z Axis Slider Label
        ;
cp5.addButton("zJogMinus") //Create Z Axis - Jog Movement
        .setPosition(110, 618) //Define Z Axis - Jog Button Position
in GUI
        .setSize(90, 40) //Define Z Axis - Jog Button Size
        .setFont(font) //Define Z Axis - Jog Button Font
        .setCaptionLabel("JOG-") //Define Z Axis - Jog Button Label
        ;
cp5.addButton("zJogPlus") //Create Z Axis + Jog Movement
        .setPosition(290, 618) //Define Z Axis + Jog Button Position
in GUI
        .setSize(90, 40) //Define Z Axis + Jog Button Size
        .setFont(font) //Define Z Axis + Jog Button Font
        .setCaptionLabel("JOG+") //Define Z Axis + Jog Button Label
        ;
cp5.addNumberbox("zJogValue") //Create Z Axis Jog Value Controller
        .setPosition(220, 618) //Define Z Axis Jog Value Controller -
Position in GUI
        .setSize(50, 30) //Define Z Axis Jog Value Controller -
Size
        .setRange(0, 20) //Define Z Axis Jog Value Controller -
Value Range
        .setFont(font) //Define Z Axis Jog Value Controller -
Font
        .setMultiplier(0.1) //Define Z Axis Jog Value Controller -
Multiplier
        .setValue(1) //Define Z Axis Jog Value Controller -
Initial Value
        .setDirection(Controller.HORIZONTAL) //Define Z Axis Jog Value Controller -
Controller Direction
        .setCaptionLabel("") //Define Z Axis Jog Value Input Label
        ;

cp5.addTextfield("xTextfield") //Create X Axis Text
        .setPosition(530, 205) //Define X Axis Text - Position in GUI
        .setSize(70, 40) //Define X Axis Text - Size
        .setFont(font) //Define X Axis Text - Font
        .setColor(255) //Define X Axis Text - Color
        .setCaptionLabel("") //Define X Axis Text - Label

```

```

;
cp5.addTextfield("yTextfield") //Create Y Axis Text
    .setPosition(680, 205) //Define Y Axis Text - Position in GUI
    .setSize(70, 40) //Define Y Axis Text - Size
    .setFont(font) //Define Y Axis Text - Font
    .setColor(255) //Define Y Axis Text - Color
    .setCaptionLabel("") //Define Y Axis Text - Label
;
cp5.addTextfield("zTextfield") //Create Z Axis Text
    .setPosition(830, 205) //Define Z Axis Text - Position in GUI
    .setSize(70, 40) //Define Z Axis Text - Size
    .setFont(font) //Define Z Axis Text - Font
    .setColor(255) //Define Z Axis Text - Color
    .setCaptionLabel("") //Define Z Axis Text - Label
;

cp5.addButton("move") //Create Move Button
    .setPosition(590, 315) //Define Move Button - Position in GUI
    .setSize(240, 45) //Define Move Button - Size
    .setFont(font) //Define Move Button - Font
    .setCaptionLabel("MOVE TO POSITION") //Define Move Button - Label
;

cp5.addButton("savePosition") //Create Save Position Button
    .setPosition(470, 520) //Define Save Position - Position in
GUI //Define Save Position - Size
    .setSize(215, 50) //Define Save Position - Font
    .setFont(font2) //Define Move Save Position - Label
    .setCaptionLabel("SAVE POSITION") //Define Move Save Position - Label
;

cp5.addButton("run") //Create Save Position Button
    .setPosition(725, 520) //Define Save Position - Position in
GUI //Define Save Position - Size
    .setSize(215, 50) //Define Save Position - Font
    .setFont(font2) //Define Save Position - Label
    .setCaptionLabel("RUN PROGRAM") //Define Save Position - Label
;

cp5.addButton("updateSA") //Create Update Button
    .setPosition(760, 590) //Define Update - Position in GUI
    .setSize(150, 40) //Define Update - Size
    .setFont(font) //Define Update - Font
    .setCaptionLabel("(Update)") //Define Update - Label
;

cp5.addButton("clearSteps") //Create Clear Button
    .setPosition(490, 650) //Define Clear - Position in GUI
    .setSize(135, 40) //Define Clear - Size
    .setFont(font) //Define Clear - Font
    .setCaptionLabel("(CLEAR)") //Define Clear - Label
;

cp5.addSlider("speedSlider") //Create Speed Slider
    .setPosition(490, 740) //Define Speed Slider - Position in GUI
    .setSize(180, 30) //Define Speed Slider - Size
    .setRange(500, 4000) //Define Speed Slider - Range
    .setColorLabel("#3269c2") //Define Speed Slider - Color Label
    .setFont(font) //Define Speed Slider - Font
    .setCaptionLabel("") //Define Speed Slider - Label
;

cp5.addSlider("accelerationSlider") //Create Acceleration Slider

```

```

        .setPosition(720, 740) //Define Acceleration Slider - Position
in GUI
        .setSize(180, 30) //Define Acceleration Slider - Size
        .setRange(500, 4000) //Define Acceleration Slider - Range
        .setColorLabel(#3269c2) //Define Acceleration Slider - Color
Label
        .setFont(font) //Define Acceleration Slider - Font
        .setCaptionLabel("") //Define Acceleration Slider - Label
        ;
        cp5.addSlider("gripperValue") //Create Gripper Slider
        .setPosition(605, 445) //Define Gripper Slider - Position in
GUI
        .setSize(190, 30) //Define Gripper Slider - Size
        .setRange(0, 100) //Define Gripper Slider - Range
        .setColorLabel(#3269c2) //Define Gripper Slider - Color Label
        .setFont(font) //Define Gripper Slider - Font
        .setCaptionLabel("") //Define Gripper Slider - Label
        ;
}

void draw() {
    background(#F2F2F2); // Define Black Background
    textSize(26);
    fill(33);
    //Define Texts
    text("Forward Kinematics", 120, 135);
    text("Inverse Kinematics", 590, 135);
    textSize(40);
    text("SCARA Robot Control", 260, 60);
    textSize(45);
    text("J1", 35, 250);
    text("J2", 35, 375);
    text("J3", 35, 500);
    text("Z", 35, 625);
    textSize(22);
    text("Speed", 545, 730);
    text("Acceleration", 745, 730);

    //println("PREV: "+accelerationSlider);
    fill(speedSlider);
    fill(accelerationSlider);
    fill(j1Slider);
    fill(j2Slider);
    fill(j3Slider);
    fill(zSlider);
    fill(j1JogValue);
    fill(j2JogValue);
    fill(j3JogValue);
    fill(zJogValue);
    fill(gripperValue);

    updateData();
    //println(data);

    saveStatus=0; //Keep savePosition variable false. Check, when button SAVE is
    pressed and makes the value true, which indicates to store the value in the
    arduino code

    // If slider moved, calculate new position of X,Y and Z with forward kinematics
    if (slider1Previous != j1Slider) {
        if (activeIK == false) { // Check whether the inverseKinematics mode is
            active, Execute Forward kinematics only if inverseKinematics mode is false

```

```

        theta1 = round(cp5.getController("j1Slider").getValue()); //Read the value
from the slider1
        theta2 = round(cp5.getController("j2Slider").getValue());
        forwardKinematics();
        myPort.write(data);
    }
}
slider1Previous = j1Slider;

if (slider2Previous != j2Slider) {
    if (activeIK == false) { // Check whether the inverseKinematics mode is
active, Execute Forward kinematics only if inverseKinematics mode is false
        theta1 = round(cp5.getController("j1Slider").getValue()); //Read the value
from the slider1
        theta2 = round(cp5.getController("j2Slider").getValue());
        forwardKinematics();
        myPort.write(data);
    }
}
slider2Previous = j2Slider;

if (slider3Previous != j3Slider) {
    if (activeIK == false) { // Check whether the inverseKinematics mode is
active, Execute Forward kinematics only if inverseKinematics mode is false
        theta1 = round(cp5.getController("j1Slider").getValue()); //Read the value
from the slider1
        theta2 = round(cp5.getController("j2Slider").getValue());
        forwardKinematics();
        myPort.write(data);
    }
}
slider3Previous = j3Slider;

if (sliderzPrevious != zSlider) {
    if (activeIK == false) { // Check whether the inverseKinematics mode is
active, Execute Forward kinematics only if inverseKinematics mode is false
        zP = round(cp5.getController("zSlider").getValue());
        myPort.write(data);
    }
}
sliderzPrevious = zSlider;

if (gripperValuePrevious != gripperValue) {
    if (activeIK == false) { // Check whether the inverseKinematics mode is
active, Execute Forward kinematics only if inverseKinematics mode is false
        gripperAdd = round(cp5.getController("gripperValue").getValue());
        gripperValue=gripperAdd+50;
        updateData();
        println(data);
        myPort.write(data);
    }
}
gripperValuePrevious = gripperValue;
activeIK = false; //Deactivate inverseKinematics so the above if statements can
be executed the next iteration

fill(33);
textSize(32);
text("X: ", 500, 290);
text(xP, 533, 290);
text("Y: ", 650, 290);
text(yP, 685, 290);
text("Z: ", 800, 290);

```

```

text(zP, 835, 290);
textSize(26);
text("Gripper", 650, 420);
text("CLOSE", 510, 470);
text("OPEN", 810, 470);
textSize(18);

if (positionsCounter >0 ) {
    text(positions[positionsCounter-1], 460, 630);
    text("Last saved position: No."+(positionsCounter-1), 460, 600);
} else {
    text("Last saved position:", 460, 600);
    text("None", 460, 630);
}
}

// FORWARD KINEMATICS
void forwardKinematics() {
    float theta1F = theta1 * PI / 180;    //Convert Degrees to Radians
    float theta2F = theta2 * PI / 180;
    xP = round(L1 * cos(theta1F) + L2 * cos(theta1F + theta2F));
    yP = round(L1 * sin(theta1F) + L2 * sin(theta1F + theta2F));
}

// INVERSE KINEMATICS
void inverseKinematics(float x, float y) {
    theta2 = acos((sq(x) + sq(y) - sq(L1) - sq(L2)) / (2 * L1 * L2));
    if (x < 0 & y < 0) {
        theta2 = (-1) * theta2;
    }

    theta1 = atan(x / y) - atan((L2 * sin(theta2)) / (L1 + L2 * cos(theta2)));

    theta2 = (-1) * theta2 * 180 / PI;
    theta1 = theta1 * 180 / PI;

//Angles adjustment depending in which quadrant the final tool coordinate x,y is
if (x >= 0 & y >= 0) {    //1st quadrant
    theta1 = 90 - theta1;
}
if (x < 0 & y > 0) {    //2nd Quadrant
    theta1 = 90 - theta1;
}
if (x < 0 & y < 0) {    //3rd Quadrant
    theta1 = 270 - theta1;
    phi = 270 - theta1 - theta2;
    phi = (-1) * phi;
}
if (x > 0 & y < 0) {    //4th Quadrant
    theta1 = -90 - theta1;
}
if (x < 0 & y == 0) {
    theta1 = 270 + theta1;
}

//Calculate "phi" angle so gripper is parallel to the X axis
phi = 90 + theta1 + theta2;
phi = (-1) * phi;

//Angle adjustment depending in which quadrant the final tool coordinate x,y is
if (x < 0 & y < 0) {    //3rd quadrant
    phi = 270 - theta1 - theta2;
}
}

```



```

    if (abs(phi) > 165) {
        phi = 180 + phi;
    }

    theta1=round(theta1);
    theta2=round(theta2);
    phi=round(phi);

    cp5.getController("j1Slider").setValue(theta1);
    cp5.getController("j2Slider").setValue(theta2);
    cp5.getController("j3Slider").setValue(phi);
    cp5.getController("zSlider").setValue(zP);
}

void controlEvent(ControlEvent theEvent) {

    if (theEvent.isController()) {
        println(theEvent.getController().getName());
    }
}

public void xTextfield(String theText) {
    //If a value is entered into the Textfield, read the value, convert to integer,
    set the inverseKinematics mode active
    xP=Integer.parseInt(theText);
    activeIK = true;
    inverseKinematics(xP, yP); //Use inverse kinematics to calculate the J1(theta1),
    J2(theta2), and J3(phi) positions
    //activeIK = false;
    println("Test; theta1: "+theta1+" theta2: "+theta2);
}
public void yTextfield(String theText) {
    yP=Integer.parseInt(theText);
    activeIK = true;
    inverseKinematics(xP, yP);
    //activeIK = false;
}
public void zTextfield(String theText) {
    zP=Integer.parseInt(theText);
    activeIK = true;
    inverseKinematics(xP, yP);
}

public void j1JogMinus() {
    int a = round(cp5.getController("j1Slider").getValue());
    a=a-j1JogValue;
    cp5.getController("j1Slider").setValue(a);
}
//Joint 1 control
public void j1JogPlus() {
    int a = round(cp5.getController("j1Slider").getValue());
    a=a+j1JogValue;
    cp5.getController("j1Slider").setValue(a);
}
//Joint 2 control
public void j2JogMinus() {
    int a = round(cp5.getController("j2Slider").getValue());
    a=a-j2JogValue;
    cp5.getController("j2Slider").setValue(a);
}
public void j2JogPlus() {
    int a = round(cp5.getController("j2Slider").getValue());
    a=a+j2JogValue;
}

```

```

    cp5.getController("j2Slider").setValue(a);
}
//Joint 3 control
public void j3JogMinus() {
    int a = round(cp5.getController("j3Slider").getValue());
    a=a-j3JogValue;
    cp5.getController("j3Slider").setValue(a);
}
public void j3JogPlus() {
    int a = round(cp5.getController("j3Slider").getValue());
    a=a+j3JogValue;
    cp5.getController("j3Slider").setValue(a);
}
//Joint 3 control
public void zJogMinus() {
    int a = round(cp5.getController("zSlider").getValue());
    a=a-zJogValue;
    cp5.getController("zSlider").setValue(a);
}
public void zJogPlus() {
    int a = round(cp5.getController("zSlider").getValue());
    a=a+zJogValue;
    ;
    cp5.getController("zSlider").setValue(a);
}

public void move() {

    myPort.write(data);
    println(data);
}

public void savePosition() {
    //Save Joint1, Joint2, Joint3 and Z position in the array

positions[positionsCounter]="J1="+str(round(cp5.getController("j1Slider").getValue()))
    +"; J2=" + str(round(cp5.getController("j2Slider").getValue()))
    +"; J3="+str(round(cp5.getController("j3Slider").getValue()))
    +"; Z="+str(round(cp5.getController("zSlider").getValue()));
    positionsCounter++;
    saveStatus = 1;
    updateData();
    myPort.write(data);
    saveStatus=0;
}

public void run() {

    if (runStatus == 0) {
        cp5.getController("run").setCaptionLabel("STOP");
        cp5.getController("run").setColorLabel(#e74c3c);

        runStatus = 1;
    } else if (runStatus == 1) {
        runStatus = 0;
        cp5.getController("run").setCaptionLabel("RUN PROGRAM");
        cp5.getController("run").setColorLabel(255);
    }
    updateData();
    myPort.write(data);
}
public void updateSA() {

```

```

    myPort.write(data);
}
public void clearSteps() {
    saveStatus = 2; //Xlear all steps / program
    updateData();
    myPort.write(data);
    println("Clear: "+data);
    positionsCounter=0;
    saveStatus = 0;
}

public void updateData() {
    data = str(saveStatus)
        +","+str(runStatus)
        +","+str(round(cp5.getController("j1Slider").getValue()))
        +","+str(round(cp5.getController("j2Slider").getValue()))
        +","+str(round(cp5.getController("j3Slider").getValue()))
        +","+str(round(cp5.getController("zSlider").getValue()))
        +","+str(gripperValue)
        +","+str(speedSlider)
        +","+str(accelerationSlider);
}

```

10. ΠΡΟΣΟΜΟΙΩΣΗ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ

10.1 Προσομοίωση του ρομπότ μέσω του λογισμικού MATLAB.

Στο κεφάλαιο αυτό περιγράφεται η εκκίνηση της διαδικασίας της προσομοίωσης του ρομποτικού βραχίονα σε εικονικό περιβάλλον χρησιμοποιώντας το λογισμικό του MATLAB καθώς και ο έλεγχος τους μέσω ορθής και αντίστροφης κινηματικής. Για την πραγματοποίηση της διαδικασίας αυτής αρχικά χρειάστηκε η εκκίνηση μιας σειριακής επικοινωνίας μεταξύ του μικροελεγκτή Arduino και του MATLAB με βάση την οποία τα δεδομένα που θα εξάγονται από το δεύτερο θα μεταφέρονται στο πρώτο σε αναγνωρίσιμη μορφή. Η εκκίνηση της επικοινωνίας γίνεται αρχικά εγκαθιστώντας μια βιβλιοθήκη εντολών στο MATLAB (“MATLAB Support Package for Arduino Hardware”) και στην συνέχεια δίνοντας τις κατάλληλες εντολές ώστε να οριστούν δεδομένα όπως η σειριακή θύρα επικοινωνίας και ο ρυθμός ανταλλαγής δεδομένων (Baud Rate). Όταν υλοποιηθεί επιτυχώς η σύνδεση, τότε με τις κατάλληλες εντολές τα δεδομένα μπορούν να αποσταλούν.

Στην συνέχεια, για το έλεγχο του ρομποτικού βραχίονα χρειάστηκε όπως και στην περίπτωση του κώδικα ελέγχου του ρομποτικού βραχίονα να χρησιμοποιηθούν οι εξισώσεις κινηματικής του ρομπότ ώστε να υλοποιείτε ο υπολογισμός της θέσης του στον χώρο.

Οπότε, έγινε η μετατροπή αυτών των εντολών σε μορφή αναγνωρίσιμη από το λογισμικό του MATLAB και προστέθηκαν οι εντολές ανάγνωσης δεδομένων. Με αυτές ο χειριστής μπορεί να εισάγει συντεταγμένες στο Command Window του MATLAB και αφού τις εισάγει πατώντας το κουμπί Enter, και να επιστραφούν οι υπολογισμένες γωνίες. Έπειτα μελετήθηκαν οι τρόποι διασύνδεσης του Arduino με το MATLAB.

Κατά την διάρκεια των ερευνών που εκτελέστηκαν, βρέθηκαν διάφοροι τρόποι δια σύνδεσης του MATLAB με το Arduino, εκ των οποίων ο πιο διαδεδομένος και εύκολος να υλοποιηθεί, λαμβάνοντας το διαθέσιμο hardware, ήταν αυτός της σειριακής επικοινωνίας. Αφού συνδέθηκε το Arduino στον υπολογιστή μέσω καλωδίου USB, στην συνέχεια ακολουθήθηκε η βηματική διαδικασία από την ιστοσελίδα του λογισμικού MATLAB, Mathworks, για την αναγνώριση και εγκατάσταση των drivers της συσκευής στο πρόγραμμα. (Set up and Configure Arduino Hardware, 2022)

Μετά την ολοκλήρωση της επιτυχημένης αναγνώρισης της συσκευής από το λογισμικό υλικό, χρειάστηκε να πραγματοποιηθεί η σύνδεση του η οποία έγινε ακολουθώντας, όπως και πριν, τις προκαθορισμένες οδηγίες του λογισμικού από την αντίστοιχη ιστοσελίδα. (Connect to Arduino Hardware, 2022)

Τέλος, συνδυάστηκε ο κώδικας υπολογισμού των κινήσεων των βηματικών μοτέρ με αυτόν της σύνδεσης του Arduino με το MATLAB, υλοποιήθηκε ο έλεγχος του ρομποτικού βραχίονα μέσω του λογισμικού MATLAB. Δυστυχώς, λόγω έλλειψης χρόνου δεν ολοκληρώθηκε η προσπάθεια υλοποίησης της τρισδιάστατης εικονικής αναπαράστασης του ρομποτικού βραχίονα. Η διαδικασία αυτή έχει τεθεί σαν μελλοντικός στόχος για περαιτέρω ανάπτυξη του θέματος.

10.2 Προσομοίωση του ρομπότ μέσω της γλώσσας προγραμματισμού Processing.

Κατά την διαδικασία της προσομοίωσης του ρομποτικού βραχίονα σε εικονικό χώρο κάνοντας χρήση την γλώσσα προγραμματισμού της processing αρχικά υλοποιήθηκε η αναπαράσταση του ρομποτικού βραχίονα σε ένα εικονικό περιβάλλον και ο έλεγχος του άκρου εργασίας με την βοήθεια του “ποντικιού” του ηλεκτρονικού υπολογιστή. Αρχικά, μετά από εκτεταμένη έρευνα στο διαδίκτυο βρέθηκε ένα άρθρο με οδηγίες για την υλοποίηση εικονικού ελέγχου ενός Articulated Robotic Arm. (Create a Robotics Simulator Using Processing, 2020)

Με βάση αυτές της οδηγίες και σε συνδυασμό με το υπάρχον σχέδιο του ρομποτικού βραχίονα τύπου scara υλοποιήθηκε η ίδια εκτέλεση κατά την οποία επιτεύχθηκε ο έλεγχος του άκρου εργασίας του ρομπότ με την χρήση του “ποντικιού” του υπολογιστή. Τέθηκαν ως επιμέρους στόχοι ο έλεγχος του ρομποτικού βραχίονα μέσω του εικονικού περιβάλλοντος που δημιουργήθηκε στην διπλωματική εργασία και η παράλληλη εικονική του αναπαράσταση, η οποία λόγω έλλειψης χρόνου δεν υλοποιήθηκε.

11. ΣΥΜΠΕΡΑΣΜΑΤΑ

Η ρομποτική είναι ένας κλάδος της μηχανικής, που έχει ως αντικείμενο τη μελέτη των μηχανών. Δραστηριοποιείται στην σχεδίαση, ανάπτυξη, την κατασκευή αλλά και των προγραμματισμό των μηχανών οι οποίες επιτελούν συγκεκριμένες λειτουργίες. Αποτελεί επίσης έναν κλάδο της επιστήμης του οποίου η ανάπτυξη απασχολεί την επιστήμη την σύγχρονη εποχή και ενδεχόμενος να συνεχίσει να την απασχολεί στο μέλλον λαμβάνοντας υπόψιν την εξέλιξη του κλάδου του αυτοματισμού.

Στην συγκεκριμένη διπλωματική εργασία επιλέχθηκε η υλοποίηση μιας τέτοιας έρευνας, ανάπτυξης και δημιουργίας ενός ρομποτικού βραχίονα με σκοπό την εκμάθηση και την μελέτη και την κατανόηση μιας τέτοιας διαδικασίας. Πιο συγκεκριμένα υλοποιήθηκε ένας ρομποτικός βραχίονας τύπου SCARA, 4 βαθμών ελευθερίας, Για την δημιουργία του ρομποτικού βραχίονα ακολουθήθηκαν και εκτελέστηκαν οι διαδικασίες, αρχικά της έρευνας αγοράς για τους διαθέσιμους πόρους και τα πιθανά κόστη. Στην συνέχεια βάση αυτής της έρευνας πραγματοποιήθηκε η σχεδίαση του μηχανολογικού αλλά και ηλεκτρολογικού σχεδίου του ρομποτικού βραχίονα σε σχεδιαστικό λογισμικό. Έπειτα, ακολούθησε η τρισδιάστατη εκτύπωση καθώς και η αγορά των απαραίτητων υλικών για την κατασκευή. Τέλος, αφού ολοκληρώθηκε η διαδικασία της συναρμολόγησης του, έγινε ο προγραμματισμός της λειτουργίας του.

Τα συμπεράσματα που προκύπτουν από την ολοκλήρωση των παραπάνω βημάτων είναι ότι η δημιουργία ενός ρομποτικού βραχίονα, σε εκπαιδευτικό επίπεδο, είναι μια απαιτητική διαδικασία η οποία παράλληλα είναι επιμορφωτική και ενδιαφέρουσα, από την οποία ο φοιτητής μπορεί να αποκομίσει πληθώρα γνώσεων. Στα θετικά επίσης ανήκει, η εμπειρία η οποία προσκομίζεται από το κάθε βήμα, όπως αυτό της σχεδίασης ή της συναρμολόγησης, η οποία αποτελεί χρήσιμο εφόδιο σε εργασιακούς χώρους που δραστηριοποιούνται σε αυτόν τον τομέα. Το βασικό μειονέκτημα όμως της ανάπτυξης της παρούσης διπλωματικής εργασίας είναι ο οικονομικός παράγοντας της κατασκευής.

Επιμέρους στόχοι της παρούσης διπλωματικής εργασίας θα μπορούσαν να τεθούν ο έλεγχος του ρομποτικού βραχίονα μέσω άλλου λογισμικού όπως είναι το MATLAB, δημιουργώντας ένα περιβάλλον ελέγχου μέσω του Simulink με οποίο θα υλοποιείται αρχικά ο έλεγχος του και στην συνέχεια η απεικόνιση του, αλλά και των κινήσεων του σε εικονικό περιβάλλον. Επιπροσθέτως ως στόχος εξέλιξης και βελτίωσης θα μπορούσε να τεθεί η υλοποίηση της εικονική αναπαράστασης του μέσω του κώδικα processing σε συνδυασμό με τον υπάρχον περιβάλλον ελέγχου του ρομποτικού βραχίονα.

Τέλος, πιθανές βελτιώσεις του ρομποτικού βραχίονα, θα μπορούσε να είναι το fine tuning των ταχυτήτων και επιταχύνσεων των βηματικών κινητήρων ώστε να μπορεί να λειτουργεί στην μέγιστη δυνατή ταχύτητα χωρίς όμως αυτό να επηρεάζει την ακρίβεια των κινήσεων του. Επίσης, θα μπορούσε να εφαρμοστεί κάποιο προστατευτικό ή κάλυψη για τα καλώδια που βρίσκονται εξωτερικά του βραχίονα. Σε περίπτωση ανακατασκευής του, για να βελτιστοποιηθεί περισσότερο η κατασκευή και να επιτευχθεί μεγαλύτερη ταχύτητα θα μπορούσε να σχεδιαστεί διαφορετικά ώστε να εγκατασταθούν μοτέρ μεγαλύτερου μεγέθους, ώστε να φέρουν την ανάλογη ροπή, άμεσα στις αρθρώσεις χωρίς να εφαρμόζεται μηχανισμός μείωσης.

13. ΒΙΒΛΙΟΓΡΑΦΙΑ

- Arduino Uno*. (2022, 11 2). Ανάκτηση από Wikipedia: https://en.wikipedia.org/wiki/Arduino_Uno
- Arduino.cc. (χ.χ.). *Arduino Reference*. Ανάκτηση 12 3, 2018, από <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/>
- Ball bearing*. (2022, 09 16). Ανάκτηση από Wikipedia: https://en.wikipedia.org/wiki/Ball_bearing
- Bearing (mechanical)*. (2022, 10 14). Ανάκτηση από Wikipedia: [https://en.wikipedia.org/wiki/Bearing_\(mechanical\)](https://en.wikipedia.org/wiki/Bearing_(mechanical))
- CNC Shield*. (2019, 11 04). Ανάκτηση από MakerHardware: https://makerhardware.net/wiki/doku.php?id=electronics:cnc_shield
- CNC V3.0 PWM pins*. (2022, 6 10). Ανάκτηση από forum.Arduino.cc: <https://forum.arduino.cc/t/cnc-v3-0-pwm-pins/1001467>
- Cobot*. (2022, June 27). Ανάκτηση από Wikipedia: <https://en.wikipedia.org/wiki/Cobot>
- Connect to Arduino Hardware*. (2022, 08 08). Ανάκτηση από MathWorks: <https://www.mathworks.com/help/supportpkg/arduinoio/ug/connect-to-arduino-hardware.html>
- Create a Robotics Simulator Using Processing*. (2020, 10 30). Ανάκτηση από Instructables: <https://www.instructables.com/Create-a-Robotics-simulator-using-Processing/>
- Everything you need to know about the Arduino Hardware*. (2022, 3 21). Ανάκτηση από Circuit Digest: <https://circuitdigest.com/article/everything-you-need-to-know-about-arduino-uno-board-hardware>
- FlexiBowl. (2020, May 11). *Scara Robot*. Ανάκτηση από Flexibowl: <https://www.flexibowl.com/scara-robot.html>
- GT2 Timing Belt*. (2015, 02 20). Ανάκτηση από RepRap: https://reprap.org/wiki/GT2_Timing_Belt
- Humanoid Robot*. (2022, September 12). Ανάκτηση από Wikipedia: https://en.wikipedia.org/wiki/Humanoid_robot
- Humanoid Robots*. (2022, July 5). Ανάκτηση από Automate: <https://www.automate.org/a3-content/service-robots-humanoid-robots>
- Intel*. (2022, July 21). Ανάκτηση από Types of Robots: <https://www.intel.com/content/www/us/en/robotics/types-and-applications.html>

Interfacing A4988 Stepper Motor Driver with Arduino. (2021, 01 11). Ανάκτηση από Electropeak: <https://electropeak.com/learn/interfacing-a4988-stepper-motor-driver-with-arduino/>

Leadscrew. (2022, 06 03). Ανάκτηση από Wikipedia: <https://en.wikipedia.org/wiki/Leadscrew>

Learning. (2021, December 16). *Benefits of Virtual Robotics in Schools.* Ανάκτηση από Learning: <https://www.learning.com/blog/benefits-of-virtual-robotics-in-schools/>

Limit Switches Explained. (2020, 10 19). Ανάκτηση από RealPars: <https://realpars.com/limit-switch/>

Planchard, M. (2015, 6 22). *West Virginia University wins NASA Robot Competition.* (solidworks.com) Ανάκτηση 12 7, 2018, από https://www.google.gr/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKewjG5M7_yY3fAhULUIAKHW01B7MQjRx6BAgBEAU&url=https%3A%2F%2Fblogs.solidworks.com%2Fteacher%2F2015%2F06%2Fwest-virginia-university-wins-nasa-robot-competition.html&psig=AOvVaw1Z8LeN5wSR

Polylactic acid. (2022, 11 7). Ανάκτηση από Wikipedia: https://en.wikipedia.org/wiki/Polylactic_acid

Robots. (2020, December 1). *RobotWorx.* Ανάκτηση από Robots: <https://www.robots.com/faq/what-are-the-main-types-of-robots>

Rosroboticslearning. (2021, October 16). *Forward Kinematics.* Ανάκτηση από Rosroboticslearning: <https://www.rosroboticslearning.com/forward-kinematics>

Set up and Configure Arduino Hardware. (2022, 10 2). Ανάκτηση από MathWorks: <https://www.mathworks.com/help/supportpkg/arduinoio/ug/configure-setup-for-arduino-hardware.html>

Slicer in 3D Printing: Definition, Features, and How it Works. (2022, 09 12). Ανάκτηση από Xometry: <https://www.xometry.com/resources/3d-printing/what-is-a-slicer-in-3d-printing/>

SPOOKY TEDDY - ARDUINO POWERED SELF-ROCKING CHAIR & ROTATING HEAD. (2018, 11 7). Ανάκτηση από www.instructables.com: <https://www.instructables.com/id/Spooky-Teddy-Arduino-Powered-Self-rocking-Chair-Ro/>

Stepper Motor Driver. (2018, 05 01). Ανάκτηση από RepRap: https://reprap.org/wiki/Stepper_motor_driver

The 4 Types of FDM 3D Printer Explained. (2022, 11 1). Ανάκτηση από 3D Sourced: <https://www.3dsourced.com/3d-printers/types-of-fdm-3d-printer-cartesian-delta/>

Thrust Bearing. (2022, 08 15). Ανάκτηση από Wikipedia: https://en.wikipedia.org/wiki/Thrust_bearing

What is a Micro Switch : Working & Its Applications. (22, 05 23). Ανάκτηση από Elprocus:
<https://www.elprocus.com/micro-switch/>

Wikipedia. (2019, November 1). *MATLAB*. Ανάκτηση από Wikipedia:
<https://en.wikipedia.org/wiki/MATLAB>

Wikipedia. (2021, August 11). *Proscressing*. Ανάκτηση από Wikipedia:
[https://el.wikipedia.org/wiki/Processing_\(%CE%B3%CE%BB%CF%8E%CF%83%CF%83%CE%B1_%CF%80%CF%81%CE%BF%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%B1%CF%84%CE%B9%CF%83%CE%BC%CE%BF%CF%8D\)](https://el.wikipedia.org/wiki/Processing_(%CE%B3%CE%BB%CF%8E%CF%83%CF%83%CE%B1_%CF%80%CF%81%CE%BF%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%B1%CF%84%CE%B9%CF%83%CE%BC%CE%BF%CF%8D))

Wikipedia. (2021, 5 15). *SCARA*. Ανάκτηση 9 12, 2022, από <https://en.wikipedia.org/wiki/SCARA>

Wikipedia. (2021, October 12). *Ρομπότ*. Ανάκτηση 7 13, 2022, από
<https://el.wikipedia.org/wiki/%CE%A1%CE%BF%CE%BC%CF%80%CF%8C%CF%84>

Wikipedia. (2022, July 23). *Inverse Kinematics*. Ανάκτηση από Wikipedia:
https://en.wikipedia.org/wiki/Inverse_kinematics

Βικιπαίδεια. (2021, 10 12). *Ρομπότ*. Ανάκτηση 7 9, 2022, από
<https://el.wikipedia.org/wiki/%CE%A1%CE%BF%CE%BC%CF%80%CF%8C%CF%84>