

Διεθνές Πανεπιστήμιο της Ελλάδος

Τμήμα Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών

---

Διπλωματική Εργασία

Πρόγραμμα Μεταπτυχιακών Σπουδών στη Ρομποτική

---

## Σχεδίαση και κατασκευή βαδίζοντος ρομπότ με την βοήθεια 3D Εκτυπωτή και Arduino

Ραφαέλα Τερζή

*Διεθνές Πανεπιστήμιο της Ελλάδος*

Σέρρες, 30-1-2024

Εργασία που υποβλήθηκε στο

Πρόγραμμα Μεταπτυχιακών Σπουδών στη Ρομποτική,

του Διεθνούς Πανεπιστημίου της Ελλάδος,

για τη μερική εκπλήρωση υποχρεώσεων για το Δίπλωμα Ειδίκευσης στη

Ρομποτική

Επιβλέπων Καθηγητής: Σπυρίδων Καζαρλής

# Περίληψη

Στα πλαίσια της παρούσας διπλωματικής εργασίας, κατασκευάστηκε ένα βαδίζον ρομπότ με τέσσερα πόδια με την χρήση 3D Εκτυπωτή και Arduino. Σκοπός του ρομπότ είναι η αυτόνομη κίνηση προς τα εμπρός, πίσω αλλά και προς τα πλάγια και η αποφυγή εμποδίων.

Αρχικά, παρουσιάζεται το πρόβλημα, επισημαίνοντας τη σημασία της διερεύνησης προβλημάτων που σχετίζονται με την ανάπτυξη ενός βαδίζοντος ρομπότ. Ακόμα, παρέχεται μια πλήρης αναφορά της τρέχουσας κατάστασης της τεχνολογίας, εμβαθύνοντας στις μεθόδους σχεδιασμού, την τρισδιάστατη εκτύπωση, το Arduino Uno, τους αισθητήρες υπερήχων, τους σερβομηχανισμούς και τα υπάρχοντα ρομπότ με πόδια.

Στη συνέχεια, περιγράφεται η μεθοδολογία που ακολουθήθηκε για την υλοποίηση του ρομπότ και αναλύεται το κύριο μέρος της εργασίας. Περιγράφεται η επιλογή του σκελετού του ρομπότ και η 3D εκτύπωση των κομματιών του, η συναρμολόγηση του, η ενσωμάτωση ενεργών στοιχείων όπως σερβομηχανισμών, Arduino και αισθητήρες. Ακόμα, παρατίθεται αναλυτικά η κίνηση των ποδιών και ο τελικός προγραμματισμός του Arduino, έπειτα από μια σειρά δοκιμαστικών προγραμμάτων.

Τέλος, πραγματοποιείται μια αξιολόγηση των συμπερασμάτων που προκύπτουν από την υλοποίηση του βαδίζοντος ρομπότ και ολοκληρώνεται με μια σύνοψη των βασικών σημείων της εργασίας, ενώ προτείνονται και κατευθύνσεις για μελλοντική επέκταση και ανάπτυξη του ρομπότ.

## ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

Ρομποτική, Ρομπότ με πόδια, Βαδίζον ρομπότ, 3D Printer, Arduino Uno, αισθητήρας υπερήχων, σερβομηχανισμοί

# Abstract

This thesis revolves around the construction of a four-legged walking robot using a 3D printer and Arduino, designed to autonomously navigate forward, backward, sideways and avoid obstacles.

The initial presentation of the problem highlights the critical need for thorough investigation in the field of walking robot development. The paper then offers a complete review of the current state of the art, thoroughly examining design methodologies, 3D printing techniques, Arduino Uno integration, ultrasonic sensors, servo motors, and existing walking robots.

The methodology used to implement the robot is then clarified, and a detailed description of the whole process is provided. The central part of the paper unfolds with a detailed narrative of the critical steps, including the selection of the robot skeleton, 3D printing of its components, the assembling procedures and the integration of active components such as servos, Arduino and sensors. The mechanics of leg movement and the final programming of the Arduino are analyzed, following a comprehensive set of test programs.

Finally, an evaluation of the conclusions drawn from the practical application of the walking robot is thoroughly discussed. The paper concludes with a brief summary, that includes the key points and suggests ways for future expansion and improvement of the robot.

## KEY WORDS

Robotics, Robot with legs, Walking Robot, 3D Printer, Arduino Uno, Ultrasonic Sensor, Servos

# Περιεχόμενα

Περίληψη .....	2
Abstract .....	3
Κατάλογος Εικόνων .....	5
Κατάλογος Πινάκων .....	6
Κατάλογος Σχημάτων .....	7
Κεφάλαιο 1: Διατύπωση του προβλήματος .....	8
1.1 Εισαγωγή .....	8
1.2 Ορισμός του προβλήματος .....	11
1.3 Οφέλη της διερεύνησης του προβλήματος .....	11
Κεφάλαιο 2: Κατάσταση της Τέχνης .....	13
2.1 Σχεδίαση και 3D Εκτύπωση .....	13
2.2 Arduino Uno .....	19
2.3 Αισθητήρες και σερβομηχανισμοί.....	23
2.4 Ρομπότ με πόδια .....	27
Κεφάλαιο 3: Μεθοδολογία .....	30
3.1 Βήματα υλοποίησης του βαδίζοντος ρομπότ .....	30
Κεφάλαιο 4: Κύριο Μέρος.....	34
4.1 Εκτύπωση μέσω 3D Printer .....	34
4.2 Υλικά και συνδεσμολογία .....	43
4.3 Συναρμολόγηση του ρομπότ.....	45
4.4 Δοκιμαστικά προγράμματα και αποτελέσματα .....	56
4.5 Κίνηση των ποδιών.....	59
4.6 Τελικός προγραμματισμός του Arduino.....	69
Κεφάλαιο 5: Αξιολόγηση συμπερασμάτων .....	72
Κεφάλαιο 6: Σύνοψη και μελλοντική επέκταση.....	74
Βιβλιογραφία .....	76
Παραρτήματα.....	77
Κώδικας Arduino .....	77

## Κατάλογος Εικόνων

Εικόνα 1: Ρομπότ με 2 πόδια, πηγή: [1] .....	9
Εικόνα 2: Ρομπότ με 4 πόδια – Mini Cheetah by MIT, πηγή: [2] .....	10
Εικόνα 3: Ρομπότ με 6 πόδια, πηγή: [3] .....	10
Εικόνα 4: Ο 3D εκτυπωτής Wanhao Duplicator 6.....	15
Εικόνα 5: Προεπισκόπηση εκτύπωσης στο Simplify3D όπου φαίνεται το brim, το raft και το αντικείμενο σε layers με infill 15%.....	17
Εικόνα 6: Διάφορα μοτίβα infill στην προβολή εκτύπωσης του Simplify3D .....	18
Εικόνα 7: Το Arduino Uno R3, πηγή: [8].....	20
Εικόνα 8: Διάγραμμα ακροδεκτών της πλακέτας Arduino Uno, πηγή: [8].....	21
Εικόνα 9: Το Sensor Shield V5.0 συνδεδεμένο επάνω στο Arduino Uno.....	22
Εικόνα 10: Ο σερβομηχανισμός Servo Micro 2.2kg.cm Plastic Gears (Waveshare SG90) ....	25
Εικόνα 11: Ένα 3D printed robot βασισμένο στο Spot και το Mini Cheetah, πηγή: [14] .....	28
Εικόνα 12: Το ρομπότ Spot από την Boston Dynamics, πηγή: [11].....	28
Εικόνα 13: Μεσαία βάση - Base Electronics στο πρόγραμμα Cura.....	36
Εικόνα 14: Πίσω βάση - Base Back στο πρόγραμμα Cura.....	36
Εικόνα 15: Μπροστά βάση - Front Base στο πρόγραμμα Cura .....	37
Εικόνα 16: Κυκλικό pin L1 (Circular Pin L1) στο πρόγραμμα Cura .....	37
Εικόνα 17: Κυκλικό pin L2 (Circular Pin L2) στο πρόγραμμα Cura .....	38
Εικόνα 18: Κυκλικό pin L3 (Circular Pin L3) στο πρόγραμμα Cura .....	38
Εικόνα 19: Κυκλικό pin L4 (Circular Pin L4) στο πρόγραμμα Cura .....	39
Εικόνα 20: Μηρός σερβομηχανισμού (Thigh Servo) στο πρόγραμμα Cura .....	39
Εικόνα 21: Μηρός (Thigh) στο πρόγραμμα Cura .....	40
Εικόνα 22: Κνήμη εξωτερική (Calf Ext) στο πρόγραμμα Cura .....	40
Εικόνα 23: Κνήμη εσωτερική (Calf Int) στο πρόγραμμα Cura .....	41
Εικόνα 24: Πόδι (Foot) στο πρόγραμμα Cura .....	41
Εικόνα 25: Τετράγωνο κλιπ (Square Clip) στο πρόγραμμα Cura.....	42
Εικόνα 26: Κυκλικό κλιπ (Circular Clip) στο πρόγραμμα Cura.....	42
Εικόνα 27: Συναρμολόγηση του σώματος του ρομπότ - βήμα 1.....	46
Εικόνα 28: Συναρμολόγηση του σώματος του ρομπότ - βήμα 2 έως 4.....	46
Εικόνα 29: Συναρμολόγηση του σώματος του ρομπότ - βήμα 5.....	47
Εικόνα 30: Συναρμολόγηση του σώματος του ρομπότ - βήμα 6 έως 7.....	47
Εικόνα 31: Συναρμολόγηση του σώματος του ρομπότ - βήμα 8.....	48
Εικόνα 32: Συναρμολόγηση του σώματος του ρομπότ - βήμα 9.....	48
Εικόνα 33: Συναρμολόγηση του σώματος του ρομπότ - βήμα 10 έως 12.....	49
Εικόνα 34: Συναρμολόγηση του σώματος του ρομπότ - βήμα 13.....	49
Εικόνα 35: Συναρμολόγηση ποδιών του ρομπότ - βήμα 1 .....	51
Εικόνα 36: Συναρμολόγηση ποδιών του ρομπότ - βήμα 2.....	51
Εικόνα 37: Συναρμολόγηση ποδιών του ρομπότ - βήμα 3 έως 4.....	52
Εικόνα 38: Συναρμολόγηση ποδιών του ρομπότ - βήμα 5 έως 8 .....	52
Εικόνα 39: Συναρμολόγηση ποδιών του ρομπότ - βήμα 9 έως 11 .....	53
Εικόνα 40: Συναρμολόγηση ποδιών του ρομπότ - βήμα 12 έως 14.....	53
Εικόνα 41: Συναρμολόγηση ποδιών του ρομπότ - βήμα 15 έως 18.....	54
Εικόνα 42: Τα 4 πόδια του ρομπότ συναρμολογημένα - βήμα 19.....	54
Εικόνα 43: Το ολοκληρωμένο ρομπότ.....	55
Εικόνα 44: Η συνάρτηση loop του προγράμματος στο Arduino IDE.....	71

## Κατάλογος Πινάκων

Πίνακας 1: Κάποια βασικά χαρακτηριστικά του 3D Printer Wanhao Duplicator 6, πηγή: [6]	15
Πίνακας 2: Παράμετροι εκτύπωσης στο πρόγραμμα Wanhao Cura.....	35
Πίνακας 3: Αναλυτικό κόστος της κατασκευής του βαδίζοντος ρομπότ .....	55
Πίνακας 4: Αρχική θέση (home position) των σερβομηχανισμών .....	57
Πίνακας 5: Προτεινόμενη περιστροφή των servo σε μοίρες για κάθε θέση: Κίνηση προς τα εμπρός [15].....	57
Πίνακας 6: Προτεινόμενη περιστροφή των servo σε μοίρες για κάθε θέση: Στροφή προς τα αριστερά [15].....	57
Πίνακας 7: Περιστροφή των servo σε μοίρες για κάθε θέση: Κίνηση προς τα εμπρός.....	59
Πίνακας 8: Περιστροφή των servo σε μοίρες για κάθε θέση: Κίνηση προς τα πίσω .....	61
Πίνακας 9: Περιστροφή των servo σε μοίρες για κάθε θέση: Στροφή προς τα δεξιά .....	63
Πίνακας 10: Περιστροφή των servo σε μοίρες για κάθε θέση: Στροφή προς τα αριστερά .....	65
Πίνακας 11: Περιστροφή των servo σε μοίρες για κάθε κίνηση-πόζα .....	66

## Κατάλογος Σχημάτων

Σχήμα 1: Λειτουργικό διάγραμμα της πλακέτας Arduino Sensor Shield V5.0, πηγή: [9] .....	22
Σχήμα 2: Αρχή λειτουργίας του αποστασιόμετρου υπερήχων HC-SR04, πηγή: [10].....	24
Σχήμα 3: Η διάρκεια των παλμών και οι αντίστοιχες γωνίες περιστροφής, πηγή: [11] .....	26
Σχήμα 4: Σχεδιάγραμμα σύνδεσης του Sensor Shield V5.0 με το Arduino Uno μέσω του προγράμματος Fritzing [16] .....	44
Σχήμα 5: Σχεδιάγραμμα συνδεσμολογίας των servo, του HC- SR04, του DC-DC Step Down Module και των μπαταριών με το Senor Shield V5.0 μέσω του προγράμματος Fritzing [16]	44
Σχήμα 6: Η κάτοψη του ρομπότ .....	56
Σχήμα 7: Προβολή των ποδιών του ρομπότ από αριστερά: Κίνηση προς τα εμπρός - 1 .....	60
Σχήμα 8: Προβολή των ποδιών του ρομπότ από αριστερά: Κίνηση προς τα εμπρός - 2 .....	60
Σχήμα 9: Προβολή των ποδιών του ρομπότ από αριστερά: Κίνηση προς τα πίσω - 1.....	61
Σχήμα 10: Προβολή των ποδιών του ρομπότ από αριστερά: Κίνηση προς τα πίσω - 2.....	62
Σχήμα 11: Προβολή των ποδιών του ρομπότ από αριστερά: Στροφή προς τα δεξιά - 1 .....	63
Σχήμα 12: Προβολή των ποδιών του ρομπότ από αριστερά: Στροφή προς τα δεξιά - 2.....	64
Σχήμα 13: Προβολή των ποδιών του ρομπότ από αριστερά: Στροφή προς τα αριστερά - 1...	65
Σχήμα 14: Προβολή των ποδιών του ρομπότ από αριστερά: Στροφή προς τα αριστερά - 2...	66
Σχήμα 15: Προβολή των ποδιών του ρομπότ από αριστερά: Κίνηση-πόζα Sit Back .....	67
Σχήμα 16: Προβολή των ποδιών του ρομπότ από αριστερά: Κίνηση-πόζα Sit Forward .....	67
Σχήμα 17: Προβολή των ποδιών του ρομπότ από αριστερά: Κίνηση-πόζα Lie.....	68
Σχήμα 18: Προβολή των ποδιών του ρομπότ από αριστερά: Κίνηση-πόζα Stop.....	68

# Κεφάλαιο 1: Διατύπωση του προβλήματος

## 1.1 Εισαγωγή

Η ρομποτική είναι ένας επιστημονικός κλάδος που επικεντρώνεται στο σχεδιασμό, την κατασκευή, τη λειτουργία και τη χρήση ρομπότ. Ένα ρομπότ είναι μια φυσική ή ψηφιακή συσκευή που μπορεί να λειτουργεί ανεξάρτητα ή ημιαυτόνομα. Τα ρομπότ συχνά αναπαράγουν την ανθρώπινη συμπεριφορά ή εκτελούν δραστηριότητες που είναι επικίνδυνες ή ανέφικτες για τους ανθρώπους. Η ρομποτική είναι ένας συνδυασμός από διάφορους τομείς όπως η επιστήμη των υπολογιστών, η μηχανική, η φυσική, τα μαθηματικά και η βιολογία. Έτσι, η ρομποτική μπορεί να χρησιμοποιήσει μια ποικιλία ιδεών και μεθόδων για την κατασκευή ευφυών μηχανών που είναι ικανές να αλληλοεπιδρούν με το πραγματικό περιβάλλον και να εκτελούν αποτελεσματικά εργασίες με μεγάλη ακρίβεια.

Τα ρομπότ με πόδια ή αλλιώς βαδίζοντα ρομπότ (walking robots) μπορούν να χαρακτηριστούν ως αυτόνομα κινούμενα ρομπότ δηλαδή ένα είδος ρομπότ που μπορεί να κινείται στο περιβάλλον του χωρίς να επιβλέπεται άμεσα από άνθρωπο ή να περιορίζεται σε μια προκαθορισμένη πορεία. Αυτά τα ρομπότ έχουν την δυνατότητα να αναπαράγουν την κίνηση του ανθρώπου ή διάφορων ζώων και μπορούν να έχουν έναν πολλαπλό αριθμό ποδιών (2, 4, 6, ...). Συνεπώς, τα ρομπότ με πόδια μπορούν να αντιμετωπίσουν ανώμαλα εδάφη, να ανέβουν σκάλες, να διασχίσουν ανώμαλες επιφάνειες και να λειτουργήσουν σε περιοχές που αποτελούν πρόκληση για τα παραδοσιακά ρομπότ με τροχούς.

Η κατασκευή ενός τέτοιου ρομπότ είναι δυνατόν να πραγματοποιηθεί με την βοήθεια ενός 3D εκτυπωτή. Η τρισδιάστατη εκτύπωση, γνωστή και ως προσθετική κατασκευή, είναι μια επαναστατική τεχνολογία που επιτρέπει τη δημιουργία τρισδιάστατων αντικειμένων με την προσθήκη υλικού στρώμα προς στρώμα. Η τεχνολογία αυτή έχει επιφέρει σημαντικές προόδους σε διάφορους κλάδους, από τη υγειονομική περίθαλψη έως και την εκπαίδευση. Η τεχνολογία Fused Deposition Modeling (FDM) είναι η πιο ευρέως χρησιμοποιούμενη τεχνολογία. Κατασκευάζεται με θέρμανση, ρευστοποίηση και εναπόθεση πλαστικών νημάτων (όπως PLA, ABS, PETG κ.λπ.) σε στρώσεις (layers). Υπάρχουν πολλές διαφορετικές τεχνολογίες τρισδιάστατης εκτύπωσης, η καθεμία με τα δικά της υλικά και τις δικές της αρχές λειτουργίας, όπως:



- Stereolithography (SLA)
- Selective Laser Sintering (SLS)
- Electron Beam Melting (EBM)
- Digital Light Process (DLP)
- Direct Metal Laser Sintering (DMLS)
- Multi Jet Fusion (MJF)
- PolyJet

Για τον έλεγχο της κίνησης ενός τέτοιου ρομπότ μπορεί να χρησιμοποιηθεί ένας μικροελεγκτής, όπως το Arduino Uno. Το Arduino είναι ουσιαστικά μια πλακέτα χαμηλού κόστους που μπορεί να προγραμματιστεί εύκολα για την ανάπτυξη κάποιου αυτοματισμού και έχει την δυνατότητα να συνδεθεί με διάφορους αισθητήρες και άλλα περιφερειακά.



*Εικόνα 1: Ρομπότ με 2 πόδια, πηγή: [1]*



*Εικόνα 2: Ρομπότ με 4 πόδια – Mini Cheetah by MIT, πηγή: [2]*



*Εικόνα 3: Ρομπότ με 6 πόδια, πηγή: [3]*

## 1.2 Ορισμός του προβλήματος

Η παρούσα διπλωματική εργασία ασχολείται με την δημιουργία ενός βαδίζοντος ρομπότ με πόδια το οποίο θα κατασκευαστεί μέσω 3D εκτυπωτή και έπειτα θα προστεθούν μηχανισμοί κίνησης (κινητήρες, σερβομηχανισμοί) που θα ελέγχονται από έναν μικροελεγκτή τύπου Arduino προκειμένου να επιτευχθεί η βάδιση.

Ο σκοπός της εργασίας είναι η δημιουργία ενός ρομπότ με πόδια (legs) που θα έχει την ικανότητα να περπατάει (walking robot). Τα μέρη του ρομπότ θα είναι από πλαστικό και θα μπορούν να εκτυπωθούν από τον 3D εκτυπωτή. Το ρομπότ μπορεί να περιλαμβάνει από 2 έως και N πόδια, ωστόσο στην συγκεκριμένη εργασία επιλέχθηκε να έχει 4 πόδια. Οι προδιαγραφές που τέθηκαν είναι οι εξής:

1. Η κατασκευή όλων των πλαστικών μερών του ρομπότ θα γίνει μέσω του 3D printer Wanhao Duplicator 6 που προμηθεύτηκε πρόσφατα ο Τομέας και η συναρμολόγησή τους.
2. Η προσθήκη μοτέρ ή σερβομηχανισμών, στην ρομποτική κατασκευή για τον έλεγχο της κίνησης των ποδιών και του βαδίσματος.
3. Η διασύνδεση των σερβομηχανισμών με μικροελεγκτή τύπου Arduino, και με την ενσωμάτωση κατάλληλης πηγής ενέργειας (π.χ. μπαταρία) για τα servo.
4. Η ανάπτυξη βασικού λογισμικού για τον έλεγχο του ρομποτικού μηχανισμού που θα μπορεί να υλοποιεί το βάδισμα του ρομπότ.
5. Η δυνατότητα το ρομπότ να έχει, όχι μόνο κίνησης προς τα εμπρός, αλλά και προς τα πίσω και ίσως και προς τα πλάγια (omnidirectional walking).
6. Η προσθήκη αποστασιόμετρου έτσι ώστε να είναι δυνατή η ανάπτυξη αλγορίθμων αυτόνομης κίνησης ή και αποφυγής εμποδίων.
7. Είναι επιθυμητή η δυνατότητα υπερπήδησης μικρών εμποδίων π.χ. μικρού σκαλοπατιού
8. Είναι επιθυμητή η σύνδεση του Arduino με PC μέσω WiFi για τον τηλεχειρισμό του μέσω του PC.

## 1.3 Οφέλη της διερεύνησης του προβλήματος

Η διεξαγωγή μιας μελέτης για ένα ρομπότ με πόδια που χρησιμοποιεί τρισδιάστατη εκτύπωση και Arduino είναι εξαιρετικά σημαντική επειδή συνδυάζει μια σειρά από σημαντικά πεδία που θα μπορούσαν να φέρουν επανάσταση σε πολλούς

τομείς της σύγχρονης τεχνολογίας και της εκπαίδευσης. Η μελέτη αυτή συνδυάζει τη ρομποτική, την τεχνολογία 3D εκτύπωσης και τον προγραμματισμό μικροελεγκτών (Arduino). Η ενασχόληση με τέτοια έργα συμβάλει στην ανάπτυξη δεξιοτήτων σε διάφορα επιστημονικά πεδία και βοηθά στην κατανόηση του τρόπου με τον οποίο διαφορετικές τεχνολογίες μπορούν να συνεργάζονται για τη δημιουργία καινοτόμων λύσεων.

Η κατασκευή ενός τέτοιου ρομπότ, επιτρέπει στους φοιτητές να μεταβούν από τις θεωρητικές γνώσεις στην πρακτική εφαρμογή, ενώ παράλληλα έρχονται σε επαφή με τεχνολογίες αιχμής. Αποκτούν έτσι εμπειρία στο σχεδιασμό, την εκτύπωση, τη συναρμολόγηση και τον προγραμματισμό ενός λειτουργικού ρομπότ. Εξερευνούν την πλατφόρμα Arduino, ενός σημαντικού και ευέλικτου εργαλείου στον τομέα των ηλεκτρονικών και της ρομποτικής και έρχονται σε επαφή με ένα ευρύ φάσμα κινητήρων και αισθητήρων. Η ενασχόλησή με αυτές τις τεχνολογίες όχι μόνο εξοπλίζει τους φοιτητές με σύγχρονες δεξιότητες, ενισχύοντας τις ικανότητές τους στην επίλυση προβλημάτων, αλλά και καλλιεργεί την προσαρμοστικότητα τους στις εξελισσόμενες τεχνολογικές τάσεις.

Επιπλέον, σε έναν συνεχώς εξελισσόμενο κόσμο η ρομποτική και η βιομηχανία διασταυρώνονται. Η τεχνολογία της ρομποτικής έχει διεισδύσει σε βιομηχανίες σε παγκόσμιο επίπεδο, αναδιαμορφώνοντας τις παραδοσιακές εργασίες και φέρνοντας επανάσταση στην κατασκευή, την αυτοματοποίηση και τις επιχειρησιακές διαδικασίες. Η ενασχόληση με τη μελέτη και την κατασκευή ρομπότ με πόδια χρησιμοποιώντας 3D εκτύπωση και Arduino παρέχει στους φοιτητές πολύπλευρα οφέλη, καθώς τους εξοπλίζει με πρακτικές δεξιότητες και γνώσεις που αναζητούνται από τις εταιρείες τεχνολογίας, τοποθετώντας τους πλεονεκτικά σε μια βιομηχανία που αγκαλιάζει όλο και περισσότερο τη ρομποτική και την αυτοματοποίηση. Επιπλέον, τους προετοιμάζει για την αυξανόμενη ζήτηση σε τομείς όπως η έρευνα και διάσωση, η εξερεύνηση και η γεωργία, όπου τα ρομπότ που είναι ικανά να περιηγούνται σε περίπλοκα εδάφη είναι ζωτικής σημασίας.

Συνοψίζοντας, τα πλεονεκτήματα που παρουσιάζονται μέσω της διερεύνησης αυτού του προβλήματος καλύπτουν ένα μεγάλο φάσμα της τεχνολογίας, της απόκτησης δεξιοτήτων και της πρακτικής μάθησης. Η ενασχόληση με την 3D εκτύπωση, το Arduino, την συνδεσιμότητα μικροελεγκτών και του εξειδικευμένου προγραμματισμού συμβάλλει ουσιαστικά στη δημιουργία έμπειρων και καινοτόμων ανθρώπων, έτοιμων να συνεισφέρουν ουσιαστικά σε μια ταχέως αναπτυσσόμενη κοινωνία.

## Κεφάλαιο 2: Κατάσταση της Τέχνης

### 2.1 Σχεδίαση και 3D Εκτύπωση

Η 3D σχεδίαση αναφέρεται στη δημιουργία τρισδιάστατων αντικειμένων με τη χρήση ψηφιακών εργαλείων και λογισμικού. Σε αντίθεση με την συνήθη 2D σχεδίαση, η οποία περιορίζεται σε μήκος και πλάτος, η τρισδιάστατη σχεδίαση δίνει και ύψος στο αντικείμενο. Η 3D σχεδίαση είναι άμεσα συνδεδεμένη με την 3D εκτύπωση, γνωστή και ως προσθετική κατασκευή. Μια πρωτοποριακή τεχνολογία που δίνει την δυνατότητα δημιουργίας τρισδιάστατων αντικειμένων με την τοποθέτηση υλικού σε στρώσεις. Η τεχνολογία αυτή έχει εξελιχθεί ραγδαία και είναι πλέον προσιτή σε μεγάλο αριθμό ανθρώπων και χρησιμοποιείται ευρέως σε διάφορους κλάδους, όπως η υγειονομική περίθαλψη, η αεροδιαστημική, η αυτοκινητοβιομηχανία, η οδοντιατρική, η εκπαίδευση, η αρχιτεκτονική κ.α.. Τα 3D μοντέλα μπορούν να κατασκευαστούν από το μηδέν χρησιμοποιώντας ένα πρόγραμμα 3D Computer-Aided Design (CAD). [4]. Υπάρχουν διάφορα προγράμματα 3D CAD, κάποια πιο φιλικά προς τους αρχάριους έως και για πιο προχωρημένους, όπως:

1. **TinkerCAD**: Κατάλληλο για αρχάριους και εκπαιδευτικούς για την εκμάθηση των βασικών στοιχείων του τρισδιάστατου σχεδιασμού
2. **SketchUp**
3. **FreeCAD**
4. **Blender**
5. **Autodesk Fusion 360**
6. **Autodesk AutoCAD**
7. **SolidWorks**
8. **CATIA**
9. **Rhino (Rhinoceros)**
10. **OpenSCAD**

Πέρα από την δημιουργία ενός 3D ψηφιακού μοντέλου ενός αντικειμένου χρησιμοποιώντας ένα από τα πολλά διαθέσιμα πακέτα λογισμικού 3D μοντελοποίησης, υπάρχει ως επιλογή και η 3D σάρωση (3D-Scan), δηλαδή να σαρωθεί κάτι που ήδη υπάρχει, ή η λήψη κάποιου αντικειμένου από διάφορες ιστοσελίδες όπου διατίθενται δωρεάν για λήψη τα STL αρχεία. Το STL είναι μια συνήθης μορφή αρχείου που

χρησιμοποιείται στην 3D εκτύπωση. Αναφέρεται, επίσης, και ως Standard Triangle Language ή Standard Tessellation Language.

Η 3D εκτύπωση μπορεί να πραγματοποιηθεί με διάφορες τεχνολογίες, με πιο δημοφιλή τεχνολογία την FDM που χρησιμοποιεί πλαστικά νήματα (PLA, ABS, PETG κ.λπ.) που θερμαίνονται, ρευστοποιούνται και τοποθετούνται σε στρώσεις ώστε να δημιουργηθεί το αντικείμενο. Με την χρήση ενός λογισμικού τεμαχισμού (slicing software), μετατρέπουμε τα τρισδιάστατα μοντέλα σε μια σειρά οδηγιών που ο τρισδιάστατος εκτυπωτής μπορεί να ακολουθήσει για την κατασκευή του κάθε στρώματος του αντικειμένου. Τέτοια προγράμματα είναι το Simplify3D, το Wanhao Cura του εκτυπωτή Wanhao Duplicator 6 κ.α.. όπου μπορούν να μετατρέψουν τα STL αρχεία σε G-Code. Ορισμένες από τις λειτουργίες των προγραμμάτων αυτών είναι:

- **Τεμαχισμός (Slicing):** Το slicing software τεμαχίζει ένα τρισδιάστατο αντικείμενο με μορφή STL ή OBJ σε στρώματα. Αυτά τα στρώματα μετατρέπονται στη συνέχεια σε οδηγίες για τον τρισδιάστατο εκτυπωτή.
- **Προφίλ υλικών:** Αυτά τα προγράμματα υποστηρίζουν ένα ευρύ φάσμα νημάτων και υλικών και συχνά περιλαμβάνουν προ-ρυθμισμένα προφίλ για κοινούς τύπους νημάτων. Οι χρήστες μπορούν επίσης να σχεδιάσουν τα δικά τους προφίλ υλικών.
- **Προεπισκόπηση εκτύπωσης:** Αυτή η λειτουργία εμφανίζει μια οπτική αναπαράσταση του μοντέλου σε φέτες, επιτρέποντας στους χρήστες να δουν πώς θα εμφανίζεται η εκτύπωση στρώμα προς στρώμα πριν την υποβάλουν στον εκτυπωτή.
- **Προσαρμόσιμες παράμετροι:** Για να αποκτήσουν την απαιτούμενη ποιότητα εκτύπωσης και τα χαρακτηριστικά, οι χρήστες μπορούν να μεταβάλλουν πολυάριθμες παραμέτρους εκτύπωσης, όπως το ύψος στρώματος, η πυκνότητα γεμίσματος, η ταχύτητα εκτύπωσης, η θερμοκρασία, οι δομές στήριξης κ.α. [5]





Εικόνα 4: Ο 3D εκτυπωτής Wanhao Duplicator 6

Πίνακας 1: Κάποια βασικά χαρακτηριστικά του 3D Printer Wanhao Duplicator 6, πηγή: [6]

<b>Printing</b>	
Print technology	Fused filament fabrication (FFF)
Build volume	20 x 20 x 17.5 cm
Layer resolution	Ultra high: 20 micron High: 60 micron Medium: 100 micron Low: 200 micron
Position precision	X 12,5 micron Y 12,5 micron Z 5 micron
Filament diameter	1.75mm
Nozzle diameter	0.4 mm
Print speed	30 mm/1 - 300 mm/s

<b>Temperature</b>	
Ambient Operating Temperature	15° - 32° C
Storage Temperature	0° - 32° C
Operating nozzle temperature	180° - 260°
Operating heated build plate temperature	50° - 100°

Κάποια βασικά δομικά στοιχεία ενός 3D εκτυπωτή είναι τα εξής:

- **Bed/Build plate:** Η επιφάνεια εκτύπωσης του τρισδιάστατου εκτυπωτή. Μπορεί να είναι κατασκευασμένο από διάφορα υλικά, όπως πλαστικό, μέταλλο ή γυαλί.
- **Build area:** Το συνολικό μέγεθος της εκτυπώσιμης περιοχής που καθορίζει το μέγιστο μέγεθος ενός αντικειμένου που μπορεί να εκτυπωθεί. Αυτό περιλαμβάνει τις διαστάσεις XYZ του πλάτους, του ύψους και του βάθους.
- **Extruder:** Η κεφαλή εκτύπωσης είναι μια μηχανοκίνητη συσκευή με δύο τμήματα: το ψυχρό άκρο και το θερμό άκρο. Το ψυχρό άκρο τραβά νήμα και το τροφοδοτεί στο θερμό άκρο, το οποίο θερμαίνει το νήμα πριν αυτό βγει από το ακροφύσιο (Nozzle) στην περιοχή κατασκευής (Build area).
- **Nozzle:** Μια μικρή μεταλλική κεφαλή με οπή σταθερού μεγέθους, μέσω της οποίας το θερμαινόμενο πλαστικό περνάει από το Extruder στην περιοχή κατασκευής (Build area). Χρησιμοποιούνται nozzle διαφορετικού μεγέθους ανάλογα με τον τύπο του εκτυπώσιμου αντικειμένου. [5]

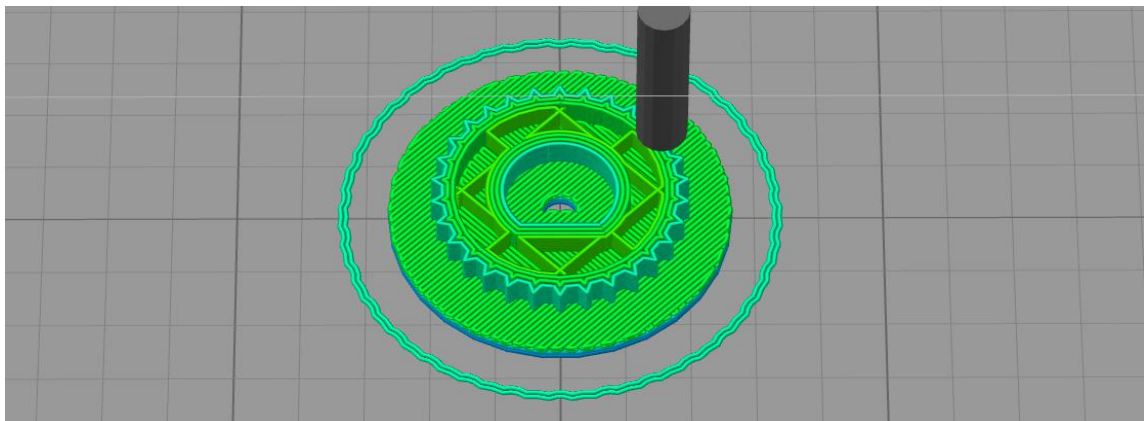
Οι χρήστες μπορούν να ρυθμίσουν πολλές παραμέτρους στο 3D printing slicing software για να επιτύχουν συγκεκριμένα αποτελέσματα στις εκτυπώσεις τους, όπως:

- **Ύψος στρώσης (Layer Height):** Το πάχος κάθε στρώματος καθορίζεται από το layer height. Τα μεγαλύτερα ύψη στρώματος μπορούν να επιταχύνουν τη διαδικασία εκτύπωσης, αλλά επιβαρύνουν την ποιότητα της επιφάνειας του αντικειμένου.
- **Πυκνότητα γεμίσματος (Infill Density):** Η ποσότητα του υλικού στο εσωτερικό της τρισδιάστατης εκτύπωσης ελέγχεται από το Infill Density και μπορεί να πάρει διάφορα μοτίβα. Η υψηλότερη πυκνότητα ενισχύει την αντοχή του αντικειμένου, αλλά αυξάνει τον χρόνο εκτύπωσης και τη ποσότητα του υλικού που απαιτείται.
- **Ταχύτητα εκτύπωσης (Print Speed):** Η ταχύτητα εκτύπωσης επηρεάζει το πόσο γρήγορα κινείται η κεφαλή εκτύπωσης (extruder), κατά την εκτύπωση. Οι

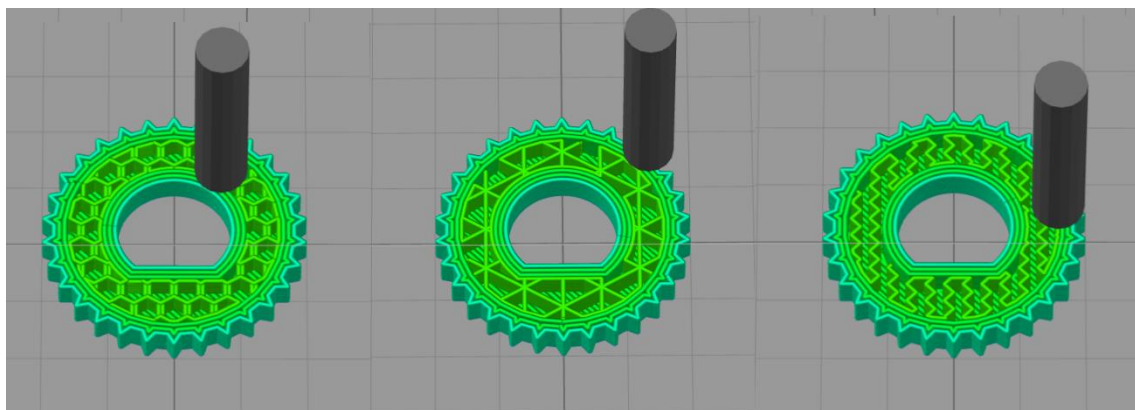


υψηλότερες ταχύτητες μπορούν να μειώσουν το χρόνο εκτύπωσης, αλλά ενδέχεται να επηρεάσουν την ποιότητα. Οι χαμηλότερες ταχύτητες μπορούν να βελτιώσουν την ακρίβεια, ειδικά για περίπλοκες λεπτομέρειες.

- **Έλεγχοι θερμοκρασίας (Temperature Controls):** Οι θερμοκρασίες του extruder και του bed έχουν αντίκτυπο στον τρόπο με τον οποίο το νήμα λιώνει και κολλάει στο build plate. Οι θερμοκρασίες που μπορούν να πάρουν ποικίλλουν ανάλογα με το υλικό. Οι σωστές ρυθμίσεις θερμοκρασίας είναι απαραίτητες για τη δημιουργία επαρκούς προσκόλλησης του αντικειμένου στο build plate και την σωστή τήξη του υλικού.
- **Ρυθμίσεις ανάσχυσης (Retraction Settings):** Οι παράμετροι Retraction ρυθμίζουν τον τρόπο με τον οποίο το νήμα ωθείται προς τα πίσω κατά τη διάρκεια κινήσεων που δεν αφορούν την εκτύπωση για να αποφευχθεί η διαρροή ή η δημιουργία κλωστών (stringing).
- **Δομές υποστήριξης (Support Structures):** Οι δομές στήριξης χρησιμεύουν ως προσωρινές “σκαλωσιές” για τα τμήματα της εκτύπωσης που προεξέχουν. Η ρύθμιση των παραμέτρων στήριξης μπορεί να έχει επίδραση στην ευκολία αφαίρεσης και την σταθερότητα του αντικειμένου κατά την διάρκεια της εκτύπωσης. Ένας άλλος τρόπος για να επιτευχθεί η σταθερότητα είναι ο διαχωρισμός του αντικειμένου σε κομμάτια.
- **Raft and Brim:** Ρυθμίσεις του raft και brim αυξάνουν την προσκόλληση και τη σταθερότητα της εκτύπωσης. Οι βάσεις (raft) είναι 4-5 στρώσεις υλικού που εκτυπώνονται πρώτα κάτω από ολόκληρο το print, ενώ τα brim είναι λεπτές γραμμές γύρω από τη βάση. Χρησιμοποιούνται ειδικά για εκτυπώσεις με μικρή βάση ή για εκτυπώσεις με τάση για παραμόρφωση. [5]



Εικόνα 5: Προεπισκόπηση εκτύπωσης στο Simplify3D όπου φαίνεται το brim, το raft και το αντικείμενο σε layers με infill 15%



Εικόνα 6: Διάφορα μοτίβα infill στην προβολή εκτύπωσης του Simplify3D

Συνοψίζοντας, τα βήματα που πρέπει ακολουθηθούν από την σχεδίαση μέχρι την ολοκληρωμένη εκτύπωση ενός αντικειμένου είναι τα εξής:

1. **Σχεδιασμός ή λήψη ενός τρισδιάστατου μοντέλου:** Δημιουργία ενός 3D μοντέλου χρησιμοποιώντας λογισμικό (CAD) ή λήψη 3D μοντέλου από το διαδίκτυο.
2. **Τεμαχισμός του μοντέλου:** Χρησιμοποιώντας λογισμικό τεμαχισμού (π.χ. Cura, Simplify3D) για την μετατροπή του 3D μοντέλου σε G-Code, ο οποίος περιέχει οδηγίες για τον 3D Printer σχετικά με τον τρόπο κατασκευής κάθε στρώματος του αντικειμένου.
3. **Τοποθέτηση του νήματος:** Εισαγωγή του νήματος στον 3D εκτυπωτή. Ο τύπος νήματος εξαρτάται από το υλικό με το οποίο θα γίνει η εκτύπωση (π.χ. PLA, ABS, PETG). Ακολουθώντας το εγχειρίδιο του εκτυπωτή για οδηγίες σχετικά με την τοποθέτηση του νήματος.
4. **Βαθμονόμηση τον εκτυπωτή:** Η σωστή βαθμονόμηση είναι απαραίτητη για επιτυχημένες εκτυπώσεις και γίνεται μια συνήθως φόρα πριν την πρώτη εκτύπωση του εκτυπωτή. Πολλοί 3D εκτυπωτές διαθέτουν ενσωματωμένες ρουτίνες βαθμονόμησης ή χειροκίνητες ρυθμίσεις.
5. **Φόρτωση του G-Code:** Μεταφορά του αρχείου G-code που δημιουργήθηκε από το λογισμικό τεμαχισμού στον 3D εκτυπωτή. Αυτό μπορεί να γίνει μέσω USB, κάρτας SD ή μέσω απευθείας σύνδεσης, ανάλογα με το μοντέλο του εκτυπωτή.
6. **Εκκίνηση της εκτύπωσης:** Εκκίνηση της διαδικασίας εκτύπωσης μέσω του Navigation Dial και της LCD οθόνης του 3D εκτυπωτή. Ο εκτυπωτής θα ακολουθήσει τις οδηγίες του G-Code για να δημιουργήσει το αντικείμενο στρώμα προς στρώμα.

7. **Προθέρμανσή του εκτυπωτή:** Ο 3D εκτυπωτής θερμαίνει το build plate και το extruder στη θερμοκρασία που επιλέχθηκε στο G-Code. Αυτό εξασφαλίζει την ομαλή ροή του νήματος κατά τη διαδικασία εκτύπωσης.
8. **Παρακολούθηση της εκτύπωσης:** Επίβλεψη της εκτύπωσης καθ' όλη τη διάρκεια της διαδικασίας για τον έλεγχο τυχόν προβλημάτων, όπως προβλήματα πρόσφυσης στο bed, εμπλοκές νημάτων ή προβλήματα πρόσφυσης στρώσεων.
9. **Μεταγενέστερη επεξεργασία:** Μόλις ολοκληρωθεί η εκτύπωση, γίνεται προσεκτική αφαίρεση του αντικείμενου από το build plate. Τα βήματα μετα-επεξεργασίας μπορεί να περιλαμβάνουν την αφαίρεση των δομών στήριξης ή την εξομάλυνση τραχειών άκρων.
10. **Αντιμετώπιση προβλημάτων:** Τα συνήθη προβλήματα περιλαμβάνουν προβλήματα πρόσφυσης στρώματος, παραμόρφωσης, κακής ποιότητας εκτύπωση ή κακή ευθυγράμμιση της εκτύπωσης.

## 2.2 Arduino Uno

Από το 2003, η Arduino παράγει πλακέτες και αρχιτεκτονική ενσωματωμένων συστημάτων ανοικτού κώδικα. Με πιο δημοφιλές το Arduino Uno R3. Το Arduino Uno βασίζεται στον μικροελεγκτή Atmel ATmega328P. Αυτός ο μικροελεγκτής είναι ο εγκέφαλος της πλακέτας Arduino, υπεύθυνος για την εκτέλεση του προγράμματος και τον έλεγχο του συνδεδεμένου υλικού.

Ο ATmega328P είναι ένας μικροελεγκτής 8-bit και βασίζεται σε αρχιτεκτονική RISC (Reduced Instruction Set Computing), η οποία μειώνει την εκτέλεση εντολών για ταχύτερη λειτουργία. Περιέχει 32 kilobytes (KB) μνήμης Flash, η οποία χρησιμοποιείται για την αποθήκευση του κώδικα, 2 KB μνήμης (SRAM) για την αποθήκευση προσωρινών δεδομένων κατά τη διάρκεια της εκτέλεσης του προγράμματος και 1 KB μνήμης μόνο για ανάγνωση (EEPROM). Έχει 23 ψηφιακές γραμμές I/O γενικού σκοπού και 32 καταχωρητές γενικού σκοπού. Ο μικροελεγκτής υποστηρίζει σειριακή επικοινωνία μέσω των διεπαφών UART (Universal Asynchronous Receiver-Transmitter), SPI (Serial Peripheral Interface) και I2C (Inter-Integrated Circuit) και περιέχει έναν αριθμό χρονομετρητών/μετρητών που απαιτούνται για τον καθορισμό ακριβών χρονικών καθυστερήσεων και τη διαχείριση σημάτων PWM (Pulse Width Modulation).

Λειτουργεί κανονικά στα 16 MHz, εξισορροπώντας την κατανάλωση ενέργειας και την απόδοση επεξεργασίας. Η πλακέτα διαθέτει 14 ψηφιακά I/O pin (με αριθμούς 0-13) από τα οποία τα 6 έχουν δυνατότητα εξόδου σήματος PWM και 6 pin αναλογικής εισόδου που χρησιμοποιούνται για την ανάγνωση αναλογικών πληροφοριών αισθητήρα. Διαθέτει USB θύρα για σύνδεση με PC μέσω της οποίας προγραμματίζεται και τροφοδοτείται με τάση 6-20V (συνιστώμενη 7-12V), ενώ λειτουργεί στα 5V. Ο προγραμματισμός του γίνεται μέσω του Arduino IDE, το οποίο υποστηρίζει τη σύνδεση με το Arduino μέσω θύρας USB και επιτρέπει στον χρήστη να δημιουργήσει ένα νέο πρόγραμμα, να το αποθηκεύσει, να το μεταγλωττίσει σε κώδικα μηχανής και να το φορτώσει στο Arduino.

Η δομή του προγράμματος αποτελείται από δύο συναρτήσεις με σταθερά ονόματα, τις `setup()` και `loop()`. Η `setup()` εκτελείται μία μόνο φορά κατά την έναρξη του προγράμματος ή μετά από κάθε επανεκκίνηση του Arduino (reset). Η `loop()` εκτελείται αμέσως μετά από την `setup()` και επαναλαμβάνεται μέχρι το επόμενο reset. Επομένως, στην `loop()` βάζουμε το κυρίως πρόγραμμα της εφαρμογής μας, το οποίο θα εκτελείται συνέχεια, όσο λειτουργεί το Arduino.

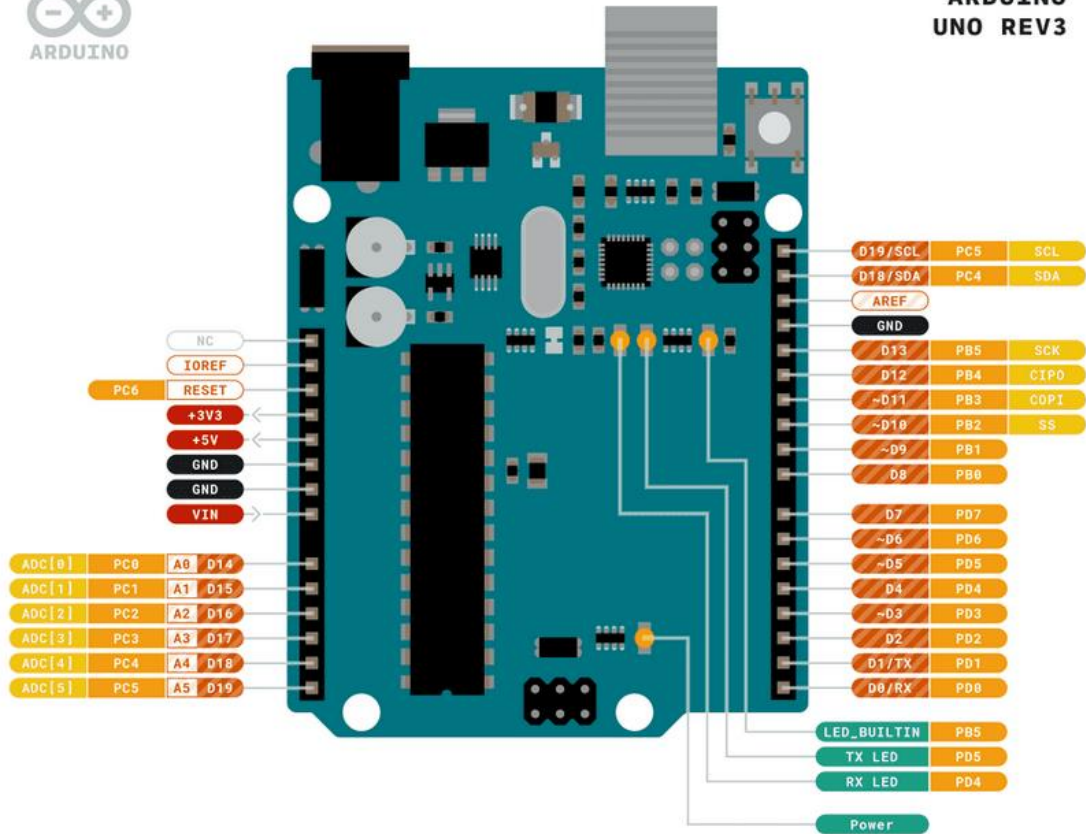
Συνεπώς, το Arduino Uno χρησιμοποιείται για ένα ευρύ φάσμα εφαρμογών σε διάφορους τομείς όπως: ρομποτική, εκπαίδευση, υγειονομική περίθαλψη, γεωργία, επιστημονική έρευνα, αυτοκινητοβιομηχανία κ.α. Χρησιμεύει ως μια ευέλικτη πλακέτα μικροελεγκτή που επιτρέπει στους χρήστες να προγραμματίζουν και να ελέγχουν εύκολα διάφορους αισθητήρες, ενεργοποιητές και άλλα ηλεκτρονικά εξαρτήματα. [7]



Εικόνα 7: Το Arduino Uno R3, πηγή: [8]



## ARDUINO UNO REV3

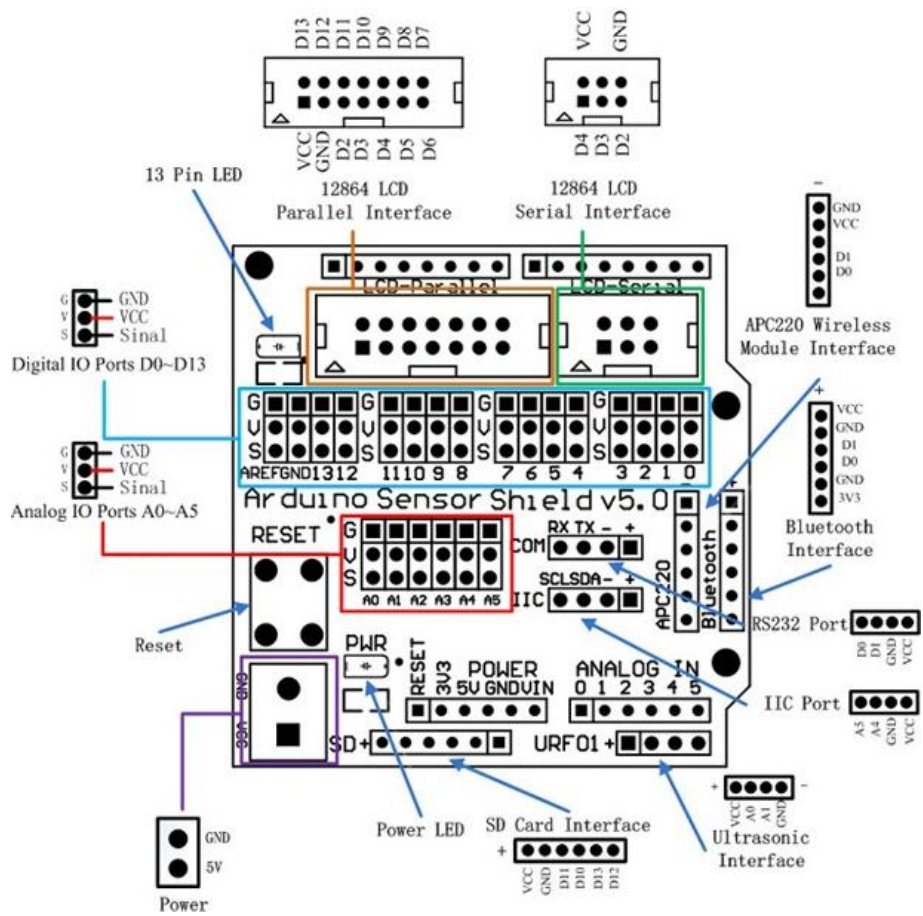


Εικόνα 8: Διάγραμμα ακροδεκτών της πλακέτας Arduino Uno, πηγή: [8]

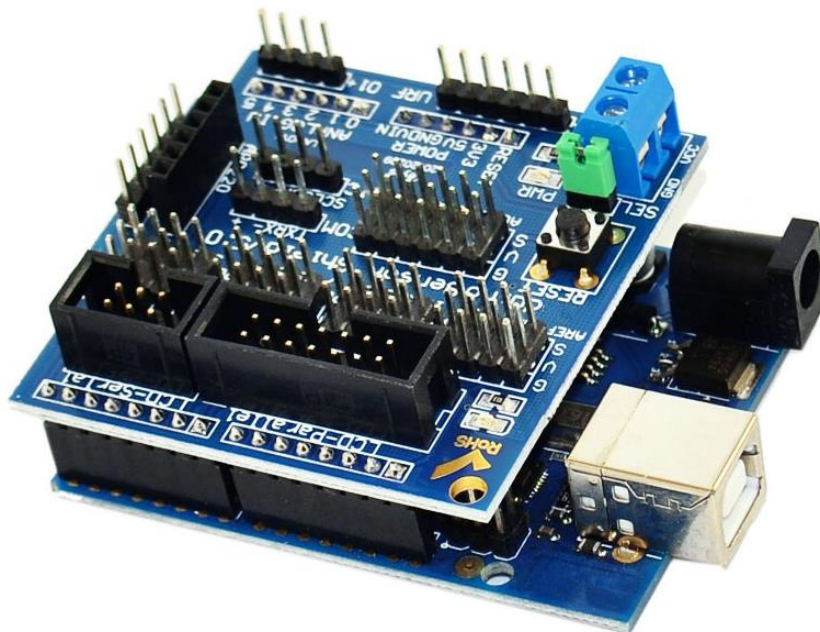
Μια πλακέτα επέκτασης για πλακέτες μικροελεγκτών Arduino είναι το Sensor Shield V5.0. Το Shield έχει σχεδιαστεί ώστε να ευθυγραμμίζεται με τα pin του Arduino Uno, διατηρώντας μια τυποποιημένη διάταξη για ευκολία χρήσης. Διαθέτει πρόσθετα pin, ψηφιακά και αναλογικά, για τη σύνδεση περισσότερων εξαρτημάτων, διευκολύνοντας τη σύνδεση πολλαπλών αισθητήρων και σερβομηχανισμών σε ένα Arduino. Μπορεί να δεχτεί εξωτερική τροφοδοσία από μπαταρίες και αυτό διευκολύνει την τροφοδοσία των συνδεδεμένων αισθητήρων και μονάδων απευθείας από την πλακέτα. Για την εύκολη επανεκκίνηση της πλακέτας Arduino και των συνδεδεμένων εξαρτημάτων διαθέτει ένα reset button. Ορισμένες εκδόσεις του Sensor Shield διαθέτουν ενσωματωμένες λυχνίες LED που μπορούν να χρησιμοποιηθούν για ένδειξη κατάστασης ή αποσφαλμάτωση. Καθώς, επίσης, μπορεί να περιλαμβάνουν ειδικές θύρες για επικοινωνία I2C και UART, απλοποιώντας τη σύνδεση συσκευών που χρησιμοποιούν αυτά τα πρωτόκολλα. Συνολικά, το Sensor Shield V5.0 απλοποιεί τη



διαδικασία σύνδεσης αισθητήρων και μονάδων σε ένα Arduino, καθιστώντας το ένα πολύτιμο εργαλείο τόσο για αρχάριους όσο και για έμπειρους χρήστες. [9]



Σχήμα 1: Λειτουργικό διάγραμμα της πλακέτας Arduino Sensor Shield V5.0, πηγή: [9]



Εικόνα 9: Το Sensor Shield V5.0 συνδεδεμένο επάνω στο Arduino Uno

## 2.3 Αισθητήρες και σερβομηχανισμοί

Σε ρομποτικές εφαρμογές αυτόνομης κίνησης παίζουν πολύ σημαντικό ρόλο οι αισθητήρες μέτρησης απόστασης. Υπάρχουν διάφοροι τύποι τέτοιων αισθητήρων, με πιο σύνηθες το κοινό αποστασιόμετρο υπερήχων HC-SR04. Η αρχή λειτουργίας του είναι η εξής: Ο αισθητήρας χρησιμοποιεί έναν πομπό για να στείλει μια δέσμη υπερήχων προς ένα αντικείμενο. Αυτή ανακλάται από το αντικείμενο, επιστρέφει πίσω και λαμβάνεται από τον δέκτη του αισθητήρα. Έτσι, ο αισθητήρας υπολογίζει την απόσταση από το αντικείμενο, βάση του χρόνου που χρειάστηκε η δέσμη αυτή για να ταξιδέψει μέχρι το αντικείμενο και πίσω.

Η απόσταση υπολογίζεται ως εξής:

$$\text{Απόσταση} = \text{χρόνος\_αποστολής\_λήψης} * \text{ταχύτητα\_ήχου} / 2$$

Η ταχύτητα του ήχου είναι 0,034 cm/msec και  $0,034 \approx 1/29$ . Συνεπώς, για να μην γίνονται πράξεις με δεκαδικά μπορούμε να πούμε:

$$\text{Απόσταση} = \text{χρόνος\_αποστολής\_λήψης} / 29 / 2, \text{ ή}$$

$$\text{Απόσταση} = \text{χρόνος\_αποστολής\_λήψης} / 58$$

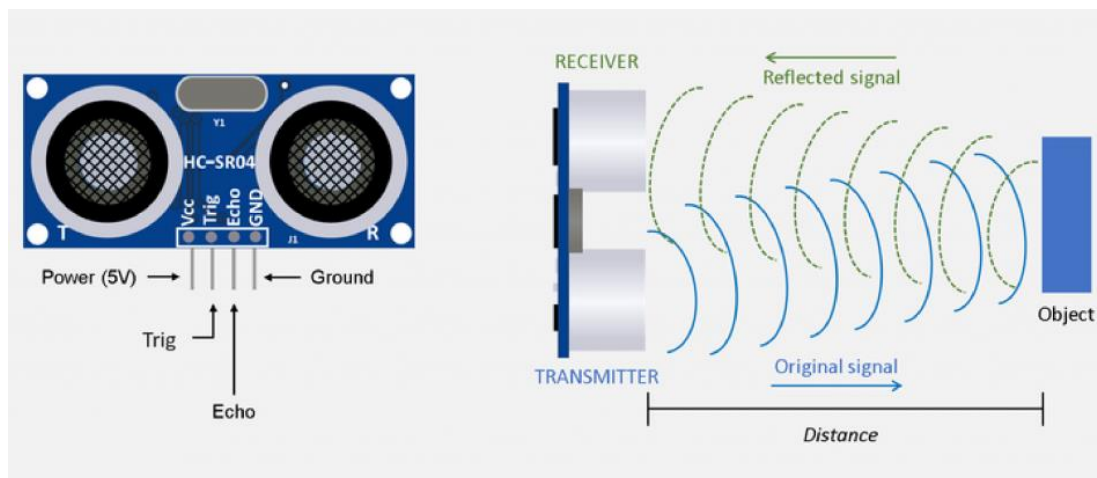
Ο αισθητήρας HC-SR04 έχει τάση τροφοδοσίας 5V, κατανάλωση ρεύματος 15mA, γωνία μέτρησης 15 μοίρες και εμβέλεια μέτρησης: 2cm – 4m. Η λειτουργία του μπορεί να επηρεαστεί από διάφορους παράγοντες του περιβάλλοντος και η μέτρηση του μπορεί να είναι ανακριβής σε μαλακές ή ακανόνιστες επιφάνειες.

Ο αισθητήρας αυτός έχει 4 pin που είναι:

1. **VCC**: σύνδεση της τάσης τροφοδοσίας που είναι τυπικά 5V,
2. **Trig**: παρέχουμε σύντομο παλμό HIGH για να παραχθεί ο υπερηχητικός παλμός και να ξεκινήσει η μέτρηση χρόνου επιστροφής,
3. **Echo**: παράγει έναν θετικό παλμό (HIGH) από την στιγμή αποστολής του υπερηχητικού παλμού έως και την λήψη της ανάκλασης, οπότε το pin αυτό γίνεται LOW. Με μέτρηση της διάρκειας αυτού του παλμού μπορούμε να υπολογίσουμε τον συνολικό χρόνο αποστολής, ανάκλασης και λήψης του υπερηχητικού παλμού,
4. **GND**: η γείωση για την τροφοδοσία του αισθητήρα.

Η ανάγνωση της διάρκειας του παλμού στο pin Echo μπορεί να γίνει με την συνάρτηση pulseIn(). Σε περίπτωση που το εμπόδιο μπροστά από τον αισθητήρα είναι πολύ μακριά (> 4 μέτρα) ή υπό γωνία, η δέσμη υπερήχων ίσως να μην ανακλαστεί ποτέ προς τον δέκτη του αισθητήρα. Για την περίπτωση αυτή, υπάρχει και η εναλλακτική σύνταξη της συνάρτησης pulseIn με τρεις παραμέτρους, όπου η 3η παράμετρος είναι ένας χρόνος σε msec και χαρακτηρίζεται ως χρόνος timeout. Όταν παρέλθει, η συνάρτηση επιστρέφει με return, ακόμα και αν ο παλμός στο pin Echo διατηρείται ακόμα HIGH, περιμένοντας την επιστροφή της υπερηχητικής δέσμης. Σε αυτή την περίπτωση η επιστρεφόμενη τιμή διάρκειας παλμού είναι 0. Η τιμή 0 σημαίνει ότι μπορεί το συγκεκριμένο pin να έχει «κολλήσει» σε μόνιμη κατάσταση HIGH και για να ξεκολλήσουμε αυτή την κατάσταση μπορούμε να κάνουμε τα εξής:

1. Ορισμός του pin ως έξοδος
2. Αποστολή στάθμης LOW στο pin
3. Ορισμός του pin ξανά ως είσοδος [7]



Σχήμα 2: Αρχή λειτουργίας του αποστασιόμετρου υπερήχων HC-SR04, πηγή: [10]



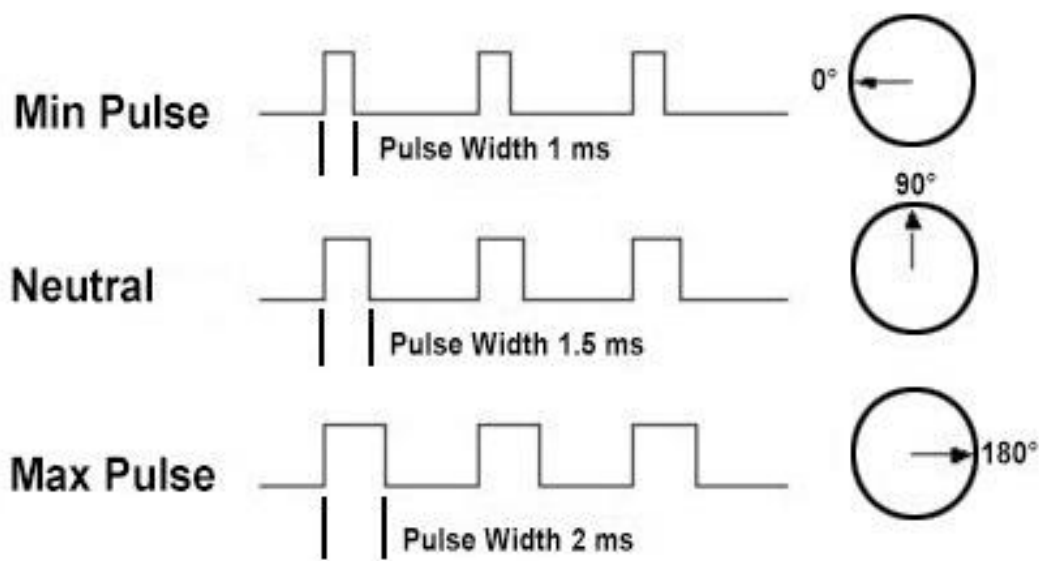


Εικόνα 10: Ο σερβομηχανισμός Servo Micro 2.2kg.cm Plastic Gears (Waveshare SG90)

Ο σερβομηχανισμός είναι ένας περιστροφικός ή γραμμικός ενεργοποιητής που ρυθμίζει με ακρίβεια τη θέση, την ταχύτητα και, σε ορισμένες περιπτώσεις, την επιτάχυνση ενός μηχανικού συστήματος. Τα σερβομηχανήματα χρησιμοποιούνται εκτενώς σε διάφορες εφαρμογές όπου απαιτείται ακριβής έλεγχος της κίνησης. Η γωνία περιστροφής ενός σερβομηχανισμού είναι συνήθως από 0 έως 180 μοίρες, ο μικρότερος έχει βάρος περίπου 9 γραμμάρια και διαστάσεις: 22.8mm × 12.2mm × 28.5mm. Η ροπή του κυμαίνεται από 1.5 έως 3 Kg\*cm και η τυπική τάση τροφοδοσίας του είναι 5V, ενώ μπορεί να λειτουργήσει και με τάσεις: 4.8 ~ 6.5V και σε θερμοκρασίες από 0 έως 55 βαθμούς κελσίου. [7]

Το Servo ελέγχεται από 3 ακροδέκτες– καλώδια:

1. **Καλώδιο Τροφοδοσίας (κόκκινο):** Συνδέεται στην τάση τροφοδοσίας 5V
2. **Καλώδιο γείωσης (μαύρο):** Συνδέεται στον αρνητικό πόλο της τροφοδοσίας
3. **Καλώδιο σήματος (πορτοκαλί ή κίτρινο ή άσπρο):** Σε αυτό παρέχεται σήμα ένα PWM, όπου η διάρκεια του παλμού καθορίζει και την γωνία στροφής της κεφαλής του Servo



Σχήμα 3: Η διάρκεια των παλμών και οι αντίστοιχες γωνίες περιστροφής, πηγή: [11]

Για τον εύκολο προγραμματισμό και χειρισμό των σερβομηχανισμών χρησιμοποιείται η βιβλιοθήκη Servo.h, η οποία επιτρέπει την οδήγηση έως και 12 servo ταυτόχρονα στο Arduino Uno. Ακόμα, επιτρέπει την σύνδεση των servo σε οποιαδήποτε pins και όχι μόνο σε PWM pins, καθώς το σήμα PWM γίνεται με software. Οι πιο βασικές εντολές για τον χειρισμό των servo είναι:

- **servo.attach():** Συνδέει το αντικείμενο servo με την ψηφιακή θύρα
- **servo.write():** Μετακινεί την κεφαλή του servo στην γωνία που δίνεται ως παράμετρος
- **servo.detach():** Αποσυνδέει το αντικείμενο servo από την θύρα με την οποία είχε γίνει attach()
- **servo.writeMicroseconds():** Παρέχει παλμό PWM στο servo διάρκειας όσο και η παράμετρος
- **servo.read():** Επιστρέφει την γωνία θέσης του servo που πήρε με την προηγούμενα εκτελεσμένη μέθοδο write();
- **servo.attached():** Επιστρέφει true αν το αντικείμενο τύπου Servo είναι συνδεδεμένο με ψηφιακή θύρα [7]

## 2.4 Ρομπότ με πόδια

Τα τετράποδα ρομπότ έχουν σκοπό να μιμηθούν την κίνηση και τη λειτουργία τετράποδων πλασμάτων στη φύση, όπως τα σκυλιά ή άλλα τετράποδα ζώα. Πολλές εταιρίες έχουν δημιουργήσει τετράποδα ρομπότ, καθώς έχουν ένα ευρύ φάσμα εφαρμογών, όπως επιχειρήσεις έρευνας και διάσωσης, έρευνα και ανάπτυξη στη ρομποτική, εκπαίδευση, βιομηχανικός αυτοματισμός κ.α. Κάποια παραδείγματα τέτοιων ρομπότ είναι:

1. **Spot από την Boston Dynamics:** Ο Spot της Boston Dynamics είναι ένα γνωστό τετράποδο ρομπότ. Προορίζεται για εφαρμογές, όπως η επιθεώρηση, η συλλογή δεδομένων και η απομακρυσμένη λειτουργία. Το Spot είναι εξοπλισμένο με κάμερες και αισθητήρες που το βοηθούν να κινείται και να αλληλεπιδρά με το περιβάλλον του. [11]
2. **ANYmal από την ANYbotics:** Το ANYmal είναι ένα τετράποδο ρομπότ πολλαπλών χρήσεων που αναπτύχθηκε για επιθεώρηση, συλλογή πληροφοριών, παρακολούθηση και παρέμβαση σε επικίνδυνες περιοχές. Μπορεί να λειτουργήσει σε δύσκολα εδάφη, να ανέβει σκάλες και να μεταφέρει διάφορα οφέλιμα φορτία. [12]
3. **Cheetah 3 & Mini Cheetah από το MIT:** Το Cheetah 3 είναι ένα τετράποδο ρομπότ που μπορεί να κινείται σαν κανονικό τσιτάχ. Η ευελιξία του και η ικανότητά του να κινείται με ευκολία σε ανώμαλα εδάφη το καθιστούν ιδανικό για εργασίες όπως η έρευνα και διάσωση και η βοήθεια σε περιπτώσεις καταστροφών. Το Mini Cheetah χρησιμεύει ως ερευνητική πλατφόρμα για τη ρομποτική, δίνοντας έμφαση στην ευελιξία, την ευκινησία και τις εκπαιδευτικές εφαρμογές. [2] [13]

Καθώς η τεχνολογία της 3D εκτύπωσης είναι πλέον ευρέως προσβάσιμη, πολλοί ερασιτέχνες και χομπίστες μπορούν να σχεδιάσουν και να κατασκευάσουν τα δικά τους τετράποδα ρομπότ για εκπαιδευτικούς σκοπούς, προσωπική περιέργεια ή συγκεκριμένες χρήσεις. Οι 3D εκτυπωτές τους δίνουν την δυνατότητα να δημιουργήσουν τα επιθυμητά εξαρτήματα για τα ρομπότ τους και σε συνδυασμό με πλακέτες όπως Arduino ή Raspberry να κατασκευάσουν ολοκληρωμένα ρομπότ. Με αφορμή γνωστά ρομπότ, όπως αυτά που αναφέρθηκαν παραπάνω, και βασιζόμενοι σε αυτά, αρκετοί άνθρωποι έχουν κατασκευάσει τα δικά τους ρομπότ σε πιο απλές εκδοχές

και τα μοιράζονται στο διαδίκτυο, παρέχοντας όλες τις απαραίτητες πληροφορίες για την κατασκευή τους. Έτσι, πολλοί άνθρωποι μπορούν να φτιάξουν μόνοι τους αυτά τα ρομπότ και να τα εξελίξουν ακόμα περισσότερο, ενώ έχουν την ευκαιρία να εξερευνήσουν τη ρομποτική και να αποκτήσουν πρακτική εμπειρία.



Εικόνα 11: Ένα 3D printed robot βασισμένο στο Spot και το Mini Cheetah, πηγή: [14]



Εικόνα 12: Το ρομπότ Spot από την Boston Dynamics, πηγή: [11]

Η ιστοσελίδα instructables είναι μια πλατφόρμα που μπορεί κανείς να ανεβάσει δικά του project διαφόρων κατηγοριών, ανάμεσα στις οποίες είναι και η ρομποτική. Το GorillaBot κατασκευάστηκε από έναν χρήστη αυτής της πλατφόρμας και είναι ένα ρομπότ με 4 πόδια και 8 σερβομηχανισμούς (2 για το κάθε πόδι) το οποίο χρησιμοποιεί ένα Arduino Nano, ενώ μπορεί να ελέγχεται από ένα τηλεχειριστήριο. Το ρομπότ αυτό μαζί με το τηλεχειριστήριο μπορεί να εκτυπωθεί εξ' ολοκλήρου με έναν 3D printer, καθώς τα STL αρχεία που χρειάζονται διατίθενται δωρεάν για λήψη, όπως επίσης και οδηγίες για την συναρμολόγηση του. Το ρομπότ έχει σχεδιαστεί για αγώνα ταχύτητας, με στόχο να καταρρίψει το ρεκόρ αυτόνομου σπριντ 10 μέτρων για τετράποδα ρομπότ, ενώ μπορεί και να ελέγχεται απομακρυσμένα μέσω τηλεχειριστηρίου. Για να εξασφαλιστεί η αυτονομία κατά τη διάρκεια του αγώνα, το ρομπότ είναι εξοπλισμένο με αισθητήρες, όπως ένα μαγνητόμετρο QMC5883L (ψηφιακή πυξίδα) για τον προσανατολισμό, έναν αισθητήρα υπερήχων HC-SR04 για τον εντοπισμό εμποδίων και την αποφυγή συγκρούσεων και έναν μετρητή βημάτων στον κώδικα για τον έλεγχο του αριθμού των βημάτων που κάνει το ρομπότ για την απόσταση των 10 μέτρων. [15] Ο σκελετός του συγκεκριμένου ρομπότ μπορεί εύκολα να χρησιμοποιηθεί και για άλλους σκοπούς, όπως ένα αυτόνομο κινούμενο ρομπότ που αποφεύγει εμπόδια, καθώς έχει θέση για τον αισθητήρα υπερήχων και μπορεί να εκτυπωθεί και να συναρμολογηθεί με σχετική ευκολία. Βάση αυτού του ρομπότ κατασκευάστηκε και το βαδίζον ρομπότ της συγκεκριμένης διπλωματικής εργασίας.

## **Κεφάλαιο 3: Μεθοδολογία**

### **3.1 Βήματα υλοποίησης του βαδίζοντος ρομπότ**

#### **1. Έρευνα για έτοιμα βαδίζοντα ρομπότ στο Internet:**

Για την υλοποίηση του βαδίζοντος ρομπότ, χρειάστηκε αρχικά να γίνει κάποια έρευνα στο διαδίκτυο για έτοιμα βαδίζοντα ρομπότ. Αυτή η έρευνα αποσκοπεί στη συλλογή πληροφοριών σχετικά με διάφορα μοντέλα, λειτουργίες, δυνατότητες και τεχνολογικά χαρακτηριστικά των συγκεκριμένων ρομπότ που θα μπορούσαν να χρησιμεύσουν ως σημαντικές πηγές αναφοράς για την επίτευξη του έργου. Με την αναζήτηση σε διαδικτυακά αποθετήρια, φόρουμ, ερευνητικές εργασίες και δημοσιεύσεις μπορεί κανείς να εντοπίσει πιθανές λύσεις σε προβλήματα που ενδέχεται να προκύψουν μελλοντικά. Συνεπώς, η έρευνα αυτή παρείχε τις απαραίτητες βασικές γνώσεις και μια ολοκληρωμένη εικόνα σχετικά με την ανάπτυξη ρομπότ με πόδια, διευκολύνοντας την διαδικασία υλοποίησης.

#### **2. Προτάσεις και επιβεβαίωση από τον επιβλέποντα:**

Σε επόμενη φάση, πραγματοποιήθηκαν κάποιες συζητήσεις με τον επιβλέποντα καθηγητή σχετικά με τις προδιαγραφές και τα χαρακτηριστικά του βαδίζοντος ρομπότ. Μια από τις σημαντικότερες αποφάσεις ήταν η επιλογή ενός κατάλληλου σκελετού για το ρομπότ. Εξετάστηκαν διάφορες προτάσεις και επιλογές με βάση την αντοχή των υλικών, το κόστος, το βάρος, την ευκολία κατασκευής και συναρμολόγησης και την δυνατότητα προσαρμογής στις προδιαγραφές που τέθηκαν, όπως την ενσωμάτωση αισθητήρων. Ο επιβλέπων καθηγητής παρείχε πολύτιμη καθοδήγηση, για να διασφαλίσει ότι ο επιλεγμένος σκελετός θα ανταποκρινόταν στους προβλεπόμενους στόχους του βαδίζοντος ρομπότ.

#### **3. Κατέβασμα των STL αρχείων:**

Έπειτα, μετά από την απόφαση και την επιβεβαίωση από τον καθηγητή για την επιλογή του συγκεκριμένου σκελετού για το ρομπότ, ήρθε η ώρα για το κατέβασμα των STL αρχείων. Τα αρχεία αυτά παρέχονται από τον δημιουργό του ρομπότ δωρεάν για λήψη και ο χρόνος λήψης τους διαρκεί μερικά λεπτά. Συνολικά είναι 24 αρχεία, με 14 από αυτά να είναι το κάθε κομμάτι μεμονωμένο, ενώ τα υπόλοιπα είναι τα κομμάτια του ρομπότ σε μικρά γκρουπάκια ώστε να μπορούν να εκτυπωθούν πολλά κομμάτια ταυτόχρονα.

#### **4. Εύρεση και εκμάθηση της λειτουργίας του 3D-Printer:**

Στην συνέχεια, εφόσον δεν υπήρχε προσωπικός 3D Εκτυπωτής, έπρεπε να λυθεί το πρόβλημα της εύρεσης και εκμάθησης της λειτουργίας ενός 3D Εκτυπωτή. Με τον 3D Printer Wanhao Duplicator 6 που παρείχε το τμήμα Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών του ΔΠΠΑΕ και με την βοήθεια του επιβλέποντα καθηγητή έγινε δυνατή η εξοικείωση με τον 3D Εκτυπωτή και η εκμάθηση της χρήσης του. Ακόμα, ήταν απαραίτητη η εγκατάσταση των προγραμμάτων τεμαχισμού, τα οποία μετατρέπουν τα τρισδιάστατα μοντέλα σε εκτυπώσιμα στρώματα. Μέσω αυτής της διαδικασίας εκπαίδευσης, αποκτήθηκαν πολύτιμες γνώσεις σχετικά με την χρήση τέτοιων προγραμμάτων, όπως το Cura και το Simplify3D, και των παραμέτρων εκτύπωσης που μπορούν να ελεγχθούν μέσα από αυτά.

#### **5. Εκτύπωση των κομματιών:**

Εφόσον πλέον υπήρχαν τα STL αρχεία και οι απαιτούμενες γνώσεις για την χρήση του 3D Εκτυπωτή, έφτασε η στιγμή της εκτύπωσης των κομματιών. Κάνοντας δοκιμές, μέσω του προγράμματος Cura, στις παραμέτρους εκτύπωσης των κομματιών, όπως την θερμοκρασία της κεφαλής και του bed, το πάχος των στρωμάτων, το ποσοστό του infill και την χρήση raft, καταλήξαμε στην δημιουργία των τελικών αρχείων G-Code τα οποία μπορούν να περαστούν μέσω κάρτας SD στον 3D Εκτυπωτή. Εφόσον έχει γίνει και η τοποθέτηση του υλικού PLA στον εκτυπωτή, ξεκινάει η διαδικασία της εκτύπωσης επιλέγοντας μέσω της οθόνη του εκτυπωτή το επιθυμητό αρχείο. Η διαδικασία τελειώνει με έλεγχο του αντικειμένου που εκτυπώθηκε και μεταγενέστερη επεξεργασία, όπως αφαίρεση στηριγμάτων, την εξομάλυνση τραχειών επιφανειών ή την αφαίρεση «κλωστών» που μπορεί να έχουν δημιουργηθεί κατά την εκτύπωση.

#### **6. Συναρμολόγηση των κομματιών:**

Σε επόμενο βήμα, έρχεται η συναρμολόγηση των κομματιών που εκτυπώθηκαν. Ακολουθώντας τις οδηγίες συναρμολογούμε πρώτα το σώμα του ρομπότ το οποίο αποτελείται από 3 κομμάτια, την μπροστά βάση που έχει και θέση για αισθητήρα υπερήχων, την μεσαία βάση όπου τοποθετούνται τα ηλεκτρονικά στοιχεία και την πίσω βάση. Οι 2 ακριανές βάσεις συνδέονται στην μεσαία και ασφαρίζονται στην θέση τους με 2 τετράγωνα κλιπ η κάθε μια. Τα πόδια αποτελούνται από 24 ξεχωριστά κομμάτια το καθένα και από την μέσα πλευρά ενώνονται με την βάση του ρομπότ, ενώ εξωτερικά με τα servo. Ωστόσο, για την σωστή σύνδεση τους απαιτείται πρώτα βαθμονόμηση των

σερβομηχανισμών, ώστε να μετακινηθούν στην αρχική τους θέση και στην συνέχεια να βιδωθούν τα πόδια στα servo. Η μετακίνηση των servo στην αρχική τους θέση γίνεται μέσω κώδικα του Arduino.

#### **7. Προσθήκη ενεργών στοιχείων (Arduino, servos, κλπ.):**

Αφού έχουμε συναρμολογήσει το σώμα του ρομπότ, μπορούμε πλέον να προσθέσουμε και τα ενεργά στοιχεία του ρομπότ. Κάθε μια από τις 2 ακριανές βάσεις έχει δυο θέσεις σερβομηχανισμών σε κάθε πλευρά, ενώ η μπροστά έχει και θέση για αισθητήρα υπερήχων. Βιδώνουμε, λοιπόν, τα servo σε κάθε βάση έχοντας συνολικά 8 servo, δηλαδή 2 για κάθε πόδι, και τοποθετούμε και τον αισθητήρα υπερήχων HC-SR04 στη θέση του. Στη συνέχεια, στην πάνω μεριά της μεσαίας βάσης βιδώνουμε με 2 μικρές βίδες το Arduino Uno, επάνω στο οποίο είναι συνδεδεμένο το Arduino Sensor Shield V5.0, ενώ στην από κάτω βιδώνουμε την βάση των 2 μπαταριών. Τέλος, συνδέουμε τις μπαταρίες με ένα DC-DC Adjustable Step Down Module, το οποίο ρυθμίζουμε να δίνει ως έξοδο 6V και έπειτα το συνδέουμε με το Sensor Shield για να τροφοδοτήσουμε το Arduino, τα servo και τον αισθητήρα υπερήχων που είναι συνδεδεμένα σε αυτό.

#### **8. Προτεινόμενη αλληλουχία κινήσεων για επίτευξη βαδίσματος:**

Έχοντας ολοκληρώσει την κατασκευή του ρομπότ, μπορούμε πλέον να περάσουμε στον προγραμματισμό του Arduino για την επίτευξη του βαδίσματος. Ο δημιουργός του σκελετού του ρομπότ χρησιμοποίησε ένα τηλεχειριστήριο για τον έλεγχο του ρομπότ και είχε δημιουργήσει κάποιες συναρτήσεις για τις κινήσεις του, όπως για παράδειγμα τις κινήσεις εμπρός, πίσω και στροφή δεξιά και αριστερά. Η προτεινόμενη αλληλουχία κινήσεων των ποδιών για μια κίνηση προς τα εμπρός ήταν η εξής: το ένα εμπρός και το αντίθετο πίσω πόδι θα παρέμεναν σταθερά, ενώ τα άλλα δυο αντίθετα πόδια να πραγματοποιούσαν μια κίνηση τεσσάρων θέσεων όπου το πόδι μετακινείται επάνω, μπροστά, κάτω και τέλος λίγο πίσω για να δώσει ώθησή προς τα εμπρός. Αντίστοιχες ήταν και οι υπόλοιπες κινήσεις.

#### **9. Δοκιμαστικά προγράμματα βαδίσματος/στροφής – αποτελέσματα:**

Βάση αυτών των προτεινόμενων συναρτήσεων για τις κινήσεις του ρομπότ πραγματοποιήθηκαν δοκιμαστικά προγράμματα βαδίσματος και στροφής. Το αρχικό ρομπότ είχε σχεδιαστεί για αγώνες ταχύτητας, με συνέπεια το ρομπότ να πηγαίνει πολύ γρήγορα και ορισμένες φορές να χάνει την ισορροπία του, εκτελώντας τις πρωτότυπες



συναρτήσεις κινήσεων. Συνεπώς, έπειτα από διάφορες δοκιμές και αλλαγές στις μοίρες που θα μετακινείται το κάθε servo, καταφέραμε να μειώσουμε λίγο την ταχύτητα του ρομπότ μικραίνοντας τον διασκελισμό του και έτσι βελτιώθηκε η ισορροπία του, ενώ παράλληλα προλάβαινε να στρίψει εγκαίρως αποφεύγοντας τα εμπόδια που συναντούσε.

#### **10. Συγγραφή του ολοκληρωμένου τελικού software και δοκιμές:**

Τέλος, έχοντας βελτιστοποιήσει τις συναρτήσεις των κινήσεων, προχωράμε στην συγγραφή του ολοκληρωμένου τελικού προγράμματος. Εφόσον στόχος του ρομπότ είναι να κινείται αυτόνομα χωρίς να το ελέγχεται από κάποιο τηλεχειριστήριο, έπρεπε να αλλάξει η συνάρτηση loop() ώστε το ρομπότ να εκτελεί τις απαραίτητες κινήσεις. Κάνοντας διάφορες δοκιμές ώστε να πραγματοποιείται σωστά η αποφυγή εμποδίων και οι στροφές και έχοντας υπόψη να μπορεί το ρομπότ να ανέβει ένα μικρό σκαλοπάτι, καταλήξαμε στο τελικό software. Αρχικά, μόλις τροφοδοτηθεί το ρομπότ με ρεύμα πραγματοποιεί μια ρουτίνα (Demo) όπου επιδεικνύει κάποιες κινήσεις που μπορεί να υποστηρίξει, όπως το να κάθεται στα δυο πίσω ή μπροστά πόδια και να «ξαπλώνει» και με τα 4 πόδια του λυγισμένα. Στη συνέχεια, ξεκινάει με κίνηση προς τα εμπρός ενώ παράλληλα παίρνει σε κάθε χρονική στιγμή μετρήσεις για την απόσταση, μέσω της κατάλληλης συνάρτησης που έχει δημιουργηθεί με σκοπό να εντοπίζει τα εμπόδια. Εάν σε κάποια στιγμή η απόσταση γίνει μικρότερη από 30 εκατοστά, το ρομπότ στρίβει προς τα αριστερά και ξανά λαμβάνει μια μέτρηση. Εάν και αυτή η μέτρηση είναι μικρότερη από 30 εκατοστά τότε στρίβει σχεδόν 180 μοίρες δεξιά αλλιώς συνεχίζει ευθεία. Με αυτόν τον τρόπο το ρομπότ καταφέρνει να κινείται αυτόνομα μέσα στον χώρο αποφεύγοντας τυχόν εμπόδια που μπορεί να συναντήσει.

## Κεφάλαιο 4: Κύριο Μέρος

### 4.1 Εκτύπωση μέσω 3D Printer

Εφόσον έχει γίνει η απαραίτητη έρευνα σχετικά με τα βαδίζοντα ρομπότ και έχει πραγματοποιηθεί η διαδικασία εκμάθησης της λειτουργίας του 3D Εκτυπωτή, γίνεται η επιλογή, σε συνεννόηση με τον επιβλέποντα καθηγητή, του σκελετού του ρομπότ που θα χρησιμοποιηθεί. Έχοντας κατεβάσει τα αρχεία με μορφή STL με τα κομμάτια του ρομπότ, ξεκινάμε με την εκτύπωση τους με τον 3D εκτυπωτή Wanhao Duplicator 6. Τα κομμάτια που πρέπει να εκτυπωθούν είναι τα εξής:

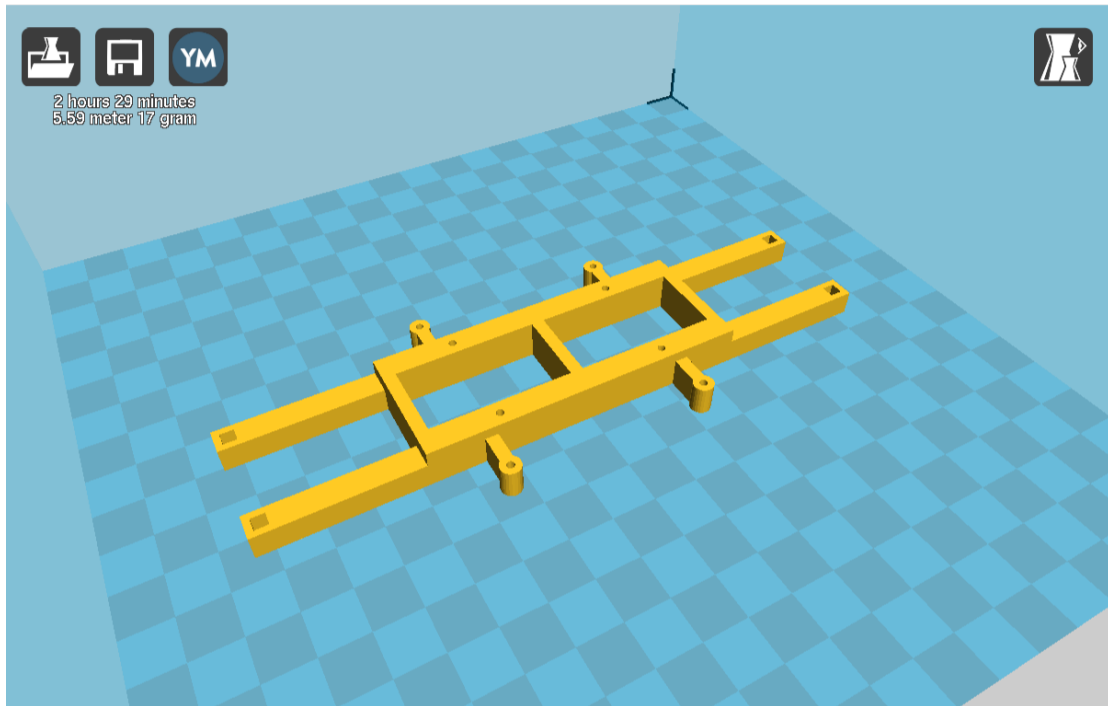
- 1x BASE ELECTRONICS (βλ. Εικόνα 13)
- 1x BASE BACK (βλ. Εικόνα 14)
- 1x BASE FRONT (βλ. Εικόνα 15)
- 8x CIRCULAR PIN L1 (βλ. Εικόνα 16)
- 4x CIRCULAR PIN L2 (βλ. Εικόνα 17)
- 4x CIRCULAR PIN L3 (βλ. Εικόνα 18)
- 4x CIRCULAR PIN L4 (βλ. Εικόνα 19)
- 8x THIGH SERVO (βλ. Εικόνα 20)
- 8x THIGH (βλ. Εικόνα 21)
- 8x CALF EXT (βλ. Εικόνα 22)
- 8x CALF INT (βλ. Εικόνα 23)
- 8x FOOT (βλ. Εικόνα 24)
- 4x SQUARE CLIP (βλ. Εικόνα 25)
- 44x CIRCULAR CLIP (βλ. Εικόνα 26)

Μέσω του προγράμματος Cura του συγκεκριμένου 3D εκτυπωτή κάνουμε τις απαραίτητες ρυθμίσεις και αλλαγές στις παραμέτρους και αποθηκεύουμε τα αρχεία σε μορφή G-CODE ώστε να τα περάσουμε μέσω SD card στον 3D εκτυπωτή. Το ελάχιστο build platform που απαιτείται για την εκτύπωση των κομματιών του ρομπότ είναι: M150mm x Π150mm x Y25mm. Το υλικό που χρησιμοποιήσαμε για την εκτύπωση ήταν PLA χρώματος λευκό και ένα καρούλι 500γρ. είναι υπεραρκετό για την εκτύπωση ενός ρομπότ. Συγκεκριμένα, σύμφωνα με το πρόγραμμα Cura, για την εκτύπωση των κομματιών χωρίς την χρήση raft χρειάζονται συνολικά 123γρ. , ενώ με την χρήση raft

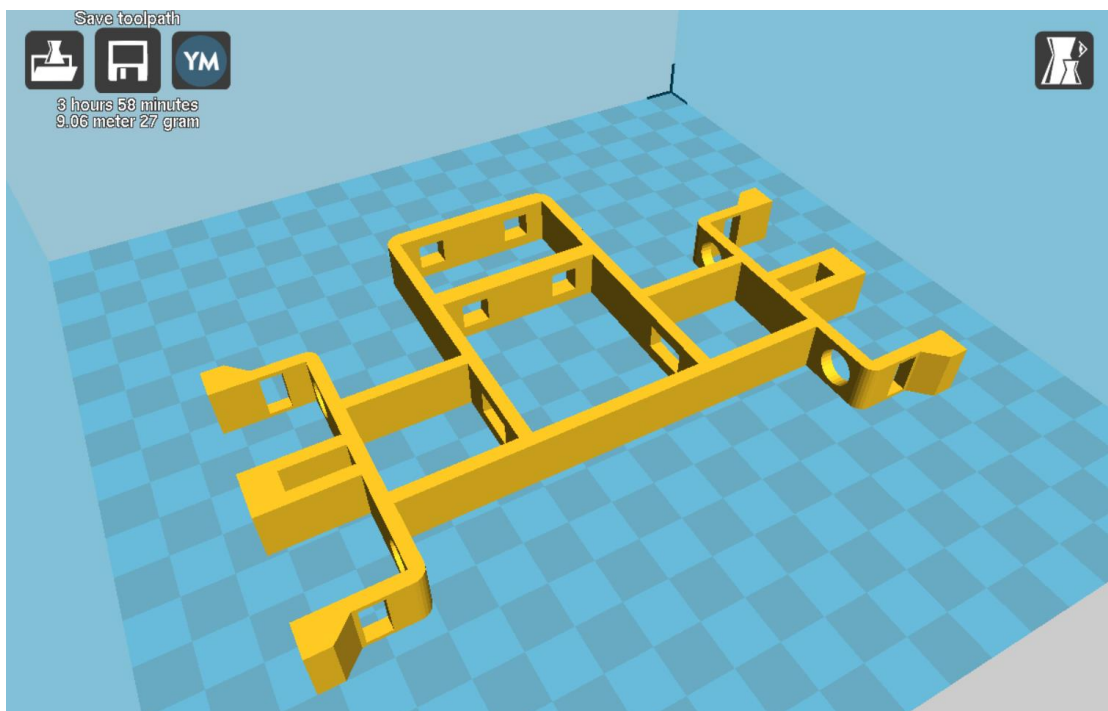
απαιτούνται 253γρ. υλικού. Όλα τα κομμάτια έχουν σχεδιαστεί έτσι ώστε να μην χρειάζονται κάποια στήριξη (support) κατά την διάρκεια της εκτύπωσης. Διάφορες δοκιμές έγιναν και στο infill των κομματιών, καθώς ένα πολύ μικρό infill δημιουργούσε πρόβλημα στις τελευταίες στρώσεις των αντικειμένων, αφού δεν υπήρχε αρκετό υλικό από κάτω για να μπορούν να εκτυπωθούν σωστά. Τελικά, ένα infill 24% έλυσε το πρόβλημα χωρίς να αυξήσει σημαντικά τον χρόνο εκτύπωσης, Ο χρόνος εκτύπωσης μπορεί να διαφέρει ανάλογα με τις παραμέτρους. Στην προκειμένη περίπτωση με τις παρακάτω παραμέτρους και χωρίς την χρήση raft ο χρόνος εκτύπωσης είναι περίπου 15 ώρες, ενώ με την χρήση raft είναι περίπου 40 ώρες. Στα πρώτα κομμάτια που εκτυπώθηκαν χρησιμοποιήθηκε το raft. Ωστόσο, επιλέχθηκε να μην χρησιμοποιηθεί στις υπόλοιπες εκτυπώσεις, επειδή σε κάποια άλλα μεγαλύτερα κομμάτια η επιφάνεια επαφής τους με το raft ήταν μεγαλύτερη και η αφαίρεση του raft ήταν εξαιρετικά δύσκολη. Η μη χρήση του raft μείωσε σημαντικά τον χρόνο εκτύπωσης και την ποσότητα υλικού που χρησιμοποιήθηκε, χωρίς να δημιουργήσει κάποιο άλλο πρόβλημα. Οι παράμετροι που χρησιμοποιήσαμε για τα κομμάτια είναι οι παρακάτω:

*Πίνακας 2: Παράμετροι εκτύπωσης στο πρόγραμμα Wanhao Cura*

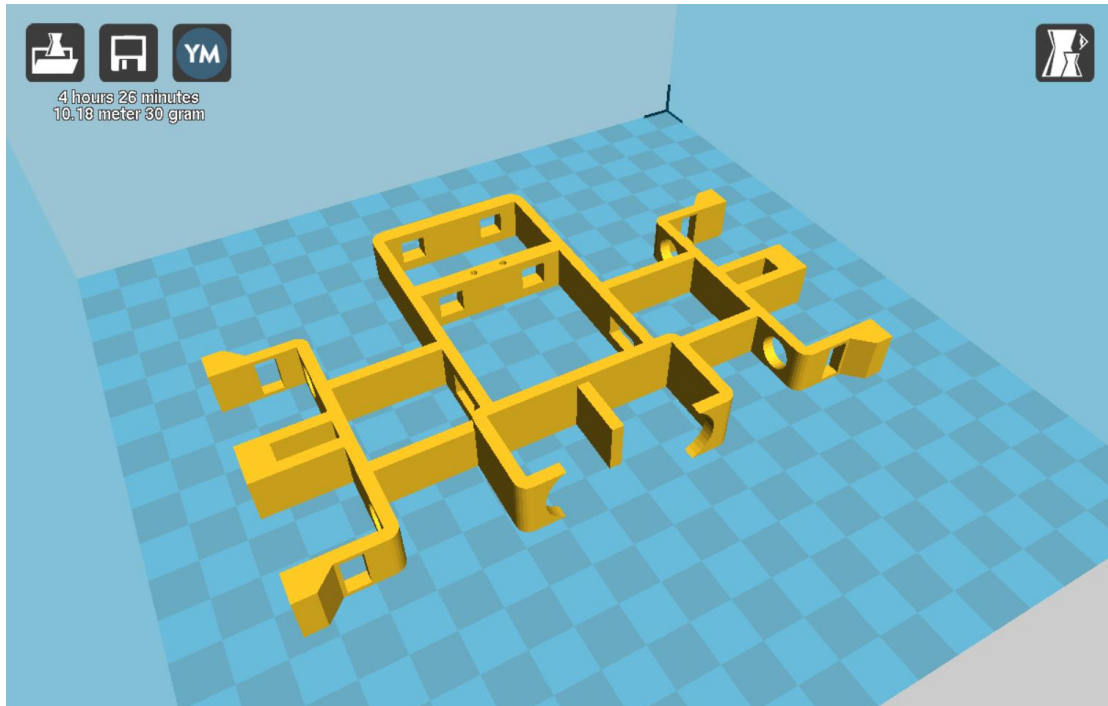
<b>Quality</b>	
Layer height (mm)	0.3
Shell thickness (mm)	0.5
<b>Fill</b>	
Bottom/Top thickness (mm)	0.6
Fill Density (%)	24
<b>Speed and Temperature</b>	
Print speed (mm/s)	50
Printing temperature (C)	235
Bed temperature (C)	60
<b>Support</b>	
Support type	None
Platform adhesion type	None
<b>Filament</b>	
Diameter (mm)	1.75
Flow (%)	100.0



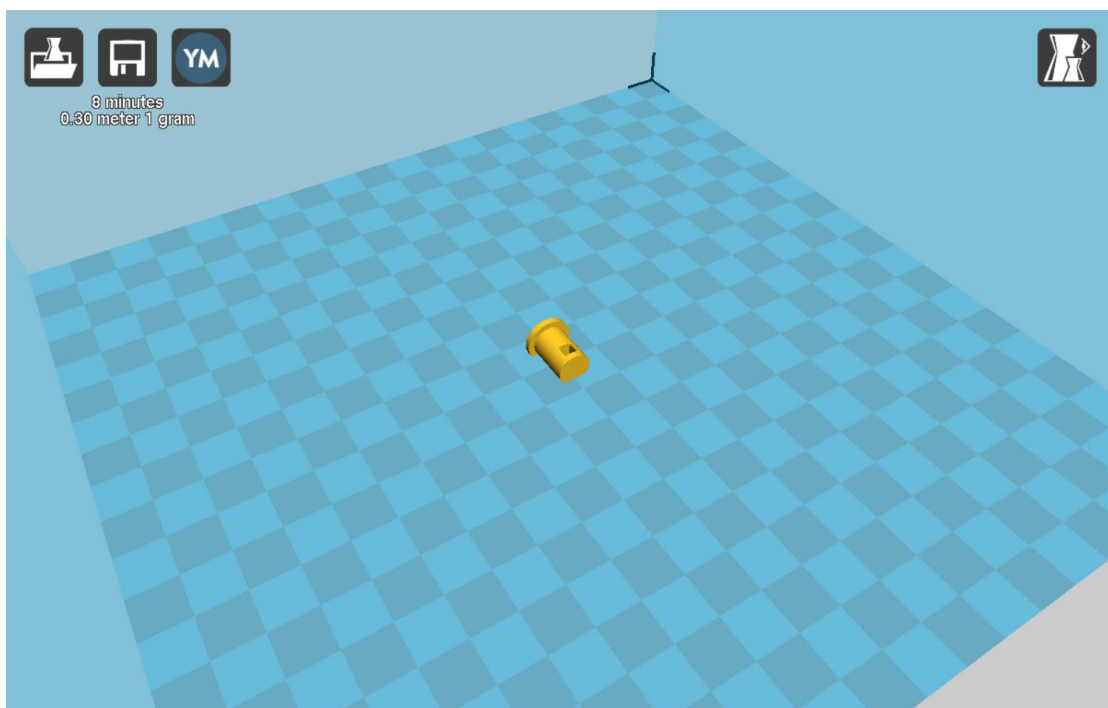
Εικόνα 13: Μεσαία βάση - Base Electronics στο πρόγραμμα Cura



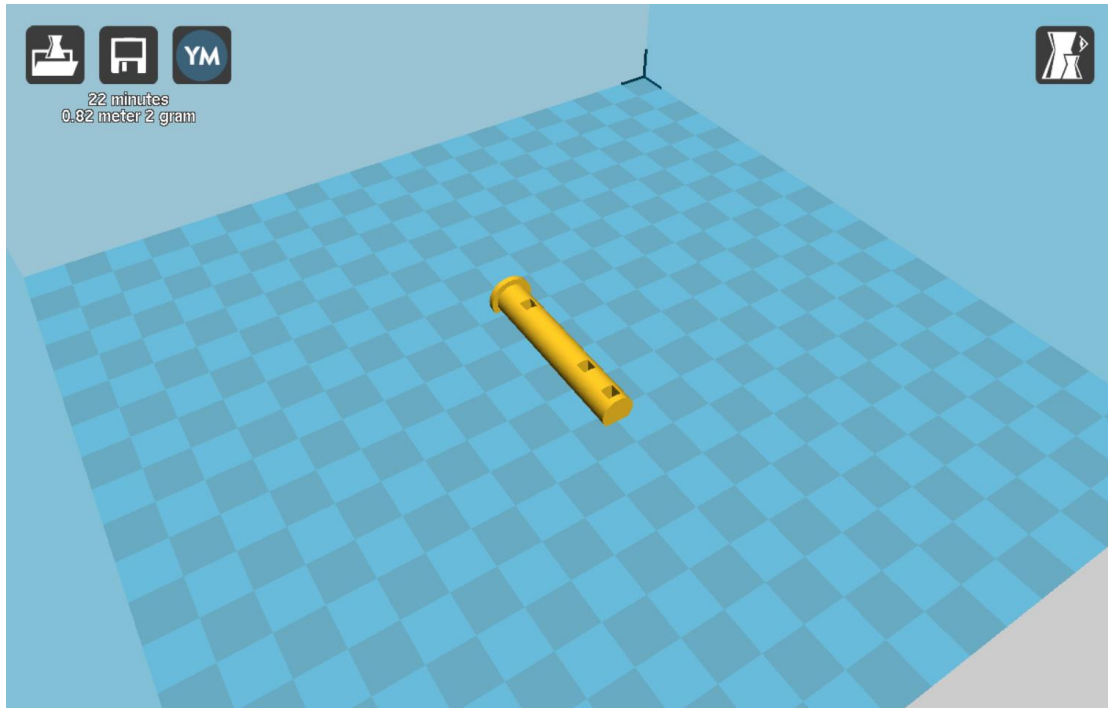
Εικόνα 14: Πίσω βάση - Base Back στο πρόγραμμα Cura



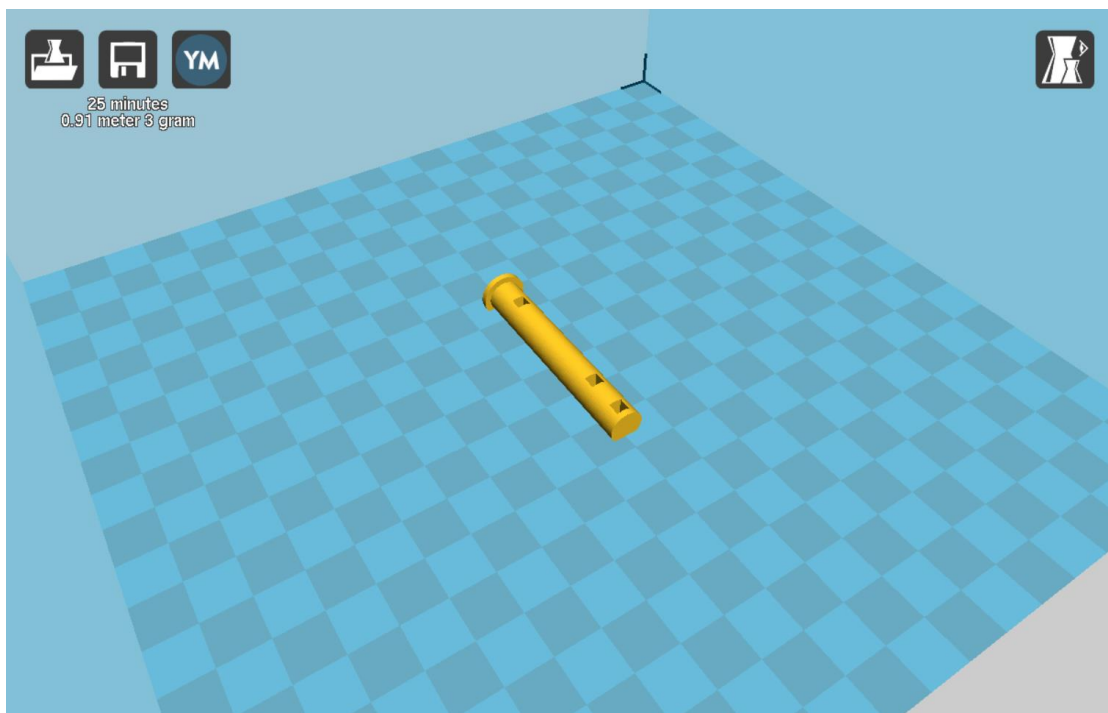
Εικόνα 15: Μπροστά βάση - *Front Base* στο πρόγραμμα *Cura*



Εικόνα 16: Κυκλικό pin L1 (*Circular Pin L1*) στο πρόγραμμα *Cura*

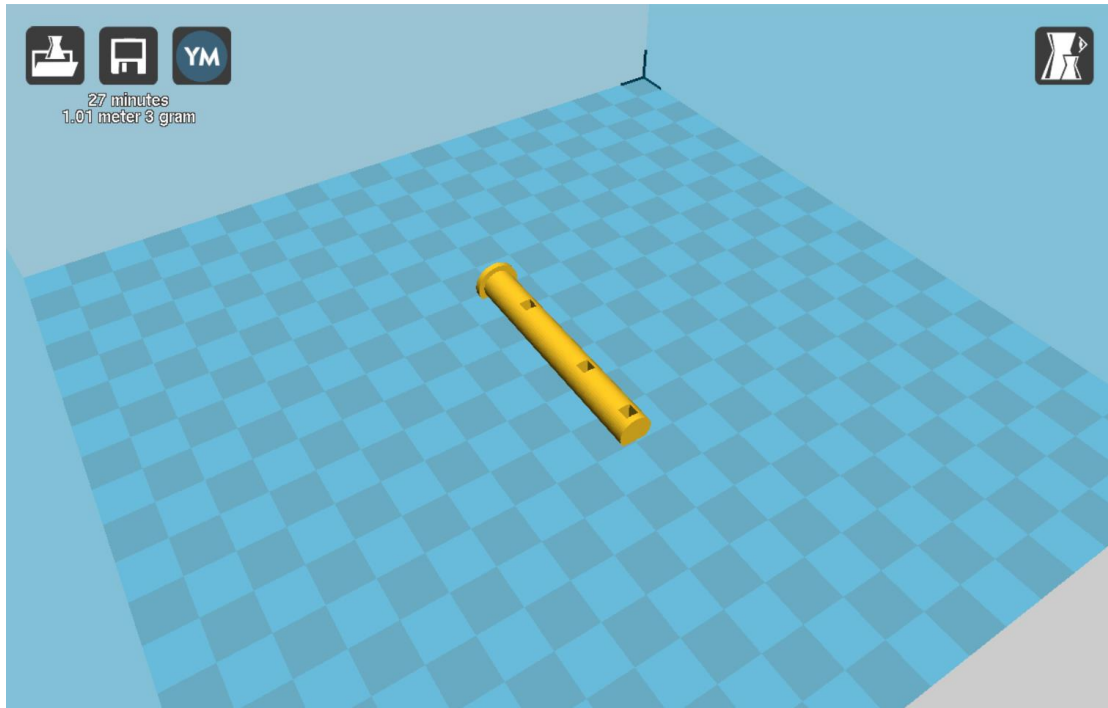


Εικόνα 17: Κυκλικό pin L2 (Circular Pin L2) στο πρόγραμμα Cura

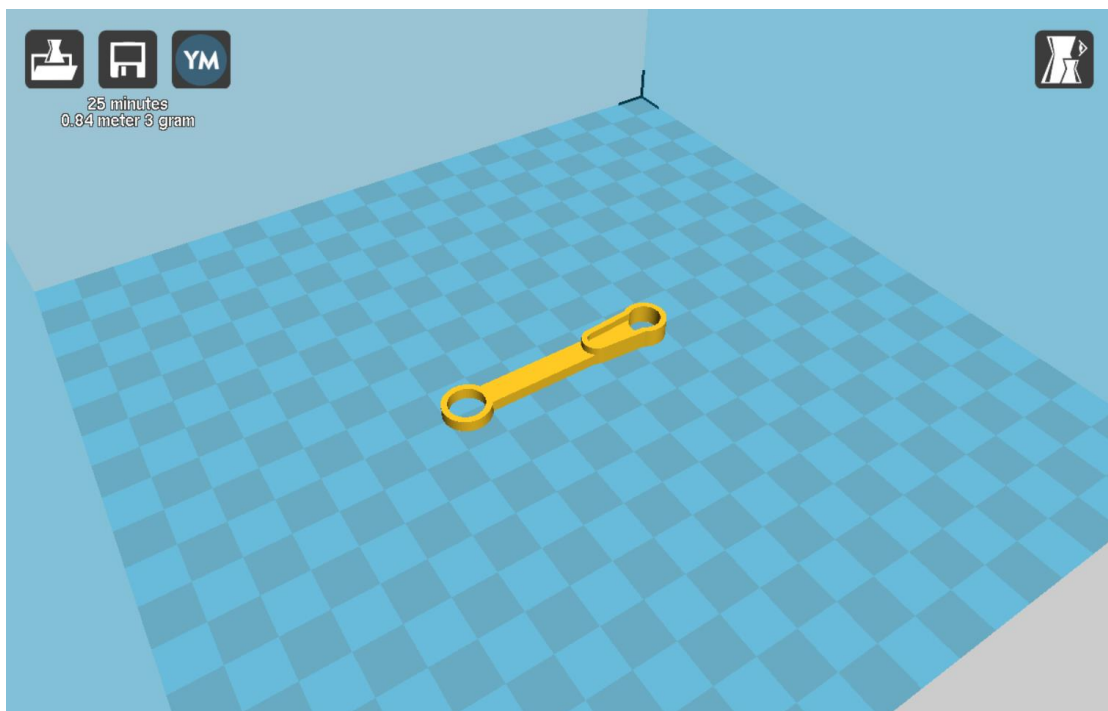


Εικόνα 18: Κυκλικό pin L3 (Circular Pin L3) στο πρόγραμμα Cura

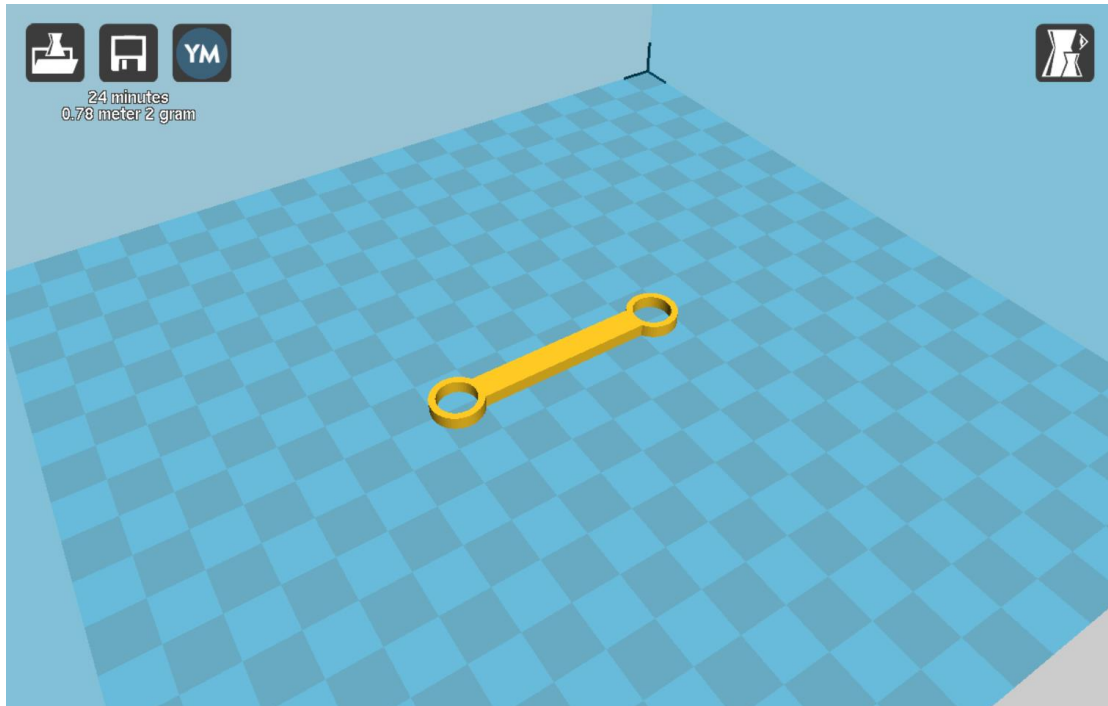




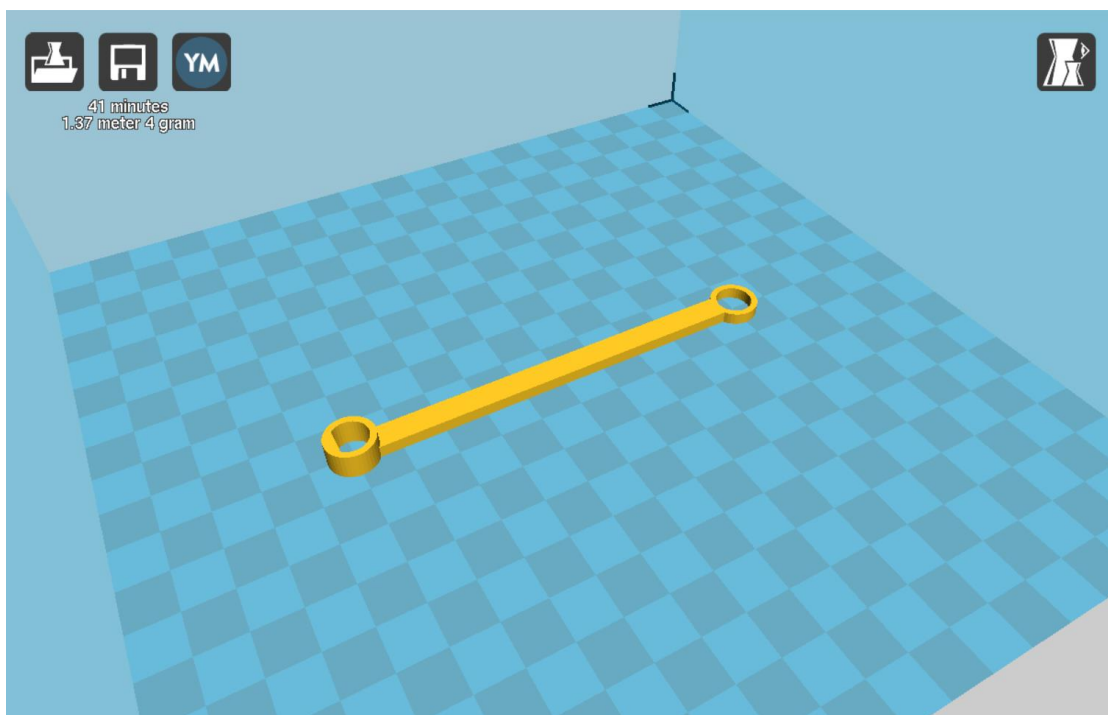
Εικόνα 19: Κυκλικό pin L4 (Circular Pin L4) στο πρόγραμμα Cura



Εικόνα 20: Μηρός σερβομηχανισμού (Thigh Servo) στο πρόγραμμα Cura

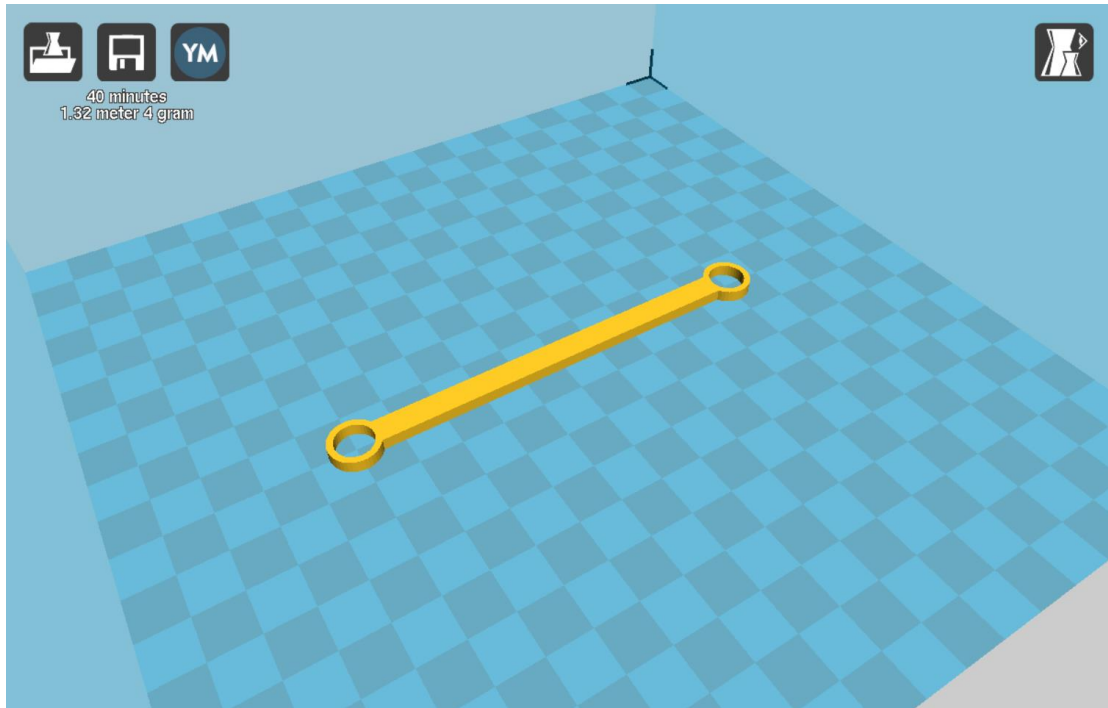


Εικόνα 21: Μηρός (Thigh) στο πρόγραμμα Cura

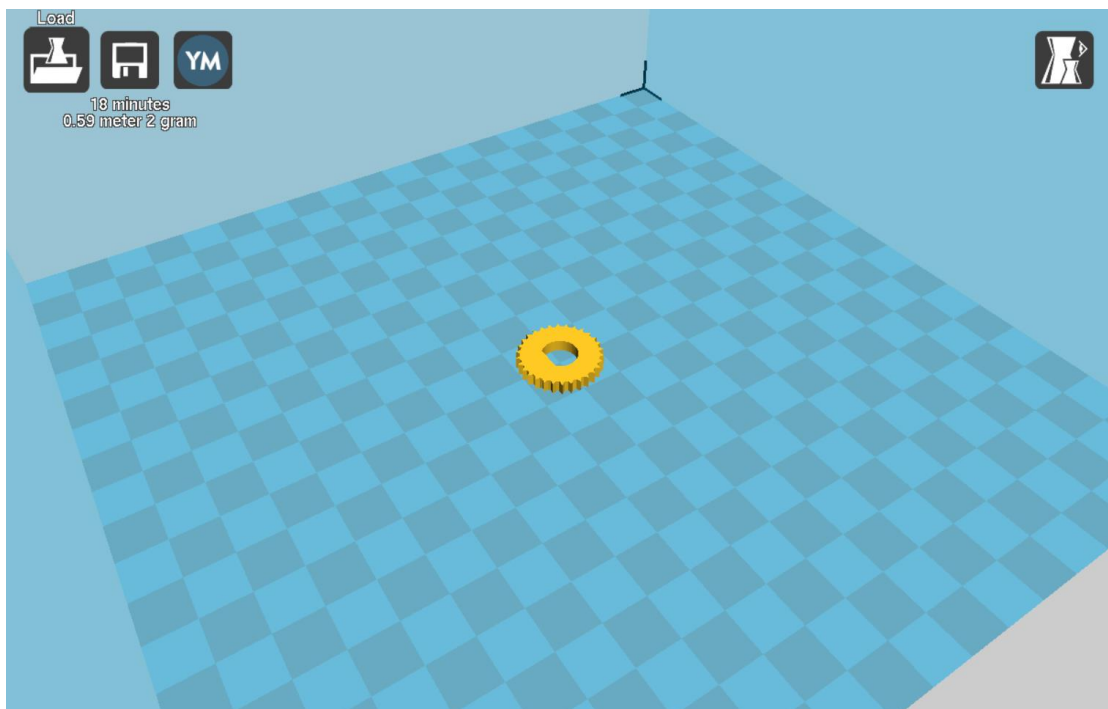


Εικόνα 22: Κνήμη εξωτερική (Calf Ext) στο πρόγραμμα Cura

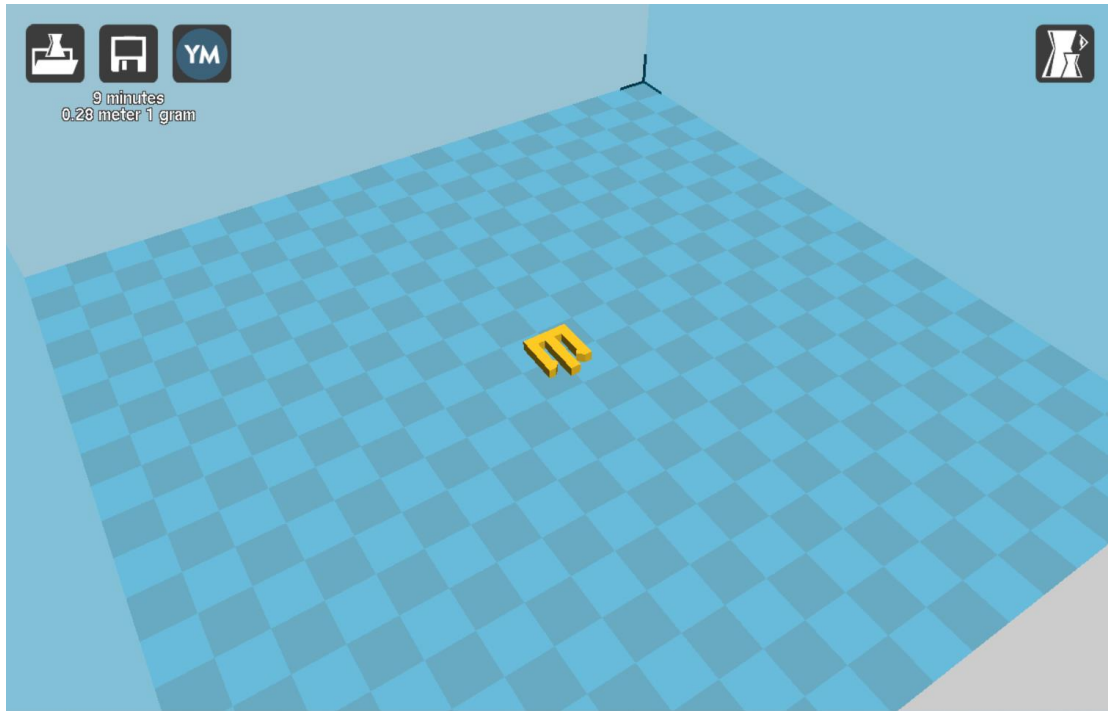




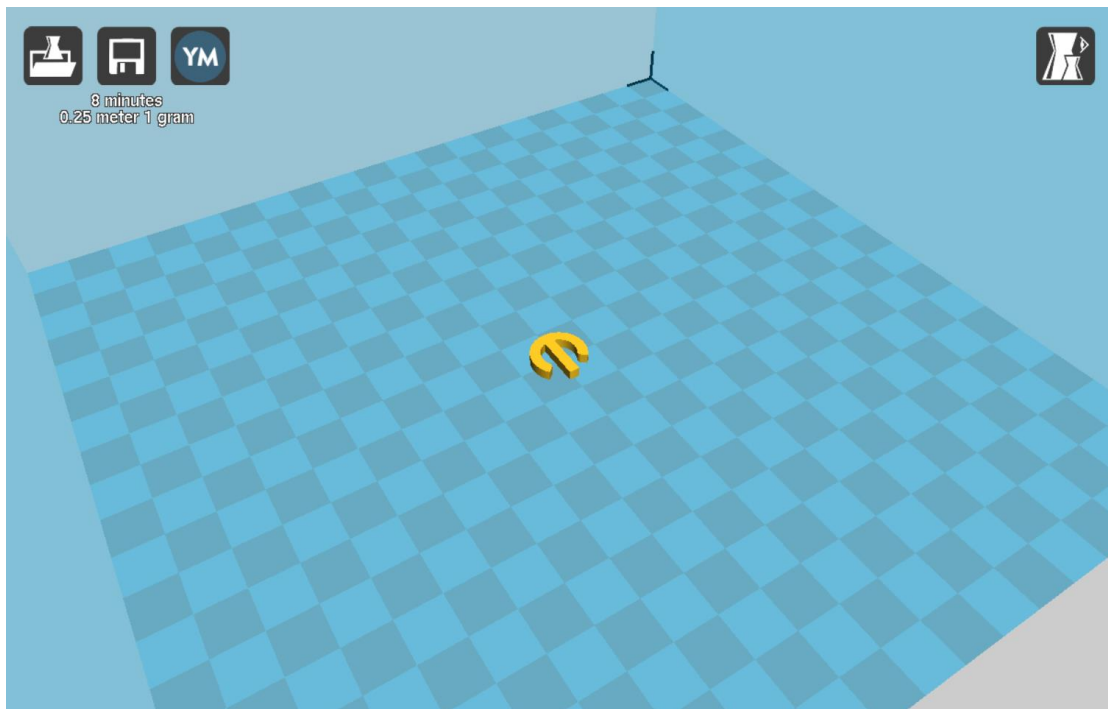
Εικόνα 23: Κνήμη εσωτερική (Calf Int) στο πρόγραμμα Cura



Εικόνα 24: Πόδι (Foot) στο πρόγραμμα Cura



Εικόνα 25: Τετράγωνο κλιπ (Square Clip) στο πρόγραμμα Cura



Εικόνα 26: Κυκλικό κλιπ (Circular Clip) στο πρόγραμμα Cura

## 4.2 Υλικά και συνδεσμολογία

Πέρα από τα κομμάτια που εκτυπώνονται από το 3D εκτυπωτή για την κατασκευή του ρομπότ θα χρειαστούν τα εξής υλικά:

- 1x Arduino Uno
- 1x Sensor Shield V5.0 για Arduino
- 8x Servo Micro analog 180 degree (Waveshare SG90)
- 1x Ultrasonic sensor HC-SR04
- 2x Επαναφορτιζόμενες Μπαταρίες 18650 Li-ion 3000 mAh 3.7V
- 1x Θήκη 2 θέσεων για μπαταρίες 18650 με καλώδια σύνδεσης και on/off διακόπτη
- 1x Φορτιστή 2 Μπαταριών Li-ion Μεγέθους 18650
- 1x DC-DC Adjustable Step Down Module με Βολτόμετρο
- 4x female to female καλώδια για το αποστασιόμετρο
- 4x Βίδες 2mm x 8mm για να βιδωθεί το Arduino και η βάση για τις μπαταρίες επάνω στην μεσαία βάση (Electronics Base) του ρομπότ

Για την συνδεσμολογία των παραπάνω ακολουθούμε την εξής διαδικασία:

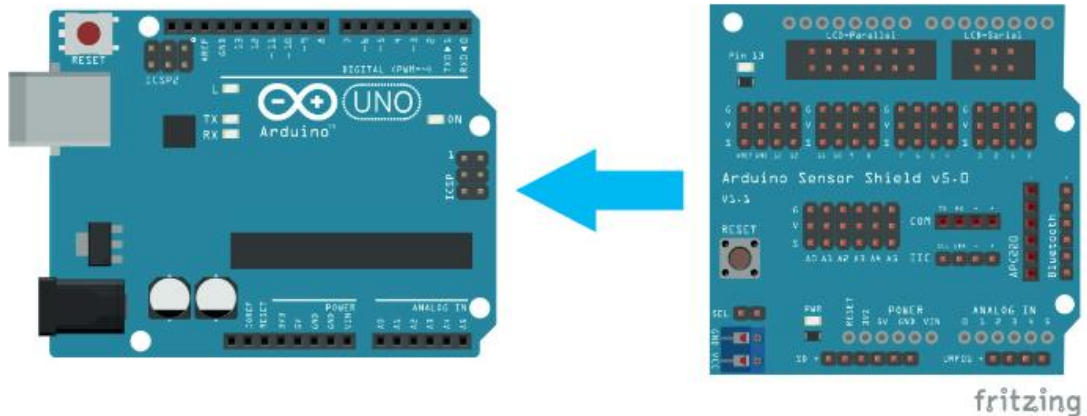
Αρχικά, «κουμπώνουμε» το Sensor Shield V5.0 επάνω στο Arduino και στη συνέχεια συνδέουμε τα servo στα ψηφιακά pin S2 έως S9 του Sensor Shield.

Συγκεκριμένα,

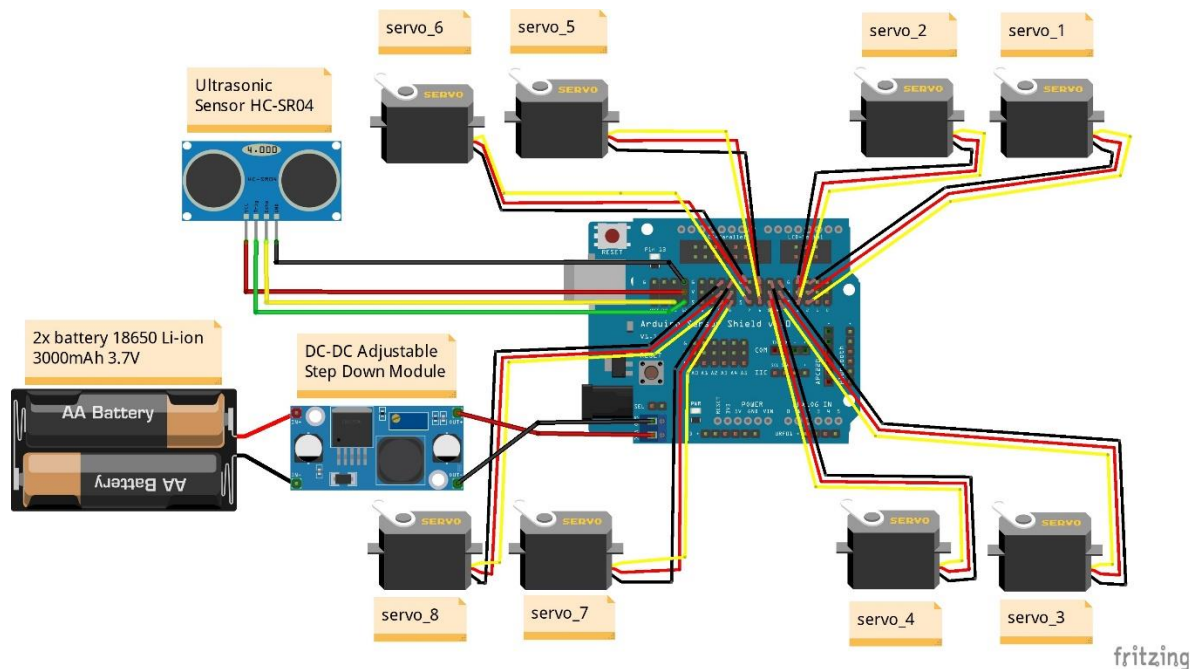
- Στο πίσω δεξί πόδι: Το πίσω servo (servo\_1) στο pin 2 και το μπροστά servo (servo\_2) στο pin 3.
- Στο πίσω αριστερό πόδι: Το πίσω servo (servo\_3) στο pin 4 και το μπροστά servo (servo\_4) στο pin 5.
- Στο μπροστά δεξί πόδι: Το πίσω servo (servo\_5) στο pin 6 και το μπροστά servo (servo\_6) στο pin 7.
- Στο μπροστά αριστερό πόδι: Το πίσω servo (servo\_7) στο pin 8 και το μπροστά servo (servo\_8) στο pin 9.

Έπειτα, συνδέουμε το αποστασιόμετρο Ultrasonic sensor HC-SR04 συνδέοντας το GRN pin με την γείωση G13 και το VCC pin με το V13 του Sensor Shield V5.0, καθώς επίσης και το Trig pin με το S12 και το Echo pin με το S13.

Ακόμα, συνδέουμε τις μπαταρίες με το DC-DC Adjustable Step Down Module το οποίο δέχεται ως είσοδο από τις μπαταρίες συνολικά 7,4V και το ρυθμίζουμε να δίνει ως έξοδο 6V. Τέλος, το συνδέουμε με την εξωτερική τροφοδοσία του Sensor Shield V5.0, το +/- με το VCC/GRN αντίστοιχα. Έτσι, τροφοδοτούμε το Arduino και τα servo παράλληλα μέσα από το Sensor Shield.



Σχήμα 4: Σχεδιάγραμμα σύνδεσης του Sensor Shield V5.0 με το Arduino Uno μέσω του προγράμματος Fritzing [16]



Σχήμα 5: Σχεδιάγραμμα συνδεσμολογίας των servo, του HC-SR04, του DC-DC Step Down Module και των μπαταριών με το Sensor Shield V5.0 μέσω του προγράμματος Fritzing [16]

### 4.3 Συναρμολόγηση του ρομπότ

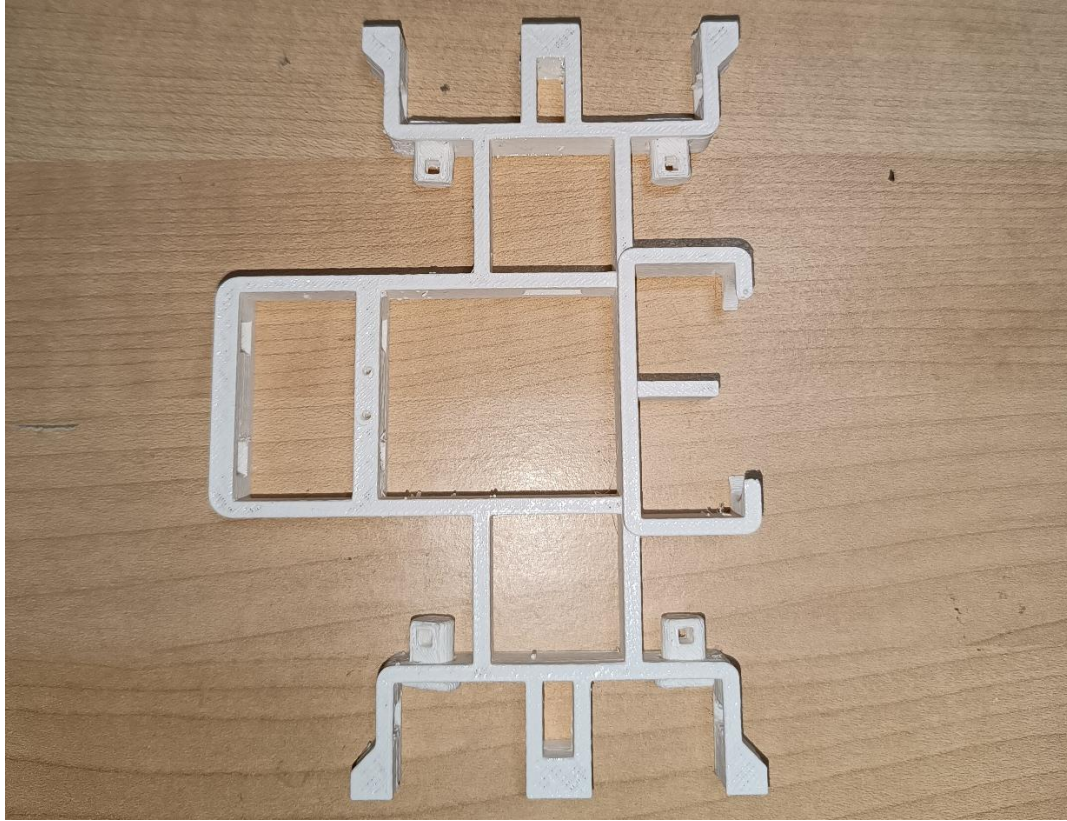
Αρχικά, ξεκινάμε την κατασκευή του ρομπότ με την συναρμολόγηση του σώματος ακολουθώντας τα εξής βήματα:

1. Τοποθετούμε ένα κυκλικό Pin L1 σε κάθε τρύπα της μπροστινής βάσης (Base Front) που βρίσκεται στις θέσεις των σερβομηχανισμών (βλ. Εικόνα 27)
2. Περνάμε τα καλώδια από τους σερβομηχανισμούς μέσα από τις υποδοχές που βρίσκονται στις θέσεις των σερβομηχανισμών
3. Τοποθετούμε τους σερβομηχανισμούς στη θέση τους
4. Στερεώνουμε τους σερβομηχανισμούς στις θέσεις τους με 2 βίδες (δεν τις σφίγγουμε πολύ γιατί μπορεί να προκληθεί ζημιά στη βάση) (βλ. Εικόνα 28)
5. Επαναλαμβάνουμε την ίδια διαδικασία για την πίσω βάση (Base Back) (βλ. Εικόνα 29)
6. Σέρνουμε την μπροστινή βάση (Base Front) στη μεσαία βάση (Base Electronics) μέσα από τις 2 τετράγωνες οπές με την θύρα USB του Arduino να βλέπει προς την μπροστά βάση (Base Front)
7. Στερεώνουμε τις δυο βάσεις στη θέση τους με 2 τετράγωνα κλιπ (βλ. Εικόνα 30)
8. Στερεώνουμε τον αισθητήρα υπερήχων στην μπροστά βάση (Base Front) (βλ. Εικόνα 31)
9. Στερεώνουμε το Arduino με το Sensor Shield στη μεσαία βάση (Base Electronics) με 2 βίδες (βλ. Εικόνα 32)
10. Σέρνουμε την πίσω βάση (Base Back) στην μεσαία βάση (Base Electronics) μέσα από τις 2 τετράγωνες οπές
11. Στερεώνουμε τις δυο βάσεις στη θέση τους με 2 τετράγωνα κλιπ
12. Στερεώνουμε το DC-DC Step Down Module επάνω στο ρομπότ με ταινία διπλής όψης (βλ. Εικόνα 33)
13. Τέλος, Στερεώνουμε την βάση της μπαταρίας στην κάτω πλευρά της μεσαίας βάσης (Base Electronics) με 2 βίδες διαγώνια (ή 4 βίδες) (βλ. **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**)

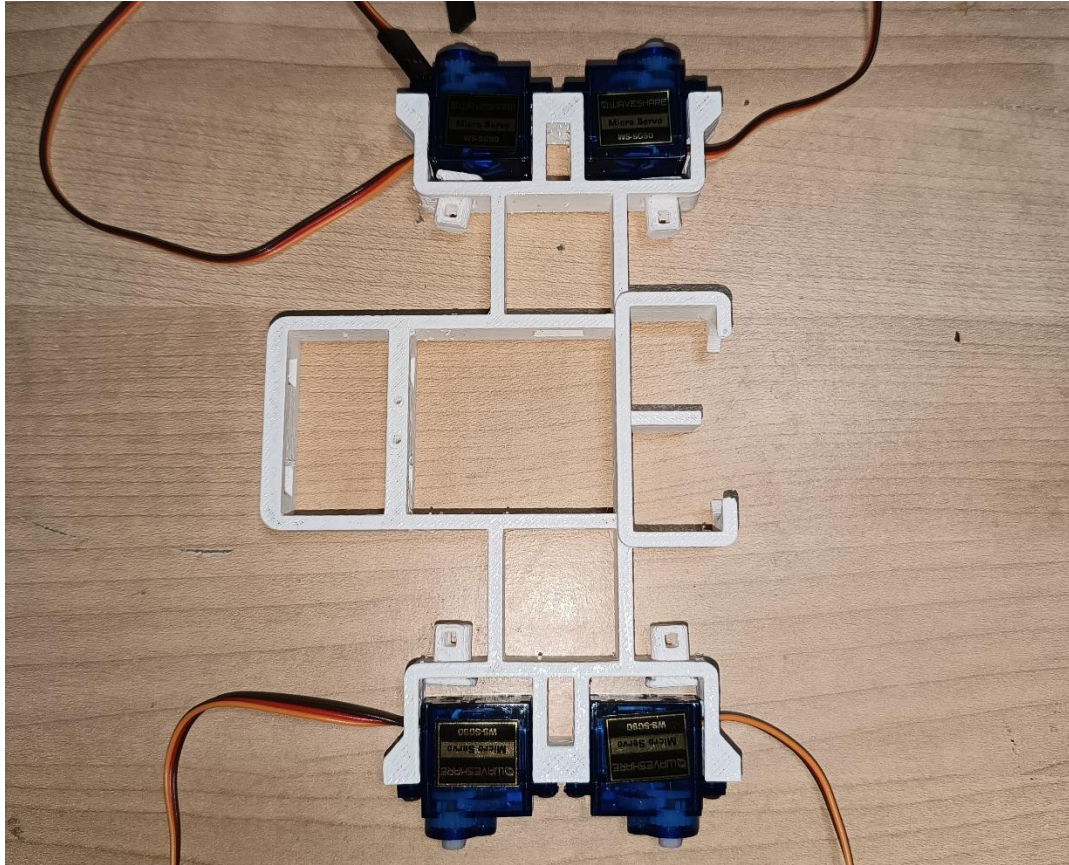
Στην συνέχεια, πρέπει να κάνουμε βαθμονόμηση (Calibration) των servo προτού συνδέσουμε τα πόδια. Ανεβάζουμε στο Arduino το πρόγραμμα, το οποίο θα αναλυθεί σε επόμενη ενότητα αυτού του κεφαλαίου, και ανοίγουμε τον διακόπτη από τις μπαταρίες για 2 δευτερόλεπτα ώστε να μετακινηθούν τα servo στην αρχική τους



θέση (home position), η οποία είναι ανάλογα είτε 0 είτε 180 μοίρες. Τότε μπορούμε να βιδώσουμε τα πόδια κάθετα με το σώμα του ρομπότ.

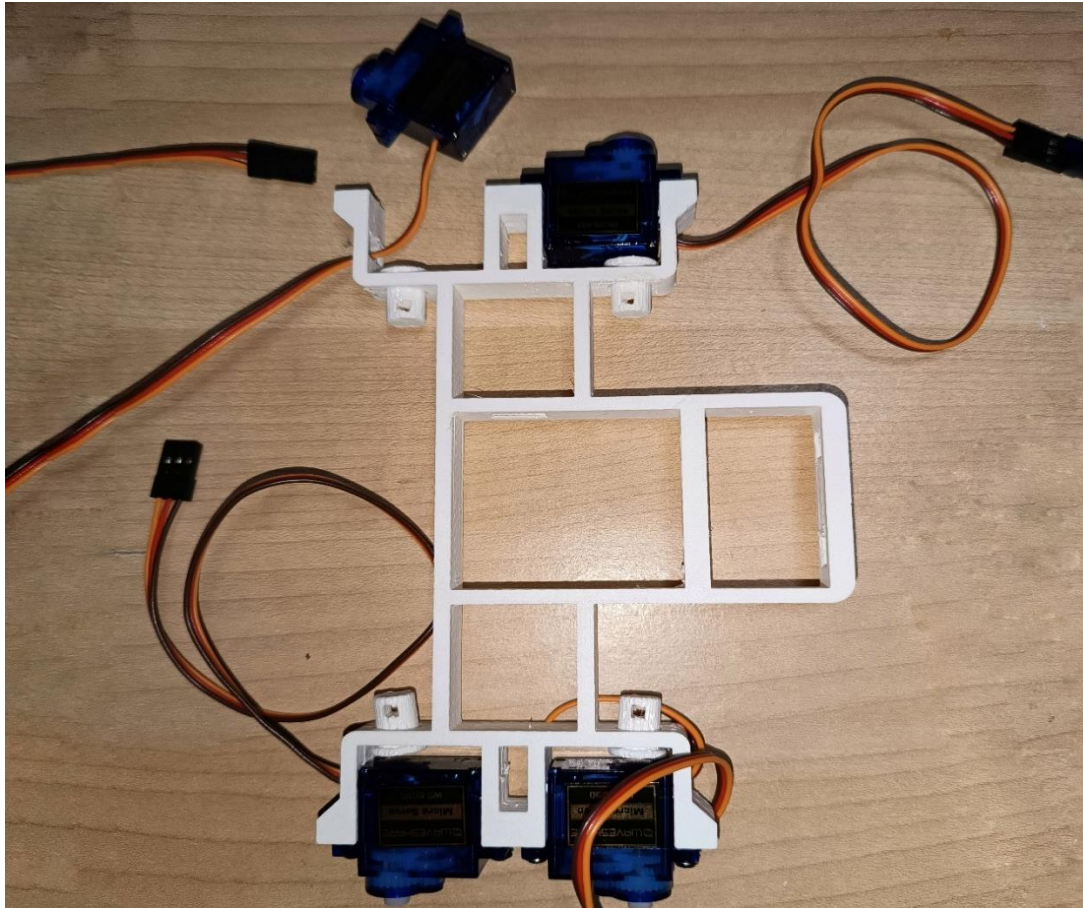


*Εικόνα 27: Συναρμολόγηση του σώματος του ρομπότ - βήμα 1*

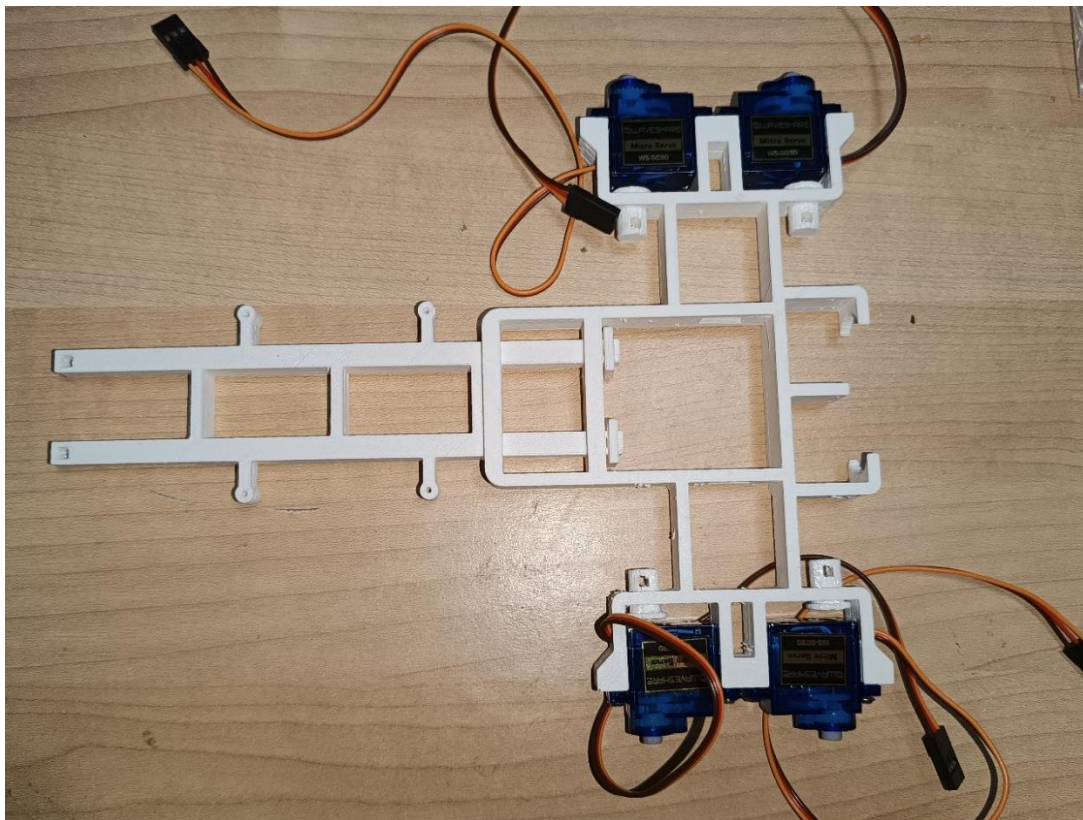


*Εικόνα 28: Συναρμολόγηση του σώματος του ρομπότ - βήμα 2 έως 4*



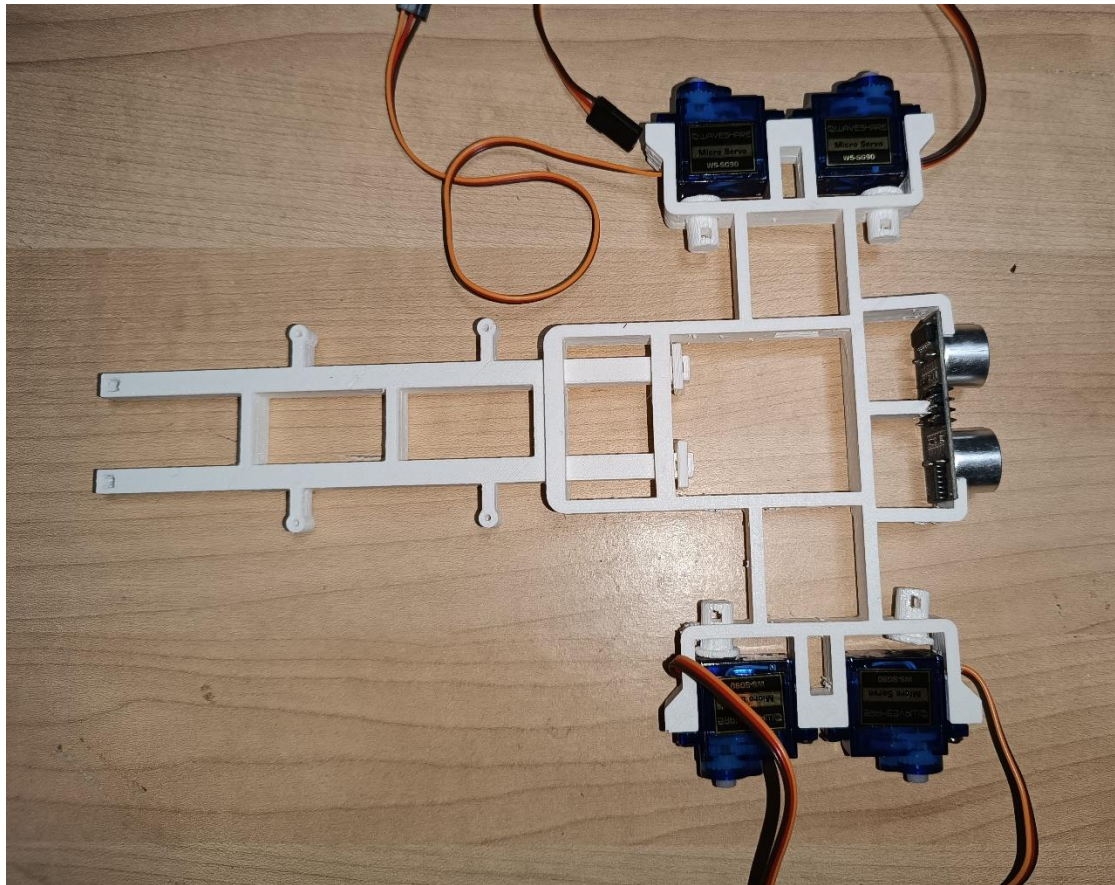


*Εικόνα 29: Συναρμολόγηση του σώματος του ρομπότ - βήμα 5*

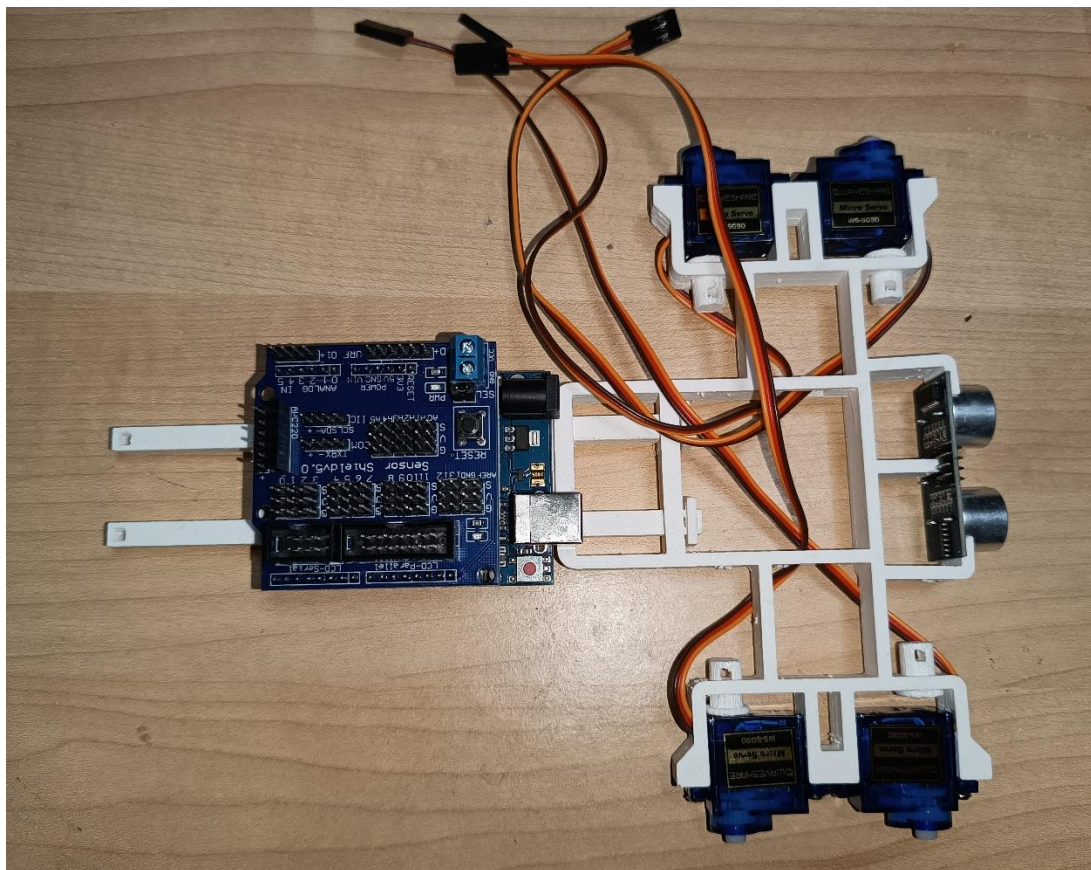


*Εικόνα 30: Συναρμολόγηση του σώματος του ρομπότ - βήμα 6 έως 7*



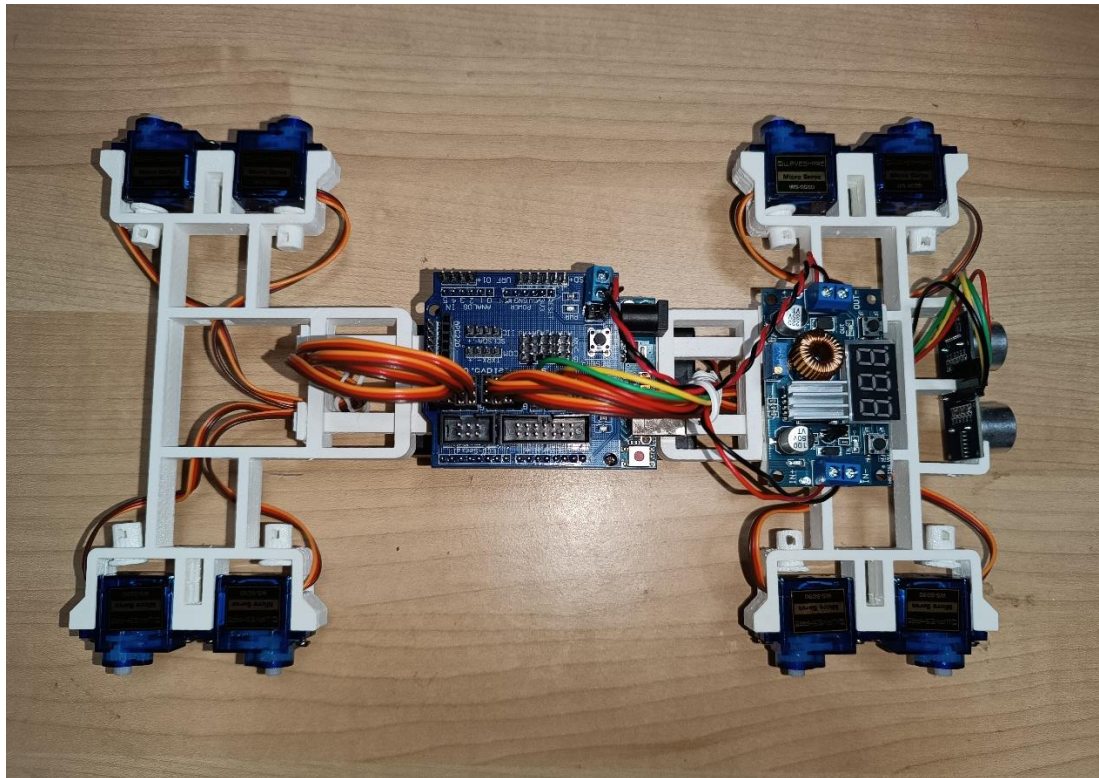


Εικόνα 31: Συναρμολόγηση του σώματος του ρομπότ - βήμα 8

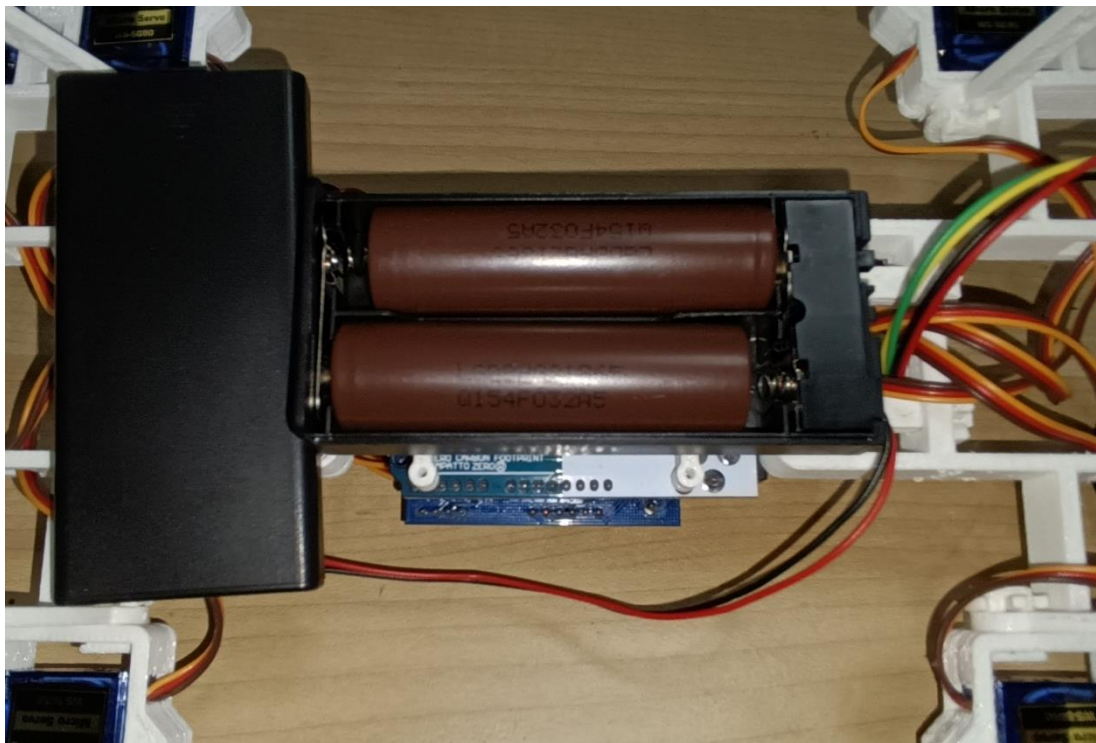


Εικόνα 32: Συναρμολόγηση του σώματος του ρομπότ - βήμα 9





Εικόνα 33: Συναρμολόγηση του σώματος του ρομπότ - βήμα 10 έως 12



Εικόνα 34: Συναρμολόγηση του σώματος του ρομπότ - βήμα 13

Για την συναρμολόγηση των ποδιών του ρομπότ ακολουθούμε τα παρακάτω βήματα:

1. Σέρνουμε 1 πόδι (Foot) μέσα από 1 κυκλικό pin L4 (Circular Pin L4) (βλ. Εικόνα 35)
2. Σέρνουμε το παχύτερο άκρο μιας εξωτερικής κνήμης (Calf Ext) μέσα από το κυκλικό pin L4 (Circular Pin L4) με την έξω πλευρά στραμμένη μακριά από το πόδι (βλ. Εικόνα 36)
3. Σέρνουμε 1 εσωτερική κνήμη (Calf Int) μέσα από το κυκλικό pin L4 (Circular Pin L4)
4. Στερεώνουμε στη θέση του με 1 κυκλικό κλιπ (Circular Clip) (βλ. Εικόνα 37)
5. Σέρνουμε 1 εσωτερική κνήμη (Calf Int) μέσα από το κυκλικό pin L4 (Circular Pin L4)
6. Σέρνουμε το παχύτερο άκρο μιας εξωτερικής κνήμης (Calf Ext) μέσα από 1 κυκλικό pin L4 (Circular Pin L4) με την έξω πλευρά στραμμένη προς το πόδι (Foot)
7. Σέρνουμε 1 πόδι (Foot) μέσα από το κυκλικό pin L4 (Circular Pin L4)
8. Στερεώνουμε στη θέση τους με 2 κυκλικά κλιπ (Circular Clip) (βλ. Εικόνα 38)
9. Σέρνουμε 1 κυκλικό pin L3 (Circular Pin L3) μέσα από το άλλο άκρο της εξωτερικής κνήμης (Calf Ext)
10. Σέρνουμε 1 μηρό servo (Thigh Servo) μέσα από 1 κυκλικό pin L3 (Circular Pin L3) με την έξω πλευρά στραμμένη προς την εξωτερική κνήμη (Calf Ext)
11. Στερεώνουμε στη θέση τους με 1 κυκλικό κλιπ (Circular Clip) (βλ. Εικόνα 39)
12. Σέρνουμε 1 μηρό (Thigh) μέσα από το κυκλικό pin L3 (Circular Pin L3)
13. Σέρνουμε το κυκλικό pin L3 (Circular Pin L3) μέσα από την άλλη εξωτερική κνήμη (Calf Ext)
14. Στερεώνουμε στη θέση του με 1 κυκλικό κλιπ (Circular Clip) (βλ. Εικόνα 40)
15. Σέρνουμε 1 μηρό servo (Thigh Servo) μέσα από 1 κυκλικό pin L2 (Circular Pin L2) με την έξω πλευρά στραμμένη προς την κεφαλή του κυκλικού pin L2 (Circular Pin L2)
16. Σέρνουμε το κυκλικό pin L2 (Circular Pin L2) μέσα από τις δύο εξωτερικές κνήμες (Calf Ext)
17. Σέρνουμε 1 μηρό (Thigh) μέσα από το κυκλικό pin L2 (Circular Pin L2)
18. Στερεώνουμε στη θέση του με 3 κυκλικά κλιπ (Circular Clip) (βλ. Εικόνα 41)
19. Επαναλαμβάνουμε όλες τις διαδικασίες για τα υπόλοιπα 3 πόδια (βλ. Εικόνα 42)





*Εικόνα 35: Συναρμολόγηση ποδιών του ρομπότ - βήμα 1*

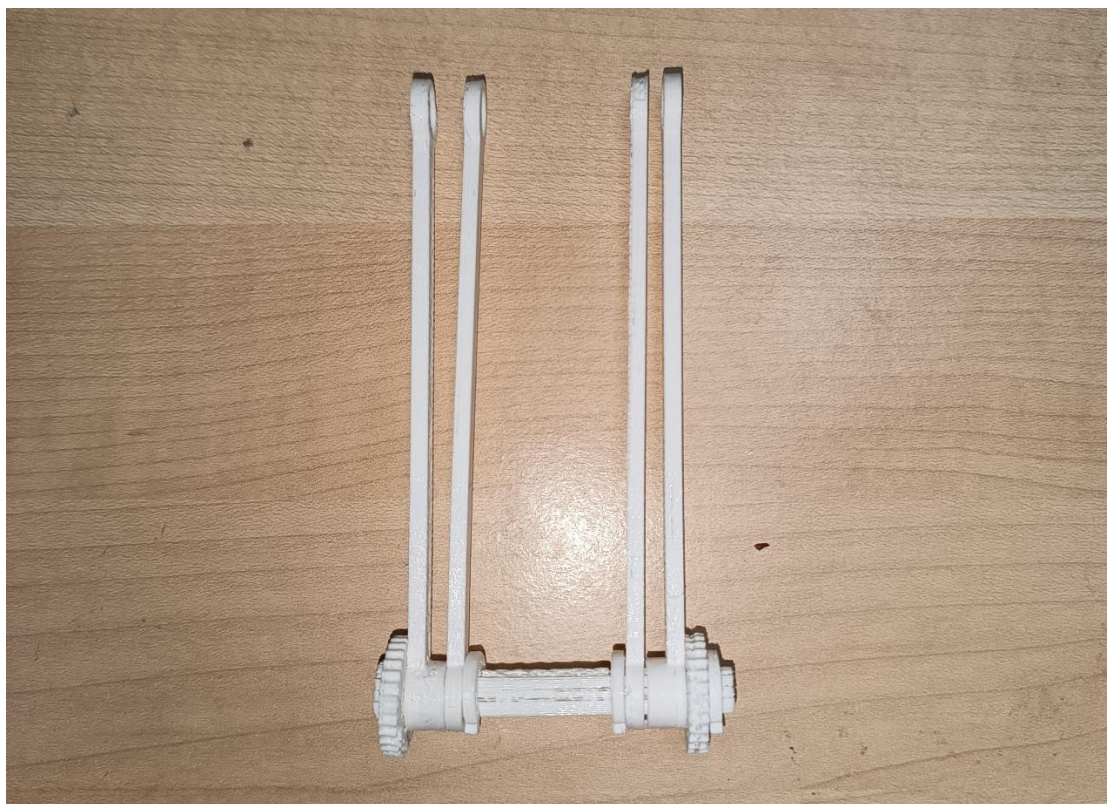


*Εικόνα 36: Συναρμολόγηση ποδιών του ρομπότ - βήμα 2*



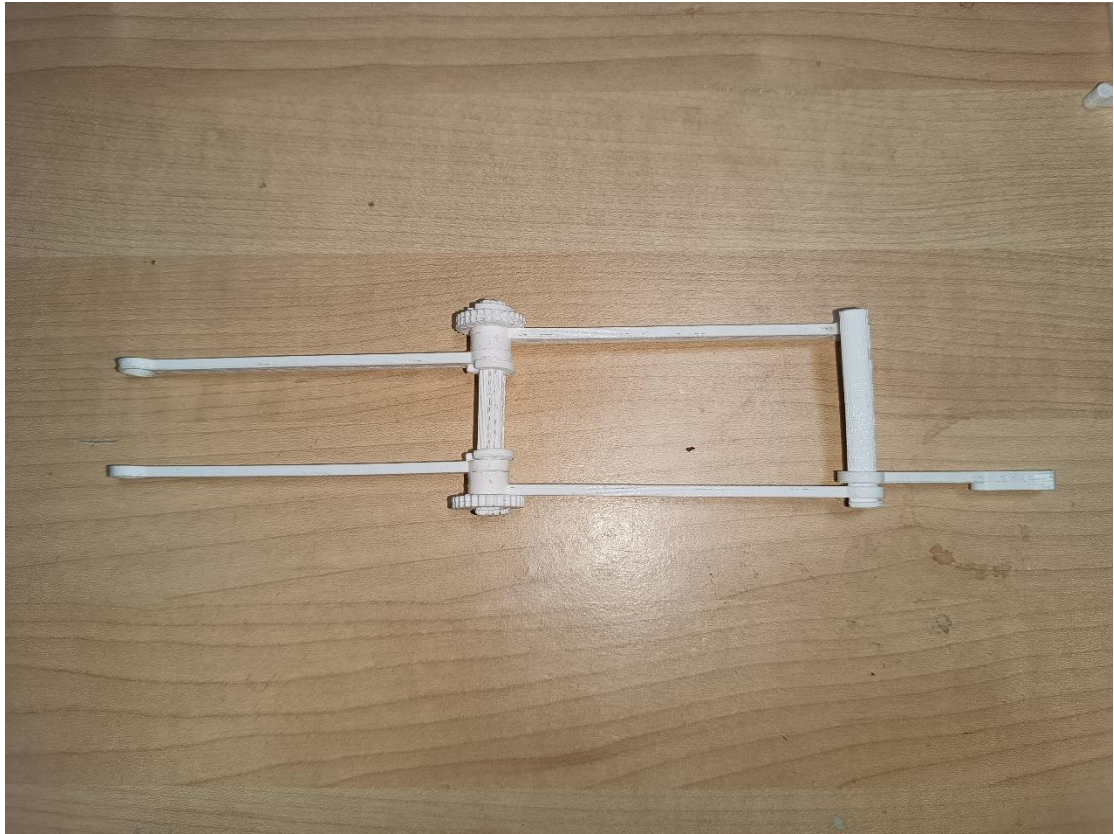


*Εικόνα 37: Συναρμολόγηση ποδιών του ρομπότ - βήμα 3 έως 4*

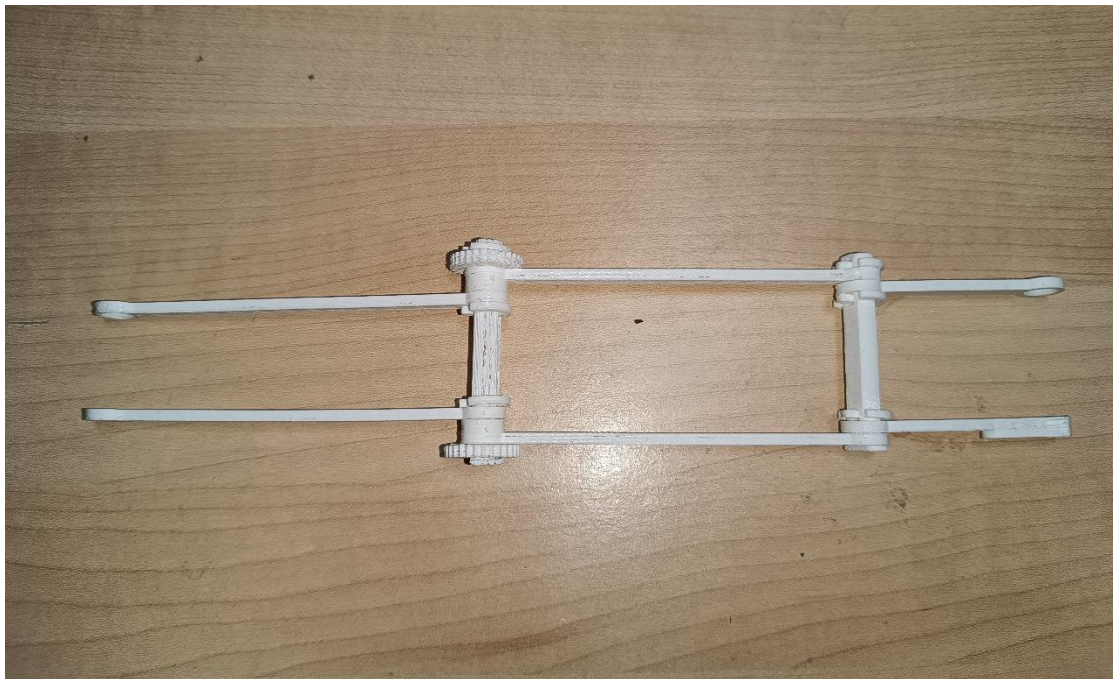


*Εικόνα 38: Συναρμολόγηση ποδιών του ρομπότ - βήμα 5 έως 8*



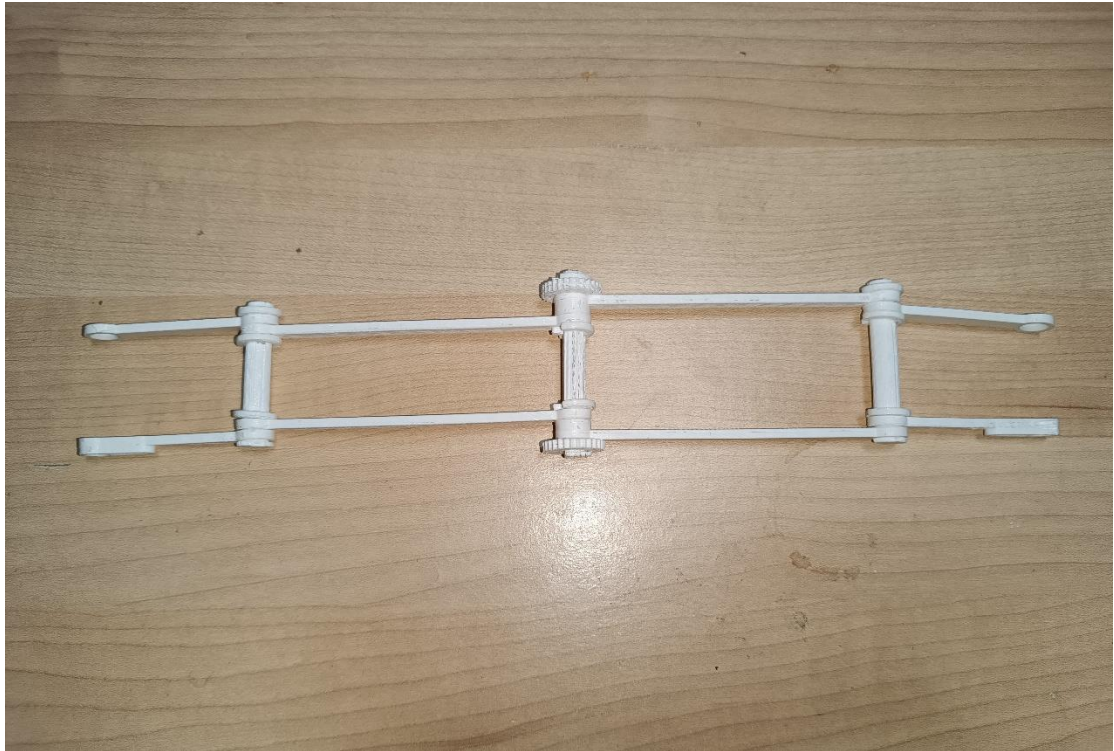


*Εικόνα 39: Συναρμολόγηση ποδιών του ρομπότ - βήμα 9 έως 11*



*Εικόνα 40: Συναρμολόγηση ποδιών του ρομπότ - βήμα 12 έως 14*





*Εικόνα 41: Συναρμολόγηση ποδιών του ρομπότ - βήμα 15 έως 18*

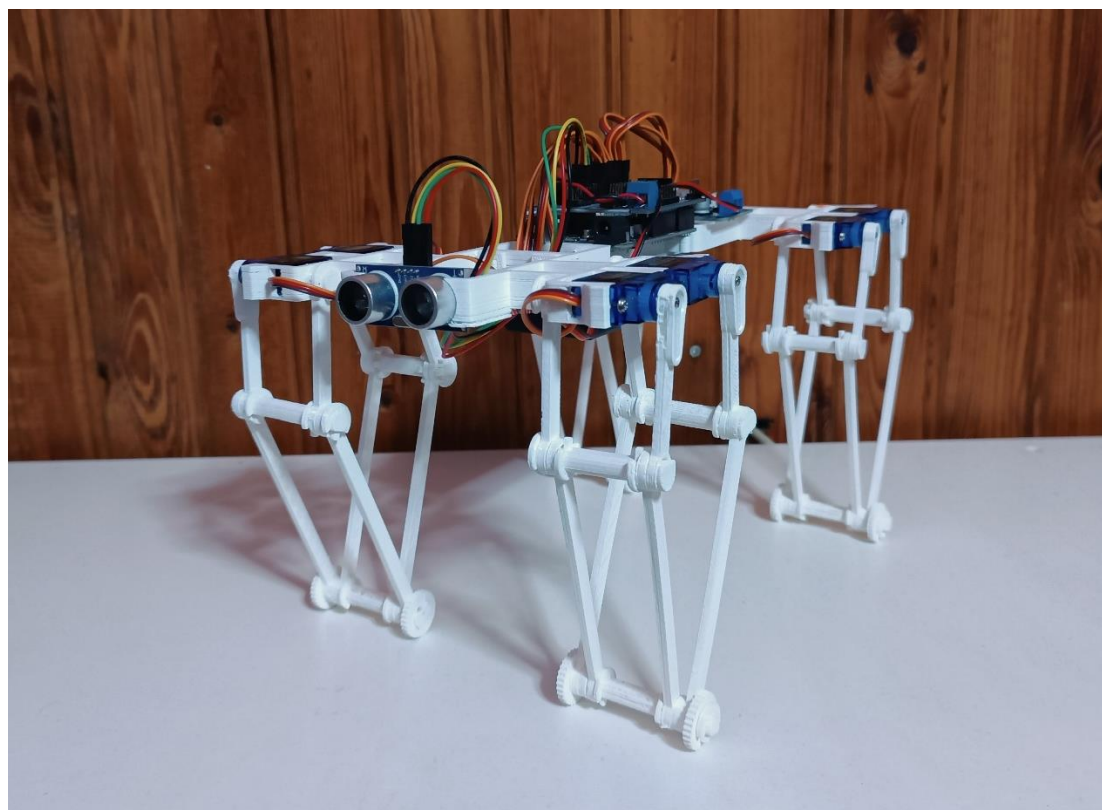


*Εικόνα 42: Τα 4 πόδια του ρομπότ συναρμολογημένα - βήμα 19*

Οι τελικές διαστάσεις του ρομπότ, συμπεριλαμβανομένων των ενεργών στοιχείων και των καλωδίων, είναι 20 εκατοστά ύψος \* 28 εκατοστά μήκος \* 18 εκατοστά πλάτος και το συνολικό κόστος κατασκευής είναι περίπου 84€ όπως φαίνεται στον παρακάτω πίνακα, χωρίς να υπολογίζεται το κόστος του υλικού PLA που χρησιμοποιήθηκε και του 3D Εκτυπωτή του Τμήματος Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών του ΔΙΠΑΕ.

Πίνακας 3: Αναλυτικό κόστος της κατασκευής του βαδίζοντος ρομπότ

Υλικά	Κόστος	Ποσότητα
Υλικό εκτύπωσης PLA	-	< 250 γραμμάρια
Arduino Uno	~ 30 €	1
Sensor Shield V5.0	2,50 €	1
DC-DC Adjustable converter	5,64 €	1
Βάση μπαταριών	1,20 €	1
Φορτιστής 2 μπαταριών	6,69 €	1
Μπαταρίες	4,39 €/τμχ	2
Σερβομηχανισμοί	3,60 €/τμχ	8
Καλώδια	0.60 €	Πακέτο 10 τμχ
<b>Σύνολο</b>	<b>~ 84,21 €</b>	



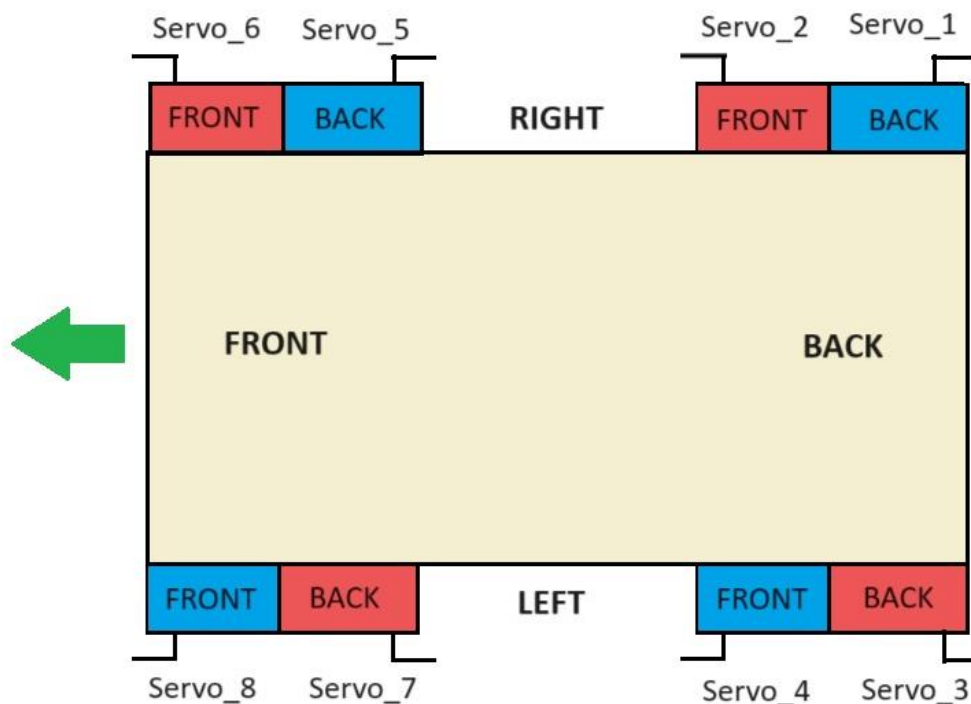
Εικόνα 43: Το ολοκληρωμένο ρομπότ



#### 4.4 Δοκιμαστικά προγράμματα και αποτελέσματα

Για να προγραμματίσουμε το Arduino ώστε να μπορεί το ρομπότ να περπατάει πρέπει να αναλύσουμε τις μοίρες που θα πρέπει να στρέφεται ο κάθε σερβομηχανισμός ώστε να εκτελείται σωστά η κάθε κίνηση. Οι κινήσεις που θα μπορεί να κάνει το ρομπότ είναι οι εξής: κίνηση προς τα μπροστά και προς τα πίσω, στροφή προς τα δεξιά και προς τα αριστερά. Καθώς επίσης και κάποιες κινήσεις-πόζες όπως: να κάθεται στα δύο πίσω πόδια και στα δύο μπροστά πόδια και τέλος να ξαπλώνει και με τα 4 πόδια λυγισμένα. Έτσι, θα δημιουργήσουμε 7 συναρτήσεις που θα κάνουν τις παραπάνω κινήσεις και επιπλέον 1 συνάρτηση που θα σταματάει το ρομπότ και θα το τοποθετεί σε μια ουδέτερη στάση όρθιο. Συνολικά, δηλαδή, θα έχουμε 8 συναρτήσεις για την κίνηση του ρομπότ.

Αρχικά, πρέπει πρώτα να δώσουμε κάποιες εντολές στη συνάρτηση setup του Arduino ώστε να έρθουν όλα τα servo στην αρχική τους θέση (home position) όπως αναφέραμε σε προηγούμενη ενότητα για να μπορέσουμε να βιδώσουμε σωστά τα πόδια. Οι θέσεις αυτές θα είναι είτε 0 είτε 180 μοίρες δηλαδή θα πρέπει η αρχική θέση του κάθε servo να είναι προς τα κάτω και τα πόδια να βιδωθούν κάθετα στο σώμα του ρομπότ.



Σχήμα 6: Η κάτοψη του ρομπότ

Πίνακας 4: Αρχική θέση (home position) των σερβομηχανισμών

Σερβομηχανισμός	Home Position σε μοίρες
Πίσω δεξί πόδι – πίσω servo: Servo_1	180
Πίσω δεξί πόδι – μπροστά servo: Servo_2	0
Πίσω αριστερό πόδι – πίσω servo: Servo_3	0
Πίσω αριστερό πόδι – μπροστά servo: Servo_4	180
Μπροστά δεξί πόδι – πίσω servo: Servo_5	180
Μπροστά δεξί πόδι – μπροστά servo: Servo_6	0
Μπροστά αριστερό πόδι – πίσω servo: Servo_7	0
Μπροστά αριστερό πόδι – μπροστά servo: Servo_8	180

Πίνακας 5: Προτεινόμενη περιστροφή των servo σε μοίρες για κάθε θέση: Κίνηση προς τα εμπρός [15]

Servo Θέση	Servo 8	Servo 7	Servo 6	Servo 5	Servo 4	Servo 3	Servo 2	Servo 1
A	90	135	0	135	180	45	90	45
B	45	0	-	-	-	-	135	180
Γ	120	0	-	-	-	-	60	180
Δ	180	45	90	45	90	135	0	135
E	-	-	135	180	45	0	-	-
Z	-	-	60	180	120	0	-	-
A	90	135	0	135	180	45	90	45

Πίνακας 6: Προτεινόμενη περιστροφή των servo σε μοίρες για κάθε θέση: Στροφή προς τα αριστερά [15]

Servo Θέση	Servo 8	Servo 7	Servo 6	Servo 5	Servo 4	Servo 3	Servo 2	Servo 1
A	0	135	45	90	90	45	135	0
B	-	-	180	135	135	180	-	-
Γ	-	-	180	60	60	180	-	-
Δ	90	45	135	0	0	135	45	90
E	135	180	-	-	-	-	180	135
Z	60	180	-	-	-	-	180	60
A	0	135	45	90	90	45	135	0

Πρώτο δοκιμαστικό πρόγραμμα που εκτελέστηκε από το ρομπότ ήταν η συνάρτηση της κίνησης προς τα εμπρός. Χρησιμοποιώντας την προτεινόμενη συνάρτηση του σχεδιαστή του ρομπότ που χρησιμοποιήσαμε, το ρομπότ κατάφερε να περπατήσει προς τα εμπρός. Με αυτήν την συνάρτηση το ρομπότ είχε μεγάλο διασκελισμό των ποδιών και κινούνταν με σχετικά μεγάλη ταχύτητα, αφού ο σκοπός του ήταν να συμμετέχει σε αγώνες ταχύτητας. Ωστόσο, λόγω της αυξημένης ταχύτητας

του, το ρομπότ έχανε την ισορροπία του και πολλές φορές αναποδογύριζε ή γονάτιζαν τα δυο του πόδια και δεν μπορούσε να ξανασηκωθεί. Συνεπώς, έπρεπε να βρεθεί μια λύση σε αυτό το πρόβλημα, διότι η μεγάλη του ταχύτητα επηρέαζε και άλλες λειτουργίες του ρομπότ, όπως την αποφυγή εμποδίων. Επιπλέον, λόγω ολίσθησης των ποδιών σε εδάφη όπως πλακάκια, έγινε δοκιμή να τοποθετηθούν λαστιχάκια στα πόδια του, αλλά επειδή δεν εφάρμοζαν καλά και ήταν πιο λεπτά από το πλάτος του ποδιού, έφευγαν κατά την κίνηση.

Συνδυάζοντας αυτήν την συνάρτηση της κίνησης προς τα εμπρός με την προτεινόμενη συνάρτηση της στροφής προς τα αριστερά, δημιουργήσαμε ένα μικρό πρόγραμμα για την αποφυγή εμποδίων. Το πρόγραμμα αυτό εκτελούσε την συνάρτηση `loop()` επαναλαμβανόμενα και έδινε στο ρομπότ κάποιες εντολές. Το ρομπότ ξεκινούσε να πηγαίνει ευθεία και μόλις εντόπιζε εμπόδια σε απόσταση μικρότερη από 15 εκατοστά, θα έστριβε αριστερά. Η μέτρηση της απόστασης γινόταν μέσω του αισθητήρα υπερήχων, με χρήση της κατάλληλης συνάρτησης. Ωστόσο, ενώ το ρομπότ εντόπιζε το εμπόδιο και ξεκινούσε να στρίψει, δεν προλάβαινε να αποφύγει το εμπόδιο χωρίς να τρακάρει. Μία λύση σε αυτό το πρόβλημα ήταν να δοκιμάσουμε να αυξήσουμε την απόσταση από την οποία το ρομπότ θα ξεκινούσε την στροφή, δηλαδή να εντόπιζε εμπόδιο σε απόσταση μικρότερη από 30 εκατοστά αυτή τη φορά. Τότε το ρομπότ θα ακινητοποιείται, θα κάνει λίγα βήματα πίσω και μετά θα στρίψει. Με αυτήν την αλλαγή βελτιώθηκε λίγο η αποφυγή εμποδίων αλλά και πάλι ορισμένες φορές δεν γινόταν έγκαιρα η στροφή ή το ρομπότ έχανε την ισορροπία του και έπεφτε.

Τελικά, κάνοντας διάφορες δοκιμές στις μοίρες που θα πήγαιναν τα `servo` σε κάθε φάση της κίνησης, έχοντας υπόψη να σηκώνονται αρκετά ψηλά τα πόδια ώστε να μπορεί το ρομπότ να ανέβει ένα μικρό σκαλοπάτι ή εμπόδιο ύψους 1-2 εκατοστών, καταλήξαμε σε αυτές που αναφέρονται στην παρακάτω ενότητα. Σε κάθε περίπτωση, οι αρθρώσεις των ποδιών δεν θα ανεβαίνουν πιο πάνω από το ύψος του σώματος, με εξαίρεση τις κινήσεις καθίσματος, διασφαλίζοντας έτσι την σταθερότητα του ρομπότ. Επίσης, αυξάνοντας και την καθυστέρηση (`delay`) ανάμεσα από κάθε φάση/κίνηση από 75 σε 100, καταφέραμε να μειώσουμε την ταχύτητα. Έτσι, εφόσον το ρομπότ έκανε πιο μικρά βήματα και με μικρότερη ταχύτητα λύθηκε και το πρόβλημα της σταθερότητας του, ενώ παράλληλα λύθηκε και το πρόβλημα την έγκαιρης αποφυγής των εμποδίων.

Η δεύτερη φάση των δοκιμαστικών προγραμμάτων αφορούν το κύριο πρόγραμμα και την συνάρτηση `loop()`. Μέχρι στιγμής το ρομπότ όταν συναντούσε εμπόδιο έστριβε μόνο αριστερά, οπότε έπρεπε με κάποιον τρόπο να συμπεριλάβουμε

και την στροφή προς τα δεξιά. Πρώτη σκέψη ήταν το ρομπότ να προχωράει ευθεία και όταν συναντούσε εμπόδιο να στρίβει πρώτα αριστερά, να παίρνει μια μέτρηση σε αυτή τη θέση, την οποία κρατούσε σε μια μεταβλητή που λεγόταν Distance\_Left και στη συνέχεια να έστριβε όλο δεξιά 180 μοίρες και να έπαιρνε και εκεί μια μέτρηση Distance\_Right. Έπειτα, θα σύγκρινε τις δυο τιμές και όποια απόσταση ήταν μεγαλύτερη θα πήγαινε προς τα εκείνη την κατεύθυνση. Ωστόσο, το πρόγραμμα αυτό δεν δούλεψε τόσο καλά και ήταν κάπως χρονοβόρο μέχρι να αποφασίσει το ρομπότ ποια κατεύθυνση θα πάρει. Συνεπώς, λαμβάνοντας υπόψη όλες τις παραπάνω δοκιμές καταλήξαμε στο τελικό πρόγραμμα που περιγράφεται αναλυτικά σε επόμενη ενότητα αυτού του κεφαλαίου.

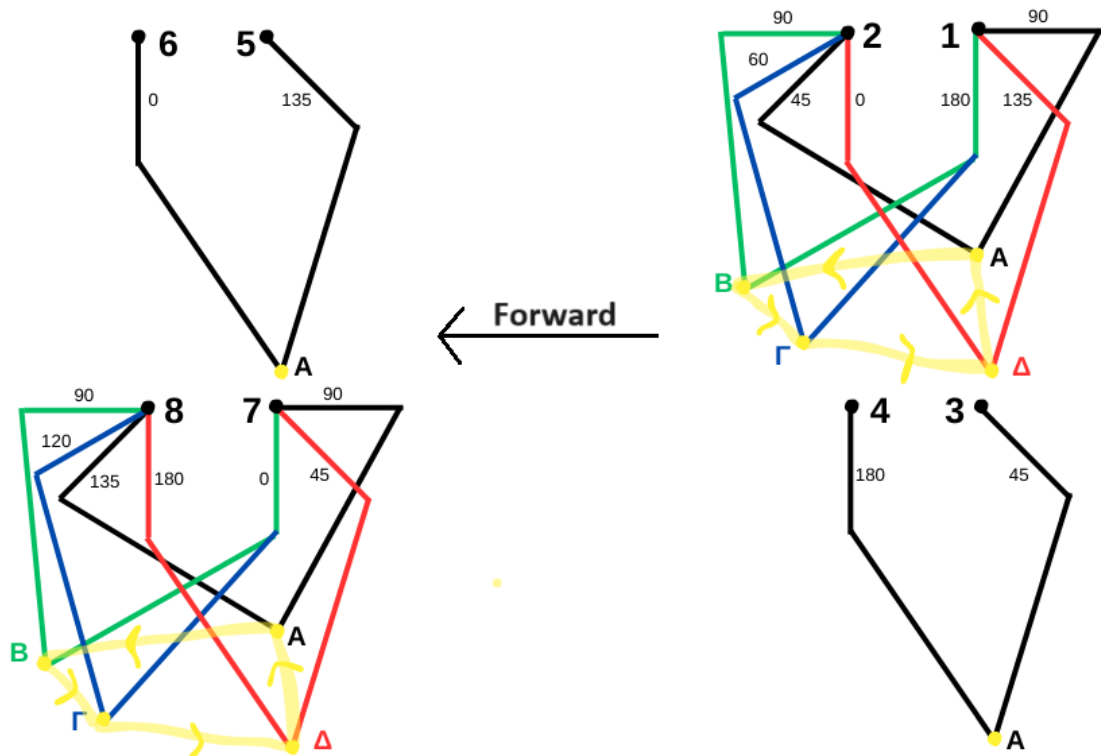
## 4.5 Κίνηση των ποδιών

Όπως αναφέραμε στην προηγούμενη ενότητα, οι κινήσεις του ρομπότ θα γίνονται με μια συνάρτηση για την κάθε μια. Για την κίνηση του ρομπότ προς τα εμπρός, θα πρέπει το ένα μπροστά πόδι και το αντίθετο πίσω πόδι να κάνουν μια κίνηση προς τα εμπρός, ενώ τα άλλα δυο πόδια να παραμένουν σταθερά για να διατηρείται η ισορροπία. Αφού έχουν ολοκληρώσει την κίνηση τους τα δύο αυτά πόδια, θα παραμένουν αυτά σταθερά και τα άλλα δύο θα κάνουν την αντίστοιχη εμπρός κίνηση.

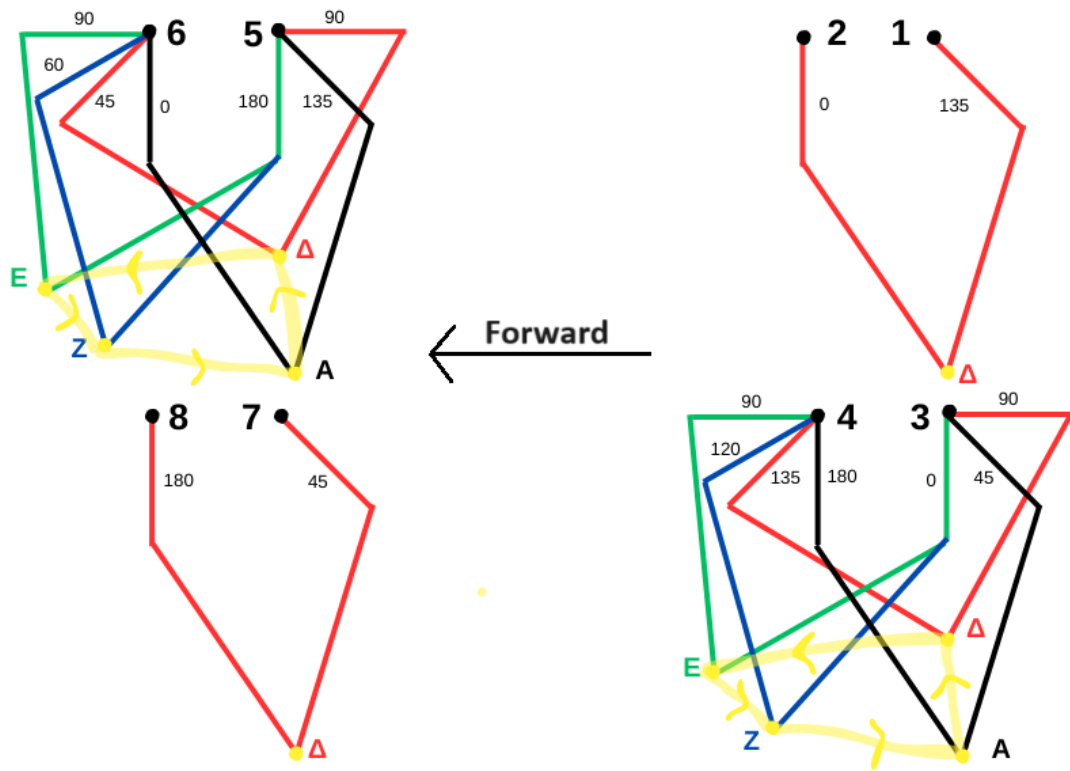
Η εμπρός κίνηση ενός ποδιού γίνεται με την τοποθέτηση του άκρου του ποδιού σε 4 θέσεις, την μια μετά την άλλη. Συνεπώς, τα δυο αντίθετα πόδια εκτελούν αυτή την κίνηση τεσσάρων θέσεων και παραμένουν στην τέταρτη ώστε να ακολουθήσουν τα άλλα δυο πόδια με την αντίστοιχη κίνηση κ.ο.κ.. Οι θέσεις αυτές φαίνονται στον πίνακα και τα σχήματα παρακάτω:

*Πίνακας 7: Περιστροφή των servo σε μοίρες για κάθε θέση: Κίνηση προς τα εμπρός*

<i>Servo</i> <i>Θέση</i>	<i>Servo</i> <b>8</b>	<i>Servo</i> <b>7</b>	<i>Servo</i> <b>6</b>	<i>Servo</i> <b>5</b>	<i>Servo</i> <b>4</b>	<i>Servo</i> <b>3</b>	<i>Servo</i> <b>2</b>	<i>Servo</i> <b>1</b>
<b>A</b>	135	90	0	135	180	45	45	90
<b>B</b>	90	0	-	-	-	-	90	180
<b>Γ</b>	120	0	-	-	-	-	60	180
<b>Δ</b>	180	45	45	90	135	90	0	135
<b>E</b>	-	-	90	180	90	0	-	-
<b>Z</b>	-	-	60	180	120	0	-	-
<b>A</b>	135	90	0	135	180	45	45	90



Σχήμα 7: Προβολή των ποδιών του ρομπότ από αριστερά: Κίνηση προς τα εμπρός - 1



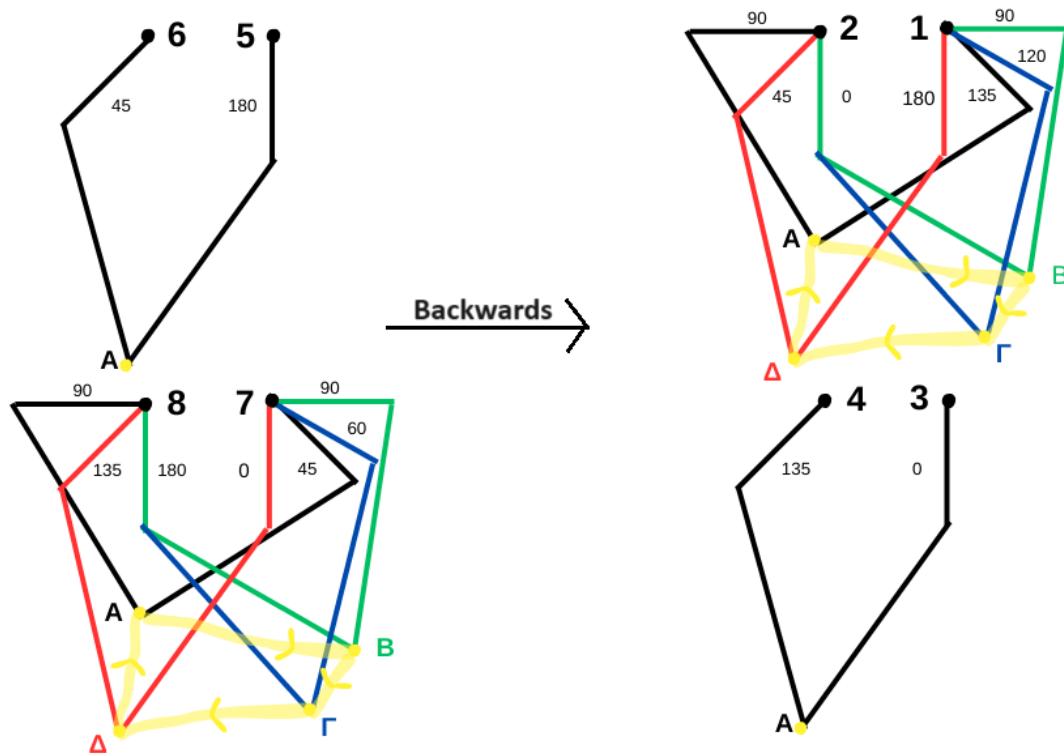
Σχήμα 8: Προβολή των ποδιών του ρομπότ από αριστερά: Κίνηση προς τα εμπρός - 2

Για την κίνηση του ρομπότ προς τα πίσω συμβαίνει ακριβώς το αντίθετο απ' ότι στην παραπάνω κίνηση. Δηλαδή θα πρέπει το ένα μπροστά πόδι και το αντίθετο πίσω πόδι να κάνουν μια κίνηση προς τα πίσω, ενώ τα άλλα δυο πόδια να παραμένουν σταθερά για να διατηρείται η ισορροπία. Αφού έχουν ολοκληρώσει την κίνηση τους τα δύο αυτά πόδια, θα παραμένουν αυτά σταθερά και τα άλλα δύο θα κάνουν την αντίστοιχη πίσω κίνηση.

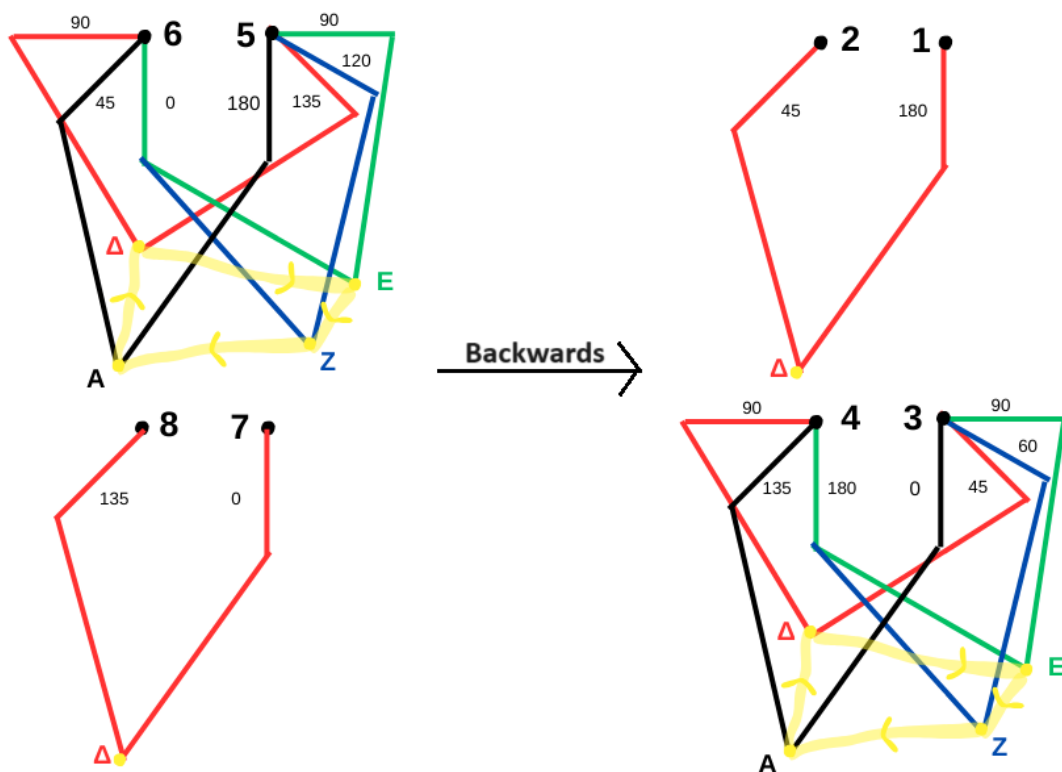
Η πίσω κίνηση ενός ποδιού γίνεται πάλι σε 4 θέσεις όπως και πριν αντίστοιχα. Συνεπώς, τα δυο αντίθετα πόδια εκτελούν αυτή την κίνηση τεσσάρων θέσεων και παραμένουν στην τέταρτη ώστε να ακολουθήσουν τα άλλα δυο πόδια με την αντίστοιχη κίνηση κ.ο.κ.. Οι θέσεις αυτές φαίνονται στον πίνακα και τα σχήματα παρακάτω:

Πίνακας 8: Περιστροφή των servo σε μοίρες για κάθε θέση: Κίνηση προς τα πίσω

Servo Θέση	Servo 8	Servo 7	Servo 6	Servo 5	Servo 4	Servo 3	Servo 2	Servo 1
A	90	45	45	180	135	0	90	135
B	180	90	-	-	-	-	0	90
Γ	180	60	-	-	-	-	0	120
Δ	135	0	90	135	90	45	45	180
E	-	-	0	90	180	90	-	-
Z	-	-	0	120	180	60	-	-
A	90	45	45	180	135	0	90	135



Σχήμα 9: Προβολή των ποδιών του ρομπότ από αριστερά: Κίνηση προς τα πίσω - 1



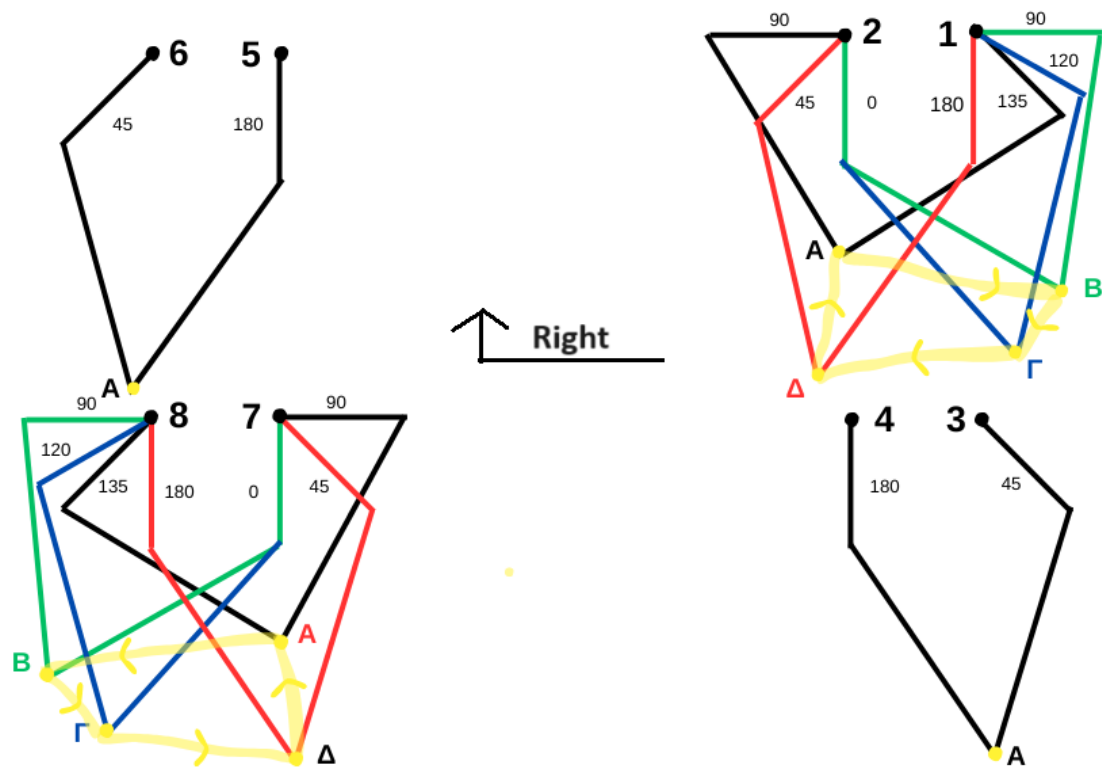
Σχήμα 10: Προβολή των ποδιών του ρομπότ από αριστερά: Κίνηση προς τα πίσω - 2

Για την επιτόπου στροφή του ρομπότ προς τα δεξιά, θα πρέπει το μπροστά αριστερό πόδι να κάνει κίνηση προς τα μπροστά, ενώ ταυτόχρονα το πίσω δεξί πόδι να κάνει κίνηση προς τα πίσω. Παράλληλα, τα άλλα δυο πόδια να παραμένουν σταθερά για να διατηρείται η ισορροπία. Αφού έχουν ολοκληρώσει την κίνηση τους τα δύο αυτά πόδια, θα παραμένουν αυτά σταθερά και τα άλλα δύο θα κινούνται αντίστοιχα, δηλαδή το μπροστά δεξί πόδι να κάνει κίνηση προς τα πίσω, ενώ το πίσω αριστερό προς τα μπροστά.

Η δεξιά στροφή γίνεται πάλι σε 4 θέσεις όπως και πριν αντίστοιχα. Συνεπώς, τα δυο αντίθετα πόδια εκτελούν την παραπάνω κίνηση τεσσάρων θέσεων και παραμένουν στην τέταρτη ώστε να ακολουθήσουν τα άλλα δυο πόδια με την αντίστοιχη κίνηση κ.ο.κ.. Γενικά, σε αυτήν την επιτόπου στροφή προς τα δεξιά τα δυο αριστερά πόδια να κινούνται προς τα μπροστά, ενώ τα δυο δεξιά θα κινούνται προς τα πίσω με τον τρόπο που αναφέραμε παραπάνω. Οι θέσεις αυτές φαίνονται στον πίνακα και τα σχήματα παρακάτω:

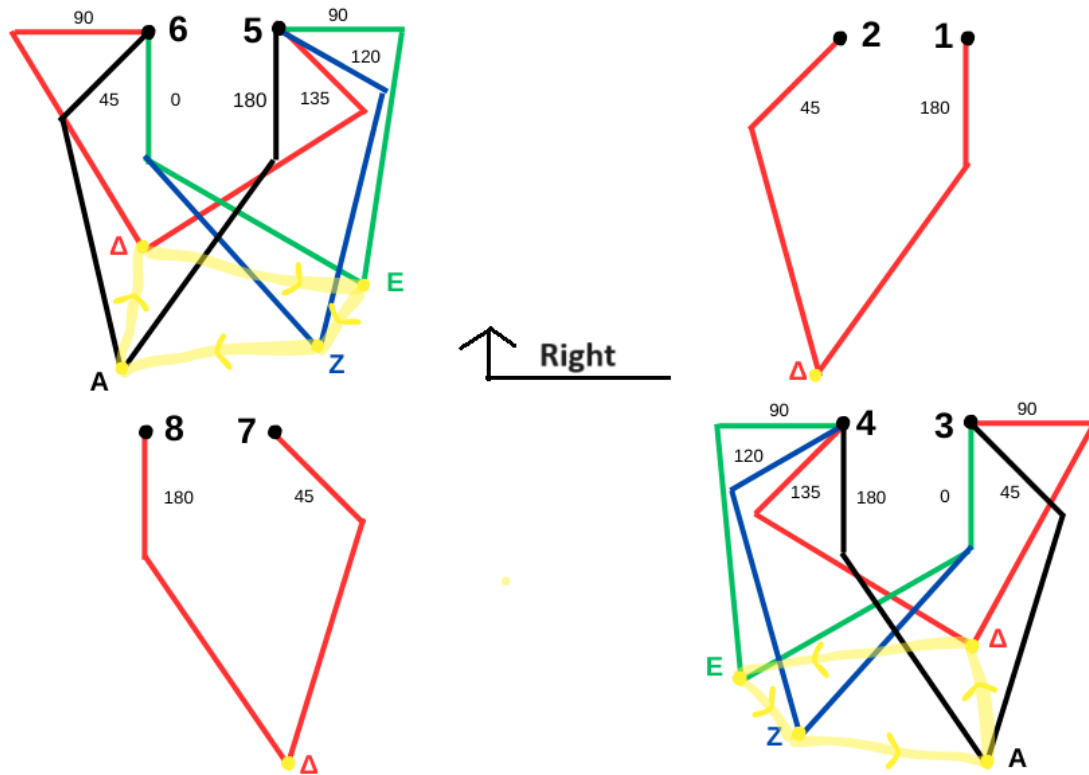
Πίνακας 9: Περιστροφή των servo σε μοίρες για κάθε θέση: Στροφή προς τα δεξιά

Servo Θέση	Servo 8	Servo 7	Servo 6	Servo 5	Servo 4	Servo 3	Servo 2	Servo 1
A	135	90	45	180	180	45	90	135
B	90	0	-	-	-	-	0	90
Γ	120	0	-	-	-	-	0	120
Δ	180	45	90	135	135	90	45	180
E	-	-	0	90	90	0	-	-
Z	-	-	0	120	120	0	-	-
A	135	90	45	180	180	45	90	135



Σχήμα 11: Προβολή των ποδιών του ρομπότ από αριστερά: Στροφή προς τα δεξιά - 1





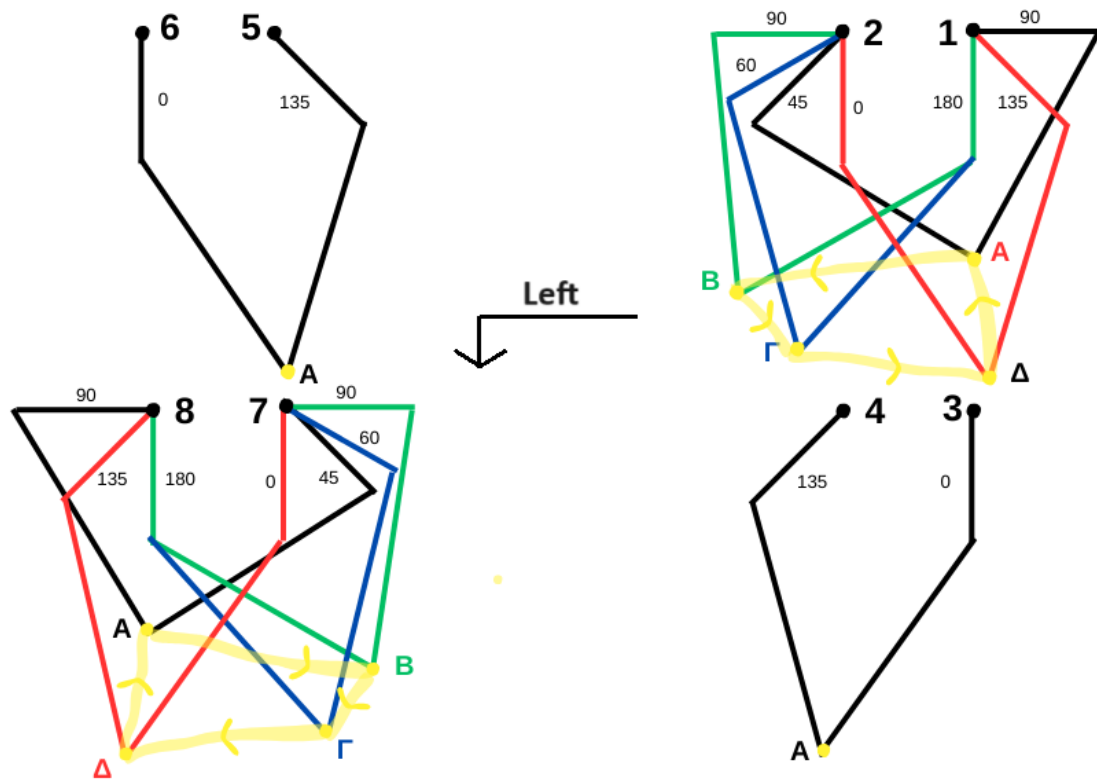
Σχήμα 12: Προβολή των ποδιών του ρομπότ από αριστερά: Στροφή προς τα δεξιά - 2

Για την επιτόπου στροφή του ρομπότ προς τα αριστερά συμβαίνει το αντίθετο από την στροφή προς τα δεξιά. Δηλαδή, θα πρέπει το μπροστά αριστερό πόδι να κάνει κίνηση προς τα πίσω, ενώ ταυτόχρονα το πίσω δεξί πόδι να κάνει κίνηση προς τα μπροστά. Παράλληλα, τα άλλα δυο πόδια να παραμένουν σταθερά για να διατηρείται η ισορροπία. Αφού έχουν ολοκληρώσει την κίνηση τους τα δύο αυτά πόδια, θα παραμένουν αυτά σταθερά και τα άλλα δύο θα κινούνται αντίστοιχα, δηλαδή το μπροστά δεξί πόδι να κάνει κίνηση προς τα μπροστά, ενώ το πίσω αριστερό προς τα πίσω.

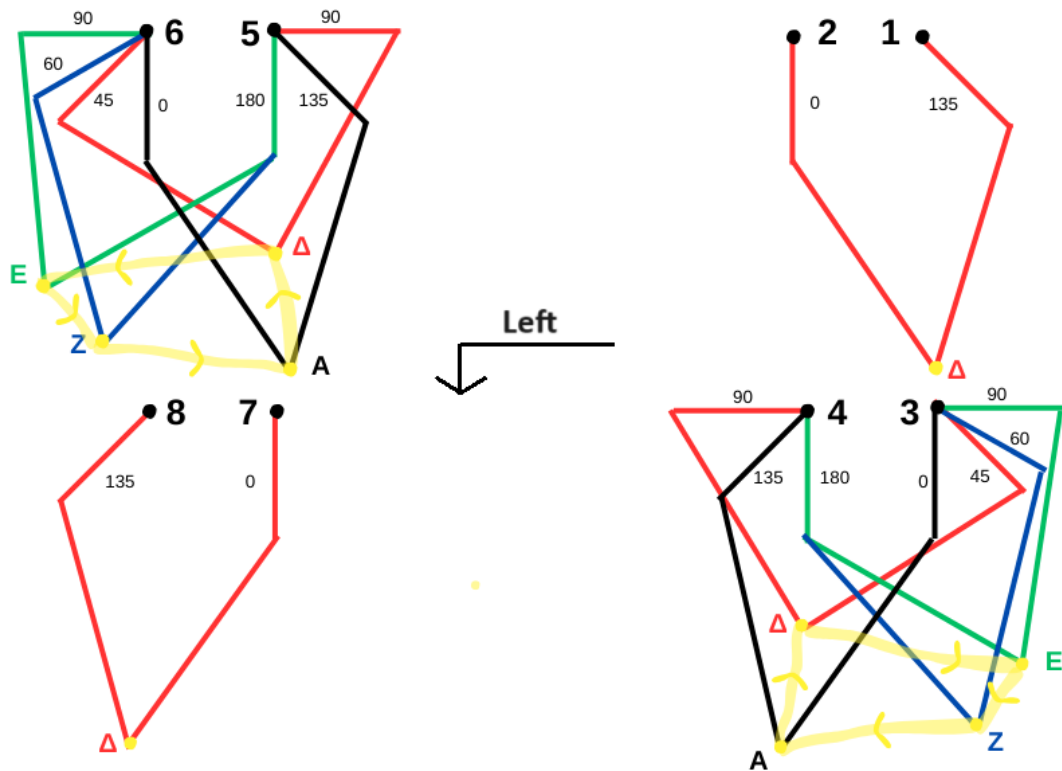
Η αριστερή στροφή γίνεται πάλι σε 4 θέσεις όπως και πριν αντίστοιχα. Συνεπώς, τα δυο αντίθετα πόδια εκτελούν την παραπάνω κίνηση τεσσάρων θέσεων και παραμένουν στην τέταρτη ώστε να ακολουθήσουν τα άλλα δυο πόδια με την αντίστοιχη κίνηση κ.ο.κ.. Γενικά, σε αυτήν την επιτόπου στροφή προς τα αριστερά τα δυο αριστερά πόδια να κινούνται προς τα πίσω, ενώ τα δυο δεξιά θα κινούνται προς τα μπροστά με τον τρόπο που αναφέραμε παραπάνω. Οι θέσεις αυτές φαίνονται στον πίνακα και τα σχήματα παρακάτω:

Πίνακας 10: Περιστροφή των servo σε μοίρες για κάθε θέση: Στροφή προς τα αριστερά

Servo Θέση	Servo 8	Servo 7	Servo 6	Servo 5	Servo 4	Servo 3	Servo 2	Servo 1
A	90	45	0	135	135	0	45	90
B	180	90	-	-	-	-	90	180
Γ	180	60	-	-	-	-	60	180
Δ	135	0	45	90	90	45	0	135
E	-	-	90	180	180	90	-	-
Z	-	-	60	180	180	60	-	-
A	90	45	0	135	135	0	45	90



Σχήμα 13: Προβολή των ποδιών του ρομπότ από αριστερά: Στροφή προς τα αριστερά - 1



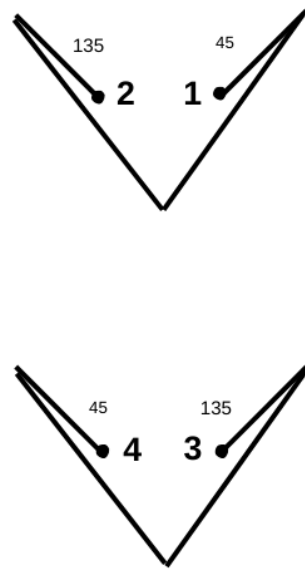
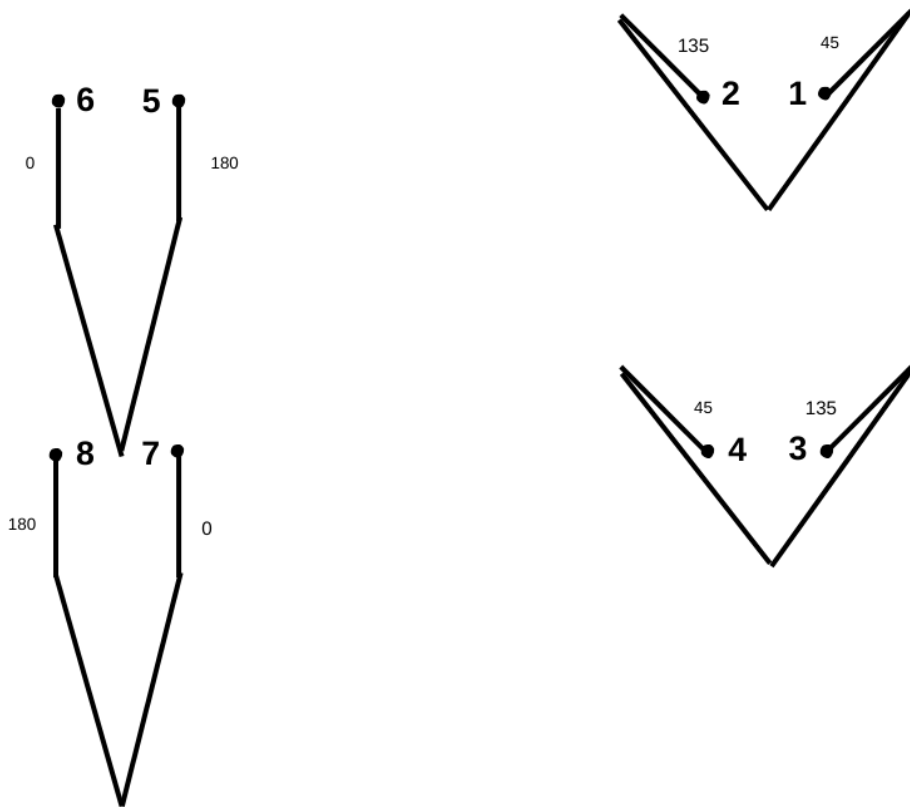
Σχήμα 14: Προβολή των ποδιών του ρομπότ από αριστερά: Στροφή προς τα αριστερά - 2

Η κίνηση του ρομπότ που κάθεται στα πίσω ή τα μπροστά πόδια είναι πιο απλή. Όταν το ρομπότ κάθεται στα πίσω πόδια, όλα τα servo των μπροστά ποδιών πηγαίνουν στην αρχική θέση (home position) που έχουμε αναφέρει παραπάνω δηλαδή στις 0 ή 180 μοίρες ανάλογα. Τα servo των πίσω ποδιών πηγαίνουν είτε στις 135 είτε στις 45 μοίρες ανάλογα. Το αντίθετο συμβαίνει όταν το ρομπότ κάθεται στα μπροστά πόδια.

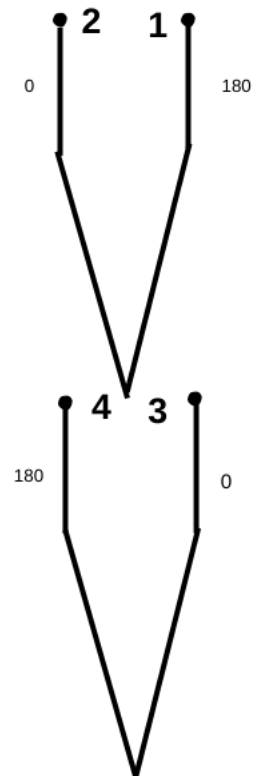
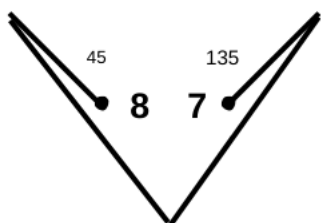
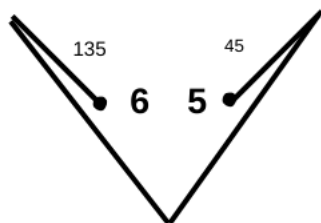
Επιπλέον, έχουμε την κίνηση που το ρομπότ ξαπλώνει λυγίζοντας και τα 4 πόδια του ταυτόχρονα. Σε αυτήν την περίπτωση τα πράγματα είναι ακόμη πιο απλά αφού και τα 8 servo πηγαίνουν στις 90 μοίρες. Τέλος, υπάρχει και μια ακόμη κίνηση η οποία σταματάει το ρομπότ και το τοποθετεί σε μια ουδέτερη πόζα. Σε αυτήν το ρομπότ στέκεται όρθιο με τα servo στις 170 ή 10 μοίρες ανάλογα. Όλα αυτά φαίνονται και στον πίνακα και τα σχήματα παρακάτω:

Πίνακας 11: Περιστροφή των servo σε μοίρες για κάθε κίνηση-πόζα

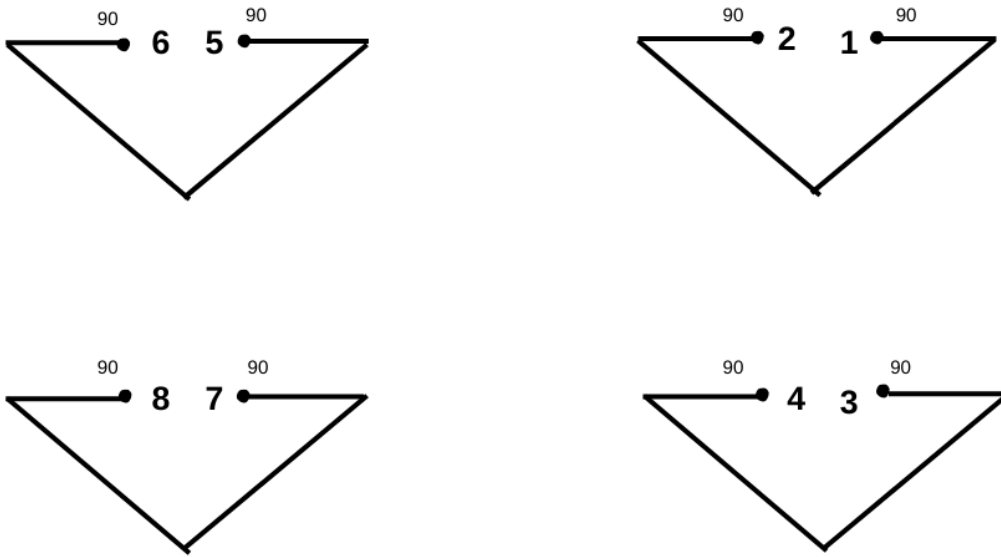
Servo Πόζα	Servo 8	Servo 7	Servo 6	Servo 5	Servo 4	Servo 3	Servo 2	Servo 1
Sit Back	180	0	0	180	45	135	135	45
Sit Forward	45	135	135	45	180	0	0	180
Lie	90	90	90	90	90	90	90	90
Stop	170	10	10	170	170	10	10	170



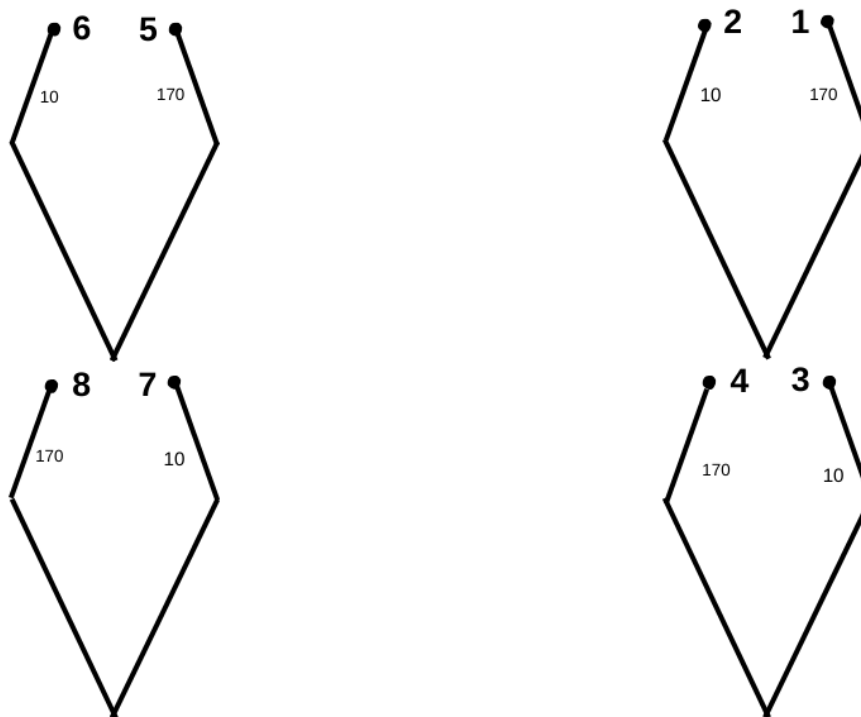
Σχήμα 15: Προβολή των ποδιών του ρομπότ από αριστερά: Κίνηση-πόζα Sit Back



Σχήμα 16: Προβολή των ποδιών του ρομπότ από αριστερά: Κίνηση-πόζα Sit Forward



Σχήμα 17: Προβολή των ποδιών του ρομπότ από αριστερά: Κίνηση-πόζα Lie



Σχήμα 18: Προβολή των ποδιών του ρομπότ από αριστερά: Κίνηση-πόζα Stop

## 4.6 Τελικός προγραμματισμός του Arduino

Όπως αναφέραμε σε προηγούμενη ενότητα, θα δημιουργήσουμε 8 συναρτήσεις για τις κινήσεις του ρομπότ, τις οποίες θα μπορούμε να χρησιμοποιήσουμε στο κύριο μέρος του προγράμματος. Επιπλέον, θα χρειαστούμε μια ακόμα συνάρτηση η οποία θα χρησιμοποιεί το αποστασιόμετρο και θα υπολογίζει την απόσταση ώστε να μπορεί το ρομπότ να αποφεύγει τα εμπόδια που συναντά. Οι συναρτήσεις αυτές είναι οι εξής:

1. **RobotForward()** : Κίνηση του ρομπότ προς τα εμπρός
2. **RobotBackward()** : Κίνηση του ρομπότ προς τα πίσω
3. **RobotRight()** : Στροφή του ρομπότ προς τα δεξιά
4. **RobotLeft()** : Στροφή του ρομπότ προς τα αριστερά
5. **RobotSitBack()** : Κίνηση-πόζα του ρομπότ που κάθεται στα πίσω πόδια
6. **RobotSitForward()** : Κίνηση-πόζα του ρομπότ που κάθεται στα μπροστά πόδια
7. **RobotLie()** : Κίνηση-πόζα του ρομπότ που ξαπλώνει με λυγισμένα και τα 4 πόδια
8. **RobotStop()** : Κίνηση-πόζα του ρομπότ που σταματάει και στέκεται όρθιο σε ουδέτερη θέση
9. **Read\_Distance()** : Ανάγνωση απόστασης και μετατροπή της σε εκατοστά

Για τις συναρτήσεις που αφορούν την κίνηση του ρομπότ χρησιμοποιούμε την εντολή `servo.write(angle)`, όπου `servo` βάζουμε εμείς τα `servo` μας όπως τα ορίσαμε και `angle` η γωνία που θέλουμε να μετακινηθεί το `servo` από 0 έως 180 μοίρες. Επίσης, χρησιμοποιούμε άλλες εντολές όπως `servo.attach(pin)` και `servo.detach(pin)` για να συνδέσουμε ή να αποσυνδέσουμε, αντίστοιχα, ένα `servo` σε ένα `pin`. Συνεπώς, με την εντολή `write` δίνουμε τις γωνίες που επιθυμούμε να μετακινηθεί το κάθε `servo` όπως αναλύσαμε στην προηγούμενη ενότητα και ανάμεσα από κάθε θέση (A έως Z) βάζουμε ένα `delay` κάποιων μικροδευτερολέπτων για να προλάβει να ολοκληρωθεί η κάθε κίνηση.

Η συνάρτηση ανάγνωσης της απόστασης, χρησιμοποιώντας το αισθητήρα υπερήχων, εκπέμπει υπερηχητικά κύματα και στη συνέχεια μετρά το χρόνο που χρειάζονται τα κύματα για να αναπηδήσουν μετά την πρόσκρουση σε κάποιο αντικείμενο. Αυτό γίνεται στέλλοντας έναν παλμό HIGH διάρκειας 10 ή περισσότερων `microseconds` και έπειτα παρέχοντας έναν σύντομο παλμό LOW για να αποσταλεί ένας καθαρός παλμός HIGH. Τέλος, μετατρέπει αυτόν τον χρόνο σε απόσταση και τον επιστρέφει σε εκατοστά.

Αφού έχουμε φτιάξει τις παραπάνω συναρτήσεις, μπορούμε να ξεκινήσουμε να γράφουμε το κύριο μέρος του προγράμματος. Αρχικά, θα πρέπει να προσθέσουμε την βιβλιοθήκη <Servo.h> για τους σερβομηχανισμούς. Στη συνέχεια, ορίζουμε τα servo που θα χρησιμοποιηθούν και τα pin για τα Trigger και Echo του αισθητήρα υπερήχων. Περνώντας, τώρα, στο setup καθορίζουμε τα pin για τα servo και τα ρυθμίζουμε στην αρχική θέση (home position) για 2 δευτερόλεπτα, ώστε να μπορέσουμε να βιδώσουμε σωστά τα πόδια κατά την συναρμολόγηση του ρομπότ, όπως αναφέρθηκε σε προηγούμενη ενότητα. Έπειτα, αποσυνδέουμε τα servo για να αποφευχθεί το τρέμουλο κατά την ενεργοποίηση. Επιπλέον, κάνουμε αρχικοποίηση της σειριακής επικοινωνίας και ορίζουμε τα pin 12, 13 ως έξοδος και είσοδος, αντίστοιχα. Τέλος, δημιουργούμε μια ρουτίνα επίδειξης (Demo), όπου θα εκτελείται μόνο μια φορά κατά την εκκίνηση του ρομπότ και στην οποία το ρομπότ θα κάθεται στα πίσω πόδια, στα μπροστά και μετά θα ξαπλώνει.

Ξεκινάμε, λοιπόν, με την συνάρτηση loop όπου πρώτα ορίζουμε τις μεταβλητές για τις αποστάσεις distanceLeft και distance, δηλαδή την απόσταση στα αριστερά και ευθεία, αντίστοιχα. Με την εντολή distance = Read\_Distance() διαβάζουμε την απόσταση που βλέπει ευθεία το ρομπότ πριν ξεκινήσει. Ύστερα, ξεκινάει το ρομπότ να κινείται ευθεία, καθώς ελέγχει την απόσταση distance. Εάν η απόσταση αυτή γίνει μικρότερη από 30 εκατοστά τότε το ρομπότ σταματάει, κάνει πίσω και με την χρήση της “for” στρίβει 4 φορές επιτόπου αριστερά. Τότε διαβάζει την απόσταση σε αυτό το σημείο (distanceLeft) και αν αυτή η απόσταση είναι μεγαλύτερη από 30 εκατοστά, συνεχίζει ευθεία, αλλιώς αν είναι μικρότερη από 30 εκατοστά δηλαδή αν υπάρχει και στα αριστερά του εμπόδιο, τότε με την χρήση της “for” στρίβει 10 φορές επιτόπου δεξιά και συνεχίζει ευθεία. Η συνάρτηση loop εκτελείται επαναλαμβανόμενα και έτσι το ρομπότ περπατάει και αποφεύγει τα εμπόδια που συναντάει.



Robot.ino

```
73 void loop()
74 {
75     // Μεταβλητές για τις αποστάσεις
76     int distanceLeft = 0; // Απόσταση που παίρνει όταν στρίβει αριστερά
77     long distance = Read_Distance(); // Απόσταση όταν προχωράει ευθεία
78
79     RobotForward();
80
81     if (distance > 0)
82     {
83         if (distance < 30)
84         {
85             RobotStop();
86             delay(100);
87             RobotBackward();
88             // Επιτόπου στροφή προς τα αριστερά 4 φορές
89             for (int i = 0; i < 4; i++)
90             {
91                 RobotLeft();
92             }
93             delay(100);
94             distanceLeft = Read_Distance();
95             if (distanceLeft > 0)
96             {
97                 if (distanceLeft < 30)
98                 {
99                     // Επιτόπου στροφή προς τα δεξιά 10 φορές
100                    for (int j = 0; j < 9; j++)
101                    {
102                        RobotRight();
103                    }
104                }
105            }
106        }
107     }
108 }
```

Output Serial Monitor

Εικόνα 44: Η συνάρτηση loop του προγράμματος στο Arduino IDE

## Κεφάλαιο 5: Αξιολόγηση συμπερασμάτων

Η διαδικασία αξιολόγησης των συμπερασμάτων μπορεί να παρέχει την ευκαιρία για εντοπισμό μελλοντικών δυνατοτήτων του ρομπότ και πρόσθετη διερεύνηση ή βελτίωση. Ένα βαδίζον ρομπότ επιδιώκει να αναπαράγει την περίπλοκη ανθρώπινη ή ζωική κίνηση, πέρα από τα τυπικά ρομπότ με τροχούς. Η πρόκληση είναι να δημιουργηθεί μια μηχανή ικανή να χειρίζεται διάφορα εδάφη με σταθερότητα και αποτελεσματικότητα. Ο σχεδιασμός του ρομπότ λαμβάνει υπόψη παράγοντες όπως η επιλογή ενός κατάλληλου σκελετού, τα ενεργά στοιχεία, η ενσωμάτωση αισθητήρων κλπ. και απαιτείται να βρεθούν λύσεις σε προβλήματα όπως η ισορροπία, η βελτιστοποίηση της βάδισης και η αλληλεπίδραση με το περιβάλλον.

Η μέθοδος υλοποίησης του βαδίζοντος ρομπότ που εφαρμόστηκε, βοήθησε στην επίτευξη των στόχων του με επιτυχία. Η επιλογή του συγκεκριμένου σκελετού ήταν ιδανική, διότι μπορούσε να εκτυπωθεί εύκολα από τον 3D Εκτυπωτή που είχαμε στην διάθεση μας και σε σχετικά σύντομο χρονικό διάστημα. Το υλικό που απαιτούσε για την πλήρη εκτύπωση ήταν λιγότερο από μισό καρούλι 500 γραμμαρίων και το τελικό βάρος του ήταν σχετικά μικρό. Επίσης, διευκόλυνε πολύ την ενσωμάτωση των σερβομηχανισμών και των υπόλοιπων ενεργών στοιχείων και αισθητήρων, καθώς υπήρχαν ειδικές θέσεις για αυτά, τοποθετώντας τα εύκολα χωρίς να χρειαστεί η δημιουργία τρυπών για τις βίδες.

Επιπλέον, το Arduino Uno είναι μια πλακέτα εύκολη στην χρήση και κατάλληλη για τέτοιου είδους κατασκευές, σε συνδυασμό με το Sensor Shield V5.0. Η επιλογή του Shield ήταν σημαντική αφού έλυσε το πρόβλημα της συνδεσμολογίας των 8 σερβομηχανισμών και του αισθητήρα υπερήχων. Πέρα από αυτό, συνέβαλε αρκετά και στο ζήτημα της τροφοδοσίας του Arduino και των υπόλοιπων στοιχείων, καθώς μέσω αυτού μπορούν να τροφοδοτηθούν όλα από μια πηγή μπαταριών συνολικού φορτίου 7,4V, βάζοντας το DC-DC adjustable converter μεταξύ των μπαταριών και του Sensor shield. Ωστόσο, είναι πιθανό κάποια στιγμή να καταναλωθούν παραπάνω volt από κάποιον σερβομηχανισμό και το Arduino να εκτελέσει μια επανεκκίνηση του προγράμματος, μέχρις ότου να λάβει ξανά τα απαραίτητα volt για την λειτουργία του. Διαφορετικά, θα έπρεπε να τροφοδοτήσουμε ξεχωριστά το Arduino και τους σερβομηχανισμούς από διαφορετικές πηγές. Αυτό θα ξεπερνούσε τον παραπάνω

κίνδυνο, αλλά θα δημιουργούσε πιο πολύπλοκες συνδεσμολογίες και θα αύξανε το βάρος του ρομπότ, επηρεάζοντας την λειτουργία του.

Ο κώδικας του Arduino που δημιουργήθηκε τελικά, κατάφερε με επιτυχία να πραγματοποιήσει την βάδιση του ρομπότ και την αποφυγή εμποδίων. Ακόμα, είναι δυνατή η υπερπήδηση μικρού εμποδίου ή σκαλοπατιού 1-2 εκατοστών με την χρήση του συγκεκριμένου κώδικα. Ωστόσο, για υψηλότερα εμπόδια 2.5-4 εκατοστών απαιτείται διαφορετικός αλγόριθμος, όπου τα πόδια του ρομπότ θα σηκώνονται υψηλότερα για να μπορεί να τα ξεπεράσει, μειώνοντας όμως την σταθερότητα και την ισορροπία του. Ακόμα, για την ομαλότερη κίνησή του σε ολισθηρά εδάφη, όπως πλακάκια, θα μπορούσαν να τοποθετηθούν λαστιχάκια σιλικόνης με κατάλληλες διαστάσεις ώστε να εφαρμόζουν καλά στα πόδια, γύρω από τα στρογγυλά κομμάτια που έρχονται σε επαφή με έδαφος για καλύτερη πρόσφυση.

Συνοψίζοντας, η χρήση της μεθοδολογίας που ακολουθήθηκε για την δημιουργία αυτού του ρομπότ προσφέρει νέες γνώσεις και ιδέες για μελλοντικές κατασκευές ρομπότ με πόδια. Ο σκελετός του ρομπότ που επιλέχθηκε διευκόλυνε την διαδικασία κατασκευής και τα ενεργά στοιχεία που χρησιμοποιήθηκαν ήταν κατάλληλα για την κλίμακα του ρομπότ και για τους συγκεκριμένους σερβομηχανισμούς, οι οποίοι ήταν μικρών διαστάσεων και απαιτούσαν λίγα volt για την λειτουργία τους. Ο συγκεκριμένος τρόπος τροφοδοσίας του Arduino και των σερβομηχανισμών λειτούργησε αρκετά καλά σε αυτήν την περίπτωση. Ωστόσο, θα ήταν επιθυμητή και η ξεχωριστή τροφοδοσία τους, έτσι ώστε να μην υπάρχει κίνδυνος να καταναλωθούν παραπάνω volt από κάποιον σερβομηχανισμό που μπορεί να ζοριστεί και έτσι να υπολειτουργήσει το Arduino επειδή δεν θα δέχεται τα απαραίτητα volt.

## Κεφάλαιο 6: Σύνοψη και μελλοντική επέκταση

Η υλοποίηση του βαδιστικού ρομπότ απαιτούσε μια μεθοδική και σύνθετη διαδικασία. Η αρχική φάση αφορούσε την έρευνα για έτοιμα ρομπότ βάδισης, συγκεντρώνοντας βασικές πληροφορίες από διάφορες πηγές. Στη συνέχεια, οι συζητήσεις με τον επιβλέποντα καθηγητή καθόρισαν κρίσιμες αποφάσεις, ιδίως την επιλογή ενός κατάλληλου σκελετού ρομπότ, λαμβάνοντας υπόψη παράγοντες όπως η ευκολία κατασκευής και η ενσωμάτωση αισθητήρων και σερβομηχανισμών.

Η υλοποίηση της κατασκευής απαιτούσε την λήψη αρχείων STL για τρισδιάστατη εκτύπωση, την εκμάθηση της χρήσης ενός 3D εκτυπωτή και την εκτύπωση των εξαρτημάτων του ρομπότ. Μετά την συναρμολόγηση των εκτυπωμένων στοιχείων ακολούθησε η εγκατάσταση ενεργών στοιχείων, όπως Arduino, σερβομηχανισμών και αισθητήρα υπερήχων.

Το επόμενο βήμα ήταν ο προγραμματισμός του Arduino για την πραγματοποίηση του βαδίσματος, ο οποίος απαιτούσε επαναληπτικές δοκιμές και βελτιστοποίηση των κινήσεων βάδισης και στροφής. Τα αποτελέσματα αποκάλυψαν ότι η ταχύτητα θα πρέπει να προσαρμοστεί για μεγαλύτερη ισορροπία και έγκαιρη αποφυγή εμποδίων. Τέλος, το ρομπότ βάδισης κατείχε την ικανότητα να διασχίζει αποτελεσματικά το περιβάλλον του, κινούμενο αυτόνομα και αποφεύγοντας τα εμπόδια.

Ένας πιθανός τρόπος για μελλοντική ανάπτυξη του έργου του ρομπότ θα ήταν η επέκταση των δυνατοτήτων του για τηλεχειρισμό μέσω Η/Υ. Μια επιλογή είναι η σύνδεση του Arduino με Η/Υ μέσω WiFi, η οποία θα επιτρέψει τον απομακρυσμένο έλεγχο και την αλληλεπίδραση σε πραγματικό χρόνο, ξεκλειδώνοντας ένα πλήθος δυνατοτήτων για την παρακολούθηση, τον προγραμματισμό και τη βελτίωση της συμπεριφοράς του ρομπότ. Η υλοποίηση της συνδεσιμότητας WiFi θα μπορούσε να περιλαμβάνει την ενσωμάτωση μιας μονάδας WiFi στο υλικό του ρομπότ και τη διαμόρφωση του Arduino ώστε να επικοινωνεί με ένα αντίστοιχο πρόγραμμα ή διεπαφή με έναν υπολογιστή.

Τα οφέλη αυτής της μελλοντικής εξέλιξης είναι πολλαπλά. Ο απομακρυσμένος έλεγχος μέσω Η/Υ θα μπορούσε να διευκολύνει τον προγραμματισμό και τις προσαρμογές των κινήσεων του ρομπότ σε πραγματικό χρόνο. Ο χρήστης θα μπορούσε δοκιμάσει διαφορετικά μοτίβα βάδισης και να αντιμετωπίσει προβλήματα από

απόσταση, βελτιώνοντας την διαδικασία ανάπτυξης. Επιπλέον, αυτή η συνδεσιμότητα θα μπορούσε να ανοίξει το δρόμο για την ενσωμάτωση αισθητήρων στο ρομπότ για τη συλλογή και τη μετάδοση δεδομένων στον υπολογιστή. Τα δεδομένα αυτά θα μπορούσαν να αναλυθούν σε πραγματικό χρόνο, παρέχοντας πολύτιμες πληροφορίες για το περιβάλλον του ρομπότ. Επιπλέον, ανοίγει δρόμους για την εφαρμογή προηγμένων χαρακτηριστικών, όπως η ρομποτική όραση, επιτρέποντας στο ρομπότ να αντιλαμβάνεται και να ανταποκρίνεται στο περιβάλλον του πιο έξυπνα.

Συμπερασματικά, η ενσωμάτωση της συνδεσιμότητας WiFi μεταξύ του Arduino και ενός υπολογιστή αποτελεί μια ενδιαφέρουσα μελλοντική εξέλιξη για το βαδίζον ρομπότ. Αυτή η βελτίωση, όχι μόνο επεκτείνει τις δυνατότητες του ρομπότ, αλλά παρέχει επίσης μια ευκαιρία για συνεχή καινοτομία, έρευνα και εξέλιξη των λειτουργιών των ρομπότ βάδισης.

## Βιβλιογραφία

- [1] D. Edwards, «Agility robotics raises 8 million to develop its two legged robot,» Robotics & Automation News , 25 06 2018. [Ηλεκτρονικό]. Available: <https://roboticsandautomationnews.com/2018/07/25/agility-robotics-raises-8-million-to-develop-its-two-legged-robot/18487/>.
- [2] J. Chu, «Mini cheetah is the first four-legged robot to do a backflip,» MIT News | Massachusetts Institute of Technology, 04 03 2019. [Ηλεκτρονικό]. Available: <https://news.mit.edu/2019/mit-mini-cheetah-first-four-legged-robot-to-backflip-0304>.
- [3] A. Evan, «Six legged robot oneups nature with faster gait,» IEEE Spectrum, 21 02 2017. [Ηλεκτρονικό]. Available: <https://spectrum.ieee.org/six-legged-robot-oneups-nature-with-faster-gait>.
- [4] J. Horvath, Mastering 3D Printing, Berkeley, CA: Apress, 2014.
- [5] S. Torta και J. Torta, 3D printing: an introduction, Dulles, Virginia: Mercury Learning and Information, 2019.
- [6] Wanhao 3D Printer User Manual, [Ηλεκτρονικό]. Available: [https://www.wanhao3dprinter.com/xiazai/D6\\_User\\_Manual\\_REV\\_A.pdf](https://www.wanhao3dprinter.com/xiazai/D6_User_Manual_REV_A.pdf).
- [7] Σ. Καζαρής, «Σημειώσεις του μαθήματος "Ενσωματωμένα Συστήματα",» 2023.
- [8] Arduino Docs | Arduino Documentation, [Ηλεκτρονικό]. Available: <https://docs.arduino.cc/>.
- [9] «Arduino Sensor Shield V5.0 User Manual,» Arduino Forum, [Ηλεκτρονικό]. Available: <https://forum.arduino.cc/t/arduino-sensor-shield-v5-apc220-manual/223457>.
- [10] «Micro bit Lesson — Using the Ultrasonic Module,» Osoyoo, 2018. [Ηλεκτρονικό]. Available: <https://osoyoo.com/2018/09/18/micro-bit-lesson-using-the-ultrasonic-module/>.
- [11] Arduino Stack Exchange, [Ηλεκτρονικό]. Available: <https://arduino.stackexchange.com/questions/60134/why-a-servo-doesnt-move-to-angles-properly>.
- [12] «Spot,» Boston Dynamics, [Ηλεκτρονικό]. Available: <https://bostondynamics.com/products/spot/>.
- [13] ANYbotics, ANYbotics: Autonomous robotic inspection solutions, [Ηλεκτρονικό]. Available: <https://www.anybotics.com/robotics/anymal/>.
- [14] J. Chu, «“Blind” Cheetah 3 robot can climb stairs littered with obstacles,» MIT News | Massachusetts Institute of Technology, 04 07 2018. [Ηλεκτρονικό]. Available: <https://news.mit.edu/2018/blind-cheetah-robot-climb-stairs-obstacles-disaster-zones-0705>.
- [15] Aaedmusa, «3D Printed Robot Dog,» Instructables, [Ηλεκτρονικό]. Available: <https://www.instructables.com/3D-Printed-Robot-Dog/>.
- [16] S. Coddington, «GorillaBot the 3D Printed Arduino Autonomous Sprint Quadruped Robot,» Instructables, 2021. [Ηλεκτρονικό]. Available: <https://www.instructables.com/GorillaBot-the-3D-Printed-Arduino-Autonomous-Sprin/>.
- [17] Fritzing Software, [Ηλεκτρονικό]. Available: <https://fritzing.org/>.

# Παραρτήματα

## Κώδικας Arduino

```
// Βιβλιοθήκη που χρειάζεται για τα servo
#include <Servo.h>

// Καθορισμός των pin για Trigger και Echo
const int trigPin = 12;
const int echoPin = 13;

// Servo που χρησιμοποιούνται
Servo servo_8;
Servo servo_7;
Servo servo_6;
Servo servo_5;
Servo servo_4;
Servo servo_3;
Servo servo_2;
Servo servo_1;

void setup()
{
  // Καθορισμός των pin για τα servo
  servo_8.attach(9);      //FRONT_BASE FRONT LEFT
  servo_7.attach(8);      //FRONT_BASE BACK LEFT
  servo_6.attach(7);      //FRONT_BASE FRONT RIGHT
  servo_5.attach(6);      //FRONT_BASE BACK RIGHT
  servo_4.attach(5);      //BACKF_BASE FRONT LEFT
  servo_3.attach(4);      //BACK_BASE BACK LEFT
  servo_2.attach(3);      //BACK_BASE FRONT RIGHT
  servo_1.attach(2);      //BACK_BASE BACK RIGHT

  // Ρύθμιση των servo στην αρχική θέση (home position)
  servo_8.write(180);
  servo_7.write(0);
  servo_6.write(0);
  servo_5.write(180);
  servo_4.write(180);
  servo_3.write(0);
  servo_2.write(0);
  servo_1.write(180);

  delay(2000);
}
```

```

// Αποσύνδεση των servo για να αποφευχθεί το τρέμουλο κατά την
ενεργοποίηση
servo_8.detach();
servo_7.detach();
servo_6.detach();
servo_5.detach();
servo_4.detach();
servo_3.detach();
servo_2.detach();
servo_1.detach();

// Αρχικοποίηση σειριακής επικοινωνίας
Serial.begin(9600);

// Ορισμός του pin 12 ως έξοδος
pinMode(trigPin, OUTPUT);
// Ορισμός του pin 13 ως είσοδος
pinMode(echoPin, INPUT);

// Ρουτίνα εκκίνησης (Demo)
RobotStop();
RobotSitBack();
delay(1000);
RobotStop();
RobotSitForward();
delay(1000);
RobotStop();
delay(1000);
Robotlie();
delay(1000);
}

void loop()
{
// Μεταβλητές για τις αποστάσεις
int distanceLeft = 0; // Απόσταση που παίρνει όταν στρίβει αριστερά
long distance = Read_Distance(); // Απόσταση όταν προχωράει ευθεία

RobotForward();

if (distance > 0)
{
if (distance < 30)
{
RobotStop();
delay(100);
RobotBackward();
// Επιτόπου στροφή προς τα αριστερά 4 φορές

```



```

    for (int i = 0; i < 4; i++)
    {
        RobotLeft();
    }
    delay(100);
    distanceLeft = Read_Distance();
    if (distanceLeft > 0)
    {
        if (distanceLeft < 30)
        {
            // Επιτόπου στροφή προς τα δεξιά 10 φορές
            for (int j = 0; j < 9; j++)
            {
                RobotRight();
            }
        }
    }
}
}
}
}
}

```

```

//Υπορουτίνα ανάγνωσης απόστασης
long Read_Distance()
{
    long duration, cm;
    // Αποστέλεται ένας παλμός HIGH διάρκειας 10 microseconds
    // Παρέχεται ένας σύντομος παλμός LOW για να αποσταλεί ένας καθαρός
    παλμός HIGH
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    // Ανάγνωση του σήματος από τον αισθητήρα
    duration = pulseIn(echoPin, HIGH,30000);
    // Επαναφορά της στάθμης του echoPin αν λαμβάνεται τιμή 0
    if (duration==0)
    {
        pinMode(echoPin, OUTPUT);
        delay(10);
        digitalWrite(echoPin, LOW);
        delay(10);
        pinMode(echoPin, INPUT);
        delay(10);
    }
    // Μετατροπή του χρόνου σε απόσταση
    cm = (duration / 58);
    return cm;
}

```

```

}

// Υπορουτίνα σταματημού του ρομπότ
void RobotStop()
{
  servo_8.write(170);
  servo_7.write(10);
  servo_6.write(10);
  servo_5.write(170);
  servo_4.write(170);
  servo_3.write(10);
  servo_2.write(10);
  servo_1.write(170);

  delay(400);

  // Αποσύνδεση των servo
  servo_8.detach();
  servo_7.detach();
  servo_6.detach();
  servo_5.detach();
  servo_4.detach();
  servo_3.detach();
  servo_2.detach();
  servo_1.detach();
}

// Υπορουτίνα κίνησης προς τα εμπρός
void RobotForward()
{
  // Καθορισμός των pin για τα servo
  servo_8.attach(9);
  servo_7.attach(8);
  servo_6.attach(7);
  servo_5.attach(6);
  servo_4.attach(5);
  servo_3.attach(4);
  servo_2.attach(3);
  servo_1.attach(2);

  servo_8.write(135);
  servo_7.write(90);
  servo_6.write(0);
  servo_5.write(135);
  servo_4.write(180);
  servo_3.write(45);
  servo_2.write(45);
  servo_1.write(90);
}

```

```
delay(100);

servo_8.write(90);
servo_7.write(0);
servo_6.write(0);
servo_5.write(135);
servo_4.write(180);
servo_3.write(45);
servo_2.write(90);
servo_1.write(180);

delay(100);

servo_8.write(120);
servo_7.write(0);
servo_6.write(0);
servo_5.write(135);
servo_4.write(180);
servo_3.write(45);
servo_2.write(60);
servo_1.write(180);

delay(100);

servo_8.write(180);
servo_7.write(45);
servo_6.write(45);
servo_5.write(90);
servo_4.write(135);
servo_3.write(90);
servo_2.write(0);
servo_1.write(135);

delay(100);

servo_8.write(180);
servo_7.write(45);
servo_6.write(90);
servo_5.write(180);
servo_4.write(45);
servo_3.write(0);
servo_2.write(0);
servo_1.write(135);

delay(100);

servo_8.write(180);
```

```

servo_7.write(45);
servo_6.write(60);
servo_5.write(180);
servo_4.write(120);
servo_3.write(0);
servo_2.write(0);
servo_1.write(135);

delay(100);
}

// Υπορουτίνα κίνησης προς τα πίσω
void RobotBackward()
{
  // Καθορισμός των pin για τα servo
  servo_8.attach(9);
  servo_7.attach(8);
  servo_6.attach(7);
  servo_5.attach(6);
  servo_4.attach(5);
  servo_3.attach(4);
  servo_2.attach(3);
  servo_1.attach(2);

  servo_8.write(90);
  servo_7.write(45);
  servo_6.write(45);
  servo_5.write(180);
  servo_4.write(135);
  servo_3.write(0);
  servo_2.write(90);
  servo_1.write(135);

  delay(100);

  servo_8.write(180);
  servo_7.write(45);
  servo_6.write(45);
  servo_5.write(180);
  servo_4.write(135);
  servo_3.write(0);
  servo_2.write(0);
  servo_1.write(90);

  delay(100);

  servo_8.write(180);
  servo_7.write(60);

```

```

servo_6.write(45);
servo_5.write(180);
servo_4.write(135);
servo_3.write(0);
servo_2.write(0);
servo_1.write(120);

delay(100);

servo_8.write(135);
servo_7.write(0);
servo_6.write(90);
servo_5.write(135);
servo_4.write(90);
servo_3.write(45);
servo_2.write(45);
servo_1.write(180);

delay(100);

servo_8.write(135);
servo_7.write(0);
servo_6.write(0);
servo_5.write(90);
servo_4.write(180);
servo_3.write(90);
servo_2.write(45);
servo_1.write(180);

delay(100);

servo_8.write(135);
servo_7.write(0);
servo_6.write(0);
servo_5.write(120);
servo_4.write(180);
servo_3.write(60);
servo_2.write(45);
servo_1.write(180);

delay(100);
}

// Υπορουτίνα στροφής προς τα αριστερά
void RobotLeft()
{
    // Καθορισμός των pin για τα servo

```

```
servo_8.attach(9);  
servo_7.attach(8);  
servo_6.attach(7);  
servo_5.attach(6);  
servo_4.attach(5);  
servo_3.attach(4);  
servo_2.attach(3);  
servo_1.attach(2);
```

```
servo_6.write(0);  
servo_5.write(135);  
servo_2.write(45);  
servo_1.write(90);  
servo_8.write(90);  
servo_7.write(45);  
servo_4.write(135);  
servo_3.write(0);
```

```
delay(100);
```

```
servo_6.write(0);  
servo_5.write(135);  
servo_2.write(90);  
servo_1.write(180);  
servo_8.write(180);  
servo_7.write(90);  
servo_4.write(135);  
servo_3.write(0);
```

```
delay(100);
```

```
servo_6.write(0);  
servo_5.write(135);  
servo_2.write(60);  
servo_1.write(180);  
servo_8.write(180);  
servo_7.write(60);  
servo_4.write(135);  
servo_3.write(0);
```

```
delay(100);
```

```
servo_6.write(45);  
servo_5.write(90);  
servo_2.write(0);  
servo_1.write(135);  
servo_8.write(135);  
servo_7.write(0);
```

```

servo_4.write(90);
servo_3.write(45);

delay(100);

servo_6.write(90);
servo_5.write(180);
servo_2.write(0);
servo_1.write(135);
servo_8.write(135);
servo_7.write(0);
servo_4.write(180);
servo_3.write(90);

delay(100);

servo_6.write(60);
servo_5.write(180);
servo_2.write(0);
servo_1.write(135);
servo_8.write(135);
servo_7.write(0);
servo_4.write(180);
servo_3.write(60);

delay(100);
}

// Υπορουτίνα στροφής προς τα δεξιά
void RobotRight()
{
  // Καθορισμός των pin για τα servo
  servo_8.attach(9);
  servo_7.attach(8);
  servo_6.attach(7);
  servo_5.attach(6);
  servo_4.attach(5);
  servo_3.attach(4);
  servo_2.attach(3);
  servo_1.attach(2);

  servo_8.write(135);
  servo_7.write(90);
  servo_4.write(180);
  servo_3.write(45);
  servo_6.write(45);
  servo_5.write(180);
  servo_2.write(90);
}

```

```
servo_1.write(135);
```

```
delay(100);
```

```
servo_8.write(90);  
servo_7.write(0);  
servo_4.write(180);  
servo_3.write(45);  
servo_6.write(45);  
servo_5.write(180);  
servo_2.write(0);  
servo_1.write(90);
```

```
delay(100);
```

```
servo_8.write(120);  
servo_7.write(0);  
servo_4.write(180);  
servo_3.write(45);  
servo_6.write(45);  
servo_5.write(180);  
servo_2.write(0);  
servo_1.write(120);
```

```
delay(100);
```

```
servo_8.write(180);  
servo_7.write(45);  
servo_4.write(135);  
servo_3.write(90);  
servo_6.write(90);  
servo_5.write(135);  
servo_2.write(45);  
servo_1.write(180);
```

```
delay(100);
```

```
servo_8.write(180);  
servo_7.write(45);  
servo_4.write(90);  
servo_3.write(0);  
servo_6.write(0);  
servo_5.write(90);  
servo_2.write(45);  
servo_1.write(180);
```

```
delay(100);
```



```

servo_8.write(180);
servo_7.write(45);
servo_4.write(120);
servo_3.write(0);
servo_6.write(0);
servo_5.write(120);
servo_2.write(45);
servo_1.write(180);

delay(100);
}

// Υπορουτίνα κίνησης που ξαπλώνει το ρομπότ
void Robotlie()
{
  // Καθορισμός των pin για τα servo
  servo_8.attach(9);
  servo_7.attach(8);
  servo_6.attach(7);
  servo_5.attach(6);
  servo_4.attach(5);
  servo_3.attach(4);
  servo_2.attach(3);
  servo_1.attach(2);

  int pos1 = 170;
  int pos2 = 10;

  // Εκτελείται η for 80 φορές, δηλαδή 80 μοίρες μέχρι να φτάσει
  // τις 90 σε όλα τα πόδια και ξαπλώνει σταδιακά
  for (int i=0; i<=80; i++)
  {
    servo_8.write(pos1);
    servo_5.write(pos1);
    servo_4.write(pos1);
    servo_1.write(pos1);
    pos1=pos1-1;

    servo_7.write(pos2);
    servo_6.write(pos2);
    servo_3.write(pos2);
    servo_2.write(pos2);
    pos2=pos2+1;

    delay(10);
  }

  delay(1000);
}

```

```

// Σηκώνεται σταδιακά
for (int i=0; i<=80; i++)
{
    servo_8.write(pos1);
    servo_5.write(pos1);
    servo_4.write(pos1);
    servo_1.write(pos1);
    pos1=pos1+1;

    servo_7.write(pos2);
    servo_6.write(pos2);
    servo_3.write(pos2);
    servo_2.write(pos2);
    pos2=pos2-1;

    delay(10);
}
}

// Υπορουτίνα κίνησης που κάθεται στα πίσω πόδια
void RobotSitBack()
{
    // Καθορισμός των pin για τα servo
    servo_8.attach(9);
    servo_7.attach(8);
    servo_6.attach(7);
    servo_5.attach(6);
    servo_4.attach(5);
    servo_3.attach(4);
    servo_2.attach(3);
    servo_1.attach(2);

    // Τα μπροστά πόδια σε όρθια θέση
    servo_8.write(180);
    servo_7.write(0);
    servo_6.write(0);
    servo_5.write(180);

    int pos1 = 170;
    int pos2 = 10;

    // Τα πίσω πόδια κάθονται σταδιακά
    for(int i=0; i<=125 ; i++)
    {
        servo_4.write(pos1);
        servo_1.write(pos1);
        pos1=pos1-1;
    }
}

```

```

servo_3.write(pos2);
servo_2.write(pos2);
pos2=pos2+1;

delay(10);
}
}

// Υπορουτίνα κίνησης που κάθεται στα μπροστά πόδια
void RobotSitForward()
{
// Καθορισμός των pin για τα servo
servo_8.attach(9);
servo_7.attach(8);
servo_6.attach(7);
servo_5.attach(6);
servo_4.attach(5);
servo_3.attach(4);
servo_2.attach(3);
servo_1.attach(2);

// Τα πίσω πόδια σε όρθια θέση
servo_4.write(180);
servo_3.write(0);
servo_2.write(0);
servo_1.write(180);

int pos1 = 170;
int pos2 = 10;

// Τα μπροστά πόδια κάθονται σταδιακά
for(int i=0; i<=125 ; i++)
{
servo_8.write(pos1);
servo_5.write(pos1);
pos1=pos1-1;

servo_7.write(pos2);
servo_6.write(pos2);
pos2=pos2+1;

delay(10);
}
}

```