

Πίνακας περιεχομένων

Contents

Η κατασκευή	2
Source code.....	3
Hardware	3
Τα μαθηματικά πίσω από τη κατασκευή.....	4
Project Fork.....	4
Χρήση/ Manual	6

Christianidis Vasileios

basilisvirus@hotmail.com

Η κατασκευή

2. ΧΩΡΙΚΟΣ ΒΡΑΧΙΟΝΑΣ 3ΒΕ

Αντικείμενο

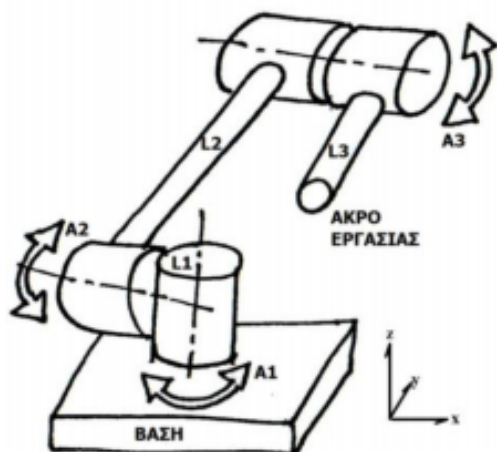
Αφορά την κατασκευή και επίδειξη ρομποτικού βραχίονα μικρού μεγέθους, με τρεις βαθμούς ελευθερίας όπως στο Σχήμα. Οι αρθρώσεις ενεργοποιούνται με κινητήρες τύπου "micro-servo" (ελαφρού τύπου / περιορισμένης γωνίας) υπό τον έλεγχο συστήματος μικροελεγκτή. Ενδεικτικά, το μήκος των συνδέσμων (L_2 , L_3) είναι 6-10cm.

Προδιαγραφή

Σκοπός της λειτουργίας του ρομποτικού συστήματος είναι η τοποθέτηση του άκρου εργασίας σε οποιαδήποτε θέση στο χώρο εργασίας όταν οι αντίστοιχες συντεταγμένες διαβιβάζονται σειριακά από τον υπολογιστή στο σύστημα μικροελεγκτή. Σε περίπτωση που οι συντεταγμένες βρίσκονται εκτός του χώρου εργασίας, αγνοούνται και παράγεται μια φωτεινή ένδειξη

Πρόσθετη προδιαγραφή

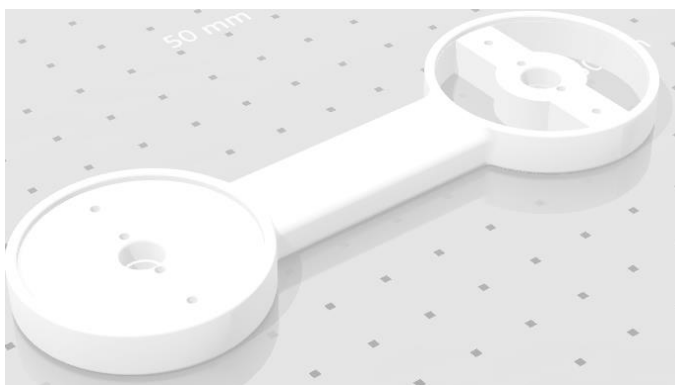
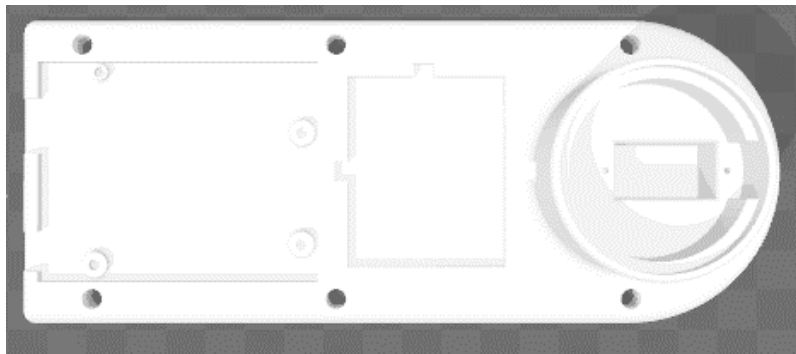
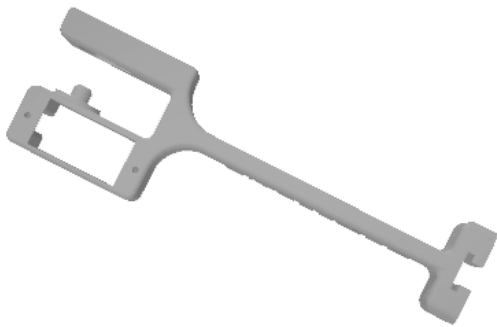
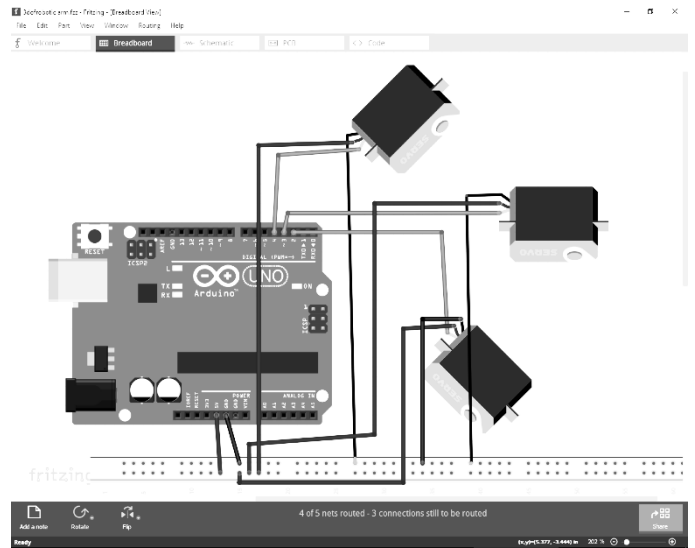
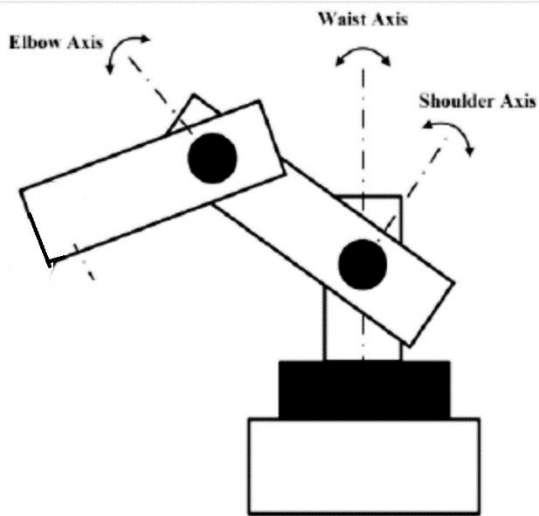
Υπολογισμός και εφαρμογή ομαλής τροχιάς (χωρίς γωνίες) η οποία να διέρχεται από τρία επιλεγόμενα σημεία, των οποίων οι συντεταγμένες διαβιβάζονται σειριακά από τον υπολογιστή στο σύστημα μικροελεγκτή.



Source code

Ο πηγαίως κώδικας: https://drive.google.com/open?id=1Mwbe595RzY48CXIqrH_VfTmWk9Hwk52V

Hardware



Τα αρχεία για τα 3d parts μπορούν να βρεθούν εδώ:

https://drive.google.com/open?id=1dwAuN0cc92pcW_Hri_FHhAWubaCeS61t

Τα οποία δε τα κατασκεύασα, αλλά τα βρήκα εδώ: <http://3dprintedrobots.blogspot.com/p/project-links.html?m=0> και είναι compatible με servo μεγέθους: 9G SG90 και MG996R (micro servos)

καποια από τα 3d printed parts είναι 10 φορές μικρότερα από το κανονικό, οπότε θα χρειαστεί να γίνει αύξηση κατά 1000% αυτών των μελών.

Το αρχείο του fritzing το ανέβασα εδώ: https://drive.google.com/open?id=1pe4eR5G9d9OuKV1X6T0yLQ4_ZlyApwpt

Τα μαθηματικά πίσω από τη κατασκευή

Βρήκα ένα pdf σε αυτή τη σελίδα:

http://www.academia.edu/9165706/Forward_and_inverse_Kinematics_complete_solutions_3DOF_good_reference_for_CrustCrawler_Smart_Arm_Users

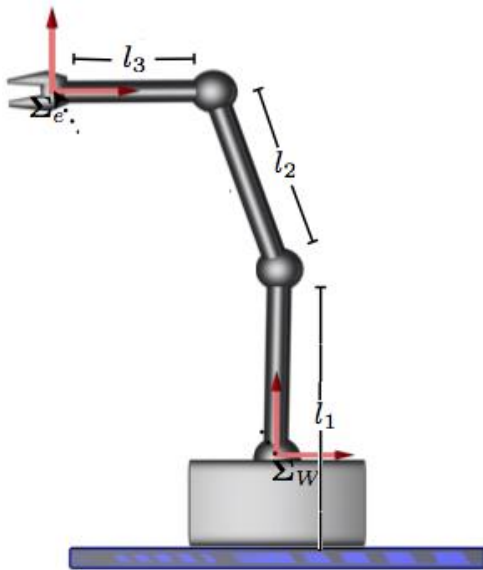
Το οποίο λύνει την αντιστροφή κινηματική για έναν 3DOF βραχίονα (3BE), στη σελίδα 36 (το s3 παίρνει και αρνητική τιμή για τη 2^η λύση αν υπάρχει).

Project Fork

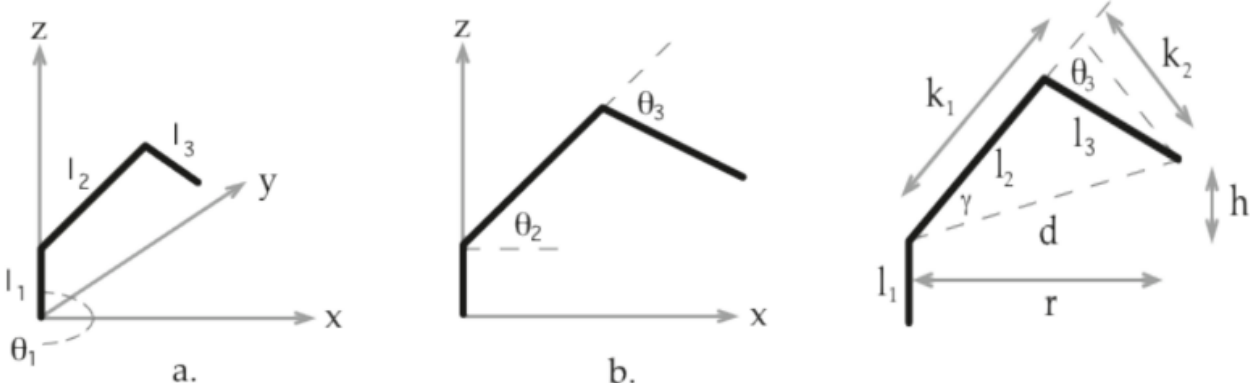
Σε περίπτωση που προσπαθήσετε να αναπαράγετε τη κατασκευή, θα πρέπει να αλλάξετε τον κωδικά ανάλογα με τον τρόπο που τοποθετήσατε τα servo σας, θα αλλάζουν οι οριακές τιμές των γωνιών και αλλα, όπως το μέγεθος των βραχιονων.

Λεπτομερή οι αλλαγές:

---Στη 16^η γραμμή του κώδικα, τα l1, l2, l3 αντιστοιχούν στα παρακάτω μήκη:

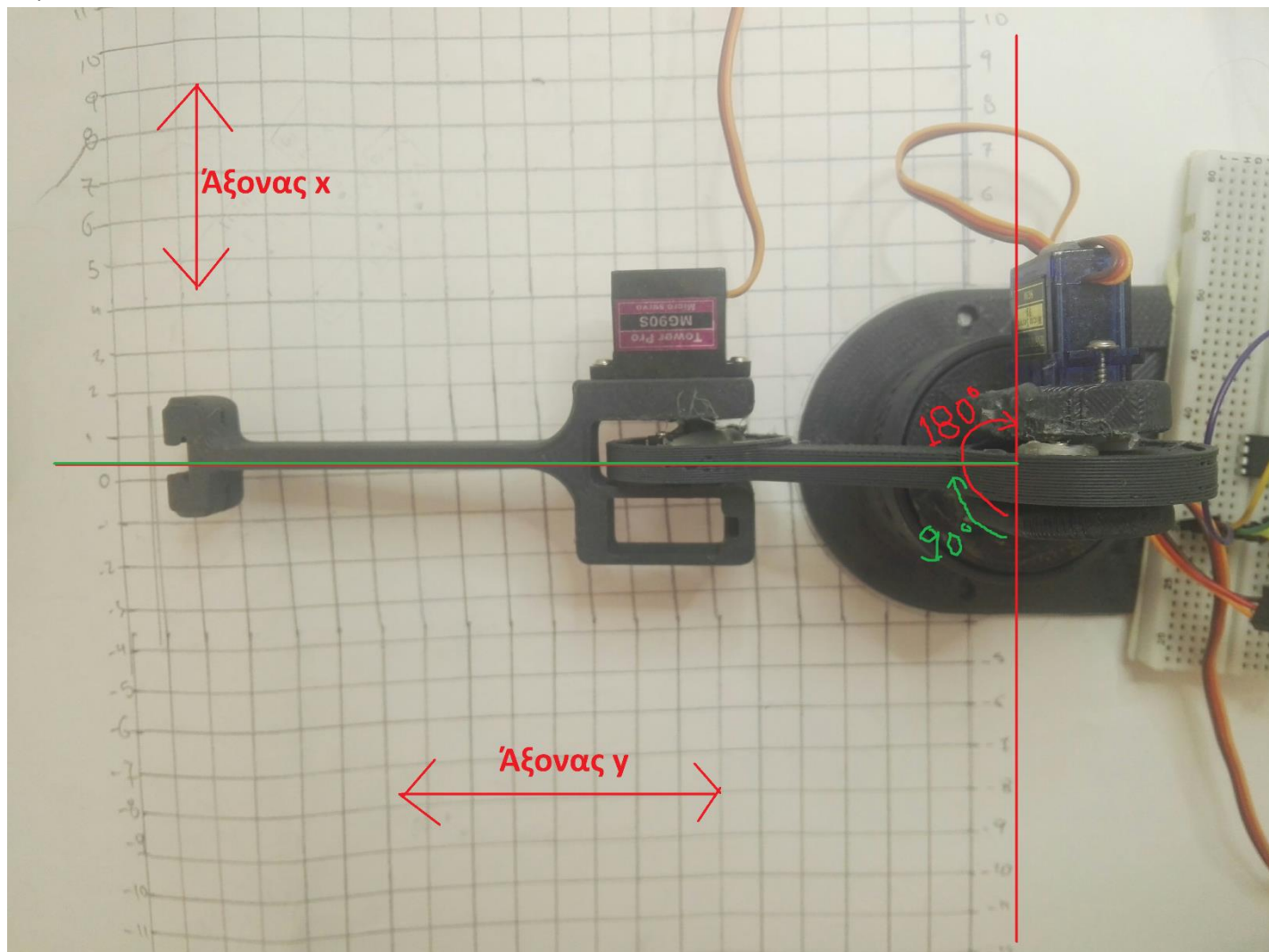


---Οι αρχικές γωνίες που αποκτάνε τα servo λόγω της βιβλιοθήκης servo.h είναι οι 90° (γωνίες servo). Αυτές τις γωνίες παίρνουν όλα τα servo τη στιγμή που χρησιμοποιείται η μέθοδος `servo.attach(pin value)` στην void `setup()`, σε οποιοδήποτε pin και αν γίνει το attach. Όμως, στη πραγματικότητα, το πρόγραμμά μας, επειδή χρησιμοποιεί ως γωνίες τις προκαθορισμένες γωνίες που έχουμε θέσει και χρησιμοποιούμε από τις μαθηματικές μας εξισώσεις, (βλεπε εικόνα παρακάτω),



Υπάρχει πιθανότητα κάποιες ή όλες οι θεωρητικές μας γωνίες να μην ανταποκρίνονται στις φυσικές μας γωνίες. Ως φυσικές γωνίες εννοούμε αυτές που πραγματικά θέλουμε και βλέπουμε στον χτισμένο βραχίονά μας και ανταποκρίνονται στη παραπάνω εικόνα. Ενώ ως θεωρητικές γωνίες, εννοούμε κάποιες άλλες γωνίες, οι οποίες χρησιμοποιεί το πρόγραμμά μας, οι οποίες συσχετίζονται με τις φυσικές μας γωνίες με μια εξίσωση, την οποία θα πρέπει και να βρούμε. Για αυτό θα χρησιμοποιήσουμε manual calibration.

Η τεχνική εδώ της χειροκίνητης βαθμονόμησης θα είναι πολύ πρόχειρη. Θα πρέπει να φτιάξετε ένα νέο πρόγραμμα στο Arduino ide, χωρίς να αλλάξετε τη κατασκευή φυσικά, και χρησιμοποιώντας την έτοιμη βιβλιοθήκη Servo.h, να δίνετε δικές σας γωνίες σε μια μια ξεχωριστά άρθρωση/servo και να δείτε αν αυτές οι γωνίες που δίνετε αντικατοπτρίζει/ανταποκρίνεται/είναι ίσες με τις φυσικές γωνίες, όχι μόνο στη τιμή αλλά και τη κατεύθυνση των γωνιών. τα θ2 και θ3 έχουν θετική κατεύθυνση προς τα επάνω (αριστερόστροφα), ενώ το θ1 μπορεί να έχει είτε δεξιόστροφη είτε αριστερόστροφη κίνηση. Το θ1 δε θα χρειαστεί καθόλου ρύθμιση, εάν τοποθετήσουμε τη βάση του βραχίονα έτσι ώστε όταν το servo να δείχνει 90° οι βραχίονες να είναι στην ευθεία με τον γ άξονα όπως παρακάτω:



Αν είμαστε ευχαριστημένοι με τη βάση, φτιάχνουμε την πρώτη μας εξίσωση, στη περίπτωση μου:

Για τη θ1:

$$TH1(Deg) = Deg$$

Για τη θ2:

$$TH2(Deg) = (-Deg) + 140$$

Για τη θ3:

$$TH3(Deg) = Deg + 125$$

Στις γραμμές κώδικα 25, 29 και 33 θα βάλετε τις δικές σας συναρτήσεις.

---Η τελευταία αλλαγή πρέπει να γίνει στους περιορισμούς γωνιών. ο βραχίονάς μας δεν μπορεί να πάρει όλες τις διαθέσιμες γωνίες που του επιτρέπουν οι μαθηματικές εξισώσεις. Εδώ ευτυχώς δεν χρειάζεται να φτιάξουμε μια εξίσωση που να μετατρέπει τα φυσικά όρια στα μαθηματικά όρια, αλλά πάλι με δικό μας calibration όπως έκανα

και φένεται στις παρακάτω εικόνες (αν ανοίξετε το link), να εκτιμήσουμε με όσο μεγαλύτερη ακρίβεια μπορούμε το range των γωνιών που θα είναι επιτρεπτό να δωθεί στον βραχίονα.

https://drive.google.com/open?id=1MX1EK6R8p5shqi_fzJayERDFiYCHO3f8

Συνολικά θα χρειαστούμε 4 όρια γωνιών. Τα όρια που θα βρείτε από τον παραπάνω τρόπο μέσω των εικόνων δεν χρειάζεται να τους αλλάξετε στις θεωρητικές τιμές, διότι όταν ζητηθεί στον βραχίονα να πάει σε ένα σημείο, πρώτα υπολογίζονται μόνο τα μαθηματικά κομμάτια (οι γωνίες που πρέπει να πάρει ο βραχίονας σε φυσικό επίπεδο) και ύστερα γίνεται η σύγκριση με τα όρια γωνιών. Και ύστερα χρησιμεύουν οι εξισώσεις που μετατρέπουν τις φυσικές γωνίες σε θεωρητικές, διότι ύστερα μπαίνουν τα servo στην όλη κατάσταση. Για τον απλο υπολογισμό του που μπορεί και που δε μπορεί να πανε τα μελη του βραχίονα δεν χρειάζονται οι εξισώσεις μετατροπής. Έτσι, ότι βρούμε στα όρια μας από τις εικόνες, είναι και αυτό που θα βάλουμε.

Τα όρια γωνιών μπορούμε να τα αλλάξουμε στη γραμμή 38 έως 41 του κώδικα.

Χρήση/ Manual

Για να χρησιμοποιηθεί ο βραχίονας, θα πρέπει να κατεβάσετε το Arduino IDE και να συνδέσετε το hardware σύμφωνα με το schematic πιο πάνω (google drive link). Ύστερα, αφού ακολουθήσετε τις οδηγίες για το “project fork” ανοίξετε το αρχείο απο το φακέλο 3Dof_final στο link, θα παρατηρήσετε στις τελευταίες γραμμές κωδικα:

```
void setup() {  
  //declare which pin are my servos attached  
  Ser_1.attach(2);  
  Ser_2.attach(3);  
  Ser_3.attach(4);  
  
  delay(3000);  
  //Start serial port  
  Serial.begin(9600);  
  Inverse Calc(0,6,2);  
}  
  
void loop() {  
  
  //add a delay  
  delay(1000);  
}
```

Την Inverse_Calc(). Για να λειτουργήσει ο βραχίονας, και να παει στο στόχο το άκρο εργασίας του, πρέπει να βάλετε τις συντεταγμένες x,y,z στην inverse calc με αυτη τη σειρα και να πατήσετε το compile & upload sketch στο Arduino.

Επίσης θα παρατηρήσετε οτι με το που φορτωσει ο κώδικας, θα κάνει μια απότομη κίνηση στις 90 μοίρες (προ-προγραμματισμού) και στα 3 σερβο). Και ύστερα θα ξεκινήσει να κινείται προς το σημείο που ορίστηκε. Καλό είναι να αποφεύγεται εκείνη η απότομη κίνηση, βάζοντας χειροκίνητα τον βραχίονα στο αρχικό σημείο που θα κινόταν επότομα με το φόρτωμα του προγράμματος.

Μπορείτε να τρέξετε την Inverse_Calc() παραπάνω από μία φορές σε κάθε upload, με αποτέλεσμα να πηγαίνει με τη σειρά στα σημεία που θέσαμε.

Σημείωση: το Y δε δέχεται αρνητικές τιμές* και το Z μπορεί να δεχτεί αρνητικές τιμές.