

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΣΕΡΡΩΝ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

**Υλοποίηση ρομποτικού βραχίονα με χρήση της
πλατφόρμα Arduino.**

Πτυχιακή Εργασία

Σέφη Τατιανή Α.Μ. 1255

Μπανάβου Νίκη Α.Μ. 2387

Επιβλέπων :Δρ. Σταύρος Βολογιαννίδης

ΣΕΡΡΕΣ, ΙΟΥΝΙΟΣ 2013

Υπεύθυνη Δήλωση

Βεβαιώνουμε ότι είμαστε συγγραφείς αυτής της πτυχιακής εργασίας και ότι κάθε υποστήριξη ή βοήθεια την οποία είχαμε για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται μέσα σε αυτήν. Επίσης, σε αυτή την πτυχιακή εργασία αναφέρονται όλες οι πηγές από τις οποίες κάναμε χρήση δεδομένων και πληροφοριών. Τέλος, βεβαιώνουμε ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμάς προσωπικά και ειδικά για τις αναγκαίες απαιτήσεις του προγράμματος σπουδών του Τμήματος Πληροφορικής & Επικοινωνιών του Τεχνολογικού Εκπαιδευτικού Ιδρύματος Σερρών.

Περιεχόμενα

Περίληψη.....	5
ΚΕΦΑΛΑΙΟ 1 ^ο	6
1.ΕΙΣΑΓΩΓΗ	6
1.1 Τι είναι το Arduino.....	6
1.2 Υλικό πλατφόρμας Arduino Uno.....	7
1.2.1 Μνήμη	8
1.2.2 Είσοδοι – Έξοδοι.....	9
1.2.3 Τροφοδοσία.....	11
1.3 Ολοκληρωμένο περιβάλλον ανάπτυξης.....	12
ΚΕΦΑΛΑΙΟ 2 ^ο	14
2. ΡΟΜΠΟΤΙΚΟΙ ΒΡΑΧΙΟΝΕΣ.....	14
2.1. Ρομποτική.....	14
2.1.1 Ιστορική Αναδρομή	16
2.1.2 Τομείς Εφαρμογής.....	17
2.2 Χαρακτηριστικά των ρομπότ.....	18
2.3 Δομή των Ρομπότ	19
2.4 Τύποι ρομποτικών βραχιόνων και χώρος εργασίας	21
2.4.1 Ρομπότ καρτεσιανών συντεταγμένων	23
2.4.2 Ρομπότ κυλινδρικών συντεταγμένων	23
2.4.3 Ρομπότ σφαιρικών συντεταγμένων	23
2.4.4 Αρθρωτό ρομπότ.....	24
2.5 Αρθρώσεις (Joints)	24
2.5.1 Είδη αρθρώσεων	24
2.5.2 Αρθρώσεις σε ρομποτικούς βραχίονες.....	25
2.6 Βαθμοί ελευθερίας.....	25
ΚΕΦΑΛΑΙΟ 3 ^ο	27
3.ΚΙΝΗΜΑΤΙΚΗ ΘΕΩΡΙΑ	27
3.1 Εισαγωγή στην κινηματική θεωρία.....	27
3.2 Ευθύ κινηματικό πρόβλημα	28
3.2.1 Αλγόριθμος Denavit-Hartenberg.....	29
3.3 Αντίστροφο κινηματικό πρόβλημα	31

ΚΕΦΑΛΑΙΟ 4 ^ο	38
4.ΣΧΕΔΙΑΣΜΟΣ – ΜΕΛΕΤΗ ΚΑΙ ΒΗΜΑΤΑ ΚΑΤΑΣΚΕΥΗΣ ΤΟΥ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ	38
4. 1 Εισαγωγή	38
4.2 Σχέδια κατασκευής του βραχίονα.....	39
4.3 Μελέτη και βήματα.....	42
ΚΕΦΑΛΑΙΟ 5 ^ο	48
5.ΥΛΟΠΟΙΗΣΗ ΠΕΙΡΑΜΑΤΩΝ ΣΕ ΓΛΩΣΣΑ WIRING.....	48
5.1 Υλοποίηση 1 ^{ου} Πειράματος-Χειρισμός του βραχίονα μέσω Η/Υ.....	48
5.2 Υλοποίηση 2 ^{ου} Πειράματος – Χειρισμός βραχίονα με κλειστή διαδρομή κίνησης.....	52
5.3 Ανάπτυξη γραφικού περιβάλλοντος διεπαφής για την μελέτη του ευθέους κινηματικού προβλήματος στο περιβάλλον του matlab	63
5.4 Ανάπτυξη γραφικού περιβάλλοντος διεπαφής για την μελέτη του αντίστροφου κινηματικού προβλήματος στο περιβάλλον του matlab	64
ΚΕΦΑΛΑΙΟ 6 ^ο	66
6.1 Σχόλια-Συμπεράσματα.....	66

Περίληψη

Στην παρούσα πτυχιακή εργασία πραγματοποιείται η σχεδίαση και υλοποίηση ενός ρομποτικού βραχίονα τριών περιστροφικών αρθρώσεων, καθώς επίσης ο προγραμματισμός και ο έλεγχος του, μέσω της πλατφόρμας του Arduino Uno. Ο ρομποτικός βραχίονας έχει προγραμματιστεί σε γλώσσα Wiring και η εξομοίωση του έγινε στο περιβάλλον του Matlab. Βασικό στόχο αποτελεί η δημιουργία προγραμμάτων, τα οποία θέτουν τον ρομποτικό βραχίονα σε εκτέλεση βασικών λειτουργιών και κινήσεων. Τέλος πραγματοποιήθηκε η παραμετροποίηση γραφικού περιβάλλοντος διεπαφής για την μελέτη του ευθέως και αντίστροφου κινηματικού προβλήματος σε περιβάλλον προσομοίωσης.

ΚΕΦΑΛΑΙΟ 1^ο

1.ΕΙΣΑΓΩΓΗ

1.1 Τι είναι το Arduino

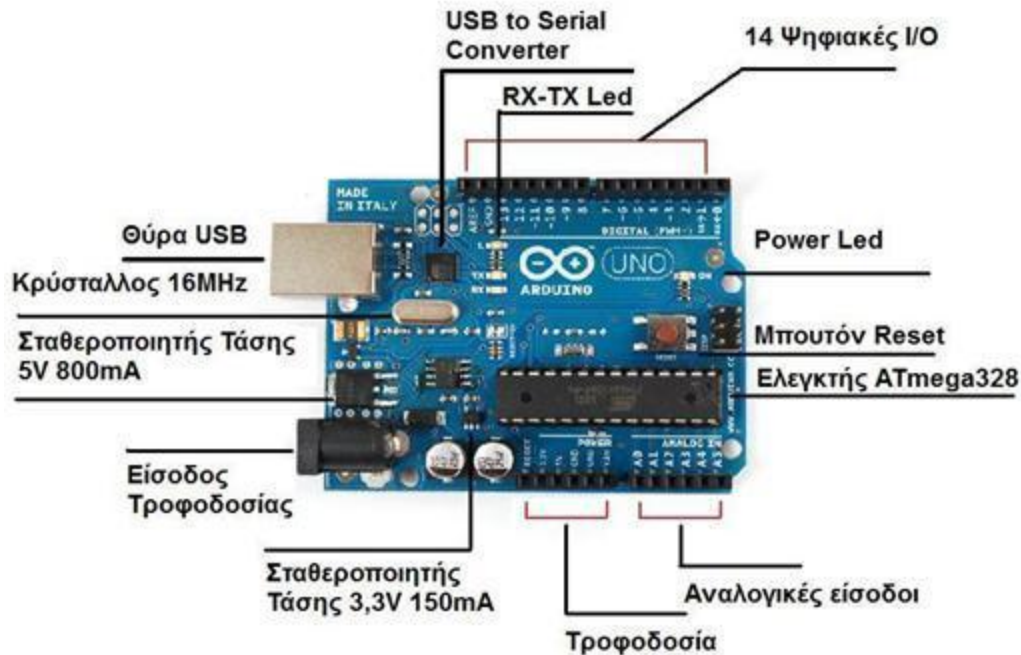
Το Arduino είναι μια υπολογιστική πλατφόρμα βασισμένη σε μια απλή μητρική πλακέτα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring .

Το arduino μπορεί να διαβάσει το περιβάλλον λαμβάνοντας σήματα από διάφορους αισθητήρες και μπορεί να το επηρεάσει ελέγχοντας φώτα, μηχανισμούς και άλλα. Ο μικροελεγκτής που βρίσκεται σε κάθε πλακέτα arduino μπορεί να προγραμματιστεί σε γλώσσα arduino programming language (βασισμένη στην Wiring) και το προγραμματιστικό περιβάλλον του arduino είναι βασισμένο στη γλώσσα Processing. Οι arduino εφαρμογές μπορούν να λειτουργούν μόνες τους (stand-alone) ή μπορούν να επικοινωνούν με διάφορα προγράμματα σε ένα υπολογιστή (π.χ Flash,Processing, MaxMSP).

Υπάρχουν διάφορες πλακέτες arduino βασισμένες σε διαφορετικούς μικροελεγκτές οι οποίες ανάλογα με τα χαρακτηριστικά έχουν και την ανάλογη τιμή. Το πρόγραμμα compiler του arduino είναι δωρεάν και διαθέσιμο από το επίσημο site (arduino.cc).

Επιπλέον, η γλώσσα προγραμματισμού, οι διάφορες βιβλιοθήκες και το ολοκληρωμένο περιβάλλον ανάπτυξης που υπάρχουν για τον προγραμματισμό της πλατφόρμας Arduino αποτελούν ανοιχτό λογισμικό προσφέροντας έτσι ανεκτίμητη γνώση σε όλους.

1.2 Υλικό πλατφόρμας Arduino Uno



Μία πλακέτα Arduino αποτελείται από έναν cmos 8 ψηφίων Atmel AVR μικροελεγκτή (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωση του σε άλλα κυκλώματα. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz. Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα bootloader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής.

Ο μικροελεγκτής σε κάθε πλατφόρμα Arduino είναι προγραμματισμένος και περιέχει έναν ειδικό φορτωτή εκκίνησης ο οποίος απλοποιεί κυρίως την διαδικασία της τοποθέτησης προγραμμάτων στην ενσωματωμένη μνήμη Flash της πλατφόρμας, με αποτέλεσμα να μην είναι απαραίτητη κάποια εξωτερική συσκευή προγραμματισμού μικροελεγκτών AVR. Οι πιο πρόσφατες πλατφόρμες Arduino μπορούν να προγραμματιστούν μέσω της θύρας USB που διαθέτουν, με την βοήθεια όμως ενός κυκλώματος μετατροπής από USB σε σειριακή όπως το FTDI FT232.

Γενικά όλες οι πλακέτες είναι προγραμματισμένες μέσω μιας σειριακής σύνδεσης RS-232, αλλά ο τρόπος με τον οποίο αυτό υλοποιείται ποικίλλει ανάλογα με την έκδοση. Οι σειριακές πλακέτες Arduino περιέχουν ένα απλό κύκλωμα αντιστροφής για την μετατροπή ανάμεσα στα σήματα των επιπέδων RS-232 και TTL.

1.2.1 Μνήμη

Ο μικροελεγκτής ATmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων:

- 2Kb μνήμης SRAM που είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματά μας για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. κατά το runtime. Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή αν γίνει reset.
- 1Kb μνήμης EEPROM η οποία μπορεί να χρησιμοποιηθεί για «ωμή» εγγραφή/ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματά μας κατά το runtime. Σε αντίθεση με την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset οπότε είναι το ανάλογο του σκληρού δίσκου.
- 32Kb μνήμης Flash, από τα οποία τα 2Kb χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware αυτό που στην ορολογία του Arduino ονομάζεται bootloader είναι αναγκαίο για την εγκατάσταση των δικών μας προγραμμάτων στον μικροελεγκτή μέσω της θύρας USB, χωρίς δηλαδή να χρειάζεται εξωτερικός hardware programmer. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή μας. Η μνήμη Flash, όπως και η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset.

1.2.2 Είσοδοι – Έξοδοι

Το Arduino διαθέτει σειριακό interface. Ο μικροελεγκτής ATmega υποστηρίζει σειριακή επικοινωνία, την οποία το Arduino προωθεί μέσα από έναν ελεγκτή Serial-over-USB ώστε να συνδέεται με τον υπολογιστή μέσω USB. Η σύνδεση αυτή χρησιμοποιείται για την μεταφορά των προγραμμάτων που σχεδιάζονται από τον υπολογιστή στο Arduino αλλά και για αμφίδρομη επικοινωνία του Arduino με τον υπολογιστή μέσα από το πρόγραμμα την ώρα που εκτελείται.

Επιπλέον, στην πάνω πλευρά του Arduino βρίσκονται 14 θηλυκά pin, αριθμημένα από 0 ως 13, που μπορούν να λειτουργήσουν ως ψηφιακές είσοδοι και έξοδοι. Λειτουργούν στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί το πολύ 40 mA ρεύμα. Μερικά από αυτά τα 14 pin, εκτός από ψηφιακές είσοδοι/έξοδοι έχουν και δεύτερη λειτουργία. Συγκεκριμένα:

- Τα pin 0 και 1 λειτουργούν ως RX και TX της σειριακής όταν το πρόγραμμά μας ενεργοποιεί την σειριακή θύρα. Έτσι, όταν λόγω χάρη το πρόγραμμά μας στέλνει δεδομένα στην σειριακή, αυτά προωθούνται και στην θύρα USB μέσω του ελεγκτή Serial-Over-USB αλλά και στο pin 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή (π.χ. ένα δεύτερο Arduino στο δικό του pin 1). Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμά μας ενεργοποιήσουμε το σειριακό interface, χάνουμε 2 ψηφιακές εισόδους/εξόδους.
- Τα pin 2 και 3 λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Συγκεκριμένα μπορούμε να τα ρυθμίσουμε μέσα από το πρόγραμμά μας ώστε να λειτουργούν αποκλειστικά ως ψηφιακές είσοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει και εκτελείται μια συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.
- Τα pin 3, 5, 6, 9, 10 και 11 μπορούν να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι με το σύστημα PWM (Pulse Width Modulation), δηλαδή το ίδιο σύστημα που διαθέτουν οι μητρικές των

υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων. Έτσι, μπορούμε να συνδέσουμε λόγω χάρη ένα LED σε κάποιο από αυτά τα pin και να ελέγξουμε πλήρως την φωτεινότητά του με ανάλυση 8bit (256 καταστάσεις από 0-σβηστό ως 255-πλήρως αναμμένο) αντί να έχουμε απλά την δυνατότητα αναμμένο-σβηστό που παρέχουν οι υπόλοιπες ψηφιακές έξοδοι. Είναι σημαντικό να καταλάβουμε ότι το PWM δεν είναι πραγματικά αναλογικό σύστημα και ότι θέτοντας στην έξοδο την τιμή 127, δεν σημαίνει ότι η έξοδος θα δίνει 2.5V αντί της κανονικής τιμής των 5V, αλλά ότι θα δίνει ένα παλμό που θα εναλλάσσεται με μεγάλη συχνότητα και για ίσους χρόνους μεταξύ των τιμών 0 και 5V.

Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN, θα βρούμε μια ακόμη σειρά από 6 pin, αριθμημένα από το 0 ως το 5. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC (Analog to Digital Converter) που είναι ενσωματωμένο στον μικροελεγκτή. Για παράδειγμα, μπορούμε να τροφοδοτήσουμε ένα από αυτά με μια τάση την οποία μπορεί να κυμαίνεται με ένα ποτενσιόμετρο από 0V ως μια τάση αναφοράς V_{ref} η οποία, αν δεν κάνουμε κάποια αλλαγή είναι προ ρυθμισμένη στα 5V. Τότε, μέσα από το πρόγραμμά μας μπορούμε να «διαβάσουμε» την τιμή του pin ως ένα ακέραιο αριθμό ανάλυσης 10-bit, από 0 (όταν η τάση στο pin είναι 0V) μέχρι 1023 (όταν η τάση στο pin είναι 5V). Η τάση αναφοράς μπορεί να ρυθμιστεί με μια εντολή στο 1.1V, ή σε όποια τάση επιθυμείτε (μεταξύ 2 και 5V) τροφοδοτώντας εξωτερικά με αυτή την τάση το pin με την σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας.

1.2.3 Τροφοδοσία

Το Arduino μπορεί να τροφοδοτηθεί με ρεύμα είτε από τον υπολογιστή μέσω της σύνδεσης USB, είτε από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής φισ των 2.1mm (θετικός πόλος στο κέντρο) και βρίσκεται στην κάτω αριστερή γωνία του Arduino. Η τάση λειτουργίας της πλατφόρμα Arduino είναι 5V.

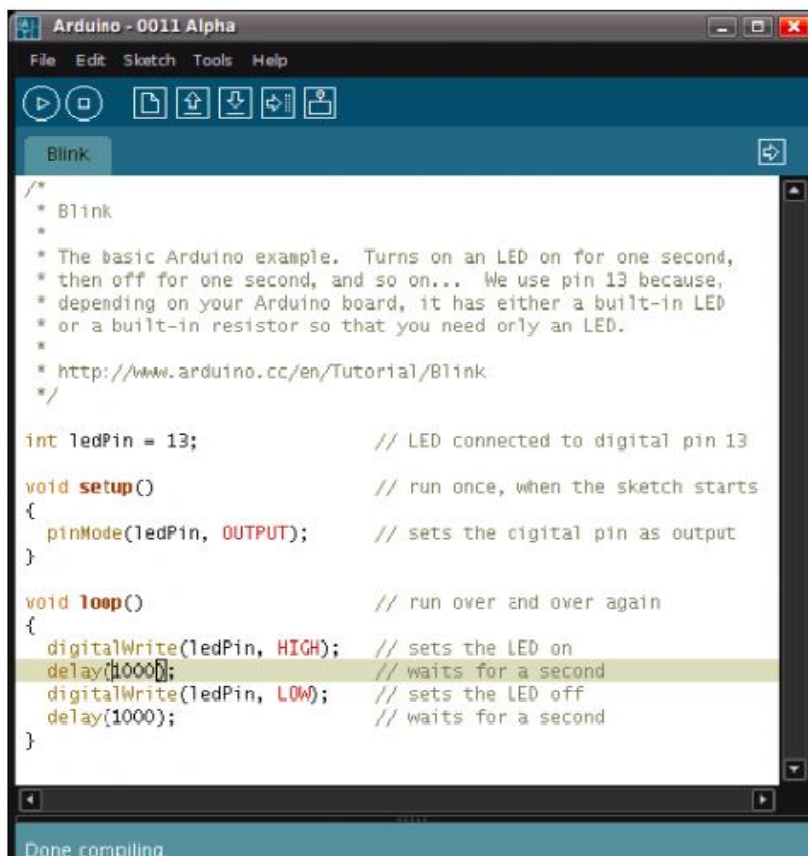
Για να μην υπάρχουν προβλήματα, η εξωτερική τροφοδοσία πρέπει να είναι από 7 ως 12V και μπορεί να προέρχεται από ένα κοινό μετασχηματιστή του εμπορίου, από μπαταρίες ή οποιαδήποτε άλλη πηγή DC. Δίπλα από τα pin αναλογικής εισόδου, υπάρχει μια ακόμα συστοιχία από 6 pin με την σήμανση POWER. Η λειτουργία του καθενός έχει ως εξής:

- Το πρώτο, με την ένδειξη RESET, όταν γειωθεί (σε οποιοδήποτε από τα 3 pin με την ένδειξη GND που υπάρχουν στο Arduino) έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.
- Το δεύτερο, με την ένδειξη 3.3V, μπορεί να τροφοδοτήσει τα εξαρτήματά μας με τάση 3.3V. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial-over-USB και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA.
- Το τρίτο, με την ένδειξη 5V, μπορεί να τροφοδοτήσει τα εξαρτήματά μας με τάση 5V. Ανάλογα με τον τρόπο τροφοδοσίας του ίδιου του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την θύρα USB (που ούτως ή άλλως λειτουργεί στα 5V), είτε από την εξωτερική τροφοδοσία αφού αυτή περάσει από ένα ρυθμιστή τάσης για να την «φέρει» στα 5V.
- Το τέταρτο και το πέμπτο pin, με την ένδειξη GND, είναι φυσικά γειώσεις.

- Το έκτο και τελευταίο pin, με την ένδειξη Vin έχει διπλό ρόλο. Σε συνδυασμό με το pin γείωσης δίπλα του, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino, στην περίπτωση που δεν μας βολεύει να χρησιμοποιήσουμε την υποδοχή του φισ των 2.1mm. Αν όμως έχουμε ήδη συνδεδεμένη εξωτερική τροφοδοσία μέσω του φισ, μπορούμε να χρησιμοποιήσουμε αυτό το pin για να τροφοδοτήσουμε εξαρτήματα με την πλήρη τάση της εξωτερικής τροφοδοσίας (7~12V), πριν αυτή περάσει από τον ρυθμιστή τάσης όπως γίνεται με το pin των 5V.

1.3 Ολοκληρωμένο περιβάλλον ανάπτυξης

Το IDE του Arduino είναι γραμμένο σε Java και μπορεί να τρέξει σε πολλαπλές πλατφόρμες. Περιλαμβάνει επεξεργαστή κώδικα (επεξεργαστή κειμένου με διάφορα εύχρηστα εργαλεία) και μεταγλωττιστή, και έχει την ικανότητα να φορτώνει εύκολα το πρόγραμμα μέσω σειριακής θύρας από τον υπολογιστή στην πλακέτα.



```
Arduino - 0011 Alpha
File Edit Sketch Tools Help
Blink
/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */
int ledPin = 13;           // LED connected to digital pin 13
void setup()               // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}
void loop()                // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}
Done compiling
```

Το περιβάλλον ανάπτυξης είναι βασισμένο στην Processing, ένα περιβάλλον ανάπτυξης σχεδιασμένο να εισαγάγει στον προγραμματισμό νέους χρήστες μη εξοικειωμένους με την ανάπτυξη λογισμικού. Η συγκεκριμένη γλώσσα προγραμματισμού προέρχεται από την Wiring, μια γλώσσα που μοιάζει με την C η οποία παρέχει παρόμοια λειτουργικότητα για μια πιο περιορισμένη σχεδίασης πλακέτα, της οποίας το περιβάλλον ανάπτυξης βασίζεται επίσης στην Processing.

Στην πραγματικότητα, το ολοκληρωμένο περιβάλλον ανάπτυξης είναι αυτό το οποίο κατά την μεταγλώττιση αποθηκεύει σε ένα προσωρινό αρχείο τον πηγαίο κώδικα του προγράμματος που έχουμε συγγράψει και στην συνέχεια προσθέτει σε αυτό όλα όσα είναι απαραίτητα (πρωτότυπα συναρτήσεων, υλοποίηση συνάρτησης `main()`, ενσωμάτωση επικεφαλίδων) ώστε να είναι δυνατή η μεταγλώττιση του.

Η μεταφορά των δεδομένων του τελικού αρχείου στην πλατφόρμα Arduino γίνεται μέσω USB. Για την επίτευξη αυτής της διαδικασίας χρησιμοποιείται από το ολοκληρωμένο περιβάλλον ανάπτυξης το εργαλείο `avrdude` και ο φορτωτής εκκίνησης της πλατφόρμας Arduino.

ΚΕΦΑΛΑΙΟ 2^ο

2. ΡΟΜΠΟΤΙΚΟΙ ΒΡΑΧΙΟΝΕΣ

2.1. Ρομποτική

Η Ρομποτική είναι εκείνος ο κλάδος της επιστήμης, που ασχολείται με τη σύλληψη, τη σχεδίαση, την κατασκευή, τη θεωρία και τις εφαρμογές των ρομπότ. Τα ρομπότ είναι μηχανές, που δεν έχουν τη μορφή ή τη συμπεριφορά του ανθρώπου, αλλά μπορούν να εκτελούν εργασίες, που κάνει ο άνθρωπος. Η χρήση τους, λοιπόν, έχει σκοπό την αντικατάσταση του ανθρώπου στην εκτέλεση εργασίας, η οποία αφορά τόσο στο φυσικό επίπεδο, όσο και στο επίπεδο λήψης απόφασης.

Η εξέλιξη των ρομπότ έχει περάσει από πολλά στάδια. Τα ρομπότ της πρώτης γενιάς δεν είχαν την ικανότητα υπολογισμού και αίσθησης, σε αντίθεση με τα ρομπότ της δεύτερης γενιάς, τα οποία διαθέτουν περιορισμένη υπολογιστική ισχύ, γλώσσες προγραμματισμού υψηλού επιπέδου και αισθητήρες ανατροφοδότησης. Τα ρομπότ της τρίτης γενιάς διαθέτουν νοημοσύνη με την έννοια ότι είναι ικανά να παίρνουν αποφάσεις κατά τη διάρκεια εκτέλεσης της εργασίας τους. Τις ικανότητες αυτές, τις αποκτούν μέσω τεχνικών της τεχνητής νοημοσύνης σε συνδυασμό με εξελιγμένες μορφές αισθητήρων αφής, δύναμης, απόστασης, όρασης, κ.ο.κ..

Τα βιομηχανικά ρομπότ είναι εξελιγμένα συστήματα αυτοματισμού, που χρησιμοποιούν ηλεκτρονικό υπολογιστή σαν μια βασική συνιστώσα του ελέγχου τους. Σήμερα, οι υπολογιστές αποτελούν ένα βασικό μέρος του βιομηχανικού αυτοματισμού. Κατευθύνουν γραμμές παραγωγής και ελέγχουν συστήματα κατασκευής (λ.χ. εργαλειομηχανές, συγκολλητές, κοπτικές διατάξεις Laser κ.α.) Τα νέα ρομπότ εκτελούν ποικίλες εργασίες στα βιομηχανικά συστήματα και γενικά συμμετέχουν στον πλήρη αυτοματισμό των εργοστασίων.

Το κύριο πλεονέκτημα του ρομπότ είναι η ευελιξία του. Μπορεί να προσαρμοστεί σε διάφορα προϊόντα στην ίδια γραμμή παραγωγής, όπως απαιτούν οι αλλαγές της αγοράς και να επαναπρογραμματιστεί έτσι, ώστε να είναι κατάλληλο για μικρές ή μεγάλες μεταβολές του παραγόμενου προϊόντος. Έτσι, προσφέρει στη βιομηχανία μαζικής παραγωγής έναν τρόπο να αντιμετωπίζει τις μεταβολές της απαιτούμενης ποσότητας ή του τύπου του προς παραγωγή προϊόντος.

Σύμφωνα με το Robot Institute της Αμερικής, ως ρομπότ μπορούμε να ορίσουμε έναν μηχανισμό σχεδιασμένο, ώστε μέσω προγραμματιζόμενων κινήσεων να μεταφέρει υλικά, τεμάχια, εργαλεία ή ειδικευμένες συσκευές με σκοπό την εκτέλεση ποικιλίας εργασιών. Ένας τέτοιος μηχανισμός περιλαμβάνει, συνήθως, τις ακόλουθες συνιστώσες: ένα μηχανολογικό υποσύστημα, ένα υποσύστημα αίσθησης και ένα σύστημα ελέγχου.

Το μηχανολογικό υποσύστημα επιτρέπει στο ρομπότ να εκτελεί την εργασία, που του έχει ανατεθεί. Αποτελείται από το υποσύστημα αίσθησης το οποίο βοηθάει το ρομπότ να συλλέγει πληροφορίες για την κατάσταση στην οποία βρίσκονται τόσο το ίδιο, όσο και το περιβάλλον του. Εκτός των άλλων, δέχεται εξωτερικές εντολές, τις επεξεργάζεται, τις μεταφράζει σε ηλεκτρική ισχύ, που θα δοθεί στους κινητήρες του ρομπότ, καθώς, επίσης, παράγει σήματα εξόδου, που θα πληροφορούν για την κατάσταση του συστήματος. Στο υποσύστημα αίσθησης περιλαμβάνονται όργανα μέτρησης, αισθητήρες, ηλεκτρονικά στοιχεία, κ.λ.π.. Επίσης συνήθως περιλαμβάνονται και μηχανισμοί, που επιτρέπουν στο ρομπότ να κινείται, όπως αρθρώσεις, σύστημα μετάδοσης κίνησης, επενεργητές –κινητήρες, οδηγούς, κ.λ.π..

2.1.1 Ιστορική Αναδρομή

Τα βιομηχανικά ρομπότ αναπτύχθηκαν ταυτόχρονα με τον υπολογιστικό αριθμητικό έλεγχο (CNC). Βεβαίως, το πρώτο ρομπότ κατασκευάστηκε το 1961, αλλά τα ρομπότ άρχισαν να παίζουν πρωτεύοντα ρόλο στη βιομηχανική παραγωγή μόνον κατά τα τέλη της δεκαετίας του 1970.

Η λέξη «ρομπότ» πρωτοεμφανίστηκε πριν από έναν αιώνα. Ο Τσέχος δραματογράφος Karel Capek, χρησιμοποίησε για πρώτη φορά το 1921 αυτόν τον όρο στο έργο του R.U.R.("Rossum's Universal Robots"). Αυτός ο όρος επινοήθηκε από την τσέχικη λέξη «robot», η οποία έχει τη σημασία της καταναγκαστικής εργασίας. Στο έργο του περιγράφεται η κατασκευή έξυπνων συσκευών, οι οποίες χρησιμοποιούνται ως υπηρέτες του ανθρώπινου δημιουργού τους.

Κατά τη δεκαετία του 1940, ο Ρώσος συγγραφέας επιστημονικής φαντασίας, Isaac Asimov, περιέγραψε το ρομπότ ως μια μηχανή με εμφάνιση ανθρώπου απαλλαγμένο, όμως από συναισθήματα. Η συμπεριφορά του υπαγορευόταν από έναν εγκέφαλο προγραμματισμένο από ανθρώπους με ιδιαίτερο τρόπο, ώστε να ακολουθεί βασικές ηθικές αρχές. Για πρώτη φορά, επίσης, χρησιμοποιήθηκε από τον Asimov ο όρος «ρομποτική» για να περιγράψει τον τομέα της επιστήμης, που ασχολείται με τα ρομπότ.

Ο όρος βιομηχανικό ρομπότ (industrial robot) καθιερώθηκε το 1954 από τον G.C. Devol (ΗΠΑ). Ο Devol περιέγραψε πώς μπορεί να κατασκευαστεί ένα ελεγχόμενο μηχανικό χέρι, το οποίο μπορεί να εκτελεί διάφορες εργασίες στη βιομηχανία. Το πρώτο βιομηχανικό ρομπότ κατασκευάστηκε και τέθηκε σε λειτουργία το 1961 από την εταιρία Unimation. Έκτοτε, τέθηκαν σε λειτουργία χιλιάδες ρομπότ στην Αμερική, στην Ιαπωνία και στην Ευρώπη.

Η πρώτη γενιά των σύγχρονων ρομπότ απείχε κατά πολύ από τα ανθρωπόμορφα μηχανήματα, καθώς οι περισσότεροι κατασκευαστές δεν είχαν σκοπό να μιμηθούν το ανθρώπινο γένος. Το δημοφιλές ρομπότ Unimate της

δεκαετίας του 1960 ήταν ικανό να κινήσει μονάχα το ένα του χέρι προς διάφορες κατευθύνσεις και να ανοιγοκλείνει την παλάμη του. Σήμερα, υπάρχουν πάνω από δύο εκατομμύρια ρομπότ Roomba, τα οποία κινούνται και εκτελούν εργασίες, που άλλοτε εκτελούνταν από τους ανθρώπους, αλλά μοιάζουν περισσότερο με γρήγορες χελώνες παρά με υπηρέτες.

2.1.2 Τομείς Εφαρμογής

Τις τελευταίες δεκαετίες σημειώνεται μία ραγδαία ανάπτυξη των επιστημών και της τεχνολογίας. Στη διάρκεια του 20ου αιώνα εμφανίστηκε για πρώτη φορά η έννοια του ρομπότ και της ρομποτικής, του επιστημονικού πεδίου, που το ρομπότ αποτελείτο αντικείμενο μελέτης του. Η Ρομποτική βρήκε αρκετά πεδία εφαρμογής όπως στη βιομηχανία, στον στρατό και αλλού.

Γενικά, τα βιομηχανικά ρομπότ μπορούμε να πούμε ότι παρέχουν:

- Υποκατάστατο ανθρώπινου χεριού
- Ευελιξία
- Ευφυΐα και ειδικούς αισθητήρες
- Ταχύτητα και ακρίβεια
- Μεγαλύτερη δύναμη από τον άνθρωπο
- Γρήγορη ανάπτυξη – ολοκλήρωση
- Οικονομικές λύσεις για μικρή παραγωγή
- Σταθερή ποιότητα προϊόντος και παραγωγικότητα

Ακόμα αξίζει να αναφερθεί η εφαρμογή της ρομποτικής τεχνολογίας και στον τομέα της ιατρικής.

Η Ρομποτική Ιατρική (Medical Robotics) έχει αναπτυχθεί τις τελευταίες κυρίως δεκαετίες και η έρευνα που γίνεται συγκεντρώνεται σε συγκεκριμένα πεδία, από τα οποία τα πιο κύρια είναι τα ακόλουθα:

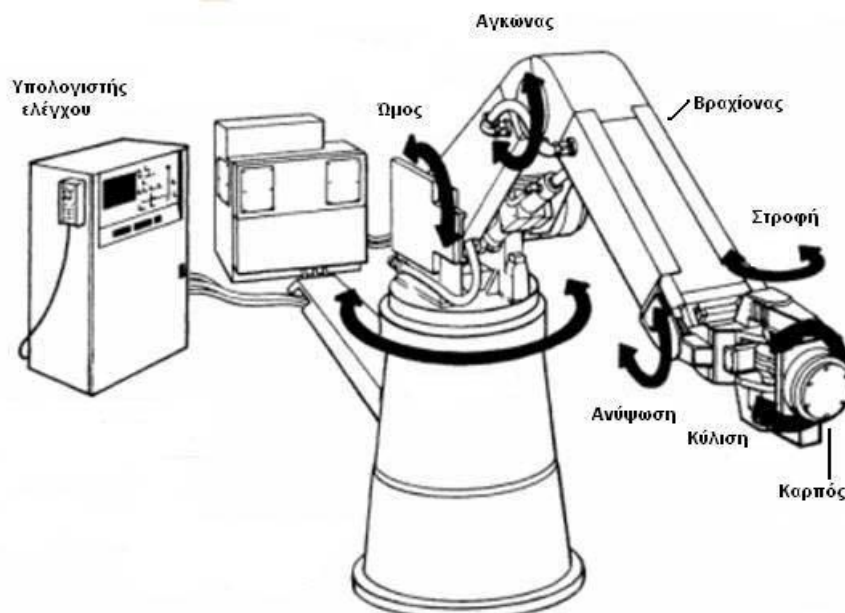
- Ρομποτική Χειρουργική (Robotic Surgery)
- Ρομποτική Αποκατάστασης (Rehabilitation Robotics)
- Προσομοιωτές Χειρουργικών Επεμβάσεων Εικονικής Πραγματικότητας με Οπτική Ανάδραση - Ανατροφοδότηση Δύναμης ή χωρίς (Training Simulators

with Haptic Display)

- Διαγνωστική Ρομποτική Τεχνολογία (Diagnostics Robotic Technology)
- Μικρορομποτική (Micro-Robotics)

2.2 Χαρακτηριστικά των ρομπότ

Η τυπική μορφή ενός ρομπότ-βραχίονα φαίνεται στην εικόνα 2.2. Βασικά, ένα βιομηχανικό ρομπότ μιμείται το ανθρώπινο χέρι. Επειδή τα βιομηχανικά ρομπότ εκτελούν διάφορους χειρισμούς, ονομάζονται και μηχανικοί χειριστές ή απλώς χειριστές. Ένα ρομπότ έχει αρθρώσεις ανάλογες αυτών του ανθρώπινου χεριού.



Εικόνα 2.2 Τυπική μορφή βιομηχανικού ρομπότ (Cincinnati Milacron)

Ο υπολογιστής ενός σύγχρονου ρομπότ περιέχει ένα πρόγραμμα ελέγχου και ένα πρόγραμμα εργασίας. Το πρόγραμμα ελέγχου δίνεται από τον κατασκευαστή και ρυθμίζει την κίνηση κάθε μιας άρθρωσης του ρομπότ. Το

πρόγραμμα εργασίας δίνεται από τον χρήστη και καθορίζει τις κινήσεις, που χρειάζονται για να εκτελεστεί κάθε φορά η επιθυμητή εργασία. Ένα πρόγραμμα εργασίας μπορεί να παραχθεί είτε περνώντας το ρομπότ από τις θέσεις που απαιτεί η συγκεκριμένη εργασία, είτε χρησιμοποιώντας κατάλληλες γλώσσες προγραμματισμού. Όταν χρησιμοποιείται μια γλώσσα προγραμματισμού, το ρομπότ περιέχει έναν κατάλληλο επεξεργαστή, που μεταφράζει το πρόγραμμα εργασίας και παρέχει τα δεδομένα, που χρειάζεται το πρόγραμμα ελέγχου για να καθοδηγήσει το ρομπότ στις επιθυμητές κινήσεις. Το πρόγραμμα ελέγχου χρησιμοποιεί το πρόγραμμα εργασίας σαν είσοδο και συνεπώς, για κάθε εργασία ο χρήστης πρέπει να γράψει ένα διαφορετικό πρόγραμμα εργασίας.

Τα πλεονεκτήματα των βιομηχανικών ρομπότ συνοψίζονται ως εξής:

- α) Ευελιξία
- β) Υψηλή παραγωγικότητα
- γ) Καλύτερη ποιότητα προϊόντος
- δ) Βελτιωμένη ποιότητα ζωής

2.3 Δομή των Ρομπότ

Το βιομηχανικό ρομπότ είναι ένας προγραμματιζόμενος μηχανικός χειριστής, ικανός να κινείται προς διάφορες διευθύνσεις, εφοδιασμένος στο άκρο του με μια διάταξη εργασίας, που ονομάζεται τελικό σημείο δράσης (ή εργαλείο), και ικανός να εκτελεί διάφορες βιομηχανικές εργασίες, που συνήθως γίνονται από τον άνθρωπο.

Ένα σύγχρονο ρομπότ περιλαμβάνει τουλάχιστον τρία βασικά δομικά μέρη:

- 1) τον χειριστή (βραχίονα, χέρι), που είναι το κινούμενο μηχανικό τμήμα,
- 2) τα στοιχεία δράσης (κινητήρες κ.λ.π.), που ενεργοποιούν τις αρθρώσεις του βραχίονα και

3) τον υπολογιστή, που αποθηκεύει και εκτελεί τα προγράμματα εργασίας και ελέγχει τις κινήσεις του ρομπότ.

Ο βραχίονας του ρομπότ περιέχει το κύριο σώμα και τον καρπό, που έχει στο τέλος του το εργαλείο (τελικό στοιχείο δράσης). Το εργαλείο μπορεί να είναι μια κεφαλή συγκόλλησης, ένα πιστόλι χρωματίσματος, ένα μηχανικό εργαλείο ή μια αρπάγη, που ανοιγοκλείνει ανάλογα με την εφαρμογή, στην οποία πρόκειται να χρησιμοποιηθεί το ρομπότ. Επειδή όλα τα εργαλεία στερεώνονται στο τέλος των ρομπότ, λέγονται και "τελικά στοιχεία δράσης".

Το κύριο σώμα (δηλαδή ο κορμός) του ρομπότ αποτελείται από μια διαδοχή συνδέσμων ή μελών (links), που συνδέονται με αρθρώσεις (joints). Οι αρθρώσεις ελέγχουν τις κινήσεις των συνδέσμων. Η ομάδα των αρθρώσεων, που ελέγχει την κίνηση του εργαλείου, ονομάζεται "καρπός". Κάθε μια από τις αρθρώσεις του βραχίονα και του καρπού παρέχει έναν βαθμό ελευθερίας στην κίνηση του τελικού στοιχείου δράσης. Έτσι, ένα ρομπότ με n βαθμούς ελευθερίας περιέχει n αρθρώσεις ή n άξονες κίνησης γενικά.

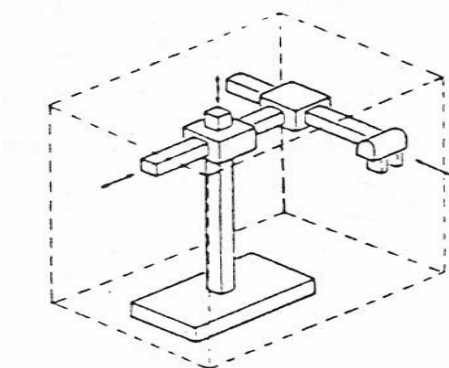
Η κίνηση του τελικού στοιχείου δράσης ρυθμίζεται ελέγχοντας τη θέση και την ταχύτητα των αξόνων κίνησης του ρομπότ. Στη ρομποτική ένας άξονας κίνησης ισοδυναμεί με έναν βαθμό ελευθερίας, ως προς τον οποίο μπορεί να κινηθεί το ρομπότ. Για να μπορέσει ένα ρομπότ να φτάσει ένα αυθαίρετο σημείο (μέσα στο χώρο εργασίας του) με έναν επιθυμητό προσανατολισμό του εργαλείου, χρειάζεται να έχει έξι άξονες (βαθμούς ελευθερίας) κίνησης. Έστω και μόνο ένας διαφορετικός προσανατολισμός του εργαλείου μπορεί να αλλάξει εντελώς τη θέση του βραχίονα του ρομπότ.

2.4 Τύποι ρομποτικών βραχιόνων και χώρος εργασίας

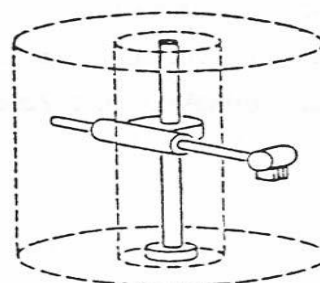
Το μηχανικό τμήμα των ρομπότ αποτελείται από μια αλληλουχία μηχανικών συνδέσμων (links) και αρθρώσεων (joints) που ενώνουν τους συνδέσμους ανά δύο. Ένα τυπικό ρομπότ αποτελείται από το κυρίως τμήμα (δηλαδή τον βραχίονα) και από τον καρπό. Τόσο ο βραχίονας (arm) όσο και ο καρπός (wrist) έχουν τρεις βαθμούς ελευθερίας, ο καθένας. Υπάρχουν όμως ρομπότ στα οποία ο καρπός έχει λιγότερους βαθμούς ελευθερίας.

Τα ρομπότ ταξινομούνται ανάλογα με το σύστημα συντεταγμένων του βραχίονα ως εξής (σχήμα. 2.1):

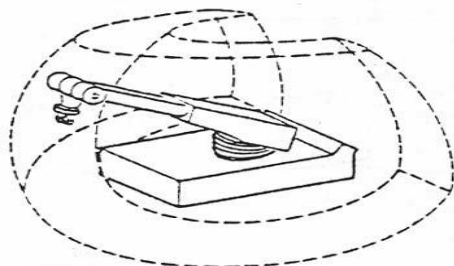
- **Καρτεσιανά:** Τρεις γραμμικοί άξονες
- **Κυλινδρικά:** Δυο γραμμικοί και ένας στροφικός άξονας
- **Σφαιρικά:** Ένας γραμμικός και δυο στροφικοί άξονες
- **Αρθρωτά:** Τρεις στροφικοί άξονες



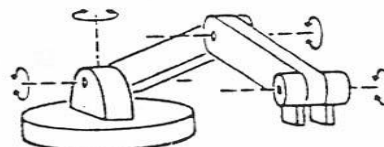
Καρτεσιανό ή
X-Y-Z ρομπότ



Κυλινδρικό
ρομπότ



Σφαιρικό ρομπότ

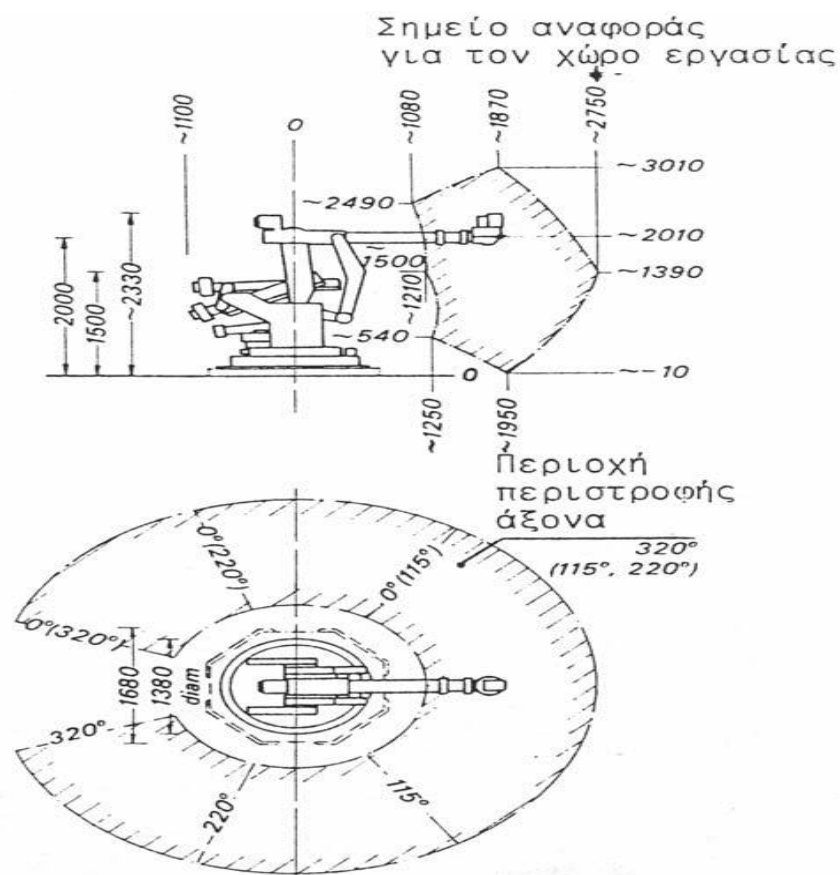


Αρθρωτό
ρομπότ

Σχήμα 2.1 Οι τέσσερις βασικοί δομικοί τύποι βιομηχανικού ρομπότ

Οι γραμμικές αρθρώσεις μπορούν να είναι αρθρώσεις ολίσθησης (συμβολικά S: sliding), ή πρισματικές (συμβολικά P: prismatic). Μια, στροφική άρθρωση συμβολίζεται με R (revolute joint). Έτσι ο τύπος ενός ρομπότ με βάση τα συστήματα συντεταγμένων των αρθρώσεων του συμβολίζεται με την αλληλουχία των συμβόλων S, P και R αρχίζοντας από τη βάση και προχωρώντας προς τον καρπό. Ένα σφαιρικό ρομπότ μπορεί να είναι της μορφής RRP, ενώ ένα αρθρωτό ρομπότ συμβολίζεται με RRR.

Ένα από τα βασικά χαρακτηριστικά ενός βιομηχανικού ρομπότ είναι το μέγεθος και το σχήμα του χώρου εργασίας του (work volume ή reach envelope). Πρέπει να σημειωθεί ότι με την προσθήκη του τελικού στοιχείου δράσης(εργαλείου) στον καρπό του ρομποτικού βραχίονα, ο χώρος εργασίας μεταβάλλεται ανάλογα με τον τύπο και το μέγεθος του εργαλείου.



Σχήμα 2.2: Χώρος εργασίας του ρομπότ

2.4.1 Ρομπότ καρτεσιανών συντεταγμένων

Το κυρίως σώμα ενός ρομπότ του τύπου αυτού αποτελείται από τρεις γραμμικούς άξονες όπως φαίνεται στο σχήμα 2.1. Η δομή μπορεί να είναι όμοια με των εργαλειομηχανών (βάση, τραπέζι εργασίας, κ.ά.) αλλά τότε ο λόγος μεταξύ του χώρου εργασίας του ρομπότ και του χώρου που καταλαμβάνει είναι μικρότερος. Γενικά, τα χαρακτηριστικά (ηλεκτρονικό υλικό;hardware, πρόγραμμα έλεγχου κλπ) ενός καρτεσιανού ρομπότ είναι όμοια με εκείνα των εργαλειομηχανών αριθμητικού ελέγχου (CNC).

2.4.2 Ρομπότ κυλινδρικών συντεταγμένων

Το κυρίως σώμα ενός ρομπότ του τύπου αυτού αποτελείται από ένα οριζόντιο βραχίονα στερεωμένο σε μια κατακόρυφη κολώνα. Η κολώνα είναι με τη σειρά της στερεωμένη πάνω σε μια περιστρεφόμενη βάση (σχήμα2.1). Ο οριζόντιος βραχίονας κινείται προς τα εμπρός και προς τα πίσω κατά τη διεύθυνση του διαμήκους άξονα του και επίσης ανεβοκατεβαίνει στη κολώνα. Κολώνα και βραχίονας στέφονται σαν ένα σώμα πάνω στη βάση γύρω από τον κατακόρυφο άξονα.

2.4.3 Ρομπότ σφαιρικών συντεταγμένων

Τα ρομπότ του τύπου αυτού αποτελούνται από μια στρεφόμενη βάση, ένα ανυψούμενο στέλεχος και ένα τηλεσκοπικό βραχίονα που κινείται προς τα μέσα και προς τα έξω όπως φαίνεται στο σχήμα 2.1. Το μέγεθος της περιστροφής μετράται από ψηφιακούς κωδικοποιητές που είναι στερεωμένοι στους άξονες. Το βασικό μειονέκτημα των σφαιρικών ρομπότ είναι και πάλι η μικρή διακριτική ικανότητα θέσης των δυο στρωφικών αξόνων που μεταβάλλεται με το μήκος του βραχίονα. Τα σφαιρικά ρομπότ, εκτός από το πλεονέκτημα της αυξημένης ταχύτητας κίνησης των στρωφικών αξόνων, έχουν και το πλεονέκτημα της αυξημένης ευελιξίας σε σχέση τόσο με τα καρτεσιανά όσο και με τα κυλινδρικά ρομπότ.

2.4.4 Αρθρωτό ρομπότ

Τα αρθρωτά ρομπότ αποτελούνται από τρία σταθερά μέλη (συνδέσμους) που ενώνονται με στρωφικές αρθρώσεις και είναι τοποθετημένα πάνω σε μια στρεφόμενη βάση (σχήμα 2.1). Η κινηματική διάταξη μοιάζει με εκείνη του ανθρώπινου χεριού. Το εργαλείο (αρπάγη) είναι ανάλογο της παλάμης και προσαρμόζεται στον πήχυ (Forearm) μέσω του καρπού. Ο "αγκώνας" συνδέει τον πήχυ με τον άνω βραχίονα και ο "ώμος" συνδέει τον άνω βραχίονα με τη βάση. Πολλές φορές στην άρθρωση του ώμου διατίθεται και μια περιστροφική κίνηση σε οριζόντιο επίπεδο όπως φαίνεται στο σχήμα 2.6.

2.5 Αρθρώσεις (Joints)

Στο χώρο των τριών διαστάσεων η σχετική κίνηση που μπορεί να πραγματοποιήσει ένα συμπαγές σώμα, ως προς ένα άλλο που θεωρείται ακίνητο, χωρίζεται σε 6 βασικές επιμέρους κινήσεις. 3 μεταφορικές κινήσεις στις διευθύνσεις των αξόνων του τρισσορθογωνίου συστήματος συντεταγμένων, και 3 περιστροφικές κινήσεις γύρω από αυτούς.

Άρθρωση (joint) ονομάζεται η κατάλληλα διαμορφωμένη διάταξη που συνδέει δύο σώματα μεταξύ τους, περιορίζοντας ένα ή, συχνότερα, περισσότερους από τους μεταξύ τους βαθμούς ελευθερίας, δηλαδή εμποδίζει την πραγματοποίηση μιας ή περισσότερων από τις παραπάνω 6 δυνατότητες κίνησης, μειώνοντας την ελευθερία κίνησης του ενός ως προς το άλλο σώμα.

2.5.1 Είδη αρθρώσεων

Οι ρομποτικοί βραχίονες χαρακτηρίζονται κυρίως από τον αριθμό και από τον τύπο των αρθρώσεων που έχουν. Τα είδη των αρθρώσεων διακρίνονται σε περιστροφικές αρθρώσεις (έναν βαθμό ελευθερίας), πρισματικές αρθρώσεις (έναν βαθμό ελευθερίας) και σφαιρικές αρθρώσεις (τρεις βαθμοί ελευθερίας) [βαθμός ελευθερίας = ανεξάρτητη κίνηση]. Έτσι για παράδειγμα ένας βραχίονας

ο οποίος έχει δυο περιστροφικές, μια πρισματική και μια σφαιρική άρθρωση, χαρακτηρίζεται ως βραχίονας έξι βαθμών ελευθερίας.

2.5.2 Αρθρώσεις σε ρομποτικούς βραχίονες

Οι ρομποτικοί βραχίονες, αν και είναι μηχανισμοί με πολύ μεγάλη ευελιξία, στην πλειονότητα τους βασίζονται σε βραχίονες που συνδέονται μεταξύ τους με αρθρώσεις ενός βαθμού ελευθερίας.

Έτσι οι ρομποτικοί βραχίονες, για να μπορέσουν να επιτύχουν την προσδοκούμενη ευελιξία, σχεδιάζονται με πολλές (συνήθως από τρεις μέχρι έξι) αρθρώσεις, κατάλληλα τοποθετημένες έτσι ώστε να επιτρέπουν στους επενεργητές να κινούνται εύκολα. Οι αρθρώσεις αυτές είναι οι:

1. **Περιστροφική άρθρωση** (revolute joint), που επιτρέπει την ελεγχόμενη περιστροφή γύρω από τον άξονά της. Η περιστροφική άρθρωση κυρίως λόγω της μεγάλης ταχύτητάς της είναι η πιο κοινή άρθρωση σε ρομποτικούς βραχίονες.
2. **Γραμμικές αρθρώσεις** που χωρίζονται σε ολίσθησης (sliding) και πρισματικές (prismatic) και επιτρέπουν την ελεγχόμενη μετατόπιση προς μία κατεύθυνση. Οι αρθρώσεις αυτές χαρακτηρίζονται για την μεγάλη ακρίβειά τους, και χρησιμοποιούνται κυρίως σε διατάξεις συναρμολόγησης.

2.6 Βαθμοί ελευθερίας

Ο αριθμός των βαθμών ελευθερίας ενός ρομποτικού βραχίονα είναι ο αριθμός των ανεξάρτητων μεταβλητών θέσης. Ο αριθμός αυτός πρέπει να είναι γνωστός και μονοσήμαντα ορισμένος για κάθε διάταξη, έτσι ώστε να είναι εφικτός ο προσδιορισμός της θέσης των τμημάτων που τη συνιστούν. Πρόκειται για μια γενική αρχή, που εφαρμόζεται σε όλους τους μηχανισμούς.

Κρίνεται σκόπιμο να επισημάνουμε τη διαφορά που υπάρχει ανάμεσα στους βαθμούς κινητικότητας ενός βραχίονα και τους βαθμούς ελευθερίας που απαιτούνται για την εκτέλεση ενός έργου. Για έναν βραχίονα το πλήθος των βαθμών κινητικότητας είναι σταθερό και ίσο με το πλήθος των αρθρώσεων του. Από την άλλη πλευρά οι βαθμοί ελευθερίας είναι άμεσα συνδεδεμένοι με το συγκεκριμένο έργο που καλείται να φέρει εις πέρας ο βραχίονας. Για την γενική περίπτωση που θέλουμε να τοποθετήσουμε και να προσανατολίσουμε ένα αντικείμενο στο χώρο απαιτούνται έξι βαθμοί ελευθερίας (τρεις για να τοποθετήσουμε ένα σημείο του αντικείμενου στο χώρο και τρεις για να προσανατολίσουμε το αντικείμενο ως προς ένα σύστημα συντεταγμένων αναφοράς) είναι προφανές ότι ένας ρομποτικός βραχίονας με 6 βαθμούς κινητικότητας μπορεί να ανταπεξέλθει σ' αυτό το έργο, όπως επίσης και σε οποιοδήποτε άλλο έργο που απαιτεί μέχρι έξι βαθμούς ελευθερίας.

ΚΕΦΑΛΑΙΟ 3^ο

3.ΚΙΝΗΜΑΤΙΚΗ ΘΕΩΡΙΑ

3.1 Εισαγωγή στην κινηματική θεωρία

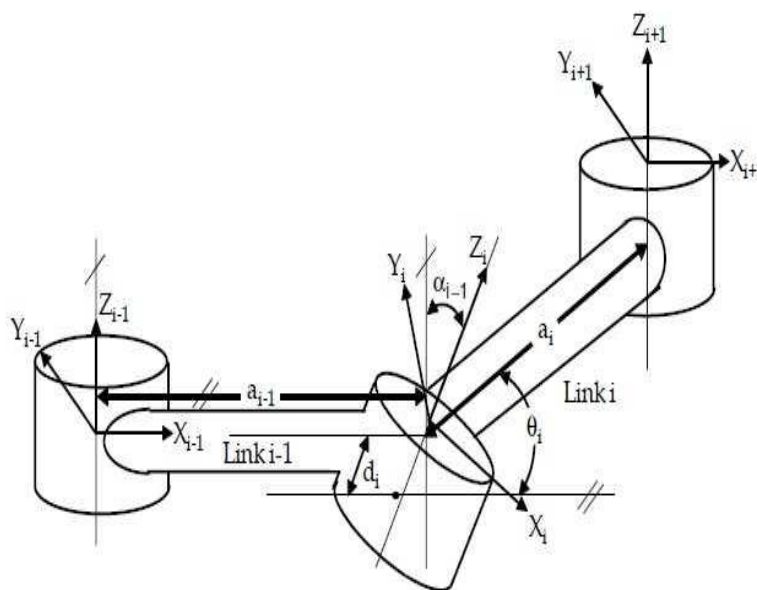
Κινηματική είναι η επιστήμη που μελετά την κίνηση χωρίς να λαμβάνει υπόψη τις δυνάμεις που την προκαλούν. Στην κινηματική ενδιαφερόμαστε για την θέση, την ταχύτητα, την επιτάχυνση και όλες τις παραγώγους των μεταβλητών που περιγράφουν τη θέση.

Η κινηματική ανάλυση ενός βραχίονα αφορά την επίλυση δύο προβλημάτων, του ευθέως και του αντίστροφου κινηματικού προβλήματος. Το ευθύ κινηματικό πρόβλημα συνίσταται την εύρεση της θέσης και του προσανατολισμού του άκρου του βραχίονα ως προς την βάση του, όταν γνωρίζουμε την θέση κάθε άρθρωσης, ενώ το αντίστροφο κινηματικό πρόβλημα συνίσταται στην εύρεση της θέσης της κάθε άρθρωσης, η οποία τοποθετεί το άκρο του βραχίονα σε μια συγκεκριμένη θέση και προσανατολισμό σε σχέση με την βάση του. Ο βαθμός δυσκολίας επίλυσης των προβλημάτων αυτών εξαρτάται από την γεωμετρία του βραχίονα.

Για την μετακίνηση του άκρου του βραχίονα από την παρούσα θέση του στην επιθυμητή είναι απαραίτητη η γνώση των γωνιών των αρθρώσεων που αντιστοιχούν στη επιθυμητή θέση. Οι γωνίες αυτές υπολογίζονται επιλύοντας το αντίστροφο κινηματικό πρόβλημα. Η μετάβαση μεταξύ δύο διαδοχικών καταστάσεων του βραχίονα γίνεται μέσω ευθύγραμμης τροχιάς με παραβολική μείξη στο επίπεδο των αρθρώσεων ώστε να επιτυγχάνεται τόσο η ομαλή κίνηση της κατασκευής όσο και η ταυτόχρονη έναρξη και λήξη της κίνησης όλων των κινητήρων.

3.2 Ευθύ κινηματικό πρόβλημα

Ένας βραχίονας αποτελείται από σειριακούς συνδέσμους οι οποίοι τοποθετούνται σε κάθε άρθρωση από τη βάση μέχρι το εργαλείο τελικής δράσης. Για να εφαρμοστεί το ευθύ κινηματικό σε ένα ρομπότ θα πρέπει να χρησιμοποιήσουμε ένα κατάλληλο κινηματικό μοντέλο. Η μεθοδολογία Denavit - Hartenberg που χρησιμοποιεί τέσσερις παραμέτρους είναι η πιο κοινή μέθοδος για την περιγραφή του κινηματικού προβλήματος ενός ρομπότ. Ένα σύστημα συντεταγμένων επισυνάπτεται σε κάθε άρθρωση και καθορίζει τις DH παραμέτρους. Βάσει της μεθόδου αυτής επιλέγονται με συγκεκριμένο τρόπο τα συστήματα συντεταγμένων που είναι προσαρμοσμένα σε κάθε σύνδεσμο του ρομποτικού βραχίονα εκτελώντας τα βήματα του αλγορίθμου όπως θα δούμε στην υποενότητα 3.2.1. Έτσι προσδιορίζεται η θέση και ο προσανατολισμός του άκρου εργαλείας ως προς το ακίνητο σύστημα συντεταγμένων.



Σχήμα 3.2 Συντεταγμένες σε γενικού τύπου βραχίονα

3.2.1 Αλγόριθμος Denavit-Hartenberg

Ο αλγόριθμος των D-H, είναι μια διαδικασία σύμφωνα με την οποία τοποθετούνται ορθοκανονικά και δεξιόστροφα Συστήματα Συντεταγμένων στους συνδέσμους του βραχίονα. Η συστηματική αυτή μέθοδος μας βοηθάει στο να ορίσουμε την σχετική θέση και τον προσανατολισμό μεταξύ δύο διαδοχικών συνδέσμων.

Βασικό στοιχείο της μεθόδου αποτελεί ο σωστός ορισμός των πλαισίων κάθε άρθρωσης. Προκειμένου να γίνει αυτό, πρέπει να τηρηθούν οι παρακάτω τέσσερις κανόνες(Denavit – Hartenberg).

1. Επιλέγουμε τον άξονα Z_i κατά μήκος του άξονα της άρθρωσης $i+1$
2. Τοποθετούμε το κέντρο O_i , στην τομή του άξονα Z_i , με την κοινή κάθετο των αξόνων Z_{i-1} και Z_i .
3. Επιλέγουμε τον άξονα x_0 κατά μήκος της κοινής καθέτου των αξόνων Z_{i-1} και Z_i με τη φορά από την άρθρωση i στην άρθρωση $i+1$.
4. Επιλέγουμε τον άξονα y_i , ώστε να έχουμε δεξιόστροφο σύστημα συντεταγμένων για το πλαίσιο i

Έχοντας ορίσει τα πλαίσια αναφοράς κάθε άρθρωσης, όπως φαίνεται στο σχήμα 3.2 το μόνο που μας απομένει είναι να προσδιορίσουμε τις τέσσερις παραμέτρους της μεθόδου που είναι οι $d_i, \theta_i, a_i, \alpha_i$. Οι παράμετροι αυτοί ορίζονται ως εξής.

- a_i = η απόσταση μεταξύ των Z_i και Z_{i+1} μετρούμενη κατά μήκος του X_i
- α_i = η γωνία μεταξύ των Z_i και Z_{i+1} μετρούμενη ως προς τον X_i
- d_i = η απόσταση μεταξύ των X_{i-1} και X_i μετρούμενη κατά μήκος του Z_i
- θ_i = η γωνία μεταξύ των X_{i-1} και X_i μετρούμενη ως προς Z_i

Εφόσον έχουμε ορίσει τα πλαίσια των αρθρώσεων και τις παραμέτρους της μεθόδου D-H, ακολουθούμε τα παρακάτω βήματα, προκειμένου να ορίσουμε τη θέση και τον προσανατολισμό του πλαισίου ως προς το πλαίσιο $i-1$.

Βήμα 1: Περιστροφή του πλαισίου $i-1$ γύρω από τον άξονα Z_{i+1} κατά γωνία θ_i

Βήμα 2:Μετατόπιση \mathbf{d}_i του πλαισίου i-1 κατά μήκος του άξονα Z_{i-1}

Βήμα 3:Μετατόπιση \mathbf{a}_i (μήκος της κοινής καθέτου) κατά τον άξονα x_0

Βήμα 4:Περιστροφή γύρω από τον άξονα X_i κατά γωνία α_i

Από το βήμα 1 και βήμα 2, παίρνουμε την μήτρα μετασχηματισμού

$$T_{\Sigma_i^{i-1}} = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ένω εφαρμόζοντας το βήμα 3 και βήμα 4 έχουμε την μήτρα μετασχηματισμού

$$T_i^{\Sigma_i} = \begin{bmatrix} 1 & 0 & 0 & \alpha_i \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Οπότε συνολικά η μήτρα μετασχηματισμού που μας δίνει τη σχετική θέση και τον προσανατολισμό του πλαισίου i ως προς το πλαίσιο i-1,είναι η ακόλουθη:

$$T_i^{i-1} = T_{\Sigma_i^{i-1}} * T_i^{\Sigma_i} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & \alpha_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & \alpha_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

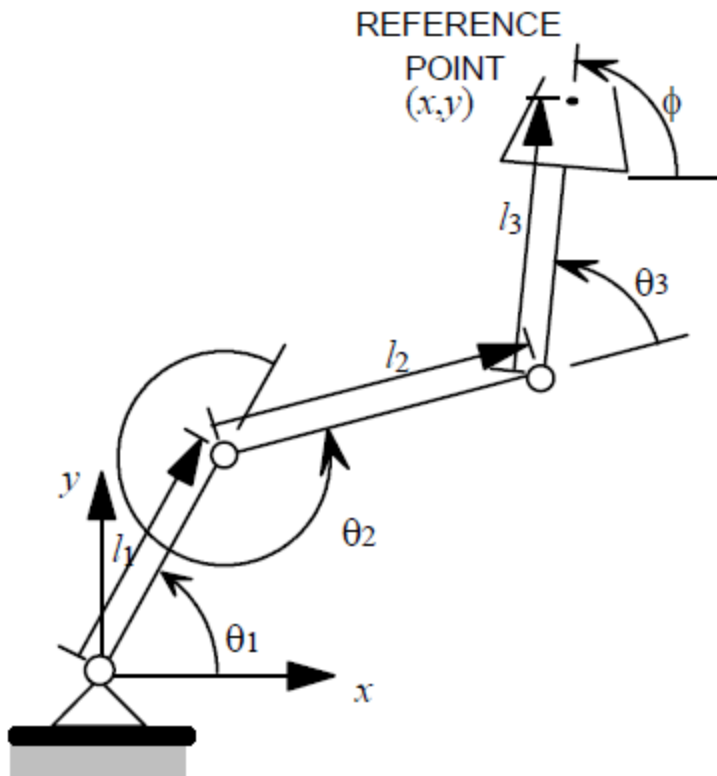
Στον παραπάνω πίνακα παραμέτρων κάθε σειρά περιέχει όλες τις απαραίτητες παραμέτρους για την δημιουργία των ομογενών μετασχηματισμών μεταξύ των πλαισίων των γειτονικών συνδέσμων.

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	L_1	0	θ_2
3	0	L_2	0	θ_3

3.1 Πίνακας παραμέτρων Denavit-Hartenberg.

3.3 Αντίστροφο κινηματικό πρόβλημα

Το αντίστροφο κινηματικό πρόβλημα αναφέρεται στον προσδιορισμό των μεταβλητών των αρθρώσεων του ρομπότ ενώ μας δίνεται η θέση και ο προσανατολισμός του εργαλείου της τελικής δράσης. Την επίλυση αυτού του προβλήματος αποτελεί η λύση μη γραμμικών αλγεβρικών εξισώσεων. Για την αριθμητική επίλυση μη γραμμικών συστημάτων εξισώσεων υπάρχουν πολλές αριθμητικές μέθοδοι που συγκλίνουν προς την λύση.



Σχήμα 3.2 .Επίπεδο ρομπότ τριών βαθμών ελευθερίας

Πίνακες μετασχηματισμού των συνδέσμων:

$${}^0_1T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

$${}^1_2T = \begin{bmatrix} c_2 & -s_2 & 0 & L_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$${}^2_3T = \begin{bmatrix} c_3 & -s_3 & 0 & L_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

Όπου $c_i = \cos \theta_i$ και $s_i = \sin \theta_i$

Από τους πίνακες μετασχηματισμού των συνδέσμων του επίπεδου βραχίονα και με τη χρήση τριγωνομετρικών ταυτοτήτων του αθροίσματος γωνιών βρίσκουμε τον 0_3T

$${}^0_3T = {}^0_1T {}^1_2T {}^2_3T = \begin{bmatrix} c_{123} & -s_{123} & 0 & l_1 c_1 + l_2 c_{12} \\ s_{123} & c_{123} & 0 & l_1 s_1 + l_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

Όπου $c_{123} = \cos(\theta_1 + \theta_2 + \theta_3)$ και $s_{123} = \sin(\theta_1 + \theta_2 + \theta_3)$

Ο 0_3T μας δίνει τη θέση και τον προσανατολισμό του τελευταίου πλαισίου του βραχίονα ως προς το ακίνητο σύστημα βάσης.

Δεδομένου ότι ο ρομποτικός βραχίονας είναι επίπεδος, μπορούμε να περιγράψουμε τα σημεία στόχευσης εύκολα, με τρεις αριθμούς x, y και φ . Η γωνία φ , ορίζει τον προσανατολισμό στο επίπεδο (σε σχέση με τον θετικό ημιάξονα X), του τρίτου μέλους.

Έτσι ο γενικός μετασχηματισμός έχει την περίπτωση αυτή την μορφή,

$${}^B_WT = \begin{bmatrix} c_\varphi & -s_\varphi & 0 & x \\ s_\varphi & c_\varphi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

Όλα τα προσβάσιμα σημεία ανήκουν στον υποχώρο που δομείται με βάση τις εξισώσεις (3.5). Από την εξίσωση της (3.4) με την (3.5), καταλήγουμε σ' ένα σύστημα τεσσάρων μη γραμμικών εξισώσεων τις οποίες πρέπει να επιλύσουμε, για να υπολογίσουμε τις θ_1, θ_2 και θ_3

$$c_{\varphi} = c_{123} \quad (3.6)$$

$$s_{\varphi} = s_{123} \quad (3.7)$$

$$x = l_1 c_1 + l_2 c_{12}, \quad (3.8)$$

$$y = l_1 s_1 + l_2 s_{12}, \quad (3.9)$$

Αθροίζοντας τα τετράγωνα των (3.8) και (3.9), καταλήγουμε στην σχέση

$$x^2 + y^2 = l_1^2 + l_2^2 + 2l_1 l_2 c_2, \quad (3.10)$$

Για τα οποία χρησιμοποιήσαμε τους τύπους

$$c_{12} = c_1 c_2 - s_1 s_2, \quad (3.11)$$

$$s_{12} = c_1 s_2 + s_1 c_2. \quad (3.12)$$

Επιλύοντας τώρα ως προς c_2 , έχουμε

$$c_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2} \quad (3.14)$$

Για να υπάρχει λύση, πρέπει το δεξιό μέλος της ισότητας (3.14) να παίρνει τιμές που ανήκουν στο διάστημα $[-1, 1]$. Ο αλγόριθμος επίλυσης, σ' αυτό το σημείο, διενεργεί τον παραπάνω έλεγχο και γνωμοδοτεί για την ύπαρξη ή όχι λύσης. Φυσιολογικά συμπεραίνουμε, ότι αν δεν ικανοποιείται η εν λόγω συνθήκη, ο στόχος βρίσκεται υπερβολικά μακριά και δεν είναι δυνατόν να προσεγγιστεί από τον βραχίονα.

Υπο την προϋπόθεση ότι ο στόχος μας βρίσκεται εντός του χώρου εργασίας, μπορούμε να υπολογίσουμε το s_2 από την παρακάτω γνωστή τριγωνομετρική ταυτότητα:

$$s_2 = \pm \sqrt{1 - c_2^2} \quad (3.15)$$

Τελικά με την χρήση της συνάρτησης δύο ορισμάτων τόξου εφαπτομένης, υπολογίζουμε τη γωνία θ_2 .

$$\theta_2 = A \tan 2(s_2, c_2) \quad (3.16)$$

Η επιλογή του προσίμου της (3.15), έχει να κάνει με τη διεύθυνση του βραχίονα. Έτσι, μπορούμε να επιλέξουμε λύση της μορφής <<αγκώνας πάνω>> ή <<αγκώνας κάτω>>, κατά την προσέγγιση του βραχίονα στο επιθυμητό σημείο. Για τον υπολογισμό της θ_2 , χρησιμοποιούμε μία αναδρομική σχέση που συχνά χρησιμοποιείται σε τέτοιου είδους προβλήματα κινηματικής. Υπολογίζουμε και το ημίτονο και το συνημίτονο, για να εκχωρήσουμε ως ορίσματα στη συνάρτηση δύο ορισμάτων τόξου εφαπτομένης. Αυτή η τακτική μας εξασφαλίζει την εύρεση όλων των λύσεων. Ακόμα εξασφαλίζει την τοποθέτηση της γωνίας στο σωστό τεταρτημόριο.

Έχοντας ως δεδομένο την τιμή της θ_2 , μπορούμε να επιλύσουμε τις (3.10) και (3.11), και να υπολογίσουμε την θ_1 . Έτσι γράφουμε τις εξισώσεις αυτές στην ακόλουθη μορφή.

$$x = k_1 c_1 - k_2 s_1, \quad (3.17)$$

$$y = k_1 s_1 + k_2 c_1, \quad (3.18)$$

όπου,

$$k_1 = l_1 + l_2 c_2, \quad (3.19)$$

$$k_2 = l_2 s_2 \quad (3.20)$$

Για να λύσουμε μια τέτοια εξίσωση προβαίνουμε σε αλλαγή των μεταβλητών. Συγκεκριμένα μεταβάλλουμε τον τρόπο αναπαράστασης των k_1 και k_2 .

Έστω ότι είναι:

$$r = +\sqrt{k_1^2 + k_2^2} \quad (3.21)$$

$$y = A \tan 2(k_2, k_1)$$

τότε,

$$k_1 = r \cos y,$$

$$k_2 = r \sin y, \quad (3.22)$$

Οι εξισώσεις (3.17) και (3.18) μπορούν τώρα να διατυπωθούν ως εξής:

$$\frac{x}{r} = \cos y \cos \theta_1 - \sin y \sin \theta_1, \quad (3.23)$$

$$\frac{y}{r} = \cos y \sin \theta_1 + \sin y \cos \theta_1 \quad (3.24)$$

Έτσι έχουμε,

$$\cos(y + \theta_1) = \frac{x}{r}, \quad (3.25)$$

$$\sin(y + \theta_1) = \frac{y}{r}, \quad (3.26)$$

Εφαρμόζοντας την συνάρτηση δύο ορισμάτων τόξου εφαπτομένης, καταλήγουμε στη σχέση :

$$y + \theta_1 = A \tan 2\left(\frac{y}{r}, \frac{x}{r}\right) = A \tan 2(y, x), \quad (3.27)$$

Στην συνέχεια υπολογίζουμε τη θ_1 , από την εξίσωση ,

$$\theta_1 = A \tan 2(x, y) - A \tan 2(k_2, k_1) \quad (3.28)$$

Χρειάζεται να σημειωθεί, ότι η επιλογή πρόσημου για τη θ_2 , επηρεάζει το πρόσημο της ποσότητας k_2 , η οποία με την σειρά της επηρεάζει την τελική τιμή

της θ_1 . Η αντικαταστάσεις που χρησιμοποιήθηκαν στις (3.21) και (3.22) αποτελούν ουσιώδεις χαρακτηριστικό της μεθόδου που επιλέγουμε για τα προβλήματα που συχνά διατυπώνονται στη μορφή που υποδεικνύεται από τις (3.10) και (3.11).Επισημαίνουμε επίσης ότι αν $x=y=0$, τότε η (3.28) είναι αόριστη, και η γωνία θ_2 μπορεί να έχει οποιαδήποτε τιμή.

Τέλος, με χρήση των εξισώσεων (3.8) και(3.9), μπορούμε να υπολογίσουμε το άθροισμα των τριών γωνιών($\theta_1+\theta_2+\theta_3$).

$$\theta_1 + \theta_2 + \theta_3 = A \tan 2(s_\varphi, c_\varphi) = \varphi \quad (3.28)$$

Από την σχέση (3.28), μπορούμε να υπολογίσουμε τη θ_3 , αφού ήδη γνωρίζουμε τις θ_1 και θ_2 .

ΚΕΦΑΛΑΙΟ 4^ο

4. ΣΧΕΔΙΑΣΜΟΣ – ΜΕΛΕΤΗ ΚΑΙ ΒΗΜΑΤΑ ΚΑΤΑΣΚΕΥΗΣ ΤΟΥ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ

4. 1 Εισαγωγή

Η κατασκευή του βραχίονα με τρεις βαθμούς ελευθερίας που αντιστοιχούν σε περιστροφικές αρθρώσεις, αποτελείται από συνδέσμους κατασκευασμένους από ξύλο πεπιεσμένου χάρτου.

Οι τρεις αρθρώσεις με παράλληλους άξονες αντιστοιχούν σε ώμο, αγκώνα και καρπό αλληλένδετα όπως ενός ανθρώπινου χεριού.

Για την επιλογή και την υλοποίηση της κατασκευής του συγκεκριμένου βραχίονα υπήρξε επισκεψιμότητα στην ιστοσελίδα <http://luckylarry.co.uk/arduino-projects/arduino-robot-arm-larryarm-v0-1/> , προκειμένου να πραγματοποιηθούν τα παραδείγματα που αναλύονται στα επόμενα κεφάλαια.

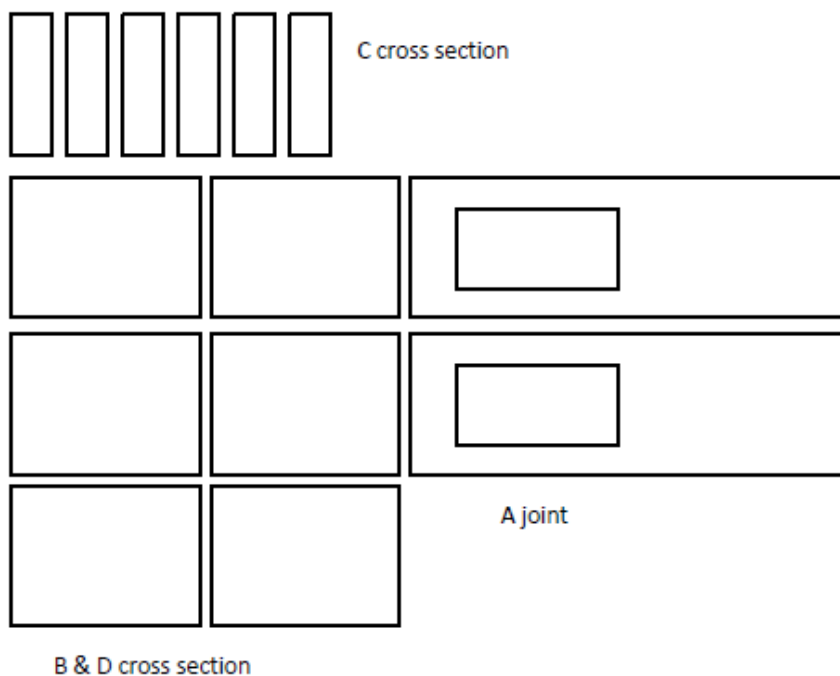
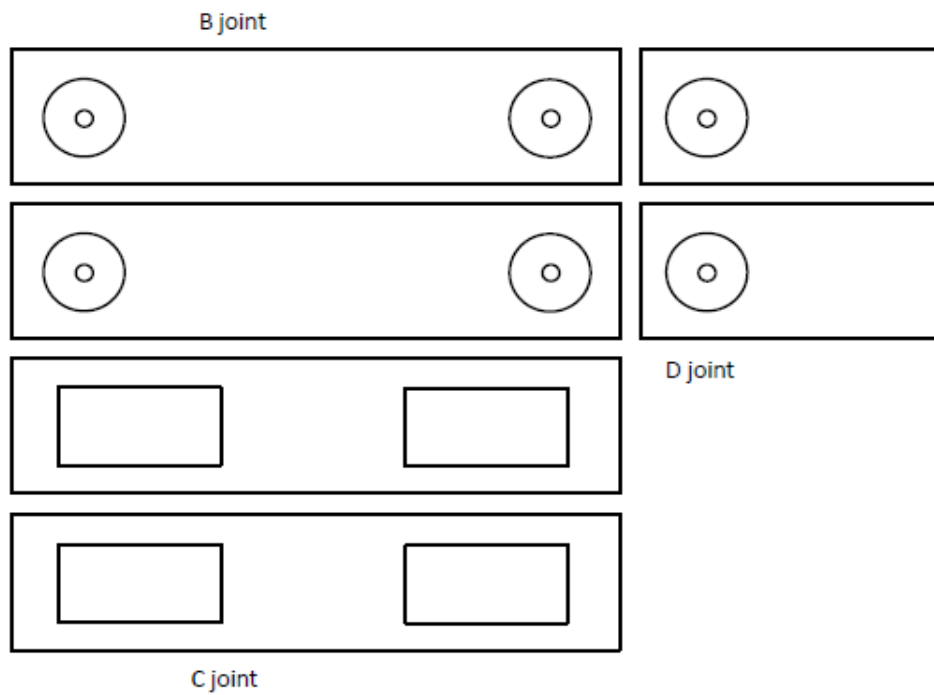
Στην ιστοσελίδα υπήρχαν έτοιμα σχέδια σε μορφή αρχείου PDF με τις διαστάσεις των ξύλων, προ συναρμολόγησης, κλίμακας 1:1.

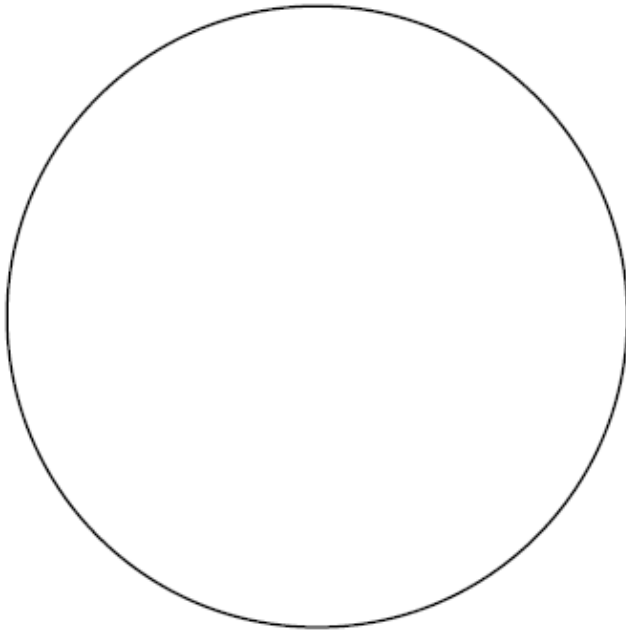
Οι παράγοντες που λήφθηκαν σοβαρά υπόψη κατά τον σχεδιασμό και την υλοποίηση ήταν οι εξής:

- Το βάρος ολόκληρου του σκελετού που θα πρέπει να είναι μικρό ώστε να μην επιβαρύνεται η κίνηση των σερβοκινητήρων
- Η στιβαρότητα της βάσης

Η επιλογή του υλικού έγινε με βάση το βάρος προκειμένου να μην επιβαρύνονται οι σερβοκινητήρες και ο σκελετός του βραχίονα κατά την κίνησή τους. Με βάση αυτό το κριτήριο επιλέχθηκε ξύλο πεπιεσμένου χάρτου πάχους 4χιλ.

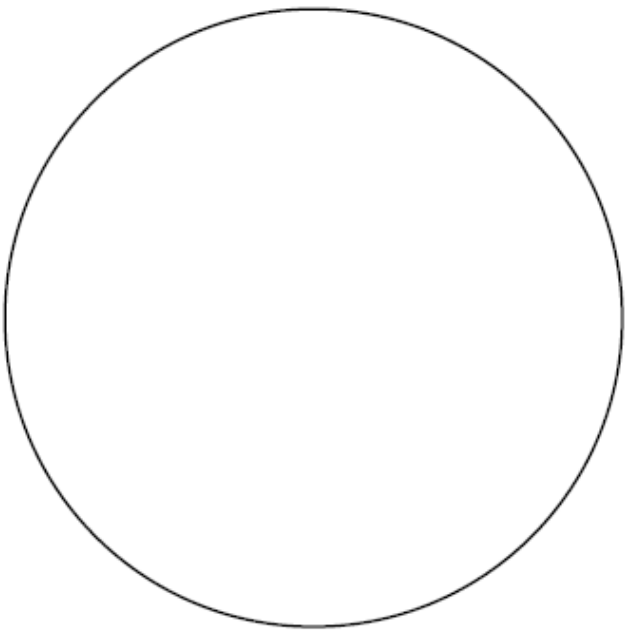
4.2 Σχέδια κατασκευής του βραχίονα





Base

A & D spacers

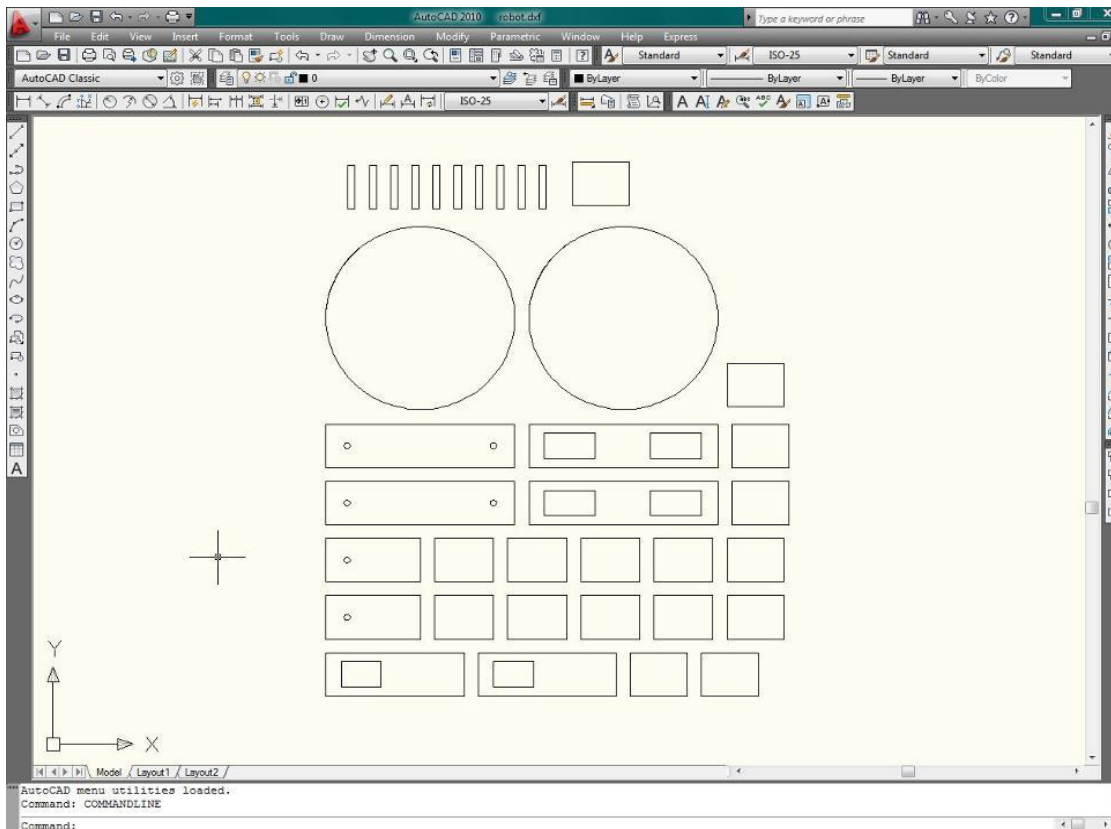


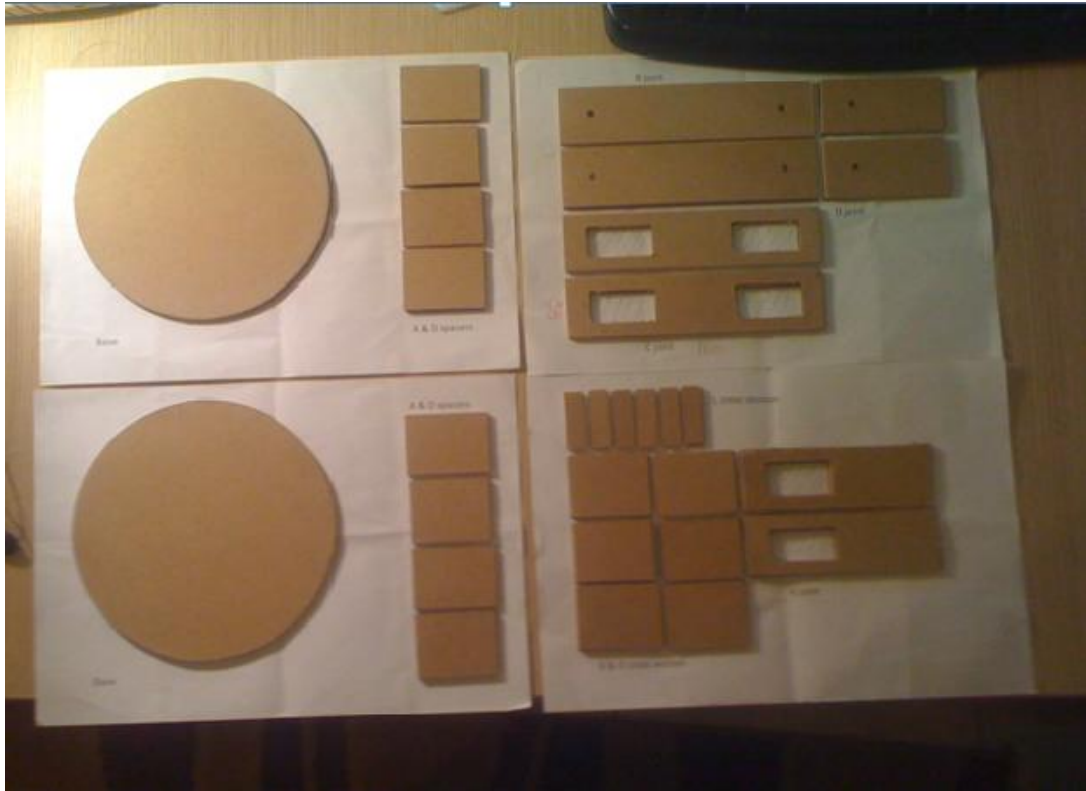
Base



A & D spacers

Σύμφωνα με τα σχέδια του αρχικού κατασκευαστή σχεδιάστηκαν στην ίδια κλίμακα 1:1 σε κατάλληλο λογισμικό AUTOCAD προκειμένου να κοπούν με ακρίβεια σε μηχάνημα CNC φρέζα τριών αξόνων.





4.3 Μελέτη και βήματα

Για να υλοποιηθεί η κατασκευή του βραχίονα έπρεπε να μελετηθούν τα σχέδια και οι εικόνες του ολοκληρωμένου βραχίονα από την ιστοσελίδα <http://luckylarry.co.uk/arduino-projects/arduino-robot-arm-larryarm-v0-1/>.

Σύμφωνα με τα παραπάνω σχέδια πραγματοποιήθηκαν τα βήματα συναρμολόγηση ως εξής:

1ο Βήμα

Τοποθέτηση σερβοκινητήρα στα τμήματα A joint με τρία A&D spacers.

Κατά την τοποθέτηση των σερβοκινητήρων υπήρξε δυσκολία κατά το πέρασμα των καλωδίων τους, χρησιμοποιώντας μαχαιράκι ακρίβειας μοντελισμού ανοίχθηκε λίγο το διάστημα στα άκρα των A joint. Για την συναρμολόγηση χρησιμοποιήθηκε στην αρχή πιστόλι θερμικής κόλλας. Μετά από έλεγχο διαπιστώθηκε ότι δεν ήταν η κατάλληλη επιλογή για να δέσουν τα μέλη της

κατασκευής και για το λόγο αυτό χρησιμοποιήθηκε εποξική 2 στοιχείων 5 λεπτών Pattex Power Epoxy.

2ο Βήμα

Τοποθέτηση των σερβοκινητήρων στα τμήματα C joint με τρία C cross section.

Πραγματοποιήθηκε και σε αυτό το βήμα άνοιγμα χρησιμοποιώντας μαχαιράκι ακρίβειας μοντελισμού ώστε να περαστούν με ευκολία τα καλώδια των σερβοκινητήρων. Η κόλληση των τμημάτων πραγματοποιήθηκε επίσης με εποξική 2 στοιχείων 5 λεπτών Pattex Power Epoxy.

3ο Βήμα

Τοποθέτηση των στρόγγυλων πλαστικών κεφαλών στα τμήματα B joint & D joint ακριβώς στα σημεία που έχουν σχεδιαστεί κατά την κοπή τους και που αποτελούν την επάνω όψη του βραχίονα, προκειμένου να εφαρμοστούν με ακρίβεια στις περιστροφικές κεφαλές των σερβοκινητήρων.

Η κόλληση των τμημάτων πραγματοποιήθηκε επίσης με εποξική 2 στοιχείων 5 λεπτών Pattex Power Epoxy.

4ο Βήμα

Τοποθέτηση των B&D cross section στα πάνω τμήματα όψης του βραχίονα B joint & D joint αντίστοιχα με σκοπό την δημιουργία κατάλληλης σταθερής απόστασης μεταξύ τους.

Σε αυτό το βήμα χρησιμοποιήθηκε τριγωνικό χαράκι προκειμένου να σχηματιστεί ορθή γωνία κατά την τοποθέτηση των B&D cross section κατά την τοποθέτηση τους ανάμεσα στα τμήματα B joint & D joint. Η κόλληση των τμημάτων πραγματοποιήθηκε επίσης με εποξική 2 στοιχείων 5 λεπτών Pattex Power Epoxy.

5ο Βήμα

Τοποθέτηση βιδών στους σερβοκινητήρες.

Προστέθηκαν βίδες στην από κάτω τους πλευρά, απέναντι ακριβώς από το κέντρο των περιστροφικών κεφαλών, ώστε να σχηματιστεί ένας νοητός άξονας, που θα διαπερνά κάθετα το επίπεδο περιστροφής του δίσκου. Αυτό έχει σαν αποτέλεσμα την σταθεροποίηση της αρθρωσης ως προς τον άξονα περιστροφής και κατ' επέκταση πιο σταθερή – ανθεκτική, με πλήρη ακρίβεια κίνησης, κατασκευή. Η κόλληση των βιδών πραγματοποιήθηκε επίσης με εποξική 2 στοιχείων 5 λεπτών Pattex Power Epoxy και για την εύρεση των σημείων κόλλησης χρησιμοποιήθηκαν χάρακας καθώς και ανεξίτηλο μαρκαδόρακι για την σημείωση αυτών.

6ο Βήμα

Κόλληση των μελών B&D cross section με τα μέλη του βραχίονα B joint & D joint αντίστοιχα.

Αρχικά τοποθετήθηκαν κατάλληλα οι πλαστικές κεφαλές που είχαν κολληθεί στα επάνω τμήματα B joint & D joint στους σερβοκινητήρες και έπειτα περάστηκαν τα κάτω τμήματα B joint & D joint κατάλληλα στα σημεία που εφάπτονται με τις βίδες. Με το πέρασμα των κάτω τμημάτων στις βίδες έγινε παράλληλα και η κόλληση στην από κάτω πλευρά των B&D cross section αντίστοιχα. Η κόλληση των τμημάτων πραγματοποιήθηκε επίσης με εποξική 2 στοιχείων 5 λεπτών Pattex Power Epoxy.

7ο Βήμα

Δημιουργία της Βάσης του βραχίονα με τα Base κομμάτια.

Στην πλευρά που κολλήθηκαν τα τρία A&D spacers στα τμήματα A joint προστέθηκαν από την επάνω τέσσερα A&D spacers και από την κάτω τρία. Αυτό είχε σαν αποτέλεσμα τον σωστό καθορισμό του ύψους της βάσης κατά την

τοποθέτηση των Base τμημάτων με την κόλληση τους στην πάνω και κάτω πλευρά των A&D spacers. Η κόλληση των τμημάτων πραγματοποιήθηκε με εποξική 2 στοιχείων 5 λεπτών Pattex Power Epoxy και επιπλέον δύο καρφιά με καρφωτικό αέρος για το καλύτερο δέσιμο αυτών.

8ο Βήμα

Προσθήκη επιπρόσθετου καλωδίου στο σερβοκινητήρα που αντιστοιχεί στον καρπό του βραχίονα.

Το καλώδιο του σερβοκινητήρα που αντιστοιχεί στον καρπό του βραχίονα έπρεπε να επεκταθεί διότι η απόσταση μέχρι την βάση σε ευθεία κίνηση του βραχίονα, αποδείχτηκε μεγαλύτερη από ότι το μήκος του καλωδίου. Για το λόγω αυτό δημιουργήθηκε καλώδιο "extension" ίδιου τύπου 14 εκατοστών.

9ο Βήμα

Τρίψιμο με γυαλόχαρτο και Χρωματισμός Βραχίονα.

Κατά την ολοκλήρωση του βραχίονα, με ένα κομμάτι γυαλόχαρτο smirnex finishing paper P320 αφαιρέθηκαν οι κόλλες που προεξείχαν και στη συνέχεια με πινέλα Νο 6 και Νο 8, χρωματίστηκε με πλαστικό χρώμα μπλε για λόγους καλαισθησίας.

10ο Βήμα

Συνδεσμολογία του μικροελεγκτή Arduino Uno με τον βραχίονα.

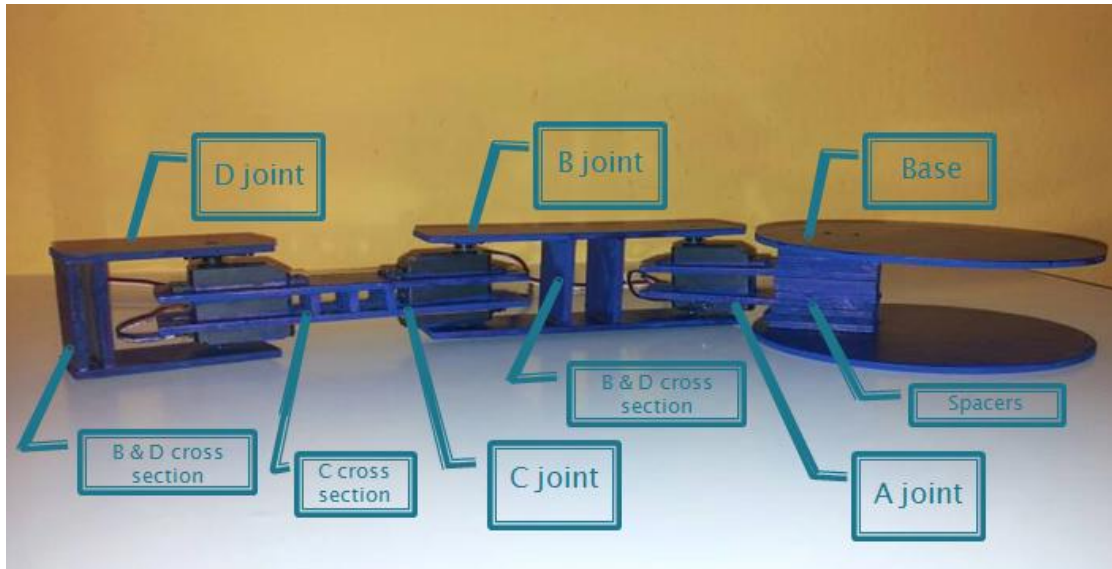
Σύνδεση των τερματικών απολήξεων των σερβοκινητήρων επάνω στο raster του arduino uno σε κάθετη διάταξη. Τα χρώματα των καλωδίων αντιστοιχίζονται ως εξής:

- Μαύρο -> Γείωση
- Κόκκινο -> Τάση 5V
- Λευκό -> Σήμα

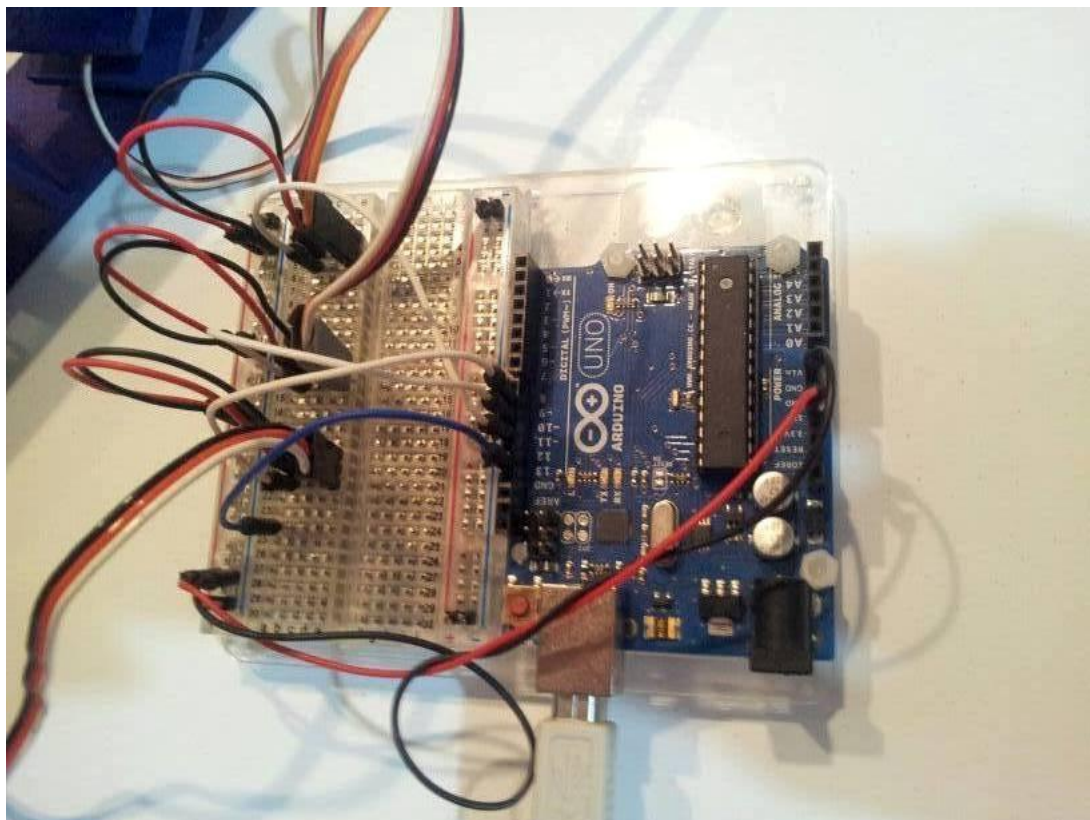
Στα αντίστοιχα πιν των απολήξεων των σερβοκινητήρων επάνω στο ράστερ προστέθηκαν ακριβώς στην ίδια σειρά, ιδίου χρώματος καλώδια, για διευκόλυνση που καταλήγουν στις αντίστοιχες θέσεις του ράστερ (γείωση στο - και τάση στο +5V) καθώς και στις αντίστοιχες ψηφιακές θέσεις σήματος.

<i>Σερβοκινητήρας</i>	<i>Θέση που Αντιστοιχεί στο βραχίονα</i>	<i>Ψηφιακή Θέση επάνω στον μικροελεγκτή</i>
<i>Servo1</i>	<i>Ωμος</i>	<i>9</i>
<i>Servo2</i>	<i>Αγκώνας</i>	<i>10</i>
<i>Servo3</i>	<i>Καρπός</i>	<i>11</i>

Στο τέλος συνδέονται δύο επιπλέον καλώδια στην γείωση και στην τάση του raster με τις αντίστοιχες θέσεις γείωσης και τάσης του μικροελεγκτή Arduino Uno (GND & 5V) και το καλώδιο τροφοδοσίας του Arduino Uno με θύρα USB του ηλεκτρονικού υπολογιστή.



Εικόνα 4.1 Επίπεδος ρομποτικός βραχίονας.



Εικόνα 4.2 Συνδεσμολογία του ρομποτικού βραχίονα στην πλακέτα του μικροελεγκτή

ΚΕΦΑΛΑΙΟ 5^ο

5.ΥΛΟΠΟΙΗΣΗ ΠΕΙΡΑΜΑΤΩΝ ΣΕ ΓΛΩΣΣΑ WIRING

5.1 Υλοποίηση 1^{ου} Πειράματος-Χειρισμός του βραχίονα μέσω Η/Υ

Το πρώτο πείραμα αφορά τον χειρισμό του βραχίονα από τον χρήστη μέσω ηλεκτρονικού υπολογιστή. Ο χρήστης μπορεί μέσω της σειριακής οθόνης να δίνει μοίρες για κάθε έναν σερβοκινητήρα τη φορά, π.χ. (1,50)enter ή (2,80)enter ή (3,90)enter (όπου αριστερά του διαχωριστή ‘,’ ο αριθμός του σερβοκινητήρα και δεξιά οι μοίρες). Στη συνέχεια το αντίστοιχο κομμάτι του βραχίονα εκτελεί την κίνηση σύμφωνα με τις μοίρες που έλαβε πηγαίνοντας στο αντίστοιχο σημείο.

Κώδικας 1^{ου} πειράματος σε γλώσσα Wiring

```
#include <VarSpeedServo.h> // βιβλιοθήκη που περιέχει τα αντικείμενα servo και
// συναρτήσεις για την χρήση των σερβοκινητήρων
VarSpeedServo myservo1; //δήλωση αντικειμένου servo με όνομα myservo1,2,3
// για τον έλεγχο των σερβοκινητήρων
VarSpeedServo myservo2;
VarSpeedServo myservo3;
char incomingByte; //δήλωση μεταβλητής τύπου χαρακτήρα
int i=0; // δήλωση μεταβλητών ακέραιου τύπου

int ServoValue=0; // μεταβλητή ακεραίου τύπου που θα κρατάει την τιμή του
// σερβοκινητήρα
int Degrees=0; // δήλωση μεταβλητής στην οποία αποθηκεύονται οι μοίρες

//συνάρτηση που παίρνει έναν χαρακτήρα και επιστρέφει τον αντίστοιχο του
//ακέραιο.

int AltoIn(char inPut)
{
    int outPut = 0;
    outPut = (int) inPut;
    outPut = inPut - 48;
    return outPut;
}
//συνάρτηση που καλείται σε περίπτωση που ο χρήστης δώσει λάθος δεδομένα
void ResetReceive()
```



```

{
//εμφάνιση μηνύματος στην σειριακή οθόνη
Serial.print("\nYou Give Wrong Value! Must Be Like : 1,150");
Serial.flush(); //ανακτά τα σειριακά δεδομένα από το Buffer
}
//συνάρτηση για τις εισερχόμενες μοίρες
int GetDegrees(int pos, int inByte)
{
char incoming; // δήλωση αλφαριθμητικού
int value[2]={0,0}; //δήλωση πίνακα τριών θέσεων
//δημιουργία ακεραίου Dgrs και εκχώρηση της τιμής της μεταβλητής inByte
int Dgrs=inByte;
if( Serial.available() )
{
pos++; //αύξηση θέσης κατά ένα
value[0] = inByte;
//ανάγνωση τέταρτου ληφθέντος χαρακτήρα
incoming = Serial.read();
inByte = AltoIn(incoming);
if(pos==3 && (inByte>=0 && inByte<10) )
{

if( Serial.available() ) // η συνθήκη ελέγχει αν υπάρχουν δεδομένα στον buffer της
σειριακής θύρας.

{
pos++;
value[1] = inByte;

incoming = Serial.read(); //διαβάζει τα σειριακά δεδομένα σαν ροή δεδομένων
που υπάρχουν στην σειριακή θύρα

inByte = AltoIn(incoming); //επειδή η serial.read επιστρέφει αλφαριθμητικό, το
μετατρέπει σε ακέραιο.

if(pos==4 && (inByte>=0 && inByte<10) )
{
// μετατροπή σε μοίρες
Dgrs = (value[0]*100) + (value[1]*10) + inByte;
if(Dgrs>=0 && Dgrs<=180)
{
return Dgrs; // επιστροφή μοιρών
}
}
else{ ResetReceive();
}
}
else{ ResetReceive(); }
}
else
{

```

```

Dgrs = (value[0]*10) + value[1]; // αν οι μοίρες είναι δεκαδικής μορφής πχ 85
return Dgrs;
}
}
else{ ResetReceive(); }
}
else{ return Dgrs; }
}

```

```

void ServoPosition(int Servo, int Position) //παίρνει ως όρισμα τον κινητήρα και
την θέση του και δίνει εντολή στον arduino να δώσει έξοδο την θέση

```

```

{
switch (Servo) // κατευθύνει την ροή του προγράμματος
{
case 1 :
myservo1.slowmove(Position,20);
break;
case 2 :
myservo2.slowmove(Position,20);
break;
default :
myservo3.slowmove(Position,20);
break;
}
}

```

```

//Αρχικοποιούμε την βασική ρουτίνα η οποία εκτελείται μόνο μία φορά κατά την
εκκίνηση του προγράμματος.

```

```

void setup() // συνάρτηση αρχικοποίησης
{
Serial.begin(2400);
Serial.flush(); // ανακτά τα δεδομένα από το Buffer
//εμφάνιση μηνύματος στην οθόνη
Serial.print("<<<----- ROBOT ARM ----->>>\n");
Serial.print("<<<----- A.T.E.I. SERRWN ----->>>\n");
//Δήλωση θέσεων των σερβοκινητήρων ε πάνω στις ψηφιακές θέσεις του
μικροελεγκτή
myservo1.attach(9);
myservo2.attach(10);
myservo3.attach(11);

}

```

`void loop()` //περιλαμβάνει τον κορμό του προγράμματος, αφού εκτελεστεί η συνάρτηση αρχικοποίησης(*Setup*) η *loop* εκτελείται συνεχώς από την στιγμή που παίρνει ρεύμα το *arduino*.

```
{
i=0; //χρησιμοποιείται για την θέση μέσα στον πίνακα
if ( Serial.available() ) //έλεγχος διαθέσιμης σειριακής θύρας κατά τη λήψη
{
//ανάγνωση πρώτου χαρακτήρα που λαμβάνει από τη σειριακή
incomingByte =Serial.read();
//μετατροπή από αλφαριθμητικό σε ακέραιο
ServoValue = AltoIn(incomingByte);
// έλεγχος για τον αριθμό των servo 1-3
if ( i==0 && (ServoValue>0 && ServoValue<4) )
{
delay(300); // Σταματάει το πρόγραμμα για το χρονικό διάστημα 0,3 (σε
milliseconds)
if ( Serial.available() )
{
i++;
incomingByte =Serial.read(); // ανάγνωση δευτέρου ληφθέντος χαρακτήρα
if ( i==1 && incomingByte=='\'' ) //έλεγχος διαχωριστή
{
delay(300);
if ( Serial.available() )
{
i++;
incomingByte =Serial.read();
Degrees = AltoIn(incomingByte);
// ανάγνωση τρίτου ληφθέντος χαρακτήρα , πρέπει να είναι αριθμοί μόνο
ανάμεσα 0-9
if ( i==2 && Degrees >=0 && Degrees<10)
{
Degrees=GetDegrees(i, Degrees);
Serial.print("\nServo :");
Serial.print(ServoValue);
Serial.print("\nDegrees :");
Serial.print(Degrees);
delay(10);
ServoPosition(ServoValue, Degrees);
}
else
{
ResetReceive(); // μήνυμα λάθους
goto end; //έξοδος από την Loop
}
}
else
{
ResetReceive();
```

```

goto end; //Έξοδος από την Loop
}
}
else
{
ResetReceive();
goto end; //Έξοδος από την Loop
}
}
else
{
ResetReceive();
goto end; //Έξοδος από την Loop
}
}
end:
delay(1000);
}
}

```

5.2 Υλοποίηση 2ου Πειράματος – Χειρισμός βραχίονα με κλειστή διαδρομή κίνησης

Ο χρήστης μπορεί μέσω της σειριακής οθόνης να δίνει μοίρες για 2 σετ κινήσεων των σερβοκινητήρων π.χ.

1^ο σετ 1,50;2,50;3,50;enter

2^ο σετ 1,80;2,80;3,80;enter

και ο βραχίονας θα εκτελεί τις κινήσεις ανάμεσα στις 50 και στις 180 μοίρες έως ότου δοθούν εκ νέου τα επόμενα σετ μοιρών από τον χρήστη.

Σημειώνεται ότι δεν είναι απαραίτητη προϋπόθεση ο αριθμός των σέρβο μέσα σε ένα σετ μοιρών να είναι με αριθμητική σειρά π.χ.

1^ο σετ 3,50;1,20;2,80; enter

2^ο σετ 1,50;2,10;3,90; enter

Επίσης υπάρχει δυνατότητα επιλογής των σερβοκινητήρων που θέλει ο χρήστης να κινήθούν π.χ.

1^ο σετ 1,10;2,20; enter

2^ο σετ 3,50;1,50; enter

Κώδικας 2^{ου} πειράματος σε γλώσσα Wiring

```
#include <VarSpeedServo.h> // δήλωση βιβλιοθήκης

// δήλωση μεταβλητής αριθμού
#define Movies 2

VarSpeedServo myservo1; //δήλωση των σερβοκινητήρων
VarSpeedServo myservo2;
VarSpeedServo myservo3;

// δήλωση μεταβλητών ακέραιου τύπου
int ServoPos=0;

int DegPos=0;

int Wrong=0;

int Switch=0;

int Packet=0;

int outPacket[Movies];

int inDegrees[Movies][3]; // νέες- εισερχόμενες τιμές
int inServo[Movies][3];

int outDegrees[Movies][3]; // παλιές- εξερχόμενες τιμές
int outServo[Movies][3];
int i=0;

//συνάρτηση που καλείται σε περίπτωση που ο χρήστης δώσει λάθος δεδομένα
void ResetReceive()
{
  Serial.print("\nYou Give Wrong Values!"); //εμφάνιση μηνύματος στην οθόνη
  Serial.print("\nMust Be Like: ServoValue-Degrees;");
  Serial.print("\nFor example : 3,134;");
  Serial.print("\nSeparate from SERVO to DEGREES can be ',' ");
  Serial.print("\nSeparate from DEGREES to SERVO can only be ',' 1,50; 2,100;
3,150;! \n\n");
  //Reset Receive Buffer
  Serial.flush(); //ανακτά τα σειριακά δεδομένα από το Buffer
```

```
}
```

```
//συνάρτηση που μετατρέπει τους αλφαριθμητικούς χαρακτήρες σε ακέραιους.
```

```
int AltoIn(char inPut)
{
  int outPut = 0;
  outPut = (int) inPut;
  outPut = inPut - 48; //ASCII 48 = 0
  return outPut;
}
```

```
//εκτελείται εφόσον όλα τα δεδομένα του χρήστη είναι σωστά για τις κινήσεις
```

```
void ServosPositions(int ServoValue, int Position)
{
  Serial.print("\nServo ");
  switch (ServoValue) // κατευθύνει την ροή του προγράμματος
  {
    case 1 :
      //εμφάνιση σερβοκινητήρα στην σειριακή οθόνη και ρύθμιση της ταχύτητας κίνησης του
      Serial.print(ServoValue); //εμφάνιση μηνύματος
      Serial.print(" - ");
      Serial.print(Position);
      myservo1.slowmove(Position,20);
      break;

    case 2 :
      Serial.print(ServoValue);
      Serial.print(" - ");
      Serial.print(Position);
      myservo2.slowmove(Position,20);

      break;
    default :
      Serial.print(ServoValue);
      Serial.print(" - ");
      Serial.print(Position);
      myservo3.slowmove(Position,20);

      break;
  }
}
```

```
//καθυστέρηση (2*1500=3000 χιλιοστά του δευτερολέπτου
```

```
delay(Movies*1500); }
```

```
void setup()
{
  Serial.begin(2400);
  Serial.flush();
  Serial.print("<<<----- ROBOT ARM ----->>>\n");
  Serial.print("<<<----- A.T.E.I. SERRWN ----->>>\n");
  myservo1.attach(9);
  myservo2.attach(10);
  myservo3.attach(11);
}
```

```
void loop()//περιλαμβάνει τον κορμό του προγράμματος
{
  char incomingByte; //δήλωση μεταβλητής τύπου χαρακτήρα
  int inValue=0; // δήλωση μεταβλητής ακέραιου τύπου

  //έλεγχος για το αν εισαχθεί κάποια μεταβλητή στη διαθέσιμη σειριακή θύρα
  if (Serial.available()>0)
  {
    while(Serial.available())
    {
      //έλεγχος αν έχει πάει κάτι στραβά τα δεδομένα δεν είναι σωστά

      if(Wrong==1) //wrong=1->λάθος
      {
        goto StopLoop; //σταματάει να τρέχει η loop

        i=0; // η μεταβλητή i μηδενίζεται
      }
      incomingByte =Serial.read(); //διαβάζονται τα εισερχόμενα byte

      //τα εισερχόμενα δεδομένα μετατρέπονται σε ακέραιο
      inValue = AltoIn(incomingByte);
    }
  }
}
```

```

if(Switch==0) //αν ο διακόπτης είναι 0 διαβάζει τον σερβοκινητήρα
{

if(incomingByte==' ') //έλεγχος αν ο διαχωριστής είναι το ';'
{

// όταν εισέρχεται ο διαχωριστής ';' ο διακόπτης γίνεται 1 και περιμένει να δοθούν
μοίρες και αυξάνεται η θέση της μεταβλητής ServoPos

Switch=1;
ServoPos++;
}

else

{

// στην περίπτωση που η τιμή που δίνει ο χρήστης για τον σερβοκινητήρα δεν
είναι από 1 έως 3
if (inValue<1 || inValue>3)
{

Wrong=1; //wrong=1->λάθος

i=0; // η μεταβλητή i μηδενίζεται

ResetReceive(); //καλείται η συνάρτηση για μήνυμα λάθους

goto StopLoop; // σταματάει η loop
}

else
{

// έλεγχος αν ο αριθμός του σερβοκινητήρα δόθηκε πάνω από μία φορά για κάθε
σετ τιμών

for(int j=0;j<Packet;j++)
{

if (inValue==inServo[i][j])
{
Serial.print("\nYou chose this servo again!");
i=0;
Wrong=1;
ResetReceive();
}
}
}
}

```



```

goto StopLoop;
}
}
inServo[i][Packet]=inValue;
}
}
}
else //στην περίπτωση που η μεταβλητή switch=1
{

if(inValue==11) //έλεγχος για το αν ο διαχωριστής είναι ';' ASCII->11=';'
{
Packet++; //αυξάνει κατά ένα για το επόμενο σετ μοιρών
Switch=0; // επιστρέφει σε 0 ->περιμένει για αριθμό σερβοκινητήρα
}

//ο αριθμός της εισερχόμενης ακέραιης μεταβλητής για τις μοίρες πρέπει να είναι
από 0 έως 9
else if(inValue>=0 && inValue<=9)

{
inDegrees[i][Packet]=inValue+(inDegrees[i][Packet] * 10);
DegPos++;

//έλεγχος αν οι μοίρες είναι από 0 έως 180
if(inDegrees[i][Packet]<0 || inDegrees[i][Packet]>180)

{
i=0;
Wrong=1;
ResetReceive();
goto StopLoop;
}

}
else //για την inValue=11
{
i=0;
Wrong=1;
ResetReceive();
goto StopLoop;
}

}

} // Τέλος της WHILE

// έλεγχοι:

```

```
//αν δοθεί σερτ 1,50 χωρίς ερωτηματικό  
//αν τα σερτ δεν είναι ίδια σε αριθμό με τις μοίρες π.χ 1,10;2;
```

```
if( (Packet>0 && inValue!=11) || Packet>DegPos || Packet!=ServoPos)  
{  
Wrong=1;  
ResetReceive();  
goto StopLoop;  
}  
if(Wrong==1)  
{  
goto StopLoop;  
}  
else  
{  
for(int j=0;j<Packet;j++)  
{  
outDegrees[i][j]=inDegrees[i][j];  
outServo[i][j]=inServo[i][j];  
outPacket[i]=Packet;  
}  
}
```

```
if(i==Movies-1) // αν οι κινήσεις δεν είναι 2  
{  
i=0;  
goto StopLoop;  
}  
i++;  
StopLoop:  
delay(100);  
  
}  
else  
{  
delay(Movies*1500);
```

```
//εκτέλεση κινήσεων
```

```
for(int k=0;k<Movies;k++)  
{  
if(i==0 && outPacket[k]>0)  
{  
Serial.print("\n[-----> Position ");  
Serial.print(k+1);  
Serial.print(" <-----]");  
for(int j=0;j<outPacket[k];j++)
```

```
{
  ServosPositions(outServo[k][j],outDegrees[k][j]);
}
}
}
//αρχικοποίηση μεταβλητών

for(int k=0;k<Movies;k++)
{
  for(int j=0;j<outPacket[k];j++)
  {
    inDegrees[k][j]=0;
    inServo[k][j]=0;
  }
}
DegPos=0;
Packet=0;
inValue=0;
Wrong=0;
ServoPos=0;
}
} //τέλος της LOOP
```

Υποσημειώσεις – επεξηγήσεις

Πίνακας 5.1 ASCII-DEC

ASCII	DEC	outPut = inPut - 48;
0	48	48-48=0;
1	49	49-48=1;
2	50	50-48=2;
3	51	51-48=3;
4	52	52-48=4;
5	53	53-48=5;
6	54	54-48=6;
7	55	55-48=7;
8	56	56-48=8;
9	57	57-48=9
;	59	59-48=11=;

«Παρουσιάζονται μόνο οι τιμές και τα σύμβολα που μετατρέπονται από τύπο χαρακτήρα σε τύπο ακέραιου, στους συγκεκριμένους κώδικες από την συνάρτηση AltoIN»

Παραδείγματα υπολογισμού μοιρών

$\text{inDegrees}[i][\text{Packet}] = \text{inValue} + (\text{inDegrees}[i][\text{Packet}] * 10);$

1^ο παράδειγμα

έστω π.χ. δίνει ο χρήστης 9 μοίρες

$\text{inDegrees}[0][0] = 0; \text{inValue} = 9; \text{inDegrees}[0][0] = 9 + (0 * 10);$

→ $\text{inDegrees}[0][0] = 9;$

2^ο παράδειγμα

έστω π.χ. δίνει ο χρήστης 75 μοίρες

1^η ΕΠΑΝΑΛΗΨΗ

$\text{inDegrees}[0][0] = 0; \text{inValue} = 7; \text{inDegrees}[0][0] = 7 + (0 * 10);$

→ $\text{inDegrees}[0][0] = 7;$

2^η ΕΠΑΝΑΛΗΨΗ

$\text{inDegrees}[0][0] = 7; \text{inValue} = 5; \text{inDegrees}[0][0] = 5 + (7 * 10);$

→ $\text{inDegrees}[0][0] = 5 + 70 = 75;$

3^ο παράδειγμα

έστω π.χ. δίνει ο χρήστης 175 μοίρες

1^η ΕΠΑΝΑΛΗΨΗ

$\text{inDegrees}[0][0] = 0; \text{inValue} = 1; \text{inDegrees}[0][0] = 1 + (0 * 10);$

→ $\text{inDegrees}[0][0] = 1;$

2^η ΕΠΑΝΑΛΗΨΗ

$\text{inDegrees}[0][0] = 1; \text{inValue} = 7; \text{inDegrees}[0][0] = 7 + (1 * 10);$

→ $\text{inDegrees}[0][0] = 7 + 10 = 17;$

3^η ΕΠΑΝΑΛΗΨΗ

$\text{inDegrees}[0][0] = 17; \text{inValue} = 5; \text{inDegrees}[0][0] = 5 + (17 * 10);$

→ $\text{inDegrees}[0][0] = 5 + 170 = 175;$

Προβλήματα κατά την εκτέλεση

Κατά την δημιουργία των προγραμμάτων ελέγχου των σερβοκινητήρων αντιμετωπίσαμε διάφορα προβλήματα. Ένα από τα ποία σημαντικά ήταν η ρύθμιση της ταχύτητας των σέρβο. Για αυτό τον σκοπό χρησιμοποιήσαμε την βιβλιοθήκη **<VarSpeedServo.h>** αντί της **<servo.h>** , προκειμένου να λυθεί το πρόβλημα.

Το πρόβλημα προέκυψε εξαιτίας του υψηλού φορτίου το οποίο μεταφέρεται κατά τη λειτουργία του βραχίονα, καθώς και της απότομης επιτάχυνσης των σερβοκινητήρων τη στιγμή κατά την οποία στέλνονται τα δεδομένα από την σειριακή οθόνη του υπολογιστή , με αποτέλεσμα να μην προλαβαίνει ο χρήστης να αντιληφθεί την κίνηση του βραχίονα.

Ένα ακόμα πρόβλημα που αντιμετωπίσαμε ήταν η μη σωστή λειτουργία/σύνδεση της θύρας(COM) του υπολογιστή με την πλακέτα του Arduino Uno, λόγω της γρήγορης ταχύτητας με την οποία τα σέρβο του βραχίονα λάμβαναν και εκτελούσαν τα δεδομένα, με αποτέλεσμα η τάση τροφοδοσίας να μην επαρκή και για τα τρία σέρβο.

Τέλος χρειάστηκε να γίνει αλλαγή του μεσαίου σερβοκινητήρα του βραχίονα, διότι από τις απότομες επιταχύνσεις, ενδεχομένως πειράχτηκαν τα γρανάζια του, με συνέπεια να την ολική καταστροφή του.

Για να αντιμετωπίσουμε λοιπόν όλα αυτά τα προβλήματα χρησιμοποιήσαμε την βιβλιοθήκη, την οποία αναφέραμε παραπάνω, όπου και κατεβάσαμε από στην ιστοσελίδα

<http://www.thebestcasescenario.com/?q=integrating+electronics+part+2>

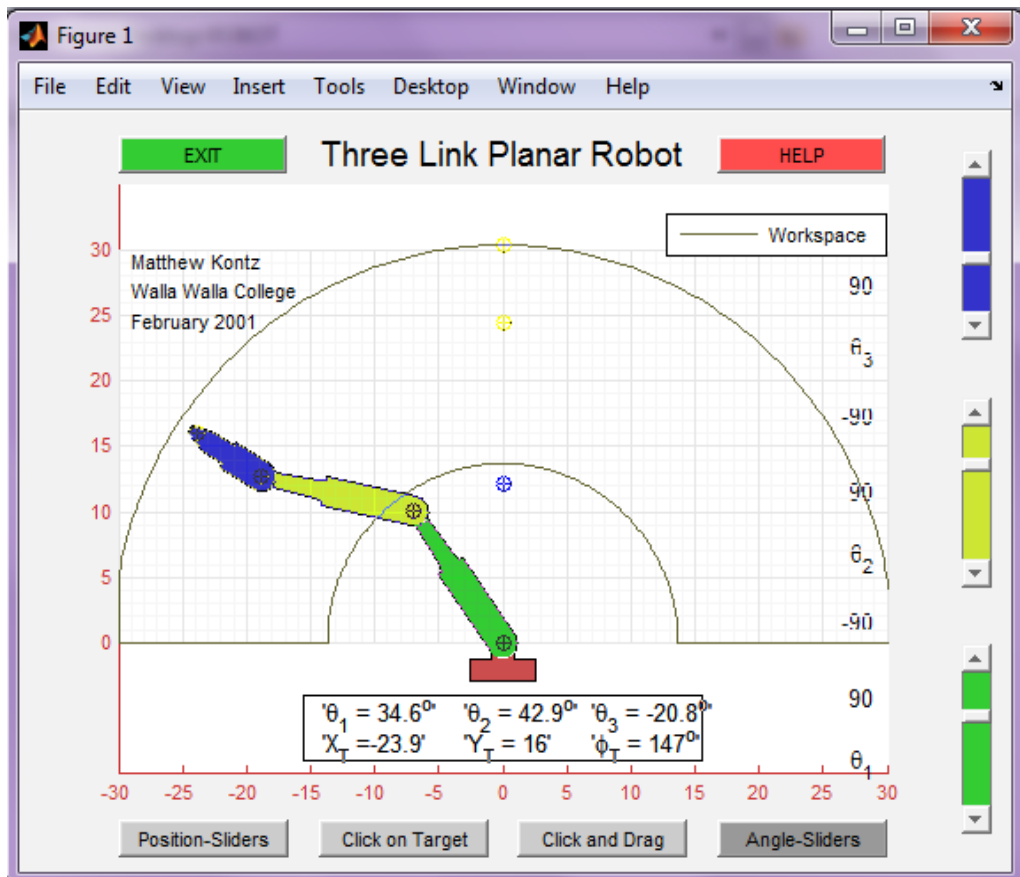
σύμφωνα με την οποία μπορούμε να ελέγξουμε την ταχύτητα περιστροφής του κάθε σέρβο.

5.3 Ανάπτυξη γραφικού περιβάλλοντος διεπαφής για την μελέτη του ευθέως κινηματικού προβλήματος στο περιβάλλον του matlab

Η γραφική διεπαφή που χρησιμοποιήσαμε για την επίλυση του ευθέως καθώς και του αντίστροφου κινηματικού προβλήματος, για το οποίο γίνεται αναφορά στην επόμενη ενότητα, υπάρχει έτοιμη στο διαδίκτυο στην σελίδα <http://www.mathworks.com/matlabcentral/fileexchange/157-robot-demo> απ' όπου και την κατεβάσαμε. Πρόκειται για μια demo εφαρμογή, όπου για να εκτελεστεί το αρχείο demorobot απαιτείται η κλήση πέντε συναρτήσεων, δύο από τις οποίες είναι οι `forkin()` και `invkin()`, που επιλύουν το ευθύ και το αντίστροφο κινηματικό πρόβλημα αντίστοιχα.

Αυτό που κάναμε εμείς ήταν να προσαρμόσουμε τον βραχίονα της εξομοίωσης σύμφωνα με τα χαρακτηριστικά του δικού μας βραχίονα (αλλάζοντας τις διαστάσεις των συνδέσεων) προκειμένου να έχουμε τα επιθυμητά αποτελέσματα.

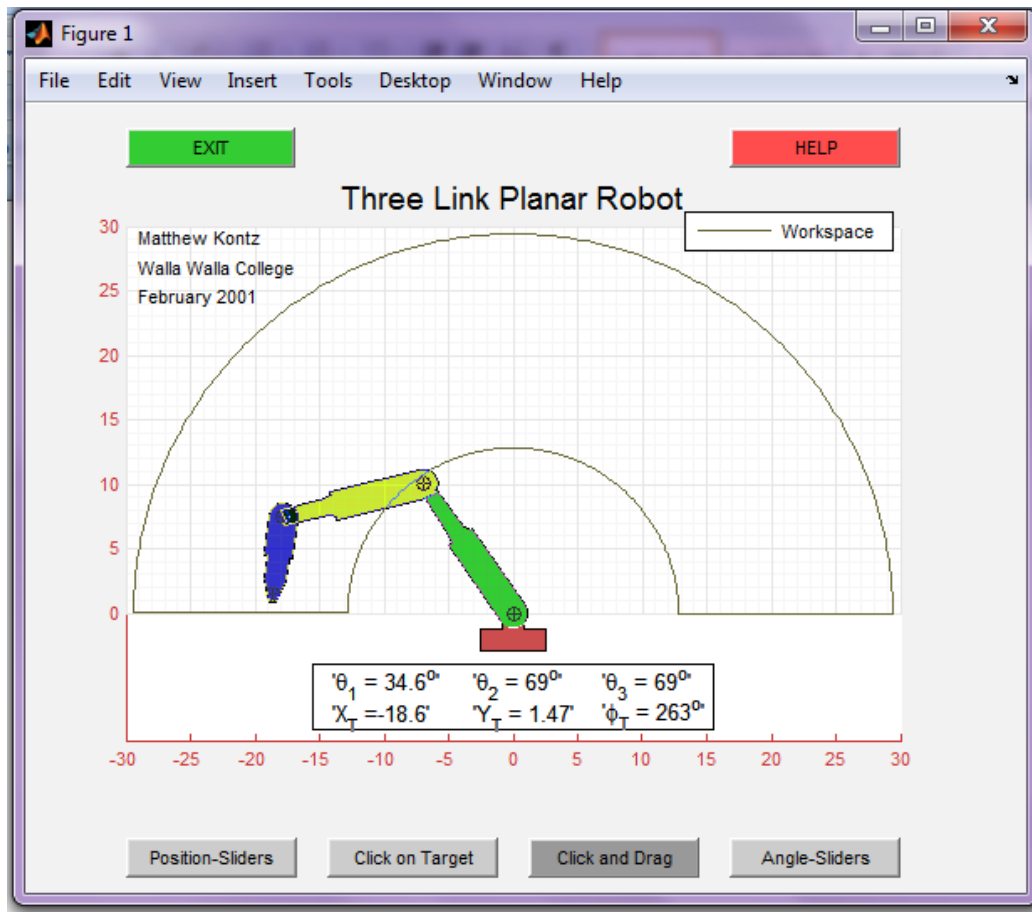
Στο σχήμα 5.3 παρουσιάζεται το γραφικό περιβάλλον διασύνδεσης χρήστη για την μελέτη του ευθέως κινηματικού προβλήματος σε περιβάλλον προσομοίωσης. Με τη χρήση του γραφικού περιβάλλοντος δίνεται η δυνατότητα στον χρήστη να εισάγει από τις μπάρες ολίσθησης τις τιμές των αρθρώσεων και με αυτές να υπολογίσει τις συντεταγμένες του εργαλείου τελικής δράσης.



Σχήμα 5.3: Γραφική διεπαφής χρήστη σε περιβάλλον προσομοίωσης ευθέως κινηματικού προβλήματος

5.4 Ανάπτυξη γραφικού περιβάλλοντος διεπαφής για την μελέτη του αντίστροφου κινηματικού προβλήματος στο περιβάλλον του matlab

Για την μελέτη του αντίστροφου κινηματικού προβλήματος αναπτύχθηκε ένα γραφικό περιβάλλον διασύνδεσης χρήστη σε περιβάλλον προσομοίωσης, που παρουσιάζεται στο Σχήμα 5.4. Το συγκεκριμένο πείραμα δίνει την δυνατότητα στον χρήστη να εισάγει τις τιμές των συντεταγμένων του εργαλείου τελικής δράσης και για κάθε τιμή συντεταγμένων, υπολογίζονται οι τιμές των αρθρώσεων του εικονικού βραχίονα.



Σχήμα 5.4 Γραφική διεπαφή χρήστη, προσομοίωση αντίστροφου κινηματικού

Είναι σημαντικό να αναφέρουμε πως οι είσοδοι που παίρνει η συνάρτηση `invkin()` είναι η ακτίνα (r) και η γωνία (φ), όπου r και φ είναι οι ισοδύναμες πολικές συντεταγμένες των x και y σε καρτεσιανό σύστημα συντεταγμένων. Το ίδιο ισχύει φυσικά και για την συνάρτηση `forkin()`, όπου οι συντεταγμένες τις οποίες υπολογίζει και επιστρέφει είναι και αυτές σε πολικό σύστημα συντεταγμένων.

ΚΕΦΑΛΑΙΟ 6^ο

6.1 Σχόλια-Συμπεράσματα

Τα πειράματα ελέγχου που παρουσιάστηκαν σ' αυτό το κεφάλαιο βασίζονται στη ύπαρξη ιδανικών μειωτήρων και στην πλήρη αντιστάθμιση των δυναμικών μη γραμμικών όρων του βραχίονα. Στην πράξη δεν είναι δυνατόν να επιτευχθεί πλήρης αντιστάθμιση των όρων αυτών, διότι η γνώση μας για τις μηχανικές παραμέτρους του βραχίονα είναι ατελής και η ύπαρξη δυναμικών όρων που έχουν αγνοηθεί στην μοντελοποίηση του βραχίονα είναι πιθανή. Έτσι το μοντέλο του βραχίονα είναι γνωστό ως ένα βαθμό βεβαιότητας. Επιπλέον ο υπολογισμός των δυναμικών όρων σε πραγματικό χρόνο σημαίνει υπολογισμό των περίπλοκων όρων σε χρόνους δειγματοληψίας της τάξης των milliseconds, γεγονός που δημιουργεί σημαντικές απαιτήσεις στο υλικό και το λογισμικό του ελεγκτή του βραχίονα. Συνήθως, το υπολογιστικό φορτίο μειώνεται με τον υπολογισμό μόνο των περισσότερο απαραίτητων δυναμικών όρων.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Ε.ΚΑΝΑΡΑΧΟΣ Ανδρέας-Καθηγητής Ε.Μ.Π, Μηχανισμοί και Ρομποτικά Συστήματα, Εκδόσεις Παπασωτηρίου, 1996

JOHN GRAIN, ΕΙΣΑΓΩΓΗ ΣΤΗ ΡΟΜΠΟΤΙΚΗ, Μηχανική και Αυτόματος έλεγχος, Εκδόσεις Τζιόλα, 2008

ΔΟΥΛΓΕΡΗ Ζωή, ΡΟΜΠΟΤΙΚΗ, κινηματική δυναμική και έλεγχος αρθρωτών βραχιόνων, Εκδόσεις Κριτική, 2007

Ρομποτική- Αυτόματη Θεωρία-Πράξη γύρω από τα ρομπότ, Εκδόσεις Σύγχρονη Έκδοση, 2006

Δρ. Βολογιαννίδης Σταύρος, Συστήματα Αυτομάτου Ελέγχου, Θεωρία και Εφαρμογές, Διδακτικές σημειώσεις, Τμήμα Πληροφορικής & Επικοινωνιών, ΤΕΙ Σερρών, 2009

Προγραμματισμένοι Ελεγκτές, Demis Collins Eammon Lane, Επιμέλεια μετάφρασης: Δρ. Νικόλαος Κ. Κωνσταντινίδης, Εκδόσεις Α. Τζιόλα, Ε Θεσσαλονίκη.

Πλήρες εγχειρίδιο της java 6, Ragers Cadenhead, Laura Lemay, Απόδοση : Ελένη Γκαγκάτσιου, Εκδόσεις Μ. Γκιούρδας, Αθήνα 2007

Ιστοσελίδες:

- 1) <http://www.wikipedia.org/>
- 2) <http://www.arduino.cc/>
- 3) <http://luckylarry.co.uk/>
- 4) <http://www.mathworks.com/>