

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΟΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ 1568

ΣΥΝΑΡΜΟΛΟΓΗΣΗ ΚΑΙ ΑΣΥΡΜΑΤΟΣ ΕΛΕΓΧΟΣ ΕΝΟΣ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ ΜΕ ΕΞΙ ΒΑΘΜΟΥΣ ΕΛΕΥΘΕΡΙΑΣ ΜΕΣΩ ΜΙΚΡΟΕΛΕΓΚΤΩΝ ARDUINO

ΦΟΙΤΗΤΗΣ:

ΚΟΥΡΤΙΝΟΣ ΠΑΝΑΓΙΩΤΗΣ (6460)

ΕΙΣΗΓΗΤΗΣ:

ΒΛΑΧΟΠΟΥΛΟΣ ΠΕΤΡΟΣ

ΠΑΤΡΑ 2017

Πρόλογος/ Περίληψη

Η παρούσα εργασία έχει σκοπό τον ασύρματο έλεγχο μέσω μικροελεγκτών Arduino ενός ρομποτικού βραχίονα με έξι βαθμούς ελευθερίας. Ειδικότερα, η εργασία περιλαμβάνει τη συναρμολόγηση του ρομποτικού βραχίονα και τον προγραμματισμό των μικροελεγκτών Arduino για τον ασύρματο έλεγχο μέσω χειριστηρίων των σερβοκινητήρων του συστήματος.

Στο πρώτο κεφάλαιο περιγράφονται αναλυτικά οι μικροελεγκτές Arduino που θα χρησιμοποιηθούν. Επίσης, γίνεται αναφορά στην αρχή λειτουργίας τους και στα κριτήρια της επιλογής τους και περιγράφονται τα βοηθητικά κυκλώματα (shields) που κάνουν δυνατό τον ασύρματο έλεγχο του ρομποτικού βραχίονα.

Στο δεύτερο κεφάλαιο παρατίθεται μια γενική προσέγγιση του τομέα της ρομποτικής, ενώ δίνεται ιδιαίτερη έμφαση στην ανάλυση των ρομποτικών βραχιόνων ως μέρος αυτής. Συγκεκριμένα, παρουσιάζονται η αρχή λειτουργίας τους και τα είδη τους, και γίνεται μια ιστορική αναδρομή της χρήσης τους στη βιομηχανία.

Τα δύο επόμενα κεφάλαια πραγματεύονται τις απαραίτητες πειραματικές διαδικασίες. Οι εν λόγω διαδικασίες αφορούν στη συναρμολόγηση του ρομποτικού βραχίονα, καθώς και στον υπολογισμό των απαραίτητων ηλεκτρικών μεγεθών με σκοπό την υλοποίησή του και τον ασύρματο έλεγχό του. Ακολουθεί ο προγραμματισμός των μικροελεγκτών Arduino, κατά τον οποίο έλαβε χώρα η επέκταση του ενός μικροελεγκτή με το απαραίτητο υλικό για τον έλεγχο και των 7 σερβοκινητήρων του συστήματος. Επίσης, μέσω προγραμματισμού των μικροελεγκτών δόθηκε η δυνατότητα ασύρματου χειρισμού του ρομποτικού βραχίονα, μέσω χειριστηρίων (thumbsticks). Για την πραγματοποίηση των ανωτέρω, χρησιμοποιήθηκαν οι κατάλληλες βιβλιοθήκες λογισμικού, καθώς και διαφορετικά shields.

Στο πέμπτο κεφάλαιο παρουσιάζονται τα συμπεράσματα της παρούσας πτυχιακής εργασίας.

Πίνακας περιεχομένων

Πρόλογος/ Περίληψη	iii
1 Εισαγωγή.....	1
1.1 Τι είναι το Arduino	1
1.2 Το υλικό της πλατφόρμας Arduino	1
1.2.1 Ο μικροελεγκτής ATmega2560	2
1.2.2 Τεχνικά χαρακτηριστικά Arduino	3
1.2.3 Τροφοδοσία.....	5
1.3 Περιβάλλον ανάπτυξης.....	5
1.3.1 Το πρόγραμμα Arduino IDE	5
1.3.2 Η γλώσσα Wiring.....	5
2 Οι ρομποτικοί βραχίονες	9
2.1 Ρομποτική.....	9
2.1.1 Ιστορική αναδρομή.....	9
2.1.2 Χαρακτηριστικά και δομή των ρομπότ	9
2.2 Ρομποτικοί βραχίονες.....	10
2.2.1 Βαθμοί ελευθερίας.....	10
2.2.2 Χαρακτηριστικά και δομή ρομποτικών βραχιόνων.....	11
2.2.3 Τύποι ρομποτικών βραχιόνων (βάσει γεωμετρικής διαμόρφωσης)	11
3 Υλικά και μέθοδοι	15
3.1 Μηχανικό σχέδιο ρομποτικού βραχίονα.....	15
3.2 Επιλογή σερβοκινητήρων	16
3.3 Χαρακτηριστικά σερβοκινητήρων	17
4 Κατασκευή βραχίονα και προγραμματισμός μικροελεγκτή	21
4.1 Συναρμολόγηση βραχίονα	21
4.1.1 Διαδικασία εγκατάστασης σερβοκινητήρων	21
4.1.2 Εγκατάσταση Βοηθητικών κυκλωμάτων (shields) στο μικροελεγκτή και σύνδεση σερβοκινητήρων.....	27
4.2 Συναρμολόγηση χειριστηρίου.....	28
4.2.1 Διαδικασία εγκατάστασης Arduino και συνδεσμολογία χειριστηρίου.....	29
4.3 Πειραματική διαδικασία.....	34
4.3.1 Σύνδεση Arduino με Η/Υ	34
4.3.2 Βιβλιοθήκες	35
4.3.3 Πείραμα ορθής λειτουργίας σερβοκινητήρων και calibration.....	36
4.3.4 Πείραμα επικοινωνίας μεταξύ Arduino Mega και Arduino Uno μέσω NRF24L01+	37

4.3.5	Ασύρματος έλεγχος ρομποτικού βραχίονα.....	40
5	Συμπεράσματα.....	53
	Βιβλιογραφία	55

1 Εισαγωγή

1.1 Τι είναι το Arduino

Το Arduino αποτελεί μια πλατφόρμα ανάπτυξης ανοιχτού κώδικα, το οποίο είναι αποτέλεσμα έργου του Wiring project. Το Wiring project είχε ως σκοπό την ανάπτυξη ενός απλού συστήματος, το οποίο βασίζεται σε μικροελεγκτή που προσφέρεται για εύκολη και γρήγορη ανάπτυξη εφαρμογών.

Πρόκειται για μία απλή μητρική πλακέτα ανοιχτού κώδικα με ενσωματωμένο ένα μικροελεγκτή, καθώς και εισόδους/εξόδους, όπου μέσω της γλώσσας Wiring δίνει τη δυνατότητα υλοποίησης από απλών μέχρι πολυσύνθετων projects.

Η δημοτικότητα της πλατφόρμας Arduino σχετίζεται με τα 4 βασικότερα πλεονεκτήματά της:

- ❑ **Κόστος:** Μπορεί κάποιος να αγοράσει την πιο απλή μορφή του Arduino (Uno) με κόστος χαμηλότερο των 12 ευρώ και το περιβάλλον ανάπτυξης παρέχεται δωρεάν.
- ❑ **Συμβατότητα:** Το λογισμικό του Arduino είναι συμβατό με τα δημοφιλέστερα λειτουργικά συστήματα (Windows, Macintosh, Linux).
- ❑ **Προσβασιμότητα:** Το περιβάλλον ανάπτυξης είναι αρκετά απλό ώστε να χρησιμοποιείται εύκολα και από αρχάριους χρήστες.
- ❑ **Ανοιχτός κώδικας:** Τα προγράμματα του Arduino είναι ανοιχτού κώδικα, συνεπώς ο καθένας μπορεί να βρει τμήματα πηγαίου κώδικα, κυρίως, στο διαδίκτυο. Έπειτα, μπορεί να τα τροποποιήσει ανάλογα με τις ανάγκες της εφαρμογής του. Επίσης, μέσα από την προσθήκη βιβλιοθηκών C++ μπορούν να προστεθούν δυνατότητες στο περιβάλλον ανάπτυξης (Παπάζογλου & Λιωνής, 2015).

1.2 Το υλικό της πλατφόρμας Arduino

Το Arduino Mega 2560 είναι μια πλακέτα μικροελεγκτή βασισμένη στο ATmega2560 της Atmel. Αποτελεί τη ναυαρχίδα των πλατφορμών Arduino δίνοντας 54 ψηφιακές εισόδους/εξόδους, από τις οποίες εξόδους οι 14 μπορούν να χρησιμοποιηθούν ως εξοδοί PWM, και 14 αναλογικές εισόδους (Microelectronics, 2016). Σε αντίθεση με τις υπόλοιπες πλατφόρμες της Arduino, το Mega διαθέτει μεγαλύτερο πλήθος ακροδεκτών. Το πλεονέκτημα αυτό προσφέρει τη δυνατότητα σύνδεσης με πολλές εξωτερικές συσκευές και διατάξεις. Εν προκειμένω, προσφέρει επιπροσθέτως τον έλεγχο πολλών σερβοκινητήρων (Παπάζογλου & Λιωνής, 2015).

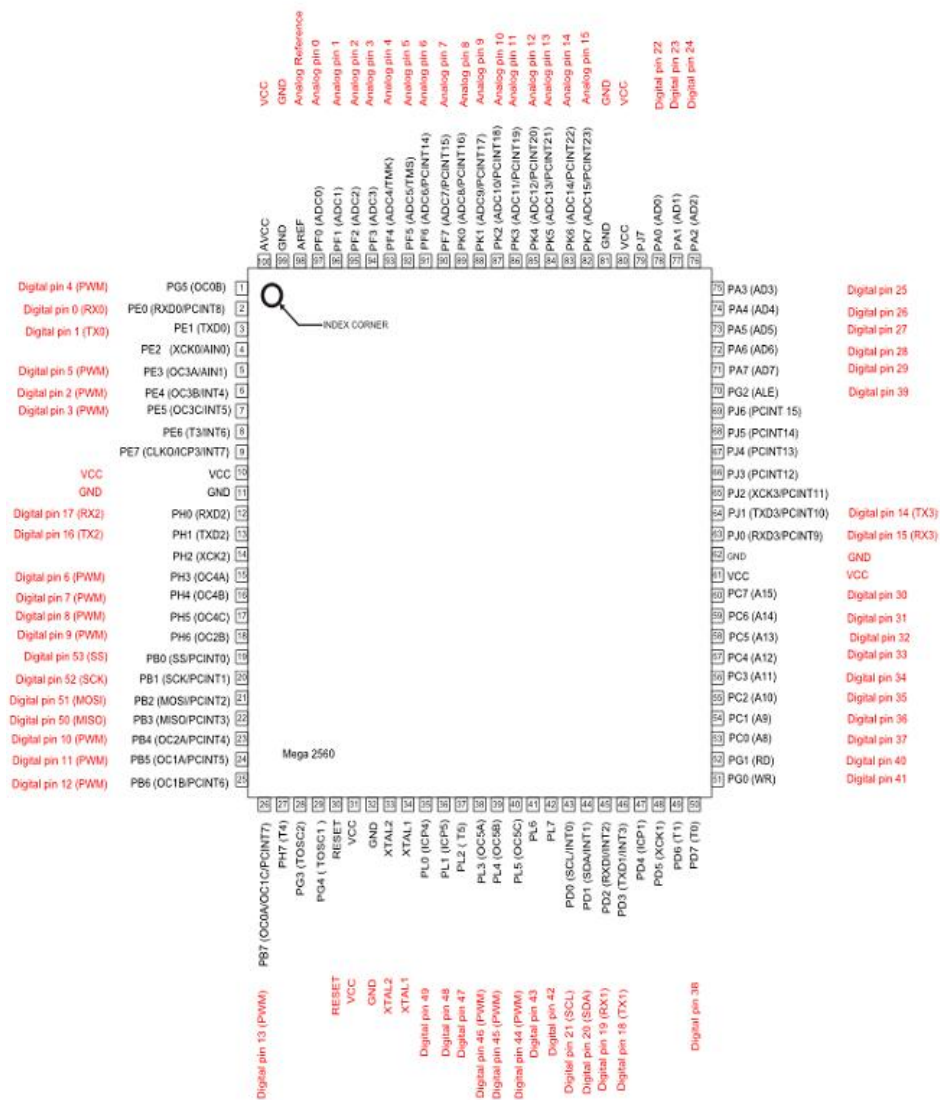


Εικόνα 1: Arduino Sainsmart Mega (Sainsmart, 2016a).

1.2.1 Ο μικροελεγκτής ATmega2560

Ο ATmega2560 είναι ένας χαμηλής κατανάλωσης Complementary Metal-Oxide-Semiconductor (CMOS) 8-bit μικροελεγκτής βασισμένος στην αρχιτεκτονική Reduced Instruction Set Computer (RISC) της AVR. Ο εν λόγω μικροελεγκτής έχει τη δυνατότητα εκτέλεσης εντολών σε ένα κύκλο ρολογιού, επιτυγχάνοντας μεταφορά δεδομένων που πλησιάζει το 1 Million Instructions Per Second (MIPS) ανά MHz.

Ο επεξεργαστικός πυρήνας Atmel AVR συνδυάζει ένα πλούσιο σετ εντολών με 32 μητρώα γενικής χρήσης. Τα εν λόγω μητρώα συνδέονται άμεσα με την αριθμητική λογική μονάδα (ALU), επιτρέποντας δύο ανεξάρτητα μητρώα να είναι προσβάσιμα με μία εντολή που εκτελείται σε ένα κύκλο ρολογιού. Η συγκεκριμένη δομή κάνει τον κώδικα που εκτελείται πιο αποτελεσματικό, ενώ επιτυγχάνει 10 φορές πιο γρήγορη εκτέλεση από τους συμβατικούς Complex Instructions Set Computer (CISC) μικροελεγκτές (Atmel, 2014).



Εικόνα 2: Pinmap 2560 (Arduino, 2017a).

1.2.2 Τεχνικά χαρακτηριστικά Arduino

Όλα τα τεχνικά χαρακτηριστικά τόσο του Arduino Mega όσο και του Uno που θα χρησιμοποιηθούν για τον απομακρυσμένο έλεγχο παρατίθενται στους παρακάτω πίνακες (Πίνακες 1 και 2):

Πίνακας 1: Τεχνικά χαρακτηριστικά Arduino Mega (Arduino, 2017b).

Μικροελεγκτής	ATmega2560
Τάση λειτουργίας	5V
Τάση εισόδου (Προτεινόμενη)	7-12V

Μέγιστη τάση εισόδου (Όριο)	6-20V
Ψηφιακά I/O Pins	54 (από αυτές 15 έχουν δυνατότητα PWM εξόδου)
Αναλογικά Pins εισόδου	16
DC Ρεύμα I/O Pin	20 mA
DC Ρεύμα για 3.3V Pin	50 mA
Μνήμη Flash	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Ταχύτητα Ρολογιού	16 MHz
LED_BUILTIN	13

Πίνακας 2: Τεχνικά χαρακτηριστικά Arduino Uno (Arduino, 2017c).

Μικροελεγκτής	ATmega328P
Τάση λειτουργίας	5V
Τάση εισόδου (Προτεινόμενη)	7-12V
Μέγιστη τάση εισόδου (Όριο)	6-20V
Ψηφιακά I/O Pins	14 (από αυτές 6 έχουν δυνατότητα PWM εξόδου)
PWM Digital I/O Pins	6
Αναλογικά Pins εισόδου	6
DC ρεύμα I/O Pin	20 mA
DC ρεύμα για 3.3V Pin	50 mA
Μνήμη Flash	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Ταχύτητα ρολογιού	16 MHz
LED_BUILTIN	13

1.2.3 Τροφοδοσία

Η τροφοδοσία του Arduino Mega2560 μπορεί να γίνει είτε μέσω θύρας USB του Η/Υ, είτε εξωτερικά. Η διάκριση μεταξύ USB και εξωτερικής τροφοδοσίας γίνεται αυτόματα από το μικροελεγκτή. Η εξωτερική τροφοδοσία μπορεί να γίνει από μετασχηματιστή AC σε DC ή μπαταρία.

Η μικροελεγκτής μπορεί να λειτουργήσει με εξωτερική τροφοδοσία από 6 έως 20 Volt. Εάν όμως η τροφοδοσία είναι μικρότερη των 7 Volt, υπάρχει πιθανότητα η τελική τάση που εισέρχεται στην πλακέτα να είναι μικρότερη των 5 Volt με αποτέλεσμα να γίνει ασταθής. Τάση μεγαλύτερη των 12 Volt δύναται να προκαλέσει υπερθέρμανση του ρυθμιστή τάσης και καταστροφή της πλακέτας.

Οι ακίδες ισχύος του μικροελεγκτή έχουν ως εξής:

Vin: Η είσοδος αυτή χρησιμοποιείται μόνο στην περίπτωση εξωτερικής πηγής τροφοδοσίας και παρέχει ρυθμιστή τάσεως, ώστε να προσφέρει τα 5 Volt που χρειάζεται από πηγές DC ή μπαταρίες του εύρους 7 έως 12 Volt. Το μέγιστο ρεύμα που τραβάει ο μικροελεγκτής είναι 50 mA.

GND: Ακίδα γείωσης.

IOREF: παρέχει την τάση αναφοράς με την οποία λειτουργεί ο μικροελεγκτής. Με σωστή ρύθμιση των Shields που εγκαθίστανται στο Arduino μέσω του IOREF επιλέγεται η κατάλληλη τροφοδοσία ή ενεργοποιούνται οι Voltage Translators στις εξόδους για λειτουργία σε 5 ή 3.3 Volt (Arduino, 2017b).

1.3 Περιβάλλον ανάπτυξης

1.3.1 Το πρόγραμμα Arduino IDE

Το πρόγραμμα Arduino IDE (Integrated Development Environment) αποτελεί το δωρεάν διαθέσιμο λογισμικό προγραμματισμού των μικροελεγκτών της οικογένειας Arduino. Το Arduino IDE είναι γραμμένο στη γλώσσα Java και αποτελεί cross-platform λογισμικό, δηλαδή μπορεί να εκτελεστεί σε πληθώρα λειτουργικών συστημάτων.

Το εν λόγω λογισμικό περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως έλεγχο συντακτικού και αυτόματη μορφοποίηση. Παράλληλα, παρέχει απλό μηχανισμό load and compile του κώδικα, όπου μετά τον έλεγχο γίνεται load στο μικροελεγκτή.

Μέσω του προγράμματος Arduino IDE, δίνεται η δυνατότητα προγραμματισμού του μικροελεγκτή με τη γλώσσα Wiring. Μέσω της σύνδεσης του μικροελεγκτή με τον Η/Υ (μέσω θύρας USB), εφόσον γίνει compile από το πρόγραμμα για έλεγχο συντακτικών και προγραμματιστικών σφαλμάτων, γίνεται η διαδικασία μετάφρασης και εκτέλεσης του κώδικα, ο οποίος παραμένει αποθηκευμένος στην μνήμη του μικροελεγκτή (Monk, 2012).

1.3.2 Η γλώσσα Wiring

Η γλώσσα Wiring αποτελεί μία γλώσσα προγραμματισμού βασισμένη στην Processing IDE, η οποία είναι γλώσσα προγραμματισμού ανοιχτού κώδικα κατασκευασμένη με κύριο σκοπό την εκμάθηση των βασικών αρχών προγραμματισμού σε αρχάριους. Η εξέλιξη της Processing καθώς και η χρήση βιβλιοθηκών C και C++ οδήγησαν στη δημιουργία της Wiring. Σκοπός της είναι η εκμάθηση προγραμματισμού μικροελεγκτών και η δημιουργία

ανεξάρτητων Project, καθώς η γλώσσα, το λογισμικό και το hardware αποτελούν προϊόντα open source.

Ένα απλό πρόγραμμα γλώσσας Wiring για χρήση σε Arduino έχει ως προϋπόθεση τον ορισμό από τον χρήστη των εξής 2 λειτουργιών:

-setup(): μία συνάρτηση που τρέχει μία φορά στην αρχή του προγράμματος, η οποία αρχικοποιεί τις ρυθμίσεις,

-loop(): μία συνάρτηση που καλείται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί (Εικόνα 3).



Εικόνα 3: Παράθυρο Arduino IDE (Ιδία σχεδίαση, 2016).

Ένα τυπικό πρώτο παράδειγμα κώδικα είναι για τον μικροελεγκτή να ανάβει και να σβήνει ένας λαμπτήρας LED (Εικόνα 4). Στο περιβάλλον Arduino IDE ο χρήστης μπορεί να γράψει ένα πρόγραμμα όπως το παρακάτω (Arduino, 2017d):

```
#define LED_PIN 13  
  
void setup () {  
  pinMode (LED_PIN, OUTPUT); // Ορισμός pin 13 ως έξοδος LED  
}  
  
void loop () {  
  digitalWrite (LED_PIN, HIGH); // High στο LED ( Ανάβει)  
  delay (1000); // καθυστέρηση 1 δευτερολέπτου (1000 ms)  
  digitalWrite (LED_PIN, LOW); // Low στο LED (Σβήνει)
```

```
delay (1000); // Καθυστέρηση 1 δευτερολέπτου  
}
```



The screenshot shows the Arduino IDE interface. The main window displays the following code:

```
/*  
 * Blink  
 * Turns on an LED on for one second, then off for one second, repeatedly.  
 *  
 * Most Arduinos have an on-board LED you can control. On the Uno and  
 * Leonardo, it is attached to digital pin 13. If you're unsure what  
 * pin the on-board LED is connected to on your Arduino model, check  
 * the documentation at https://www.arduino.cc  
 *  
 * This example code is in the public domain.  
 *  
 * modified 8 May 2014  
 * by Mark Spitznagel  
 */  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin 13 as an output.  
  pinMode(13, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);           // wait for a second  
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW  
  delay(1000);           // wait for a second  
}
```

At the bottom of the IDE, the memory usage is displayed:

```
Sketch uses 1,344 bytes (8%) of program storage space. Maximum is 32,768 bytes.  
Global variables use 0 bytes (0%) of dynamic memory, leaving 2,048 bytes for local variables. Maximum is 2,048 bytes.
```

Εικόνα 4: Πρόγραμμα Blink Led (Ιδία σχεδίαση, 2016).

2 Οι ρομποτικοί βραχίονες

2.1 Ρομποτική

Η ρομποτική ως επιστήμη αποτελεί κλάδο της μηχανολογίας, της ηλεκτρολογίας και της επιστήμης των υπολογιστών. Ασχολείται με το σχεδιασμό, την κατασκευή, τη λειτουργία και την εφαρμογή αυτόνομων μηχανών, καθώς και τα συστήματα πληροφορικής για τον έλεγχό τους και την επεξεργασία των πληροφοριών (Oxford University Press, 2017).

Οι εν λόγω τεχνολογίες ασχολούνται με ρομπότ που μπορούν να αντικαταστήσουν τον άνθρωπο σε επικίνδυνα περιβάλλοντα ή διαδικασίες κατασκευής. Επίσης, μπορεί να μοιάζουν με τους ανθρώπους στην εμφάνιση, τη συμπεριφορά, ή και τη γνώση. Πολλά από τα σημερινά ρομπότ είναι εμπνευσμένα από τη φύση, αναπτύσσοντας τον τομέα της βιοεμπνευσμένης ρομποτικής (Μπαρμπουλέτου, 2013).

2.1.1 Ιστορική αναδρομή

Η λέξη ρομποτική προέρχεται από τη λέξη ρομπότ. Η πρώτη αναφορά της λέξης γίνεται το 1920 από τον Τσέχο συγγραφέα Karel Čapek στο έργο του R.U.R (Rossum's Universal Robots) και προέρχεται από τη σλάβικη λέξη *robota* που σημαίνει εργασία. Στο συγκεκριμένο έργο η λέξη ρομπότ αναφερόταν σε πλάσματα που μπορούσαν να εκληφθούν ως άνθρωποι, ιδέα παρόμοια με αυτή των σύγχρονων ανδροειδών (Encyclopædia Britannica, 2017).

Η ιδέα της δημιουργίας ρομπότ χρονολογείται στην κλασσική εποχή, αλλά η έρευνα και η ανάπτυξη τέτοιων μηχανών δεν άλλαξε σημαντικά μέχρι τα τέλη του 20^{ου} αιώνα. Σήμερα, η ρομποτική είναι ένα ταχέως αναπτυσσόμενο τεχνολογικό πεδίο με πληθώρα εφαρμογών. Αυτές εκτείνονται από την βιομηχανία και την καθημερινή ανθρώπινη χρήση, έως και για εξαιρετικά επικίνδυνες για ανθρώπους εργασίες. Παραδείγματα αποτελούν ο χειρισμός χημικών, η στρατιωτική εφαρμογή, η έρευνα σε επικίνδυνα και μη βιώσιμα περιβάλλοντα (ορυχεία, ναυάγια), κ.ά. (Nocks, 2008).

2.1.2 Χαρακτηριστικά και δομή των ρομπότ

Σύμφωνα με το Robot Institute of America, ο όρος ρομπότ αναφέρεται σε ένα μηχανισμό που έχει σχεδιαστεί ώστε μέσω προγραμματιζόμενων κινήσεων να μεταφέρει υλικά, αντικείμενα, εργαλεία ή εξειδικευμένες συσκευές με σκοπό την εκτέλεση ενός συνόλου εργασιών. Το γεγονός ότι η ρομποτική αποτελεί τον συνδυασμό τριών διαφορετικών επιστημονικών κλάδων κρίνει απαραίτητο, ανεξαρτήτως του μεγέθους και της εργασίας του, να αποτελείται από διαφορετικά υποσυστήματα:

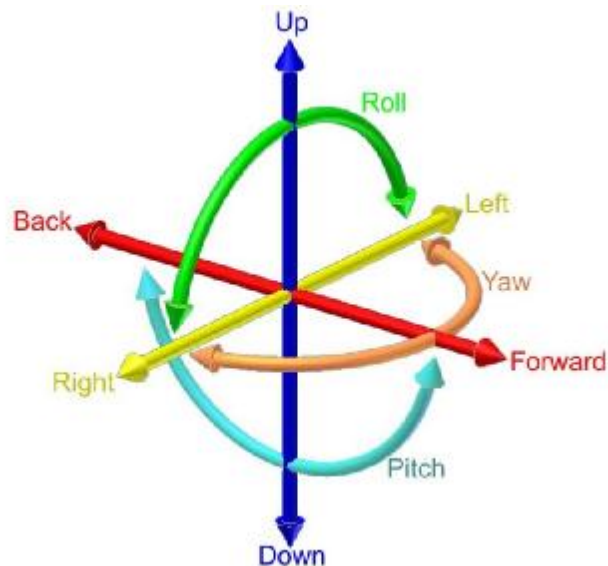
- ☐ Το μηχανολογικό υποσύστημα που είναι υπεύθυνο για την εκτέλεση του έργου, το οποίο αποτελείται από μηχανισμούς που επιτρέπουν στο ρομπότ να κινείται (π.χ. αρθρώσεις, σύστημα μετάδοσης κίνησης, ενεργοποιητές-κινητήρες, οδηγοί, κ.λπ.),
- ☐ Το υποσύστημα αίσθησης που είναι υπεύθυνο για τη συγκέντρωση πληροφοριών, οι οποίες σχετίζονται τόσο με την κατάσταση του ρομπότ όσο και με την κατάσταση του περιβάλλοντος. Στο εν λόγω υποσύστημα περιλαμβάνονται όργανα μετρήσεων, αισθητήρες, ηλεκτρονικά στοιχεία, κ.ά.
- ☐ Το υποσύστημα ελέγχου συνδυάζει κατάλληλα την αίσθηση με τη δράση, έτσι ώστε το ρομπότ να λειτουργεί αποτελεσματικά και με τον επιθυμητό τρόπο. Ο ελεγκτής που διαθέτει το συγκεκριμένο υποσύστημα, αναλαμβάνει την επίβλεψη και το

συντονισμό ολόκληρου του συστήματος που αποτελεί το ρομπότ (Κυριακόπουλος και Παπαδόπουλος, χ.χρ).

2.2 Ρομποτικοί βραχίονες

2.2.1 Βαθμοί ελευθερίας

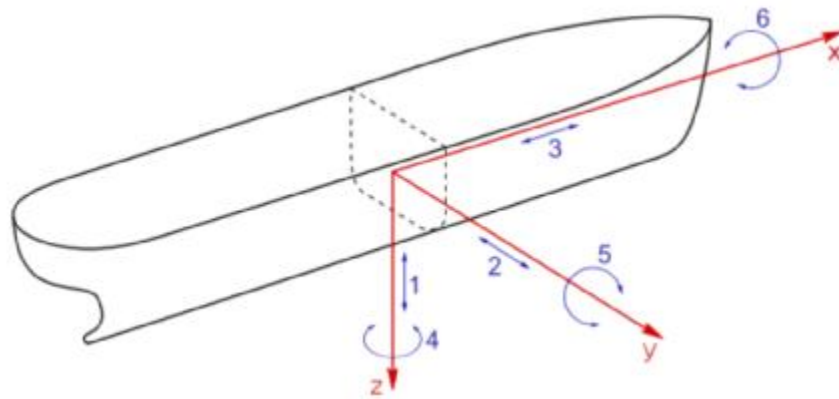
Σε πολλά επιστημονικά πεδία, οι βαθμοί ελευθερίας ενός συστήματος είναι ο αριθμός των παραμέτρων του συστήματος. Στη μηχανική ως βαθμοί ελευθερίας ορίζεται το πλήθος των ανεξάρτητων συντεταγμένων που χρειάζονται για να περιγραφεί η κίνηση του συστήματος. Για παράδειγμα, η θέση ενός τραίνου που κινείται κατά μήκος μιας τροχιάς καθορίζεται από την απόστασή του κατά μήκος της τροχιάς. Όμως, για τον ορισμό της θέσης ενός αντικειμένου στο χώρο χρειάζονται 3 συντεταγμένες (x,y,z), διότι υπάρχουν τρεις διαστάσεις. Παράλληλα, η περιστροφική κίνηση στο χώρο περιγράφεται από 3 γωνίες, οι οποίες έχουν σημείο αναφοράς αντίστοιχα τις συντεταγμένες (x,y,z).



Εικόνα 5: Οι 6 βαθμοί ελευθερίας (Wikimedia, 2017).

Ένα απλό παράδειγμα που βοηθά στην κατανόηση των 6 βαθμών ελευθερίας είναι η κίνηση ενός άκαμπτου σώματος, όπως ένα πλοίο στη θάλασσα. Η κίνηση του αποτελείται από:

1. Κίνηση πάνω-κάτω (ανύψωση)
2. Κίνηση αριστερά-δεξιά(ταλάντωση)
3. Κίνηση εμπρός-πίσω
4. Περιστροφή αριστερά δεξιά (εκτροπή, yawing)
5. Κλίση εμπρός-πίσω (pitching)
6. Στροφή από τη μία πλευρά στην άλλη (κύλιση, rolling) (Summary of ship movement, 2016).



Εικόνα 6: Κίνηση πλοίου (Wikimedia, 2015).

Η αναφορά των βαθμών ελευθερίας στη ρομποτική, ουσιαστικά, εκφράζει τη δυνατότητα κίνησης και περιστροφής του ρομπότ στο χώρο. Δηλαδή, ένας ρομποτικός βραχίονας 6 βαθμών ελευθερίας έχει τη δυνατότητα κίνησης και περιστροφής στις 3 διαστάσεις αντίστοιχα (Pennestri et al. 2005).

2.2.2 Χαρακτηριστικά και δομή ρομποτικών βραχιόνων

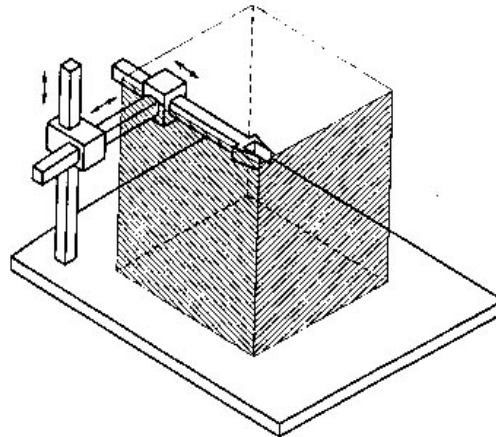
Ο σκοπός ενός ρομποτικού βραχίονα είναι να μιμηθεί το ανθρώπινο χέρι. Έτσι, οι αρθρώσεις του είναι ανάλογες αυτών του ανθρώπινου χεριού («ώμος», «αγκώνας» και «καρπός»). Ο καρπός μπορεί να περιστρέφεται γύρω από τρεις άξονες. Οι εν λόγω άξονες αφορούν στους «διαμήκη», «εγκάρσιο» και «κάθετο». Ο «διαμήκης» άξονας έχει τη διεύθυνση των δακτύλων (η κίνηση ονομάζεται κύλιση – roll). Ο «εγκάρσιος» βρίσκεται στο επίπεδο της παλάμης (κίνηση ανύψωσης – pitch) και είναι κάθετος ως προς τα δάκτυλα. Ενώ, ο «κάθετος» είναι κάθετος ως προς τους άλλους δύο (κίνηση στροφής – yaw), δηλαδή προς το επίπεδο της παλάμης.

Οι αρθρώσεις οδηγούνται από κινητήρες ή γενικότερα από στοιχεία δράσης, που μπορεί να είναι πνευματικά, υδραυλικά ή ηλεκτρικά. Τα κινητήρια στοιχεία δράσης (κινητήρες, έμβολα, κ.α.) δίνουν στα ρομπότ πολύ μεγαλύτερη ισχύ από αυτήν που μπορεί να αναπτύξει ο άνθρωπος (Craig, 2006)

2.2.3 Τύποι ρομποτικών βραχιόνων (βάσει γεωμετρικής διαμόρφωσης)

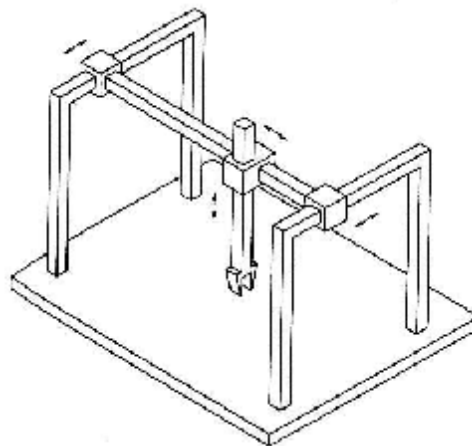
Ανάλογα με τον τύπο και την διαδοχή των αρθρώσεων ενός βραχίονα, καθώς και με τη γεωμετρία τους και το χώρο εργασίας τους, γίνεται η ταξινόμηση των ρομποτικών βραχιόνων σε διάφορες κατηγορίες οι οποίες αναφέρονται παρακάτω:

- ☐ Καρτεσιανοί Βραχίονες: η καρτεσιανή γεωμετρία υλοποιείται με τρεις διαδοχικές πρισματικές αρθρώσεις. Οι άξονες των αρθρώσεων αυτών είναι ανά δύο κάθετοι μεταξύ τους (Εικόνα 7). Η καρτεσιανή δομή παρέχει μεγάλη δυσκαμψία και σταθερή ακρίβεια σε ολόκληρο το χώρο εργασίας, οποίος είναι ένα παράλληλο επίπεδο. Βασικό μειονέκτημα της κατασκευής είναι η μειωμένη επιδεξιότητα κίνησης, λόγω της πρισματικής φύσης των αρθρώσεων.



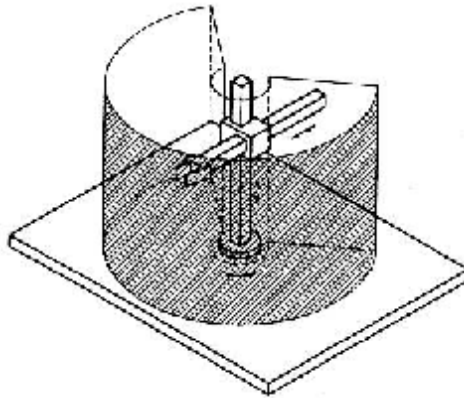
Εικόνα 7: Καρτεσιανός βραχίονας (Κυριακόπουλος & Παπαδόπουλος, χ.χρ).

- 8 Βραχίονες Gantry: παρόλο που είναι ουσιαστικά καρτεσιανοί, έχουν μία ειδοποιό διαφορά στον τρόπο προσέγγισης του αντικειμένου ενδιαφέροντος (Εικόνα 8). Ειδικότερα, οι εν λόγω βραχίονες προσεγγίζουν το αντικείμενο από πάνω, τη στιγμή που οι κλασικοί καρτεσιανοί προσεγγίζουν το αντικείμενο από το πλάι. Άμεσες συνέπειες της εν λόγω διαφοροποίησης είναι η αύξηση του χώρου εργασίας και της δυσκαμψίας, καθώς επίσης και η δυνατότητα χειρισμού μεγάλων και βαρέων αντικειμένων.



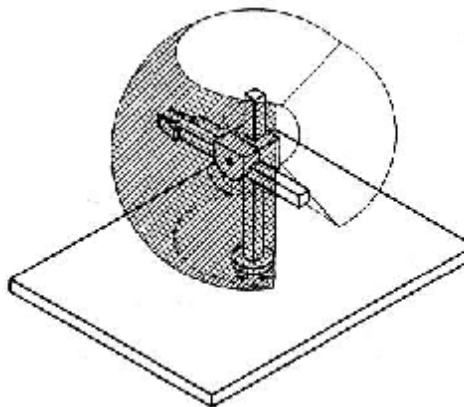
Εικόνα 8: Βραχίονας Gantry (Κυριακόπουλος & Παπαδόπουλος, χ.χρ).

- 9 Κυλινδρικοί Βραχίονες: η πρώτη πρισματική άρθρωση της καρτεσιανής δομής έχει αντικατασταθεί από μία περιστροφική άρθρωση (Εικόνα 9). Οι συγκεκριμένοι βραχίονες χαρακτηρίζονται από καλή δυσκαμψία. Όμως, η ακρίβεια της θέσης του καρπού μειώνεται καθώς η οριζόντια μετατόπιση αυξάνεται. Ο χώρος εργασίας στην εν λόγω περίπτωση είναι ένα τμήμα κυλίνδρου. Σημαντικό μειονέκτημα της συγκεκριμένης γεωμετρίας είναι το ότι ο βραχίονας εισέρχεται στο χώρο εργασίας και τον περιορίζει.



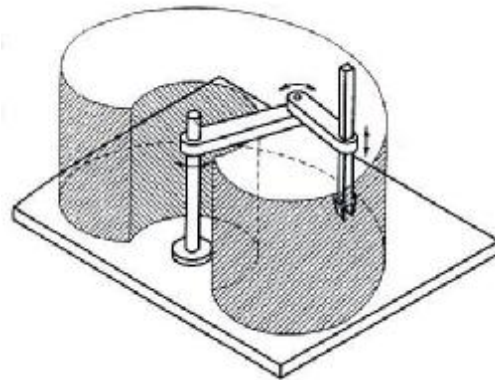
Εικόνα 9: Κυλινδρικός βραχίονας (Κυριακόπουλος & Παπαδόπουλος, χ.χρ).

- 14 Σφαιρικοί Βραχίονες: στους εν λόγω βραχίονες αντικαθίσταται πλέον και η δεύτερη πρισματική άρθρωση της καρτεσιανής δομής με περιστροφική (Εικόνα 10). Η μηχανολογική πολυπλοκότητα αυξάνεται, ενώ η δυσκαμψία μειώνεται. Επιπλέον, η ακρίβεια του καρπού μειώνεται με την αύξηση της ακτινικής απόστασης. Ο χώρος εργασίας είναι ένα τμήμα σφαίρας και περιέχει ένα μέρος της βάσης, με άμεση συνέπεια τη δυνατότητα χειρισμού αντικειμένων που βρίσκονται στο έδαφος.




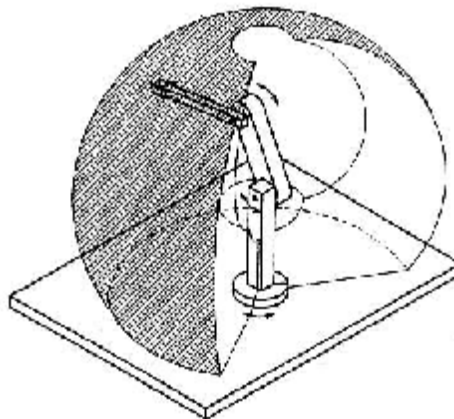
Εικόνα 10: Σφαιρικός βραχίονας (Κυριακόπουλος & Παπαδόπουλος, χ.χρ).

- 15 Βραχίονες SCARA: η γεωμετρία SCARA είναι ειδική σε σχέση με τους προαναφερθέντες βραχίονες και περιλαμβάνει δύο περιστροφικές και μία πρισματική άρθρωση. Οι εν λόγω αρθρώσεις είναι τοποθετημένες κατά τέτοιο τρόπο ώστε οι άξονες κίνησης να είναι παράλληλοι μεταξύ τους (Εικόνα 11). Το όνομα SCARA προέρχεται από τα αρχικά των λέξεων Selective Compliance Assembly Robot Arm. Η συγκεκριμένη γεωμετρία παρέχει μεγάλη δυσκαμψία σε κατακόρυφη φόρτιση και ελαστικότητα σε οριζόντια. Η ακρίβεια τοποθέτησης του καρπού μειώνεται με την αύξηση της απόστασής του από τον άξονα της πρώτης άρθρωσης.



Εικόνα 11: Βραχίονας Scara (Κυριακόπουλος & Παπαδόπουλος, χ.χρ).

-  Ανθρωπομορφικοί Βραχίονες: η ανθρωπομορφική γεωμετρία υλοποιείται με τρεις διαδοχικές περιστροφικές αρθρώσεις. Ειδικότερα, ο άξονας περιστροφής της πρώτης άρθρωσης είναι κατακόρυφος και κάθετος στους άξονες περιστροφής των επομένων δύο αρθρώσεων, οι οποίοι είναι παράλληλοι μεταξύ τους (Εικόνα 12). Η συγκεκριμένη δομή παρέχει τη μεγαλύτερη επιδεξιότητα από όλες τις προηγούμενες, καθώς όλες οι αρθρώσεις είναι περιστροφικές. Ωστόσο, η ακρίβεια του καρπού δεν είναι σταθερή εντός του χώρου εργασίας, ο οποίος έχει τη μορφή σφαίρας. (Κυριακόπουλος & Παπαδόπουλος, χ.χρ)



Εικόνα 12: Ανθρωπομορφικός βραχίονας (Κυριακόπουλος & Παπαδόπουλος, χ.χρ).

3 Υλικά και μέθοδοι

3.1 Μηχανικό σχέδιο ρομποτικού βραχίονα

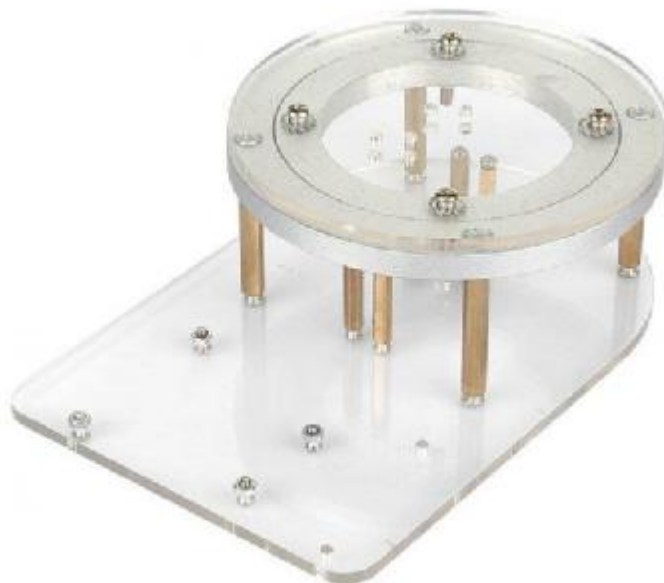
Η κατασκευή του ρομποτικού βραχίονα με 6 βαθμούς ελευθερίας, που αντιστοιχούν σε αρθρώσεις, αποτελείται από συνδέσμους οι οποίοι είναι σχεδιασμένοι για την τοποθέτηση σερβοκινητήρων κατασκευασμένους από αλουμίνιο (Εικόνα 13).



Εικόνα 13: Σύνδεσμοι αλουμινίου βραχίονα (Sainsmart, 2017b).

Οι παράγοντες που λήφθηκαν σοβαρά υπόψη κατά την επιλογή και την υλοποίηση ήταν τόσο το βάρος του σκελετού, το οποίο πρέπει να είναι μικρό ώστε να μην επιβαρύνεται η κίνηση των σερβοκινητήρων, όσο και η στιβαρότητα της κατασκευής.

Η βάση του βραχίονα αποτελείται από μία πλατφόρμα στην οποία θα εγκατασταθεί ένας ακόμη σερβοκινητήρας που θα δίνει τη δυνατότητα περιστροφής 360° (Εικόνα 14) (Sainsmart, 2017).

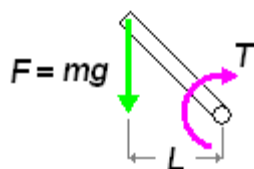


Εικόνα 14: Περιστρεφόμενη βάση βραχίονα (Sainsmart, 2017c).

3.2 Επιλογή σερβοκινητήρων

Η επιλογή των σερβοκινητήρων έγινε βάσει ορισμένων παραγόντων. Αρχικά, η τάση τροφοδοσίας στην πλειοψηφία των ηλεκτρονικών κατασκευών με σερβοκινητήρες είναι στα 5-7 Volts DC. Ωστόσο, τα κυριότερα κριτήρια επιλογής σερβοκινητήρων για έναν βραχίονα είναι η ροπή του εκάστοτε κινητήρα καθώς και η δυνατότητα ελέγχου του (περιστροφή, ταχύτητα, ακρίβεια).

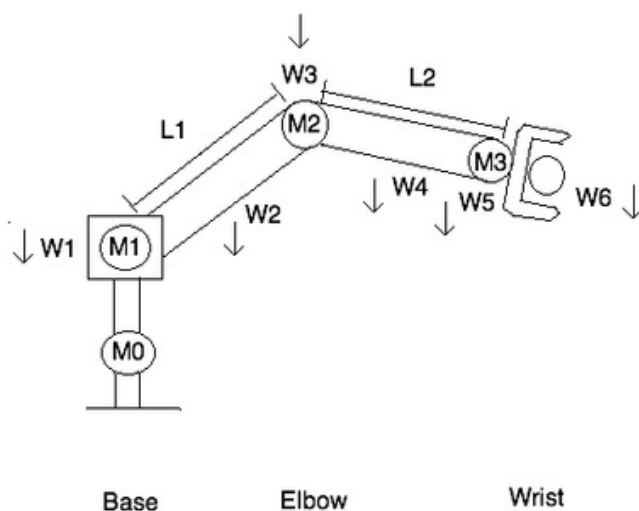
Ως ροπή ορίζεται το διανυσματικό μέγεθος που έχει μέτρο ίσο προς το γινόμενο της δύναμης επί την απόσταση της δύναμης από το σημείο. Κατά όμοιο τρόπο, ροπή δύναμης ως προς τον άξονα είναι το διανυσματικό μέγεθος που έχει ως μέτρο το γινόμενο της δύναμης επί την απόσταση της δύναμης από τον άξονα και φορέα τον άξονα (Σχέδιο 1).



Σχέδιο 1: Αναπαράσταση ροπής (Ιδία σχεδίαση, 2016).

Ο ορισμός της ροπής δίνεται από τη σχέση $T=F*L$ (οπού F η δύναμη και L η απόσταση που ασκείται αυτή). Ως δύναμη στο ρομποτικό βραχίονα ορίζεται το βάρος του κάθε άξονα $F=m*g$ (οπού m η μάζα του αντικειμένου και g η επιτάχυνση της βαρύτητας).

Αντικαθιστώντας, παρατηρείται ότι η ροπή που πρέπει να ασκηθεί ανά κάθε βραχίονα ορίζεται ως $T=(m*g)*L$. Συνεπώς, το μέγιστο φορτίο βρίσκεται στους σερβοκινητήρες, οι οποίοι βρίσκονται στις αρθρώσεις στη βάση του βραχίονα και είναι μικρότερο όσο προσεγγίζεται η άκρη του βραχίονα (gripper) (Σχέδιο 2).



Σχέδιο 2: Αναπαράσταση δυνάμεων σε ρομποτικό βραχίονα (Ιδία σχεδίαση, 2016).

Λόγω της πολυπλοκότητας του ακριβούς υπολογισμού της ροής στις τρεις διαστάσεις, έπειτα από προσέγγιση επιλέχθηκαν 3 σερβοκινητήρες με ροπή 20kg/cm. Από τους εν λόγω σερβοκινητήρες, οι 2 τοποθετήθηκαν στη βάση της κατασκευής και ένας στην πρώτη άρθρωση. Στις 2 τελευταίες αρθρώσεις, καθώς και στον gripper τοποθετήθηκαν 3 σερβοκινητήρες με μικρότερη ροπή 9.5-10 kg/cm (σημειώνεται ότι όλες οι ροπές είναι για την τάση λειτουργίας των 5V) (Serway, 1990).

3.3 Χαρακτηριστικά σερβοκινητήρων

Οι τρεις σερβοκινητήρες μεγάλης ροής που επιλέχθηκαν είναι οι DS3218mg Digital Servo (Εικόνες 15 και 16). Στον Πίνακα 3 παρατίθενται τα τεχνικά τους χαρακτηριστικά.

Πίνακας 3: Τεχνικά χαρακτηριστικά DS3218mg (Aliexpress, 2017a)

Κατασκευαστής: Dsservo

Μοντέλο: DS3218mg High Torque Digital servo

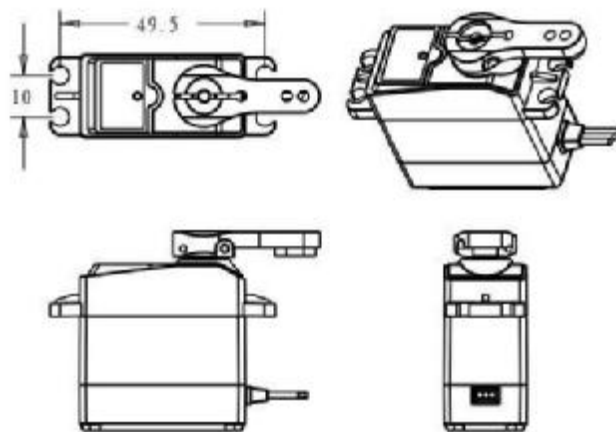
Ροπή (5volt): 19kg/cm

Ταχύτητα περιστροφής: 0.16sec/ 60

Τάση λειτουργίας:	4.8 ~ 6.8 DC Volts
Τύπος γρاناζιών:	Χαλκού και αλουμινίου



Εικόνα 15: Σερβοκινητήρας DS3218mg (Aliexpress, 2017a).



Εικόνα 16: Σχέδιο σερβοκινητήρα DS3218mg (Aliexpress, 2017a).

Οι σερβοκινητήρες χαμηλότερης ροπής που επιλέχθηκαν είναι οι MG996R Tower-pro (Εικόνα 17). Στον Πίνακα 4 παρατίθενται τα τεχνικά τους χαρακτηριστικά:

Πίνακας 4: Χαρακτηριστικά Tower pro MG996R (Torq Pro & Tower Pro 2014).

Κατασκευαστής:	Tower pro
Μοντέλο:	MG996R
Ροπή (5volts):	9.4kg/cm
Ταχύτητα περιστροφής:	0.17 s/60
Τάση λειτουργίας:	4.8V-7.2V
Τύπος γραναζιών:	Χαλκού και αλουμινίου



Εικόνα 17: Σερβοκινητήρας MG 996R και σχέδιο (Torq Pro & Tower Pro 2014).

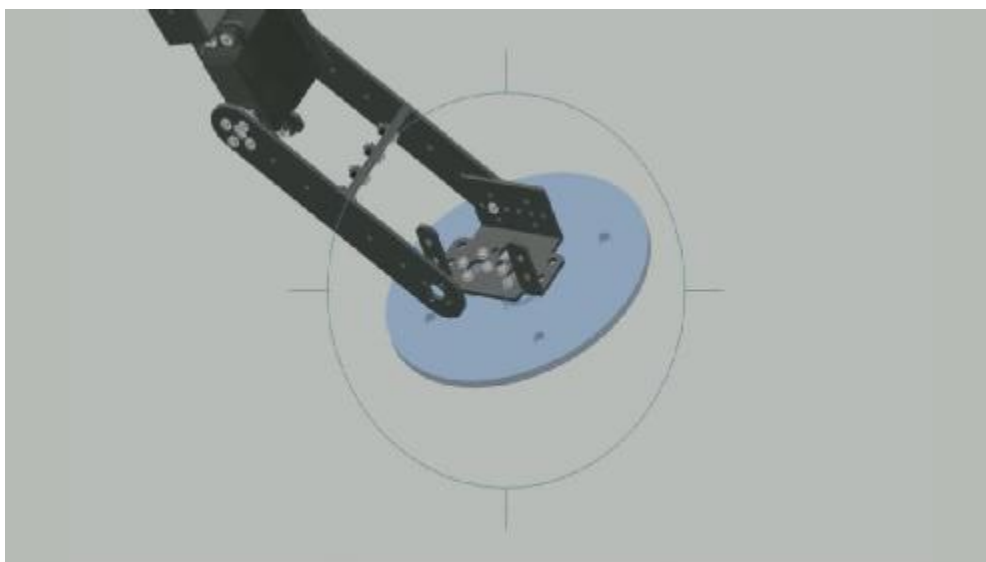
4 Κατασκευή βραχίονα και προγραμματισμός μικροελεγκτή

4.1 Συναρμολόγηση βραχίονα

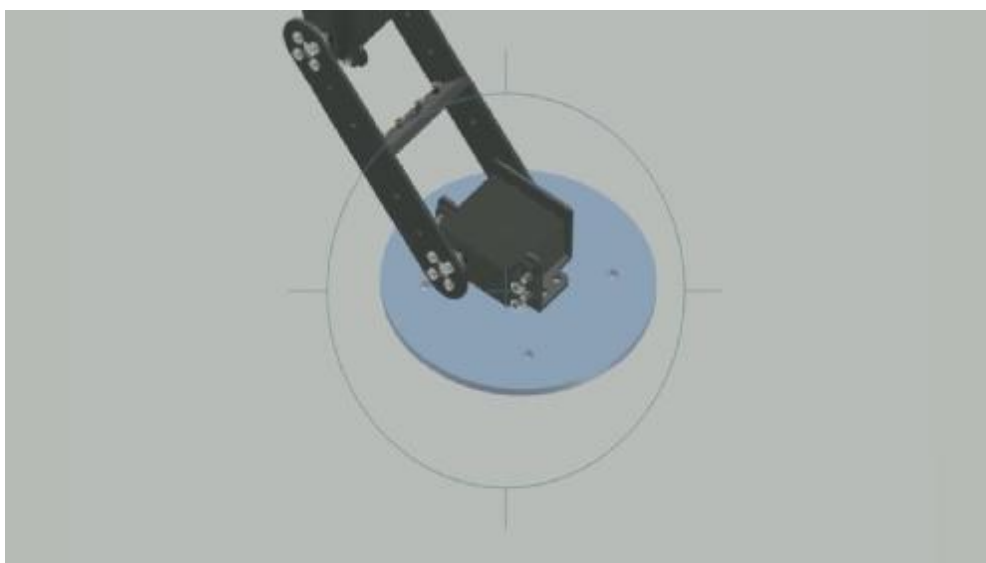
4.1.1 Διαδικασία εγκατάστασης σερβοκινητήρων

Η διαδικασία της εγκατάστασης των σερβοκινητήρων και της συναρμολόγησης του βραχίονα γίνεται με την βοήθεια των οπών στους συνδέσμους αλουμινίου. Οι συγκεκριμένες οπές έχουν την απαραίτητη διάμετρο για την εγκατάσταση των σερβοκινητήρων, καθώς και για την ένωση των εκάστοτε αξόνων μεταξύ τους. Τα βήματα που ακολουθήθηκαν για τη συναρμολόγηση του βραχίονα και την εγκατάσταση των σερβοκινητήρων είναι τα εξής:

1. Βήμα πρώτο: Συναρμολόγηση περιστρεφόμενης βάσης, εγκατάσταση σερβοκινητήρων και εγκατάσταση κατακόρυφου συνδέσμου (Εικόνες 18 και 19).

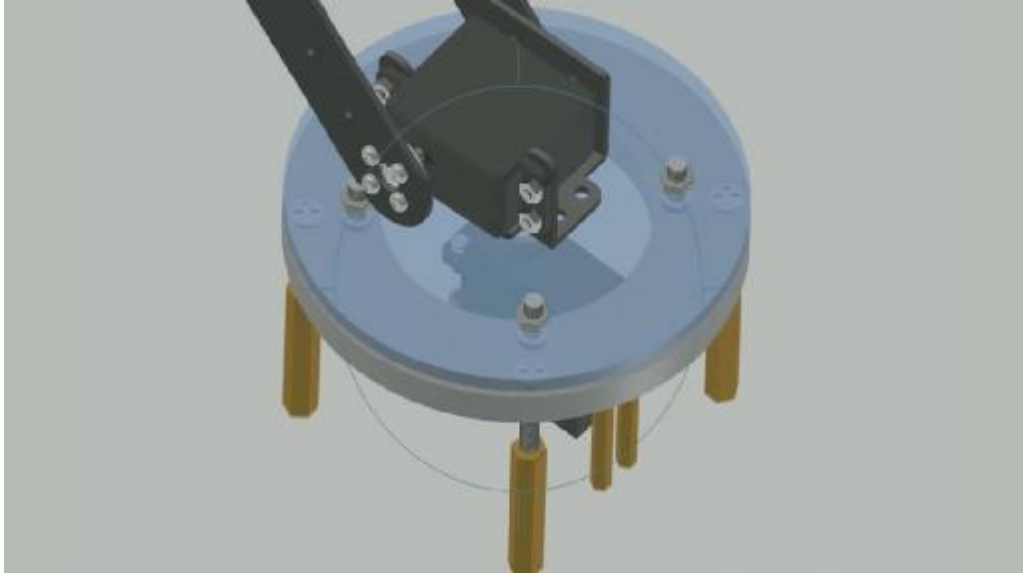


Εικόνα 18: Εγκατάσταση συνδέσμου βάσης (Ιδία σχεδίαση, 2017).

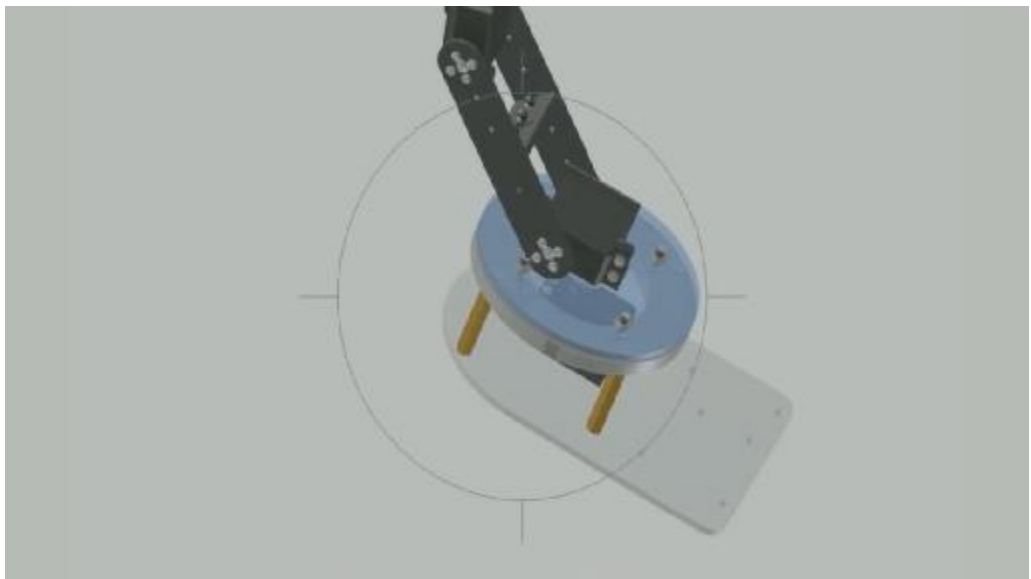


Εικόνα 19: Εγκατάσταση σερβοκινητήρα βάσης (Ιδία σχεδίαση, 2017).

- ☐ Βήμα δεύτερο: Εγκατάσταση σερβοκινητήρα περιστροφής και τελική μορφή βάσης (Εικόνες 20 και 21).



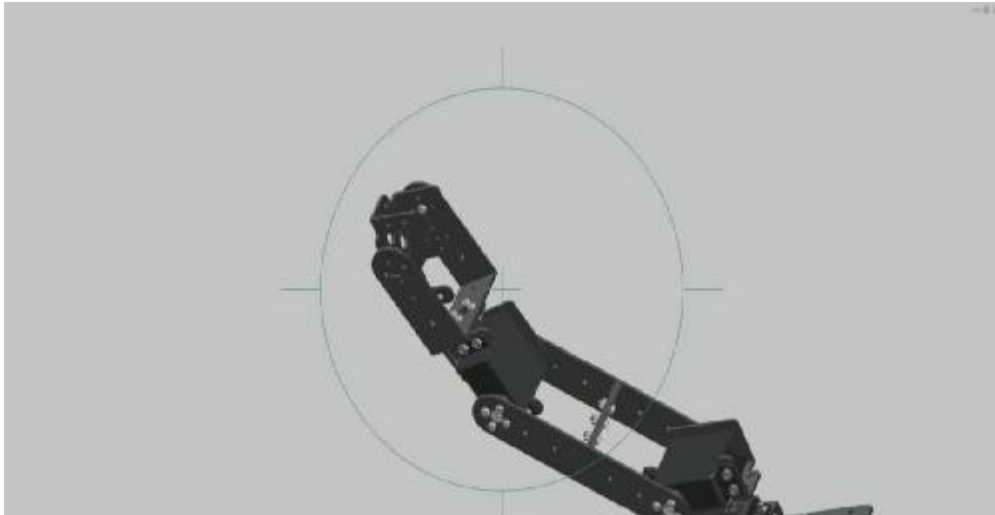
Εικόνα 20: Εγκατάσταση σερβοκινητήρα περιστροφής (Ιδία σχεδίαση, 2017).



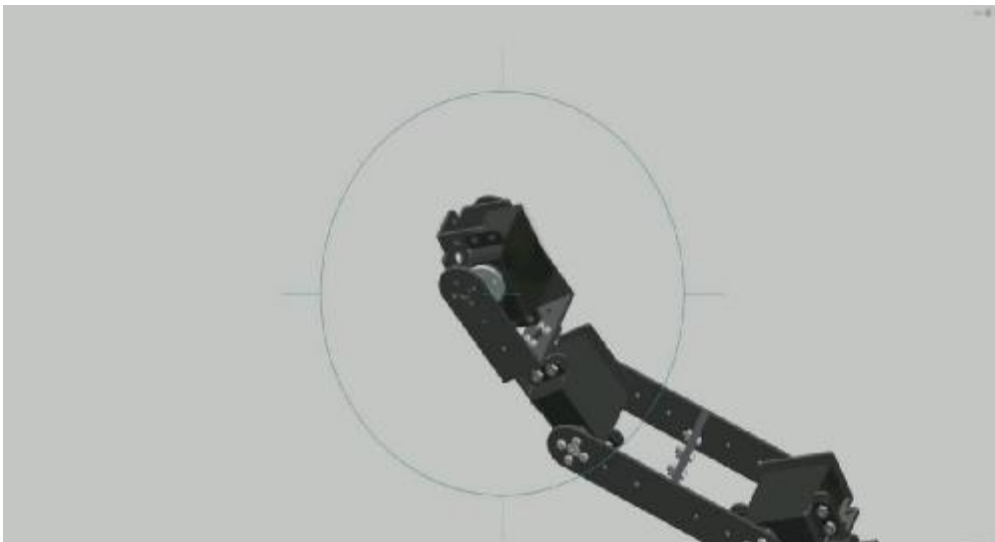
Εικόνα 21: Τελική μορφή βάσης βραχίονα (Ιδία σχεδίαση, 2017).

Όπως φαίνεται στην εικόνα 21, πάνω στον σερβοκινητήρα της βάσης του βραχίονα εφαρμόστηκε ο πρώτος άξονας του βραχίονα, ο οποίος μαζί με τη βάση δίνουν στο σύνολο την κίνηση περιστροφής και την κατακόρυφη κίνηση.

□ Βήμα τρίτο: Εγκατάσταση συνδέσμων και σερβοκινητήρων $turn$ και yaw (Εικόνες 22 και 23).



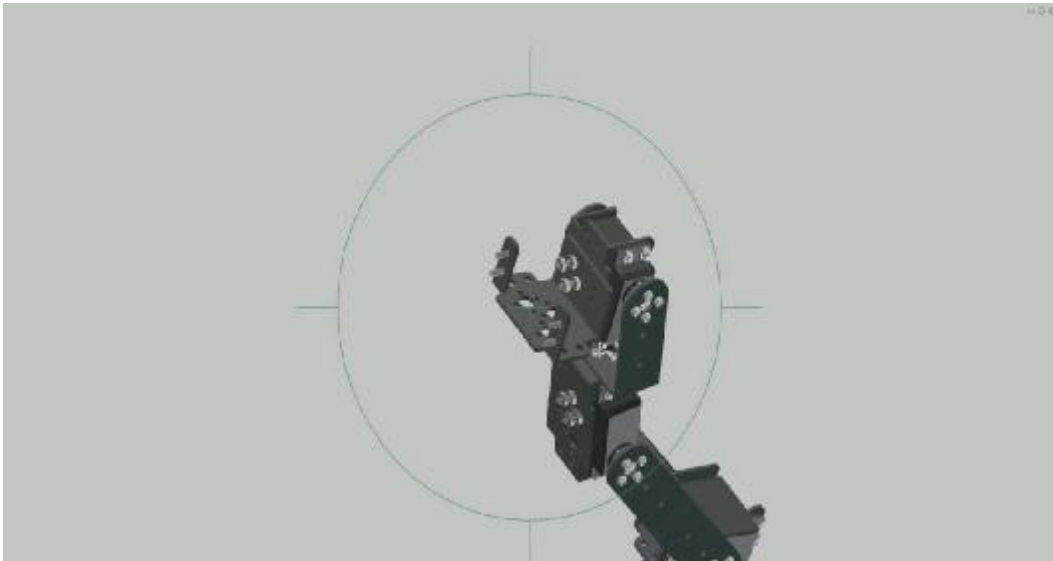
Εικόνα 22: Εγκατάσταση συνδέσμου yaw (Ιδία σχεδίαση, 2017).



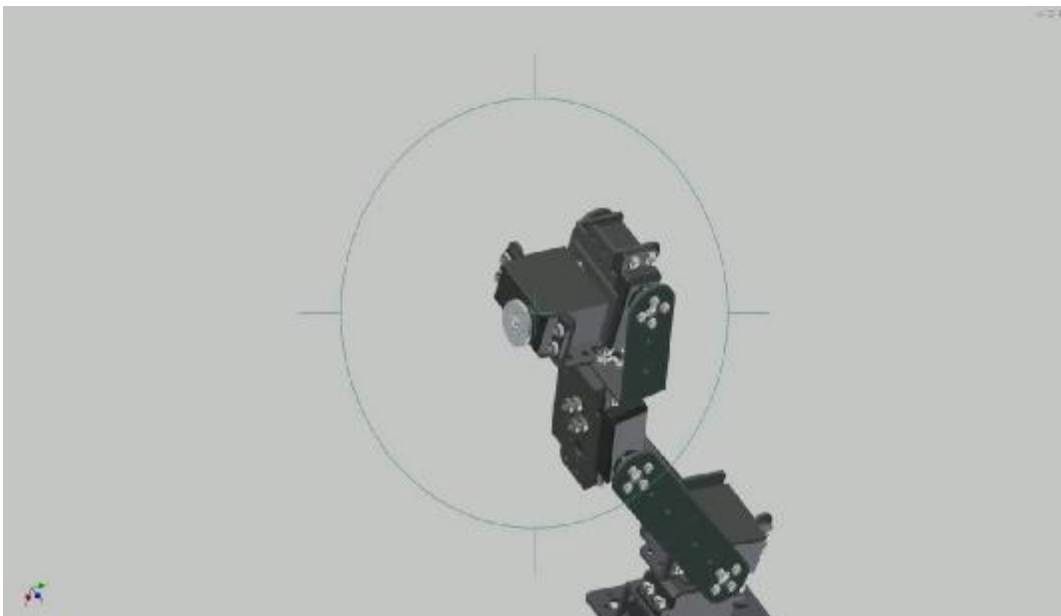
Εικόνα 23: Εγκατάσταση σερβοκινητήρα yaw (Ιδία σχεδίαση, 2017).

Στην κορυφή του κατακόρυφου άξονα, όπως φαίνεται και στην εικόνα 23, γίνεται εγκατάσταση των συνδέσμων και των σερβοκινητήρων που επιτρέπουν την κίνηση $turn$ και yaw .

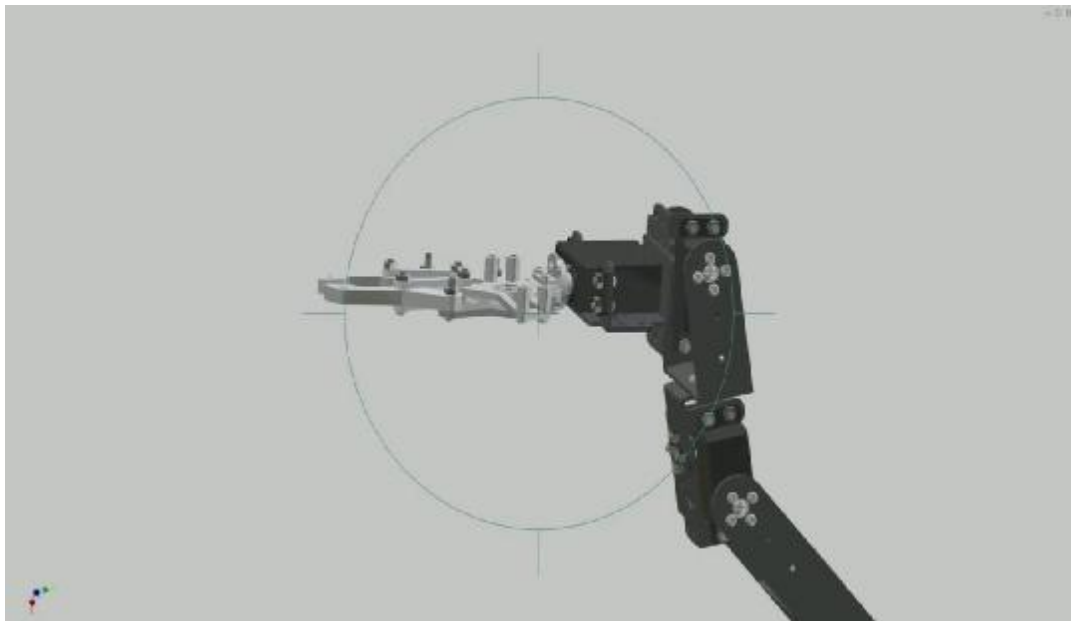
☐ Βήμα τέταρτο: Εγκατάσταση gripper (Εικόνες 24, 25, 26 και 27).



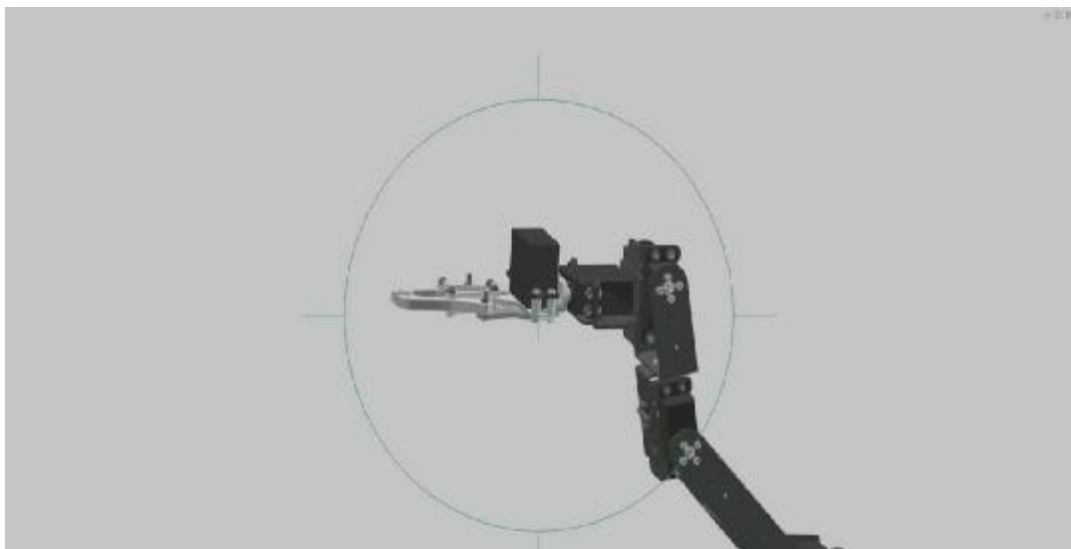
Εικόνα 24: Εγκατάσταση συνδέσμου Turn (Ιδία σχεδίαση, 2017).



Εικόνα 25: Εγκατάσταση σερβοκινητήρα Turn (Ιδία σχεδίαση, 2017).

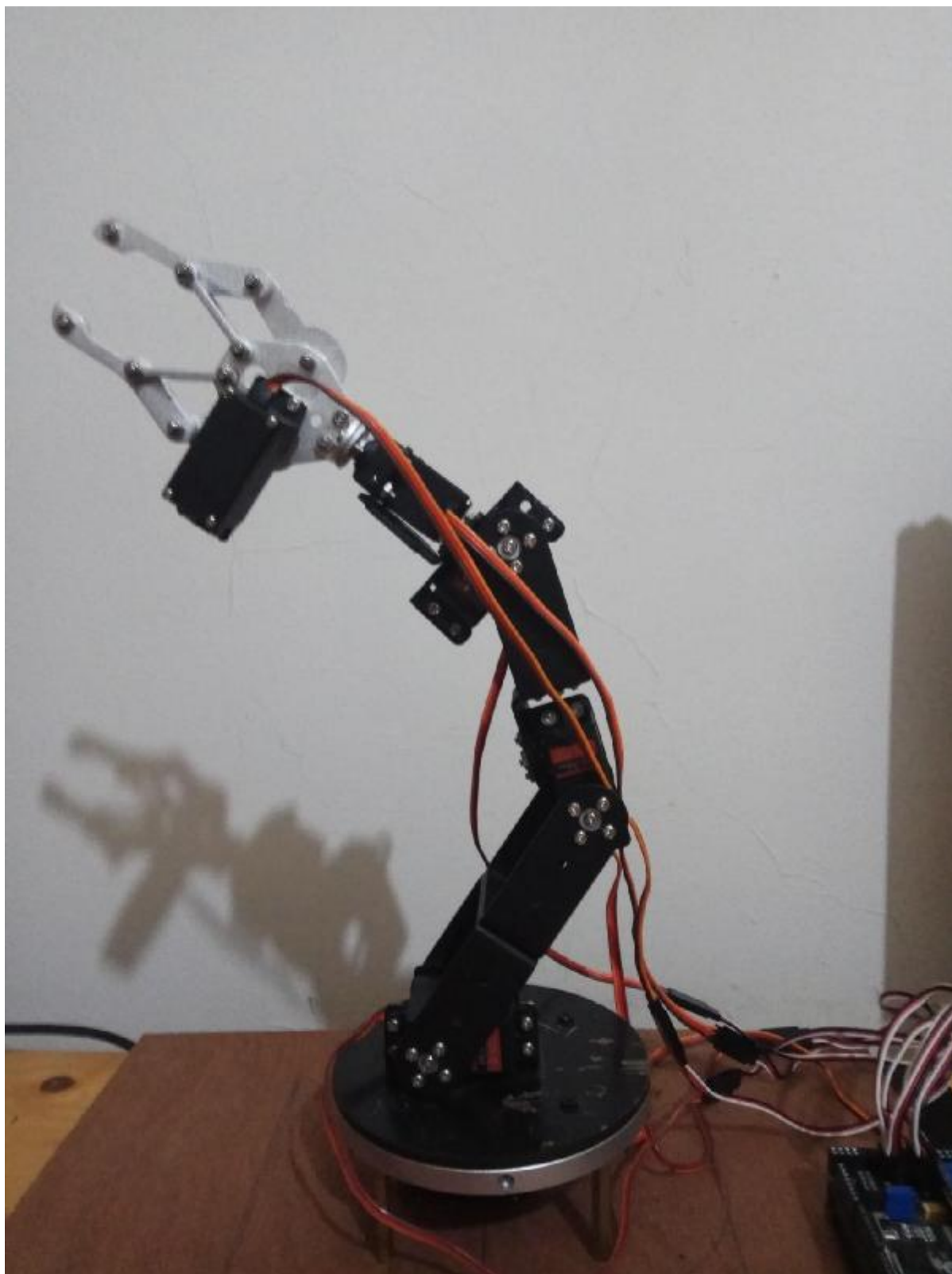


Εικόνα 26: Εγκατάσταση Gripper (Ιδία σχεδίαση, 2017).



Εικόνα 27: Εγκατάσταση σερβοκινητήρα Gripper (Ιδία σχεδίαση, 2017).

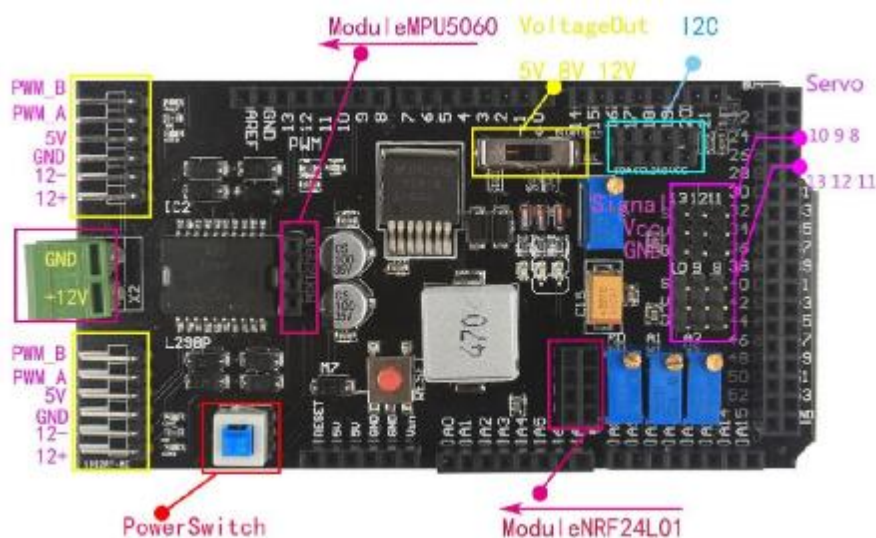
Στο τελικό βήμα γίνεται η εγκατάσταση των σερβοκινητήρων που επιτρέπουν την περιστροφή του gripper, καθώς και του τελευταίου σερβοκινητήρα, ο οποίος επιτρέπει στο gripper να ανοίγει και να κλείνει (Εικόνα 28).



Εικόνα 28: Τελική μορφή ρομποτικού βραχίονα (Ιδία λήψη, 2017).

4.1.2 Εγκατάσταση Βοηθητικών κυκλωμάτων (shields) στο μικροελεγκτή και σύνδεση σερβοκινητήρων

Ο μικροελεγκτής Arduino Mega 2650, που χρησιμοποιήθηκε για τον έλεγχο των σερβοκινητήρων, δε δύναται να τροφοδοτήσει ταυτόχρονα και τους 6 σερβοκινητήρες. Το συγκεκριμένο πρόβλημα δημιουργείται τόσο λόγω του περιορισμού σε αναλογικές εξόδους, όσο και του περιορισμού ρεύματος που μπορεί να «δώσει» ο μικροελεγκτής την ίδια χρονική στιγμή στις PWM εξόδους. Για το λόγο αυτό χρησιμοποιήθηκε το InstaBots Robot Controller Shield for Arduino MEGA2560 R3 Robot Arm Control της Sainsmart (Εικόνα 29) (Πίνακας 5).



Εικόνα 29: Sainsmart InstaBots Robot Controller Shield for Arduino MEGA2560 R3 Robot Arm Control (Sainsmart, 2017d).

Πίνακας 5: Χαρακτηριστικά InstaBots Robot Controller Shield for Arduino MEGA2560 R3 Robot Arm.(Sainsmart,2017d).

Τάση εισόδου 12Volt (Δυνατότητα χρήσης μπαταρίας 3S lipo)

Τάση/ρεύμα εξόδου 5Volt-5Ampere, 8Volt-4Ampere (ρυθμιζόμενο)

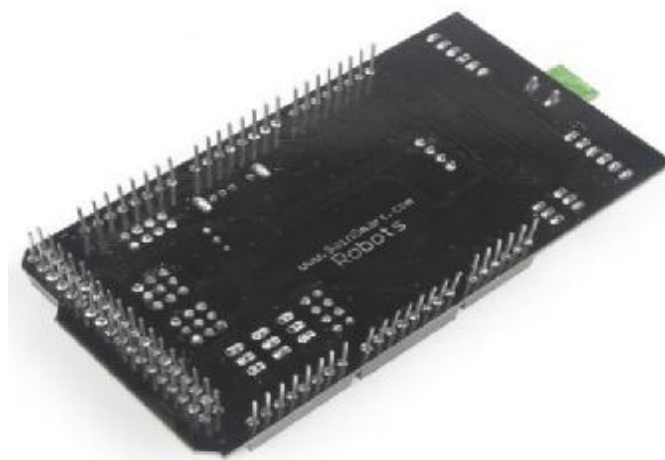
Έξοδοι PWM 6 PWM έξοδοι για 6 σερβοκινητήρες

Είσοδοι Module ModuleNRF24L01, ModuleMPU5060, I2C Οθόνη

Το συγκεκριμένο shield, το οποίο είναι ειδικά σχεδιασμένο για αυτού του είδους τις εφαρμογές, δίνει τη δυνατότητα του ελέγχου των 6 σερβοκινητήρων ταυτόχρονα, χωρίς να ελαττώσει τις εισόδους που παρέχει το Arduino mega. Επίσης, χρησιμοποιώντας το Vin του shield το οποίο λειτουργεί σε τάση 12Volt δίνεται η δυνατότητα χρήσης εξωτερικού μετασχηματιστή πρίζας, ο οποίος θα μπορέσει να τροφοδοτήσει με το απαραίτητο ρεύμα και τους 6 σερβοκινητήρες με την χρήση ρελέ. Το εν λόγω εγχείρημα θα ήταν αδύνατο με το Arduino, λόγω του περιορισμού ρεύματος που το συνοδεύει.

Όπως φαίνεται και στην εικόνα 29, το shield προσφέρει εκτός από τις εξόδους του Arduino mega και τις 6 εξόδους για του σερβοκινητήρες, συγκεκριμένες εξόδους για modules, όπως την έξοδο ModuleNRF24L01 η οποία θα χρησιμοποιηθεί για τον ασύρματο έλεγχο του βραχίονα. Εκτός του εν λόγω shield παρέχονται είσοδοι για modules γυροσκοπίου, οθόνης και έξτρα κινητήρων οι οποίοι δε θα χρησιμοποιηθούν στη συγκεκριμένη εφαρμογή.

Όσον αφορά στην εγκατάσταση του shield στον μικροελεγκτή Arduino Mega είναι εξαιρετικά απλή. Όπως η πλειοψηφία των shields, τα οποία είναι σχεδιασμένα για τους μικροελεγκτές Arduino, το μόνο που αρκεί είναι η εφαρμογή των pins πάνω στην πλακέτα του Arduino mega (Εικόνα 30) (Sainsmart, 2017d).



Εικόνα 30: Sainsmart InstaBots Robot Controller Shield πίσω μέρος (Sainsmart,2017d).

4.2 Συναρμολόγηση χειριστηρίου

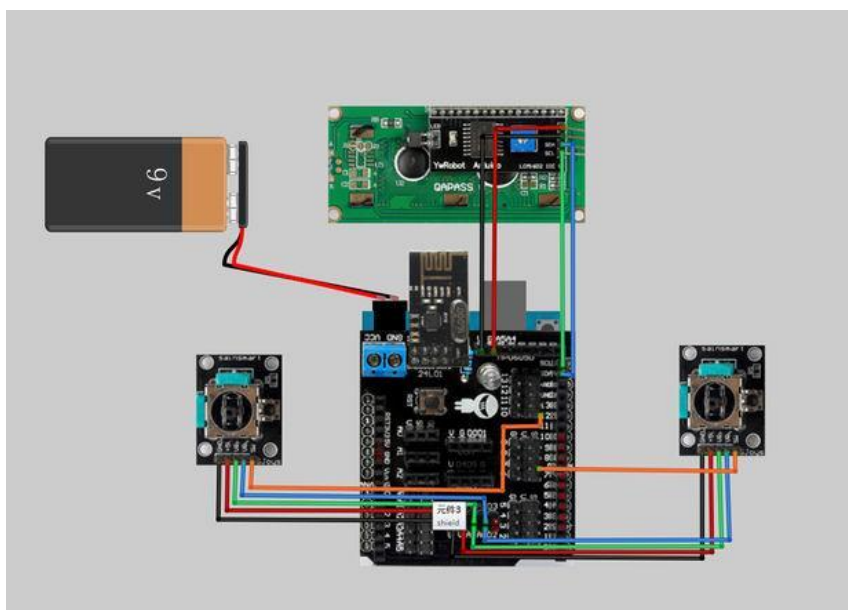
Το χειριστήριο με το οποίο ελέγχεται ο βραχίονας βασίζεται στο μικροελεγκτή Arduino Uno. Με τη βοήθεια του shield instabots remote της Sainsmart δίνεται η δυνατότητα να εγκατασταθούν στο Arduino UNO τα thumbsticks ελέγχου, η LCD οθόνη, καθώς και το NRF24L01+ Wireless Transceiver Module 2.4GHz (Εικόνα 31) το οποίο κάνει δυνατό τον ασύρματο έλεγχο. (Sainsmart, 2017e)



Εικόνα 31: NRF24L01+ Wireless Transceiver Module 2.4GHz ISM band (Sainsmart, 2017e).

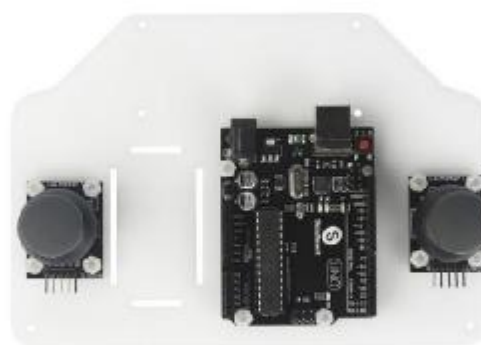
4.2.1 Διαδικασία εγκατάστασης Arduino και συνδεσμολογία χειριστηρίου

Όπως φαίνεται στην εικόνα 32, για την εγκατάσταση τόσο της οθόνης όσο και των thumbsticks, εφαρμόστηκε στο Arduino Uno το remote shield της Sainsmart. Όπως και στο Arduino Mega, η εγκατάστασή του είναι εύκολη, εφόσον εφαρμόζεται στα ήδη υπάρχοντα pin του Arduino Uno χωρίς την ανάγκη για χρήση καλωδίων.

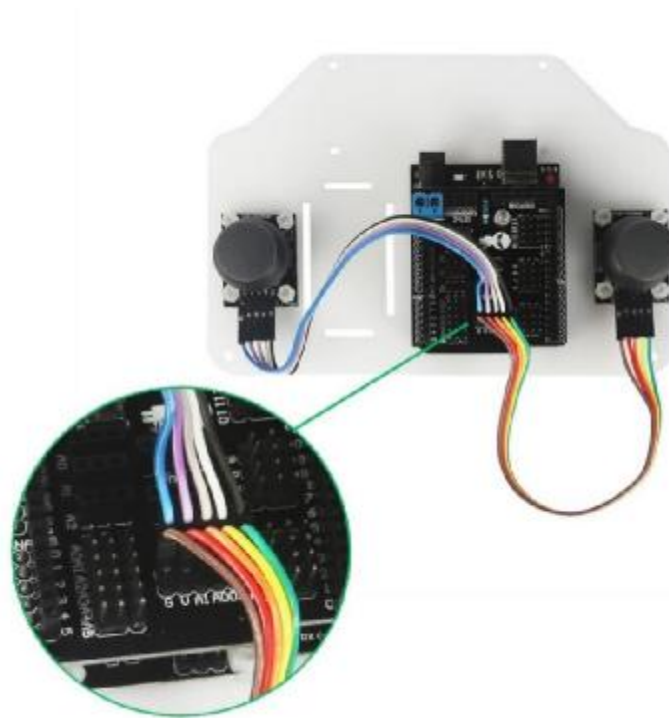


Εικόνα 32: Σχέδιο συναρμολόγησης χειριστήριου (Ιδία σχεδίαση,2017).

Το thumb stick (ή αλλιώς analog stick) αποτελεί μια συσκευή εισόδου για τον ελεγκτή που χρησιμοποιείται για διαδιάστατες εισόδους. Ουσιαστικά, η τιμή της εισόδου βασίζεται στη θέση της προεξοχής του σε σχέση με την κεντρική θέση. Το πλεονέκτημα της χρήσης τους, σε σχέση με τη χρήση διακοπών button, είναι ότι λειτουργούν με συνεχή ηλεκτρική ενέργεια που «τρέχει» μέσω ποτενσιόμετρων, προσφέροντας μεγαλύτερη ακρίβεια σε σχέση με τη χρήση απλών διακοπών button. Για τον λόγο αυτό χρησιμοποιήθηκε η αναλογική είσοδος του shield, διότι το σήμα είναι αναλογικό και όχι ψηφιακό. Αξίζει να σημειωθεί ότι ένα thumb stick αντιστοιχεί στην κίνηση 2 σερβοκινητήρων στο βραχίονα (Εικόνες 33 και 34) (Wikipedia, 2017).

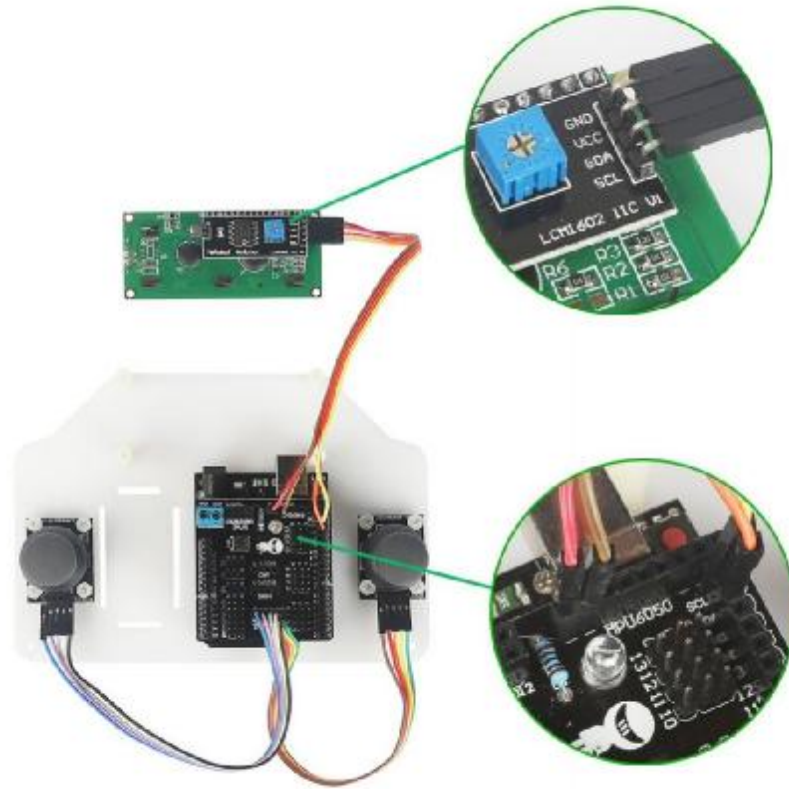


Εικόνα 33: Εγκατάσταση Uno και Thumbsticks στο χειριστήριο (Sainsmart,2017f).

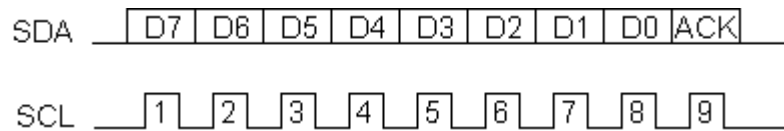


Εικόνα 34: Εγκατάσταση Shield και σύνδεση thumbsticks (Sainsmart, 2017f).

Η σύνδεση μιας οθόνης τύπου LCD (Εικόνα 35) στο shield του μικροελεγκτή Arduino Uno αποτελείται από 4 συνδέσεις: GND, 5V, SDA και SCL. Οι εισόδους GND και 5V αποτελούν την τροφοδοσία της οθόνης και αντίστοιχα συνδέονται με τις εισόδους V και G του shield. Η είσοδος SDA αποτελεί την είσοδο δεδομένων της οθόνης. Από την εν λόγω είσοδο τα δεδομένα που καταχωρούνται στη μνήμη του χειριστηρίου, οδηγούνται στην οθόνη για να προβάλλονται στο χρήστη. Η είσοδος SCL της οθόνης αποτελεί την είσοδο ρολογιού. Το ρολόι καθορίζει τη συχνότητα με την οποία τα δεδομένα μεταφέρονται από τη θέση μνήμης του ελεγκτή και εμφανίζονται στην οθόνη, όπως φαίνεται και στο σχέδιο 3.



Εικόνα 35: Εγκατάσταση Οθόνης LCD (Sainsmart, 2017f).



Σχέδιο 3: Διάγραμμα χρονισμού οθόνης I2C (Stack Exchange Inc., 2017).

Όσον αφορά στην τροφοδοσία του Arduino και του shield, για να πραγματοποιηθεί ασύρματος χειρισμός του βραχίονα, χρησιμοποιήθηκε ένας αντάπτορας μπαταρίας 9V (Εικόνα 36). Ο συγκεκριμένος αντάπτορας συνδέεται στην είσοδο τροφοδοσίας του Arduino, ούτως ώστε να παράσχει αρκετή ενέργεια τόσο για τη λειτουργία του μικροελεγκτή όσο και της οθόνης.



Εικόνα 36: Αντάπτορας μπαταρίας 9V (Aliexpress, 2017b).

Το τελευταίο module που εγκαθίσταται είναι το NRF24L01+ Wireless Transceiver Module 2.4GHz, ίδιο με αυτό που εγκαταστάθηκε στο shield του βραχίονα. Η επικοινωνία των δύο NRF24L01+ είναι αυτή που προσφέρει ασύρματο έλεγχο. Για αισθητικούς λόγους μπορεί να εφαρμοστεί το καπάκι του χειριστηρίου, το οποίο καλύπτει τις συνδέσεις (Εικόνα 37) (Sainsmart, 2017f).



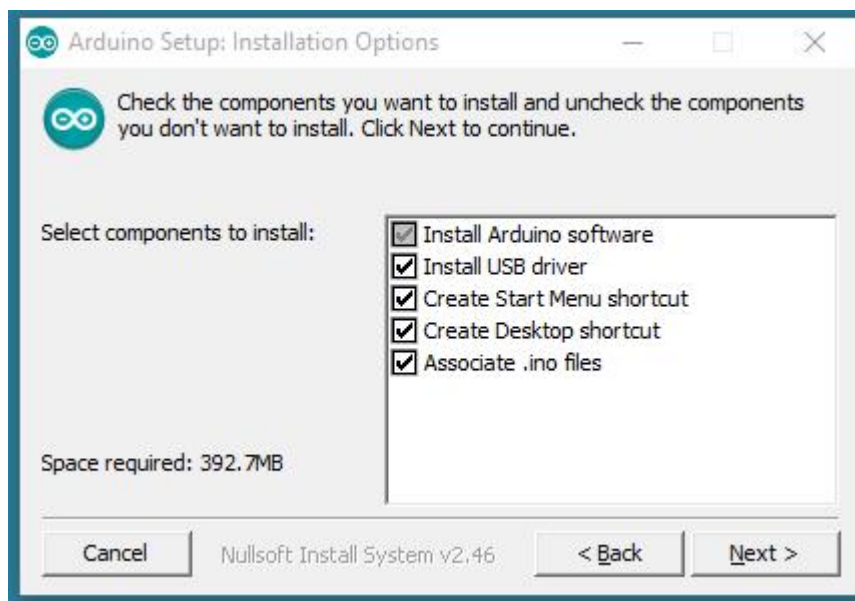
Εικόνα 37: Τελική μορφή χειριστηρίου (Sainsmart, 2017f).

4.3 Πειραματική διαδικασία

4.3.1 Σύνδεση Arduino με Η/Υ

Η σύνδεση μιας πλακέτας Arduino σε έναν ηλεκτρονικό υπολογιστή είναι μια σχετικά εύκολη διαδικασία. Αρχικά, απαιτείται η εγκατάσταση του προαναφερόμενου Arduino IDE το οποίο βρίσκεται στην εξής ιστοσελίδα: <https://www.Arduino.cc/en/main/software>.

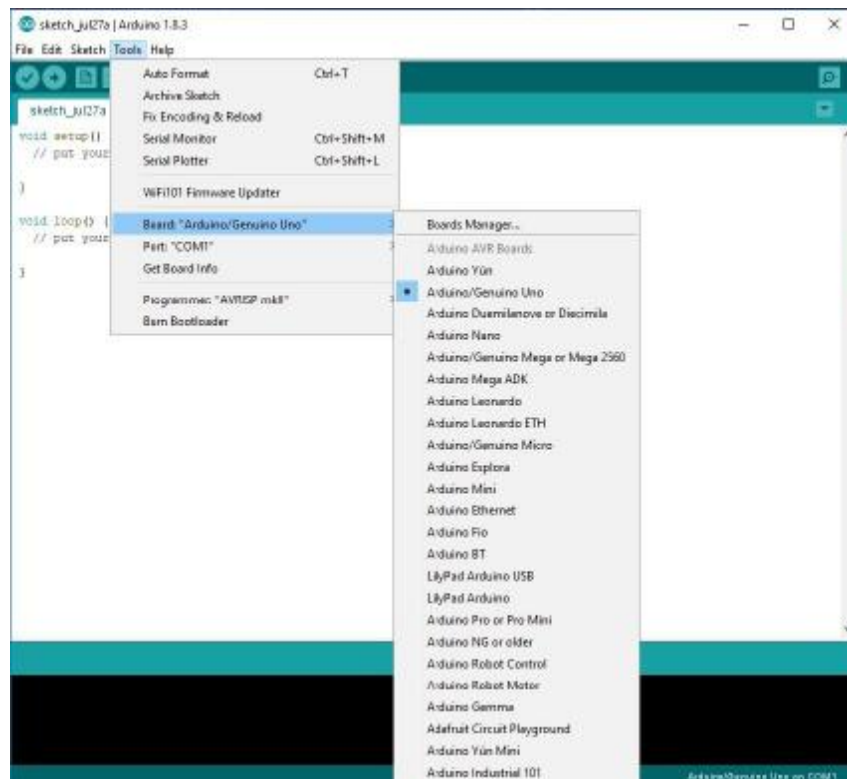
Πρόκειται για ένα εκτελέσιμο .exe αρχείο, το οποίο εκτός από το περιβάλλον προγραμματισμού των μικροελεγκτών Arduino περιλαμβάνει και τους απαραίτητους οδηγούς (drivers) με τους οποίους αναγνωρίζεται και επικοινωνεί ο μικροελεγκτής με τον ηλεκτρονικό υπολογιστή (Εικόνα 38) (Arduino, 2017e).



Εικόνα 38: Παράθυρο εγκατάστασης Arduino IDE (Arduino, 2017e).

Μετά την εγκατάσταση του προγράμματος, με τη σύνδεση του μικροελεγκτή με ένα καλώδιο USB τύπου A-B, ο υπολογιστής αναγνωρίζει τη συσκευή και είναι έτοιμος να ανεβάσει οποιονδήποτε κώδικα στο Arduino.

Όσον αφορά στη διαδικασία upload, δηλαδή το ανέβασμα του κώδικα στο Arduino, πραγματοποιούνται τα εξής βήματα: επιλογή Tools, επιλογή boards, επιλογή του συνδεδεμένου μικροελεγκτή και επιλογή port για τη θύρα στην οποία είναι συνδεδεμένος (Εικόνα 39).



Εικόνα 39: Επιλογή board στο Arduino IDE (Ιδία λήψη, 2017).

4.3.2 Βιβλιοθήκες

Οι βιβλιοθήκες είναι αρχεία γραμμένα σε γλώσσα προγραμματισμού C ή C ++ (αρχεία με επέκταση .c, .cpp) που παρέχουν στο προς υλοποίηση πρόγραμμα πρόσθετη λειτουργικότητα (π.χ. δυνατότητα ελέγχου οθόνης LCD ή ασύρματη επικοινωνία κ.λπ.) και αποτελούν βασικό κομμάτι του κώδικα. Η χρήση τους είναι απλή, καθώς εισάγοντας στην αρχή του κώδικα την εντολή #include και το όνομα της βιβλιοθήκης η οποία έχει αποθηκευθεί στον υποφάκελο libraries στο φάκελο εγκατάστασης του Arduino IDE (Arduino, 2017f).

Για τις ανάγκες του ρομποτικού βραχίονα χρησιμοποιήθηκε πληθώρα βιβλιοθηκών λόγω της πολυπλοκότητας της λειτουργίας του. Πιο αναλυτικά:

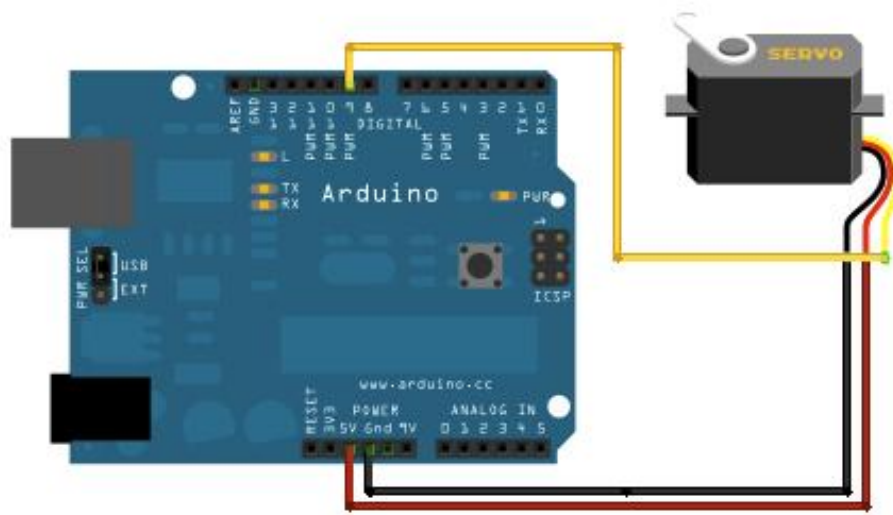
- Βιβλιοθήκη servo.h: Η συγκεκριμένη βιβλιοθήκη επιτρέπει στο μικροελεγκτή Arduino τον έλεγχο σερβοκινητήρων. Οι σερβοκινητήρες έχουν ενσωματωμένα γρανάζια και έναν άξονα που μπορεί να ελεγχθεί με σχετική ακρίβεια. Ο έλεγχός τους περιορίζεται σε γωνίες μεταξύ 0 και 180 μοιρών. Κατά τον έλεγχο με ποτενσιόμετρο (εν προκειμένω thumb stick) οι τιμές μεταξύ 0-180 μοιρών μεταφράζονται σε τιμές μεταξύ 0 έως 1023 (Arduino, 2017g).
- Βιβλιοθήκη LiquidCrystal_I2C: Η εν λόγω βιβλιοθήκη αποτελεί τροποποίηση της βιβλιοθήκης LiquidCrystal, η οποία επιτρέπει τον έλεγχο μιας οθόνης LCD μέσω I2C (μορφή επικοινωνίας πολλαπλών masters-slaves) (Codebender Inc. 2017).
- Βιβλιοθήκη SPI.h: Το Serial Peripheral Interface (SPI) είναι ένα σύγχρονο πρωτόκολλο σειριακών δεδομένων που χρησιμοποιείται από μικροελεγκτές για επικοινωνία με μία ή περισσότερες περιφερειακές συσκευές γρήγορα και σε μικρές αποστάσεις. Μπορεί, επίσης, να χρησιμοποιηθεί για επικοινωνία μεταξύ

δύο μικροελεγκτών. Στην παρούσα κατασκευή το εν λόγω πρωτόκολλο επικοινωνίας χρησιμοποιήθηκε για τον ασύρματο έλεγχο και την ανταλλαγή δεδομένων των 2 μικροελεγκτών (Arduino, 2017h).

- ☐ Βιβλιοθήκη NRF24L01.h: Η συγκεκριμένη βιβλιοθήκη είναι απαραίτητη για τον προγραμματισμό του NRF24L01+ Wireless transceiver module. Παρέχει τον ορισμό των εισόδων-εξόδων και των θέσεων μνήμης του Module για την επίτευξη επικοινωνίας. (Github Inc, 2017a).
- ☐ Βιβλιοθήκες Mirf.h και MirfHardwareSpiDriver.h: Ομοίως με τη βιβλιοθήκη NRF24L01.h, οι εν λόγω βιβλιοθήκες είναι απαραίτητες για την επικοινωνία μεταξύ των 2 Wireless transceiver module καθώς καθορίζουν χαρακτηριστικά όπως το συγχρονισμό και τα power states του module. Τονίζεται ότι λειτουργούν μέσω συνεργασίας με τις βιβλιοθήκες SPI.h και NRF24L01.h (Github Inc, 2017b).

4.3.3 Πείραμα ορθής λειτουργίας σερβοκινητήρων και calibration

Όπως προαναφέρθηκε, ο έλεγχος των σερβοκινητήρων μέσω του μικροελεγκτή Arduino γίνεται με την βοήθεια της βιβλιοθήκης servo.h. Η σύνδεση των σερβοκινητήρων αποτελείται από 3 καλώδια: power, ground και signal. Συνδέοντας το σερβοκινητήρα στις θύρες 5Volt και ground του μικροελεγκτή, καθώς και σε μία από τις θύρες PWM, γίνεται δυνατός ο έλεγχος για την ορθή λειτουργία του (Εικόνα 40). Το εν λόγω πείραμα έλαβε χώρα και στους 6 σερβοκινητήρες που εγκαταστάθηκαν στο βραχίονα με σκοπό τον έλεγχο τους. Έτσι, εντοπίστηκαν τα σημεία min και max του κάθε σερβοκινητήρα, τα οποία είναι εξαιρετικά χρήσιμα για τον προγραμματισμό και την υλοποίηση του βραχίονα.



Εικόνα 40: Σχέδιο σύνδεσης σερβοκινητήρα σε μικροελεγκτή Arduino (Arduino, 2017i).

```
#include <Servo.h> // Εντολή εισαγωγής βιβλιοθήκης servo.h
Servo myservo; // Δημιουργία αντικειμένου σέρβοκινητήρα
int pos = 0; // Μεταβλητή θέσης σερβοκινητήρα, αρχικοποίηση στο σημείο 0
```

```

void setup() {
  myservo.attach(9); // Ορισμός του pin 9 ως είσοδος σερβοκινητήρα
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // Κίνηση σερβοκινητήρα από 0-180 μοίρες
    // με βήμα 1
    myservo.write(pos);           // Μετακίνηση του σερβοκινητήρα στην τιμή της
    // μεταβλητής pos
    delay(15);                    // Καθυστέρηση 15ms για να φτάσει στην θέση ο σερβοκινητήρας
  }
  for (pos = 180; pos >= 0; pos -= 1) { // Κίνηση σερβοκινητήρα από 180-0 μοίρες
    myservo.write(pos);           // Μετακίνηση του σερβοκινητήρα στην τιμή της
    // μεταβλητής pos
    delay(15);                    // Καθυστέρηση 15ms για να φτάσει στη θέση ο σερβοκινητήρας
  }
}

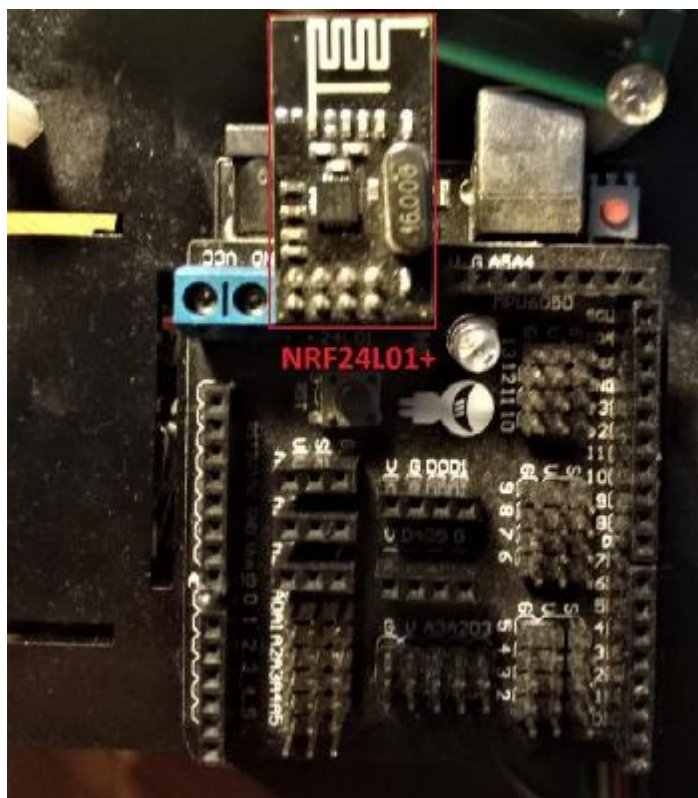
```

Ενδεικτικά στον παρών κώδικα, μετά την αρχικοποίηση της μεταβλητής θέσης του σερβοκινητήρα, ακολουθείται μια διαδικασία βηματικής κίνησής του από το σημείο 0 έως το 180. Όταν ο σερβοκινητήρας φτάσει στην τελική θέση (pos = 180) επιστρέφει με το ίδιο βήμα στο σημείο 0.

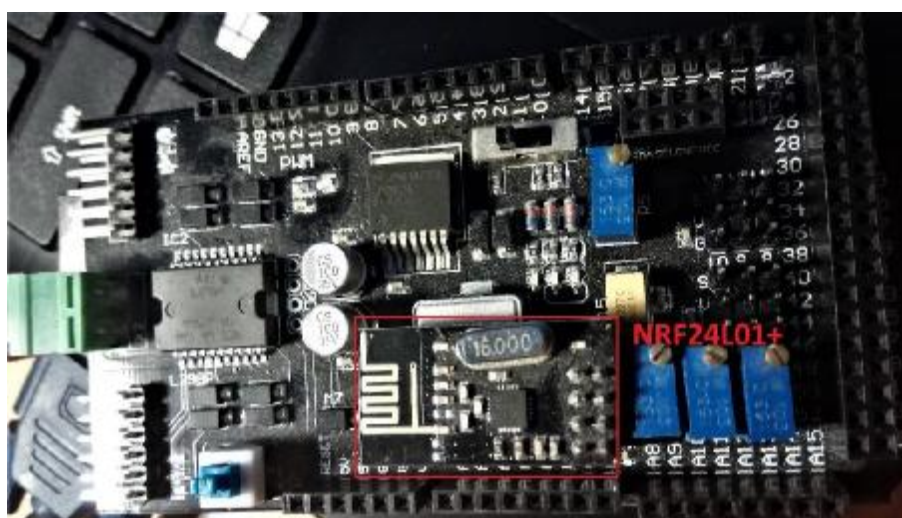
Μετά τη συγγραφή του προγράμματος και αφού έγινε compile, έγινε upload του προγράμματος στο Arduino uno. Τα αποτελέσματα έδειξαν ότι η συμπεριφορά και των 6 σερβοκινητήρων ήταν η αναμενόμενη. Συγκεκριμένα, ξεκίνησαν από το σημείο 0, με βήμα 1, έπειτα έφτασαν στο μέγιστο σημείο των 180 μοιρών και τέλος επέστρεψαν πάλι με βήμα 1 στην αρχική τους θέση 0.

4.3.4 Πείραμα επικοινωνίας μεταξύ Arduino Mega και Arduino Uno μέσω NRF24L01+

Σκοπός του πειράματος είναι η κατανόηση του τρόπου επικοινωνίας μεταξύ των δύο μικροελεγκτών με την χρήση του Module NRF24L01+. Πρόκειται για ένα βασικό παράδειγμα στο οποίο ο ένας μικροελεγκτής (Arduino Uno) έχει το ρόλο του Server, και ο δεύτερος (Arduino Mega) το ρόλο του receiver. Μέσω της συγκεκριμένης επικοινωνίας αποστέλλεται διαρκώς ένα μήνυμα, το οποίο εμφανίζεται στο serial monitor του Arduino IDE, ώστε να ελεγχθεί η επιτυχία της σύνδεσης. Αξίζει να σημειωθεί ότι το πείραμα έλαβε χώρα με τα δυο module εγκατεστημένα επάνω στα shield, οπότε η σύνδεσή του διαφέρει από την τυπική σύνδεση στους μικροελεγκτές (Εικόνες 41 και 42). Το εν λόγω εγχείρημα έλαβε χώρα προκειμένου να ελεγχθεί η ορθή λειτουργία όλων των Modules.



Εικόνα 41: Εγκατάσταση NRF24L01+ στο Shield του Arduino Uno (Ιδία λήψη, 2017).



Εικόνα 42: Εγκατάσταση NRF24L01+ στο Shield του Arduino Mega (Ιδία λήψη, 2017).

Κώδικας Server_test

```
#include <SPI.h> // Δήλωση βιβλιοθηκών
#include <Mirf.h>
#include <nRF24L01.h>
#include <MirfHardwareSpiDriver.h>
void setup(){
  Serial.begin(9600); //Ορισμός ταχύτητας σειριακής επικοινωνίας
  Mirf.spi = &MirfHardwareSpi; // εκκίνηση βιβλιοθήκης
  Mirf.init();
  Mirf.setTADDR((byte *)"serv1"); //ορισμός διεύθυνσης Transmit
  Mirf.payload = 32; //Μέγεθος δεδομένων αποστολής
  Mirf.config();
  Serial.println("Beginning ... "); //Μήνυμα αρχής επικοινωνίας
}
void loop(){
  Mirf.send((byte *) "Hello"); //Στέλνει το μήνυμα "hello" συνεχόμενα
  delay(1000); // Αποστολή δεδομένων ανά δευτερόλεπτο
}
```

Κώδικας Reciever_test

```
#include <SPI.h> // Δήλωση βιβλιοθηκών
#include <nRF24L01.h>
#include <MirfHardwareSpiDriver.h>
#include <Mirf.h>

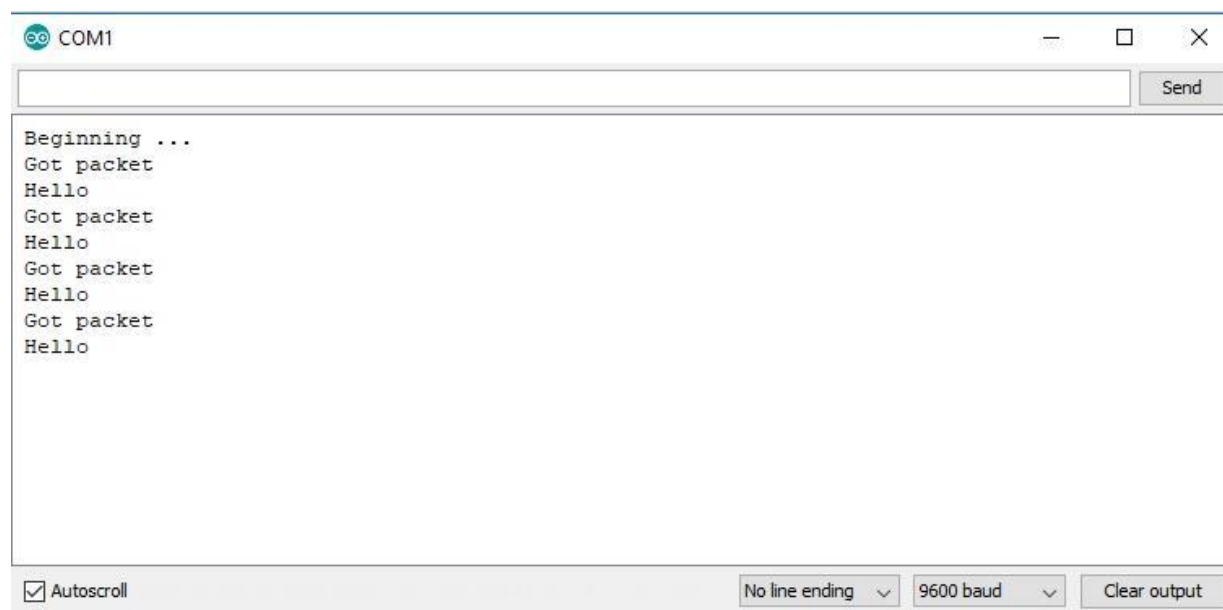
void setup(){
  Serial.begin(9600); //Ορισμός ταχύτητας σειριακής επικοινωνίας
  Mirf.spi = &MirfHardwareSpi;
  Mirf.csnPin = 48; // Ορισμός pin CSN στο Arduino
  Mirf.cePin = 53; // Ορισμός pin CE στο Arduino
  Mirf.init();
  Mirf.setRADDR((byte *)"serv1"); //ορισμός διεύθυνσης Receive
  Mirf.payload = 32;
  Mirf.config();
  Serial.println("Beginning ... "); //Μήνυμα αρχής επικοινωνίας
}

void loop(){
  int data[32]; // Μεταβλητή αρχείου αποστολής
  if(!Mirf.isSending() && Mirf.dataReady()){ // Όταν λάβει δεδομένα
    Serial.println("Got packet"); // Μήνυμα επιτυχούς αποστολής
    Mirf.getData((byte *) &data);
    Serial.write(byte(data[0])); //h
    Serial.write(byte(data[1])); //e
    Serial.write(byte(data[2])); //l
    Serial.write(byte(data[3])); //l
    Serial.write(byte(data[4])); //o
    Serial.println("");
  }
}
```

Όπως φαίνεται και στα σχόλια του κώδικα, τόσο του Server_test όσο και του Receiver_test, μετά τη δήλωση των βιβλιοθηκών, γίνεται ο ορισμός των διευθύνσεων

επικοινωνίας. Στη συνέχεια, αποστέλλεται κάθε δευτερόλεπτο ένα πακέτο από τον Server στον Receiver, ενώ ο Receiver επιβεβαιώνει την επιτυχή αποστολή του πακέτου με το μήνυμα “Got packet”.

Κάνοντας compile το πρόγραμμα και ανοίγοντας το serial monitor με το Arduino Mega συνδεδεμένο στον υπολογιστή μέσω USB, εμφανίζεται το παράθυρο της εικόνας 43. Παρατηρείται ότι μετά από κάθε δευτερόλεπτο λαμβάνεται το πακέτο, το οποίο είναι η λέξη “Hello”.

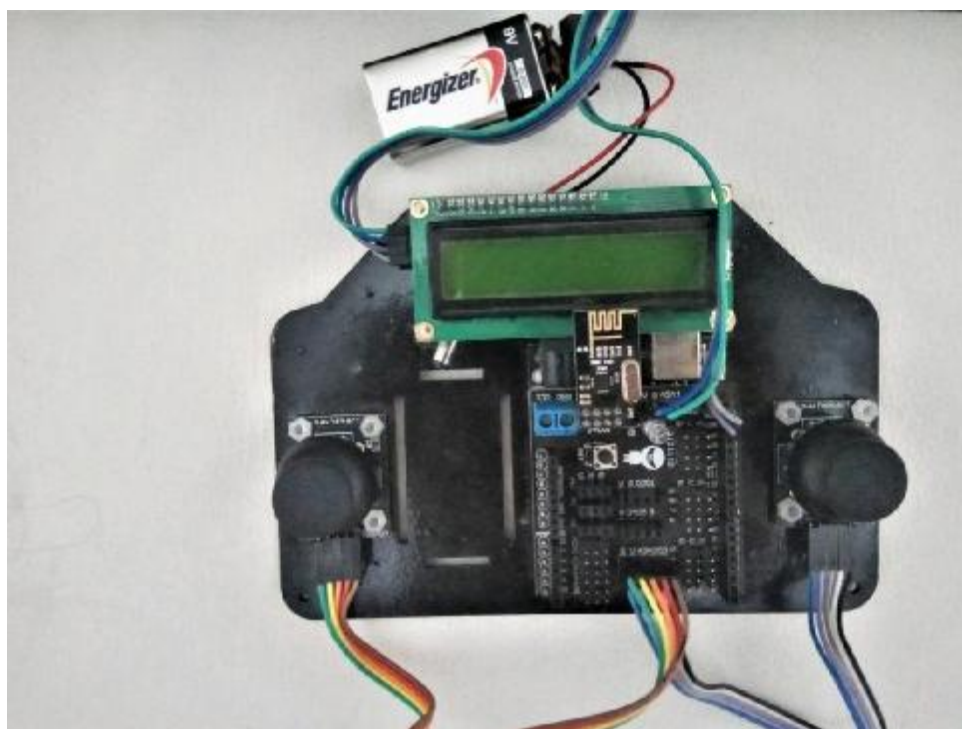


Εικόνα 43: Serial Monitor Receiver πειράματος επικοινωνίας (Ιδία σχεδίαση, 2017).

4.3.5 Ασύρματος έλεγχος ρομποτικού βραχίονα

Εφαρμόζοντας την ασύρματη επικοινωνία μεταξύ δύο μικροελεγκτών Arduino για την ανταλλαγή δεδομένων, καθώς και των έλεγχου σερβοκινητήρων μέσω του μικροελεγκτή που αναφέρθηκαν στα προηγούμενα κεφάλαια, κατέστη δυνατή η υλοποίηση του ασύρματου ελέγχου του ρομποτικού βραχίονα.

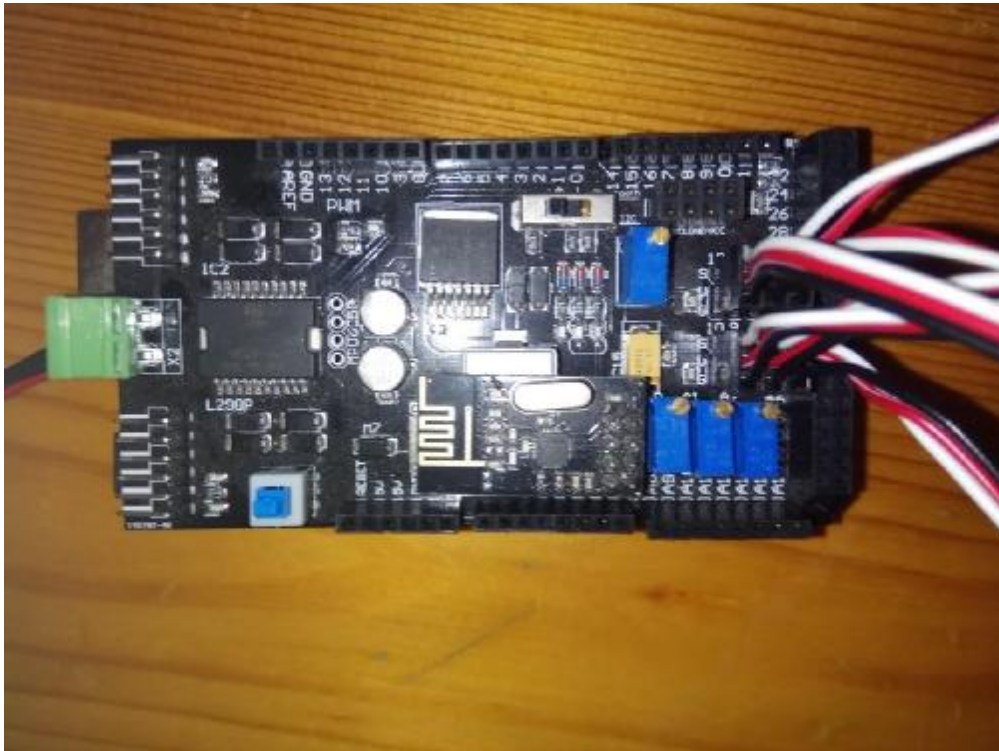
Ο προγραμματισμός του ασύρματου ελέγχου γίνεται σε 2 μέρη. Το πρώτο μέρος του κώδικα βρίσκεται στο Arduino Uno, το οποίο αποστέλλει δεδομένα από τα αναλογικά χειριστήρια και λαμβάνει δεδομένα για τη θέση του ρομποτικού βραχίονα (Εικόνα 44). Το δεύτερο μέρος βρίσκεται στο Arduino Mega, το οποίο λαμβάνει δεδομένα από το ασύρματο χειριστήριο και αποστέλλει δεδομένα της θέσης του κάθε σερβοκινητήρα (Εικόνα 45). Τονίζεται ότι μετά τη συναρμολόγηση του βραχίονα και του χειριστηρίου, έγινε η σύνδεση των σερβοκινητήρων στο shield Sainsmart InstaBots Robot Controller Shield όπως παρατίθεται στον πίνακα 6.



Εικόνα 44: Τελική μορφή ασύρματου χειριστηρίου (Πηγή: Ιδία λήψη, 2017)

Πίνακας 6: Θέσης σύνδεσης σερβοκινητήρων

Σερβοκινητήρας	Θέση στο shield
Σερβοκινητήρας βάσης περιστροφής	Θέση 8
Σερβοκινητήρας βάσης κίνηση πάνω-κάτω	Θέση 11
Σερβοκινητήρας "αγκώνα" κίνηση πάνω-κάτω	Θέση 12
Σερβοκινητήρας "καρπού" κίνηση πάνω-κάτω	Θέση 9
Σερβοκινητήρας "καρπού" περιστροφική κίνηση	Θέση 10
Σερβοκινητήρας Clamp	Θέση 13



Εικόνα 45: Σύνδεση σερβοκινητήρων στο Shield (Ιδία λήψη, 2017).

Κατά τη συγγραφή του κώδικα του χειριστήριου, προκειμένου να καταστεί δυνατός ο ικανοποιητικός έλεγχος του ρομποτικού βραχίονα, λήφθηκαν υπόψη οι παρακάτω παράμετροι:

- ☐ Πραγματοποίηση σύνδεσης και επικοινωνίας με το μικροελεγκτή του βραχίονα, ενημερώνοντας μέσω της αλλαγής της οθόνης από το αρχικό μήνυμα στα δεδομένα που δέχεται από τους σερβοκινητήρες,
- ☐ Εναλλαγή των σερβοκινητήρων που ελέγχονται μέσω του button του αναλογικού χειριστήριου, και
- ☐ Ενημέρωση σε πραγματικό χρόνο για τις θέσεις και των 6 σερβοκινητήρων με σκοπό την ενημέρωση του χρήστη (Μετάφραση και μεταφορά δεδομένων στην οθόνη).

Κώδικας χειριστήριου (Server) :

```
#include "SPI.h" //Δήλωση βιβλιοθηκών
#include "Mirf.h"
#include "nRF24L01.h"
#include "MirfHardwareSpiDriver.h"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
#define ADC_SAMPLES 1
```



```

int Button_Counter; //Μετρητής button και delay πατήματος
int Button_Delay;
struct Axis { //Μεταβλητές θέσης αναλογικών χειριστηρίων
  uint16_t axis_1;
  uint16_t axis_2;
  uint16_t axis_3;
  uint16_t axis_4;
  uint16_t axis_5;
  uint16_t axis_6;
  uint16_t axis_7;
  uint16_t axis_8;
};
Axis axis_x;

struct Gesture {
  uint16_t pwn_1;
  uint16_t pwn_2;
  uint16_t pwn_3;
  uint16_t pwn_4;
  uint16_t pwn_5;
  uint16_t pwn_6;
};
Gesture datain;

void setup() {
  Serial.begin(115200);
  Wire.begin();

  Mirf.spi = &MirfHardwareSpi; // Εκκίνηση επικοινωνίας
  Mirf.init();
  Mirf.setRADDR((byte *)"contr"); // Ορισμός διευθύνσεων Receive και
  Mirf.setTADDR((byte *)"armed"); // Transmit
  Mirf.payload = 16; // Μέγεθος byte δεδομένων αποστολής
  Mirf.config();

  lcd.init(); // Αρχικοποίηση LCD οθόνης
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print(" ROBOT_ARM ");
  lcd.setCursor(0, 1);
  lcd.print("KOURTINOS_PANOS");
}

void loop() {
  uint16_t i;
  uint32_t sum[4];
  digitalWrite(2, HIGH); //Πύθμιση button

  while (digitalRead(2) == LOW) {
    Button_Delay++;
    delay(1);
  }
  if (Button_Delay > 10) {
    Button_Counter ++;
  }
  Button_Delay = 0;
}

```

```

sum[0] = 0;
sum[1] = 0;
sum[2] = 0;
sum[3] = 0;
for (i = 0; i < ADC_SAMPLES; i++) { //Ανάγνωση αναλογικών χειριστηρίων
    sum[0] += analogRead(A0);
    sum[1] += analogRead(A1);
    sum[2] += analogRead(A2);
    sum[3] += analogRead(A3);
}
if (Button_Counter % 2 != 0) { // Εάν το button_counter είναι ζυγός ή
    axis_x.axis_1 = sum[0] / ADC_SAMPLES; // μονός αριθμός αλλάζει ο
    axis_x.axis_2 = sum[1] / ADC_SAMPLES; // έλεγχος στους άλλους σερβοκινητήρες.
    axis_x.axis_4 = sum[2] / ADC_SAMPLES;
    axis_x.axis_5 = sum[3] / ADC_SAMPLES;
    axis_x.axis_3 = axis_x.axis_3;
    axis_x.axis_6 = axis_x.axis_6;
}
else {
    axis_x.axis_1 = sum[0] / ADC_SAMPLES;
    axis_x.axis_2 = sum[1] / ADC_SAMPLES;
    axis_x.axis_3 = sum[2] / ADC_SAMPLES;
    axis_x.axis_6 = sum[3] / ADC_SAMPLES;
    axis_x.axis_5 = axis_x.axis_5;
    axis_x.axis_4 = axis_x.axis_4;
}
while (digitalRead(2) == 0)
{
    delay(1);
}
Serial.println(Button_Counter); // Μετρητής στη σειριακή οθόνη για
Mirf.send((byte *)&axis_x); //έλεγχο λειτουργίας
while (Mirf.isSending()) {
}

if (!Mirf.isSending() && Mirf.dataReady()) {

    Mirf.getData((byte *) &datain);
    lcd.setCursor(0, 0);

    if (Button_Counter % 2 != 0) { //Εμφάνιση στην οθόνη των 4
        //lcd.clear(); // σερβοκινητήρων
        lcd.print("s1: ");
        printServoMeas(datain.pwn_1);
        lcd.print(" s2: ");
        printServoMeas(datain.pwn_2);
        lcd.setCursor(0, 1);
        lcd.print("s3: ");
        printServoMeas(datain.pwn_4);
        lcd.print(" s4: ");
        printServoMeas(datain.pwn_5);
    }
    Else // Εμφάνιση των άλλων 2 σερβοκινητήρων
    {
        //lcd.clear();
        lcd.print("sA: ");
        printServoMeas(datain.pwn_6);
        lcd.print(" sB: ");
        printServoMeas(datain.pwn_3);
        lcd.setCursor(0, 1);
    }
}

```

```

    lcd.print(" ");
  }
}
}

// Συνάρτηση για στοιχισμένη εμφάνιση των μετρήσεων των σερβοκινητήρων
// στην LCD οθόνη
inline void printServoMeas(uint16_t meas)
{
  if (meas < 100 && meas >= 10) //μορφοποίηση της οθόνης ανάλογα με την
  { // τιμή των δεδομένων
    lcd.print(" ");
    lcd.print(meas);
    lcd.print(" ");
  }
  else if (meas < 10)
  {
    lcd.print(" ");
    lcd.print(meas);
    lcd.print(" ");
  }
  else
  {
    lcd.print(meas);
    lcd.print(" ");
  }
}
}

```



Εικόνα 46: Εκκίνηση χειριστηρίου (Ιδία λήψη, 2017).



Εικόνα 47: Εμφάνιση δεδομένων 4 σερβοκινητήρων και έλεγχος τους (Ιδία λήψη, 2017).



Εικόνα 48: Εμφάνιση δεδομένων 2 σερβοκινητήρων και έλεγχος τους (Ιδία λήψη, 2017).

Όπως φαίνεται από τις εικόνες 46 έως 48, το χειριστήριο έχει 3 καταστάσεις λειτουργίας. Στην εικόνα 46 απεικονίζεται η πρώτη κατάσταση κατά την οποία εμφανίζεται ένα αρχικό μήνυμα. Κατά τη διάρκεια του μηνύματος ο μικροελεγκτής προσπαθεί να επικοινωνήσει με το ρομποτικό βραχίονα. Αν η επικοινωνία τους είναι επιτυχής, ο μικροελεγκτής του χειριστηρίου μεταβαίνει στη δεύτερη κατάσταση. Κατά τη δεύτερη κατάσταση εμφανίζονται στην οθόνη οι θέσεις των 4 από τους 6 σερβοκινητήρες, οι οποίοι ελέγχονται από τα 2 αναλογικά χειριστήρια (Εικόνα 47). Ταυτόχρονα με τον έλεγχο, λαμβάνει χώρα η διαδικασία ανατροφοδότησης της θέσης των σερβοκινητήρων και εμφάνισής της στην οθόνη. Ομοίως με τη δεύτερη κατάσταση του χειριστηρίου, με το πάτημα του δεξιού αναλογικού μοχλού ο μικροελεγκτής μεταβαίνει στην τρίτη κατάσταση, κατά την οποία ελέγχει τους 2 τελευταίους σερβοκινητήρες sA και sB, οι οποίοι με τη σειρά τους ελέγχουν το gripper του ρομποτικού βραχίονα, ακολουθώντας παρόμοια μέθοδο ανατροφοδότησης της θέσης και ενημέρωσης στην οθόνη (Εικόνα 48).

Ομοίως με τον κώδικα του χειριστηρίου, κατά τη συγγραφή του κώδικα του ρομποτικού βραχίονα λήφθηκαν υπόψη οι εξής παράγοντες:

- ☐ Ταυτόχρονη αποστολή και λήψη δεδομένων από και προς το χειριστήριο σε πραγματικό χρόνο,
- ☐ Επιτυχής μετάφραση των αποδεχόμενων δεδομένων από θέση των αναλογικών μοχλών και των κατευθύνσεων σε κίνηση των σερβοκινητήρων (κάθε μοχλός αντιστοιχεί σε 2 σερβοκινητήρες),
- ☐ Μετά το πάτημα του αναλογικού μοχλού, μεταφορά του ελέγχου στους υπόλοιπους 2 σερβοκινητήρες του βραχίονα, και
- ☐ Η ταχύτητα των κινήσεων να είναι όσο το δυνατόν πιο ομαλή προσομοιώνοντας την κίνηση ανθρώπινου χεριού με ομαλές μεταβάσεις.

Κώδικας ρομποτικού βραχίονα (client):

```
#include <SPI.h> // Δήλωση βιβλιοθηκών
#include <Mirf.h>
#include <nRF24L01.h>
#include <MirfHardwareSpiDriver.h>
#include <Servo.h>

#define SERVO_WAIST 8 // 11 στο shield
#define SERVO_ELBOW 9 // 12 στο shield
#define SERVO_CLAMP 10 // 13 στο shield
#define SERVO_ROTATE 11 // 8 στο shield
#define SERVO_WRIST 12 // 9 στο shield
#define SERVO_WRIST_ROTATE 13 // 10 στο shield

//Δήλωση instances σερβοκινητήρων
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;
Servo servo5;
Servo servo6;

float k_1;
float k_2;
float k_3;
float k_4;
```

```

float k_5;
float k_6;
float k_1_Temp, k_2_Temp, k_3_Temp, k_4_Temp, k_5_Temp, k_6_Temp;

float speed_1; //Μεταβλητές ταχύτητας σερβοκινητήρων
float speed_2;
float speed_3;
float speed_4;
float speed_5;
float speed_6;

// Τύπος Δεδομένων για αποθήκευση μετρήσεων από το χειριστήριο (θέση αναλογικών)
struct Axis
{
  uint16_t axis_1;
  uint16_t axis_2;
  uint16_t axis_3;
  uint16_t axis_4;
  uint16_t axis_5;
  uint16_t axis_6;
  uint16_t axis_7;
  uint16_t axis_8;
};
Axis axis_x;

// Τύπος Δεδομένων που θα αποστέλλονται στο χειριστήριο
// κάθε μεταβλητή της δομής κρατάει την τρέχουσα θέση του σερβοκινητήρα
struct Gest
{
  uint16_t pwm1;
  uint16_t pwm2;
  uint16_t pwm3;
  uint16_t pwm4;
  uint16_t pwm5;
  uint16_t pwm6;
};

Gest dataout;

// Συνάρτηση αρχικοποίησης θέσης κάθε σερβοκινητήρα
inline void myservosetup()
{
  int sea, seb, sec, sed, see, sef, myspeed, pos;

  // Διάβασμα τωρινής θέσης κινητήρων
  sea = servo1.read();
  seb = servo2.read();
  sec = servo3.read();
  sed = servo4.read();
  see = servo5.read();
  sef = servo6.read();

  myspeed = 500; //ορισμός μεταβλητής ταχύτητας κίνησης

  // Τμηματική μετάβαση στην αρχική θέση
  for (pos = 0; pos <= myspeed; pos += 1)
  {
    servo1.write(int(map(pos, 1, myspeed, sea, 66)));
    servo2.write(int(map(pos, 1, myspeed, seb, 90)));
    servo3.write(int(map(pos, 1, myspeed, sec, 50)));
  }
}

```

```

servo4.write(int(map(pos, 1, myspeed, sed, 90)));
servo5.write(int(map(pos, 1, myspeed, see, 120)));
servo6.write(int(map(pos, 1, myspeed, sef, 90)));
delay(1);
}
}

void setup() {
  Serial.begin(115200);

  // Συσχέτιση των instances των σερβοκινητήρων με τα servo κανάλια
  servo1.attach(SERVO_WAIST);
  servo2.attach(SERVO_ELBOW);
  servo3.attach(SERVO_CLAMP);
  servo4.attach(SERVO_ROTATE);
  servo5.attach(SERVO_WRIST);
  servo6.attach(SERVO_WRIST_ROTATE);

  delay(1000);

  myservosetup();

  delay(2000);

  // Αρχικοποιήσεις και setup του transceiver για λήψη και αποστολή δεδομένων
  Mirf.cePin = 53;
  Mirf.csnPin = 48;
  Mirf.spi = &MirfHardwareSpi;
  Mirf.init();

  // Η διεύθυνση του transceiver του βραχίονα είναι η λέξη armed
  // Το χειριστήριο στέλνει σε αυτή τη διεύθυνση δεδομένα αλλά και λαμβάνει από αυτή τη διεύθυνση
  πακέτα με πληροφορία για την τρέχουσα θέση των servo
  Mirf.setRADDR((byte *)"armed");

  // Μέγιστο μέγεθος πακέτου είναι τα 16 bytes
  Mirf.payload = 16;
  Mirf.config();
  delay(1000);
}

#define SPEED_DIVIDER 6000.00
void loop() {
  // Αν ο transceiver δεν είναι σε κατάσταση αποστολής δεδομένων
  // Έλεγχος λήψης δεδομένων από το χειριστήριο
  // και τα αποθήκευση στη μεταβλητή axis_x
  if (!Mirf.isSending() && Mirf.dataReady()) {
    Mirf.getData((byte *) &axis_x);

  // Αποστολή στη διεύθυνση contr (χειριστήριο) των προηγούμενων θέσεων των servos
  Mirf.setTADDR((byte *)"contr");
  Mirf.send((byte *)&dataout);

  // Αναμονή μέχρι τα δεδομένα να σταλούν επιτυχώς
  while (Mirf.isSending());
}

```

```

// Μετάφραση των κινήσεων των μοχλών σε θέση servo
if (axis_x.axis_1 > 530)
{
    k_1_Temp = map(axis_x.axis_1, 531, 1023, 0, 500) / SPEED_DIVIDER;
}
else if (axis_x.axis_1 < 470)
{
    k_1_Temp = map(axis_x.axis_1, 0, 469, -500, 0) / SPEED_DIVIDER;
}
else {
    k_1_Temp = 0;
}

if (axis_x.axis_2 > 530)
{
    k_2_Temp = -map(axis_x.axis_2, 531, 1023, 0, 500) / SPEED_DIVIDER;
}
else if (axis_x.axis_2 < 470)
{
    k_2_Temp = -map(axis_x.axis_2, 0, 469, -500, 0) / SPEED_DIVIDER;
}
else {
    k_2_Temp = 0;
}

if (axis_x.axis_3 > 530)
{
    k_3_Temp = -map(axis_x.axis_3, 531, 1023, 0, 500) / SPEED_DIVIDER;
}
else if (axis_x.axis_3 < 470)
{
    k_3_Temp = -map(axis_x.axis_3, 0, 469, -500, 0) / SPEED_DIVIDER;
}
else {
    k_3_Temp = 0;
}

if (axis_x.axis_4 > 530) {
    k_4_Temp = -map(axis_x.axis_4, 531, 1023, 0, 500) / SPEED_DIVIDER;
}
else if (axis_x.axis_4 < 470) {
    k_4_Temp = -map(axis_x.axis_4, 0, 469, -500, 0) / SPEED_DIVIDER;
}
else {
    k_4_Temp = 0;
}

if (axis_x.axis_5 > 530) {
    k_5_Temp = -map(axis_x.axis_5, 531, 1023, 0, 500) / SPEED_DIVIDER;
}
else if (axis_x.axis_5 < 470) {
    k_5_Temp = -map(axis_x.axis_5, 0, 469, -500, 0) / SPEED_DIVIDER;
}
else {
    k_5_Temp = 0;
}

if (axis_x.axis_6 > 530) {
    k_6_Temp = -map(axis_x.axis_6, 531, 1023, 0, 500) / SPEED_DIVIDER;
}
}

```



```

else if (axis_x.axis_6 < 470) {
  k_6_Temp = -map(axis_x.axis_6, 0, 469, -500, 0) / SPEED_DIVIDER;
}
else {
  k_6_Temp = 0;
}

// Αλλαγή της θέσης των servo βάσει της προηγούμενης θέσης και της
// τρέχουσας ληφθείσας θέσης
k_1 += 0.02 * (k_1_Temp - k_1);
k_2 += 0.02 * (k_2_Temp - k_2);
k_3 += 0.02 * (k_3_Temp - k_3);
k_4 += 0.02 * (k_4_Temp - k_4);
k_5 += 0.02 * (k_5_Temp - k_5);
k_6 += 0.02 * (k_6_Temp - k_6);

// Υπολογισμός μετάβασης στη νέα θέση
speed_1 = min(90, max(-90, speed_1 += k_1));
speed_2 = min(90, max(-90, speed_2 += k_2));
speed_3 = min(90, max(-90, speed_3 += k_3));
speed_4 = min(90, max(-90, speed_4 += k_4));
speed_5 = min(90, max(-90, speed_5 += k_5));
speed_6 = min(90, max(-90, speed_6 += k_6));

delay(2);

// Μετάφραση της δεκαδικής τιμής σε τιμή με εύρος από 0 έως 180
// Η βιβλιοθήκη των servo χρησιμοποιεί τιμές με τέτοιο εύρος
dataout.pwm1 = speed_1 + 90;
dataout.pwm2 = speed_2 + 90;
dataout.pwm3 = map(speed_3 + 90, 0, 180, 18, 51);
dataout.pwm4 = speed_4 + 90;
dataout.pwm5 = speed_5 + 90;
dataout.pwm6 = speed_6 + 90;

// Συγγραφή των νέων υπολογισμένων τιμών στα servo motors
servo1.write(dataout.pwm1);
servo2.write(dataout.pwm2);
servo3.write(dataout.pwm3);
servo4.write(dataout.pwm4);
servo5.write(dataout.pwm5);
servo6.write(dataout.pwm6);
}

```

Σύμφωνα με τα σχόλια του κώδικα, αρχικά, γίνεται ο ορισμός των μεταβλητών, η αντιστοίχιση των σερβοκινητήρων στο μικροελεγκτή και η εγκαθίδρυση επικοινωνίας μεταξύ σερβοκινητήρα-χειριστηρίου. Μετά την εφαρμογή μιας αρχικής θέσης εκκίνησης του σερβοκινητήρα ακολουθεί το μέρος μετάφρασης της κίνησης των αναλογικών χειριστηρίων σε κίνηση σερβοκινητήρων. Κατά τη διάρκεια της εν λόγω διαδικασίας, γίνεται συσχετισμός της υπάρχουσας θέσης και της θέσης στην οποία είναι επιθυμητό να κινηθεί ο βραχίονας μέσω των μεταβλητών k_1 έως k_6 , k_1_temp έως k_6_temp και $speed_1$ έως $speed_6$. Στη συνέχεια, μέσω της βιβλιοθήκης `servo_h` που χρησιμοποιήθηκε, οι συγκεκριμένες τιμές μετατρέπονται από δεκαδικές σε τιμές με εύρος 0-180 μοίρες. Αξίζει να σημειωθεί ότι η μεταβλητή `SPEED_DIVIDER` διαιρεί την κίνηση των αναλογικών με μία σταθερά που ορίστηκε μετά από πειραματισμό με σκοπό την ομαλότερη κίνηση του βραχίονα. Τέλος,

ομοίως με το μικροελεγκτή του χειριστηρίου, γίνεται αμφίδρομη ανταλλαγή δεδομένων, σε πραγματικό χρόνο, των θέσεων των σερβοκινητήρων με σκοπό τόσο τον έλεγχο όσο και την απεικόνιση των θέσεων στο χρήστη μέσω της οθόνης.

5 Συμπεράσματα

Η κατασκευή και ο προγραμματισμός ενός ρομποτικού βραχίονα είναι ένα θέμα που προσφέρει τόσο τεχνικές όσο και προγραμματιστικές προκλήσεις. Αξίζει να σημειωθεί ότι υπήρξαν αρκετά προβλήματα κατά τη διάρκεια της κατασκευής τόσο στο κατασκευαστικό όσο και στο προγραμματιστικό κομμάτι της εργασίας, τα οποία αντιμετωπίστηκαν με πειραματισμό και διερεύνηση.

Η ανάπτυξη μιας τέτοιου είδους εφαρμογής, τόσο προγραμματιστικά όσο και στο κομμάτι της υλοποίησής του, εκτός από την εκπαιδευτική της αξία, μπορεί να εκμεταλλευτεί και εμπορικά για τον ασύρματο έλεγχο πληθώρας μηχανημάτων. Τα εν λόγω μηχανήματα λειτουργούν σε περιβάλλον όπου είναι αδύνατο να επέμβει ο άνθρωπος, προσφέροντας ακόμα μεγαλύτερη ακρίβεια και μειώνοντας τον κίνδυνο για ανθρώπινο σφάλμα.

Κατά τη διάρκεια της εργασίας και της πειραματικής διαδικασίας επετεύχθη η λειτουργία ενός ρομποτικού βραχίονα, καθώς και η ασύρματη επικοινωνία των μικροελεγκτών. Είναι προφανές πως υπάρχουν περιθώρια βελτίωσης και επέκτασης της παρούσας εργασίας όσον αφορά στον έλεγχο, την ακρίβεια και πληθώρα μεγεθών τα οποία δε λήφθηκαν υπόψη λόγω της πολύπλοκης φύσης της ρομποτικής και κινηματικής μηχανικής. Ωστόσο, η παρούσα αποτελεί ένα καλό παράδειγμα αρχής λειτουργίας ασύρματου ελέγχου, μικροελεγκτών και ρομποτικής με σκοπό την περαιτέρω ενασχόληση με τους εν λόγω τομείς.

Τέλος, αξίζει να σημειωθεί ότι εκτός από τον ασύρματο έλεγχο μέσω χειριστηρίου, η παρούσα εφαρμογή μπορεί να αποτελέσει και κομμάτι προγραμματισμού κινήσεως και να λειτουργήσει ανεξάρτητα από κάποιον χρήστη. Αυτό μπορεί να γίνει αξιοποιώντας τη μνήμη του Arduino Mega που χρησιμοποιήθηκε.

Βιβλιογραφία

- Craig, J., 2008, Εισαγωγή στη ρομποτική, Μηχανική και έλεγχος. Μπφ. Κωφίδης Νικόλας. Θεσσαλονίκη: Εκδόσεις Τζιόλα.
- Κυριακόπουλος, Κ., Παπαδόπουλος, Ε., χ.χρ, "Εισαγωγικές Σημειώσεις Ρομποτικής: Συστήματα Ευφυούς Ελέγχου & Ρομποτική. Αθήνα: Εθνικό Μετσόβιο Πολυτεχνείο. Ανασύρθηκε την 1/11/2016 από την ιστοσελίδα:
http://courseware.mech.ntua.gr/ml23419/robotics_pdf/intro.pdf
- Μπαρμπουλέτου, Β., 2013, "Διάταξη ελαφρών τριβο-πολυμερών αρθρώσεων και έλεγχος αυτών με τεχνητούς πνευματικούς μύες και PLC". Πάτρα: Διπλωματική Εργασία, Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, Πανεπιστήμιο Πατρών.
- Παπάζογλου, Π., Λιωνής Π-Σ., 2015, Ανάπτυξη εφαρμογών με το *Arduino*. Θεσσαλονίκη: Εκδόσεις Τζιόλα.
- Monk, S., 2012. *Programming Arduino: Getting Started with Sketches*. USA: McGraw-Hill Inc.
- Nocks, L., 2008. *The Robot: The Life Story of a Technology*. USA: The Johns Hopkins University Press.
- Pennestri, E., Cavacece, M., Vita, L. 2005. *On the computation of degrees-of-freedom: A didactic perspective*. ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference Long Beach, California, USA, September 24-28,2005. pp. 1733-1741.
- Serway, R., 1990, *Physics for scientists & engineers* Τόμος 1 μηχανική. Ιδιωτική έκδοση.

Διαδικτυακή

AliExpress (2017a)

<https://www.aliexpress.com/item/1x-15Kg-feedback-servo-Metal-gear-Android-Robot-Servo-Digital-servo-for-Robot-diy-excellent-servo/32240916087.html> (Ανασύρθηκε στις 24/8/17)

AliExpress (2017b)

<https://www.aliexpress.com/item/Experimental-9V-Battery-Snap-Power-Cable-to-DC-clip-Male-Line-Adapter-for-Arduino-UNO-R3/32473633663.html> (Ανασύρθηκε στις 24/8/17)

Atmel (2014)

http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf (Ανασύρθηκε την 28/7/2016)

Arduino (2017a)

<https://www.arduino.cc/en/Hacking/PinMapping2560> (Ανασύρθηκε την 12/09/2017)

Arduino (2017b)

<https://www.arduino.cc/en/Main/arduinoBoardMega2560/> (Ανασύρθηκε την 12/09/2017)

Arduino (2017c)

<https://store.arduino.cc/arduino-uno-rev3> (Ανασύρθηκε την 12/09/2017)

Arduino (2017d)
<https://www.Arduino.cc/en/Tutorial/Blink> (Ανασύρθηκε την 12/09/2017)

Arduino (2017e)
<https://www.Arduino.cc/en/Guide/Windows> (Ανασύρθηκε στις 27/7/17)

Arduino (2017f)
<https://www.Arduino.cc/en/Main/Libraries> (Ανασύρθηκε στις 27/7/17)

Arduino (2017g)
<https://www.arduino.cc/en/Reference/Servo> (Ανασύρθηκε στις 27/7/17)

Arduino (2017h)
<https://www.arduino.cc/en/Reference/SPI> (Ανασύρθηκε στις 27/7/17)

Arduino (2017i)
<https://www.arduino.cc/en/Tutorial/Sweep> (Ανασύρθηκε στις 27/7/17)

Codebender Inc. (2017)
https://codebender.cc/library/LiquidCrystal_I2C (Ανασύρθηκε στις 28/7/17)

Encyclopedia Britannica, Inc. (2017).
<https://www.britannica.com/biography/Karel-Capek> (Ανασύρθηκε την 12/09/2017)

Github (2017a)
<https://github.com/maniacbug/RF24> (Ανασύρθηκε στις 28/7/17)

Github (2017b)
<https://github.com/aaronds/arduino-nrf24l01/tree/master/Mirf> (Ανασύρθηκε στις 28/7/17)

Microelectronics (2016)
<http://www.microelectronics.com/datasheets/ArduinoMega2560.pdf> (Ανασύρθηκε την 27/7/16)

Oxford University Press (2017)
<http://www.oxforddictionaries.com/definition/english/robotics> (Ανασύρθηκε την 12/09/2017)

Sainsmart (2017a)
<https://www.sainsmart.com/products/mega-2560-r3-atmega2560-16au-arduino-compatible> (Ανασύρθηκε την 12/09/2017).

Sainsmart (2017b)
<https://www.sainsmart.com/products/6-axis-mechanical-desktop-robotic-arm> (Ανασύρθηκε την 12/09/2017).

Sainsmart (2017c)
<https://www.Sainsmart.com/Sainsmart-rotatable-platform-for-robotic-arm.html> (Ανασύρθηκε στις 24/8/17)

Sainsmart (2017d)
<https://www.Sainsmart.com/Sainsmart-instabots-robot-controller-shield-for-Arduino-mega2560-r3-robot-arm-control.html> (Ανασύρθηκε στις 24/8/17)

Sainsmart (2017e)

<https://www.Sainsmart.com/Arduino-compatibles-1/wireless-3g-module/nrf24l01-wireless-transceiver-module-2-4ghz-ism-band.html> (Ανασύρθηκε στις 24/8/17)

Sainsmart (2017f)

http://wiki.sainsmart.com/index.php/SainSmart_DIY_6-Axis_Servo_Control_Robot_Arm_Combo_kit (Ανασύρθηκε στις 24/8/17)

Stack Exchange Inc. (2017)

<https://electronics.stackexchange.com/questions/104004/relationship-between-clock-speed-and-i2c-bus-speed> (Ανασύρθηκε την 1/5/2017)

Torq Pro & Tower Pro (2014)

<http://www.towerpro.com.tw/product/mg995-robot-servo-180-rotation/> (Ανασύρθηκε στις 24/8/17)

Wikimedia (2015)

https://commons.wikimedia.org/wiki/File:Brosen_shipsmovemensonthewave.svg
(Ανασύρθηκε την 12/09/2017)

Wikimedia (2017)

https://commons.wikimedia.org/wiki/File:6DOF_en.jpg (Ανασύρθηκε την 12/09/2017)

Wikipedia (2017)

https://en.wikipedia.org/wiki/Analog_stick (Ανασύρθηκε την 1/5/2017)