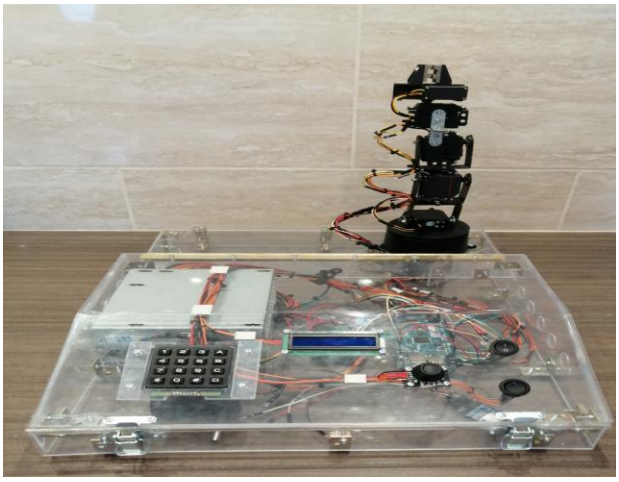
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ

ΘΕΜΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

"ΕΛΕΓΧΟΣ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ ΜΕ ΧΡΗΣΗ JOYSTICK, ΑΝΤΙΓΡΑΦΗ ΚΑΙ ΑΥΤΟΜΑΤΗ ΑΝΑΠΑΡΑΓΩΓΗ ΤΗΣ ΚΙΝΗΣΗΣ"



ΟΝΟΜΑΤΑ ΦΟΙΤΗΤΩΝ:

ΠΑΠΑΔΗΜΗΤΡΟΠΟΥΛΟΣ ΑΘΑΝΑΣΙΟΣ Α.Μ. 44564

ΠΕΝΝΙΝΞ ΑΛΕΞΑΝΔΡΟΣ Α.Μ. 44565

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

Δρ.ΜΙΧΑΗΛ ΠΑΠΟΥΤΣΙΔΑΚΗΣ

ΑΙΓΑΛΕΩ, ΝΟΕΜΒΡΙΟΣ 2019

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο / Η κάτωθι υπογεγραμμένος / η ....ΠΕΝΝΙΝΕ ΑΛΕΞΑΝΔΡΟΣ....,

του .....ΖΑΝ-ΚΑΡΛΟ....., με αριθμό μητρώου ...44565..... φοιτητής / τρια του Τμήματος **Μηχανικών Βιομηχανκής Σχεδίασης και Παραγωγής,** του **Πανεπιστημίου Δυτικής Αττικής** πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασης της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

Ο Δηλών

Ημερομηνία

01/11/2019

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο / Η κάτωθι υπογεγραμμένος / η ...ΠΑΠΑΔΗΜΗΤΡΟΠΟΥΛΟΣ ΑΘΑΝΑΣΙΟΣ

του ....ΑΝΔΡΕΑ....., με αριθμό μητρώου ...44564..... φοιτητής / τρια του Τμήματος **Μηχανικών Βιομηχανκής Σχεδίασης και Παραγωγής,** του **Πανεπιστημίου Δυτικής Αττικής** πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασης της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

Ο Δηλών

Ημερομηνία

01/11/2019

ΠΕΡΙΛΗΨΗ

Στην παρούσα πτυχιακή εργασία παρουσιάζουμε την έρευνα και την υλοποίηση της κατασκευής και του προγραμματισμού ενός ρομποτικού βραχίονα έξι βαθμών ελευθερίας. Τα σενάρια , τα οποία θα υλοποιεί το ρομποτικό σύστημα του οποίου η κίνηση ελέγχεται μέσω ενός χειριστηρίου Joystick θα είναι, αφενός η χειροκίνητη λειτουργία (Manual) και αφετέρου η αντιγραφή της κίνησης και η επανεκτέλεσή της (Record and Playback).

Πιο συγκεκριμένα, θα εμβαθύνουμε στον τρόπο λειτουργίας του ρομποτικού βραχίονα, στα επιμέρους υποστηρικτικά μέλη που το απαρτίζουν, καθώς και τα πρωτόκολλα επικοινωνίας που χρησιμοποιήθηκαν από τον μικροελεγκτή στο κομμάτι του προγραμματισμού για τον έλεγχο αυτών.

ABSTRACT

In this thesis we present the research and the implementation of the construction and programming of a robotic arm of six degrees of freedom. The scenarios implemented by this robotic system, which is controlled via a Joystick controller, on the one hand will be the creation of a Manual operation, and from the other hand, the Copy and the Playback of the motion which was captured.

More specifically, we will delve deeper into how the robotic arm works, the individual support members that make it up, and the communication protocols used by the microcontroller in the programming part to control them.

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα πτυχιακή εργασία εκπονήθηκε στο πλαίσιο των σπουδών μας, στο τμήμα Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής του Πανεπιστημίου Δυτικής Αττικής (πρώην τμήμα Μηχανικών Αυτοματισμού Α.Ε.Ι. Πειραία Τ.Τ.). Αποτέλεσε ένα ιδιαιτέρως σημαντικό μέσο για την προσωπική μας εμβάθυνση στον κλάδο της Ρομποτικής και στα πολλαπλά αντικείμενα που παρέχει στους σπουδαστές και στους μετέπειτα επαγγελματίες ο προγραμματισμός, καθώς και οι τομείς της μηχανικής και της ηλεκτρολογικής – ηλεκτρονικής. Η προσπάθεια ήταν μεγάλη, μολαταύτα η εξέλιξή μας τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο ήταν σημαντική. Αυτή η προσπάθεια δεν θα είχε υλοποιηθεί χωρίς τη συμβολή και την καθοδήγηση των καθηγητών μας. Ιδιαίτερες ευχαριστίες οφείλουμε στον Αναπληρωτή Καθηγητή κ. Μ. Παπουτσιδάκη, συντονιστή της πτυχιακής εργασίας μας, καθώς και τον κ. Μ. Χατζόπουλο για την εποπτεία της μελέτης μας και τη στήριξή μας στην ερευνητική μας δραστηριότητα. Δεν θα μπορούσαμε, επίσης, να παραγνωρίσουμε τη συμβολή της οικογένειάς μας κατά τη συνολική διάρκεια των σπουδών μας, οι οποίοι συνέβαλαν και υλικά και ηθικά. Ευχαριστούμε τους φίλους μας και συμφοιτητές μας για τους γόνιμους διαλόγους που πραγματοποιήσαμε και τη συνολική τους στήριξη, όχι μόνο κατά τη διάρκεια της εκπόνησης της πτυχιακής εργασίας, αλλά και στα συνολικά έτη των σπουδών μας. Ιδιαίτερες ευχαριστίες οφείλουμε στον φίλο μας Γιώργο Χαριτάτο, για τη συμβολή του στη φιλολογική επιμέλεια του κειμένου. Ευχόμαστε η εργασία μας να αποτελέσει εφαλτήριο για περαιτέρω έρευνα και εξέλιξη στον χώρο μας, είτε ακαδημαϊκά είτε στους ευρύτερους εργασιακούς τομείς.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1

ΕΡΜΗΝΕΙΑ ΔΟΜΗ ΚΑΙ ΕΦΑΡΜΟΓΕΣ ΤΗΣ ΡΟΜΠΟΤΙΚΗΣ

Εισαγωγή……..7

1.1 Τομέας μελέτης της Ρομποτικής…….....……7

1.2 Δομή του Ρομποτικού βραχίονα……8

1.2.1 Εφαρμογές της ρομποτικής……10

ΚΕΦΑΛΑΙΟ 2

Ο ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΗΣ ARDUINO ΚΑΙ ΟΙ ΒΟΗΘΗΤΙΚΕΣ ΠΛΑΚΕΤΕΣ

Εισαγωγή……12

2.1 Δομή και χαρακτηριστικά του Μικροεπεξεργαστή……12

2.2 Πλακέτα επέκτασης εξωτερικής μνήμης SD card……...14

2.2.1 SPI Πρωτόκολλο επικοινωνίας με τον μικροελεγκτή……..……15

2.3 Οθόνη απεικόνισης χαρακτήρων……..20

2.3.1 I2C Πρωτόκολλο επικοινωνίας με τον μικροελεγκτή……..21

2.4 Πληκτρολόγιο διαστάσεων 4Χ4……..23

2.5 Χειριστήριο Joystick……25

2.6 Σερβοκινητήρες……28

ΚΕΦΑΛΑΙΟ 3

ΜΕΘΟΔΟΛΟΓΙΑ ΕΡΕΥΝΑΣ ΥΛΟΠΟΙΗΣΗΣ ΚΑΙ ΠΡΟΓΡΑΜΑΤΙΣΜΟΥ ΤΟΥ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ

Εισαγωγή……....32

3.1 Συνδεσμολογία και κατασκευή του ρομποτικού βραχίονα…….32

3.2 Προγραμματισμός και επεξήγηση του κώδικα……...40

3.3 Μελλοντική εξέλιξη……61

3.4 Βιβλιογραφία και ιστοτοπογραφία…….…….62

ΚΕΦΑΛΑΙΟ 1

ΕΡΜΗΝΕΙΑ ΔΟΜΗ ΚΑΙ ΕΦΑΡΜΟΓΕΣ ΤΗΣ ΡΟΜΠΟΤΙΚΗΣ

Εισαγωγή

Αυτό το κεφάλαιο έχει σκοπό την εισαγωγή στον τεχνολογικό κλάδο που ασχολείται με την παραγωγή μηχανών, που θα διευκολύνουν την καθημερινότητα του ανθρώπου. Πιο συγκεκριμένα, θα γίνει αναφορά των χαρακτηριστικών, της δομής και των τομέων εφαρμογής ενός ρομποτικού βραχίονα.

1.1) **Τομέας μελέτης της Ρομποτικής**

Ο κλάδος της ρομποτικής μελετά τις μηχανές εκείνες όπου μπορούν να αντικαταστήσουν τον άνθρωπο στην εκτέλεση μιας εργασίας, συνδυάζοντας τη δραστηριότητα με τη διαδικασία λήψης αποφάσεων. Αυτές οι μηχανές θεωρούνται από τον τεχνολογικό τομέα πως είναι ικανές να προσαρμόζονται στο περιβάλλον το οποίο λειτουργούν, καθώς και να αλληλεπιδρούν με αυτό λαμβάνοντας δεδομένα μέσα από αισθητήρια, με τα οποία είναι εξοπλισμένο.

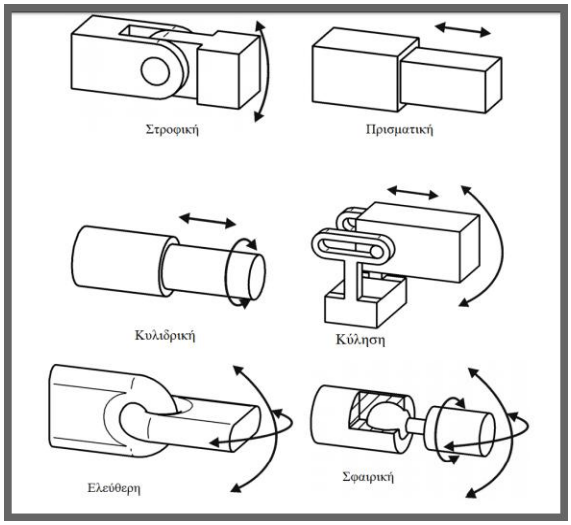
Μια ρομποτική μηχανή είναι μια σύνθεση αντίληψης και δράσης. Το κομμάτι της αντίληψης επιτυγχάνεται με διάφορες μονάδες αισθητηρίων όπου συλλέγουν δεδομένα για τον χώρο εργασίας στο περιβάλλον που βρίσκονται, καθώς και δεδομένα του ίδιου του μηχανισμού τους για τον έλεγχο της εύρυθμης λειτουργίας του. Στην άλλη περίπτωση, το στοιχείο δράσης τους οφείλεται στη μηχανική δομή τους όπου δίνει τη δυνατότητα στο ρομποτικό σύστημα να μετακινείται στο περιβάλλον εργασίας του, καθώς και τη μετακίνησή του για την υλοποίηση της εργασίας που έχει προγραμματιστεί για να εκτελέσει.

Τη σύζευξη των αισθητηρίων μονάδων και του στοιχείου δράσης του ρομπότ την αναλαμβάνει το σύστημα ελέγχου, στο οποίο συμπεριλαμβάνονται όλοι οι περιορισμοί της

μηχανής, του χώρου εργασίας του ρομπότ, του ανθρώπινου παράγοντα, καθώς και τα πρότυπα ασφαλείας και λειτουργίας.[1]

1.2) **Δομή του Ρομποτικού βραχίονα**

Η πιο διαδεδομένη μηχανή Ρομπότ είναι ένας ρομποτικός βραχίονας ο οποίος, όπως προαναφέρθηκε σε προηγούμενο κεφάλαιο, αποτελείται από τη μηχανική δομή, όπως και το σύστημα έλεγχου του. Τα δομικά του κυρίως μέρη μπορεί να είναι μηχανικά (γρανάζια, δοκοί, ροδές κ.τ.λ.), ηλεκτρονικά, αισθητήρες καθώς και κινητήρες. Η δομή του σώματός του για τη συγκράτηση και την ενεργοποίηση της κίνησής του, αποτελείται από αρθρώσεις και συνδέσμους. Οι αρθρώσεις χωρίζονται σε περιστροφικές και πρισματικές, όπου στις μεν προκαλείται περιστροφή ανάμεσα σε δυο συνδέσμους και στις πρισματικές εξασφαλίζεται η ευθύγραμμη κίνηση κατά μήκος του άξονα της άρθρωσης. Στην εικόνα 1.2 παρουσιάζονται οι εν λόγω μορφές. Αυτές οι δυο αρθρώσεις, όταν συνδυαστούν, επιφέρουν το κέρδος της ευελιξίας και της αυτονομίας της κίνησης και του ελέγχου.



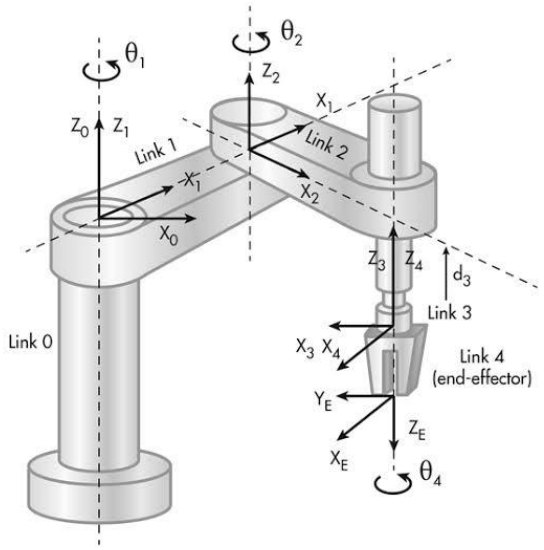
Εικόνα 1.2) Τύποι αρθρώσεων[2]

Η γεωμετρία του ρομποτικού βραχίονα χωρίζεται σε τρεις κατηγορίες· την καρτεσιανή, την κυλινδρική – σφαιρική και την ανθρωπόμορφη γεωμετρία τύπου SCARA, που

[1] Ζωή Δουλγέρη, *Ρομποτική· κινηματική, δυναμική και έλεγχος αρθρώσεων βραχίονα*, Αθήνα, Κριτική, 2007, σελ. 13 – 25.
[2] Βλ. https://ilektroaytomatismoi.blogspot.com/2018/07/blog-post_4.html [Τελευταία ανάκτηση: 3 Αυγούστου 2019].

παρουσιάζεται στην εικόνα 1.2.1. Ο ρομποτικός βραχίονας χαρακτηρίζεται από τους βαθμούς ελευθερίας του, που ορίζονται από το πλήθος των αρθρώσεων που το συγκροτούν.

Οι αρθρώσεις του ρομποτικού βραχίονα περιλαμβάνουν ενεργοποιητές, όπως ηλεκτρικούς, υδραυλικούς ή πνευματικούς κινητήρες. Ταυτόχρονα, ανάλογα με την εφαρμογή που χρειάζεται να υλοποιήσει μια τέτοια μηχανή προσαρμόζεται και διαφορετικό εργαλείο ή αρπάγη στο σημείο δράσης του. Το σημείο αυτό που συγκρατεί το εργαλείο δράσης ονομάζεται τελικό στοιχείο δράσης του ρομποτικού βραχίονα.



Εικόνα 1.2.1) Ρομποτικός βραχίονας τύπου SCARA

Επιπρόσθετα, στη δομή του βρίσκονται τα ηλεκτρονικά ισχύος που ενεργοποιούν την τροφοδοσία των κινητήρων, των αισθητήρων μέσω των οποίων επικοινωνεί ο ελεγκτής, καθώς και τα κυκλώματα ενίσχυσης του σήματος.

Τέλος, στο κομμάτι του ελεγκτή βρίσκεται το πρόγραμμα που έχει υλοποιηθεί για την εκάστοτε εφαρμογή, το οποίο είναι προγραμματισμένο σε κάποια γλώσσα υψηλού επιπέδου, το λογισμικό ελέγχου της θέσεως και της ταχύτητας των κινητήριων μέσων και το κινηματικό λογισμικό, το οποίο γνωρίζοντας τη θέση των αρθρώσεων καθορίζει τη θέση του άκρου. Το τελευταίο λειτουργεί εντελώς αντίθετα με το αντίστροφο κινηματικό, στο οποίο γνωρίζοντας τη

θέση του άκρου προσαρμόζονται τα σημεία των αρθρώσεων. Επίσης, περιλαμβάνει και βοηθητικό λογισμικό, που είναι υπεύθυνο για την εύρυθμη λειτουργία του βραχίονα εντοπίζοντας και ενημερώνοντας τον χρήστη για λάθη.[3]

1.2.1) **Εφαρμογές της ρομποτικής**

Η χρησιμότητα των ρομποτικών μηχανών συνέβαλε ενεργά στην αυτοματοποίηση της παραγωγικής διαδικασίας στις βιομηχανίες, διότι ένα τέτοιο σύστημα αυξάνει την παραγωγικότητα, συμβάλλει στη μεγάλη βελτίωση της ποιότητας, έχει μεγαλύτερη αντοχή και διάρκεια στον χρόνο, και συνάμα μπορεί να λειτουργεί με μεγάλη επαναληψιμότητα, πράγμα το οποίο είναι δύσκολο για τον άνθρωπο λόγω φυσικής υπόστασης. Ωστόσο, ένα από τα αρνητικά ενός τέτοιου συστήματος είναι η απουσία ανεπτυγμένης νοημοσύνης σε εργασίες που εκτελούνται σε άγνωστο για αυτό περιβάλλον.

Ο κλάδος της ρομποτικής βρίσκει εφαρμογή σε βιομηχανίες και βιοτεχνίες, όπως βιομηχανία αυτοκινήτων (παράδειγμα στην εικόνα 1.2.3), βιομηχανίες τσιμέντου, διαδικασία συναρμολόγησης ηλεκτρονικών, βιομηχανίες φαρμάκων, βιοτεχνίες μετάλλων, βιοτεχνίες ενδυμάτων κ.λπ.

Συνοπτικά, κάποιες από τις εφαρμογές ενός ρομποτικού συστήματος είναι η μεταφορά υλικών, η συναρμολόγηση, η συγκόλληση μεταλλικών κατασκευών, η συγκόλληση ηλεκτρικών στοιχείων, όπως επίσης βαφές, εκσκαφή, εργασίες σε ραδιενεργό περιβάλλον, καθώς και η χρήση τους στην Ιατρική για χειρουργεία μεγάλης ακρίβειας, όπως φαίνεται και στην εικόνα 1.2.2.[4]

---

[3] Βλ. Ζωή Δουλγέρη, ό. π.
[4] Φώτης Ν. Κουμπουλής – Βασίλης Γ. Μέρτζιος, *Εισαγωγή στη Ρομποτική*, Αθήνα, Παπασωτηρίου, 2002, σελ. 13 – 14.

Εικόνα 1.2.2) Ρομποτικός βραχίονας για λαπαροσκόπικη χειρουργική

Εικόνα 1.2.3) βιομηχανία αυτοκινήτων

ΚΕΦΑΛΑΙΟ 2

Ο ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΗΣ ARDUINO ΚΑΙ ΟΙ ΒΟΗΘΗΤΙΚΕΣ ΠΛΑΚΕΤΕΣ

Εισαγωγή

Σκοπός αυτού του κεφαλαίου είναι η παρουσίαση των χαρακτηριστικών και της δομής των αναπτυξιακών εργαλείων, καθώς και των βοηθητικών πλακετών (Breakout Boards) που χρησιμοποιήθηκαν για τη λειτουργία του ρομποτικού βραχίονα της παρούσας πτυχιακής εργασίας. Πιο συγκεκριμένα, κυρίαρχο ρόλο για τη διασφάλιση του έλεγχου των ενεργοποιητών του ρομποτικού βραχίονα σήμανε η ηλεκτρονική πλακέτα μικροελεγκτή ανοικτού κώδικα. Παράλληλα, οι βοηθητικές πλακέτες συνέβαλαν στην τροφοδοσία των ενεργοποιητών, στην επέκταση της μνήμης του μικροελεγκτή, στον οπτικό έλεγχο της λειτουργίας του προγράμματος μέσω μιας οθόνης γραφικής απεικόνισης χαρακτήρων, καθώς και σε περιφερειακό μέσο, όπως το πληκτρολόγιο και το χειριστήριο Joystick, τα οποία συμβάλλουν στην λειτουργία του προγράμματος. Συνάμα, θα παρουσιάσουμε και τα πρωτόκολλα επικοινωνίας μεταξύ του ελεγκτή και των βοηθητικών πλακετών.

2.1) **Δομή και χαρακτηριστικά του Μικροεπεξεργαστή**

Για τις ανάγκες του ελέγχου του ρομποτικού βραχίονα χρησιμοποιήσαμε το αναπτυξιακό εργαλείο Arduino. Το Arduino είναι μια ηλεκτρονική πλατφόρμα ανοικτού κώδικα βασισμένη στην ιδέα της εύκολης χρήσης υλικού και λογισμικού. Οι πλατφόρμες αυτές έχουν τη δυνατότητα μέσω τον εισόδων και εξόδων που υπάρχουν πάνω σε αυτήν να διαβάζουν αισθητήρες, να αντιληφθούν το πάτημα ενός κουμπιού κ.ά, καθώς και να ενεργοποιήσουν κάποια ενδεικτική λυχνία Led ή κάποιον κινητήρα, να επικοινωνήσουν με κάποια άλλη συσκευή κ.ά.

Μια τέτοια πλατφόρμα συμπεριλαμβάνει έναν μικροελεγκτή, τον ATmega, όπου ποικίλει σε εκδόσεις και δυνατότητες και ο οποίος από την κατασκευή του περιέχει μια έκδοση

Bootloader, όπου δεν χρειάζεται εξωτερικό προγραμματισμό. Επίσης, περιλαμβάνει έναν γραμμικό σταθεροποιητή τάσης στα 5V και στα 3.3V, έναν κρυσταλλικό ταλαντωτή 16MHz, ψηφιακές και αναλογικές εισόδους/εξόδους και ενδεικτικές λυχνίες Led, όπου η μία υποδηλώνει ότι η πλατφόρμα βρίσκεται σε λειτουργία.

Η άλλη μπορεί να χρησιμοποιηθεί για την πρώτη επαφή με τον προγραμματισμό του, ενεργοποιώντας την με βασικό σχεδιάγραμμα χωρίς εξωτερικές συνδέσεις, και άλλες δύο που δείχνουν την επικοινωνία με τον υπολογιστή και αναβοσβήνουν πολύ γρήγορα καθώς ανεβαίνει ένας κώδικας στην πλατφόρμα. Επιπρόσθετα, έχει μια θύρα USB όπου χρησιμοποιείται για τη σύνδεση με τον υπολογιστή, για την αποστολή δεδομένων, για την τροφοδοσία της πλατφόρμας, καθώς και για σειριακή επικοινωνία με την οθόνη του προγράμματος της πλατφόρμας. Βασικό είναι και το στοιχείο ενός κουμπιού, όπου επαναφέρει (Reset) τον μικροελεγκτή στην αρχική κατάσταση χωρίς να διαγράφει το πρόγραμμα που του τοποθετήθηκε από τον χρήστη, σε περίπτωση που υπάρχει.

Η γλώσσα προγραμματισμού της πλατφόρμας αυτής είναι η Processing, η οποία είναι μια γλώσσα προγραμματισμού ανοικτού κώδικα που προέρχεται από το ολοκληρωμένο περιβάλλον ανάπτυξης της πλατφόρμας και η οποία είναι γραμμένη σε Java. Το σχέδιο, στο οποίο ξεκινά να γράφει ο χρήστης τον κώδικά του που συντάσσεται σε γλώσσα C/C++, αποτελείται από έναν επεξεργαστή πηγαίου κώδικα, έναν μεταγλωττιστή, καθώς και εργαλείο εντοπισμού σφαλμάτων. Ταυτόχρονα, περιλαμβάνει και κάποιες βιβλιοθήκες που χρησιμοποιούνται για την παροχή τους σε επιπλέον λειτουργίες που διευκολύνουν τον χρήστη, όπως οι περισσότερες πλατφόρμες προγραμματισμού. Υπάρχουν πολλές διαφορετικές πλατφόρμες όπου ποικίλουν οι δυνατότητές τους ανάλογα με τη ζήτηση και τη χρησιμότητα στις απαιτήσεις της κάθε κατασκευής.

Για τις ανάγκες της συγκεκριμένης διεργασίας του ρομποτικού βραχίονα χρησιμοποιήσαμε την πλατφόρμα ARDUINO MEGA ADK REV3 (Βλ. εικόνα 2.1), η οποία έχει τα εξής χαρακτηριστικά. Βάσει της επίσημης σελίδας της εταιρίας της πλατφόρμας, το ARDUINO MEGA έχει για μικροελεγκτή τον ATmega2560. Αποτελείται από μια διέπαφη υποδοχή για σύνδεση με Android τηλέφωνα, 54 πόρτες, είτε εισόδους είτε εξόδους (I/O), εκ των οποίων οι 15 μπορούν να χρησιμοποιηθούν ως έξοδοι με σήμα εξόδου διαμόρφωσης πλάτους παλμού PWM (Pulse Width Modulation), 16 αναλογικές πόρτες εισόδου και εξόδου (I/O), και έναν κρυσταλλικό ταλαντωτή 16MHz. Επιπρόσθετα, περιλαμβάνει μια θύρα USB, ένα βύσμα

τύπου jack για την τροφοδοσία του, και όπως οι περισσότερες πλακέτες, ένα κουμπί επαναφοράς Reset button.[5]

Εικόνα 2.1) Μικροελεγκτής ARDUINO MEGA ADK REV3

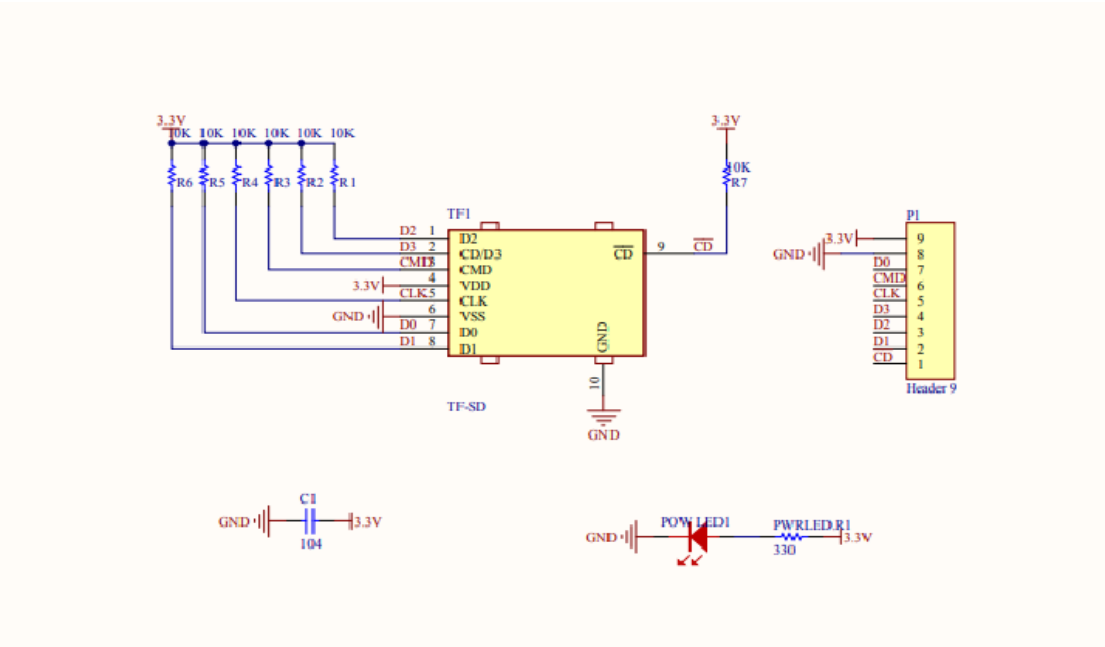
Πιο συγκεκριμένα, η εισαγόμενη τάση που είναι απαραίτητη για την εύρυθμη λειτουργία της συγκεκριμένης πλατφόρμας είναι 7V έως 12V, ενώ τα όρια είναι μεταξύ 6V και 20V. Το συνεχές ρεύμα που παρέχεται στην έξοδο της κάθε εισόδου/εξόδου – πόρτας είναι 40mA, και στην έξοδο των 3.3V είναι 50mA. Στο κομμάτι της μνήμης του διαθέτει 4KB EEPROM, 256KB Flash Memory όπου τα 8KB χρησιμοποιούνται από το Boot loader, 8KB SRAM, και 16MHz ταχύτητα ρολογιού (Clock Speed). Οι διαστάσεις της πλατφόρμας είναι μήκος 105.52mm, πλάτος 53.3 mm, και στο σύνολό της ζυγίζει 36g.[6]

2.2) **Πλακέτα επέκτασης εξωτερικής μνήμης SD card**

Αυτή η πλακέτα επέκτασης του Arduino χρησιμοποιείται για τη μεταφορά δεδομένων από και προς μια κάρτα SD. Η πλακέτα αυτή μας επιτρέπει να διαβάσουμε, καθώς και να καταγράψουμε δεδομένα στο έργο μας. Πάνω στο υλικό της πλακέτας υπάρχει μια θυρίδα όπου

[5] Απαραίτητη για τη μελέτη στάθηκε η ιστοσελίδα https://www.arduino.cc/ [Τελευταία ανάκτηση: 3 Αυγούστου 2019].
[6] Βλ. https://store.arduino.cc/arduino-mega-adk-rev3 [Τελευταία ανάκτηση: 3 Αυγούστου 2019].

χρησιμοποιείται για την τοποθέτηση μιας κάρτας μνήμης, τα χαρακτηριστικά της οποίας επιλέγονται από τον χρήστη ανάλογα με τις ανάγκες της κάθε εφαρμογής. Η κυκλωματική διάταξη της πλακέτας αυτής, όπως φαίνεται στην εικόνα 2.2, στα σημεία εξόδου της είναι άμεσα συμβατή με τον μικροελεγκτή Arduino, χρησιμοποιώντας δύο σημαντικά πρωτόκολλα για την επικοινωνία τους, όπου το ένα είναι το SPI και το άλλο SDIO.[7]



Εικόνα 2.2) Σχέδιο πλακέτας μονάδας μνήμης

2.2.1) **SPI Πρωτόκολλο επικοινωνίας με τον μικροελεγκτή**

Το πρωτόκολλο SPI (Serial Peripheral Interface), είναι ένας δίαυλος σύνδεσης και μεταφοράς δεδομένων μεταξύ ενός μικροελεγκτή και μιας περιφερειακής συσκευής, όπως έναν αισθητήρα, μια κάρτα επέκτασης μνήμης κ.ά. Χρησιμοποιεί ξεχωριστές γραμμές επικοινωνίας με τις περιφερειακές συσκευές για τον συγχρονισμό του παλμού του ρολογιού καθώς και τη

[7] Βλ. https://grobotronics.com/microsd-card-breakout.html [Τελευταία ανάκτηση: 3 Αυγούστου].

γραμμή μεταφοράς δεδομένων, επιλέγοντας και διαφορετική γραμμή για τη συσκευή που επιθυμεί να επικοινωνήσει.

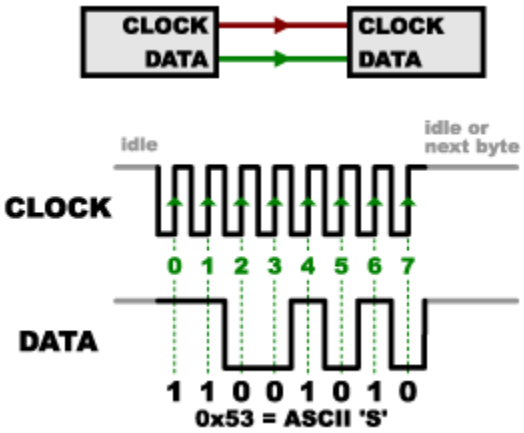
Γενικά, μια κοινή σειριακή πόρτα επικοινωνίας μεταξύ δύο συσκευών ονομάζεται ασύγχρονη, γιατί δεν υπάρχει έλεγχος κατά τη μεταφορά δεδομένων, αν λήφθηκε σωστά η πληροφορία, καθώς και δεν εγγυάται ότι και οι δυο πλευρές τρέχουν ακριβώς με τον ίδιο ρυθμό. Στην περίπτωση των υπολογιστικών συστημάτων (όπως ένας ηλεκτρονικός υπολογιστής), που συνήθως βασίζονται σε συγχρονισμένα συστήματα, το φαινόμενο αυτό θα προκαλούσε πρόβλημα, όταν κατά τη σύνδεση δυο συστημάτων υπάρχει διαφορά στον ρυθμό συγχρονισμού του ρολογιού του καθενός.

Η λύση σε αυτήν την περίπτωση των ασύγχρονων σειριακών συνδέσεων είναι να προσθέτουν bits εκκίνησης και τερματισμού σε κάθε Byte βοηθώντας τον συγχρονισμό του δέκτη στα δεδομένα όταν αυτά φτάνουν. Και οι δύο συσκευές πρέπει από την αρχή να συμφωνήσουν εκ των προτέρων σχετικά με την ταχύτητα μετάδοσης. Ελαφρές διαφορές στο ρυθμό μετάδοσης δεν είναι πρόβλημα, επειδή ο δέκτης επανασυγχρονίζεται στην αρχή κάθε Byte.

Παραδείγματος χάρη, σε πειραματικό επίπεδο παρατηρήθηκε ότι, εάν στον κώδικα του μικροελεγκτή Arduino ενεργοποιηθεί η σειριακή επικοινωνία μεταξύ αυτού και του υπολογιστή με την εντολή Serial.begin(9600) – όπου 9600 είναι ο ρυθμός δειγματοληψίας των bits ανά λεπτό – και στο σειριακό παράθυρο τοποθετηθεί διαφορετική τιμή στην επιλογή του ρυθμού, δεν θα εμφανιστούν τα δεδομένα ή θα είναι αλλοιωμένα. Αυτό οφείλεται στο γεγονός ότι ο δέκτης δειγματίζει τα δυαδικά ψηφία σε πολύ συγκεκριμένες χρονικές στιγμές.

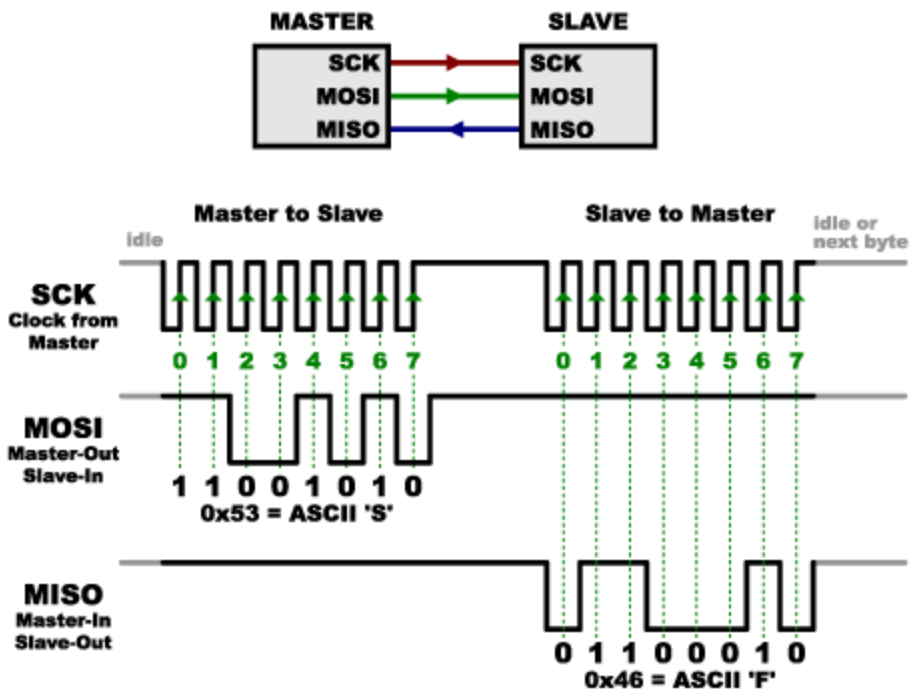
Το πρωτόκολλο SPI έχει διαφορετικό τρόπο λειτουργίας. Είναι ένας συγχρονισμένος δίαυλος μεταφοράς και λήψης δεδομένων χρησιμοποιώντας ξεχωριστές γραμμές για μεταφορά δεδομένων, και συγχρονισμό του παλμού του ρολογιού, όπου διατηρεί και τις δύο συσκευές σε τέλειο συγχρονισμό. Το ρολόι είναι ένα ταλαντευόμενο σήμα, όπου πληροφορεί τον δέκτη ακριβώς πότε πρέπει να γίνεται η δειγματοληψία των δυαδικών ψηφίων στη γραμμή δεδομένων. Αυτό σε λογικές στάθμες αντιστοιχεί είτε από την άνοδο στην κάθοδο είτε το ανάποδο, ανάλογα με τις οδηγίες του κάθε κατασκευαστή. Επομένως, όταν εντοπίσει ο δέκτης μια τέτοια αλλαγή,

θα διαβάσει τη γραμμή των δεδομένων, ώστε να λάβει το επόμενο κομμάτι, όπως φαίνεται και στην εικόνα 2.2.1.



Εικόνα 2.2.1) Το διάβασμα της γραμμής δεδομένων γίνεται από την κάθοδο προς την άνοδο του σήματος του παλμού του ρολογιού

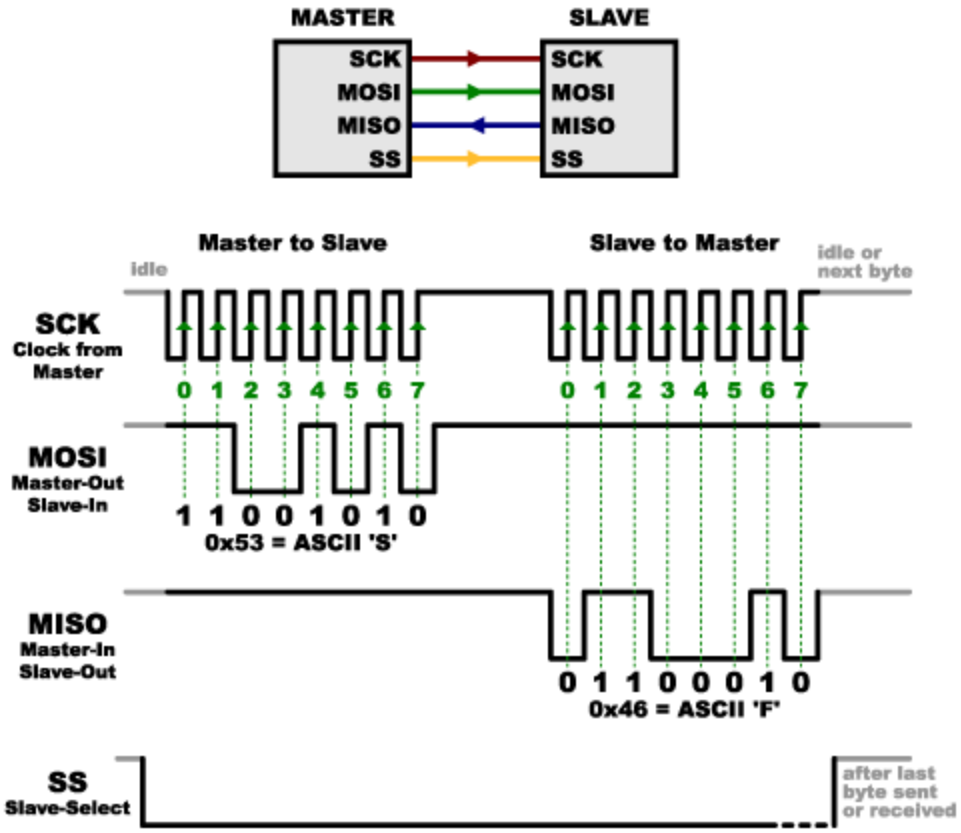
Ο τρόπος λήψης των δεδομένων γίνεται με τον ακόλουθο τρόπο. Στο πρωτόκολλο αυτό η συσκευή που παράγει τον παλμό συγχρονισμού του ρολογιού (όπου η γραμμή αυτή του δίαυλου ονομάζεται CLK ή SCK), είναι μοναδική και ονομάζεται αφέντης (Master). Η περιφερειακή συσκευή που θα επικοινωνήσει μαζί του ονομάζεται υποτελής (Slave). Στον δίαυλο αυτό μπορούν να συνδεθούν πολλές υποτελείς συσκευές, όμως στην αντίθετη περίπτωση υπάρχει μόνο μια συσκευή – Αφέντης. Τα δεδομένα αποστέλλονται από μια συσκευή – Αφέντη σε μια υποτελή συσκευή μέσω μιας γραμμής δεδομένων που ονομάζεται MOSI (Master Out – Slave In), και όταν η υποτελής συσκευή θελήσει να στείλει μια απάντηση πίσω στη συσκευή του αφέντη, την πραγματοποιεί μέσω μιας τρίτης γραμμής δεδομένων, την MISO (Master In – Slave Out), όπως φαίνεται και στην εικόνα 2.2.2.



Εικόνα 2.2.2) Γραμμές αποστολής (MOSI) και λήψης (MISO) δεδομένων του δίαυλου

Επειδή πάντοτε η συσκευή του αφέντη παράγει τον παλμό του ρολογιού, πρέπει να γνωρίζει από την αρχή πότε πρέπει να επιστρέψει δεδομένα και πόσα δεδομένα θα επιστραφούν σε σχέση με την ασύγχρονη σειριακή επικοινωνία, όπου τυχαία ποσά δεδομένων στέλνονται προς οποιαδήποτε κατεύθυνση ανά πάσα στιγμή. Παρόλα αυτά, δεν δημιουργείται σοβαρό πρόβλημα, διότι το SPI πρωτόκολλο χρησιμοποιείται γενικά για να «επικοινωνήσει» με αισθητήρες που έχουν μια συγκεκριμένη δομή. Λόγω του ότι το SPI έχει ξεχωριστές γραμμές αποστολής και λήψης, σε ορισμένες περιπτώσεις μπορεί να λάβει και να μεταδώσει δεδομένα ταυτόχρονα.

Τέλος, στον δίαυλο υπάρχει και ακόμα μια γραμμή που ονομάζεται SS (Slave Select) ή CS (Chip Select). Αυτή η γραμμή είναι υπεύθυνη για την επιλογή της συσκευής με την οποία θα επικοινωνήσει, ειδικά όταν στον δίαυλο υπάρχουν περισσότερες από μια υποτελείς συσκευές. Επίσης, δραστηριοποιεί τη συσκευή του Αφέντη για την αποστολή ή τη λήψη δεδομένων (Βλ.εικόνα 2.2.3).



Εικόνα 2.2.3) Γραμμή επιλογής υποτελούς συσκευής

Αυτή η γραμμή διατηρείται σε κανονικά υψηλή στάθμη (λογικό 1), πράγμα που αποσυνδέει τις υποτελείς συσκευές από τον δίαυλο. Προτού αποσταλούν τα δεδομένα στην υποτελή συσκευή, η γραμμή γίνεται χαμηλή (λογικό 0), η οποία την ενεργοποιεί. Όταν τελειώσει, η γραμμή γίνεται πάλι υψηλή. Σε έναν καταχωρητή μετατόπισης, αυτό αντιστοιχεί στην είσοδο «μανδάλωσης», η οποία μεταφέρει τα ληφθέντα δεδομένα στις γραμμές εξόδου.[8]

[8] Για τις περισσότερες πληροφορίες, καθώς και τις εικόνες του ανωτέρω κεφαλαίου βλ. https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all [Τελευταία ανάκτηση: 3 Αυγούστου 2019].

2.3) **Οθόνη απεικόνισης χαρακτήρων**

Μια οθόνη είναι μια συσκευή, όπου επιτρέπει την προβολή πληροφοριών σε καθορισμένο χρόνο. Υπάρχουν διάφορες διαστάσεις μεγέθους καθώς και δυνατότητες υψηλής ανάλυσης. Στον τεχνικό τομέα και ειδικά στη ρομποτική χρησιμοποιούνται για τον οπτικό έλεγχο της κατάστασης μιας εφαρμογής, καθώς ο τελευταίος μπορεί να γίνει και απομακρυσμένα. Αυτός ο οπτικός έλεγχος παρέχει στον χρήστη το πλεονέκτημα να παρακολουθεί τη διαδικασία ενός έργου και να ειδοποιείται από αυτό το σύστημα για τυχόντα σφάλματα.

Για τη διευκόλυνση και τον έλεγχο της κατάστασης του ρομποτικού βραχίονα χωρίς την άμεση παρέμβαση του Η/Υ χρησιμοποιήσαμε μια βασική οθόνη υγρών κρύσταλλου μεγέθους 16 χαρακτήρων και 2 γραμμών, όπως φαίνεται στην εικόνα 2.3, με μπλε φόντο και άσπρο χρώμα χαρακτήρων της εταιρίας OEM. Η συγκεκριμένη πλακέτα έχει τάση λειτουργιάς τα 5V και ρυθμιζόμενο κοντράστ μέσω ενός ποτενσιομέτρου. Παράλληλα, ο τρόπος διασύνδεσης της οθόνης με τον μικροελεγκτή υποστηρίζεται μέσω του πρωτοκόλλου I2C, που θα αναλυθεί σε επόμενο κεφάλαιο.[9]

Εικόνα 2.3) Οθόνη χαρακτήρων 16X2 - I2C

[9] Οι πληροφορίες ανακτήθηκαν από την κάτωθι ιστοσελίδα. Αντιστοίχως, από εκεί πραγματοποιήθηκε και η αγορά του εν λόγω προϊόντος. Αναλυτικότερα, βλ. https://grobotronics.com/basic-16x2-character-lcd-white-on-blue-5v-i2c-protocol.html#group_94505d95f7dd2f47b-1 [Τελευταία ανάκτηση: 20 Αυγούστου 2019].

2.3.1) **I2C Πρωτόκολλο επικοινωνίας με τον μικροελεγκτή**

Το I2C είναι ένα παγκόσμιο πρότυπο που εφαρμόζεται σε παρά πολλές συσκευές και χρησιμοποιείται για τη μεταφορά δεδομένων μέσω του δίαυλου της δομής του σε διάφορες αρχιτεκτονικές μελέτες έλεγχου, όπως σε προηγμένα υπολογιστικά συστήματα ελέγχου υπολογιστών.

Δυο γραμμές επικοινωνίας χρησιμοποιούνται στη δομή του πρότυπου αυτού. Η μια είναι η γραμμή του ρολογιού του δίαυλου (SCL) και η άλλη η γραμμή μεταφοράς των δεδομένων (SDA) μεταξύ των συσκευών που είναι συνδεδεμένες σε αυτόν. Οι συσκευές μπορεί να είναι οθόνη, πληκτρολόγιο, επεξεργαστής κ.λπ. Κάθε συσκευή που είναι συνδεδεμένη στον δίαυλο έχει και μια μοναδική διεύθυνση, όπου μπορεί να λειτουργήσει είτε σαν εκπομπός είτε σαν λήπτης δεδομένων, ανάλογα με τη λειτουργία της κάθε συσκευής. Παραδείγματος χάρη, μια οθόνη μπορεί να είναι μόνο λήπτης σε σχέση με μια μνήμη όπου μπορεί να είναι και τα δυο.

Το πρότυπο αυτό περιέχει το μοντέλο δυαδικής επικοινωνίας αφέντη/σκλάβου (master/slave), όπου μια συσκευή έχει μονόδρομο έλεγχο πάνω σε μια ή περισσότερες άλλες συσκευές. Ταυτόχρονα, είτε στην εκπομπή είτε στη λήψη, μια συσκευή μπορεί να θεωρηθεί ως αφέντης η ως υποτελής κατά τη μεταφορά δεδομένων. Μια συσκευή με την ιδιότητα του αφέντη είναι αυτή που ξεκινά να μεταφέρει δεδομένα στον δίαυλο και παράγει το σήμα του παλμού του ρολογιού για να επιτραπεί η μεταφορά των δεδομένων. Σε αυτήν την περίπτωση, κάθε συσκευή που έχει διεύθυνση η οποία έχει δοθεί από τον αφέντη θεωρείται υποτελής. Μπορούν να υπάρχουν περισσότερες από δύο συσκευές ως αφέντες.

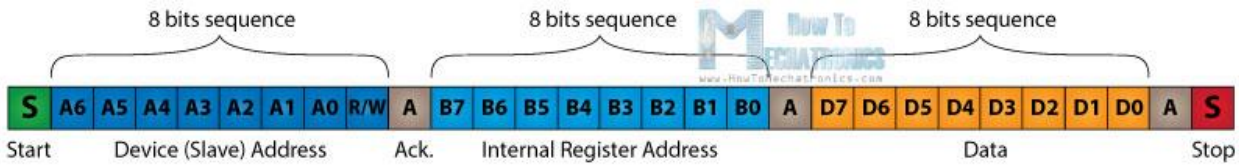
Και τα δύο σήματα, αυτό της μεταφοράς των δεδομένων καθώς και το σήμα του παλμού του ρολογιού, είναι ημιαμφίδρομης κατεύθυνσης γραμμές, οι οποίες είναι συνδεδεμένες με όλες τις συσκευές που είναι στον δίαυλο, καθώς και με αντιστάσεις συνδεδεμένες ως προς την τάση αναφοράς (Pull-up Resistor). Η τιμή των αντιστάσεων σε συνδυασμό με τη χωρητικότητα του δίαυλου καθορίζουν τη μέγιστη ταχύτητά του. Συνήθως, χρησιμοποιούνται αντιστάσεις της τάξεως των 1 έως 10 ΚΩ, οι οποίες δεν μπορούν να απουσιάζουν, γιατί οι δυο γραμμές θα βρίσκονται μόνιμα σε λογικό 0 και θα αποτρέπεται η λειτουργία του δίαυλου.

Όλες οι συναλλαγές ξεκινούν και τελειώνουν με μια ακολουθία έναρξης και λήξης. Καθώς μια συσκευή αφέντη θελήσει να επικοινωνήσει με μια συσκευή σκλάβου στέλνει στο

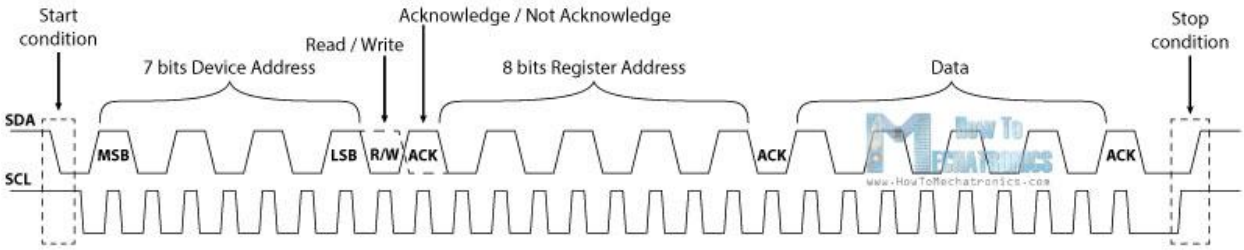
δίαυλο μια ακολουθία έναρξης. Μια αλλαγή από υψηλό σε χαμηλό στη γραμμή των δεδομένων ορίζει μια συνθήκη εκκίνησης. Συνάμα μια αλλαγή από χαμηλό σε υψηλό στη γραμμή των δεδομένων ορίζει την ακολουθία λήξης. Για να επιτευχθεί αυτή η αλλαγή πρέπει η ακολουθία στη γραμμή του παλμού του ρολογιού να βρίσκεται σε υψηλή στάθμη (λογικό 1).

Το σήμα δεδομένων μεταφέρεται σε ακολουθίες των 8 bit (Βλ. εικόνα 2.3.1). Έτσι, μετά από την κατάσταση έναρξης εμφανίζεται η πρώτη ακολουθία των 8 bit όπου υποδεικνύει τη διεύθυνση του υποτελούς. Μετά από κάθε ακολουθία στέλνεται ένα κομμάτι που ενημερώνει τη συσκευή του αφέντη από την δευτερεύουσα συσκευη ότι έλαβε την πρώτη ακολουθία δυαδικών αριθμών. Επομένως, η κύρια συσκευή, αν πάρει την τιμή Acknowledge θα παραδώσει τον έλεγχο της γραμμής των δεδομένων στην εξαρτημένη συσκευή. Εάν δεν παραλάβει η δευτερεύουσα συσκευή την ακολουθία, η κατάσταση είναι μη αναγνωρίσιμη και σημαίνει πως δεν έλαβε επιτυχώς την προηγούμενη ακολουθία. Αυτό μπορεί να οφείλεται σε πολλούς λόγους· για παράδειγμα, η υποτελής συσκευή μπορεί να είναι απασχολημένη ή να μην καταλαβαίνει τα αρχεία ή να μην μπορεί να λάβει περισσότερα δεδομένα κ.ο.κ. Στην περίπτωση αυτή, η κύρια συσκευή αποφασίζει πως θα δράσει. Στην συνεχεία μετά το πρώτο Byte αποστέλλεται μια ακολουθία τον 8 bit που αφορά τους εσωτερικούς καταχωρητές της υποτελούς συσκευής. Κατόπιν έρχεται πάλι κομμάτι ενημέρωσης και, τέλος, μετά από αυτό εμφανίζεται η ακολουθία 8 bit δεδομένων προς αποστολή, τα οποία και τελειώνουν με μια ειδική κατάσταση διακοπής.

Στην πρώτη ακολουθία όπου γίνεται η διευθυνσιοδότηση έχει χωρητικότητα 7bit, ξεκινώντας από το σημαντικότερο ψηφίο του δυαδικού αριθμού και καταλήγοντας στο λιγότερο σημαντικό ψηφίο, καθώς το $8^{ο}$ bit χρησιμοποιείται για την υπόδειξη του αφέντη αν θα διαβάσει (τιμή του $8^{ου}$ bit/ λογικο 1) ή θα γράψει (τιμή του $8^{ου}$ bit/ λογικο 0) στην υποτελή συσκευή. Κατά την έναρξη μιας αποστολής ο δίαυλος θεωρείται απασχολημένος και μετά το πέρας την λήξης της διαδικασίας θεωρείται ελεύθερος (Βλ. εικόνα 2.3.2).



Εικόνα 2.3.1) Πρωτόκολλο επικοινωνίας I2C



Εικόνα 2.3.2) Χαρακτηρισμός των bit του πρωτοκόλλου[10]

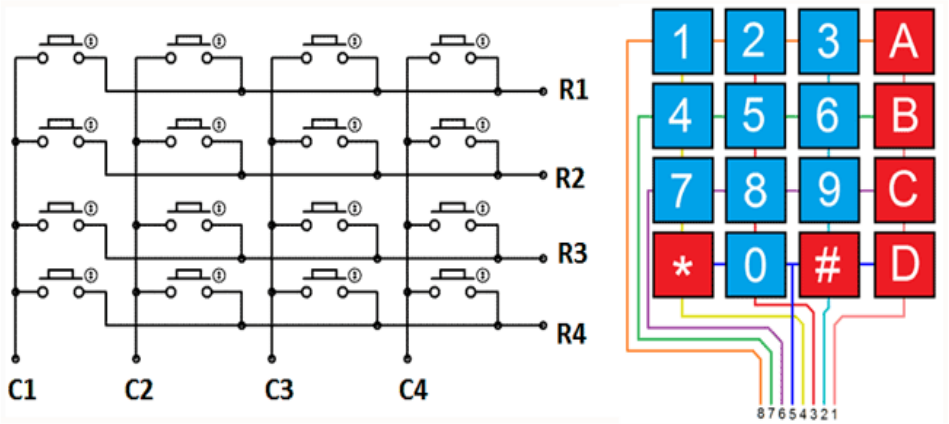
2.4) **Πληκτρολόγιο διαστάσεων 4X4**

Το πληκτρολόγιο είναι μια συσκευή, η οποία εισαγάγει χαρακτήρες σε ένα υπολογιστικό σύστημα όπως ο υπολογιστής ή κάποιος σταθμός εργασίας. Στην παρούσα πτυχιακή εργασία χρησιμοποιήθηκε πληκτρολόγιο μήτρας τεσσάρων γραμμών και τεσσάρων στηλών (Βλέπε εικόνα 2.4), το οποίο χρειάστηκε για την επιλογή κάποιων εντολών σε σημεία του προγράμματος.

Εικόνα 2.4) Πληκτρολόγιο 4X4

[10] Για τη μελέτη της ανάλυσης του πρωτοκόλλου πραγματοποιήθηκε επίσκεψη στην παρακάτω ιστοσελίδα. Βλ. https://howtomechatronics.com/tutorials/arduino/how-i2c-communication-works-and-how-to-use-it-with-arduino/, καθώς και στην επίσημη σελίδα της εταιρίας παραγωγής του, χρησιμοποιώντας το εγχειρίδιο χρήσης του. Βλ. https://www.nxp.com/docs/en/user-guide/UM10204.pdf [Τελευταία ανάκτηση: 1 Σεπτεμβρίου 2019].

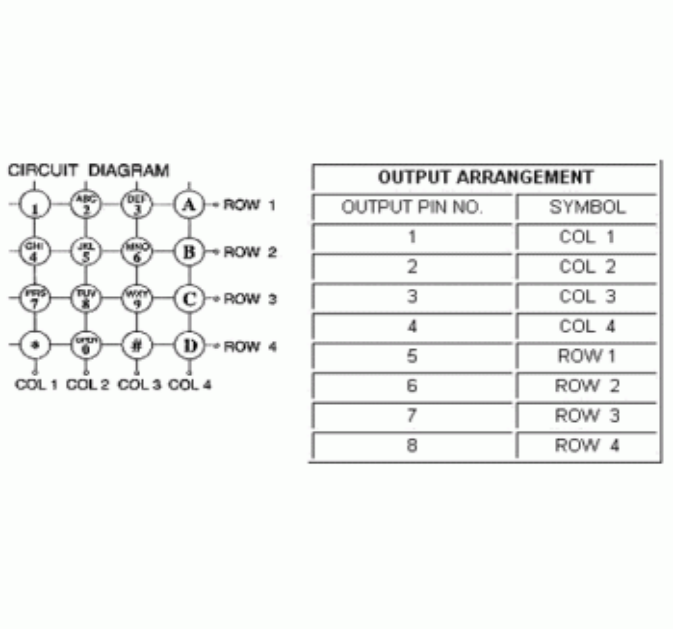
Το πληκτρολόγιο αυτό αποτελείται από ένα σύνολο 16 κουμπιών δηλαδή τέσσερα κουμπιά σε κάθε σειρά. Οι χαρακτήρες που παράγονται είναι οι «1, 2, 3, 4, 5, 6, 7, 8, 9, 0, #, *, A, B, C, D». Οι ακροδέκτες των κουμπιών συνδέονται μεταξύ τους σύμφωνα με το διάγραμμα του κατασκευαστή που παρουσιάζεται στην εικόνα 2.4.1.



Εικόνα 2.4.1) Σχεδιάγραμμα πληκτρολογίου

Όπως φαίνεται στο σχέδιο στην πρώτη σειρά, μια τερματική γραμμή συνδέει τα τέσσερα κουμπιά μεταξύ τους, και σε μια άλλη τερματική γραμμή είναι συνδεδεμένα μεταξύ τους άλλα τέσσερα πλήκτρα της πρώτης στήλης. Το ίδιο συμβαίνει και στις άλλες τρεις γραμμές/στήλες. Αυτός είναι και ο λόγος όπου στην έξοδο του πληκτρολογίου υπάρχουν 8 καλώδια σύνδεσης με τον μικροελεγκτή.[11] Στη συγκεκριμένη περίπτωση, τα πρώτα τέσσερα καλώδια αφορούν τις στήλες και τα τέσσερα επόμενα τις γραμμές, όπως φαίνεται και στο σχήμα της εικόνας 2.4.2.:

[11] Βλ. https://circuitdigest.com/microcontroller-projects/keypad-interfacing-with-8051-microcontroller [Τελευταία ανάκτηση: 1 Σεπτεμβρίου 2019].



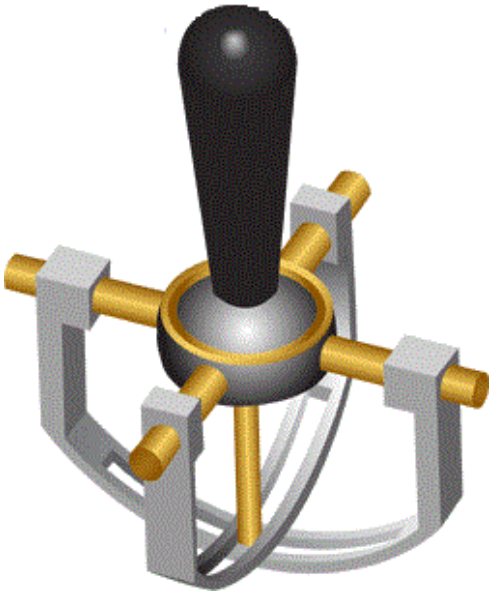
Εικόνα 2.4.2) Σχεδιάγραμμα εξόδων πληκτρολογίου

Τα τεχνικά χαρακτηριστικά του είναι τα εξής: Η μηχανολογική του αντοχή είναι 1000000 κύκλοι, μέγιστη αντίσταση επαφής 200 ΚΩ, μέγιστη δύναμη πατήματος 1N, μέγιστη υποβληθείσα τάση 24V-DC, μέγιστη ισχύς 20mA.[12]

2.5) **Χειριστήριο Joystick**

Η βασική ιδέα ενός μοχλού είναι να μεταφράσει τη θέση του μοχλού (στους άξονες X και Y) σε ηλεκτρονικές πληροφορίες, όπου μπορεί να επεξεργαστεί ένας μικροελεγκτής. Στον X άξονα, η κίνηση του μοχλού είναι δεξιά – αριστερά, και στον Y άξονα η κίνηση είναι πάνω – κάτω. Ο σχεδιασμός του μοχλού αυτού αποτελείται από δυο ποτενσιόμετρα, καθώς και τον μηχανισμό Gimbal. Όταν περιστρέφεται το joystick, η λαβή του αντίχειρα μετακινεί μια στενή ράβδο που κάθεται σε δύο περιστρεφόμενους άξονες με σχισμές (Βλ. εικόνα 2.5, Μηχανισμός Gimbal).

[12] Βλ. https://grobotronics.com/keypad-matrix-4x4.html [Τελευταία ανάκτηση: 1 Σεπτεμβρίου 2019].

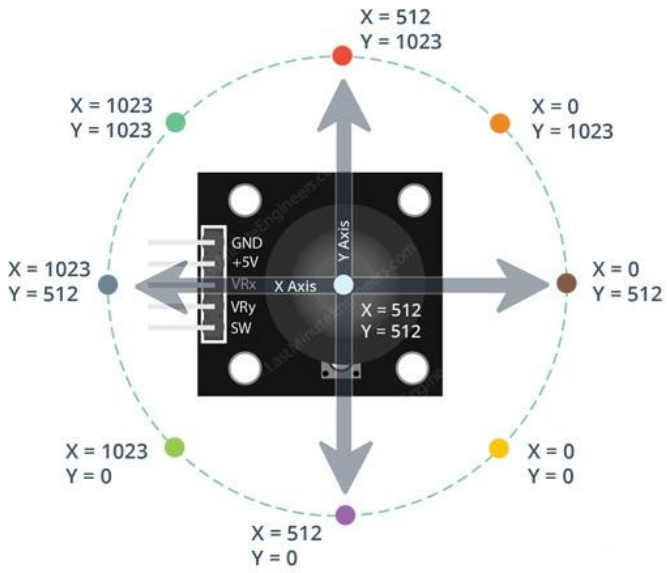


Εικόνα 2.5) Μηχανισμός Gimbal

Τα δυο ποτενσιόμετρα συνδέονται σε κάθε άξονα του χειριστηρίου τα όποια, όταν εφαρμοστεί διαφορά δυναμικού στα άκρα τους, επιστρέφουν μια τιμή τάσης στην έξοδό τους, η οποία είναι ανάλογη των αντιστάσεων που τα αποτελούν. Σε αυτήν την περίπτωση θέλουμε να διαβάσουμε αυτή τη μεταβλητή τιμή, ώστε να βρούμε τη φυσική θέση του μοχλού. Η τελευταία μπορεί να διαβαστεί στην περίπτωση του μικροελεγκτή Arduino στους αναλογικούς ακροδέκτες του με τη χρήση ενός μετατροπέα του επεξεργαστή από αναλογικό σε ψηφιακό σήμα. Η ανάλυση του μετατροπέα του Arduino έχει ανάλυση 10 bit, που σημαίνει ότι οι τιμές που θα εμφανίζει θα κυμαίνονται από 0 έως 1023 σε κάθε άξονα ξεχωριστά. Όταν ο μοχλός βρίσκεται στο κέντρο θεωρητικά πρέπει να έχει την τιμή 512. Πρακτικά, αυτό λόγω σφάλματος, για παράδειγμα αστοχία υλικού ή διαφορετική τιμή τάσης κ.ά. μπορεί να αποκλίνει από την θεωρητική αυτή τιμή.

Στην παρακάτω εικόνα (2.5.1) φαίνονται οι κατευθύνσεις του μοχλού στους άξονες X και Y καθώς και οι θεωρητικές τιμές που παρατηρούνται μετά την ανάλυση του σήματος του Arduino από την έξοδο των ποτενσιόμετρων.[13]

[13] Βλ. https://lastminuteengineers.com/joystick-interfacing-arduino-processing/ [Τελευταία ανάκτηση: 1 Σεπτεμβρίου 2019].



Εικόνα 2.5.1) Τιμές του χειριστηρίου σε διάφορες πιθανές θέσεις του

Στη βάση του χειριστηρίου υπάρχει ένα Μπουτόν, το οποίο σε κατάσταση ηρεμίας παραμένει κανονικά ανοικτή η επαφή του. Ο ακροδέκτης αυτός διαβάζεται από τους ψηφιακούς ακροδέκτες του Arduino. Ο τρόπος λειτουργίας του, σε περίπτωση που θελήσει ο χρήστης να το χρησιμοποιήσει, είναι ότι, κατά την ανάγνωση του μπουτόν, όταν αυτό δεν έχει πατηθεί επιστρέφει λογικό 1/High, ενώ όταν πατηθεί κλείνει η επαφή του, η οποία συνδέεται με τη γείωση και επιστρέφει λογικό 0/Low. Όλο το συγκρότημα του Joystick χειριστηρίου παρουσιάζεται στην παρακάτω εικόνα (2.5.2)[14]:

Εικόνα 2.5.2) Joystick

[14] Βλ. https://grobotronics.com/thumb-joystick-module.html [Τελευταία ανάκτηση: 1 Σεπτεμβρίου 2019].

2.6) **Σερβοκινητήρες**

Κινητήρες είναι ηλεκτρικές μηχανές, οι οποίες μετατρέπουν την ηλεκτρική ενέργεια σε μηχανική. Η λειτουργία των κινητήρων βασίζεται στο φαινόμενο κατά το οποίο, όταν ένας ρευματοφόρος αγωγός βρεθεί μέσα σε μαγνητικό πεδίο, τότε εμφανίζεται μια δύναμη που ασκείται στον αγωγό, με αποτέλεσμα την ενεργοποίηση της κίνησής του. Οι κινητήρες αυτοί αποτελούνται από τον δρομέα και τον στάτη. Ο στάτης παράγει το μαγνητικό πεδίο, το οποίο είναι σταθερό στον χώρο. Ο δρομέας περιέχει την περιέλιξη του τυμπάνου, όπου συνδέεται με τον αγωγό που διαρρέεται από ρεύμα. Η κατεύθυνση της κίνησης του αγωγού εξαρτάται από τη φορά της ροπής του ρεύματος μέσα στην περιέλιξη του τυμπάνου.[15]

«Η πρώτη ηλεκτρική μηχανή συνεχούς ρεύματος κατασκευάστηκε το 1832 από τον Pixii. Αργότερα, το 1866, ο Siemens ανακάλυψε την αρχή λειτουργίας του δυναμό. Πριν από μερικά χρόνια οι ασύγχρονες μηχανές ήταν οι περισσότερο χρησιμοποιημένες μηχανές στον έλεγχο της κίνησης. Με την ανάπτυξη της ηλεκτρονικής και των ηλεκτρονικών ισχύος οι μηχανές συνεχούς ρεύματος κατέλαβαν πάλι τη θέση τους στις βιομηχανικές εφαρμογές. Οι κινητήρες συνεχούς ρεύματος χρησιμοποιούνται σήμερα σε ηλεκτρονικά οχήματα, σε μηχανές ξυρίσματος, σε παιδικά παιχνίδια, σε μηχατρονικές πλατφόρμες, στην ρομποτική, στη βιομηχανία, κλπ.»[16]

Οι κινητήρες χωρίζονται σε δυο μεγάλες κατηγορίες σε σχέση με την τροφοδοσία και την δομή τους, σε αυτούς με συνεχές ρεύμα και σε κινητήρες εναλλασσόμενου ρεύματος, καθώς και τις υποκατηγορίες τους. Στην παρούσα πτυχιακή εργασία θα ασχοληθούμε με κινητήρες συνεχούς ρεύματος και, πιο συγκεκριμένα, με τους σερβοκινητήρες, στους οποίους θα γίνει ανάλυση σε επόμενη παράγραφο.

Ένας κινητήρας σέρβο είναι μια ηλεκτρική μηχανή, η οποία μπορεί να σπρώξει η να περιστρέψει ένα αντικείμενο με μεγάλη ακρίβεια. Παρόλο το μικρό μέγεθος και βάρος τους επιφέρουν υψηλή ροπή. Οι κινητήρες αυτοί αποτελούνται από έναν κινητήρα συνεχούς ρεύματος, οι οποίοι λειτουργούν με ηλεκτρική ενέργεια και οι οποίοι περιστρέφονται σε υψηλές στροφές, αλλά παρόλα αυτά έχουν χαμηλή ροπή στρέψης. Για τον λόγο αυτό, μια διάταξη από

---

[15] David M. Auslander – Carl J. Kempf, *Μηχατρονική. Προσαρμοστικά μηχανικών συστημάτων*, μτφρ. – επιμ. Herbert Tanner, Αθήνα, Πανεπιστημιακές Εκδόσεις Ε.Μ.Π., 1998, σελ. 157 – 158.

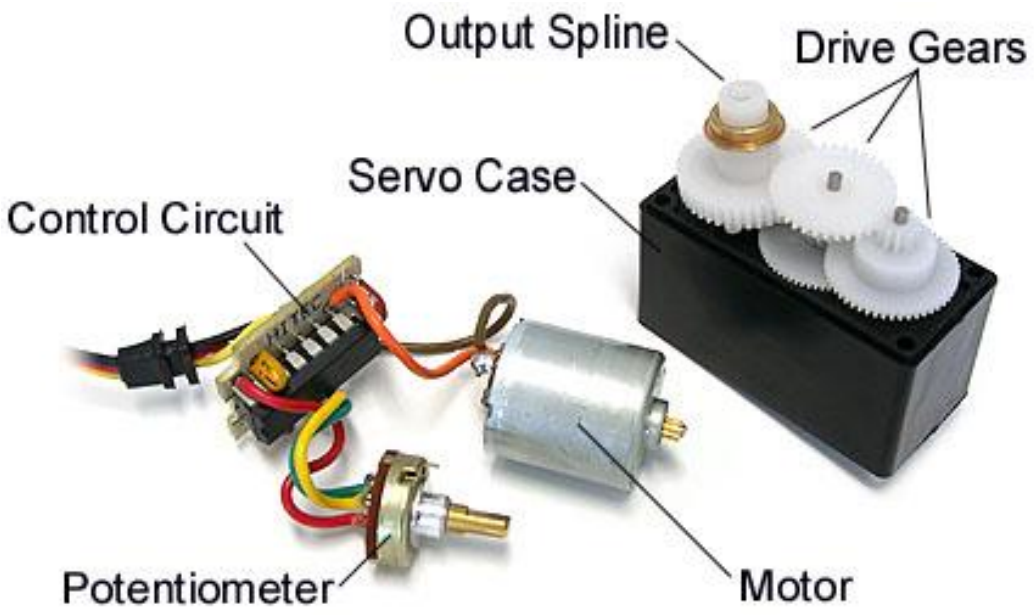
[16] Γ. Πολίτης – Γ. Τσεκούρας, *Ηλεκτρικές Μηχανές· Δομικό Στοιχείο των Συστημάτων Αυτομάτου Ελέγχου*, Αθήνα, Τμήμα Αυτοματισμού, Σ.Τ.Ε. – Τ.Ε.Ι. Πειραιά, 2009, σελ. 105.

γρανάζια που βρίσκεται μέσα στο συγκρότημα του κινητήρα είναι κατανεμημένη, έτσι ώστε να λειτουργεί ως μειωτήρας ταχύτητας αυξάνοντας παράλληλα την ροπή. Τα γρανάζια αυτά είναι κατασκευασμένα από πλαστικό για έναν κινητήρα με μικρότερες δυνατότητες, καθώς και μικρότερου κόστους, και μεταλλικής κατασκευής για κινητήρες υψηλότερων απαιτήσεων συνάμα και υψηλότερου κόστους. Επίσης, αποτελείται από μία συσκευή ανίχνευσης θέσης – συνήθως ένα ποτενσιόμετρο – και ένα κύκλωμα ελέγχου, όπως φαίνεται και στην εικόνα 2.6.

Στην κατηγορία αυτών των κινητήρων υπάρχουν τρεις βασικοί τύποι· ο ένας είναι ο σερβοκινητήρας θέσης περιστροφής. Ο άξονας εξόδου του περιστρέφεται περίπου μισό κύκλο, δηλαδή 180 μοίρες. Διαθέτει φυσικές στάσεις στον μηχανισμό μετάδοσης για να αποτρέψει την υπέρβαση αυτών των ορίων.

Σε επομένη κατηγορία ανήκουν οι συνεχούς περιστροφής σερβοκινητήρες, στους οποίους ο άξονας μπορεί να γυρίζει προς οποιαδήποτε κατεύθυνση για επ' αόριστον. Το σήμα ελέγχου, παρά τη ρύθμιση της στατικής θέσης του σέρβο, δίνεται ως εντολή για τη διευθυνιοδότηση της κατεύθυνσης και της ταχύτητας περιστροφής. Αναλυτικότερα, το εύρος των πιθανών εντολών αναγκάζει το σέρβο να περιστρέφεται δεξιόστροφα ή αντίθετα προς τη φορά των δεικτών του ρολογιού, όπως επιθυμείται, με διαφορετική ταχύτητα, ανάλογα με το σήμα εντολής.

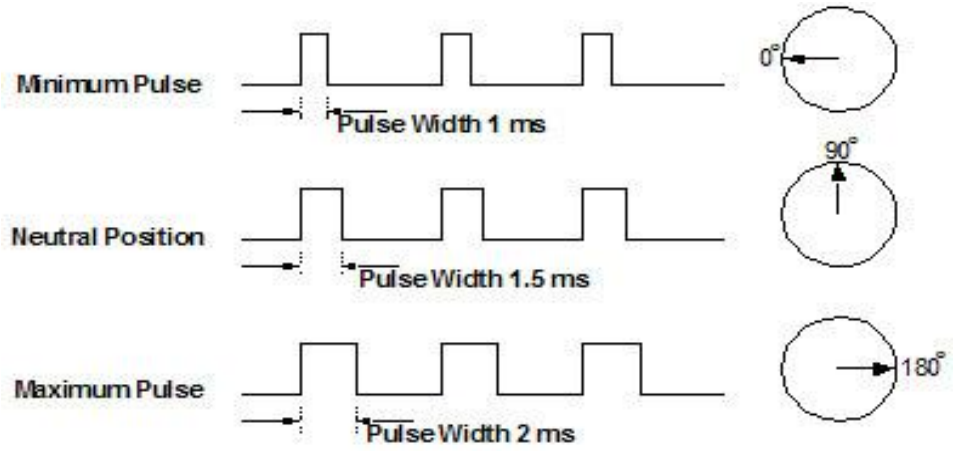
Η τρίτη κατηγορία αφορά το γραμμικό σερβοκινητήρα, ο οποίος είναι όπως αυτός της περιστροφής θέσης, αλλά με πρόσθετες ταχύτητες και μηχανισμό γραναζιών για να αλλάξει την κίνηση από περιστροφική σε εμπρόσθια και οπίσθια.



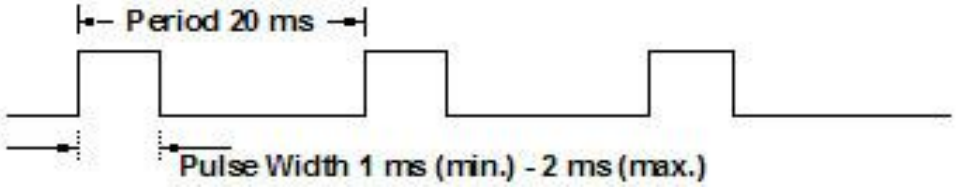
Εικόνα 2.6) Δομή του σερβοκινητήρα[17]

Το σήμα έλεγχου για την περιστροφή του άξονα σε επιθυμητή γωνία καθορίζεται από το σήμα διαμόρφωσης εύρους του παλμού που παράγει ο μικροελεγκτής. Αυτός ο παλμός στέλνεται σε συγκεκριμένες εξόδους του ελεγκτή και εμφανίζεται κάθε 20ms. Το εύρος του παλμού καθορίζει προς το ποια κατεύθυνση, καθώς και το πόσο μακριά θα στρέφεται ο κινητήρας. Ένας παλμός 1.5ms θα κάνει τον κινητήρα να πάει στην θέση των 90 μοιρών, η οποία χαρακτηρίζεται και ως ουδέτερη. Όταν φτάσει παλμός μικρότερος σε εύρος από 1.5ms τότε θα στραφεί αριστερόστροφα από την ουδέτερη θέση και όταν είναι μεγαλύτερος θα στραφεί δεξιόστροφα από αυτήν. Γενικά, το ελάχιστο εύρος του παλμού είναι το 1ms και χαρακτηρίζεται ως η θέση των 0 μοιρών, ενώ το μέγιστο εύρος του παλμού είναι 2ms, όπου βρίσκονται οι 180 μοίρες, όπως παρουσιάζεται και στην εικόνα 2.6.1 και 2.6.2.[18]

[17] Οι ανωτέρω εικόνες και μερικές πληροφορίες αντλήθηκαν από την κάτωθι πηγή. Αναλυτικότερα, βλ. https://www.servocity.com/how-does-a-servo-work [Τελευταία ανάκτηση 16 Σεπτεμβρίου 2019].
[18] Βλ http://gr.china-steppermotor.com/news/baigela-7733774.html & https://circuitdigest.com/article/servo-motor-basics [Τελευταία ανάκτηση: 16 Σεπτεμβρίου 2019].



Εικόνα 2.6.1) Εύρος παλμού και αντιστοίχηση σε μοίρες



Εικόνα 2.6.1) Εύρος και περίοδος λήψης του παλμού

ΚΕΦΑΛΑΙΟ 3

ΜΕΘΟΔΟΛΟΓΙΑ ΕΡΕΥΝΑΣ ΥΛΟΠΟΙΗΣΗΣ ΚΑΙ ΠΡΟΓΡΑΜΑΤΙΣΜΟΥ ΤΟΥ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ

Εισαγωγή

Στο παρόν κεφάλαιο θα γίνει παρουσίαση της μεθοδολογίας που χρησιμοποιήθηκε για την κατασκευή και τον προγραμματισμό του ρομποτικού βραχίονα. Επιπρόσθετα, θα γίνει ανάλυση του κώδικα, ενώ παράλληλα θα γίνει αναφορά και στα προβλήματα που προέκυψαν κατά την πρακτική υλοποίηση της κατασκευής. Τέλος, θα γίνει αναφορά σε όλα τα μέσα όπως τη βιβλιογραφία η τους υπερσυνδέσμους που χρησιμοποιήθηκαν για την μελέτη της παρούσας εργασίας καθώς και μελλοντικές εξελίξεις αυτού του ρομποτικού συστήματος.

3.1 **Συνδεσμολογία και κατασκευή του ρομποτικού βραχίονα**

Η μεθοδολογία που ακολουθήθηκε για την επίτευξη του στόχου, ο οποίος είναι η κατασκευή και ο προγραμματισμός του ρομποτικού βραχίονα, θα αναλυθεί ενδελεχώς στο παρόν υποκεφάλαιο. Αφετηρία της εργασίας στάθηκε η μελέτη του μικροελεγκτή Arduino· αναλυτικότερα η εστίαση έγινε στον τρόπο λειτουργίας του, στα χαρακτηριστικά που παρέχει στους ακροδέκτες του, καθώς και στις εντολές που χρησιμοποιούνται σε συνεργασία με τις βιβλιοθήκες για το κομμάτι του προγραμματισμού.

Εν συνεχεία, αξιοποιήθηκαν οι σειριακοί αριθμοί των κινητήρων για την εύρεση των χαρακτηριστικών τους, ώστε να υπολογιστεί η συνολική ισχύς που θα χρειαστεί για την τροφοδοσία τους. Στην παρούσα εφαρμογή χρησιμοποιήθηκαν οι σερβοκινητήρες DF05BB και DF15MG της εταιρίας DFROBOT, ένας FS5115M της εταιρίας FEETECH, ένας κινητήρας HS-

311της εταιρίας HITEC, και δύο SM-S4315M της εταιρίας SPRINGRC. Παρακάτω, θα παρουσιαστούν και τα χαρακτηρίστηκα τους.[19]

Ο σερβοκινητήρας DF05BB, ο τύπος γραναζιών του οποίου είναι πλαστικός, παρουσίαζει τα εξής τεχνικά χαρακτηριστικά:

Τάση λειτουργίας	4.8 – 6 V
Ρεύμα λειτουργίας	0.1 – 0.8 A
Παραγόμενη ροπή	4.8 kg/cm (4.8V), 5.5kg/cm(6V)
Μέση τιμή της ταχύτητας	60 rpm
Εύρος περιστροφής	0-120 degrees
Θερμοκρασία λειτουργίας	0-55 °C
Μέγεθος	41x20x38mm
Βάρος	48g

Ο σερβοκινητήρας DF15MG, ο τύπος γραναζιών του οποίου είναι μεταλλικός, παρουσίαζει τα εξής τεχνικά χαρακτηριστικά:

Τάση λειτουργίας	4.8 – 7.2V
Ρεύμα λειτουργίας	180mA (4.8V) – 220mA (6V)
Παραγόμενη ροπή	10 kg/cm (4.8V), 12kg/cm(6V) 15kg/cm(7.2V)
Ταχύτητα χωρίς φορτίο	0.17sec/ 60 deg (4.8V) 0.25sec/ 60deg (6V)
Εύρος περιστροφής	0-150 degrees
Θερμοκρασία λειτουργίας	0-60 °C
Μέγεθος	80x60x60mm
Βάρος	180g

[19] Βλ. https://www.dfrobot.com/product-146.html & http://www.feetechrc.com/.html & https://hitecrcd.com/ & http://www.springrc.com/en/.html & https://www.servocity.com/ [Τελευταία ανάκτηση: 17 Σεπτεμβρίου 2019].

Ο σερβοκινητήρας FS-5115M, ο τύπος γραναζιών του οποίου είναι μεταλλικός, παρουσίαζει τα εξής τεχνικά χαρακτηριστικά:

Τάση λειτουργίας	4.8 – 6V
Ρεύμα λειτουργίας με φορτίο Χωρίς φορτίο	1200 mA (4.8V) – 1500mA (6V) 160 mA (4.8V) - 190 mA (6V)
Παραγόμενη ροπή με φορτίο	14kg/cm (4.8V) – 15.5kg/cm (6V)
Ταχύτητα χωρίς φορτίο	0.18sec/ 60 deg (4.8V) 0.16sec/ 60deg (6V)
Εύρος περιστροφής	0-180 degrees +-5%
Απαιτούμενος παλμός	500μs-2500 μs
Μέγεθος	40.8x20.1x38mm
Βάρος	56g

Ο σερβοκινητήρας SM-S4315M ο τύπος γραναζιών του οποίου είναι μεταλλικός, παρουσίαζει τα εξής τεχνικά χαρακτηριστικά:

Τάση λειτουργίας	4.8 – 6V
Ρεύμα λειτουργίας χωρίς φορτίο Με φορτίο	0.58A 2.20A
Παραγόμενη ροπή με φορτίο	13kg/cm (4.8V) – 15.1kg/cm (6V)
Ταχύτητα χωρίς φορτίο	0.21sec/ 60 deg (4.8V) 0.17sec/ 60deg (6V)
Εύρος περιστροφής	0-180 degrees
Μέγεθος	41.3x20.7x40mm
Βάρος	61g

Τέλος, ο σερβοκινητήρας HS-311, ο τύπος γραναζιών του οποίου είναι πλαστικός, παρουσίαζει τα εξής τεχνικά χαρακτηριστικά

Τάση λειτουργίας	4.8 – 6V
Ρεύμα λειτουργίας χωρίς φορτίο Με φορτίο	160mA(4.8V), 180mA(6V) -
Παραγόμενη ροπή με φορτίο	3.02kg/cm (4.8V) – 3.53kg/cm (6V)
Ταχύτητα χωρίς φορτίο	0.19sec/ 60 deg (4.8V) 0.15sec/ 60deg (6V)
Εύρος περιστροφής	0-180 degrees
Απαιτούμενος παλμός	575μs-2460 μs
Μέγεθος	1.57x0.78x1.43mm
Βάρος	42.8g

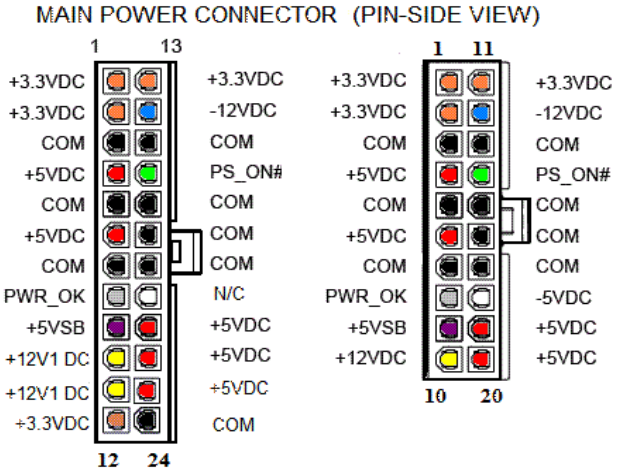
Παρατηρήθηκε ότι η τιμή του αθροίσματος όλων των ρευμάτων των κινητήρων όταν αυτοί βρίσκονται υπό φορτίο στα ανώτατα όριά τους είναι ≈7A. Αν αυτοί οι κινητήρες τροφοδοτηθούν με μια πηγή συνεχούς ρεύματος της τάξης των 5V τότε θα χρειαστεί μια πηγή ισχύος ≈35Watt. Στη συγκεκριμένη όμως εφαρμογή ο ρομποτικός βραχίονας δεν χρησιμοποιεί ταυτόχρονα τους κινητήρες με πλήρες φορτίο, επομένως και με μικρότερη ένταση ρεύματος, αλλά με κατώτερο όριο τα ≈3-4A οι κινητήρες μπορούν να λειτουργήσουν.

Ένα από τα χαρακτηριστικά του μικροελεγκτή είναι ότι η έξοδος στους ακροδέκτες του δεν μπορεί να υποστηρίξει μεγάλα φορτία, καθώς παρέχει στην έξοδο τους μόνο 40mA, όπως προαναφέρθηκε σε προγενέστερο τμήμα της εν λόγω μελέτης. Για τον λόγο αυτό, για την τροφοδοσία των κινητήρων χρησιμοποιήθηκε μια εξωτερική πηγή τάσης, η όποια προέρχεται από ένα τροφοδοτικό ηλεκτρονικού υπολογιστή.

Το μοντέλο του τροφοδοτικού είναι της εταιρίας Channel Well Technology Co και είναι το μοντέλο CWT PSP-575P2P Switching Power Supply ισχύος 400W (Βλ. Εικόνα 3.1). Πιο συγκεκριμένα, απομονώθηκαν κάποια καλώδια (Βλ. Εικόνα 3.1.1), όπως αυτό της γείωσης (Μαύρο), της παροχής των 5V (Κόκκινο) και το καλώδιο το οποίο ενεργοποιεί το τροφοδοτικό (Πράσινο). Παρόλο που μπορεί να ενεργοποιηθεί από τον κεντρικό διακόπτη, δεν λειτουργεί αν

δεν γίνει η γεφύρωσή του με κάποιο από τα καλώδια της γείωσης ή αν δεν είναι συνδεδεμένο σε μητρική πλακέτα υπολογιστή.

Εικόνα 3.1) Τροφοδοτικό 400W CWT PSP-575P2P Switching Power Supply



Εικόνα 3.1.1) Καλωδίωση πλεξούδας του τροφοδοτικού εικοσιτεσσάρων ακροδεκτών[20]

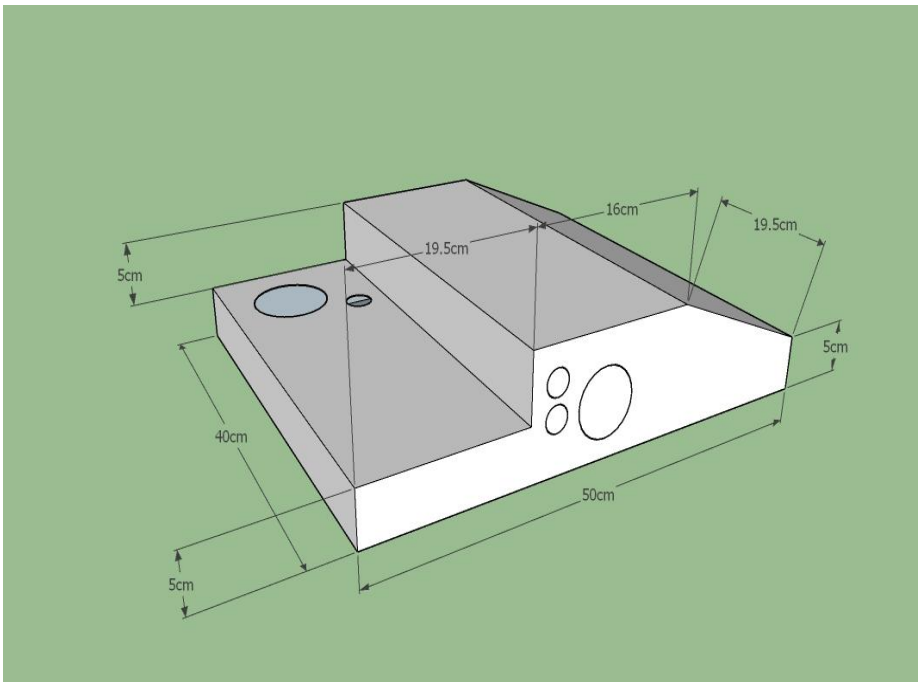
[20] Βλ. http://www.smpspowersupply.com/connectors-pinouts.html [Τελευταία ανάκτηση: 17 Σεπτεμβρίου].

Στη συνέχεια, θεωρήθηκε σωστό από μεριάς μας προτού προχωρήσουμε στο κομμάτι της κατασκευής της βάσης και τον παρελκόμενων, καθώς και της σύνδεσης όλου του βραχίονα, να πειραματιστούμε στον τρόπο λειτουργίας του ενός κινητήρα μέσω τον εντολών που δίνονταν από τον μικροελεγκτή.

Παρόλα αυτά, όταν τελειοποιήθηκε το κομμάτι του προγραμματισμού του ρομποτικού βραχίονα δημιουργήσαμε μια κατασκευή όπου θα συγκρατεί αυτόν και όλα τα παρελκόμενά του.

Στη συγκεκριμένη περίπτωση σχεδιάσαμε μια ιδιοκατασκευή τύπου κονσόλας. Το υλικό που χρησιμοποιήθηκε για τον σκελετό της είναι από φύλλα Plexiglass. Για όλον τον σκελετό χρειάστηκαν τρία φύλλα διαστάσεων 50x50 cm. Τοποθετηθήκαν τα μέτρα της κάθε πλευράς πάνω σε αυτά και στη συνέχεια κοπήκαν με χαμηλή ταχύτητα με μία σέγα.

Οι διαστάσεις του κάθε επιμέρους κομματιού που κόπηκε, καθώς και το αρχικό σχέδιο της κονσόλας παρουσιάζονται στην εικόνα 3.1.2. Το σχέδιο αυτό σχεδιάστηκε στο πρόγραμμα σχεδιασμού SKETCHUP για γραφική απεικόνιση μόνο.



Εικόνα 3.1.2) Σχέδιο και διαστάσεις της κονσόλας[21]

[21] Για το σχέδιο χρησιμοποιήθηκε το παρακάτω πρόγραμμα: https://www.sketchup.com/

Όταν ολοκληρώθηκε η διαδικασία κοπής των φύλλων αυτών πάρθηκαν τα μέτρα στα όποια τοποθετηθήκαν μεταλλικές γωνίες συγκράτησης. Επιπρόσθετα, το πάνω μέρος της κατασκευής που είναι ένα ενιαίο κομμάτι θερμάνθηκε κατά μήκος της επιφάνειας με χρήση φλόγιστρου και πήρε μια κλήση ώστε να τοποθετηθεί αυτούσιο. Κατόπιν, με τη χρήση του τρυπανιού δημιουργήθηκαν οι τρύπες στις οποίες περάστηκαν μετρικές βίδες για τη συγκράτηση των επιφανειών. Επίσης, συγκεντρώθηκαν όλα τα παρελκόμενα, τα οποία τοποθετήθηκαν στην πρόσοψη, καθώς και εντός της κονσόλας, και πάρθηκαν μέτρα ώστε να γίνουν και σε αυτές τις επιφάνειες τρύπες για τη συγκράτηση τους.

Το πάνω μέρος της κονσόλας συγκρατείται με μεταλλικό μεντεσέ, ώστε να ανοίγει όταν χρειαστεί να υπάρξει πρόσβαση εντός της κονσόλας, και δυο μεταλλικά κλιπ εξωτερικά για να μην ανοίγει σε περίπτωση χρήσης της. Αριστερά και δεξιά, στα πλαϊνά της κονσόλας, δημιουργήθηκαν οπές για τη διέλευση του αέρα και για την παροχή ψύξης στα παρελκόμενα, όπως και το τροφοδοτικό. Στη συνέχεια, ελήφθησαν τα μέτρα της καλωδίωσης της κάθε συσκευής, έγιναν οι απαραίτητες αποφλοιώσεις και κολλήθηκαν με καλάι. Επίσης, για την μόνωσή τους χρησιμοποιήθηκαν θερμοσυστελόμενα.

Στην επιφάνεια χειρισμού της κονσόλας υπάρχουν δυο κουμπιά, ένα για την ενεργοποίηση του τροφοδοτικού και ένα για την ενεργοποίηση του μικροελεγκτή Arduino. Επιπρόσθετα, υπάρχει το πληκτρολόγιο KEYPAD, η οθόνη υγρών κρύσταλλου LCD και ο μοχλός JOYSTICK. Εντός της κονσόλας τοποθετήθηκε το τροφοδοτικό, ο μικροελεγκτής, μια μπαταριοθήκη όπου συγκρατεί μια μπαταρία 9V για την τροφοδοσία του μικροελεγκτή, μια δοκιμαστική πλακέτα τύπου Breadboard, όπου είναι τοποθετημένη η πλακέτα συγκράτησης της κάρτας μνήμης καθώς και κάποια καλώδια.

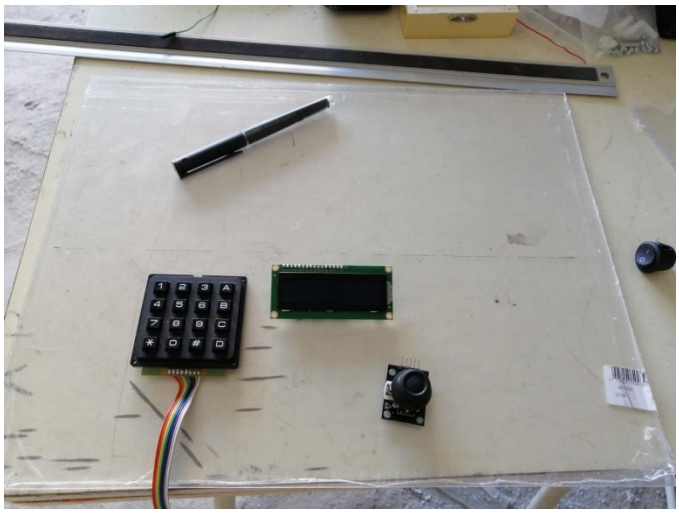
Για τη διαχείριση των καλωδίων χρησιμοποιήθηκαν δεμάτια και αυτοκόλλητα συγκράτησης καλωδίων. Όλες οι καλωδιώσεις τοποθετηθήκαν με τέτοιον τρόπο, ώστε να υπάρχει περιθώριο να ανοίξει το πάνω μέρος της κονσόλας, καθώς και να μην υπάρχει κίνδυνος να τραβηχτεί κάποιο από τα σημεία εφαρμογής τους.

Τα προβλήματα που συναντήθηκαν κατά τη διαδικασία κοπής των φύλλων ήταν ότι το ένα από τα τρία φύλλα πιθανών να ήταν πιο συμπαγές από την κατασκευή του, με αποτέλεσμα να υπάρχει μεγαλύτερη ταλάντωση στη λάμα της σέγας και κατά συνέπεια να σπάει το φύλλο

κατά το πέρας του. Για τον λόγο αυτό και επειδή το υλικό αυτό ανεβάζει εύκολα θερμοκρασία, δόθηκε μεγαλύτερος αέρας στα μέτρα κατά την κοπή του κάθε σκέλους. Αυτό είχε σαν αντίκτυπο τη μερική ασυμμετρία στην κοπή, η οποία παρόλα αυτά διορθώθηκε με τη χρήση λίμας.

Παρακάτω παρουσιάζονται κάποιες εικόνες της κονσόλας, καθώς και η τελική της μορφή.

Εικόνα 3.1.3 λήψη μέτρων και κοπή υλικού



Εικόνα 3.1.4 λήψη μέτρων και κοπή υλικού

Εικόνα 3.1.5 Παρουσιάζεται η τελική μορφή της κατασκευής.

3.2 **Προγραμματισμός και επεξήγηση του κώδικα**

Γενικά, στην αρχή χρησιμοποιήθηκε ένας κινητήρας, και για την κατανόηση της χρήσης τους μέσα από διάφορες εντολές και χρήση βιβλιοθηκών, καθώς και για να προφυλαχτούν οι σύνδεσμοι και το σύστημα μετάδοσης του κάθε σερβοκινητήρα. Παράλληλα συνδέθηκε και ο μοχλός, ο όποιος χρησιμοποιείται για την κίνηση του κάθε κινητήρα και κατά συνέπεια την κίνηση του ρομποτικού βραχίονα. Το πληκτρολόγιο είναι υπεύθυνο για την επιλογή της κάθε άρθρωσης καθώς και την επιλογή της αντιγραφής και επανεκτέλεσης της κίνησης, όπως και της διαγραφής του αρχείου που δημιουργείται στην κάρτα μνήμης. Η οθόνη προστέθηκε για τον οπτικό έλεγχο σε κάποια σημεία που χρειάστηκε παρακολούθηση η συνολική εφαρμογή, όπως για παράδειγμα ποια άρθρωση είναι ενεργοποιημένη ή αν βρίσκεται σε κατάσταση αντιγραφής ή επανεκτελεσης της κίνησης καθώς και τυχόντα σφάλματα. Κατά την παρουσίαση του κώδικα προγραμματισμού θα γίνει αναλυτική επεξήγηση των παραπάνω.

Στη συνέχεια, παρουσιάζονται όλες οι συνδέσεις που έγιναν. Πιο συγκεκριμένα, η οθόνη γραφικής απεικόνισης χαρακτήρων LCD έχει τέσσερεις εξόδους. Αρχικά, την τροφοδοσία, η οποία είναι της τάξεως των 5V και η λήψη της έγινε από την έξοδο του Arduino. Η άλλη έξοδος είναι της γείωσης, η οποία συνδέθηκε στη δοκιμαστική πλακέτα που βρίσκονται συνάμα όλες οι γειώσεις των επιμέρους κυκλωμάτων. Τέλος, οι επόμενες δυο έξοδοι την οθόνης είναι η γραμμή της σειριακής επικοινωνίας των δεδομένων και του συγχρονισμού του παλμού του ρολογιού, οι οποίες συνδέθηκαν στις εξόδους του ελεγκτή στις θύρες 20 (SDA) και 21 (SCL).

Το χειριστήριο JOYSTICK αποτελείται από πέντε εξόδους· την τροφοδοσία, την αναλογική έξοδο του ποτενσιόμετρου της οριζόντιας και της κάθετης κίνησης του μοχλού, την γείωση, και την έξοδο του κουμπιού. Η τροφοδοσία είναι της τάξεως των 5V και λήφθηκε από την έξοδο του μικροελεγκτή. Η γείωση συνδέθηκε στη δοκιμαστική πλακέτα μαζί με όλες τις υπόλοιπες. Η έξοδος που αφορά τον οριζόντιο άξονα (X) συνδέθηκε στην είσοδο A0 του μικροελεγκτή και η έξοδος που αφορά τον κάθετο άξονα (Y) συνδέθηκε στην είσοδο A2 του ίδιου.

Η πλακέτα της κάρτας μνήμης SD παρέχει έξι εξόδους στη δομή της, εκ των οποίων μια είναι της τροφοδοσίας της τάξεως των 3.3V που λήφθηκε από την έξοδο του μικροελεγκτή, επόμενη είναι η γείωση, η οποία συνδέθηκε μαζί με τις υπόλοιπες στη δοκιμαστική πλακέτα. Οι επόμενες τέσσερις είναι η MISO που συνδέθηκε στην ψηφιακή είσοδο 50, η MOSI που συνδέθηκε στην ψηφιακή είσοδο 51, η SCL που συνδέθηκε στην ψηφιακή είσοδο 52, και τέλος η SS που συνδέθηκε στην ψηφιακή είσοδο 53 του Arduino.

Το πληκτρολόγιο KEYPAD παρέχει οχτώ εξόδους στην δομή του, οι οποίες αφορούν τις γραμμές και τις στήλες που υπάρχουν τα αντίστοιχα κουμπιά. Πιο συγκεκριμένα, οι έξοδοι που αφορούν τις γραμμές συνδέθηκαν στις ψηφιακές εισόδους του Arduino 34, 36, 38, 40 και οι γραμμές που αφορούν τις στήλες συνδέθηκαν στις ψηφιακές εισόδους 26, 28, 30, 32.

Όσον αφορά τους κινητήρες, αποτελούνται από τρία καλώδια, τα οποία είναι η τροφοδοσία, η γείωση καθώς και το σήμα έλεγχου τους. Η τροφοδοσία τους είναι συνδεδεμένη στα 5V της εξόδου του τροφοδοτικού. Η γείωσή τους συνδέθηκε με τις γειώσεις όλων των επιμέρους πλακετών. Το σήμα ελέγχου του καθένα είναι συνδεδεμένο στις ψηφιακές εισόδους

του Arduino 3, 5, 6, 9, 11 και 12, οι οποίες είναι υπεύθυνες για την παραγωγή του σήματος PWM.

Ο μικροελεγκτής Arduino τροφοδοτείται από μια εξωτερική πηγή μπαταριάς 9V που συνδέεται στη θηλυκή υποδοχή τύπου JACK της πλακέτας. Ενδιάμεσα, για τον έλεγχο της ενεργοποίησής του παρεμβάλλεται ένας διακόπτης, όπου βρίσκεται στην επιφάνεια χωρισμού της κονσόλας.

Τέλος, έχουν δημιουργηθεί και γραμμές αναμονής. Η μια είναι από την έξοδο του τροφοδοτικού παροχής 5V (σε περίπτωση που εξαντληθεί η μπαταρία τροφοδοσίας του μικροελεγκτή για την άμεση παρέμβαση στην τροφοδοσία του αν δεν υπάρχει άμεση αντικατάστασή της). Επιπρόσθετα, στη δομή των κινητήρων έγινε παρέμβαση στη γραμμή εξόδου του ποτενσιόμετρου (μεσαίο πόδι), με την κόλληση ενός καλωδίου και τη διεξαγωγή από το σώμα του κινητήρα, με αποτέλεσμα να διαβάζονται οι τιμές των τάσεων από τον μικροελεγκτή σαν ανάδραση, δημιουργώντας έτσι ένα σύστημα κλειστού βρόγχου και εκτός του συστήματος έλεγχου του κινητήρα (Feedback). Αυτές οι τιμές αντιστοιχούν στην ακριβή θέση του άξονα του κινητήρα. Αυτό δίνει το πλεονέκτημα να χρησιμοποιηθούν οι εν λόγω πληροφορίες από τον χρήστη για τον εντοπισμό της ακριβής θέσης ενός κινητήρα σε κάποια τυχαία θέση, τον πλήρη έλεγχο ότι έφτασε στην δοθείσα θέση μέσω κάποιας εντολής, καθώς και για την διακρίβωση της θέσης του κινητήρα με τις αντίστοιχες εντολές ελέγχου του προγράμματος.

Στη συνέχεια, παρουσιάζεται ο κώδικας, καθώς και η επεξήγηση του:

Στο περιβάλλον εργασίας του Arduino, όπως και στις περισσότερες πλατφόρμες προγραμματισμού μπορει να γίνει επέκταση της λειτουργίας του προγράμματος χρησιμοποιώντας κάποια επιπλέον προγράμματα τα οποία παρέχουν επιπλέον λειτουργίες. Αυτά τα προγράμματα ονομάζονται βιβλιοθήκες και διευκολύνουν τον χρήστη στη δημιουργία του κώδικα. Οι βιβλιοθήκες είναι δημιουργημένες από άλλους προγραμματιστές και βρίσκονται στην επίσημη σελίδα στο διαδίκτυο της εταιρίας κατασκευής του Arduino. Μαζί με αυτές δίνονται και διάφορες εντολές που συνεργάζονται με τον κώδικα των βιβλιοθηκών. Στην συγκεκριμένη περίπτωση χρησιμοποιήθηκαν οι παρακάτω βιβλιοθήκες:

```
#include <SPI.h>
#include <SD.h>
```

Η συγκεκριμένη περιέχει κώδικα που αφορά το πρωτόκολλο επικοινωνίας SPI, και χρειάζεται για την επικοινωνία με την κάρτα μνήμης SD το οποίο λαμβάνει χώρα στις ψηφιακές ακίδες του μικροελεγκτή όπως προαναφέραμε σε προηγούμενη παράγραφο . Για την εγγραφή και την ανάγνωση από την κάρτα μνήμης χρειάζεται και η SD βιβλιοθήκη.

```
#include <VarSpeedServo.h>
```

Η παραπάνω βιβλιοθήκη έχει δημιουργηθεί για τον έλεγχο 8 σερβοκινητήρων. Πιο συγκεκριμένα χρησιμοποιείται για την ρύθμιση της ταχύτητας και της θέσης του κινητήρα όπου αυτοί οι παράμετροι ορίζονται σε μια εντολή που θα σχολιασθεί παρακάτω. Αυτή η βιβλιοθήκη είναι μια επέκταση της **Servo.h** βιβλιοθήκης.

```
#include <Keypad.h>
```

Η βιβλιοθήκη αυτή χρησιμοποιείται για την σύνδεση πληκτρολογίων keypad τύπου πίνακα με το Arduino. Προωθεί το Hardware Abstraction, το οποίο με τη σειρά του στοχεύει στην ομαλότερη επικοινωνία ανάμεσα στο υλικό και το λογισμικό.

```
#include <Wire.h>
```

Η Wire.h είναι μια βιβλιοθήκη που μας επιτρέπει να επικοινωνήσουμε με συσκευές I2C/TWI.

```
#include <LiquidCrystal_I2C.h>
```

Η βιβλιοθήκη αυτή επιτρέπει τον έλεγχο της οθόνης γραφικής απεικόνισης χαρακτήρων που στην δομή του υλικού της φέρει δυο εξόδους που αφορούν την σύνδεση του δίαυλου I2C κάνοντας χρήση συναρτήσεων όμοιων με αυτές της κοινής LiquidCrystal βιβλιοθήκης.

```
LiquidCrystal_I2C lcd(0x27,16,2);
```

Εδώ δίνεται διεύθυνση στην υποτελή συσκευή της LCD οθόνης, σε συνδυασμό με τους χαρακτήρες (16) και τις γραμμές (2) που την αποτελούν.

```
int counter=0;
int counter1=0;
int counter2=0;
```

Οι παραπάνω ακέραιου τύπου μεταβλητές χρησιμοποιούνται στο κυρίως πρόγραμμα με στόχο την αυτοσυγκράτηση των τιμών του κουμπιού που ελέγχει τον εκάστοτε κινητήρα, της έναρξης καθώς και του πέρατος της εγγραφής αλλά και της αναπαραγωγής αντίστοιχα.

```
int piny = A2;
int pinx = A0;
```

Οι συγκεκριμένες δύο εντολές αντιστοιχίζουν τους ακροδέκτες του joystick που ορίζουν τον άξονα y και τον άξονα x με τις επιθυμητές αναλογικές εισόδους.

```
VarSpeedServo servo1;
int pin1 = A5;
int angle1=70;
int value1;
int value2;
int feedback1;
long int forservo1 = 5000;
```

Στο κομμάτι αυτό δηλώνονται οι μεταβλητές που αφορούν τον κινητήρα. Συγκεκριμένα η αναλογική είσοδος της ανάδρασης (pin1), η αρχική τιμή της θέσης του κινητήρα σε μοίρες (angle1), η τιμή του joystick σε ανάλυση 10-bit από 0 μέχρι 1023 (value1), η τιμή της ανάδρασης σε ανάλυση 10-bit από 0 μέχρι 1023 (value2), η αντιστοίχιση του εύρους τιμών της ανάδρασης με αυτό των μοιρών (feedback1) και η αρχική τιμή του κινητήρα όχι σε μοίρες πια αλλά σε μία τιμή αρκετά μεγαλύτερη η οποία θα μας δώσει το προνόμιο της επιτάχυνσης αλλά και της επιβράδυνσης της ταχύτητας του κινητήρα ανάλογα με τις απαιτήσεις της κίνησης, σκεπτικό το οποίο θα αναλυθεί παρακάτω. Η ίδια διαδικασία ακολουθείται για τη δήλωση των μεταβλητών και των υπόλοιπων κινητήρων (servo2-servo6).

```
VarSpeedServo servo2;
int pin2 = A6;
int angle2=100;
int value3;
int value4;
int feedback2;
long int forservo2 = 7500;
```

```
VarSpeedServo servo3;
int pin3=A7 ;
int angle3=75;
int value5;
int value6;
int feedback3;
long int forservo3 = 7500;

VarSpeedServo servo4;
int pin4 = A8;
int angle4=98;
int value7;
int value8;
int feedback4;
long int forservo4 = 7500;

VarSpeedServo servo5;
int pin5 = A9;
int angle5=90;
int value9;
int value10;
int feedback5;
long int forservo5 = 7500;

VarSpeedServo servo6;
int pin6 = A10;
int angle6=90;
int value11;
int value12;
int feedback6;
long int forservo6 = 10000;

File dataFile;
```

Στην παραπάνω εντολή ορίζουμε μεταβλητή για την αλληλεπίδραση με το αρχείο που θα δημιουργηθεί στην εξωτερική μνήμη (SD card).

```
struct datastore {
  int feedb1;
  int feedb2;
  int feedb3;
  int feedb4;
  int feedb5;
  int feedb6;
};
```

Στο παραπάνω σετ εντολών δημιουργούμε μια δομημένη μεταβλητή με έξι ορίσματα στα οποία θα αποθηκεύονται οι τιμές των γωνιών των κινητήρων. Αυτά τα ορίσματα λαμβάνουν συγκεκριμένο χώρο στην μεταβλητή datastore. Αυτός ο τρόπος προσέγγισης έχει σαν βασικό του πλεονέκτημα την ταυτόχρονη καταχώρηση των τιμών των γωνιών όλων των κινητήρων στα επιμέρους ορίσματα.

```
const int chipSelect = 53;
```

Σε αυτήν την εντολή ορίζεται ως σταθερά η θύρα 53 του μικροελεγκτή στην οποία είναι συνδεδεμένη η υποτελής συσκευή της κάρτας μνήμης.

```
char customKey;
const byte ROWS = 4;
const byte COLS = 4;
```

Στις παραπάνω γραμμές του κώδικα ορίζουμε μια μεταβλητή τύπου χαρακτήρα την `customKey` για να αποθηκευτεί η τιμή του κουμπιού που επιλέξαμε να πατήσουμε μια δεδομένη χρονική στιγμή. Στη συνέχεια δημιουργούνται δύο σταθερές για την δήλωση του πλήθους των γραμμών και των στηλών του πληκτρολογίου.

```
char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
```

Σε αυτό το κομμάτι του κώδικα γίνεται η αντιστοίχιση των γραμμών και των στηλών με τα σύμβολα των αντίστοιχων πλήκτρων.

```
byte rowPins[ROWS] = {34, 36, 38, 40};
byte colPins[COLS] = {26, 28,30,32};
```

Στη συνέχεια ορίζεται η σύνδεση της κάθε γραμμής και στήλης του πληκτρολογίου στις αντίστοιχες θύρες εισόδου του μικροελεγκτή.

```
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS,
COLS);
```

Με την παραπάνω εντολή γίνεται η αρχικοποίηση και η αντιστοίχιση του πλήθους των χαρακτήρων, των γραμμών καθώς και των στηλών με τις αντίστοιχες εξόδους του πληκτρολογίου που θα χρησιμοποιηθούν ως είσοδοι στον μικροελεγκτή.

```
/*Setup Function*/
void setup() {
servo1.attach(3);
servo2.attach(5);
servo3.attach(6);
servo4.attach(9);
servo5.attach(11);
servo6.attach(12);
```

Παραπάνω γίνεται η προσάρτηση και η ενεργοποίηση των κινητήρων μέσω των αντίστοιχων ψηφιακών pin.

```
servo1.write(angle1,10,true);
servo2.write(angle2,10,true);
servo3.write(angle3,10,true);
servo4.write(angle4,10,true);
servo5.write(angle5,10,true);
servo6.write(angle6,10,true);
```

Μετακίνηση των κινητήρων στις αρχικές τους θέσεις με μικρή ταχύτητα για την αποφυγή σπασμωδικών κινήσεων.

```
lcd.init();
lcd.backlight();
```

Με την χρήση των παραπάνω εντολών επιτυγχάνεται η ενεργοποίηση της lcd οθόνης καθώς και του φωτισμού της.

```
Serial.begin(9600);
```

Έναρξη της σειριακής επικοινωνίας μεταξύ του μικροελεγκτή και του υπολογιστή σε συνδυασμό με τον καθορισμό του ρυθμού μετάδοσης των δεδομένων.

```
pinMode(chipSelect,OUTPUT);
pinMode(10,OUTPUT);
pinMode(3,OUTPUT);
pinMode(5,OUTPUT);
pinMode(6,OUTPUT);
pinMode(9,OUTPUT);
pinMode(11,OUTPUT);
pinMode(12,OUTPUT);
```

Ορισμός των ψηφιακών pin ελέγχου των κινητήρων ως εξόδους.

```
while (!Serial) {
;
}

lcd.setCursor(0,0);
lcd.print("Initializing SD:");

Serial.print("Initializing SD card...");

if (!SD.begin(chipSelect)) {

lcd.setCursor(2,1);
lcd.print("Card failed!");
Serial.println("Card failed, or not present");
return;
}
lcd.setCursor(0,1);
lcd.print("card initialized");
delay(1500);
lcd.clear();
Serial.println("card initialized.");

}
```

Στη συνέχεια με την εντολή while (!**Serial**) αναμένεται η επίτευξη της σειριακής σύνδεσης. Κατόπιν ελέγχεται η παρουσία της κάρτας μνήμης και εκτυπώνεται το αντίστοιχο μήνυμα.

```
/*Loop Function*/

void loop() {

customKey = customKeypad.getKey();
```

Με την παραπάνω εντολή το πρόγραμμα είναι δομημένο με τέτοιον τρόπο ώστε να αναμένει το χρήστη να επιλέξει κάποιον κινητήρα για εκκίνηση.

```
if (customKey=='1') {
lcd.clear();
lcd.setCursor(4,0);
lcd.print("MOTION 1");
counter=1;}
if (counter==1){
motion1();

}

if (customKey == '2') {
lcd.clear();
lcd.setCursor(4,0);
lcd.print("MOTION 2");
counter=2;}
if (counter==2){
motion2();
}

if (customKey=='3') {
lcd.clear();
lcd.setCursor(4,0);
lcd.print("MOTION 3");
counter=3;}
if (counter==3){
```

```
motion3();
}

if (customKey == '4') {
lcd.clear();
lcd.setCursor(4,0);
lcd.print("MOTION 4");
counter=4;}
if (counter==4){
motion4();
}

if (customKey == '5') {
lcd.clear();
lcd.setCursor(4,0);
lcd.print("MOTION 5");
counter=5;}
if (counter==5){
motion5();
}

if (customKey == '6') {
lcd.clear();
lcd.setCursor(4,0);
lcd.print("MOTION 6");
counter=6;}
if (counter==6){
motion6();
}
```

Στις παραπάνω γραμμές κώδικα, αν ο χρήστης επιλέξει μέσω του πληκτρολογίου κάποιον αριθμό από το 1 έως το 6, η μεταβλητή counter που λειτουργεί ως αυτοσυγκράτηση θα ελεγχθεί και θα καλέσει την αντίστοιχη ρουτίνα για την κίνηση του επιλεγμένου κινητήρα, προβάλλοντας ταυτόχρονα στην lcd οθόνη το όνομα αυτής προς διευκόλυνση του χρήστη.

```
if (customKey == '*') {
lcd.clear();
lcd.setCursor(5,1);
```

```
lcd.print("RECORD");
counter1=counter1+1;}

if (counter1==1){

record();

}else if (counter1==2){
counter1=0;
lcd.clear();
lcd.setCursor(0,1);
lcd.print("RECORD  COMPLETE");
delay(3000);
lcd.clear();

}
```

Με την επιλογή του πλήκτρου «*» ο μετρητής counter1 παίρνει την τιμή 1, ξεκινώντας παράλληλα την διαδικασία την εγγραφής μέσω της ρουτίνας «record». Συνάμα εάν το συγκεκριμένο πλήκτρο πατηθεί για δεύτερη φορά η διαδικασία της εγγραφής ολοκληρώνεται. Επιπρόσθετα προβάλλονται και τα αντίστοιχα μηνύματα.

```
if (customKey == '#') {
lcd.clear();
lcd.setCursor(4,1);
lcd.print("PLAYBACK");
counter2=1;}

if (counter2==1){

playback();
}
```

Σε αντίθετη περίπτωση αν επιλεγεί το πλήκτρο «#» ο μετρητής counter2 είναι αυτός που θα πάρει την τιμή 1 και η ρουτίνα της αναπαραγωγής της κίνησης των κινητήρων (playback) θα κληθεί. Σε αντίθεση με την ρουτίνα εγγραφής ο τερματισμός της αναπαραγωγής δεν μπορεί να επιτευχθεί πατώντας για δεύτερη φορά το αντίστοιχο πλήκτρο παρά μόνο αν αυτό συμβεί

μέσα στην ίδια τη ρουτίνα λόγω της ύπαρξης του βρόγχου επανάληψης while μέσα σε αυτήν. Τέλος εμφανίζονται και τα αντίστοιχα μηνύματα.

```
if (customKey == '0') {
lcd.clear();
lcd.setCursor(4,1);
lcd.print("SD CLEAR");
SD.remove("servopos.dat");
}
}
```

Στο παραπάνω κομμάτι κώδικα πατώντας το πλήκτρο «0» επιτυγχάνεται η διαγραφή του αρχείου που δημιουργείται στην κάρτα μνήμης. Αυτό το αρχείο χρησιμοποιείται για την αποθήκευση της κίνησης των κινητήρων. Παράλληλα εμφανίζεται και το ανάλογο μήνυμα.

```
/*Motion1*/

void motion1() {

value9 = analogRead(pinx);
value10 = analogRead(pin5);
feedback5 = map(value10, 96, 413, 0, 150);
//value9 = map(value9, 0, 1023, 1, 59);
if (value9 <= 530 && value9 >= 505) {
value9 = 512;
}
forservo5 = forservo5 + (value9 - 512);

if (forservo5 < 1) {
forservo5 = 1;
}
if (forservo5 > 15000) {
forservo5 = 15000;
}
angle5 = map(forservo5, 1, 15000, 0, 150);
servo5.write(angle5,30,true);
delay(1);
}
```

```
/*Motion2*/

void motion2() {
value7 = analogRead(piny);
value8 = analogRead(pin4);
feedback4 = map(value8, 115, 436, 10, 170);
if (value7 <= 540 && value7 >= 470) {
value7 = 512;
}
forservo4 = forservo4 + (value7 - 512);
if (forservo4 < 1) {
forservo4 = 1;
}
if (forservo4 >15000) {
forservo4 = 15000;
}
angle4 = map(forservo4, 1,15000, 10, 170);
servo4.write(angle4,15,true);
delay(1);
}

/*Motion3*/

void motion3() {
value5 = analogRead(piny);
value6 = analogRead(pin3);
feedback3 = map(value6,117, 436, 10, 170);
if (value5 <= 540 && value5 >= 470) {
value5 = 512;
}
forservo3 = forservo3 + (value5 - 512);
if (forservo3 < 1) {
forservo3 = 1;
}
if (forservo3 > 15000) {
forservo3 = 15000;
}
angle3 = map(forservo3, 1,15000, 170, 10);
```

```
servo3.write(angle3,30,true);
delay(1);
}

/*Motion4*/

void motion4() {
value3 = analogRead(piny);
value4 = analogRead(pin2);
feedback2 = map(value4, 57, 592, 0, 180);
//value3 = map(value3, 0, 1023, 1, 59);
if (value3 <= 530 && value3 >= 470) {
value3 = 512;
}
forservo2 = forservo2 + (value3 - 512);
if (forservo2 < 1) {
forservo2 = 1;
}
if (forservo2 > 15000) {
forservo2 = 15000;
}
angle2 = map(forservo2, 1,15000, 0, 180);
servo2.write(angle2,30,true);
delay(1);
}

/*Motion5*/

void motion5() {
value1 = analogRead(pinx);
value2 = analogRead(pin1);
feedback1 = map(value2, 138, 340, 10, 110);
// value1 = map(value1, 0, 1023, 1, 59);
if (value1 <= 530 && value1 >= 505) {
value1 = 512;
}
forservo1 = forservo1 + (value1 - 512);
if (forservo1 < 1) {
```

```
forservo1 = 1;
}
if (forservo1 > 10000) {
forservo1 = 10000;
}
angle1 = map(forservo1, 1, 10000, 544, 2400);
servo1.write(angle1,255,true);
}

/*Motion6*/

void motion6() {
value11 = analogRead(pinx);
value12 = analogRead(pin6);
feedback6 = map(value12, 95, 358, 10, 170);
// value11 = map(value11, 0, 1023, 1, 59);
if (value11 <= 530 && value11 >= 505) {
value11 = 512;
}
forservo6 = forservo6 + (value11 - 512);
if (forservo6 < 1) {
forservo6 = 1;
}
if (forservo6 > 20000) {
forservo6 =20000;
}
angle6 = map(forservo6, 1, 20000, 10, 170);
servo6.write(angle6,150,true);
delay(1);
}
```

Όπως προαναφέρθηκε με την επιλογή των πλήκτρων 1 έως 6 καλείται η αντίστοιχη ρουτίνα κίνησης του εκάστοτε κινητήρα. Αναλυτικότερα με τις εντολές value# = analogRead(pinx ή piny) και value# = analogRead(pin#) ,όπου # το νούμερο ανάλογα με τον επιλεγμένο κινητήρα, διαβάζουμε τις τιμές της θέσης του joystick αλλά και της ανάδρασης του κινητήρα. Έπειτα τις αποθηκεύουμε στις αντίστοιχες μεταβλητές. Στη συνέχεια κάνουμε αντιστοίχιση των ακραίων τιμών της ανάδρασης του κινητήρα με τις ακραίες τιμές των μοιρών του, σε περίπτωση που θελήσει ο χρήστης μέσω της ανάδρασης να έχει τον ακριβή έλεγχο της

θέσης του κινητήρα. Κατόπιν οριοθετούμε τις τιμές του joystick χρησιμοποιώντας τη μέθοδο της βαθμονόμησης, επειδή λόγω κατασκευής η τιμή της θέσης αυτής αποκλίνει κατά έναν βαθμό από το κέντρο. Οριοθέτηση γίνεται και στις τιμές της μεταβλητής forservo# για να μην υπάρξει τιμή εκτός των επιθυμητών ορίων.

Το σκεπτικό πίσω από την κίνηση του κινητήρα με ελεγχόμενη από το χρήστη επιτάχυνση ή επιβράδυνση συνοψίζεται στην παρακάτω εντολή: forservo# = forservo# + (value# - 512); . Έχοντας τον μοχλό στην αρχική του θέση η τιμή της παρένθεσης γίνεται 0 άρα η τιμή της μεταβλητής forservo δεν αλλάζει και ο κινητήρας δεν κινείται. Αν ο μοχλός μετακινηθεί αριστερά ή δεξιά η τιμή της παρένθεσης αποκτά αρνητικό ή θετικό πρόσημο αντίστοιχα. Επίσης όσο περισσότερο αποκλίνει η τιμή της θέσης του μοχλού από το κέντρο, τόσο μεγαλύτερο θα είναι το νούμερο που προστίθεται ή αφαιρείται από την αρχική τιμή του forservo στοιχείο που καθορίζει την επιτάχυνση ή την επιβράδυνση του κινητήρα.

Με πιο απλά λόγια η μεταβλητή forservo έχει ως ορίσματα μεγάλες τιμές από τις οποίες αφαιρούνται ή προστίθενται αρκετά μικρότερες τιμές όπως αυτή της θέσης του μοχλού για να επιτευχθεί ομαλότερη κίνηση στην κίνηση των κινητήρων. Αν χρησιμοποιούνταν μικρότερες τιμές και με βάση την ταχύτητα εκτέλεσης του κώδικα, οι αφαιρέσεις ή οι προσθέσεις θα γινόντουσαν τόσο γρήγορα που δε θα μπορούσε να γίνει αντιληπτό το φαινόμενο της εναλλαγής ταχυτήτων, αλλά και να μην μπορεί να αξιοποιηθεί από τον χρήστη. Οι τιμές αυτές της μεταβλητής forservo ποικίλουν ανάλογα τη ροπή στρέψης του εκάστοτε κινητήρα καθώς και την ταχύτητα περιστροφής του. Επιπρόσθετα αντιστοιχίζονται οι τιμές της τωρινής τιμής της μεταβλητής forservo, την μικρότερη και μεγαλύτερη τιμη της καθώς και αυτές των ορίων μέσα στα οποία θέλουμε να κινείται με ασφάλεια ο κινητήρας. Τέλος δίνεται η εντολή για την κίνηση του κινητήρα σε συνδυασμό με την επιθυμητή ταχύτητα που μας επιτρέπεται βάσει των εντολών της βιβλιοθήκης VarSpeedServo.

```
/*Function Record*/

void record() {
dataFile = SD.open("servopos.dat", FILE_WRITE);
struct datastore myData;
myData.feedb1 = angle1;
myData.feedb2 = angle2;
```

```
myData.feedb3 = angle3;
myData.feedb4 = angle4;
myData.feedb5 = angle5;
myData.feedb6 = angle6;

Serial.print(myData.feedb1);
Serial.print("\t");
Serial.print(myData.feedb2);
Serial.print("\t");
Serial.print(myData.feedb3);
Serial.print("\t");
Serial.print(myData.feedb4);
Serial.print("\t");
Serial.print(myData.feedb5);
Serial.print("\t");
Serial.println(myData.feedb6);

dataFile.write((const uint8_t *)&myData, sizeof(myData));
dataFile.close();
}
```

Αρχικά στην συνάρτηση που υλοποιήθηκε για την εγγραφή των θέσεων των κινητήρων (record) ανοίγουμε την κάρτα SD και δημιουργούμε ένα αρχείο με τίτλο «servopos.dat». Λαμβάνουμε υπόψιν μας τον κανόνα 8+3 που θέλει το όνομα του αρχείου να μην ξεπερνάει τους 8 χαρακτήρες συν άλλους 3 για την επέκταση. Το όρισμα «FILE_WRITE» επεξηγεί τον λόγο για τον οποίο δημιουργήθηκε το συγκεκριμένο αρχείο που δεν είναι άλλος από την εγγραφή δεδομένων. Η χρήση της κάρτας SD έχει ένα βασικό μειονέκτημα, αυτό της αργής εγγραφής μεγάλου όγκου δεδομένων το οποίο επηρεάζει την ταχύτητα κίνησης των κινητήρων αφού οι δυο ρουτίνες αλληλοεξαρτώνται και λειτουργούν παράλληλα. Για την καταπολέμηση του φαινομένου χρησιμοποιήθηκαν δύο μέθοδοι. Πρώτον η πειραματική εύρεση της κατάλληλης τιμής για την μεταβλητή forservo έτσι ώστε να μην φαίνεται η επιβράδυνση της κίνησης κατά τη λειτουργία εγγραφής και δευτερευόντως αυτή της εγγραφής των δεδομένων σε γλώσσα μηχανής και όχι σε αρχείο κειμένου. Η βιβλιοθήκη της SD στο Arduino υποστηρίζει την εγγραφή τέτοιου είδους δεδομένων εξαιρετικά καλά.

Αφού λοιπόν αποθηκεύσουμε τις τιμές των γωνιών του κάθε κινητήρα στην αντίστοιχη επιμέρους μεταβλητή μιας νέας μεταβλητής με όνομα myData, προβάλλουμε τις τιμές αυτές στην σειριακή απεικόνιση του Arduino μέσω της εντολής **Serial**.print. Στη συνέχεια υπάρχει η εντολή dataFile.write((const uint8_t *)&myData, sizeof(myData)). Η εντολή «&myData» βρίσκει την διεύθυνση της μνήμης όπου είναι αποθηκευμένα τα δεδομένα. Αφού είναι τύπου «struct datastore» πρέπει να αλλαχτεί σε τύπο «unsigned byte». Αυτή η ενέργεια απαιτείται για την εύρυθμη συνεργασία με τη συνάρτηση «write» και επιτυγχάνεται με την εντολή (const uint8_t *)&myData. Τώρα η συνάρτηση «write» βλέπει τη μεταβλητή myData σαν ένα πίνακα από bytes. Η εντολή sizeof(myData) είναι μία συνάρτηση που μας επιστρέφει το μέγεθος των μεταβλητών και αν τα έξι επιμέρους στοιχεία της μεταβλητής myData είναι για παράδειγμα στο σύνολό τους 16 bytes θα δεσμεύσει τον χώρο αυτό στην SD για να αποθηκευτούν τα δεδομένα. Τέλος κλείνουμε τη μεταβλητή dataFile μετά από κάθε εγγραφή με την εντολή dataFile.close().[22]

```
/*Function Playback*/

void playback() {
dataFile = SD.open("servopos.dat",FILE_READ);
if (!dataFile) {
counter2=0;
// Serial.println("second time pressed #");
return;
}
else{
while (dataFile.available()) {
struct datastore myData;
dataFile.read((uint8_t *)&myData, sizeof(myData));

servo1.write(myData.feedb1,30,true);
servo2.write(myData.feedb2,30,true);
servo3.write(myData.feedb3,30,true);
servo4.write(myData.feedb4,20,true);
servo5.write(myData.feedb5,30,true);
servo6.write(myData.feedb6,30,true);
```

[22] Βλ. https://hackingmajenkoblog.wordpress.com/2016/03/25/fast-efficient-data-storage-on-an-arduino/ [Τελευταία ανάκτηση: 17 Σεπτεμβρίου 2019].

```
Serial.print(myData.feedb1);
Serial.print("\t");
Serial.print(myData.feedb2);
Serial.print("\t");
Serial.print(myData.feedb3);
Serial.print("\t");
Serial.print(myData.feedb4);
Serial.print("\t");
Serial.print(myData.feedb5);
Serial.print("\t");
Serial.println(myData.feedb6);

counter=0;

customKey = customKeypad.getKey();
if (customKey == '#'){
servo1.write(70,10,true);
servo2.write(100,10,true);
servo3.write(75,10,true);
servo4.write(98,10,true);
servo5.write(100,10,true);
servo6.write(100,10,true);
counter2=0;
lcd.clear();
lcd.setCursor(4,0);
lcd.print("PLAYBACK");
lcd.setCursor(3,1);
lcd.print("CANCELLED!");
delay(1500);
lcd.clear();
lcd.setCursor(2,0);
lcd.print("SET ROBOT TO");
lcd.setCursor(0,1);
lcd.print("REFERENCE POINT");
delay(1500);
lcd.clear();

return;
```

```
}
}
dataFile.close();
Serial.println("ekleise o fakelos");
}
}
```

Η συνάρτηση αναπαραγωγής της αποθηκευμένης κίνησης των κινητήρων (playback) λειτουργεί αν και για όσο είναι διαθέσιμη η μεταβλητή του αρχείου (dataFile). Αρχικά αφού ανοίξει το αρχείο μέσα στην κάρτα SD στο οποίο είναι αποθηκευμένες οι τιμές των γωνιών των κινητήρων, στοχεύει στην ανάκτηση των δεδομένων αυτών μέσω της εντολής dataFile.read((uint8_t *)&myData, sizeof(myData)). Έπειτα με τις εντολές servo1.write(myData.feedb1,30,true) κ.ο.κ δίνουμε εντολή στους κινητήρες να μετακινηθούν στη θέση που τους ορίζουν οι επιμέρους μεταβλητές feedb με τις ταχύτητες που τους ορίζουμε εμείς αφού η ταχύτητα ανάγνωσης της κάρτας SD διαφέρει από την ταχύτητα εγγραφής. Στη συνέχεια δίνουμε στη μεταβλητή counter την τιμή 0 διότι δεν επιθυμούμε την κίνηση των κινητήρων με χειροκίνητο τρόπο. Παράλληλα οι αποθηκευμένες θέσεις των κινητήρων προβάλλονται στην σειριακή απεικόνιση χωρισμένες σε στήλες. Όπως προαναφέρθηκε λόγω του ότι δεν μπορούμε να σταματήσουμε στο κυρίως πρόγραμμα την αυτόματη αναπαραγωγή πατώντας κάποιο πλήκτρο και συγκεκριμένα αυτό της δίεσης (#), δίνουμε τη δυνατότητα στο χρήστη μέσα στο βρόγχο επανάληψης να μπορεί να τερματίσει τη συνάρτηση playback μέσα από την εντολή customKey = customKeypad.getKey(). Πατώντας το πλήκτρο της δίεσης η μεταβλητή counter2 θα πάρει την τιμή 0 και οι κινητήρες θα επιστρέψουν με αργή ταχύτητα στην αρχική τους θέση. Εν παραλλήλω θα εμφανιστούν και τα αντίστοιχα μηνύματα στην lcd οθόνη, με την εντολή return θα βγούμε από το βρόγχο επανάληψης της while και αφού κλείσουμε τη μεταβλητή dataFile θα επιστρέψουμε στο κυρίως πρόγραμμα.

3.3) **Μελλοντική εξέλιξη**

Η αναγκαιότητα συνεχούς εξέλιξης της τεχνολογίας βρίσκει αντίκτυπο στις μεγάλες δυνατότητες που μας προσφέρει ο προγραμματισμός. Μελλοντικά η παρούσα εργασία για παράδειγμα θα μπορούσε να αναπτυχθεί περαιτέρω οδηγώντας σε μεγαλύτερες δυνατότητες και ακρίβεια. Έχοντας αφήσει σε εκκρεμότητα τη δυναμική λειτουργία της ανάδρασης παρέχεται η δυνατότητα να περάσουμε από τη μειονεκτική θέση ενός συστήματος ανοικτού βρόγχου σε αυτή ενός κλειστού. Επεξηγηματικά αυτό σημαίνει πως ο χρήστης θα έχει πια τον απόλυτο έλεγχο για το αν οι κινητήρες έχουν οδηγηθεί με ακρίβεια στο σημείο που τους έχει υποδειχθεί. Ακόμα αν για κάποιο λόγο υπάρξει έστω και μια μικρή απόκλιση από την επιθυμητή τιμή με τη λειτουργία της ανάδρασης ο κινητήρας θα επιστρέψει σε αυτήν. Όσον αφορά το κομμάτι της κινησιολογίας θα ήταν δυνατό να υπάρξει μεγαλύτερη ευχρηστία μέσω της κινηματικής θεωρίας καθώς και της χρήσης αισθητήρων όπως για παράδειγμα αυτού της πίεσης, ο οποίος σε συνδυασμό με κατάλληλο προγραμματισμό θα απέτρεπε την αρπάγη από τον τραυματισμό εύθραυστων αντικειμένων ακόμα και κάτω από συνθήκες λάθος χειρισμού από τον χρήστη.

3.4) **Βιβλιογραφία και ιστοτοπογραφία**

Γενική βιβλιογραφία

Auslander David M. – Kempf Carl J, *Μηχατρονική. Προσαρμοστικά μηχανικών συστημάτων*, μτφρ. – επιμ. Herbert Tanner, Αθήνα, Πανεπιστημιακές Εκδόσεις Ε.Μ.Π., 1998.

Δουλγέρη Ζωή, *Ρομποτική· κινηματική, δυναμική και έλεγχος αρθρώσεων βραχίονα*, Αθήνα, Κριτική, 2007.

Κουμπουλής Φώτης Ν. –Μέρτζιος Βασίλης Γ., *Εισαγωγή στη Ρομποτική*, Αθήνα, Παπασωτηρίου, 2002.

Πολίτης Γ. – Τσεκούρας Γ., *Ηλεκτρικές Μηχανές· Δομικό Στοιχείο των Συστημάτων Αυτομάτου Ελέγχου*, Αθήνα, Τμήμα Αυτοματισμού, Σ.Τ.Ε. – Τ.Ε.Ι. Πειραιά, 2009.

Ιστοτοπογραφία

https://ilektroaytomatismoi.blogspot.com/2018/07/blog-post_4.html
[Τελευταία ανάκτηση: 3 Αυγούστου 2019].
https://www.arduino.cc/
[Τελευταία ανάκτηση: 3 Αυγούστου 2019].
https://store.arduino.cc/arduino-mega-adk-rev3
[Τελευταία ανάκτηση: 3 Αυγούστου 2019].
https://grobotronics.com/microsd-card-breakout.html
[Τελευταία ανάκτηση: 3 Αυγούστου].
https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all
[Τελευταία ανάκτηση: 3 Αυγούστου 2019].
https://grobotronics.com/basic-16x2-character-lcd-white-on-blue-5v-i2c-protocol.html#group_94505d95f7dd2f47b-1
[Τελευταία ανάκτηση: 20 Αυγούστου 2019].
https://www.nxp.com/docs/en/user-guide/UM10204.pdf
[Τελευταία ανάκτηση: 1 Σεπτεμβρίου 2019].
https://howtomechatronics.com/tutorials/arduino/how-i2c-communication-works-and-how-to-use-it-with-arduino/

[Τελευταία ανάκτηση: 1 Σεπτεμβρίου 2019].
https://circuitdigest.com/microcontroller-projects/keypad-interfacing-with-8051-microcontroller
[Τελευταία ανάκτηση: 1 Σεπτεμβρίου 2019].
https://grobotronics.com/keypad-matrix-4x4.html
[Τελευταία ανάκτηση: 1 Σεπτεμβρίου 2019].
https://lastminuteengineers.com/joystick-interfacing-arduino-processing/
[Τελευταία ανάκτηση: 1 Σεπτεμβρίου 2019].
https://grobotronics.com/thumb-joystick-module.html
[Τελευταία ανάκτηση: 1 Σεπτεμβρίου 2019].
https://www.servocity.com/how-does-a-servo-work
[Τελευταία ανάκτηση 16 Σεπτεμβρίου 2019].
http://gr.china-steppermotor.com/news/baigela-7733774.html
[Τελευταία ανάκτηση: 16 Σεπτεμβρίου 2019].
https://circuitdigest.com/article/servo-motor-basics
[Τελευταία ανάκτηση: 16 Σεπτεμβρίου 2019].
https://www.dfrobot.com/product-146.html
[Τελευταία ανάκτηση: 17 Σεπτεμβρίου 2019].
http://www.feetechrc.com/.html
[Τελευταία ανάκτηση: 17 Σεπτεμβρίου 2019].
https://hitecrcd.com/
[Τελευταία ανάκτηση: 17 Σεπτεμβρίου 2019].
http://www.springrc.com/en/.html
[Τελευταία ανάκτηση: 17 Σεπτεμβρίου 2019].
https://www.servocity.com/
[Τελευταία ανάκτηση: 17 Σεπτεμβρίου 2019].
http://www.smpspowersupply.com/connectors-pinouts.html
[Τελευταία ανάκτηση: 17 Σεπτεμβρίου 2019].
https://www.sketchup.com/

https://hackingmajenkoblog.wordpress.com/2016/03/25/fast-efficient-data-storage-on-an-arduino/

[Τελευταία ανάκτηση: 18 Σεπτεμβρίου 2019].