

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΑΥΤΟΜΑΤΙΣΜΟΥ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ ΜΕ
ΜΝΗΜΕΣ»



Επιβλέπων Καθηγητής:

ΜΙΧΑΛΗΣ ΠΑΠΟΥΤΣΙΔΑΚΗΣ

Στοιχεία φοιτητών:

ΜΑΝΔΑΡΑΚΑ ΜΑΡΙΑ ΑΜ: 40371

ΣΦΑΚΙΑΝΟΣ ΙΩΑΝΝΗΣ ΑΜ: 40463

ΑΙΓΑΛΕΩ, ΦΕΒΡΟΥΑΡΙΟΣ 2016

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο / Η κάτωθι υπογεγραμμένος / η Στακιάς Γιάννης....., του Γεωργίου....., με αριθμό μητρώου 40463..... φοιτητής / τρια του Τμήματος **Μηχανικών Αυτοματισμού Τ.Ε.** του Α.Ε.Ι. Πειραιά Τ.Τ. 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.»

Επίσης δηλώνω υπεύθυνα ότι έχω παρακολουθήσει το σεμινάριο συγγραφής και εκπόνησης πτυχιακής εργασίας που διοργανώνεται από το Τμήμα Μηχανικών Αυτοματισμού Τ.Ε. κατά το Χειμερινό/Εαρινό Εξάμηνο του Ακ. Έτους 2015-2016

Ο Δηλών

Ημερομηνία

α) Στακιάς

16/2/2016

β) Μ. Μανδραράκη

ΜΑΡΙΑ ΜΑΝΔΡΑΡΑΚΑ
ΤΟΥ ΕΜΜΑΝΟΥΗΛ
Α.Μ.: 40371

Περίληψη

Η παρούσα πτυχιακή εργασία έχει ως στόχο την εκπαίδευση και εξοικείωση των μαθητών και φοιτητών με την λειτουργία ενός αυτόματου ρομποτικού βραχίονα. Αντιπροσωπεύει τους ρομποτικούς βραχίονες που χρησιμοποιούνται στην βαριά βιομηχανία, όπου ο σωστός χειρισμός και προγραμματισμός τους είναι αναγκαίος, τόσο για την σωστή λειτουργία της βιομηχανίας, όσο και για την ασφάλεια των εργαζομένων. Μετά την εξοικείωση του μαθητευόμενου με τον χώρο λειτουργίας του βραχίονα, και τον χειρισμό του με ασφάλεια, θα είναι σε θέση να τον προγραμματίσει με μεγαλύτερη άνεση. Η χρήση και ο προγραμματισμός του βραχίονα στα σχολεία θα είναι ένα σημαντικό πλεονέκτημα για τους μελλοντικούς φοιτητές, προκειμένου να γνωρίσουν έμπρακτα τον αυτοματισμό για πρώτη φορά με ασφάλεια. Οι φοιτητές θα μπορούν να επαναλάβουν της βασικές αρχές ενός ρομποτικού βραχίονα, και να προβούν σε δυσκολότερες εφαρμογές.

Η λειτουργία χωρίζεται σε δυο μέρη. Το πρώτο μέρος βασίζεται στις μνήμες, προκειμένου να δείξουν στον ενδιαφερόμενο τις δυνατότητες κίνησης του ρομποτικού βραχίονα, σχηματίζοντας συγκεκριμένα σχήματα που επιλέχθηκαν από εμάς, ανάλογα με το ποιο κουμπί (button) έχει επιλέγει. Τα σχήματα αυτά πραγματοποιούνται σε άξονες $x-y$ ή σε $y-z$. Επιλέχθηκαν ο κύκλος και η καρδιά ως τα βασικά σχήματα. Ο κύκλος επιλέχθηκε προκειμένου να δείξει ότι ενώ υπάρχουν περιορισμοί στις κινήσεις των σερβοκινητήρων, με τον κατάλληλο κώδικα ο καθένας μπορεί να τους ξεπεράσει. Η καρδιά αντιπροσωπεύει ένα συνδυαστικό σχήμα δυο ημικυκλίων και δυο κάθετων γραμμών, με απώτερο σκοπό να μπορεί ο ενδιαφερόμενος να κατανοήσει ότι μπορεί να δημιουργήσει σχήματα ημιτελή καθώς και παραλληλόγραμμα, ή τριγωνικά. Το keypad περιλαμβάνει και ένα button που πραγματοποιεί ελικοειδή κίνηση (spiral). Αυτή η κίνηση επιλέχθηκε ώστε ο μαθητευόμενος να μπορεί να δει πως μπορεί να προγραμματίσει κινήσεις τριών διαστάσεων $x-y-z$. Η αυτοματοποιημένη λειτουργία χρησιμοποιεί την θεωρία της αντίστροφη κινηματικής, προκειμένου να πραγματοποιήσει της επιθυμητές κινήσεις. Το δεύτερο μέρος αποτελείται από την χειροκίνητη λειτουργία του εν λόγω βραχίονα με το χειριστήριο.

Για την πραγματοποίηση της εφαρμογής χρησιμοποιήθηκε ο ρομποτικός βραχίονας, μοντέλου: 6 DOF Robotic Arm. Ο επεξεργαστής που χρησιμοποιήθηκε στον προγραμματισμό για τις κινήσεις του βραχίονα, είναι το Arduino Mega ADK μέσω του οποίου υπάρχει επικοινωνία με το πληκτρολόγιο (keypad). Κάθε button υποδηλώνει μια πλήρη κίνηση με επιστροφή στην αρχική θέση. Πέρα των αυτοματοποιημένων κινήσεων, το

keypad περιλαμβάνει και το button επιλογής της χειροκίνητης λειτουργίας. Για την παύση και την επανεκκίνηση της χειροκίνητης λειτουργίας υπάρχει ένα μαύρο button απομονωμένο από το keypad. Για την άμεση παύση όλης της εφαρμογής υπάρχει το κόκκινο button. Το σύστημα του βραχίονα τροφοδοτείται μέσω του τροφοδοτικού υπολογιστή και ενός κυκλώματος κανονικοποιητή (regulator).

Abstract

The aim of this thesis is to educate and familiarize students of both lower and higher education with the use of an automatic robotic arm. This sample is widely used for industrial purposes; it is essential that these robotic arms are used and programmed carefully, not only to promote their accurate industrial use, but also to ensure the safety of the work force. As soon as one gets better acquainted with the functions and correct handling of the robotic arm, he will be able to control and program it a lot more easily. Furthermore, their use in schools will contribute in introducing a new set of abilities for students to come, helping them come into contact with automation for the very first time. Thereafter, the students in question will be able to replicate the fundamental principles of robotic arms and, more importantly, proceed into more difficult applications.

There are two parts that contribute in the functioning procedure of a robotic arm. The first part is based on memory, which demonstrate the capacities of the robotic arm in terms of movement; they form specific types of shapes depending on which button has been pressed by the operator. These shapes are performed in an axis x-y or y-z. For this study, the two chosen types were a circular and a heart-shaped one. The circle was chosen in an attempt to show that, even though there are certain restrictions in the movements of the servomotors, everyone can overcome them using the appropriate code. The shape of the heart was accomplished through a combinatorial shape of two semi-circles and two vertical lines, which would naturally help the operator in realizing his potential in creating imperfect, parallelogram as well as triangular shapes. The keypad consists of a button that produces spiral movement. This movement was chosen in order for the students to understand how to program three-dimensional movements (x-y-z). This automatic function implicates the theory and equations of inverse kinematics, in order to achieve the intended movements. The second part is mainly associated with the manual function of said robotic arm with the nunchuck.

This practice was carried out by using the 6 DOF Robotic arm. The processor that was applied throughout the programming procedure for the movements of the robotic arm was the Arduino Mega ADK; this model provides a link with the keypad and each button leads to a complete movement and direct return to the initial position. Apart from these automatic movements, the keypad includes a button that allows manual use. In order to pause or restart this manual function, there is also a black button which is isolated from the keypad itself. Furthermore, a red button is also installed in case there is a need for immediate cancellation of

the practice. Lastly, the robotic system is fueled by the power supply (of a computer) and a regulator.

Περιεχόμενα

Ευχαριστίες	9
Εισαγωγή	10
ΚΕΦΑΛΑΙΟ 1	11
Ιστορική αναδρομή του Ρομποτικού βραχίονα και της Σύγχρονης Εκπαίδευσης	11
ΚΕΦΑΛΑΙΟ 2	12
Εφαρμογές Εξελιγμένων Ρομποτικών Βραχιόνων	12
2.1. Ρομποτικός Βραχίονας Συγκολλήσεων	12
2.2. Ρομποτικός Βραχίονας Παλετοποιήδης Φαρμακοβιομηχανίας	13
2.3. Ρομποτικός Βραχίονας Συσκευαστής Φαρμακοβιομηχανίας	14
2.4. Ρομποτική Χειρουργική	15
2.5. Ρομποτικός Σκακιστής	16
ΚΕΦΑΛΑΙΟ 3	18
Κατασκευή	18
3.1. Βραχίονας	18
3.2. Χειριστήριο	19
3.3. Keypad	20
3.4. Μικροεπεξεργαστής Arduino Mega ADK	21
Συνδεσμολογία	23
3.5. Τροφοδοσία.....	23
3.6. Κανονικοποιητής / Ρυθμιστής (Regulator).....	25
3.7. Button παύσης και εκκίνησης της χειροκίνητης λειτουργίας.....	27
ΚΕΦΑΛΑΙΟ 4	28
Πρόγραμμα και η Επεξήγησή του	28
ΚΕΦΑΛΑΙΟ 5	39
Μελλοντική εξέλιξη	39

Επίλογος	39
Παράρτημα	41
1. Introduction.....	41
2. System Description.....	42
3. The Interface Environment.....	43
4. Conclusion and Further Work.....	49
Βιβλιογραφία	50
Λίστα Εικόνων	52

Ευχαριστίες

Η παρούσα εργασία έχει ένα συγκεκριμένο σκοπό, να γίνει το πρώτο εισαγωγικό βήμα, προκειμένου να ωθήσει νέα άτομα στον τεχνολογικό τομέα, και ειδικότερα στον αυτοματισμό. Δεν θα ήταν εφικτό να ολοκληρωθεί η εργασία αυτή, χωρίς την καθοδήγηση του κυρίου Μιχαήλ Παπουτσιδάκη και του κυρίου Μάκη Χατζόπουλου για την βοήθεια τους στην υλοποίηση της παρούσας εργασίας. Επίσης οφείλουμε ένα μεγάλο ευχαριστώ στον κύριο Κωσταντίνο Εφρεμίδη, που με την πολύτιμη βοήθεια του, η εργασία είναι στον πιο επιτυχημένο της βαθμό. Τους οφείλουμε ένα ευχαριστώ εις βάθη καρδιάς.

Εισαγωγή

Η εξέλιξη του Homo Erectus στηρίχτηκε στην συσσωρευτική γνώση της κατασκευής και χρήσης εργαλείων, γεγονός που οδήγησε στη μετεξέλιξη του πολιτισμού του και στην απαρχή της τεχνολογικής ανάπτυξης.

Στην σύγχρονη εποχή της τεχνολογικής γιγάντωσης τα παιδιά έρχονται ευθύς εξαρχής σε επαφή με την τεχνολογία, γεγονός που καθιστά απαραίτητη τη χρήση των μέσων αυτής, στην εκπαιδευτική διαδικασία, ώστε να επέλθει η εξοικείωση με αυτή και η πρόκτηση των βασικών δεξιοτήτων.

Η εισαγωγή του ρομποτικού βραχίονα στην εκπαιδευτική διαδικασία θα συμβάλει ώστε οι νέοι να γνωρίσουν και να κατανοήσουν σε βάθος μια πτυχή της τεχνολογικής ανάπτυξης ώστε να συνδράμουν και οι ίδιοι στην περαιτέρω εξέλιξή της.

ΚΕΦΑΛΑΙΟ 1

Ιστορική αναδρομή του Ρομποτικού βραχίονα και της Σύγχρονης Εκπαίδευσης

Η φιλοδοξία του ανθρώπου να δημιουργήσει μηχανές που θα του έμοιαζαν τόσο στην μορφή όσο και στην λειτουργία, τη συναντάμε για πρώτη φορά στην ελληνική μυθολογία. Ο Τάλος, ο μυθικός χάλκινος γίγαντας που κατασκεύασε ο Ήφαιστος για να προστατεύει την Κρήτη από τους εισβολείς, αποτελεί το πρώτο “αυτόματο ρομπότ” στην ανθρωπινή ιστορία. Η λέξη robot εισήχθηκε στο αγγλικό λεξιλόγιο το 1921, από τον θεατρικό συγγραφέα Karel Capek στο έργο του Rossum's Universal Robots. (βλέπε [1])

Η Unimation ήταν η πρώτη εταιρεία που παρήγαγε ρομπότ, ιδρύθηκε από τον George C. Devol και τον Joseph F. Engelberger το 1956. Το πρώτο μοντέλο που κατασκεύασε ήταν ένας χειριστής ο οποίος εκτελούσε μια σειρά επαναλαμβανόμενων κινήσεων. Η λειτουργία του ήταν η μεταφορά αντικειμένων από ένα σημείο σε κάποιο άλλο, για αποστάσεις 4 μέτρων το πολύ, και ονομάζονταν "μηχανές προγραμματισμένων μεταφορών". Ο Victor Scheinman στο Πανεπιστήμιο του Στάνφορντ, 1969, ανακάλυψε το "βραχίονα του Στάνφορντ". Αποτελούσε τον πρώτο ρομποτικό σύστημα με έξι βαθμούς ελευθερίας (έξι αρθρώσεις) που μπορούσε να πραγματοποιεί αυθαίρετες κινήσεις στο χώρο με ακρίβεια. Τη δεκαετία που ακολούθησε, ιδρύθηκαν νέες εταιρίες οι οποίες λόγω ανταγωνισμού διαμόρφωσαν τον τομέα της ρομποτικής οπός είναι γνωστός σήμερα. (βλέπε [2])

Η ραγδαία εξέλιξη της τεχνολογίας αναπόφευκτα επηρέασε όλους τους τομείς της ανθρώπινης εξέλιξης και ασφαλώς η εκπαίδευση δεν θα μπορούσε να μην αποτελέσει ένα από τα άμεσα πεδία εφαρμογής της. Το 1960, εισήχθησαν δοκιμαστικά σε κάποια σχολεία το ραδιόφωνο και η τηλεόραση και αποτέλεσαν τα πρώτα βήματα για την οπτικό-ακουστική διδασκαλία. Οι παιδαγωγοί και ψυχολόγοι παρατήρησαν την θετική επιρροή των τεχνολογικών μέσων στους μαθητούμενους και ενέργησαν προκειμένου να εισαχθούν περισσότερες τεχνολογικές εφαρμογές στην παιδεία. Μέχρι το έτος του 2000, η παιδεία παγκοσμίως εμπλούτισε τον εκπαιδευτικό εξοπλισμό, τα επιμορφωτικά υλικά και την εξατομικευμένη μάθηση. (βλέπε [3])

ΚΕΦΑΛΑΙΟ 2

Εφαρμογές Εξελιγμένων Ρομποτικών Συστημάτων

Τα ρομποτικά συστήματα και ιδιαίτερα οι ρομποτικοί βραχίονες, χρησιμοποιούνται ευρέως σε όλους τους σημαντικούς τομείς την καθημερινότητά μας. Η εξέλιξη της τεχνολογίας έχει συνδεθεί άμεσα με την εξέλιξη της βιομηχανίας. Ο ρομποτικός βραχίονας χρησιμοποιείται σε εφαρμογές συσκευασίας, όπως εγκιβωτισμός, παλετοποίηση - αποπαλετοποίηση καθώς και για ειδικές εργασίες. Αλλάζει κεφαλή ανάλογα με την εκάστοτε εφαρμογή, αποκτώντας εξαιρετική ευελιξία. Κάποια τυπικά παραδείγματα παρουσιάζονται στις παρακάτω παραγράφους.

2.1. Ρομποτικός Βραχίονας Συγκολλήσεων

Ένα από τα βασικότερα εργαλεία της βιομηχανίας είναι ο ρομποτικός βραχίονας συγκολλήσεων. Λειτουργεί συνήθως με 6 βαθμούς ελευθερίας και σε άξονες X-Y-Z. Στην άκρη του βραχίονα, έχει τον συγκολλητικό πυρσό και είναι συνδεδεμένος με τη μηχανή συγκόλλησης, ρυθμίζοντας μέσω προγράμματος τη λειτουργία και τις παραμέτρους της, ανάλογα με το εκάστοτε κατεργαζόμενο υλικό. Προσφέρει βέλτιστη ακρίβεια και ταχύτητα συγκόλλησης, ενώ εξασφαλίζει άριστη και σταθερή ποιότητα του παραγόμενου προϊόντος, σε ελάχιστο χρόνο. Χρησιμοποιείται για την συγκόλληση υλικών, συνήθως βαρέων μετάλλων, τον συναντάμε συχνά σε αυτοκινητοβιομηχανία, σε διυλιστήρια και σε άλλες παρόμοιες εφαρμογές. (βλέπε [4]) Στην παρακάτω εικόνα φαίνεται ένας τυπικός ρομποτικός βραχίονας συγκολλήσεων:



Εικόνα 2.1. Διάταξη Ρομποτικού Βραχίονα Συγκολλήσεων.

Προσπέλαση στις 2/2/2016. Υπερσύνδεσμος: <http://weldinganswers.com/can-you-benefit-from-robotic-arc-welding/> (Προσοχή: Στο θεωρητικό μέρος αυτής της εργασίας όλες οι εικόνες και τα σχήματα που χρησιμοποιήθηκαν, είναι αποκλειστικά για εκπαιδευτική χρήση.)

2.2. Ρομποτικός Βραχίονας Παλετοποίησης Φαρμακοβιομηχανίας

Τα ρομποτικά αυτά συστήματα κινούνται σε 4 με 5 άξονες, έχουν την δυνατότητα ανύψωσης φορτίων μέχρι και 35 κιλών και λειτουργούν σε ταχύτητα κίνησης της τάξης των 180 μέτρων το λεπτό. Όλη η διαδικασία και τα επιμέρους βήματα, μέχρι και την έξοδο, είναι πλήρως αυτοματοποιημένα. Η λειτουργία τους χωρίζετε σε δυο σκέλη, την συλλογή και την αποθήκευση φαρμάκων.

Οι παραγγελίες των κιβωτίων με τα φάρμακα συλλέγονται και κωδικοποιούνται. Κάθε παραγγελία έχει έναν ειδικό κωδικό. Η κωδικοποίηση των φαρμάκων και καταχώριση των πληροφοριών δηλώνονται στο σύστημα WMS (Warehouse Management System), ένα ευφυές σύστημα αυτόματης επιλογής κωδικών, και επιλογής της βέλτιστης θέσης αποθηκευτικών ραφιών. Τα φάρμακα τοποθετούνται σε κωδικοποιημένες, ανάλογα με την παραγγελία, παλέτες, και μεταφέρονται μέσω των ταινιόδρομων (ιμάντων) στο σημείο τελικού ελέγχου, όπου ελέγχονται και συσκευάζονται. Η αποθήκευση των φαρμάκων.

Αφού τα δεδομένα των φαρμάκων έχουν καταχωρηθεί στο WMS, τα παλέτα μεταφέρονται με τους ιμάντες μέχρι το σημείο παραλαβής των γερανών στοίβαξης. Οι γερανοί έχουν ένα ειδικό λογισμικό και αποτελούν μια σειρά διαδικασιών, δοσμένες από το σύστημα WMS. Ανάλογα με το ποια σειρά ραφιών θα πρέπει να τοποθετηθεί η κάθε παλέτα, ο βραχίονας την τοποθετεί στη θέση που το σύστημα έχει επιλέξει, όπου και αποθηκεύεται. (βλέπε [5])



Εικόνα 2.2. Διάταξη Ρομποτικού Βραχίονα Παλετοποίησης Φαρμακοβιομηχανίας

Προσπέλαση στις 2/2/2016. Υπερσύνδεσμος: <http://www.razor.tv/video/969190/robots-help-to-dispense-medicine-at-tan-tock-seng-hospital>

2.3. Ρομποτικός Συσκευαστής Φαρμακοβιομηχανίας

Τα τελευταία χρόνια στον υγειονομικό χώρο έχει κάνει την εμφάνισή του ο ρομποτικός συσκευαστής φαρμακείου. Λειτουργεί πάντα σε κλειστό χώρο, για την διατήρηση του αποστειρωμένου περιβάλλοντος, καθώς και για την προστασία του από εξωτερικούς παράγοντες. Αφού καταχωρηθεί, στο σύστημα του υπολογιστή του φαρμακείου, η αίτηση της παραγγελίας του πελάτη, από τον υπάλληλο, ο βραχίονας κινείται μεταξύ των αποθηκευτικών ραφιών και συλλέγει τα σωστά φάρμακα. Εντοπίζει την τοποθεσία του συγκεκριμένου φαρμάκου, που του ζητήθηκε, χρησιμοποιώντας τον σαρωτή γραμμωκόδικα (Bar-code scanner). Μέσω ενός συστήματος Bar-code, κωδικοποιεί και τοποθετεί μια ετικέτα πάνω στο πακέτο με τα στοιχεία του πελάτη και τις πληροφορίες για την σωστή λήψη του φαρμάκου. Ύστερα στέλνει το πακέτο, μέσω ενός συστήματος με μάντα, στον χώρο παράδοσης. Ο υπάλληλος συγκρίνει την παραγγελία με το πακέτο και το παραδίδει στον πελάτη. (βλέπε [6])



Εικόνα 2.3. Ρομποτικός Συσκευαστής Φαρμακείου.

Προσπέλαση στις 13/8/2015. Υπερσύνδεσμος: <http://www.cambridge-news.co.uk/Addenbrooke-8217-s-pharamacy-team-reduces/story-28348464-detail/story.html>

2.4. Ρομποτική Χειρουργική

Η ιδέα της ρομποτικής χειρουργικής ξεκίνησε από το ενδιαφέρον της NASA και του στρατού των Ηνωμένων Πολιτειών της Αμερικής, ώστε ειδικοί χειρουργοί να μπορούν να πραγματοποιούν επεμβάσεις σε αστροναύτες, με επείγοντα χειρουργικά προβλήματα υγείας και τραυματισμένοι στρατιώτες στο πεδίο της μάχης. Το 1995 δημιουργήθηκε το ρομποτικό σύστημα da Vinci, το μοναδικό μέχρι σήμερα σύστημα ρομποτικής χειρουργικής. Πραγματοποιεί εγχειρήσεις με την ελάχιστη δυνατή επέμβαση στον οργανισμό του ασθενούς και εξασφαλίζει στον χειρουργό την επιτυχή και ασφαλή διεξαγωγή της επέμβασης. Είναι εφοδιασμένο με εύκαμπτα λαπαροσκοπικά εργαλεία που θυμίζουν την ελευθερία κίνησης του ανθρώπινου καρπού και δίνει την δυνατότητα στον χειρουργό να εκτελεί πολύπλοκες κινήσεις ακριβείας. Επίσης διαθέτει ένα σύστημα φακών τρισδιάστατης απεικόνισης, το οποίο μεγεθύνει το χειρουργικό πεδίο μέχρι και 15 φορές. Το σύστημα da Vinci αποτελείται από τρία βασικά μέρη, όπως φανερωθεί στην εικόνα παρακάτω: την τροχήλατη βάση τρισδιάστατης απεικόνισης, την τροχήλατη βάση βραχιόνων και την κονσόλα του χειρουργού. (βλέπε [7])



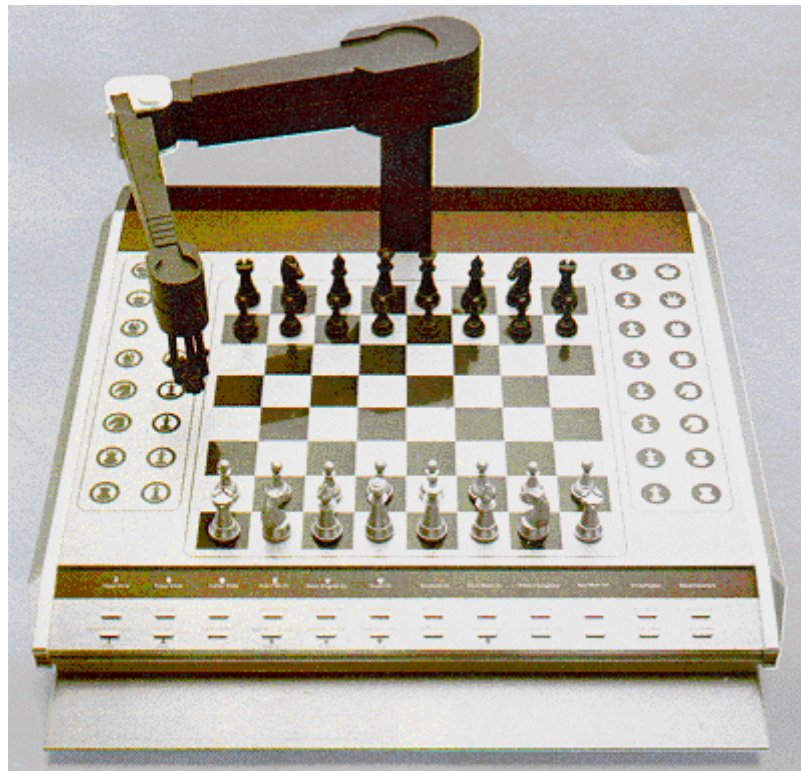
Εικόνα 2.4. Ο Εξοπλισμός της Ρομποτικής Χειρουργικής.

Προσπέλαση στις 19/7/2015. Υπερσύνδεσμος: <http://urology-athens.gr/%CE%B5%CF%80%CE%B5%CE%BC%CE%B2%CE%AC%CF%83%CE%B5%CE%B9%CF%82/32-%CF%81%CE%BF%CE%BC%CF%80%CE%BF%CF%84%CE%B9%CE%BA%CE%AE-%CF%87%CE%B5%CE%B9%CF%81%CE%BF%CF%85%CF%81%CE%B3%CE%B9%CE%BA%CE%AE.html>

2.5. Ρομποτικός Σκακιστής

Ο ρομποτικός σκακιστής συναντάται, τους τελευταίους τρεις αιώνες, με πολλές διαφορετικές διατάξεις και κατασκευές, αλλά όλες βασίζονται στην ίδια αρχή λειτουργίας. Αποτελείται από έναν αισθητήρα (συνήθως στον πίνακα του σκακιού), που ειδοποιεί τον υπολογιστή για το πια κίνηση πραγματοποιήθηκε. Επίσης, ο κώδικάς του είναι πολύ συγκεκριμένος, και η τεχνογνωσία του στηρίζεται σε μια «μηχανή σκακιού», προκειμένου να παράγει ο ρομποτικός σκακιστής την εκάστοτε κίνησή του. Η δυσκολία του παιχνιδιού ρυθμίζεται μέσω του κώδικά του. Η λειτουργία του πραγματοποιείται σε πραγματικό χρόνο και πραγματικό περιβάλλον. Ο ρομποτικός σκακιστής χρειάζεται ένα λογισμικό σύστημα το οποίο είναι ικανό να διαχειρίζεται ένα ιδικό hardware, για να δημιουργεί ένα εύκολα διαχωρίσιμο περιβάλλον, το οποίο να υποστηρίζει αλλαγές του λογισμικού του (software). Το σύστημα πραγματοποιεί αναδραστικούς έλεγχους θέσης και προσανατολισμού. Λόγο της ασφαλούς λειτουργίας του ο ρομποτικός σκακιστής χρησιμοποιείται από τους ψυχιάτρους ως

για θεραπεία για αντιστικά άτομα, και πλέον αποτελεί ένα μέσο ψυχαγωγίας για τους λάτρεις του παιχνιδιού. (βλέπε [8])



Εικόνα 2.5. Ρομποτικός Σκακιστής.

Προσπέλαση στις 19/7/2015. Υπερσύνδεσμος: <https://chessdailynews.com/the-chess-machine/>

ΚΕΦΑΛΑΙΟ 3

Κατασκευή

Σε αυτό το κεφάλαιο αναλύονται όλα τα εξαρτήματα που χρησιμοποιήθηκαν για την πραγματοποίηση της παρούσας πτυχιακής εργασίας.

3.1. Βραχίονας

Ο ρομποτικός βραχίονας που επιλέχθηκε για την υλοποίηση της παρούσας πτυχιακής εργασίας, είναι το μοντέλο: 6 DOF Robotic Arm. Καλύπτει όλες τις βασικές κινήσεις, για την προσομοίωση του ως ανθρωπινό χέρι, τις οποίες θέλαμε να μελετήσουμε και να παρουσιάσουμε. Λόγο του μικρού του μεγέθους και τον περιορισμό της ημικυκλικής κίνησης, επιτυγχάνουμε την αποφυγή τραυματισμού των μαθητευόμενων. Όπως αναφέρθηκε οι σερβοκινητήρες πραγματοποιούν ημικυκλική κίνηση της τάξης των 180 μοιρών. Όμως η κατασκευή του βραχίονα εμποδίζει κάποιους κινητήρες να κάνουν την πλήρη κίνησή τους. Αυτό το σφάλμα της κίνησης του βραχίονα επιλέχτηκε να γίνει σκόπιμα, προκειμένου να μπορεί να εφαρμοστεί η θεωρία της κινηματικής, όπως θα εξηγηθεί στην συνέχεια. Λειτουργεί με έξι (6) βαθμούς ελευθερίας. Οι χαρακτηριστικοί βαθμοί ελευθερίας του είναι: περιστρεφόμενη βάση, ομαλή κίνηση ώμου, αγκώνα, καρπού και λειτουργία της αρπάγης. (βλέπε [9]) Αποτελείτε από τα παρακάτω μέλη:

1. Gripper x1
2. U shape brackets x3
3. Multifunction brackets x4
4. L shape bracket x1
5. Bearing x3
6. Rotation Base x1
7. Hitec 311 Servo x2
8. DSS-M15 Servo x2
9. DF05BB Servo x1
10. HS422 Servo x1
11. Screw sets x16

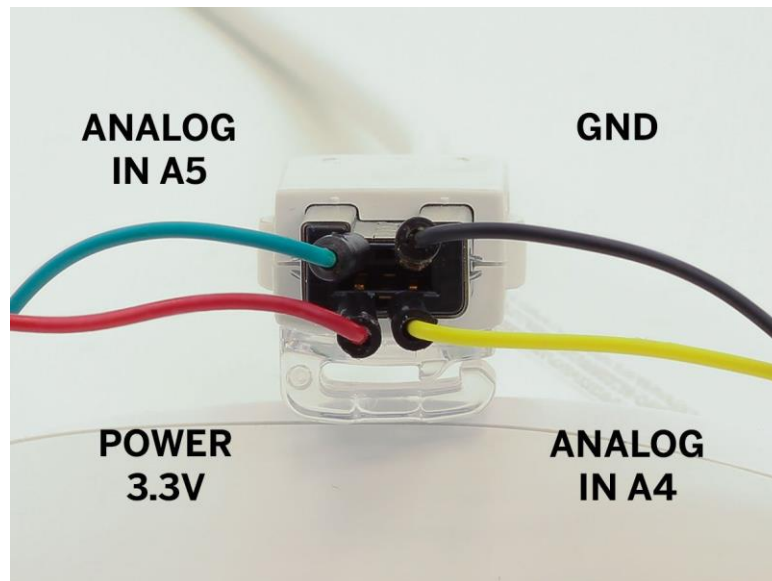
Χαρακτηρίστηκα του ρομποτικού βραχίονα:

1. Τάση: +4.8-7.2 V
2. Ρεύμα: 2000mA

3. Ύψος του βραχίονα: 320mm
4. Βάρος του βραχίονα: 600g

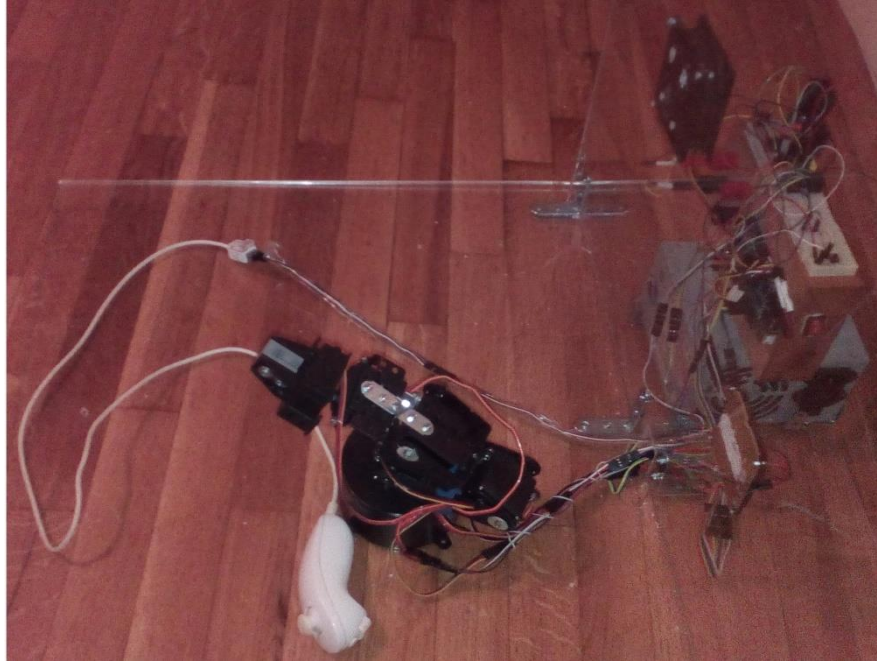
3.3. Χειριστήριο

Το χειριστήριο του βραχίονα που χρησιμοποιείται είναι το κλασικό χειριστήριο του Wii. Αντλαμβάνεται τρεις διαστάσεις ως άξονες χ-ψ-ζ. Το χειριστήριο nunchuck έχει 4 καλωδιώσεις. Το κόκκινο καλώδιο είναι η τροφοδοσία 3.3 Volt, το άσπρο είναι η γείωση, το κίτρινο είναι συριακό ωρολογιακό σήμα SCL (Serial Clock Line), και το πράσινο είναι το συριακό σήμα μετάδοσης δεδομένων SDA (Serial Data Line). Επίσης είναι συμβατό με το πρωτόκολλο I2C Bus. Το πρωτόκολλο αυτό αποτελείται από τα καλώδια SCL και SDL. Το SCL χρησιμοποιείται για να συγχρονίζει τα δεδομένα που μεταφέρονται. Το SDA ρυθμίζει σε ποια κατεύθυνση θα κινηθεί το σήμα και σε ποια συσκευή απευθύνεται και επιτρέπει στα δεδομένα να μεταφερθούν από τον αισθητήρα στον μικροεπεξεργαστή και αντίστροφα. Το σήμα που στέλνεται δεν μπορεί να είναι υψηλό (υψίσυχνο) και η λειτουργία του στηρίζεται στην φιλοσοφία κυρίου-εξαρτημένου (master-slave). (βλέπε [10])



Εικόνα 3.1. Καλωδιώσεις Nunchuck

Προσπέλαση στις 19/11/2015. Υπερσύνδεσμος: <http://arduinoarts.com/2014/07/tutorial-wii-nunchuck-pan-tilt-servo/>



Εικόνα 3.2. Ο Ρομποτικός Βραχίονας με το Χειριστήριο.

3.4. Keypad

Το αριθμητικό πληκτρολόγιο (Keypad) λειτουργεί χωρίς τροφοδοσία, και συνδέεται, μέσω του Breadboard, με το Arduino. Έχει οχτώ (8) επαφές (pins), όπου κάθε pin αντιστοιχεί σε μια γραμμή ή μια στήλη. Αποτελείται από 16 πλήκτρα (buttons) με αρχική κατάσταση «ανοιχτού διακόπτη». Ο μικροελεγκτής αντιλαμβάνεται ποιο πλήκτρο ενεργοποιείτε, όταν το πλήκτρο αυτό πατηθεί, και ο ανοιχτός διακόπτης κλείνει, επιτρέποντας να περάσει ρεύμα στα αντίστοιχα δυο pins που αντιστοιχούν στο πλήκτρο. Τα πλήκτρα A, B, C και D υπάρχουν για μελλοντική χρήση. Θα καθορίζουν την διαβάθμιση της δυσκολίας της λειτουργίας του ρομποτικού βραχίονα. Η επιλογή A θα αποτελεί την παρούσα λειτουργία του βραχίονα. Οι επιλογές B, C, D αναφέρονται στο τελευταίο κεφάλαιο.



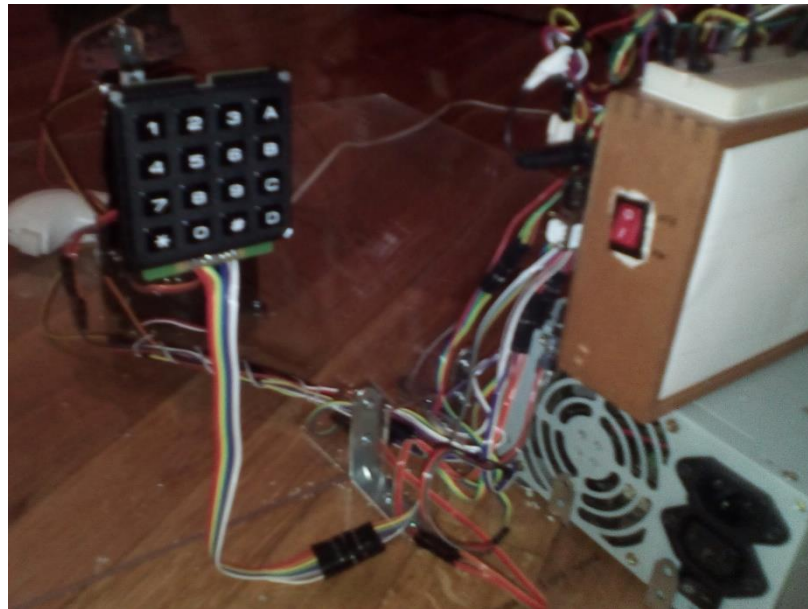
Εικόνα 3.3. Αριθμητικό Πληκτρολόγιο (Keypad).

Προσπέλαση στις 2/2/2016. Υπερσύνδεσμος:

http://www.vetco.net/catalog/product_info.php?products_id=14280

Χαρακτηριστικά του Keypad: (βλέπε [11])

1. Contact Rating: 20 mA, 24 VDC
2. Contact Resistance: 200 Ohms Max
3. Life: 1.000.000 Cycles Pre Key
4. Contact: Conductive Rubber
5. Size (Key Face Cutout): 60 (W) x 56.75 (H) x 3.5 (D)



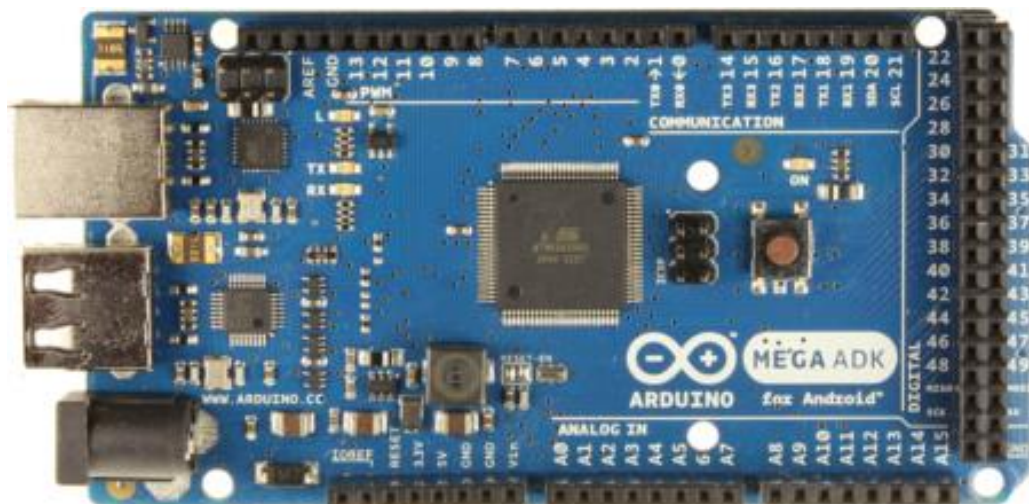
Εικόνα 3.4. Το Keypad Συνδεδεμένο με τον Μικροεπεξεργαστή.

3.4. Μικροεπεξεργαστής “Arduino Mega ADK”

Τα Arduino είναι ευέλικτοι μικροεπεξεργαστές και ενώ δεν είναι ο προτιμότερος μικροεπεξεργαστής για πολύπλοκες εφαρμογές, η μεγάλη ποικιλία του και τα εύχρηστα εργαλεία που προσφέρει, το κάνει ιδιαίτερα δημοφιλή. Ουσιαστικά είναι μια πλατφόρμα «ανοιχτού κώδικα». Υπάρχουν πολλές διαφορετικές εκδόσεις του Arduino, που κυρίως διαφοροποιούνται στην ποσότητα της διαθέσιμης μνήμης, στον αριθμό των διαθέσιμων pins εισόδων και εξόδων, στον αριθμό των σειριακών διασυνδέσεων (interface), καθώς και στην ύπαρξη των επιπλέον χαρακτηριστικών εξαρτημάτων. Για τις ανάγκες της πτυχιακής, χρειαστηκε να χρησιμοποιησουμε το Arduino Mega ADK, έναντι του Arduino Uno, το οποίο

αναφέρετε στην δήλωση της παρούσας πτυχιακής. Τα χαρακτηριστικά του Arduino Mega ADK και οι διαφορές του με το Arduino Uno παρουσιάζονται παρακάτω.

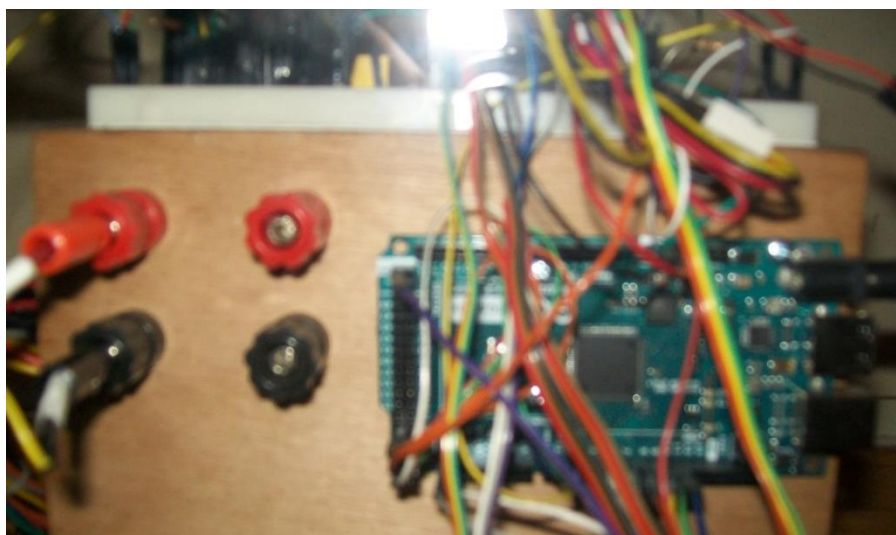
Το βασικό του πλεονέκτημα του Arduino Mega ADK είναι ότι συμπεριλαμβάνει 54 ψηφιακά, 16 αναλογικά pins και 4 σειριακές πορτές (ports), ενώ το Arduino Uno, μόνο 14 ψηφιακά και 6 αναλογικά pins. Το Mega ADK διαθέτει μεγαλύτερη μνήμη (8 KB SRAM και 4KB EEPROM), χρησιμοποιώντας παρόμοιο μικροεπεξεργαστή με το Uno και δεν διαφέρουν στον προγραμματισμό, που είναι μια πιο σύγχρονη παραλλαγή της C++. (βλέπε [12])



Εικόνα 3.5. Arduino Mega ADK.

Προσπέλαση στις 20/9/2015. Υπερσύνδεσμος:

<https://www.arduino.cc/en/Main/ArduinoBoardMegaADK>



Εικόνα 3.6. Ο Μικροεπεξεργαστή και η Συνδεσμολογίες του.

Χαρακτηρίστηκα του Arduino Mega ADK:

1. Microcontroller	ATmega2560
2. Operating Voltage	5V
3. Input Voltage (recommended)	7-12V
4. Input Voltage (limits)	6-20V
5. Digital I/O Pins	54 (of which 15 provide PWM output)
6. Analog Input Pins	16
7. DC Current per I/O Pin	40 mA
8. DC Current for 3.3V Pin	50 mA
9. Flash Memory	256 KB of which 8 KB used by bootloader
10. SRAM	8 KB
11. EEPROM	4 KB
12. Clock Speed	16 MHz
13. USB Host Chip	MAX3421E
14. Length	101.52 mm
15. Width	53.3 mm
16. Weight	36 g

Συνδεσμολογία

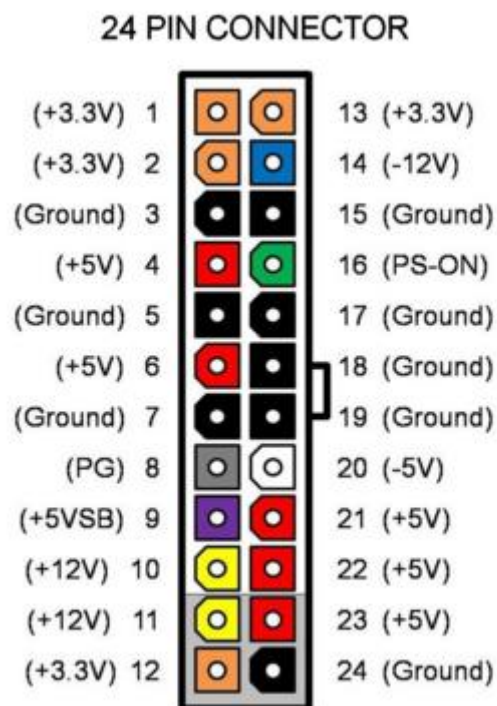
Στην συνέχεια παρουσιάζονται οι συνδεσμολογίες μεταξύ των προαναφερθέντων εξαρτημάτων.

3.5. Τροφοδοσία

Η τροφοδοσία που χρησιμοποιήθηκε αποτελείται από ένα τροφοδοτικό ηλεκτρονικού υπολογιστή, της εταιρίας CWT. Το τροφοδοτικό προμηθεύει το κύκλωμα με τάση των 12Volt και με ρεύμα των 3 Amper, περίπου, καθώς και την γείωση. Από τις εικόνες παρακάτω φαίνεται η συνδεσμολογία εντός του κουτιού που επιλέχτηκε, για την προστασία της, καθώς και την χρωματική σημασία των καλωδίων του τροφοδοτικού.



Εικόνα 3.7. Συνδεσμολογία Εντος του Κουτιου.

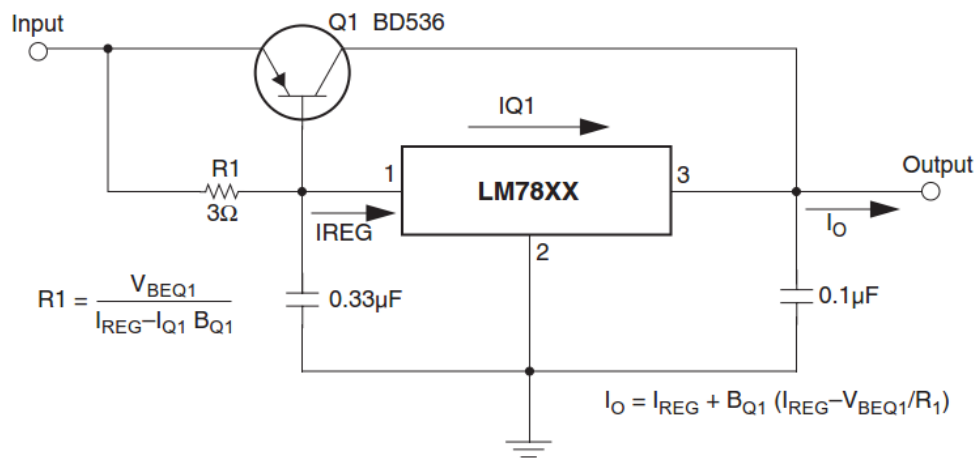


Εικόνα 3.8. Χρωματική Κωδικοποίηση Καλωδίων.

Προσπέλαση στις 1/8/2015. Υπερσύνδεσμος: <http://makezine.com/projects/computer-power-supply-to-bench-power-supply-adapter/>

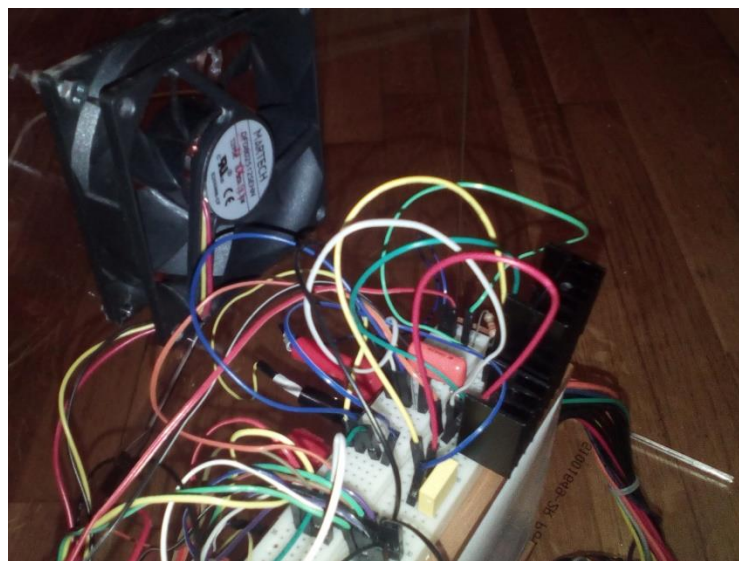
regulator. Το regulator χρησιμοποιείτε προκειμένου να λάβει μια τάση (Volt) και να την μειώσει, ανάλογα τα εξαρτήματα που το απαρτίζουν και την συνδεσμολογία του κυκλώματός του, αλλά να μην επηρεάζει το ρεύμα (Amber).

Το συγκεκριμένο κύκλωμα regulator χρησιμοποιείτε να μειώσουμε τα 12 Volt σε 6 Volt. Οι μαθηματικοί τύποι που αναγράφονται, χρησιμοποιούνται για να εξετάσουμε ποια αντίσταση (R1) απαιτείται να συνδεθεί στο κύκλωμα και ποιο είναι το ρεύμα (Io) που λαμβάνει ο βραχίονας. (βλέπε [14]) Η έξοδος του regulator τροφοδοτεί όλους τους σερβοκινητήρες με 6 Volt και 3 Amber, καθώς και με την γείωση που είναι κοινή με του τροφοδοτικού και του Arduino, προκειμένου να παράξει το Arduino παλμό σήματος.



Εικόνα 3.10. Κυκλώμα Regulator.

FAIRCHILD. Σεπτέμβριος 2014. Σελίδα 20. Προσπέλαση στις 22/8/2015. Υπερσύνδεσμος:
<https://www.fairchildsemi.com/datasheets/lm/LM7805.pdf>

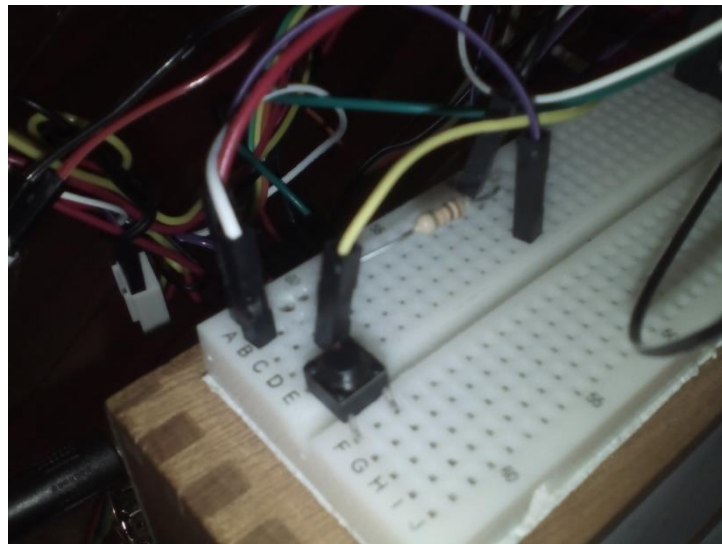


Εικόνα 3.11. Συνδεσμολογία του Regulator.

Παρατήρηση: Επειδή το κύκλωμα του regulator θερμαίνεται πολύ, έχουμε προσθέσει ένα ανεμιστηράκι στο Arduino προκειμένου να το ψύχει.

3.7. Button παύσης και εκκίνησης της χειροκίνητης λειτουργίας

Το button που χρησιμοποιήθηκε είναι ανοιχτής κατάστασης NO (Normally Open). Συνδέσαμε στην πάνω-αριστερά άκρη του button μια αντίσταση 10kΩ, που είναι συνδεδεμένη την γείωση. Από την ίδια άκρη συνδέσαμε ένα καλώδιο με το pin 2 του Arduino, προκειμένου να χρησιμοποιήσουμε την κυριότερη διακοπή (interrupt) του Arduino, ώστε το σήμα του button να έχει προτεραιότητα στο πρόγραμμα. Η κάτω-αριστερή άκρη του button συνδέεται με την τάση των 5 Volt του Arduino. (βλέπε [15])



Εικόνα 3.12. Συνδεσμολογία του Button Παύσης και Εκκίνησης της Χειροκίνητης Λειτουργίας.

ΚΕΦΑΛΑΙΟ 4

Πρόγραμμα και η Επεξήγησή του

Το κυρίως πρόγραμμα και η επεξήγηση του, παρουσιάζετε παρακάτω. Προκειμένου να επεκτείνουμε το περιβάλλον προγραμματισμού του Arduino Mega ADK, χρησιμοποιήσαμε μερικές “βιβλιοθήκες”(Libraries). Οι βιβλιοθήκες παρέχουν επιπλέον λειτουργικότητα στον προγραμματισμό, και υπάρχουν οι βασικές βιβλιοθήκες που παρέχει το Arduino, οι βιβλιοθήκες που έχουν δημιουργηθεί από προγραμματιστές για βελτίωση των ήδη υπάρχοντων βιβλιοθηκών, αλλά και βιβλιοθήκες για ιδικές εργασίες.

```
#include <VarSpeedServo.h>           //Βιβλιοθήκη για τον έλεγχο των
σερβοκινητήρων. (βλέπε [16])
```

```
#include <Keypad.h>                 //Βιβλιοθήκη για την λειτουργία του Keypad. (βλέπε [17])
```

```
#include <Wire.h>                   //Βιβλιοθήκη για την ανταλλαγή δεδομένων μεταξύ του
αισθητήρα του χειριστηρίου και του Arduino.
```

```
#include <ArduinoNunchuk.h>         //Βιβλιοθήκη για τον έλεγχο του χειριστηρίου
(βλέπε [18])
```

Η εντολή `#define` μας επιτρέπει να ονοματίσουμε μια σταθερή τιμή πριν την έναρξη του κύριου προγράμματος.

```
#define ZEROX 344                   //Δηλώνουμε τα όρια που κινείται ο βραχίονας με το
χειριστήριο σε άξονες X - Y - Z για την διαμέτρηση του χειριστηρίου. (για τον Z άξονα
συμπεριλαμβάνεται και το ύψος της βάσης του βραχίονα) (βλέπε [19])
```

```
#define ZEROY 344
```

```
#define ZEROZ 410
```

```
#define WII_NUNCHUK_I2C_ADDRESS 0x52 //Ορίζουμε την διεύθυνση του
nunchuck στο Arduino. Επίσης το wii nunchuck έχει ενσωματωμένο το σύστημα
επικοινωνίας I2C bus το οποίο είναι υπεύθυνο για την μεταφορά δεδομένων από μια συσκευή
σε μια άλλη.
```

```
int counter; // μετρητής δεδομένων του nunchuck
```

```
uint8_t data[6]; // δείκτης για την αποθήκευση της εξόδου του ArduinoNunchuck
```

```
ArduinoNunchuk nunchuck = ArduinoNunchuk();
```

```

const byte ROWS = 4; //Δημιουργούμε αντικείμενο keypad (βλέπε [20])

const byte COLS = 4;

char keys[ROWS][COLS] = { //Απεικόνιση του Keypad στον κωδικά.

    {'1','2','3','A'},

    {'4','5','6','B'},

    {'7','8','9','C'},

    {'*','0','#','D'}

};

byte rowPins[ROWS] = {4,5,6,7}; //Σύνδεση των γραμμών του keypad, στα pins
του Arduino

byte colPins[COLS] = {8,9,10,11}; //Σύνδεση των κολόνων

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

Οι διαστάσεις του βραχίονα σε χιλιοστά (mm).

#define BASE_HGT 66.00 //Υψος βάσης.

#define HUMERUS 92.00 //Βραχιόνιο οστό. (Ωμος προς αγκώνα)

#define ULNA 92.00 //Ωλένιο οστό. (Αγκώνας προς καρπό)

#define GRIPPER 160.00 //Απόσταση της αρπάγης με φορτίο, παράλληλη από το
έδαφος

#define ftl(x) ((x)>=0?(long)((x)+0.5):(long)((x)-0.5)) //Κανονικοποίηση τιμών,
προκειμένου να εξομαλύνουμε τις κινήσεις των σερβοκινητήρων

Αρχικοί υπολογισμοί.

float hum_sq = HUMERUS*HUMERUS;

float uln_sq = ULNA*ULNA;

Ορίζουμε αντικείμενα της βιβλιοθήκης.

VarSpeedServo BAS;

VarSpeedServo SHL;

VarSpeedServo ELB;

```



```
VarSpeedServo WRI;
```

```
VarSpeedServo WRO;
```

```
VarSpeedServo GRI;
```

Μεταβλητές για το button παύσης της χειροκίνητης λειτουργίας. (βλεπε [15])

```
boolean startprogram = true;
```

```
volatile boolean emstop = false;
```

```
void setup(){
```

```
    nunchuk.init();
```

Ορίζουμε τα pins του κάθε σερβοκινητήρα στο Arduino.

```
BAS.attach(22);
```

```
SHL.attach(23);
```

```
ELB.attach(24);
```

```
WRI.attach(25);
```

```
WRO.attach(26);
```

```
GRI.attach(27);
```

```
servo_park(); //Καλεί την υποσυνάρτηση "Αρχική θέση".
```

```
Serial.begin( 115200 );
```

```
while (! Serial);
```

attachInterrupt(0, emstop_ISR, RISING); // Όταν το button interrupt αφήνεται, καλείται η υποσυνάρτηση emstop_ISR.

```
Wire.begin(); // Ενώνει το I2C BUS με την διεύθυνση του 0x52 του nunchuck
```

Wire.beginTransmission(WII_NUNCHUK_I2C_ADDRESS); //Εκπέμπει στην συσκευή 0x52

```
Wire.write(0xF0);
```

```
Wire.write(0x55);
```

```
Wire.endTransmission(); // Σταματά την μετάδοση
```

```

Wire.beginTransmission(WII_NUNCHUK_I2C_ADDRESS);

Wire.write(0xFB);

Wire.write(0x00);    //Στέλνει 0 memory address

Wire.endTransmission();

delay( 100 );}

void loop(){          //Η κυρία συνάρτηση του προγράμματος που τρέχει σε
απεριόριστο χρόνο.

    char key = keypad.getKey();    //Η μεταβλητή key παίρνει την τιμή του button που
έχουμε πατήσει κάθε φορά.

    if (key){          //Ελέγχει την τιμή που έχει δοθεί στην μεταβλητή "key".

        switch (key){    // Επιβεβαιώνει ότι κάποιο button όντος έχει πατηθεί. Μπορεί
να γραφτεί και ως: (key != NO_KEY)

            if (startprogram == true){

                case 'A': // Η περίπτωση που πατείτε το button A πραγματοποιεί τις παρακάτω
κινήσεις μέχρι την πρώτη εντολή break{ που συναντάει ο κώδικας.

                    while (emstop == false){

                        nunchuk.update();

Wire.requestFrom(WII_NUNCHUK_I2C_ADDRESS, 6); //Ζητα δεδομένα απο
nunchuck

                        counter = 0;

                        while (Wire.available()){

                            data[counter++] = Wire.read();}    //δεχεται byte σαν integer

Wire.beginTransmission(WII_NUNCHUK_I2C_ADDRESS);

Wire.write(0x00);

Wire.endTransmission();

                        if(counter >= 5){

```

```
double nunchuckaccelX = ((data[2] << 2) + ((data[5] >> 2) & 0x03) - ZEROX);//
```

Υπολογισμός συντεταγμένων για τους άξονες X ,Y,Z

```
double nunchuckaccelY = ((data[3] << 2) + ((data[5] >> 4) & 0x03) - ZEROY);
```

```
double nunchuckaccelZ = ((data[4] << 2) + ((data[5] >> 6) & 0x03) - ZEROZ);
```

```
if (nunchuk.cButton ==! 1){ //συνθηκη για την κινιση των servo BAS και
```

SHL

```
if (nunchuk.zButton ==! 1){
```

```
int value = constrain(nunchuckaccelX, -180, 180);
```

```
value = map(value, -180, 180, 0, 180); //Δημιουργεί συντεταγμένων σε μοίρες
```

```
BAS.write((180-value),70);
```

```
int value1 = constrain(nunchuckaccelY, -180, 180);
```

```
value1 = map(value1, -180, 180, 0, 180);
```

```
SHL.write((180-value1),70);
```

```
delay(80);}}
```

```
if (nunchuk.cButton == 1) {
```

```
if (nunchuk.zButton ==! 1){
```

```
int value2 = constrain(nunchuckaccelX, -180, 180);
```

```
value2 = map(value2, -180, 180, 0, 180);
```

```
WRO.write((180-value2),70);
```

```
int value3 = constrain(nunchuckaccelY, -180, 180);
```

```
value3 = map(value3, -180, 180, 0, 180);
```

```
ELB.write(-(180-value3),70);
```

```
delay(80);}}
```

```
if (nunchuk.cButton ==! 1) {
```

```
if (nunchuk.zButton == 1){
```

```
int value4 = constrain(nunchuckaccelX, -180, 180);
```

```
value4 = map(value4, -180, 180, 0, 180);  
GRI.write((180-value4),70);  
  
int value5 = constrain(nunchuckaccelY, -180, 180);  
value5 = map(value5, -180, 180, 0, 180);  
WRI.write((180-value5),70);  
  
delay(80);}}}}  
  
break;  
  
case '1':  
circle(); // Καλεί την υποσυνάρτηση circle().  
break;  
  
case '2':  
heart();  
break;  
  
case '3':  
circlezaxis();  
break;  
  
case '4':  
heartzaxis();  
break;  
  
case '5':  
spiral();  
break;  
  
delay(100);}}  
  
servo_park();}} // Επιστροφή στην αρχική θέση.
```

Οι θέσεις του βραχίονα χρησιμοποιώντας την αντίστροφη κινηματική. Το z είναι το ύψος, το y είναι η απόσταση από το κέντρο της βάσης, το x είναι τα ακριανά σημεία (σημείο

προς σημείο). Τα y, z μπορούν να είναι μόνο θετικά. Το $angle$ δηλώνει την γωνία της αρπάγης σε σχέση με το έδαφος. Όταν το $angle$ είναι 0, τότε η αρπάγη είναι παράλληλη με το έδαφος. Το παρακάτω κομμάτι του κώδικα, αποτελείται από τις εξισώσεις της αντίστροφης κινηματικής στην γλώσσα του Arduino. Η θεωρία της αντίστροφης κινηματικής χρησιμοποιείται στην σύγχρονη ρομποτική καθώς διευκολύνει τον έλεγχο του ρομποτικού συστήματος. Ο χειριστής χρειάζεται μόνο να δώσει συντεταγμένες στον χώρο κατάστασης του συστήματος, με αποτέλεσμα το μηχάνημα να "αντιλαμβάνεται" τον πιο άμεσο τρόπο για να εκτελέσει την εντολή που του δόθηκε. Επειδή όμως αποτελείτε από μια σειρά εξισώσεων και αλγορίθμων, υπάρχει μεγάλη πιθανότητα σφάλματος λογικής, αλλά με την σωστή μελέτη του χώρου κατάστασης, των περιορισμών και των δυνατοτήτων του συστήματος, το σφάλμα λογικής ελαττώνετε σημαντικά. (βλέπε [21])

```
void set_arm( float x, float y, float z, float grip_angle_d ){
    float grip_angle_r = radians( grip_angle_d );
    float bas_angle_r = atan2( x, y );
    float rdist = sqrt(( x * x ) + ( y * y ));
    y = rdist;
    float grip_off_z = ( sin( grip_angle_r )) * GRIPPER;
    float grip_off_y = ( cos( grip_angle_r )) * GRIPPER;
    float wrist_z = ( z - grip_off_z ) - BASE_HGT;
    float wrist_y = y - grip_off_y;
    float s_w = ( wrist_z * wrist_z ) + ( wrist_y * wrist_y );
    float s_w_sqrt = sqrt( s_w );
    float a1 = atan2( wrist_z, wrist_y );
    float a2 = acos((( hum_sq - uln_sq ) + s_w ) / ( 2 * HUMERUS * s_w_sqrt ));
    float shl_angle_r = a1 + a2;
    float shl_angle_d = degrees( shl_angle_r );
    float elb_angle_r = acos(( hum_sq + uln_sq - s_w ) / ( 2 * HUMERUS * ULNA ));
    float elb_angle_d = degrees( elb_angle_r );
```

```
float elb_angle_dn = -( 180.0 - elb_angle_d );
float wri_angle_d = ( grip_angle_d - elb_angle_dn ) - shl_angle_d; //(-20)
```

Παρακάτω φαίνονται οι συντεταγμένες που υπολόγισε ο κώδικας, μαζί με το σφάλμα κίνησης του εκάστοτε σερβοκινητήρα.

```
BAS.write(90- degrees(bas_angle_r));
SHL.write(shl_angle_d* 0.88888);
ELB.write((180-elb_angle_d)*0.77777);
WRI.write((wri_angle_d)+90 * 1.33333);}
```

Εδώ φαίνονται οι υποσυναρτήσεις που χρησιμοποιούμε για το πρόγραμμά μας.

```
void servo_park(){ //Αρχική θέση του βραχίονα με μοίρες.
    BAS.write(90.00);
    SHL.write(150.00);
    ELB.write(140.00);
    WRI.write(90.00);
    WRO.write(30.00);
    GRI.write(0);
    return;}

void emstop_ISR(void){ //Παύση και Επανεκκίνηση της χειροκίνητης
    servo_park();
    emstop = !emstop; //Αναστρέφει την παρούσα κατάσταση της μεταβλητής emstop
    startprogram = !startprogram;
    servo_park();}

void circle(){ // Κύκλος σε άξονες χ-ψ.
    #define RADIUS 20.0
```

```

float angle = 0;

float xaxis,yaxis;

for( angle = 0.0; angle <= 360.0; angle += 1.00 ) {
    xaxis = RADIUS * sin( radians( angle )) + 200.00;
    yaxis = RADIUS * cos( radians( angle )) + 200.00;
    set_arm(xaxis, yaxis, 120, 0);
    delay( 10 );}

void heart(){ // Καρδιά σε άξονες χ-ψ.

#define RADIUSH 20.0

float xaxis,yaxis;

float varx;

for( float angle = 0.0; angle <= 180.0; angle += 1.00 ) {
    xaxis = RADIUS * sin( radians( angle )) + 200.00;
    yaxis = RADIUS * cos( radians( angle )) + 190.00;
    set_arm(xaxis , yaxis, 120, 0);
    delay( 10 );}

for( float angle = 0.0; angle <= 180.0; angle += 1.00 ) {
    xaxis = RADIUS * sin( radians( angle )) + 200.00;
    yaxis = RADIUS * cos( radians( angle )) + 150.00;
    set_arm(xaxis , yaxis, 120 , 0);
    delay( 10 );}

for(float varx=130; varx<170; varx +=1){
    xaxis = -1.5*varx + 395;
    yaxis = varx;
    set_arm (xaxis , yaxis, 120 , 0);
    delay(10);}

```



```
for(float varx=170; varx<=210; varx +=1){  
    xaxis = 1.5*varx - 115;  
    yaxis = varx;  
    set_arm (xaxis , yaxis, 120 , 0);  
    delay(10);}}  
  
void spiral(){  
#define RADIUS 20.0  
  
float angle = 0;  
float height = 0;  
float xaxis,yaxis,zaxis;  
  
for( angle = 0.0 , height = 0.0; angle <= (360.0 *5); height +=0.05 , angle += 1.00 ) {  
    xaxis = RADIUS * sin( radians( angle )) + 150.00;  
    yaxis = RADIUS * cos( radians( angle )) + 150.00;  
    zaxis = 50.0 + height;  
    set_arm(xaxis, yaxis, zaxis, -25);  
    delay( 10 );}}  
  
void circlezaxis(){ // Κύκλος σε άξονες ψ-ζ.  
#define RADIUS 20.0  
  
float angle = 0;  
float zaxis,yaxis;  
  
for( angle = 0.0; angle <= 360.0; angle += 1.00 ) {  
    zaxis = RADIUS * sin( radians( angle )) + 150.00;  
    yaxis = RADIUS * cos( radians( angle )) + 170.00;  
    set_arm(0, yaxis, zaxis, -25);  
    delay( 10 );}}  
  
void heartzaxis(){ // Καρδιά σε άξονες ψ-ζ.
```

```
#define RADIUSH 30.0

float angle = 360.0;

float zaxis,yaxis;

float varx;

for( float angle = 0.0; angle <= 180.0; angle += 1.00 ) {

    zaxis = RADIUS * sin( radians( angle )) + 150.00;

    yaxis = RADIUS * cos( radians( angle )) + 168.00;

    set_arm(0 , yaxis, zaxis, -25);

    delay( 10 );}

for( float angle = 0.0; angle <= 180.0; angle += 1.00 ) {

    zaxis = RADIUS * sin( radians( angle )) + 150.00;

    yaxis = RADIUS * cos( radians( angle )) + 108.00;

    set_arm(0 , yaxis, zaxis, -25);

    delay( 10 );}

for(float varx=78; varx<138; varx +=1){

    zaxis = -0.8333*varx + 215;

    yaxis = varx;

    set_arm (0 , yaxis, zaxis, -25);

    delay(10);}

for(float varx=138; varx<=198; varx +=1){

    zaxis = 0.8333*varx -15;

    yaxis = varx;

    set_arm (0 , yaxis, zaxis, -25);

    delay(10);}}
```

ΚΕΦΑΛΑΙΟ 5

Μελλοντική εξέλιξη

Η παρούσα πτυχιακή εργασία αποτελεί τον προθάλαμο για μια νέα πτυχή στην εκπαίδευση. Υπάρχουν αρκετές βελτιώσεις τις οποίες θα πρέπει να γίνουν προκειμένου ο συγκεκριμένος ρομποτικός βραχίονας, να ανταπεξέρχεται στις προσδοκίες τόσο των μαθητών όσο και των φοιτητών.

Μια πρώτη επέκταση που αποσκοπούμε να εφαρμόσουμε, είναι η χρήση ενός μεγαλύτερου ρομποτικού βραχίονα με περισσότερους βαθμούς ελευθερίας και άξονες κινήσεων, ώστε να μπορούν οι σερβοκινητήρες να κινούνται με κυκλικές κινήσεις (360 μοιρών) αντί για ημικυκλικές (180 μοιρών). Με αυτόν τον τρόπο θα μπορούν οι μαθητευόμενοι να διακρίνουν καλύτερα της κινήσεις του βραχίονα και να εξοικειώνετε σε μια μεγαλύτερη περιοχή λειτουργίας. Επίσης η χρήση μιας οθόνης εμφάνισης μηνυμάτων από τον κώδικα, που θα εμφανίζει μηνύματα σφαλμάτων, κατάστασης του βραχίονα κ.α., ώστε να μπορούν οι μαθητευόμενοι να άμεση επικοινωνία με τον τυποποιημένο κωδικά.

Ο κύριος επιθήματος στόχος είναι να μπορούν οι μαθητευόμενοι να προγραμματίζουν οι ίδιοι τον βραχίονα, με την εφαρμογή των νευρωνικών δικτύων και την χρήση του προγράμματος Matlab, χωρίς όμως να εμποδίζονται οι μνήμες του οι οποίες θα είναι συνεχώς σε προτεραιότητα εκτέλεσης των συγκεκριμένων εντολών. Έτσι εξασφαλίζετε η ασφάλεια τόσο των μαθητευόμενων όσο και του ιδίου του βραχίονα, και δίνει την δυνατότητα μιας πιο κατανοητής αντίληψη του προγράμματος του βραχίονα και των δυνατοτήτων του.

Ως μακροπρόθεσμο στόχο, έχουμε ορίσει την σύνδεση του ρομποτικού βραχίονα με ιδικά διαδραστικά προγράμματα, που θα αναφέρετε στην τριτοβάθμια εκπαίδευση συγκεκριμένα, και θα αποσκοπεί στο γεγονός οι μαθητευόμενοι, μέσα από ένα ευχάριστο και ασφαλής περιβάλλον να μελετούν και να επιχειρούν δυσκολότερες εφαρμογές.

Επίλογος

Είναι ξεκάθαρο ότι τα αυτοματοποιημένα συστήματα έχουν εισχωρήσει σε βάθος στην καθημερινότητά του σύγχρονου ανθρώπου. Διευκολύνουν τις εργασίες, αυξάνουν την ασφάλειά, επιταχύνουν τον ρυθμό παραγωγής, και συμμετέχουν σε μεγάλο βαθμό στην εξέλιξη του πολιτισμού. Με την πάροδο του χρόνου, οι αυτοματοποιημένες μηχανές έχουν συμβάλει στην ανάπτυξη τις υγείας, τις βιομηχανίας, της ψυχαγωγίας, και δεν θα πρέπει η παιδεία να αποτελεί εξαίρεση. Προκειμένου να εξασφαλίσουμε όμως την συνεχή

τεχνολογική ανάπτυξη, καθιστά τε αναγκαίο, από τα νέα άτομα να αφομοιώσουν πλήρως την τεχνογνωσία, να κατανοήσουν την αλληλεπίδραση των θετικών επιστημών και με μια νέα οπτική, να συνδράμουν και εκείνοι με τον τρόπο τους στην βελτίωση της ζωής μας.

Παράρτημα

1. Introduction

Robots operating in laboratory environments during the 90's are still working on the field and research is based on them in the manufacturing process. The programming part of for those devices is the most difficult part that comes from use limitations although they are still in operation owing to their control interfaces. During the past decades robotic industries pay attention to reliable constructions with modern interface that are friendly to the user in order to operate the robot. These robots provide many options in general use rather than perform more applications like, welding, collecting masses and much more. The main dependency for all robots is a controller connected to a PC that the user can program for any work needed though the environment like DOS restrict the interfacing dexterities. Moreover, the operator consists of high torque function to upper electrical parts like servo and DC motors, which secures that the degrees of freedom the construction requires will be met. Devices for electricity safety and extensive wiring are used to connect all electrical equipment with mechanical parts together to each other. The close loop systems include sensors that are throughout controllable for instance the SCORBOT ER-III. The robot operation in real time sends information from encoders to motors about the speed or the condition of each part of the robotic arm.

More productive and steady research have replaced the old one that uses control software. The following hardware uses the LabView program. With that program, users can manipulate the functions of the robot with more convenient and limber way. To prove the effectiveness of this software many experiments have been performed and the results will be presented in this paper with characterized examples. Below, the main interface control panel and the results of the system communication with the user will be extensively explained. This specific robot includes controllable electrical components that can contribute in modern day use. This research contains modern software technology of electrical equipment. LabView is a graphical program continuously upgraded since 1986 by National Instruments. It provides an efficient interface for engineering operations such as algorithms in a graphical environment, developing programs, interfacing computers with instruments or collecting data. In the past years it has been proven the capacity of that programming implement in many similar projects and researches as the one presented here. Anderson Pereira Correia (see [1]) and Ramirez Cortes J.M. (see [2]) performed using LabView for programming and controlling reports of autonomous robotic carriage. Presume upon of all features of LabView, researchers can achieve mass control of their autonomous robots, navigation and real time image transfer.

Some of the researchers have documented the different kinds between robots and manipulators with this transcendent software. Shobhita Ann Job (see [3]) proposal on control scheme for managing signals of a four axis arm for pick and place was based on LabView.

C. Chandra Mouli (see [4]) managed to control a heavy duty robotic manipulator with five degrees of freedom working with servo motors while Roland Szabo and Aurel Gonteam (see [5]) presented stereoscopic cameras to control a four axis robotic arm in a face recognition application. LabView is glorified by researchers as the elite available program to illustrate projects. Saeid Moslehpour (see [6]) managed to establish communication between SCORBOT ER-III and LabView. Based on the accomplishment of Saeid Moslehpour this project found literatures in considerations. The collaboration that expanded the automated efficiency and control capacity that supports the procedures for the system to user safety, is represented in the following chapters.

2. System Description

The ER-III is a robotic manipulator which consists of a generic format of the arm and a controller. This robotic manipulator is built of standard materials and electronics parts. Also it has been declared the winner due to his reputation of being safe and controlled properly.



Fig. 1. The Robotic Arm and the Complemented Controller.

The E-III is a robotic arm of 5 axis (degrees of freedom) which means that there 5 joints of control for handling .The more axis there is ,the more improvement in control and dexterity the robotic arm can demonstrate .The number 1 axis is the base which gives the ability of rotation to the robotic arm. The next axis is the shoulder which is similar to the third axis, the elbow. Next is the fourth axis, the wrist which is similar to that of a human wrist

and that's why is described as two axis. Finally, the last axis, the fifth is twisting the wrist, despite the fact that wrist mechanical part mimics the human wrist, this rotation move for some robotics arms can be a full 360 degrees or more which is impossible for the human wrist. Lastly, the ER-III is equipped with a pair of grippers that are capable of been opened or closed, also grippers are not considered as an axis movement but have the same mechanism theory.

In this mechanic contraction there is also a black box the control which is responsible for the control of the robotic arm. This black box contains electronic parts that give the ability to interface between the computer software and the robotic arm. In the surface of the electronic box there are switches for testing the motors and lights. The computer software which controls the ER-III is a DOS based program. The name of the software program is 'SCORBASE' which is turned on whenever the robotic arm is connected to the PC ,and will automatically load and the main programming window will appear like in figure 2.

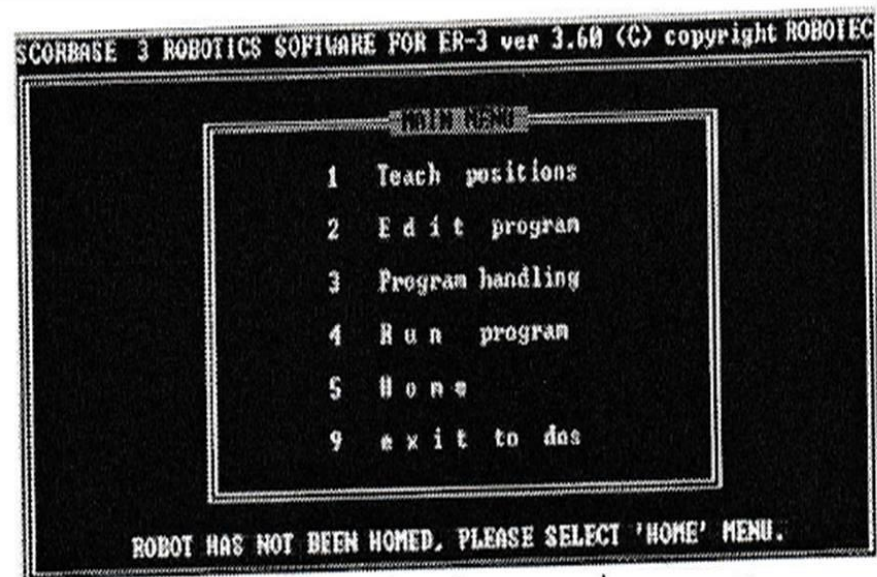


Fig. 2. The DOS Based Programming Interface.

It would be good to state the motivation of this project, and this arm is designed to introduce a system of a user friendly interface with many more capabilities and improvements like, Lab View.

3. The Interface Environment

Inspection and transport operations are expanding manufacture robots that require flexible, precise and sensitive part handling. Automated, industrial or laboratory applications use robotic arms to achieve smarter and higher implementation control. That allows robots to

change components and equipment to handle the random changing conditions that interact with the environment of all the safely operations for the user. Upper robots needed those applications that can interact with their work space through feedback signals and data. The graphical system NI LabView enables new functions for manufacture or educational robots by unify feedback, human machine interfaces (HMIs) and control data into one to create a handful environment for the user. It was no easy to complete this project with such software that communicates with the standard controller of the SCORBOT ER-III arm via the available RS-232 port. This project starts up from primary movements of the operator and upgrading the LabView code and introduce complex work of the robotic motion. Consequently, the buttons for right – left rotation movement of the robot base and a serial communication port installation supplied from the primary interface format of figure 3. Also, for basic open loop control it has been included a step and a speed selection regulator.

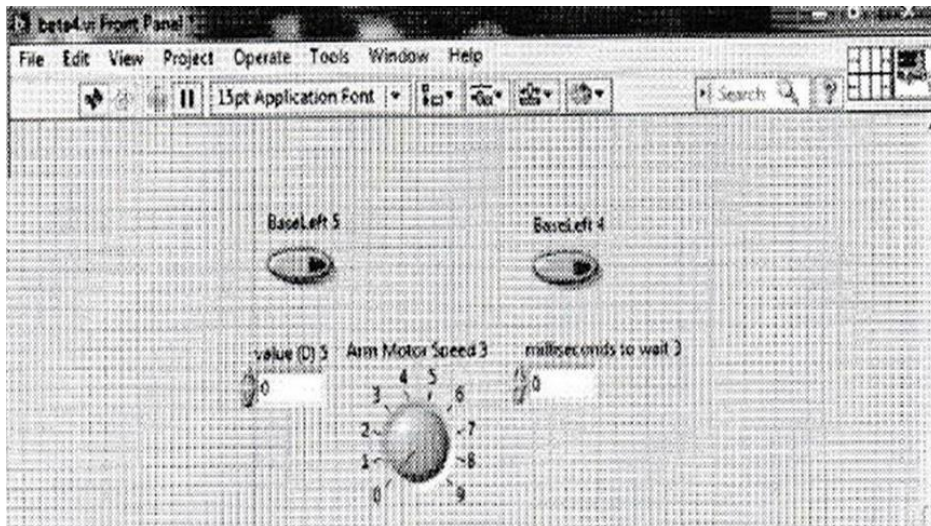


Fig. 3. The Interface Window for Basic Robot Handlings.

The figure 4 presents the first application that is supported by the code, where toolboxes, commands and libraries are used to built routines and loops of the code.

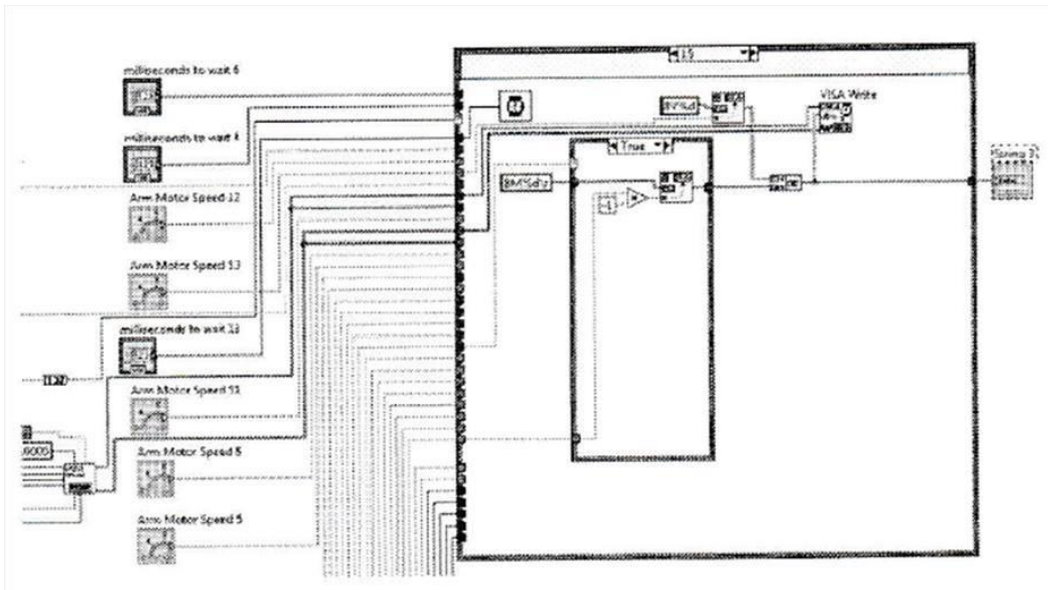


Fig. 5. The LabView Code For Controlling the Speed of 8 Motors.

SCORBOT ER-III has a pre-programmed default command by the constructor which is called “home position” and it applied on every robot industrial. This command is used for storage and position for the arm purposes. In figure 6 shows another part of the LabView code for executing all the above tasks and focused on the home position command. The robotic arm stops its operation and returns to the home position once the top left button ‘home position’ is activated, in a non interrupted procedure.

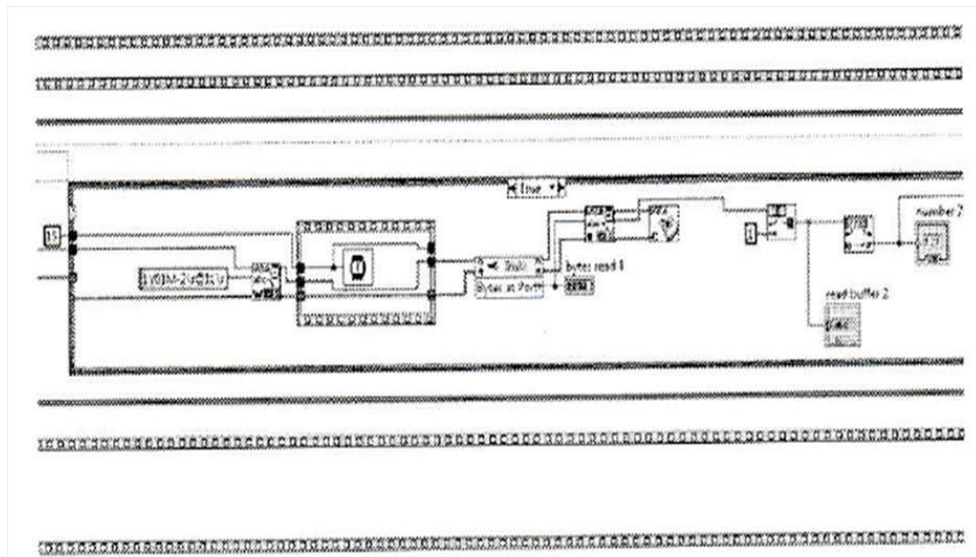


Fig. 6. The LabView Code for Robot’s Home Position Command.

The eight regulators set the step and the speed for all the eight motors. The seven servo motors move the robotic arm and the eighth operates the regional conveyor. The following figure shown the layout of the main interface panel that the user can control the robotic arm.

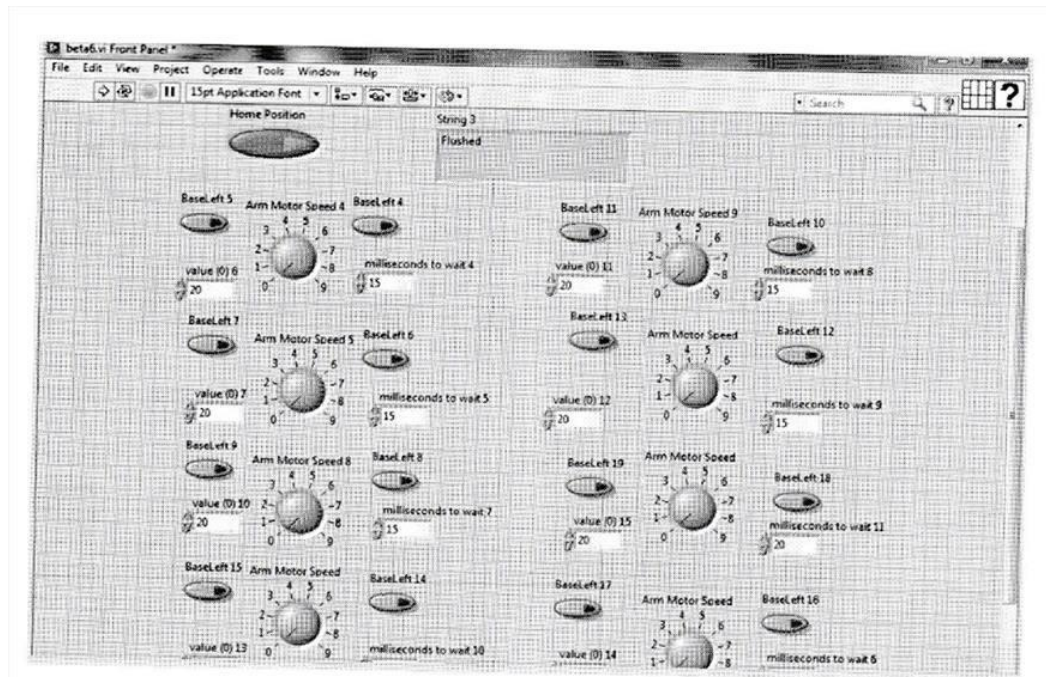


Fig. 7. The Programming Interface Panel.

There is a monitor at the top of the panel for serial port recordings. On the top left side there is the “home position” button. Much information can be provided from position limit switches mounted about the robot’s location and they compose input data to the controller.

It must be taken under serious consideration to the part that embodies safety precautions for all kind of robots. If there are random obstructions or any damage of the robot while it’s working, it could cause a serious accident to the user or the possible goods that the arm transfers. In an attempt to avoid that possibility the code was designed and performed in an automated process. To force the arm stands still there is an emergency stop that stops the data transmission and the monitor shows a warning message when the interface button is pressed. It is up to user to decide to return the robot to its home position or to terminate the process. The status control of all limit switches of the robotic arm represented from a case loop inside a while loop, in figure 6 above. Moreover, a part of the code was compiled for reading all the data that comes from the program and ‘freezes’ the robot in case an encoder stop is activated. In figure 8 is illustrated a case of safety operation from experimenting with this project.

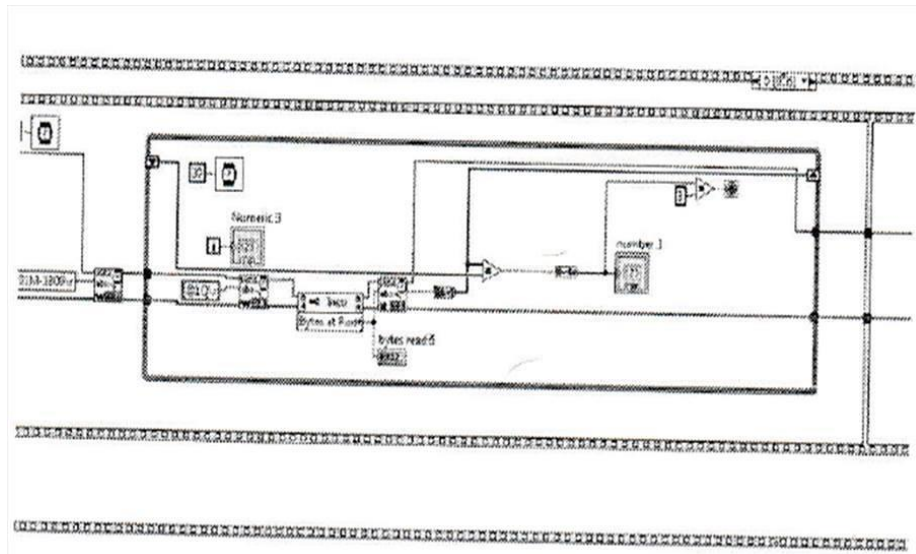


Fig. 8. The Code to Ensure the Safety of the Robot Operation.

To complete the safety precaution process, clockwise and anticlockwise rotation trends of the encoder were inclusive in the code below in figure 9. With that way any interruptions or crashes of the arm operation will not cause any damage to the user.

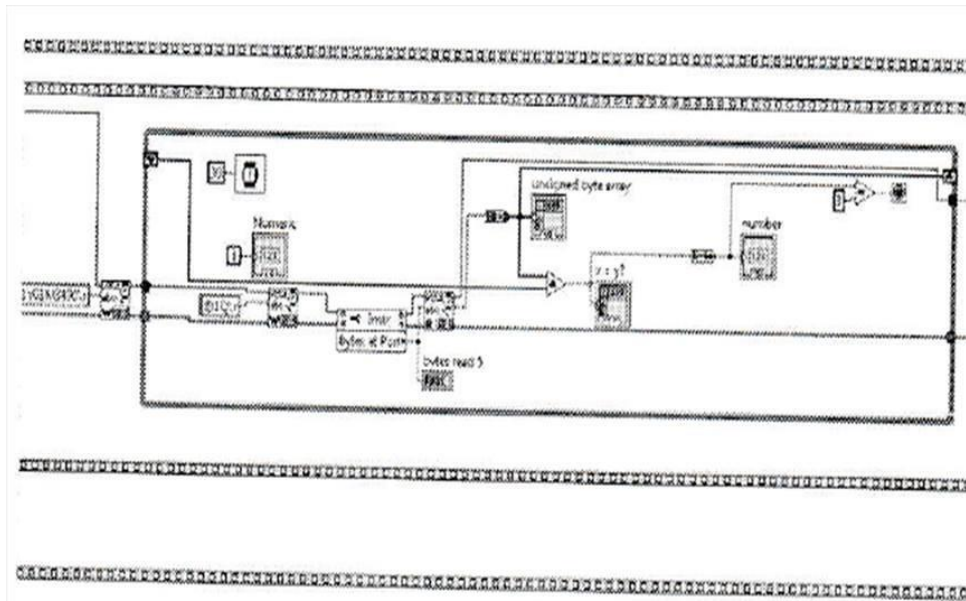


Fig. 9. The Code to Ensure the Safety of the Robot Operation.

The LabView never produced this kind of safety precautions, as authors are aware of, thus expanding the software control of the robot is more productive. Many operations are designed in this interface like, transferring obstacles, pick and place and much more. To accomplish that a researcher have to compile more code that do not fit in the project cause of

their big size. The robotic arm responded satisfactory and the interface used can accomplish the function with overall ideas.

4. Conclusion and Further Work

Some conclusions can now be stated after many months of continuous experimentation with the plant. Repeatability and accuracy in all modes of operation have been considered in high standards because they were key factors that could determine the efficiency of the method proposed in this paper. This modern interface design is making the programming process much more simple for the user and it is considered to ensure better results than the older DOS based software of the SCORBOT ER-III. In consideration of safety, for the first time of the current robot's operation with this LabView interface there has been an important improvement. This has been seen earlier in details, and produces a significant advantage of this programming method in comparison to any other.

The safety consideration will gain a separate interface window and will be programmed to memorize the movements of all robot axis allowing it to expand more in a few years. That will allow the user to continue the operation from the point the robot was stopped and finish it, rather than cancel it. A interesting improvement could be the adoption of sensors around the arm to multiply the feedback signals in the interface. With the help of Lab View this project can manage these improvements and provide the user with more data of the robot's position in real time. Lastly, the great advantage of this project is that it can benefit from the modern programming software.

Βιβλιογραφία

- [1] Άρθρο “Ρομπότ. Η λέξη που προέκυψε από την καταναγκαστική εργασία και τους σκλάβους. Την επινόησε ένας θεατρικός συγγραφέας το 1921, που προειδοποίησε για την εξέλιξη των μηχανών και πολέμησε τον ναζισμό.” 27/12/2014. Προσπέλαση στις 8/3/2015. Υπερσύνδεσμος: <http://www.mixanitouxronou.gr/rompot-i-lexi-pou-proekipse-apo-tin-katanagkastiki-ergasia-ke-tous-sklavous-tin-epinoise-enas-theatrikos-singrafeas-to-1921-pou-proidopiise-gia-tin-exelixa-ton-michanon-ke-polemise-to-nazismo/>
- [2] Προσπέλαση στις 8/3/2015. Υπερσύνδεσμος: https://el.wikipedia.org/wiki/%CE%92%CE%B9%CE%BF%CE%BC%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CE%BA%CE%AC_%CF%81%CE%BF%CE%BC%CF%80%CF%8C%CF%84
- [3] Προσπέλαση στις 8/3/2015. Υπερσύνδεσμος: https://el.wikipedia.org/wiki/%CE%95%CE%BA%CF%80%CE%B1%CE%B9%CE%B4%CE%B5%CF%85%CF%84%CE%B9%CE%BA%CE%AE_%CF%84%CE%B5%CF%87%CE%BD%CE%BF%CE%BB%CE%BF%CE%B3%CE%AF%CE%B1
- [4] Προσπέλαση στις 15/5/2015. Υπερσύνδεσμος: <http://www.plant-management.gr/index.php?id=14943>
- [5] Άρθρο "Πραγματικές λύσεις αυτοματισμού με ρομπότ". Κωνσταντίνος Ταχτσίδης και Μαίρη Ευθυμιάδου. Προσπέλαση στις 23/9/2015. Υπερσύνδεσμος: <http://www.supply-chain.gr/articles.php?artic=699#.VrCFtbKLTIW>
- [6] Προσπέλαση στις 12/8/2015. Υπερσύνδεσμος: <https://www.urmc.rochester.edu/pharmacy/about/safety/robotics.aspx>
- [7] Προσπέλαση στις 19/7/2015. Υπερσύνδεσμος: <http://urology-athens.gr/%CE%B5%CF%80%CE%B5%CE%BC%CE%B2%CE%AC%CF%83%CE%B5%CE%B9%CF%82/32-%CF%81%CE%BF%CE%BC%CF%80%CE%BF%CF%84%CE%B9%CE%BA%CE%AE-%CF%87%CE%B5%CE%B9%CF%81%CE%BF%CF%85%CF%81%CE%B3%CE%B9%CE%BA%CE%AE.html>
- [8] Προσπέλαση στις 19/7/2015. Υπερσύνδεσμος: <http://www.chessplayingrobot.com/>

- [9] Προσπέλαση στις 3/12/2014. Υπερσύνδεσμος: http://www.dfrobot.com/index.php?route=product/product&product_id=192#.VrC2OLKLTIW
- [10] Προσπέλαση στις 10/11/2015. Υπερσύνδεσμος: <http://www.robot-electronics.co.uk/i2c-tutorial>
- [11] Minikits. 2/10/2014. Σελίδα 2. Προσπέλαση στις 18/4/2015. Υπερσύνδεσμος: <https://www.minikits.com.au/doc/Keypad-Data.pdf>
- [12] Προσπέλαση στις 16/9/2015. Υπερσύνδεσμος: <https://www.arduino.cc/en/Main/ArduinoBoardMegaADK>
- [13] Jason P. Smith. Προσπέλαση στις 1/8/2015. Υπερσύνδεσμος: <http://makezine.com/projects/computer-power-supply-to-bench-power-supply-adapter/>
- [14] FAIRCHILD. Σεπτέμβριος 2014. Σελίδα 20. Προσπέλαση στις 6/8/2015. Υπερσύνδεσμος: <https://www.fairchildsemi.com/datasheets/lm/LM7805.pdf>
- [15]] Προσπέλαση στις 28/12/2015. Υπερσύνδεσμος: https://www.youtube.com/watch?v=TK0jrd_dYrg
- [16] Προσπέλαση στις 3/5/2015. Υπερσύνδεσμος: <https://github.com/netlabtoolkit/VarSpeedServo>
- [17] Προσπέλαση στις 17/3/2015. Υπερσύνδεσμος: <http://playground.arduino.cc/Code/Keypad>
- [18] Προσπέλαση στις 22/11/2015. Υπερσύνδεσμος: <https://github.com/GabrielBianconi/ArduinoNunchuk>
- [19] Προσπέλαση στις 23/11/2015. Υπερσύνδεσμος: <http://arduinoarts.com/2014/07/tutorial-wii-nunchuck-pan-tilt-servo/>
- [20] Προσπέλαση στις 25/3/2015. Υπερσύνδεσμος: <http://www.codeproject.com/Articles/710475/Using-matrix-keypad-with-Arduino>
- [21] Προσπέλαση στις 11/7/2015. Υπερσύνδεσμος: <https://www.circuitsathome.com/mcu/robotic-arm-inverse-kinematics-on-arduino>

Λίστα Εικόνων

Εικόνα 2.1. Διάταξη Ρομποτικού Βραχίονα Συγκολλήσεων.....	11
Εικόνα 2.2. Διάταξη Ρομποτικού Βραχίονα Παλετοποίησης Φαρμακοβιομηχανίας.....	13
Εικόνα 2.3. Ρομποτικός Συσκευαστής Φαρμακείου.....	14
Εικόνα 2.4. Ο Εξοπλισμός της Ρομποτικής Χειρουργικής.....	15
Εικόνα 2.5. Ρομποτικός Σκακιστής.....	16
Εικόνα 3.1. Καλωδιώσεις Nunchuck.....	18
Εικόνα 3.2. Ο Ρομποτικός Βραχίονας με το Χειριστήριο.....	19
Εικόνα 3.3. Αριθμητικό Πληκτρολόγιο (Keypad).....	19
Εικόνα 3.4. Το Keypad Συνδεδεμένο με τον Μικροεπεξεργαστή.	20
Εικόνα 3.5. Arduino Mega ADK.....	21
Εικόνα 3.6. Ο Μικροεπεξεργαστή και η Συνδεσμολογίες του.	21
Εικόνα 3.7. Συνδεσμολογία Εντός του Κουτιού.....	23
Εικόνα 3.8. Χρωματική Κωδικοποίηση Καλωδίων.....	23
Εικόνα 3.9. Συνδεσμολογία Εκτός του Κουτιού.	24
Εικόνα 3.10. Κυκλωμα Regulator.....	25
Εικόνα 3.11. Συνδεσμολογία του Regulator.	25
Εικόνα 3.12. Συνδεσμολογία του Button Παύσης και Εκκίνησης της Χειροκίνητης Λειτουργίας.	26

Τμήμα Αυτοματισμού

Σχολή Τεχνολογικών Εφαρμογών

Ανώτατο Εκπαιδευτικό Ίδρυμα Τεχνολογικού Τομέα (Α.Ε.Ι. Τ.Τ.) Πειραιά

Copyright © ΜΑΝΔΑΡΑΚΑ Ε. ΜΑΡΙΑ και ΣΦΑΚΙΑΝΟΣ Γ. ΙΩΑΝΝΗΣ

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας πτυχιακής εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για μη κερδοσκοπικό σκοπό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτή την πτυχιακή εργασία εκφράζουν τους συγγραφείς και δε πρέπει να ερμηνευτεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Τμήματος Αυτοματισμού της Σχολής Τεχνολογικών Εφαρμογών Α.Ε.Ι. Τ.Τ. Πειραιά.